Journal of Hydrology xxx (xxxx) xxx

Contents lists available at ScienceDirect

Journal of Hydrology

journal homepage: www.elsevier.com/locate/jhydrol

Research papers

Fusing stacked autoencoder and long short-term memory for regional multistep-ahead flood inundation forecasts

I-Feng Kao, Jia-Yi Liou, Meng-Hsin Lee, Fi-John Chang *

Department of Bioenvironmental Systems Engineering, National Taiwan University, Taipei 10617, Taiwan

ARTICLE INFO

This manuscript was handled by Andras Bardossy, Editor-in-Chief, with the assistance of Niels Schuetze, Associate Editor

Keywords:
Regional flood inundation
Stacked autoencoder (SAE)
Recurrent neural network (RNN)
Long short-term memory (LSTM)
Multistep-ahead forecast

ABSTRACT

Reliable and accurate regional multistep-ahead flood forecasts during extreme events are crucial and beneficial to flood disaster management and preparedness. Hydrologic uncertainty associated with the nonlinear dependence structure of flood inundation dynamics makes flood inundation forecasting fundamentally challenging. This study proposes a novel machine learning-based model (SAE-RNN) that hybrids the stacked autoencoder (SAE) with a recurrent neural network (RNN) for providing accurate and timely information to support emergency management in areas impacted by flood hazards. The proposed SAE-RNN model uses SAE to compress (encode) the high-dimensional flood inundation depths in a wide region into a low-dimensional latent space representation (flood features), uses RNN to forecast multistep-ahead flood features based on regional rainfall patterns, and finally uses SAE to reconstruct (decode) the multistep-ahead forecasts of flood features into regional flood inundation depths. A large number of hourly datasets of flood inundation depths collected in Yilan County of Taiwan formed the case study, where each dataset contains 169,797 grids of inundation depth. The datasets were divided into three independent datasets for use in training, validating and testing stages. The models' results showed that RMSE values were very small (<0.09 m) and R^2 values were high (>0.95) in all the cases (1- up to 3-hour-ahead forecasts in three stages). We conclude that the reason why the proposed SAE-RNN models are capable of attaining favorable regional multistep-ahead flood inundation forecasts could be owing to two core strategies: the effective continual extraction of the nonlinear dependence structure from flood inundation dynamics for lessening hydrologic uncertainty by virtue of SAE; and the nonlinear conversion of rainfall sequences into future flood features by virtue of RNN.

1. Introduction

The explosive growth in urban development has occurred worldwide over the last decades, where the transformation from rural to urban land use has produced a significant reduction in the absorption capacity of the watershed, causing an increase in runoff and producing high peak flows rapidly, especially during extreme events. An urban flash flood resulting from heavy rains and may endanger human lives and damage property as well as initiate a cascade of environmental and health impacts. Although flood control infrastructures have been steadily increased, flood damage has continued to increase dramatically due, in part, to the urban development in flood-prone areas. Flash floods have long been common in Asian cities with growing urbanization and extreme rainfall, which have driven increasingly severe and frequent flood events (Luo et al., 2018). To mitigate flood damages, emergency management authorities may rely on real-time flood forecast systems that provide sufficient lead time for evacuation and asset protection in urban watersheds during extreme rainfall events. However, the development of these systems is rather complicated due to spatio-temporal variations and uncertainty in rainfall distributions alongside complex rainfall–runoff relationships. Moreover, urban flood observation equipment is not installed pervasively such that the monitoring datasets of regional flood inundations are rare. Thus, regional flood inundation datasets under various hydro-metrological conditions could be obtained mainly from simulation models. Nevertheless, flood simulations are commonly conducted based on hydraulics theorems with given hydrological conditions and thus require a large amount of computation time, which cannot meet the needs of real-time flood forecasts. For a city undergoing fast urban development, its regional meteorological conditions would interact with intensive human activities and climate change, giving rise to the non-stationary process between rainfall and runoff, with features difficult to be captured by a static learning mechanism.

* Corresponding author.
E-mail address: changfj@ntu.edu.tw (F.-J. Chang).

https://doi.org/10.1016/j.jhydrol.2021.126371
Received 19 October 2020; Received in revised form 7 January 2021; Accepted 21 April 2021
Available online 26 April 2021

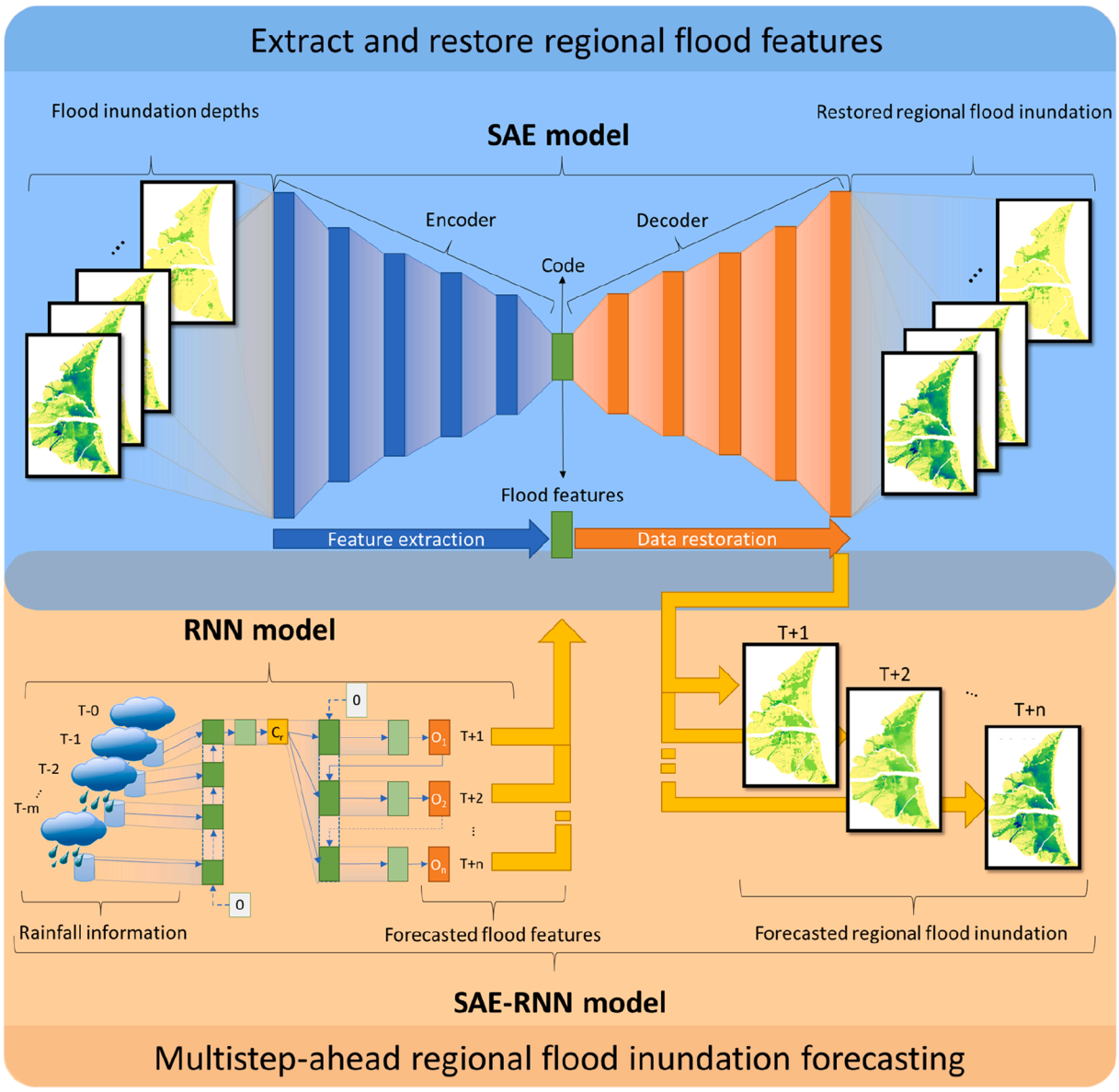
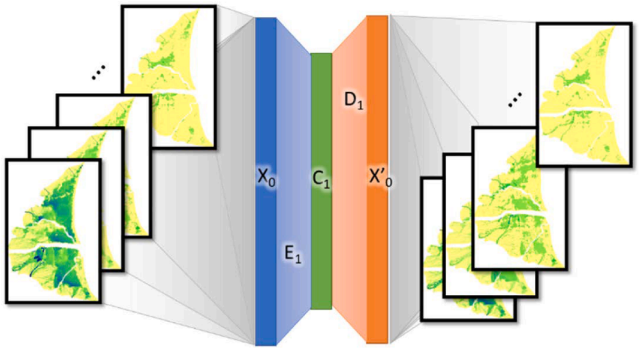


Fig. 1. Architectures of SAE-RNN model.

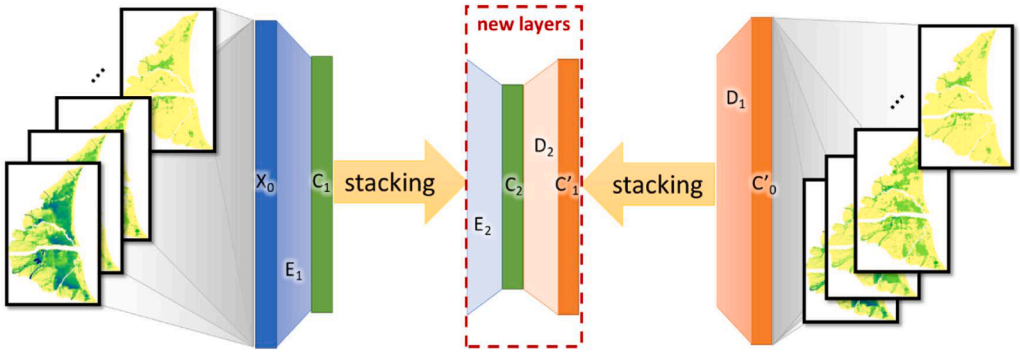
Therefore, regional flood inundation forecasting remains one of the most challenging tasks in hydrology.

Artificial neural networks (ANNs) have been developed rapidly in the field of data science over the last decades. ANNs can effectively simulate highly nonlinear and complex systems through learning their relationships between input and output patterns, without requiring the parameter settings of physical processes. ANNs have become a popular and useful tool for predicting various hydrological variables, such as precipitation (e.g., Shafaei et al., 2016; Valipour, 2016; Nanda et al., 2019) and regional flood inundation depths (e.g., Chang et al., 2014, 2018, 2019). Many studies in the past decade showed that ANNs could improve the accuracy and reliability of rainfall-runoff prediction (e.g., Badrzadeh et al., 2015; Chang, 2020; Chang and Tsai, 2016; Chen et al., 2013; Humphrey et al., 2016; Noori and Kalin, 2016; Nourani, 2017; Nourani et al., 2014; Puttinaovarat and Horkaew, 2020; Ren et al., 2019; Shoaib et al., 2018; Tan et al., 2018; Taormina et al., 2015; Tsai et al., 2014; Xie et al., 2019; Zhou et al., 2020). Deep learning neural networks utilize network architectures of multiple ($\geq$2) hidden layers to capture the inherent features of data layer-by-layer without prior knowledge and produce good performance in time series forecasting (Abbasi et al., 2020; Zhou et al., 2019). The volume, velocity, and variety of hydrological data keeps growing on account of the large-scale deployment of sensors, and sensing data have the potential to be transformed into actions to revolutionize hydrological fields owing to the availability of computational resources and the popularity of deep learning (Tong et al., 2018; Sit et al., 2020). Bai et al. (2019) combined the deep learning and recursive modeling for inflow forecasting and concluded that such combination could benefit the exploration of complex features in the inflow forecasting. Deep learning has also been applied to noise removal in recent years (Bi et al., 2019a, 2019b; Yu et al., 2019; Kidoh et al., 2020; Li et al., 2020).

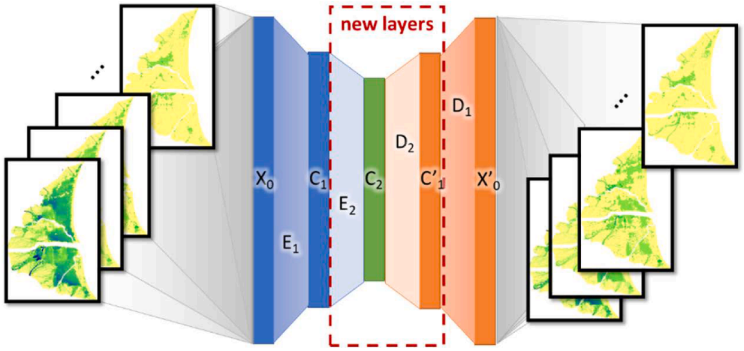
Autoencoder (AE) with an encoder-decoder framework is a type of neural networks for dimensionality reduction (Wang et al., 2016), feature detection and extraction (Zabalza et al., 2016; Liu et al., 2017a), and sequence-to-sequence prediction (Kao et al., 2020; Orland et al., 2019; Xiang et al., 2020a, 2020b) lately. The encoder of AE could largely reduce the encoding input dimension while the decoder of AE could reconstruct the reduced (encoded) representation into the output as close as possible to its original input. One key advantage of AE is that this model can extract useful features continuously during propagation



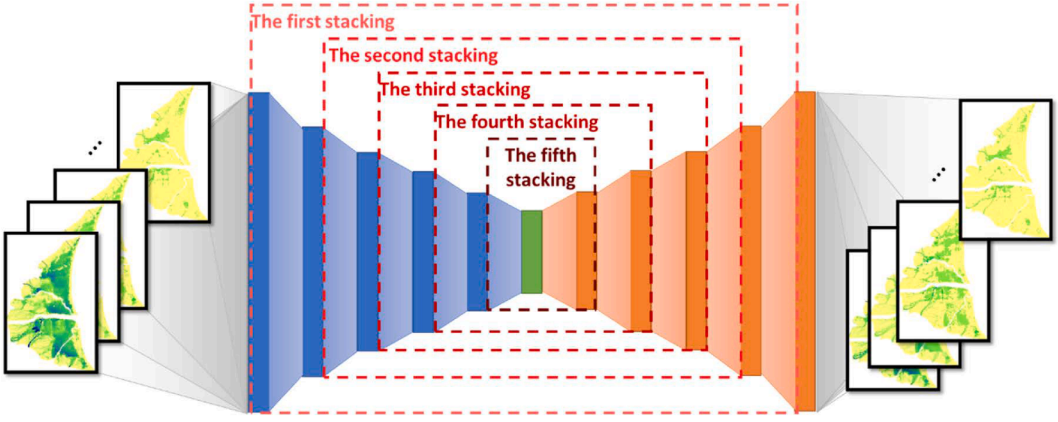
(a) The structure of SAE_1



(b) Stacking the encoder and decoder of SAE_1 on new layers.



(c) The structure of SAE_2.



(d) The structure of SAE_5.

Fig. 2. The process of building SAE in this study.

and filter out useless information (Liu et al., 2017b). With dimension reduction in input information, the efficiency of the model learning process of AE models can be increased. The encoder and decoder structure of AE is usually multi-layer and can be constructed by stacking, hence it is called "stacked autoencoder (SAE)". Li et al. (2016) used SAE to extract and learn useful features from a large number of data for better accommodating the characteristics of daily reservoir inflow. Jiao et al. (2018) proposed a model combining SAE and back propagation algorithm for the short-term wind power forecasting. Liu et al. (2019) adopted a stacked denoising autoencoder for short-term electric load forecasting. In this study, SAE is used to extract regional flood features and serves as a restorer for the multistep-ahead forecasts of regional flood inundation depths.

The long short-term memory (LSTM) configured with a recurrent neural network (RNN) architecture is a type of deep neural networks (DNNs) (Hochreiter and Schmidhuber, 1997). The LSTM is used to deal with the exploding and vanishing gradient problems that may occur when training RNNs with long-term lags. Recently, the LSTM has been implemented to explore its capability in time series forecasting of river flood (Le et al., 2019; Ding et al., 2020; Ni et al., 2020), low-flow time series forecasting (Sahoo et al., 2019), and water table depth (Zhang et al., 2018a, 2018b; Jeong and Park, 2019) as well as to learn long-term dependencies within hydrological catchments (Kratzert et al., 2019) and model rainfall-runoff processes (Sezen et al., 2019; Kao et al., 2020). For neural networks, the sequence-to-sequence learning trains a model by converting a sequence from one domain into another domain (Wiseman and Rush, 2016; Chiu et al., 2018). Sequence-to-sequence models embedded with LSTM have gained marvelous achievements in various fields, like anomaly detection (Fengming et al., 2017), image segmentation (Marmanis et al., 2018), video recognition (Zhu et al., 2017; Zhu and Zabaras, 2018), time series forecasting (Du et al., 2018; Zaytar and El Amrani, 2016; Zhang et al., 2018a,b; Zhou et al., 2019), and rainfall-runoff processes (Kratzert et al., 2018a, 2018b, 2019; Kao et al., 2020; Xiang et al., 2020a, 2020b).

The prosperous development of deep learning algorithms offers powerful tools to handle massive data, and deep learning methods often outperform conventional machine learning ones in many fields. Inspired by these, we intend to propose a deep learning based approach for regional multistep-ahead flood inundation forecasts. ANNs, SAE and DNNs have the merits to explore in-depth how regional rainfall sequences can be mapped onto urban flood inundation sequences for reliably and accurately making multistep-ahead regional flood inundation forecasts. In this study, we aim to develop a novel methodology that hybrids SAE with an LSTM-based RNN for constructing a regional multistep-ahead forecast model to provide timely inundation information in support of emergency management at areas threatened by flood hazards. The novelty of this study lies in the use of sequence-to-sequence SAE (continual feature extraction of flood inundation dynamics during propagation and dynamic filtering of useless information) and RNN (nonlinear conversion of rainfall sequences into future flood features) to map regional rainfall sequences onto urban flood inundation sequences. The remainder of this study is organized as follows. Section 2 presents the framework of the proposed methodology. Section 3 introduces the case study and materials. Section 4 presents the results and discussion of the methods applied to multistep-ahead regional flood forecasts. Conclusions are then drawn in Section 5.

2. Methodology

In this study, we propose a novel machine learning methodology (SAE-RNN) that integrates SAE with an LSTM-based RNN for accurately making regional multistep-ahead flood inundation forecasts. The framework of the proposed methodology is displayed in Fig. 1. The main merit of the methodology can be three-fold: (1) using SAE to sequentially compress the huge dimensional flood inundation depths (input datasets) into a low dimensional code (a vector of flood features) and then sequentially decompress and reconstruct the code into flood inundation depths that have the same dimension as input datasets; (2) using an LSTM-based RNN model to perform a nonlinear conversion of the input (rainfall sequences) into a specific number of rainfall features and then decode the features with the model's feedback of the previous flood features to predict future flood features; and (3) fusing the flood features predicted by the LSTM-based RNN model as the code of SAE to sequentially reconstruct multistep-ahead regional flood inundation forecasts. The methods adopted are described in the following sections.

2.1. Stacked autoencoder (SAE)

AE is an ANN that has two main components, encoder and decoder. The encoder is responsible for compressing the high dimensional input data into the low dimensional latent-space representation (code) while the decoder is responsible for reconstructing the code into the output. A sacked autoencoder (SAE) is a DNN with multiple hidden layers, where the encoding and decoding layers are typically pairwise symmetrical (Fig. 2). The SAE architecture with a one-layer encoder and a one-layer decoder is denoted as SAE_1, as shown in Fig. 2(a). The encoder, the code, and the decoder are defined as follows.

$$encoder_1 : R^{n_0} \rightarrow R^{n_1} \tag{1}$$

$$decoder_1 : R^{n_1} \rightarrow R^{n_0} \tag{2}$$

$$d, n \in R$$

$$C_1 = encoder_1(X_0) \tag{3}$$

$$X'_0 = SAE_1(X_0) = decoder_1(C_1) \tag{4}$$

$$X_0, X'_0 \in R^{n_0}, C_1 \in R^{n_1}$$

$$L(X_0, X'_0) = \|X_0 - X'_0\|^2 \tag{5}$$

where C_1 denotes an n_1-dimensional code (vector); and X_0 and X'_0 are the input and output vectors with dimension n_0, respectively. SAE is trained to minimize the reconstruction error, $L(X_0, X'_0)$. The training of SAE is performed through the backpropagation of the reconstruction error. In this study, the input vector contains regional flood inundation depths (high dimensional grid data) and the code produced by the encoder can be regarded as the feature values of regional flood inundation data.

The fully connected layer (FCL) is implemented in each neuron of all the layers, as shown below.

$$y_j^n = f\left(\sum_i w_{ij} y_i^{n-1} + b_j^n\right) \tag{6}$$

where y_j^n is the output of the jth neuron in Layer n; y_i^{n-1} is the output of the ith neuron in Layer n-1; w_{ij} is the link weight between y_i^{n-1} and y_j^n; b_j^n is the bias of the jth neuron in Layer n; and f is the Rectified Linear Unit (ReLU) function of the neuron. The ReLU function is a type of activation functions commonly used in DNNs, and its formula is shown below.

$$ReLU(x) = \max(0, x) \tag{7}$$

The value of the differential ReLU function falls within the range between 0 and 1, and therefore the ReLU function in AE can mitigate the gradient vanishing or exploding problems during network training.

We next add two new FCLs (Fig. 2(b)) to SAE_1 to form SAE_2 (Fig. 2(c)), where the trained weights of $encoder_1$ and $decoder_1$ are retained. The process of SAE_2 transforms then n_0-dimensional input into n_2-dimensional code and then reconstructs the transformed code into an n_0-dimensional output, shown as follows.

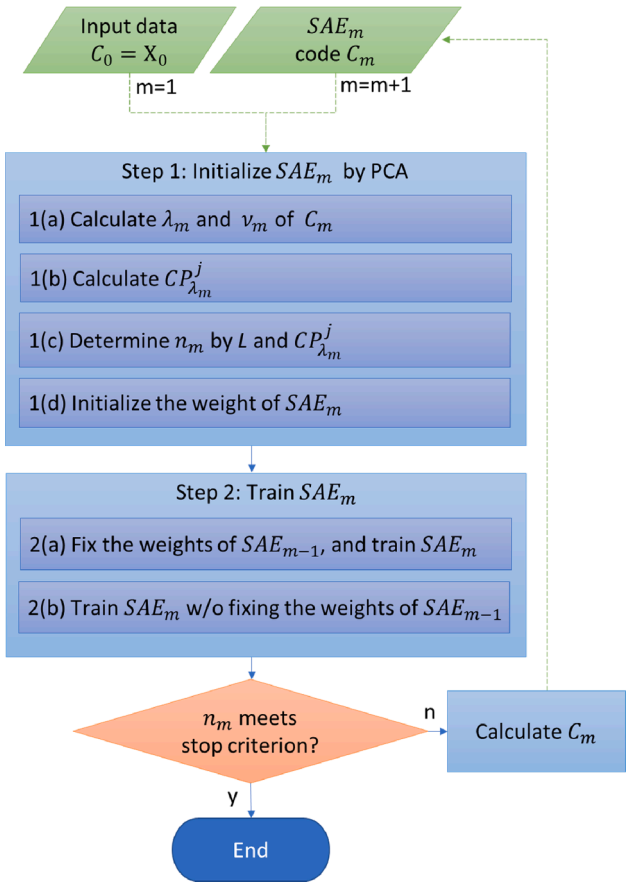


Fig. 3. Flow chart of training SAE using PCA results.

$$C_2 = encoder_2(encoder_1(X_0)) \tag{8}$$

$$SAE_2(X_0) = decoder_1(decoder_2(C_2)) \tag{9}$$

$$X_0 \in R^{n_0}, C_2 \in R^{n_2}$$

where $encoder_2$ and $decoder_2$ are both of one single FCL with initialization. C_2 denotes an n_2-dimensional code (vector).

The next is to expand the entire stacking process to the *mth* stacking of AE (SAE_m), as shown below.

$$C_m = encoder_m(encoder_{m-1}(\cdots encoder_1(X_0)\cdots)) \tag{10}$$

$$SAE_m(X_0) = decoder_1(decoder_2(\cdots decoder_m(C_m)\cdots)) \tag{11}$$

$$C_m \in R^{n_m}$$

$$n_0 > n_1 > \cdots > n_m \in R$$

where $encoder_m$ and $decoder_m$ are the newly added encoding and decoding layers in the stacking process, respectively. C_m denotes an n_m-dimensional code (vector) that is compressed by m encoders (from $encoder_1$ to $encoder_m$). The total number of layers in $SAE_m$is 2 m + 1 (an encoding part with m FCLs + a code with 1 FCL + a decoding part with m FCLs). The architecture of SAE_5 is shown in Fig. 2(d).

2.1.1. Principal component analysis (PCA)

In order to increase model effectiveness and stability when training the weights of SAE, the principal component analysis (PCA) is used to reduce the dimension of each code in the entire stacking process, where the eigenvector of PCA is implemented as the initial weights of the newly added layer ($encoder_m$ and $decoder_m$). The training of SAE involves weight initialization and weight adjustment (Fig. 3). The initialization step (Step 1) is divided into four sub-steps. Sub-step 1(a) uses PCA to calculate the eigenvalues (λ_m) and eigenvectors (v_m) of the previous code (C_{m-1}). If m is 1, C_{m-1} is the input vector (X_0). Sub-step 1(b) calculates $P^j_{\lambda_m}$ (the ratio of the *jth* eigenvalue to the sum of all eigenvalues), sorts the percentages in descending order, and calculates the cumulative percentage ($CP^j_{\lambda_m}$) of $P^j_{\lambda_m}$. $P^j_{\lambda_m}$ and $CP^j_{\lambda_m}$are as shown below.

$$P^j_{\lambda_m} = \frac{\lambda^j_m}{\sum_{k=1}^{n_m} \lambda^k_m} \tag{12}$$

$$CP^j_{\lambda_m} = \sum_{k=1}^{j} P^k_{\lambda_m} \tag{13}$$

where λ^j_m is the *j*th eigenvalue for the initialization of $encoder_m$ and $decoder_m$ in SAE_m.

Sub-step 1(c) determines a suitable dimension (n_m) for the code of SAE_m by comparing the preset value L and $CP^j_{\lambda_m}$, shown as follows.

$$n_m = j, \ when \ CP^j_{\lambda_m} > L, \ and \ j < m \tag{14}$$

Sub-step 1(d) initializes the weight values of $encoder_m$ and $decoder_m$.

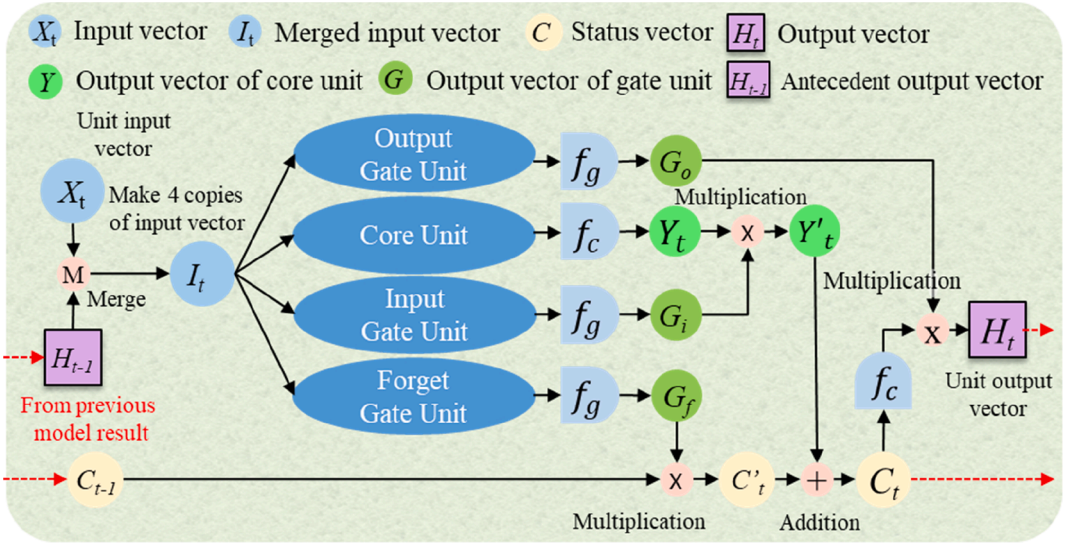


Fig. 4. Architecture of the LSTM unit.

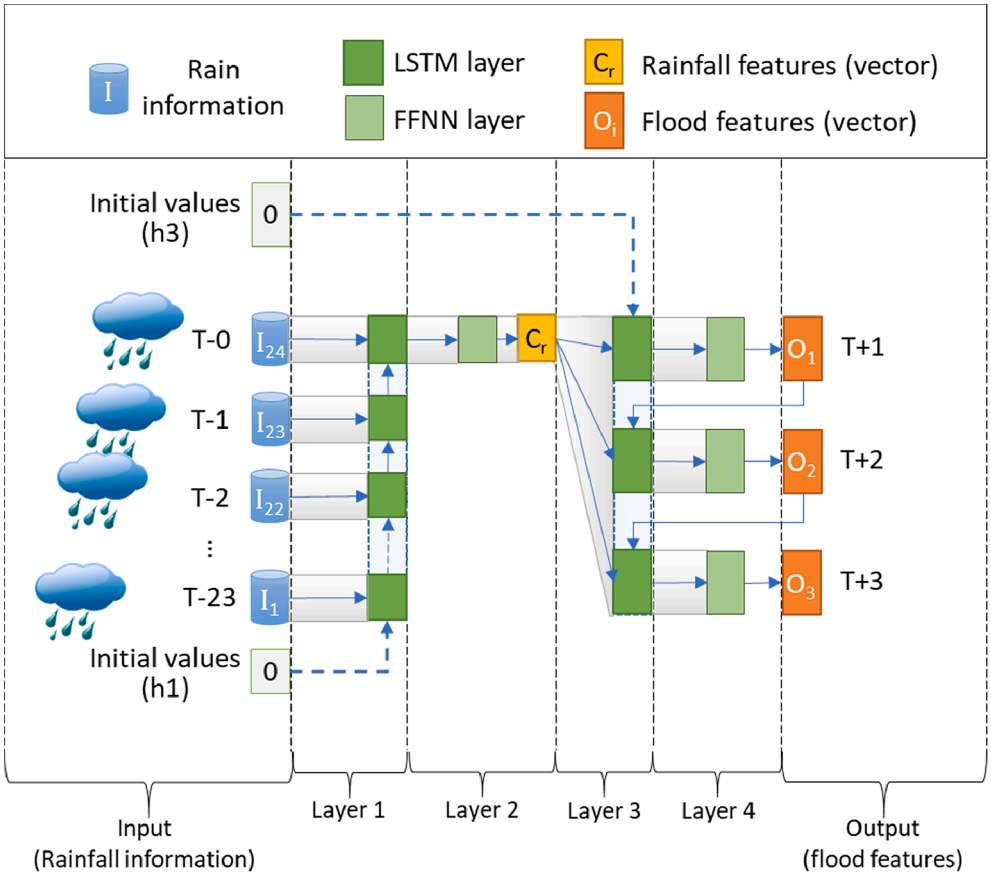


Fig. 5. Architectures of the LSTM-based RNN model.

The sum of the product of inputs and weights ($\sum_i w_{ij}x_i$) is equal either to the linear transformation equation if the initial bias (b_j^n) is set as 0 (Eq. (6)) or to the calculation of the *j*th principal component (p_j) in PCA, shown below.

$$p_j = \sum_i w_{ij}x_i \tag{15}$$

where x_i is the value of the *ith* coordinate in the original coordinate space; p_j is the value of the *jth* coordinate in the new coordinate space; and w_{ij} is the transform coefficient. Therefore, this study uses the first *j* eigenvalues as the initial weights, shown below.

$$w^{encoder_m} = \nu_{mj} \tag{16}$$

$$w^{decoder_m} = \nu_{mj}^T \tag{17}$$

$$n_{m-1} > n_m = j$$

where ν_{mj} is the vector spanned by the first *j*th rows of eigenvector ν_m; $w^{encoder_m}$ is the weights with $n_m \times n_{m-1}$ elements initialized by the eigenvector (ν_{mj}) for $encoder_m$; and $w^{decoder_m}$ is the weights with $n_{m-1} \times n_m$ elements initialized by the transpose of the eigenvector ($\nu_m^T j$) for $decoder_m$.

After Step 1, the structure of SAE_m is determined and its weight values are initialized. Step 2 will train SAE_m. Sub-step 2(a) fixes the weights from SAE_{m-1} and trains the weights of the newly added $encoder_m$ and $decoder_m$. Sub-step 2(b) trains all the weights of SAE_m to ensure the coverage and stabilization of weights. If the dimension (n_m) of C_m reaches the criterion, the training process of SAE_m will be terminated, otherwise return to Step 1 to continue the new training process of SAE_{m+1}.

2.2. LSTM-based RNN model

2.2.1. Long short-term memory (LSTM)

The LSTM unit has several architectures. A common architecture comprises a core unit (the memory part) and three gate units (input, output and forget gates) that direct the information flow inside the LSTM unit (Fig. 4). The computation steps are shown as follows.

$$I_t = H_{t-1} + X_t \tag{18}$$

$$Y_t = f_c(W_c \cdot I_t + b_c) \tag{19}$$

$$G_i = f_g(W_i \cdot I_t + b_i) \tag{20}$$

$$G_f = f_g(W_f \cdot I_t + b_f) \tag{21}$$

$$G_o = f_g(W_o \cdot I_t + b_o) \tag{22}$$

$$C_t = G_i \cdot Y_t + G_f \cdot C_{t-1} \tag{23}$$

$$H_t = f_c(C_t) \cdot G_o \tag{24}$$

where the subscript t denotes the time step; H_t is the output vector; I_t is the merged input vector that combines the antecedent output vector H_{t-1} with the input vector X_t; Y_t is the output vector of the core unit; G_i, G_f and G_o are the output vectors (gate values) obtained from the input, forget and output gate units, respectively; and C_t is the new cell state vector of long-term memory, which will return to the LSTM unit when being reused. The core unit activation function f_c is the Hyperbolic Tangent function, shown as follow.

$$f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{25}$$

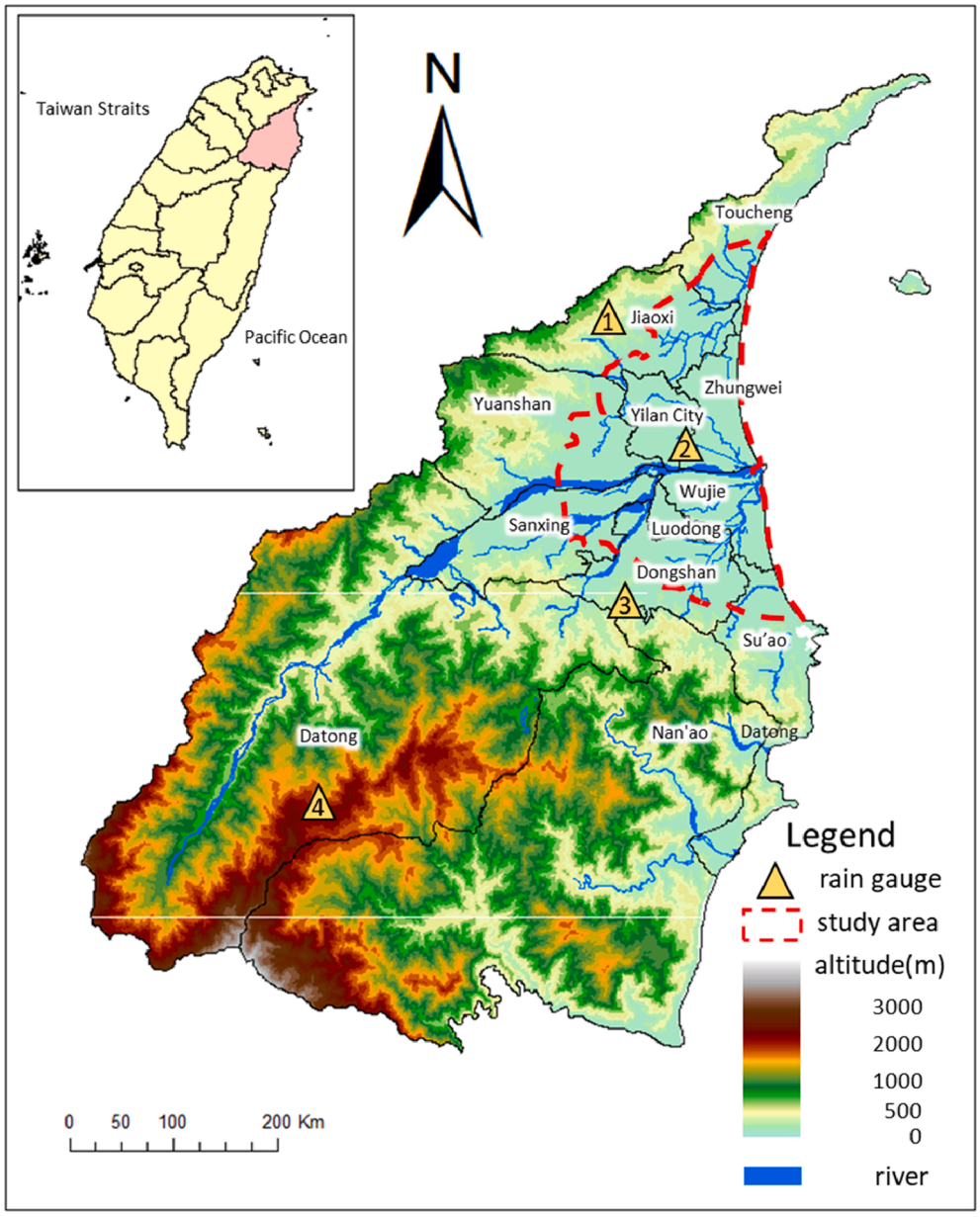


Fig. 6. Locations of Yilan County's inundation-prone region and nearby rain gauges.

The gate unit activation function f_g is the Hard Sigmoid function, shown as follow.

$$f(x) = \max[0, \min(1, 0.2x + 0.5)] \quad (26)$$

In addition, C_t is not transformed by f_c in this step such that it is much easier to keep the information of Y_t for the next reuse. In the first time step ($t = 0$), the initial values of H_0 and C_0 are zero vectors.

2.2.2. LSTM-based RNN models

The architecture of the LSTM-based RNN model has four layers, as shown in Fig. 5. The first layer consists of LSTM units, which read the input information of k-hour antecedent regional rainfall. Because LSTM has an algorithm to discard (forget) a part of the previous information, the LSTM-based RNN model can retain important rainfall features and abandon (neglect) useless rainfall information. The second layer consists of an FFNN layer, which performs a non-linear conversion of the output of the first layer into a low-dimensional rainfall features. The FFNN layer can increase the nonlinear fitting effect and the efficiency of the weight adjustment algorithm. The third layer implements another LSTM-based model to produce multistep-ahead forecasts. The inputs of this layer contain the output vector (rainfall features, C_r) from the second layer and the forecasted vector (flood features) at the previous time step (O_i). In other words, rainfall features are the main factor affecting the 1-hour-ahead forecast of flood features while the 2- and 3-hour-ahead forecasts will be adjusted using the 1-hour-ahead forecast and the 1- and 2-hour-ahead forecasts, respectively, which would avoid a complete consistency of the current result with the antecedent results. The fourth layer consists of an FFNN layer to make 1- to 3-hour-ahead outputs (flood features).

2.3. SAE-RNN model

As mentioned above, SAE firstly compresses a huge dimension of regional flood inundation depths into a low dimensional code of flood features and then sequentially decompresses and reconstructs flood features into regional flood inundation depths, which is mainly an encode-decode process. Then the LSTM-based RNN model estimates multistep-ahead flood features based on regional rainfall sequences.

Table 1
Rainfall information of 31 historical rainfall events.

Event name	24-h cumulative rainfall (mm)			
	Rain gauge No. 1	Rain gauge No. 2	Rain gauge No. 3	Rain gauge No. 4
Typhoon_1999-0605	186.5	172.5	277.5	24.0
Typhoon_2000-0708	56.1	68.6	89.0	21.5
Typhoon_2000-0821	54.0	43.5	74.0	64.0
Storm_2000-1029	46.1	31.6	0.1	17.5
Storm_2000-1103	218.5	194.5	187.5	53.5
Storm_2000-1111	72.5	70.5	154.5	12.5
Storm_2000-1213	131.5	147.5	164.0	31.5
Storm_2000-1219	132.1	27.6	120.0	28.0
Typhoon_2001-0511	133.0	150	134.0	5.5
Storm_2001-0923	80.0	18.5	23.0	35.0
Typhoon_2001-0924	102.6	144.0	327.0	53.5
Typhoon_2001-1015	22.5	10.6	10.1	10.5
Storm_2001-1208	147.1	41.1	27.1	77.0
Typhoon_2002-0702	53.1	76.6	26.1	90.0
Typhoon_2003-0819	96.5	2.1	55.5	77.0
Typhoon_2003-0831	136.1	124.5	87.5	25.0
Storm_2003-0910	41.0	26.5	36.5	35.0
Typhoon_2004-0607	110.1	223.5	301.0	43.0
Typhoon_2004-0823	0.6	0.6	9.0	13.5
Storm_2004-0907	24.1	69.1	59.6	34.0
Typhoon_2004-1024	74.1	98.1	68.5	23.5
Typhoon_2005-0716	27.1	53.5	145.5	48.0
Typhoon_2005-0830	45.0	57.0	59.5	3.0
Typhoon_2005-0930	91.6	91.6	57.6	3.0
Storm_2006-0709	330.5	140.0	84.0	7.0
Typhoon_2006-0914	22.1	32.6	42.6	6.5
Typhoon_2007-0806	56.0	35.0	20.0	5.5
Typhoon_2007-0816	61.5	58.5	107.0	4.5
Storm_2007-1013	148.0	114.5	169.5	9.0
Storm_2007-1105	32.1	32.1	263.6	6.5

That is to say, based on a large number of regional flood inundation depths generated from various rainfall patterns, we want to demonstrate the proposed SAE-RNN model can effectively extract the features of regional rainfall patterns to accurately forecast multistep-ahead regional flood inundation depths (maps).

2.4. Evaluation metrics

This study adopts the root mean square error (RMSE) and the coefficient of determination (R^2) to evaluate the forecast results and use the mean absolute error (MAE) to show the error distribution of forecast results in the grid map.

An RMSE value represents the error between the forecasted and observed values, which can be calculated by the following equation.

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{t=1}^{N}(d_i - y_i)^2} \tag{29}$$

where N is the number of samples; d_i is the target output value; and y_i is the model output value.

Table 2
24-h cumulative rainfall with various return periods at four rain gauges.

Recurrence interval (year)	24-h cumulative rainfall (mm)			
	Rain gauge No. 1	Rain gauge No. 2	Rain gauge No. 3	Rain gauge No. 4
10	477.7	364.7	501.3	872.6
25	576.1	439.0	605.7	1057.6
100	721.9	548.9	764.1	1331.6
200	794.2	603.4	844.7	1467.3

R^2 is commonly used to evaluate the linear correlation between model outputs and target outputs, which can be calculated by the following equation.

$$R^2 = \left[\frac{\sum_{t=1}^{N}\left(d_i - \overline{d}\right)(y_i - \overline{y})}{\sqrt{\sum_{t=1}^{N}(d_i - \overline{d})^2 \sum_{t=1}^{N}(y_i - \overline{y})^2}}\right]^2 \tag{30}$$

where $\overline{d}$ is the mean of target outputs; and $\overline{y}$ is the mean of model outputs.

An MAE value represents the absolute error between the forecasted and observed values, which can be calculated by the following equation.

$$\mathrm{MAE} = \frac{1}{N}\sum_{t=1}^{N}|d_i - y_i| \tag{31}$$

3. Study area and materials

Taiwan is located in the north-western Pacific Ocean, where the activities of subtropical jet streams are frequent. The Yilan County in north-eastern Taiwan spans an area of approximately 2143 km^2 and is selected as the study area because this county has a long history of flooding problems that continuously threaten the lives and livelihoods of residents. The historical observed data of inundation depths are so rare that it is almost inevitable to find alternative datasets for building flood forecast models. In this study, the synthetic hydrographs of inundation depths for various typhoon events and the designed rainfall patterns were obtained from the Water Resources Agency in Taiwan. These synthetic hydrographs and designed rainfall patterns were validated by various simulation models, such as the HEC-1 model, the Storm Water Management Model (SWMM), and the two-dimensional non-inertial overland flow simulation model. Investigative data consist of 31 historical rainfall events and 24 designed rainfall events with various return periods at four rain gauges. The investigation focuses only on the inundation-prone region, which is of about 271 km^2. The region is divided into a total of 169,797 grids, with a grid resolution of 40 m × 40 m (169,797 grids ≅ 271 km^2/(40 m × 40 m)). Fig. 6 shows the inundation-prone region and the locations of four rain gauges.

Table 1 displays historical rainfall events, and Table 2 displays 24-h cumulative rainfall with various return periods at four rain gauges. The 24-h hydrographs of flood inundation depths corresponding to these 55 events (31 historical and 24 designed events) were used to establish the forecast models of regional flood inundation depths, where 40 events (i.e. 40 events × 24 h = 960 samples) were for training, 6 events (i.e. 6 events × 24 h = 144 samples) were for validation, and 9 events (i.e. 9 events × 24 h = 216 samples) were for testing. In brief, we have collected 1320 (55 events × 24 h) hourly datasets (samples), where each dataset contains 169,797 grids of regional flood inundation depths. The total number of data used in this study are 224,132,040 (55 events × 24 h × 169,797 grids), which is very large.

Table 3
Performance of SAE models.

SAE model[a]	Number of neurons in each hidden layer		L	$CP^j_{\lambda_m}$ [c]	Dimension of C_m	Restore error[d] (RMSE) (m)	
	encoder	decoder				Training	Validation
SAE_1	X_0,C_1	C_1, X'_0	0.999	0.999	144	0.030	0.042
SAE_2	X_0, 144,C_2	C_2, 144,X'_0	0.999	0.999	22	0.031	0.039
SAE_3	X_0, 144, 22,C_3	C_3, 22, 144,X'_0	0.990	0.993	14	0.040	0.039
SAE_4	X_0, 144, 22, 14,C_4	C_4, 14, 22, 144,X'_0	0.990	0.993	7	0.031	0.039
SAE_5	X_0, 144, 22, 14, 7,C_5	C_5, 7, 14, 22, 144,X'_0	0.990	0.996	5	0.031	0.033
SAE_6	X_0, 144, 22, 14, 7, 5,C_6	C_6, 5, 7, 14, 22, 144,X'_0	–	–	4	0.030	0.032
SAE_7	X_0, 144, 22, 14, 7, 5, 4,C_7	C_7, 4, 5, 7, 14, 22, 144,X'_0	–	–	3	0.030	0.032
SAE_8	X_0, 144, 22, 14, 7, 5, 4, 3,C_8	C_8, 3, 4, 5, 7, 14, 22, 144,X'_0	–	–	2	0.046	0.042
SAE_{6R}[b]	X_0, 144, 22, 14, 7, 5,C_{6R}	C_{6R}, 5, 7, 14, 22, 144,X'_0	–	–	4	Min: 0.084 Max: 0.397	Min: 0.063 Max: 0.300

[a] X_0 and X'_0denote the input vector (dimension: 169,797) and the output vector (dimension: 169,797), respectively. C_m denotes the code of SAE_m.
[b] $P^j_{\lambda_m}$ is the ratio of the *jth* eigenvalue to the sum of all eigenvalues.
[c] SAE_{6R} that uses the random method on weight initialization for the SAE model has the same number of neurons in hidden layers asSAE_6.
[d] A restore error in RMSE is the loss of flood data during the encode-decode process in SAE.

4. Results and discussion

4.1. Dimension reduction of regional flood data by SAE

This study constructed nine SAE models with eight different structures of hidden layers to explore the effect of hidden layers on forecasting. The numbers of neurons in hidden layers, the dimension of the code, and the restore errors (RMSE) in training and validation stages for each SAE model are shown in Table 3. In the beginning, the L value of Eq. (14) was set as 0.999 and the PCA result of input data X_0 (dimension: 169,797 grids) indicated the dimension of C_1 was 144. Besides, the RMSE values of training and validation stages for SAE_1were 0.030 m and 0.042 m, respectively. The L value for SAE_2 was also set as 0.999, but the L values of SAE_3 up to SAE_5 were changed as 0.990. In the stacking and training process of SAE, the obtained dimensions of C_2 up to C_5 were 22, 14, 7, and 5, respectively. The RMSE values of SAE_2 up to SAE_5 in training and validation stages fell between0.031 m and 0.040 m. According to the PCA results for C_5, all the $CP^j_{\lambda_5}$ obtained (j < m in Eq. (14)) were smaller than L ($CP^4_{\lambda_5} = 0.98$) such that the cumulative percentage can't provide the dimension information of the code (C_6) for SAE_6. Therefore, this study decided to set the dimension of the next code as the dimension of the current code minus 1 for $m > 5$. That is to way, SAE_6, SAE_7 and SAE_8 were constructed and trained using 4-, 3- and 2-dimensional codes, respectively. The RMSE values of SAE_6 and SAE_7 in training and validation stages fell within the range between 0.030 m and 0.032 m. The RMSE values of SAE_8 in training and validation stages fell within the range between 0.042 m and 0.046 m.

Overall, among the nine models SAE_7 with a 3-dimensioal code is the best SAE model because its restore error is the smallest. Although SAE_8 has the lowest dimension of the code, its restore error, however, is significantly large. Comparing SAE_4 and SAE_5, the restore error in the validation stage decreases. We speculate that this is the advantage of adding more hidden layers by using a nonlinear activation function, which allows the model to explain the spatial variability of regional flood data with fewer codes. This phenomenon may also explain why the dimension of the code decreases but the restore error does not increase significantly during the stacking process from SAE_5 to SAE_7. That is to say, a deeper network structure commonly enhances the compressibility and restorability of the SAE model but has a limitation in the dimensionality of the code.

In order to compare the difference in weight initialization between PCA and the random method (refer to SAE_6 and SAE_{6R}, respectively), the related RMSE values for SAE_{6R} performed 10 rounds fell within 0.084 m and 0.397 m in the training stage and within 0.063 m – 0.300 m in the validation stage. Obviously, the training process of the SAE model would be difficult to converge due to the uncertainty caused by random initialization. In contrast, such training would be stable and accurate if PCA is used to initialize weights. For instance, the restore error could be reduced by 49% up to 64% if random initialization was replaced by PCA (Table 3). Finally, SAE_5 up to SAE_8 were used to build the regional flood forecast models in the following subsection.

Table 4
Performance (overhead) of dimension reduction associated with the number of the leading principal components of PCA for SAE.

Number of leading principal component (j)	Cumulative percentage (CP)	Number of weights in SAE (N = 169,797 × j × 2 + 169,797 + j)	Computer data storage (=N × $8/2^{20}$)	SAE_1 Training time
1	87.8%	509,392	3.89 MiB	<1 min
2	94.5%	848,987	6.48 MiB	<1 min
10	99.0%	3,565,747	27.20 MiB	5 min
144	99.9%	49,071,477	374.39 MiB	30 min
398	99.99%	135,328,607	1032.48 MiB	73 min
643	99.999%	218,529,382	1667.25 MiB	162 min

Model efficiency would be affected mainly by the leading principal components to be retained in the dimension reduction process, where the first round of dimension reduction for 169,797 grids would be the most important. After trial and error, this study determined 144 was the best dimensionality in the first round. Taking SAE_1 as an example, Table 4 presents the performance (overhead) of dimension reduction associated with the number of the leading principal components of PCA. Referring to Table 4, it is foreseen that the proposed algorithms can still be effective as the scale of data increases, owing to the robust architecture that hybrids SAE and PCA.

4.2. Multiple-hour-ahead forecasts of flood features by LSTM-based RNN models

The rainfall records collected at four rain gauges were used to forecast the flood features compressed by SAE. According to the architecture shown in Fig. 5, there are three hyper-parameters to be determined: the time delay k of inputs (rainfall data); the horizon of multistep-ahead forecasting; and the numbers of neurons in the first, second and third layers. The time delay k of rainfall data was determined, by the try and error procedure, as 24, 16, 8 and 4 h in this study. The results shown in Table 5 indicated the RMSE values increase as k decreases in all three stages, which suggests 24 h would be the best setting of the time delay of inputs (rainfall patterns) in this case study.

In order to meet the demand of disaster prevention, this study focused on 1- to 3-hour-ahead forecasts. As known, using a large number

Table 5
Performance of the LSTM-based RNN model with different time delays of inputs (rainfall data).

Time delay of inputs (rainfall data)	RMSE			R^2		
	Training	Validation	Testing	Training	Validation	Testing
24 h	0.20	0.19	0.23	0.97	0.96	0.95
16 h	0.31	0.28	0.29	0.94	0.92	0.92
8 h	0.45	0.41	0.50	0.84	0.80	0.74
4 h	0.55	0.58	0.66	0.76	0.61	0.55

Table 6
Performance of LSTM-based RNN models.

SAE model	RNN model	n_{C_r} [a]	RMSE			R^2		
			Training	Validation	Testing	Training	Validation	Testing
SAE_5	RNN-A1	5	0.41	0.35	0.40	0.85	0.87	0.81
	RNN-A2	10	0.30	0.24	0.28	0.92	0.94	0.91
	RNN-A3	15	0.24	0.24	0.25	0.95	0.94	0.92
	RNN-A4 [b]	**20**	**0.21**	**0.22**	**0.24**	**0.96**	**0.95**	**0.93**
	RNN-A5	25	0.27	0.25	0.26	0.94	0.93	0.92
SAE_6	RNN-B1	5	0.38	0.29	0.32	0.89	0.90	0.89
	RNN-B2	10	0.28	0.23	0.25	0.94	0.94	0.94
	RNN-B3	15	0.24	0.22	0.25	0.95	0.94	0.93
	RNN-B4	**20**	**0.20**	**0.19**	**0.23**	**0.97**	**0.96**	**0.95**
	RNN-B5	25	0.24	0.19	0.22	0.96	0.96	0.95
SAE_7	RNN-C1	5	0.25	0.20	0.26	0.94	0.95	0.92
	RNN-C2	10	0.17	0.15	0.18	0.97	0.97	0.96
	RNN-C3	**15**	**0.14**	**0.13**	**0.16**	**0.98**	**0.98**	**0.97**
	RNN-C4	20	0.19	0.13	0.17	0.97	0.98	0.97
	RNN-C5	25	0.15	0.14	0.15	0.98	0.98	0.97
SAE_8	RNN-D1	5	0.54	0.38	0.42	0.76	0.81	0.75
	RNN-D2	**10**	**0.12**	**0.10**	**0.12**	**0.99**	**0.99**	**0.98**
	RNN-D3	15	0.15	0.11	0.15	0.98	0.98	0.97
	RNN-D4	20	0.13	0.11	0.13	0.99	0.98	0.98
	RNN-D5	25	0.11	0.10	0.12	0.99	0.99	0.98

[a] n_{C_r} is the dimension of the code (C_r) of rainfall patterns for the LSTM-based RNN models.
[b] The selected RNN models are marked in bold.

Table 7
Performance of SAE-RNN models.

SAE-RNN model	SAE model	Dimension of the code	RNN model	Time step	RMSE (m)			R^2		
					Training	Validation	Testing	Training	Validation	Testing
SAE-RNN 1	SAE_5	5	RNN-A4	T + 1	0.08	0.06	0.08	0.97	0.96	0.97
				T + 2	0.09	0.07	0.09	0.97	0.96	0.96
				T + 3	0.08	0.06	0.08	0.98	0.97	0.97
SAE-RNN 2	SAE_6	4	RNN-B4	T + 1	0.07	0.06	0.07	0.98	0.96	0.96
				T + 2	0.07	0.06	0.08	0.98	0.96	0.97
				T + 3	0.06	0.06	0.07	0.98	0.97	0.97
SAE-RNN 3	SAE_7	3	RNN-C3	T + 1	0.08	0.06	0.07	0.97	0.96	0.97
				T + 2	0.08	0.06	0.08	0.98	0.96	0.97
				T + 3	0.07	0.07	0.09	0.98	0.96	0.96
SAE-RNN 4	SAE_8	2	RNN-D2	T + 1	0.08	0.07	0.08	0.97	0.95	0.96
				T + 2	0.09	0.07	0.08	0.97	0.96	0.96
				T + 3	0.08	0.07	0.08	0.97	0.96	0.97

of neurons might only have little effect on model's performance but would significantly increase the number of rounds of trial and error. For simplicity, the numbers of neurons (n_{C_r}) the three flood forecast models (T + 1, T + 2, and T + 3) were set the same (5, 10, 15, 20 and 25) and the corresponding results are shown in Table 6. The results indicated a model with the smallest n_{C_r} (i.e. 5) would lead to comparatively large RMSE values and comparatively small R^2 values whereas a model with a large n_{C_r} value (i.e. 25) not only took much time to train but also failed to achieve the best performance. Thus, the suitable value of n_{C_r} for producing reliable and accurate results would be somewhere among 10, 15 and 20.

4.3. Regional multiple-hour-ahead forecasts of flood depths by SAE-RNN models

The performance of four selected SAE-RNN models for multistep-ahead regional flood inundation forecasts are shown in Table 7. As shown, the SAE-RNN models are, in general, reliable and accurate. For instance, the RMSE values of 1- to 3-hour-ahead forecasting fell between 0.07 m and 0.09 m while the R^2 values fell between 0.96 and 0.97 in the testing stage.

In order to further evaluate the four SAE-RNN models, we used three more performance indicators: (1) the regional flood forecast error of the SAE-RNN model; (2) the restore error of the SAE model; and (3) the flood

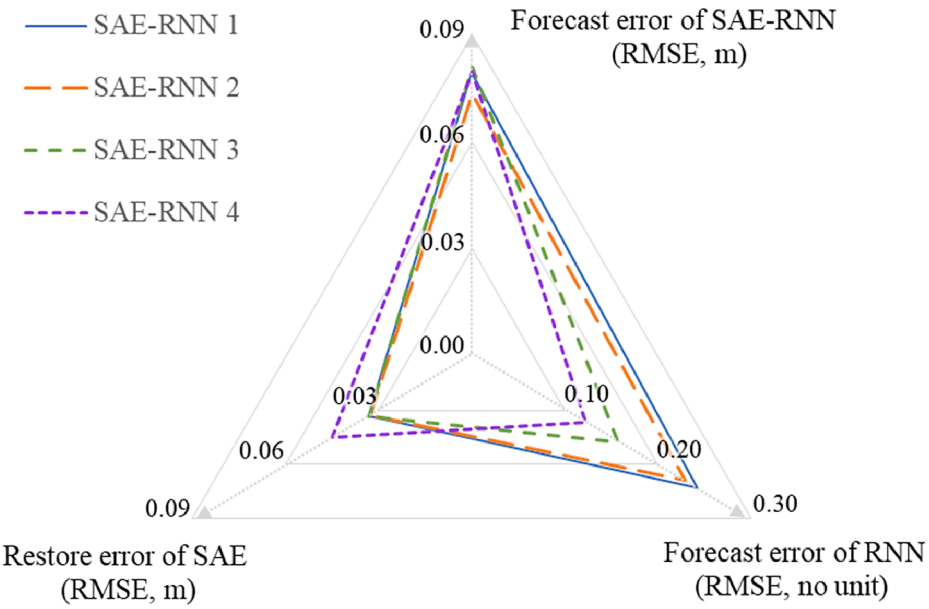


Fig. 7. Model performance evaluation in the testing stages. Top: the forecast error of SAN-RNN (RMSE values of 1- to 3-hour-ahead regional flood forecasts). Bottom left: the restore error of SAE (loss of flood data during the encode-decode process). Bottom right: the forecast error of RNN (RMSE values of 1- to 3-hour-ahead forecasts on flood features).

feature forecast error of the LSTM-based RNN model. Fig. 7 presents the results of four models in testing stages using these three indicators. For simplicity, the results are the averages of errors associated with 1- to 3-hour-ahead forecasts. As shown, the SAE-RNN 2 model has the smallest regional flood inundation forecast error (RMSE = 0.073 m). Besides, SAE-RNN models do not make obvious difference in the restore error except that the SAE-RNN 4 model has the largest error (RMSE = 0.045 m). This suggests that the dimension insufficiency in the code (flood features) will increase the restore error. Thus, the dimension of the code should be greater than 2.

Fig. 8 further show the performance of the SAE-RNN 2 model in the testing stage (216 datasets), which displays the RMSE and R^2 values in space (i.e. all the 169,797 grids) at three selected time levels (i.e. 1- to 3-hour-ahead). The RMSE and R^2 values of each grid were calculated based on the forecasted and the target flood inundation depths of 216 datasets. According to Fig. 8(a)–(c), it is observed that there is no significant difference in the spatial distribution of RMSE values at T + 1, T + 2, and T + 3. The RMSE values of all the grids were <0.3 m, with more than 78% of the values <0.1 m. The results clearly indicated that the SAE-RNN 2 model could, in general, appropriately make regional 1- to 3-hour-ahead forecasts in the study area. According to Fig. 8(d)–(f), it is apparent that there is no much difference in the spatial distribution of R^2 at T + 1, T + 2, and T + 3. Moreover, about 90% of the R^2 values were higher than 0.95. High R^2 values occurred at most of the inundation prone areas while relatively low R^2 values (<0.8) only occurred at the areas with low maximum flood inundation depths (<0.1 m). The relatively low R^2 values in these grids (areas) are reasonable and acceptable.

As for RNN models, we notice that the flood feature (code) forecast error decreases as the dimension of the code decreases. For instance, SAE-RNN 4 with a 2-dimensional code has the smallest RMSE value while SAE-RNN 1 with a 5-dimensional code has the biggest RMSE value. It appears that an increase in the dimension of the code will accumulate errors systematically. Thus, a higher dimension of the code might cause a negative impact on the accuracy of regional flood inundation forecasts. The controversy between the restore error of SAE and the forecast error of RNN suggests that a model with a suitable dimension of the code to balance these two indicators can provide reliable and accurate regional flood inundation forecasts.

4.4. Testing the SAE-RNN model by typhoon events

Two storm events were used to further test the SAE-RNN model. This study selected the SAE-RNN 2 model that, in general, has the best performance in the testing stage (Table 7). The first event is Event D06 (a designed event), which is one of the test events with 24-h hydrographs of flood inundation depths. The 24-hour cumulative rainfalls of this event at Rain Gauges No. 1 – No. 4 were 794.2 mm, 364.7 mm, 605.7 mm, and 1331.6 mm, respectively. The maximum inundation depths over the study area is shown in Fig. 9(a). It is obvious that many places are seriously inundated (e.g., over 2 m). We notice that the RMSE values of 1- to 3-hour-ahead regional flood inundation forecasts were <0.081 m and the R^2 values ranged between 0.978 and 0.986. The distribution of the forecasted maximum inundation depths is shown in Fig. 9(b). The results demonstrates that the two inundation distribution maps (simulated vs. forecasted) look quite similar.

The second event is Typhoon Mitag in 2019. The 72-hour cumulative rainfalls of this event at Rain Gauges No. 1 – No. 4 were 5338 mm, 261 mm, 239 mm, and 450 mm, respectively. There is no monitoring flood inundation dataset available, while there are four major inundated locations (a – d) reported and recorded by the government (Fig. 9(c)). The distribution of the forecasted maximum inundation depths is shown in Fig. 9(d). We noticed that our forecasted results contained all the flooding locations recorded by the government. Comparing Fig. 9(c) and 9(d), all the four reported inundated locations are within our forecasted inundation area.

Fig. 10 shows the forecast results of the SAE-RNN 2 model at the 24th hour of Event D06, where Fig. 10(a) presents the T + 1 outputs at the 23rd hour, Fig. 10(b) presents the T + 2 outputs at the 22nd hour, Fig. 10 (c) presents the T + 3 outputs at the 21st hour, and Fig. 10(d) presents the target values at the 24th hour. It is noted that the target values were generated (obtained) by SWMM. In general, the distribution of the forecasted flood inundation depths is similar (slightly higher) to that of the target values. Besides, there is no significant difference between the output values at T + 1 up to T + 3. The results indicated that the SAE-RNN 2 model could provide reliable and suitable flooding locations and inundation depths.

Overall, the forecasted results provide further evidence that the proposed SAE-RNN model not only can effectively map (project)

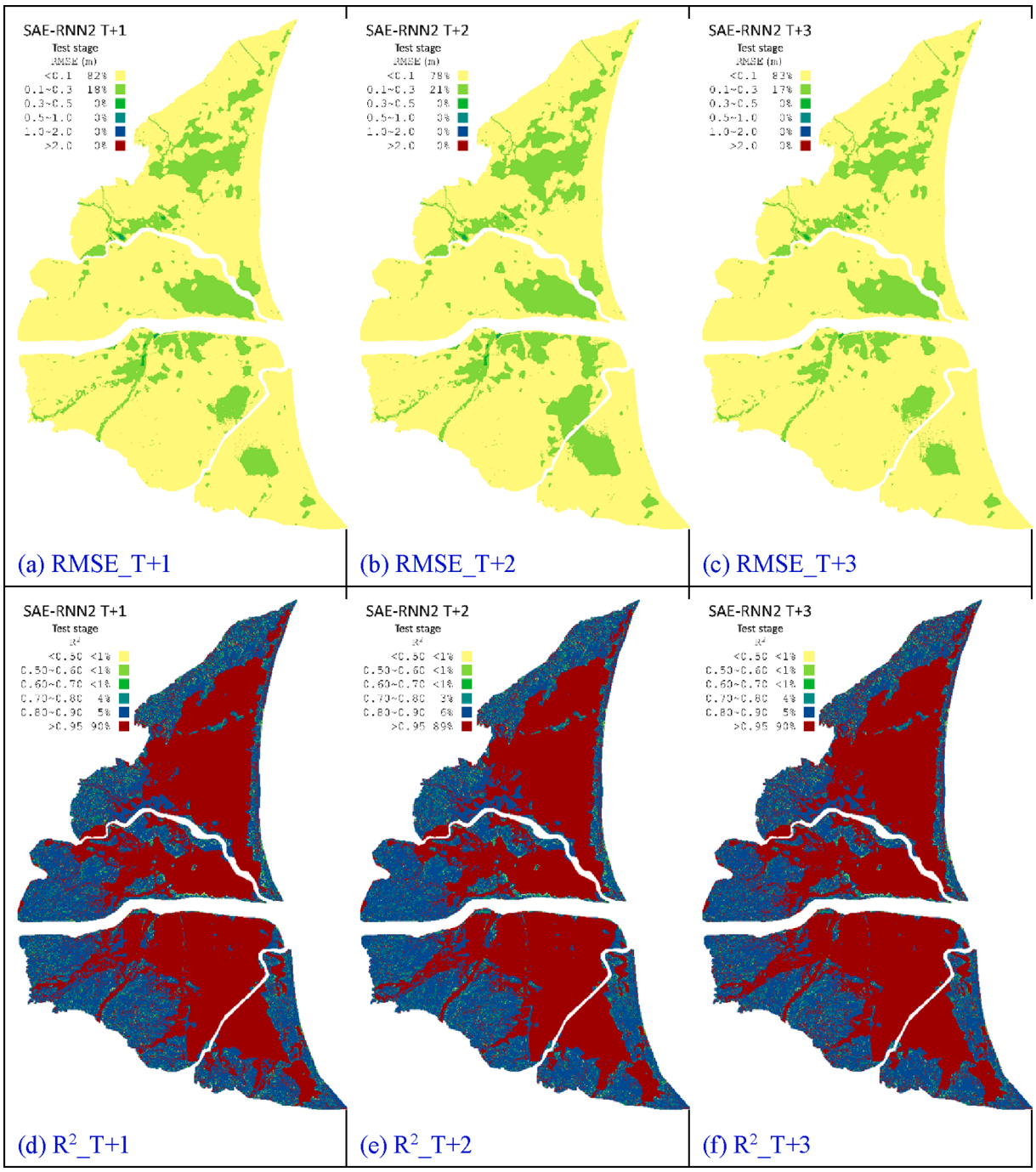


Fig. 8. Performance of the SAE-RNN 2 model in the testing stage, with valuation metrics being calculated in space (169,797 grids) at three time levels (i.e. 1- to 3-hour-ahead forecast).

regional rainfall patterns onto regional flood inundation depths but also can suitably make 3-hour-ahead forecasts of the regional inundation depths in practical applications.

4.5. Visualization of the spatio-temporal distribution of regional flood inundation

As mentioned above, the SAE model can reduce the huge dimensionality of regional flood inundation depths into a low-dimensional code of flood features. We like to present the values of low-dimensional code by the constructed SAE model to visually explore its flood features along with the storm events. The 2-dimensional flood features encoded by SAE_8 can be easily displayed in the x-y coordinate plane, and the 2-dimensional flood features over time can well represent the track of regional inundation depths of a storm event. This study attempts to display 2-dimensional flood features and their corresponding regional flood inundation depths in the same plane to visualize the temporal and spatial distribution of flood information in the study area.

The first step uses SAE_8 (encode mode) to calculate 2-dimensional hourly flood features for all the real and designed rainfall events, resulting a total of 1320 points (55 events × 24 hourly data). The distribution of 1230 points (two-dimensional flood features) is illustrated with grids (5 × 6), as shown in Fig. 11(a). As shown in Tables 1 and 2, the rainfall amounts of historical rainfall events are much smaller than those of designed rainfall events. We notice that all the codes associated with real rainfall events are categorized only into the upper-left corner

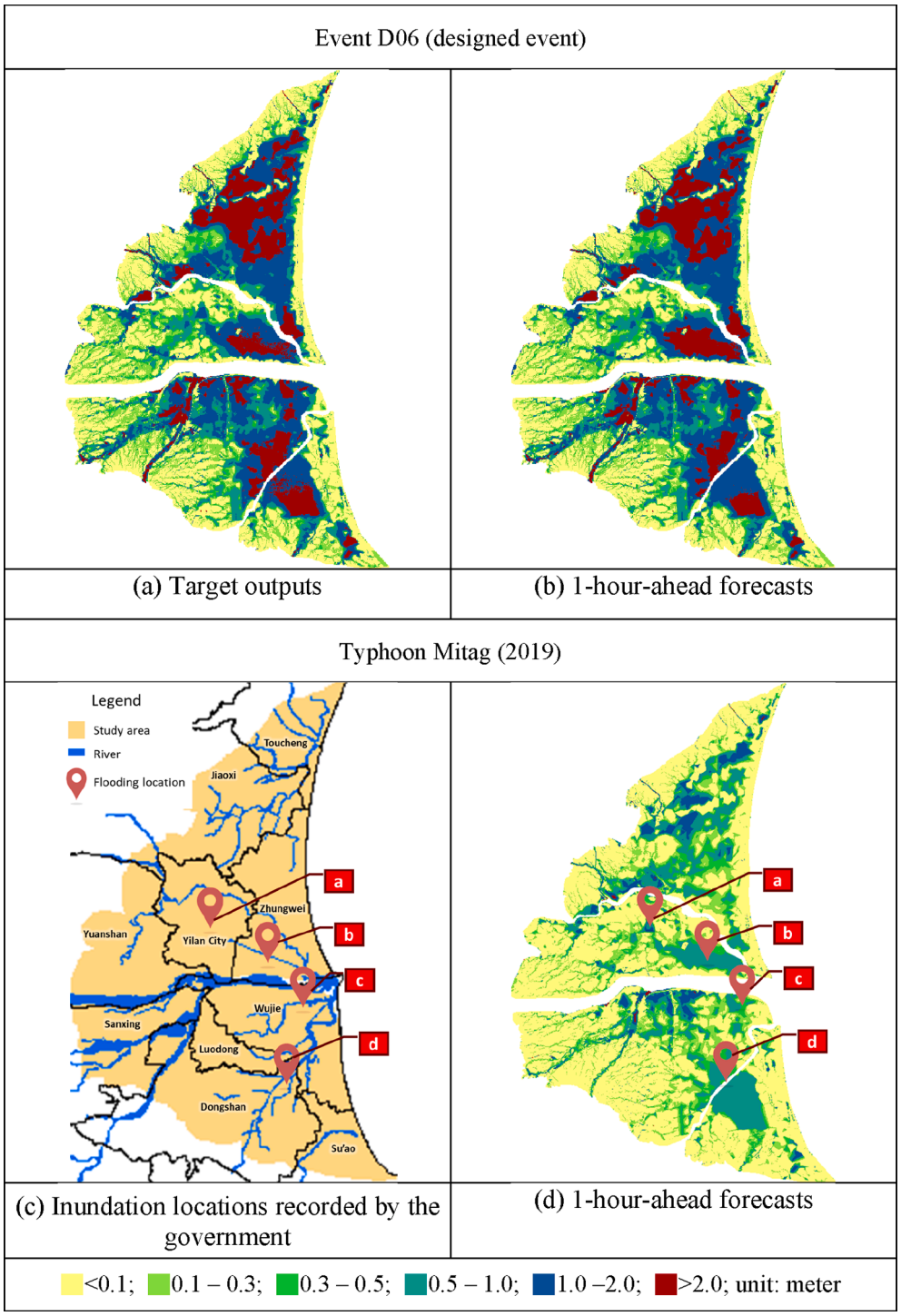


Fig. 9. Comparison results of the maximum flood inundation depths for two test events. (a) and (c) are the distribution of the maximum flood inundation depths for two test events. (b) and (d) are the 1-hour-ahead forecasts by SAE-RNN for two test events. Four major inundation locations shown in (c) are: a. Yilan City; b. North Bank of the Lanyang River (South of Zhungwei County); c. South Bank of the Lanyang River (Wujie County); and d. North and South Bank of the Dongshan River (Dongshan County).

while the codes associated with designed rainfall events are widely spread in the designed x-y coordinate plane. The second step delivers the coordinates of the centroid of each grid that contains at least one point. We then use the decoder of SAE_8 to convert the flood features (coordinates) of each centroid into regional inundation depths, as shown in Fig. 11(b). As shown, the upper-left grids represent regional flood maps with lower values while the lower-right grids represent regional flood maps with higher values.

To visualize the track of regional flood inundation depths of a storm event by the proposed model, we took Event D06 as an example. Fig. 11 (c) displays all the 24 points of Event D06, where the red line represents these points connected in chronological order. It is obvious that the 24 points (the red line) scatter over eight grids only. It is obvious that the 24 points (the red line) scatter over eight grids only. Thus, the regional flood maps of the centroids of these eight grids are displayed for the purpose of comparison. We next plot the regional inundation map of the point the closest to the centroid of the grid, as shown in Fig. 11(d). By comparing Fig. 9(c) and (d), it is found that the values of flood features corresponding to the centroid (Fig. 11(c)) and the selected point (Fig. 11 (d)) the nearest to the centroid in each grid in the course of the red line are close to each other, where the two regional flood maps in each grid are quite similar. Fig. 11 clearly displays that the proposed model can provide the 2-dimensional flood features together with their corresponding regional flood inundation depths over the period of a storm

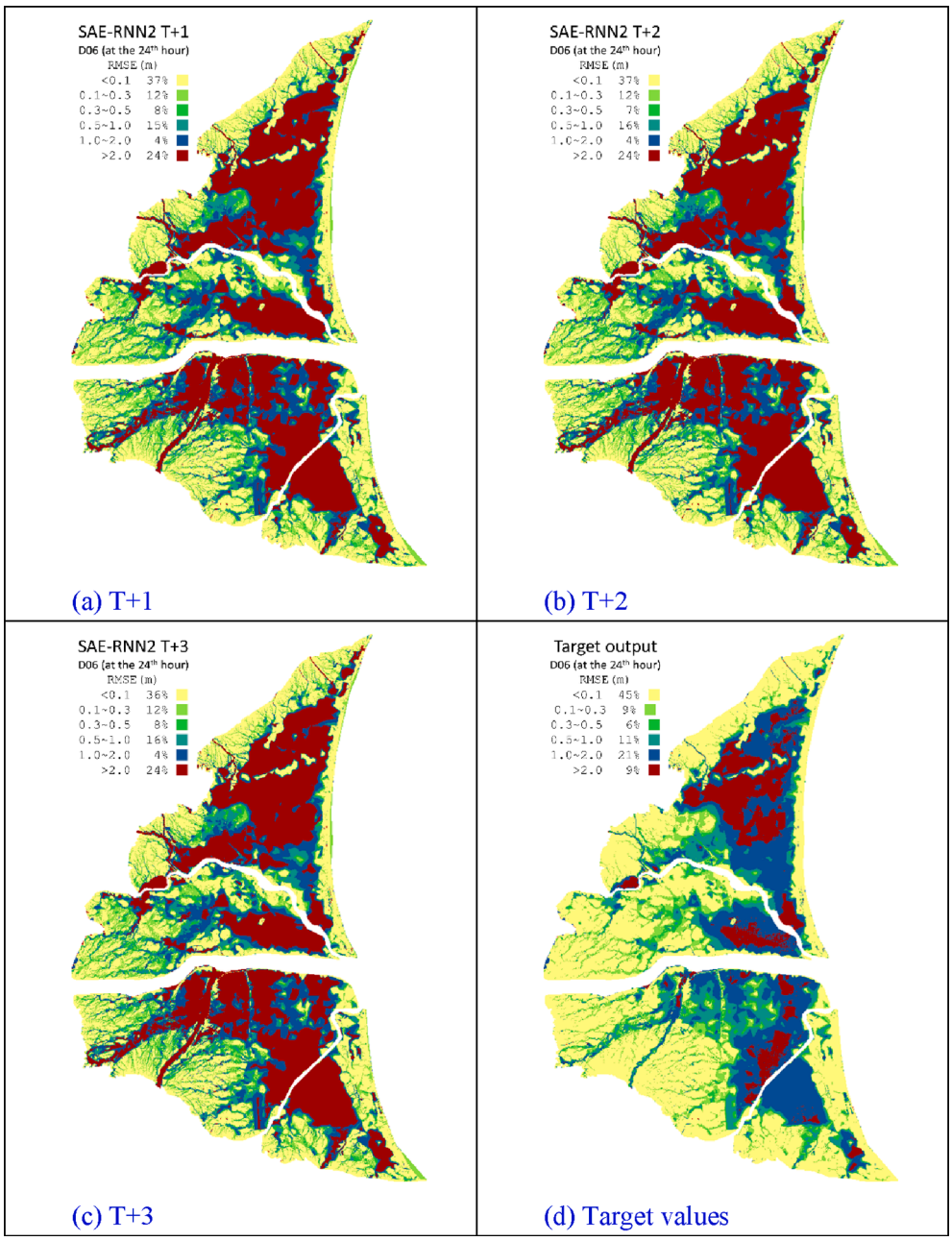


Fig. 10. Forecast results of the SAE-RNN 2 model at the 24th hour of Event D06. (a) T + 1 (the 23rd hour), (b) T + 2 (the 22nd hour), and (c) T + 3 (the 21st hour), as well as (d) the target values at the 24th hour.

event in the same plane to visualize its temporal and spatial flood information in the study area. We would like to note that the 2-dimensional flood features (codes) generated by the model (SAE_8) can be visualized in the x-y coordinate plane and explored further, however, we cannot interpret (extract) their physical meanings, which would be a weakness of this approach.

5. Conclusions

This study develops a big-data mining model to make accurate multistep-ahead regional flood inundation maps based on regional rainfall patterns. We propose a novel SAE-RNN methodology that uses SAE to compress the high-dimensional flood inundation depths into a low-dimensional flood features, uses RNN to forecast flood features based on regional rainfall patterns, and then uses SAE to reconstruct the regional flood inundation depths based on the forecasted flood features. A large number (1320 datasets) of regional rainfall patters with corresponding regional flood inundation depths (169,797 grids) of Yilan County in Taiwan were used to train and validate various SAE-RNN models. The results demonstrate that the models, in general, can provide reliable and accurate forecasts, with RMSE values falling between 0.07 m and 0.09 m and R^2 values higher than 0.95 in all the training, validating and testing stages. The findings of this study are summarized below.

1. The SAE model can effectively reduce the huge dimension of regional flood inundation depths to extract and restore their flood features. The validation and testing results show that the restore errors (RMSE) of the SAE models fall between 0.032 m and 0.042 m in this case study, which are very small, as compared with the highly varied regional inundation depths.

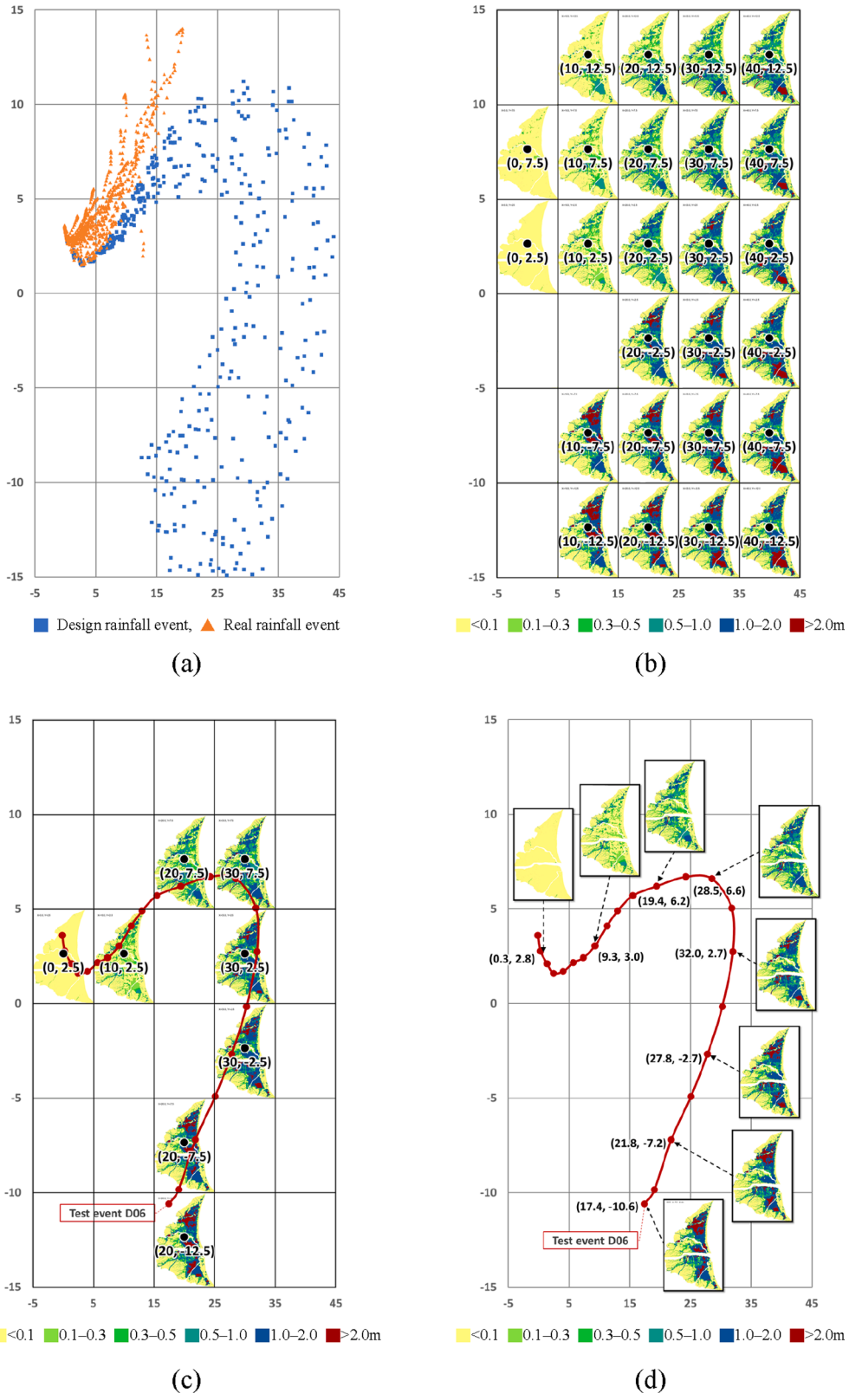


Fig. 11. Visualization of regional flood data. (a) The features of the flood inundation distribution of a flood event encoded by SAE. (b) The flood inundation distribution of equally spaced flood features points restored by SAE. (c) The overlap of the flood features of the D06 test event and the flood inundation distribution of equally spaced flood features points. (d) The original flood inundation distribution of the D06 test event.

2. Compared with the random initialization method, using PCA to initialize network weights of SAE can significantly increase the effectiveness and stability of the SAE model during training and greatly reduce the restore error. Thus, we suggest initializing the weights of the RNN model with the use of PCA.
3. A code with a suitable dimension that balances two indicators (i.e., the restore error of SAE and the flood feature forecast error of RNN) can provide reliable and accurate flood forecasts, which would constitute the best SAE-RNN model.
4. The proposed SAE-RNN model, i.e. 2-code SAE_8, can provide the plot of 2-dimensional flood features and their corresponding regional flood inundation depths over the period of a storm event in a plane, where the temporal and spatial flood information in the study area can be visualized and explored further.

We conclude the proposed SAE-RNN methodology can effectively extract the features of regional rainfall-flood depths and suitably project regional rainfall onto regional flood features to accurately make multistep-ahead forecasts of regional inundation depths in practical applications.

There is often some noise in various data, and therefore noise reduction or removal would be required during modelling. For instance, Bi et al. (2019a, 2019b)) proposed to combine the Savitzky-Golay for noise filtering and wavelet decomposition with stochastic configuration networks for data frequency representation to predict service workload at future time slots. In this study, the extraction of regional flood inundation features through SAE could remove most of the noise from inundation data. However, for rainfall data, the use of LSTM would not be sufficient enough for noise removal or reduction. Our future study will consider adopting noise reduction methods to filter rainfall information before incorporating it into the deep learning models.

CRediT authorship contribution statement

I-Feng Kao: Conceptualization, Data curation, Formal analysis, Methodology, Resources, Software, Validation, Visualization, Writing - original draft. **Jia-Yi Liou:** Data curation, Formal analysis, Methodology, Validation. **Meng-Hsin Lee:** Formal analysis, Methodology, Validation. **Fi-John Chang:** Conceptualization, Formal analysis, Funding acquisition, Investigation, Project administration, Supervision, Writing - review & editing.

Declaration of Competing Interest

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

Acknowledgements

The authors gratefully acknowledge the financial support of the Water Resources Agency, Ministry of Economic Affair (Grant No. MOEAWRA1090316) and the Ministry of Science and Technology (Grant No. 107-2621-M-002-004-MY3), Taiwan, R.O.C. on this research. Simulated data provided by the Water Resources Agency, Taiwan and the Hydrotech Research Institute of National Taiwan University are greatly appreciated. The authors would like to thank the Editors and anonymous Reviewers for their constructive comments that are greatly contributive to the revision of the manuscript.

References

Abbasi, M., Farokhnia, A., Bahreinimotlagh, M., Roozbahani, R., 2020. A hybrid of Random Forest and Deep Auto-Encoder with support vector regression methods for accuracy improvement and uncertainty reduction of long-term streamflow prediction. J. Hydrol. 125717.

Badrzadeh, H., Sarukkalige, R., Jayawardena, A.W., 2015. Hourly runoff forecasting for flood risk management: Application of various computational intelligence models. J. Hydrol. 529, 1633–1643.

Bai, Y., Bezak, N., Sapač, K., Klun, M., Zhang, J., 2019. Short-Term Streamflow Forecasting Using the Feature-Enhanced Regression Model. Water Resour. Manage. 33 (14), 4783–4797.

Bi, J., Yuan, H., Zhang, L., Zhang, J., 2019a. SGW-SCN: An integrated machine learning approach for workload forecasting in geo-distributed cloud data centers. Inf. Sci. 481, 57–68.

Bi, J., Yuan, H., Zhou, M., 2019b. Temporal prediction of multiapplication consolidated workloads in distributed clouds. IEEE Trans. Autom. Sci. Eng. 16 (4), 1763–1773.

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.

Chang, F.J., Tsai, M.J., 2016. A nonlinear spatio-temporal lumping of radar rainfall for modeling multi-step-ahead inflow forecasts by data-driven techniques. J. Hydrol. 535, 256–269.

Chang, L.C., Chang, F.J., Yang, S.N., Tsai, F.H., Chang, T.H., Herricks, E.E.,, 2020. Self-organizing maps of typhoon tracks allow for flood forecasts up to two days in advance. Nat. Commun. 11 (1), 1–13.

Chang, L.C., Amin, M., Yang, S.N., Chang, F.J., 2018. Building ANN-based regional multi-step-ahead flood inundation forecast models. Water 10 (9), 1283.

Chang, L. C., Chang, F. J., Yang, S. N., Kao, I., Ku, Y. Y., Kuo, C. L., Amin, I., 2019. Building an Intelligent Hydroinformatics Integration Platform for Regional Flood Inundation Warning Systems. Water 2019, 11(1), 9.

Chen, P.A., Chang, L.C., Chang, F.J., 2013. Reinforced recurrent neural networks for multi-step-ahead flood forecasts. J. Hydrol. 497, 71–79.

Chiu, C.C., Sainath, T.N., Wu, Y., Prabhavalkar, R., Nguyen, P., Chen, Z., Kannan, A., Weiss, R.J., Rao, K., Gonina, E., Jaitly, N., Li, B., Chorowski, J., Bacchiani, M., 2018. State-of-the-art speech recognition with sequence-to-sequence models. In: In 2018 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 4774–4778.

Ding, Y., Zhu, Y., Feng, J., Zhang, P., Cheng, Z., 2020. Interpretable spatio-temporal attention LSTM model for flood forecasting. Neurocomputing 403, 348–359.

Du, S., Li, T., Horng, S.J., 2018. In: December. Time series forecasting using sequence-to-sequence deep learning framework. IEEE, pp. 171–176.

Fengming, Z., Shufang, L., Zhimin, G., Bo, W., Shiming, T., Mingming, P., 2017. Anomaly detection in smart grid based on encoder-decoder framework with recurrent neural network. J. China Univ Posts Telecommun 24 (6), 67–73.

Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9 (8), 1735–1780.

Humphrey, G.B., Gibbs, M.S., Dandy, G.C., Maier, H.R., 2016. A hybrid approach to monthly streamflow forecasting: integrating hydrological model outputs into a Bayesian artificial neural network. J. Hydrol. 540, 623–640.

Jeong, J., Park, E., 2019. Comparative applications of data-driven models representing water table fluctuations. J. Hydrol. 572, 261–273.

Jiao, R., Huang, X., Ma, X., Han, L., Tian, W., 2018. A model combining stacked auto encoder and back propagation algorithm for short-term wind power forecasting. IEEE Access 6, 17851–17858.

Kao, I.F., Zhou, Y., Chang, L.C., Chang, F.J., 2020. Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting. J. Hydrol. 124631.

Kidoh, M., Shinoda, K., Kitajima, M., Isogawa, K., Nambu, M., Uetani, H., Yamashita, Y., 2020. Deep learning based noise reduction for brain MR imaging: tests on phantoms and healthy volunteers. Magnetic Resonance Med. Sci. 19 (3), 195.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M., 2018a. Rainfall–runoff modelling using long short-term memory (LSTM networks. Hydrol. Earth Syst. Sci. 22 (11), 6005–6022.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M., 2018b. Rainfall–runoff modelling using long short-term memory (LSTM) networks. Hydrol. Earth Syst. Sci. 22 (11), 6005–6022.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., Nearing, G., 2019. Benchmarking a catchment-aware Long Short-Term Memory Network (LSTM for large-scale hydrological modeling. arXiv preprint arXiv 1907.08456.

Le, X.H., Ho, H.V., Lee, G., Jung, S., 2019. Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting. Water 11 (7), 1387.

Li, A., Du, C., Volkow, N.D., Pan, Y., 2020. A deep-learning-based approach for Biophotonics 13 (10), e202000084.

Li, C., Bai, Y., Zeng, B., 2016. Deep feature learning architectures for daily reservoir inflow forecasting. Water Resour. Manage. 30 (14), 5145–5161.

Liu, F., Xu, F., Yang, S., 2017a. In: A flood forecasting model based on deep learning algorithm via integrating stacked autoencoders with BP neural network. IEEE, pp. 58–61.

Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., Alsaadi, F.E., 2017. A survey of deep neural network architectures and their applications. Neurocomputing 234, 11–26.

Liu, P., Zheng, P., Chen, Z., 2019. Deep learning with stacked denoising auto-encoder for short-term electric load forecasting. Energies 12 (12), 2445.

Luo, P., Mu, D., Xue, H., Ngo-Duc, T., Dang-Dinh, K., Takara, K., Nover, D., Schladow, G., 2018. Flood inundation assessment for the Hanoi Central Area, Vietnam under historical and extreme rainfall conditions. Sci. Rep. 8 (1), 1–11.

Marmanis, D., Schindler, K., Wegner, J.D., Galliani, S., Datcu, M., Stilla, U., 2018. Classification with an edge: improving semantic image segmentation with boundary detection. ISPRS J. Photogramm. Remote Sens. 135, 158–172.

Nanda, T., Sahoo, B., Chatterjee, C., 2019. Enhancing real-time streamflow forecasts with wavelet-neural network based error-updating schemes and ECMWF meteorological predictions in Variable Infiltration Capacity model. J. Hydrol. 575, 890–910.

Ni, L., Wang, D., Singh, V.P., Wu, J., Wang, Y., Tao, Y., Zhang, J., 2020. Streamflow and rainfall forecasting by two long short-term memory-based models. J. Hydrol. 583, 124296.
Noori, N., Kalin, L., 2016. Coupling SWAT and ANN models for enhanced daily streamflow prediction. J. Hydrol. 533, 141–151.
Nourani, V., 2017. An emotional ANN (EANN) approach to modeling rainfall-runoff process. J. Hydrol. 544, 267–277.
Nourani, V., Baghanam, A.H., Adamowski, J., Kisi, O., 2014. Applications of hybrid wavelet–artificial intelligence models in hydrology: A review. J. Hydrol. 514, 358–377.
Orland, E., Roering, J.J., Thomas, M.A., Mirus, B.B., 2019. Deep Learning as a tool to forecast hydrologic response for landslide-prone hillslopes. Geophys. Res. Lett. e2020GL088731.
Puttinaovarat, S., Horkaew, P., 2020. Flood forecasting system based on integrated big and crowdsource data by using machine learning techniques. IEEE Access 8, 5885–5905.
Ren, J., Ren, B., Zhang, Q., Zheng, X., 2019. A novel hybrid extreme learning machine approach improved by K nearest neighbor method and fireworks algorithm for flood forecasting in medium and small watershed of loess region. Water 11 (9), 1848.
Sahoo, B.B., Jha, R., Singh, A., Kumar, D., 2019. Long short-term memory (LSTM recurrent neural network for low-flow hydrological time series forecasting. Acta Geophys. 67 (5), 1471–1481.
Sezen, C., Bezak, N., Bai, Y., Šraj, M., 2019. Hydrological modelling of karst catchment using lumped conceptual and data mining models. J. Hydrol. 576, 98–110.
Shafaei, M., Adamowski, J., Fakheri-Fard, A., Dinpashoh, Y., Adamowski, K., 2016. A wavelet-SARIMA-ANN hybrid model for precipitation forecasting. J. Water Land Develop. 28 (1), 27–36.
Shoaib, M., Shamseldin, A.Y., Khan, S., Khan, M.M., Khan, Z.M., Sultan, T., Melville, B. W., 2018. A comparative study of various hybrid wavelet feedforward neural network models for runoff forecasting. Water Resour. Manage. 32 (1), 83–103.
Sit, M., Demiray, B.Z., Xiang, Z., Ewing, G.J., Sermet, Y., Demir, I., 2020. A comprehensive review of deep learning applications in hydrology and water resources. Water Sci. Technol. 82 (12), 2635–2670.
Tan, Q.F., Lei, X.H., Wang, X., Wang, H., Wen, X., Ji, Y., Kang, A.Q., 2018. An adaptive middle and long-term runoff forecast model using EEMD-ANN hybrid approach. J. Hydrol. 567, 767–780.
Taormina, R., Chau, K.W., Sivakumar, B., 2015. Neural network river forecasting through baseflow separation and binary-coded swarm optimization. J. Hydrol. 529, 1788–1797.
Tong, C., Li, J., Lang, C., Kong, F., Niu, J., Rodrigues, J.J., 2018. An efficient deep model for day-ahead electricity load forecasting with stacked denoising auto-encoders. J. Parallel Distrib. Comput. 117, 267–273.
Tsai, M.J., Abrahart, R.J., Mount, N.J., Chang, F.J., 2014. Including spatial distribution in a data-driven rainfall-runoff model to improve reservoir inflow forecasting in Taiwan. Hydrol. Process. 28 (3), 1055–1070.
Valipour, M., 2016. Optimization of neural networks for precipitation analysis in a humid region to detect drought and wet year alarms. Meteorol. Appl. 23 (1), 91–100.
Wang, Y., Yao, H., Zhao, S., 2016. Auto-encoder based dimensionality reduction. Neurocomputing 184, 232–242.
Wiseman, S., Rush, A. M., 2016. Sequence-to-sequence learning as beam-search optimization. arXiv 1606.02960v2.
Xiang, Z., Yan, J., Demir, I., 2020. A rainfall-runoff model with LSTM-based sequence-to-sequence learning. Water Resources Research 56(1), e2019WR025326.
Xiang, Z., Yan, J., Demir, I., 2020. A rainfall-runoff model with LSTM-based sequen-to-sequence learning. Water resources research, 56(1), e2019WR025326.
Xie, T., Zhang, G., Hou, J., Xie, J., Lv, M., Liu, F., 2019. Hybrid Forecasting Model for Non-stationary Daily Runoff Series: A Case Study in the Han River Basin, China. J. Hydrol. 123915.
Yu, S., Ma, J., Wang, W., 2019. Deep learning for denoising. Geophysics 84 (6), V333–V350.
Zabalza, J., Ren, J., Zheng, J., Zhao, H., Qing, C., Yang, Z., Qing, C., Yang, Z., Du, P., Marshall, S., 2016. Novel segmented stacked autoencoder for effective dimensionality reduction and feature extraction in hyperspectral imaging. Neurocomputing 185, 1–10.
Zaytar, M.A., El Amrani, C., 2016. Sequence to sequence weather forecasting with long short-term memory recurrent neural networks. Int. J. Computer Appl. 143 (11), 7–11.
Zhang, J., Zhu, Y., Zhang, X., Ye, M., Yang, J., 2018a. Developing a Long Short-Term Memory (LSTM based model for predicting water table depth in agricultural areas. J. Hydrol. 561, 918–929.
Zhang, J., Zhu, Y., Zhang, X., Ye, M., Yang, J., 2018b. Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. J. Hydrol. 561, 918–929.
Zhou, Y., Chang, F.J., Chang, L.C., Kao, I.F., Wang, Y.S., 2019. Explore a deep learning multi-output neural network for regional multi-step-ahead air quality forecasts. J. Cleaner Prod. 209, 134–145.
Zhou, Y., Guo, S., Xu, C.Y., Chang, F.J., Yin, J., 2020. Improving the reliability of probabilistic multi-step-ahead flood forecasting by fusing unscented Kalman filter with recurrent neural network. Water 12 (2), 578.
Zhu, L., Xu, Z., Yang, Y., Hauptmann, A.G., 2017. Uncovering the temporal context for video question answering. Int. J. Comput. Vision 124 (3), 409–421.
Zhu, Y., Zabaras, N., 2018. Bayesian deep convolutional encoder–decoder networks for surrogate modeling and uncertainty quantification. J. Comput. Phys. 366, 415–447.