Acta Oceanol. Sin., 2021, Vol. 40, No. 11, P. 104–118
https://doi.org/10.1007/s13131-021-1763-9
http://www.aosocean.com
E-mail: ocean2@hyxb.org.cn

Multi-step ahead short-term predictions of storm surge level using CNN and LSTM network

Bao Wang[1, 2], Shichao Liu[2], Bin Wang[2], Wenzhou Wu[3], Jiechen Wang[1, 4], Dingtao Shen[5]*

[1] Jiangsu Provincial Key Laboratory of Geographic Information Science and Technology/Key Laboratory for Land Satellite Remote Sensing Applications of Ministry of Natural Resources/School of Geography and Ocean Science, Nanjing University, Nanjing 210023, China

[2] Key Laboratory of Marine Hazards Forecasting, National Marine Environmental Forecasting Center, Ministry of Natural Resources, Beijing 100081, China

[3] State Key Laboratory of Resources and Environmental Information System, Institute of Geographic Sciences and Natural Resources Research, Chinese Academy of Sciences, Beijing 100101, China

[4] Jiangsu Center for Collaborative Innovation in Geographical Information Resource Development and Application, Nanjing University, Nanjing 210023, China

[5] Changjiang River Scientific Research Institute, Changjiang Water Resources Commission, Wuhan 430010, China

Received 7 October 2020; accepted 13 November 2020

Abstract

Storm surges pose significant danger and havoc to the coastal residents' safety, property, and lives, particularly at offshore locations with shallow water levels. Predictions of storm surges with hours of warning time are important for evacuation measures in low-lying regions and coastal management plans. In addition to experienced predictions and numerical models, artificial intelligence (AI) techniques are also being used widely for short-term storm surge prediction owing to their merits in good level of prediction accuracy and rapid computations. Convolutional neural network (CNN) and long short-term memory (LSTM) are two of the most important models among AI techniques. However, they have been scarcely utilised for surge level (SL) forecasting, and combinations of the two models are even rarer. This study applied CNN and LSTM both individually and in combination towards multi-step ahead short-term storm surge level prediction using observed SL and wind information. The architectures of the CNN, LSTM, and two sequential techniques of combining the models (LSTM–CNN and CNN–LSTM) were constructed via a trial-and-error approach and knowledge obtained from previous studies. As a case study, 11 a of hourly observed SL and wind data of the Xiuying Station, Hainan Province, China, were organised as inputs for training to verify the feasibility and superiority of the proposed models. The results show that CNN and LSTM had evident advantages over support vector regression (SVR) and multilayer perceptron (MLP), and the combined models outperformed the individual models (CNN and LSTM), mostly by 4%–6%. However, on comparing the model computed predictions during two severe typhoons that resulted in extreme storm surges, the accuracy was found to improve by over 10% at all forecasting steps.

Key words: storm surge, prediction, CNN, LSTM, combination

Citation: Wang Bao, Liu Shichao, Wang Bin, Wu Wenzhou, Wang Jiechen, Shen Dingtao. 2021. Multi-step ahead short-term predictions of storm surge level using CNN and LSTM network. Acta Oceanologica Sinica, 40(11): 104–118, doi: 10.1007/s13131-021-1763-9

1 Introduction

Storm surge is a phenomenon of abnormal sea level rise caused by strong atmospheric disturbances, such as tropical cyclones (also called typhoons and hurricanes) and extratropical cyclones. It is one of the most dangerous marine disasters and notoriously threatens human lives and properties. Surge level (SL) forecasting plays a pivotal role in reducing flooding risk in low-lying regions and protecting human lives and maritime activities in coastal regions. Storm surges are mainly driven by meteorological forcings (i.e., pressure gradients and wind velocities) and are characterised by high variability and uncertainty (Mel et al., 2014). Currently, the main methods of SL forecasting consist of experienced predictions and numerical models. Experienced predictions, based on similarity analysis and data statistics, may have overreliance on personal experiences. Numerical models, which physically simulate tidal processes using large amounts of initial data, can successfully capture sea level variations, but their utilisation is computationally expensive and time consuming.

Given their rapid development and wide application, artificial intelligence (AI) techniques have been gradually employed in research on SL prediction because of their high performances in realising the nonlinear mechanism of sea level variations caused by hydro-meteorological factors.

The following is a summary of related literature in this area. Cox et al. (2002) refined a linear model using an artificial neural

Foundation item: The National Key Research and Development Program of China under contract No. 2016YFC1402609; the Open Fund of the Key Laboratory of Marine Hazards Forecasting, Ministry of Natural Resources under contract No. LOMF 1804; the National Natural Science Foundation of China under contract No. 42077438.
*Corresponding author, E-mail: dingtaoshen@outlook.com

network (ANN) with forecasted wind information to improve short-term water level predictions. Sztobryn (2003) constructed and verified a model using a multilayer perceptron (MLP) with wind data for sea level changes during storms. Then, back-propagation neural network (BPNN) models were applied to tide and storm surge predictions, considering not only short-term multi-variable measuring data (Chang and Lin, 2006; Liang et al., 2008) but also typhoon factors (Lee, 2006, 2009; Kim et al., 2016) as model inputs. Rajasekaran (2008) employed support vector regression (SVR) to forecast typhoon-induced storm surges and demonstrated that it is an efficient method compared to numerical models and neural networks.

To improve prediction accuracy, many hybrid models have been proposed. You and Seo (2009) developed a cluster neural network using a cluster analysis methodology. Harmonic analysis, component analysis, Kalman filter, and other methods have also been used to supplement neural networks to improve forecasting accuracy (Lee, 2008; Balas et al., 2010; Mok et al., 2016). Neural-fuzzy, an integration of neural network and harmonic analysis, was certificated as an effective and feasible technique to forecast sea level (Karimi et al., 2013; Zhang et al., 2017; Wang et al., 2020). Kim (2015) developed a time-dependent surrogate model of storm surges based on an ANN with synthetic simulations of hurricanes. Yin and Wang (2016) proposed a novel online sequential extreme learning machine (ELM) based on an improved Gath-Geva fuzzy segmentation algorithm. Imani (2018) examined the applicability and capability of ELM and relevance vector machine models for predicting sea level variations. El-Diasty et al. (2018) combined a neural network with wavelet decomposition to improve prediction accuracy. Kim et al. (2019) described a novel systematic and objective selection procedure for the development of an ANN-based storm surge forecasting model.

In recent years, convolutional neural network (CNN) and long short-term memory (LSTM) models have proved their superiority and reliability in time series forecasting (Li et al., 2019; Sagheer and Kotb, 2019; Wang et al., 2019; Song et al., 2020), but they have scarcely been used in storm surge or sea level predictions. Ishida et al. (2020) developed a coastal sea level estimation model at an hourly temporal scale based on LSTM using wind speed and direction, mean sea level pressure, and air temperature as input data. Further, the two methods are mainly applied in other fields. For example, in wind speed forecasts, Zhao et al. (2020) innovatively applied a one-dimensional CNN to excavate the timing coupled information in data, while Araya et al. (2020) used sub-sequences of different timescales as input for multiple LSTM models to model complex temporal dynamics. Xiao et al. (2019) predicted short- and mid-term sea surface temperature by an LSTM–AdaBoost ensemble learning method. Other applications include day-ahead solar irradiance prediction, short-term load forecasting, and multi-step ahead flood forecasting (Qing and Niu, 2018; Sadaei et al., 2019; Dong et al., 2020; Kao et al., 2020).

As mentioned previously, a CNN is good at collecting salient features from an original signal, while LSTM is better at learning temporal features. However, each model has its advantages and disadvantages, and the combination of the two methods may yield stronger prediction ability (Shi et al., 2015). Many studies have integrated CNN with LSTM for time series predictions. Oh et al. (2018) realised high-performance automated diagnosis of arrhythmia by merging CNN and LSTM models with irregular time series data. Liu et al. (2018) and Chen et al. (2019) built a multifactor spatiotemporal correlation model based on a combination of CNN and LSTM models for wind speed forecasting and compared it with SVR, MLP, CNN, LSTM, and other models. Pak et al. (2020) achieved better prediction results than MLP and LSTM based on an effective CNN–LSTM in predicting PM2.5 concentration of Beijing City by constructing a spatiotemporal feature vector that reflects both linear and nonlinear correlations between parameters. Utilizing time series data, Petersen et al. (2019) proposed a deep neural network model consisting of convolutional and LSTM layers that could capture the non-static spatiotemporal correlations of variability in urban bus travel times with a better accuracy than LSTM in stand-alone mode.

From the summarisation of the literature above, two features can be highlighted. (1) Short-term multi-step SL forecasting and the applications of AI techniques in this field have been attracting attention from researchers recently. (2) CNN and LSTM have significant potential in predicting SL. In the present study, we applied CNN and LSTM models and their two sequential combinations, LSTM–CNN and CNN–LSTM, to make reliable and accurate multi-step-ahead short-term SL forecasts for the first time with lead times of 1 h, 2 h, 4 h and 6 h. It was assumed in this study that in addition its own inertia, SL is primarily influenced by local wind, and other hydro-meteorological factors (such as air pressure) were considered to be implicitly related to local wind (Nitsure et al., 2014). Firstly, sample data were generated according to the time-sharing SL distribution of differently sized regions of interest (ROIs) around the research site. Secondly, multiple models, including SVR, MLP, CNN, LSTM, LSTM–CNN, and CNN–LSTM, were constructed and trained for multi-step ahead SL forecasting. Finally, different measurement indexes and statistical analyses were used to measure the superiority of the proposed models. To demonstrate the applicability of the CNN and LSTM models in multi-step-ahead surge forecasting, this study utilised SL and wind series data of the Xiuying Station in Hainan Province, China, as a case study.

The remainder of this paper is organised as follows. Section 2 presents the methodology and framework of the proposed models. Section 3 introduces the case study and materials. Section 4 presents the results and discussion of the methods applied to multi-step ahead storm surge forecasting. Finally, Section 5 presents the conclusions and future research.

2 Methodology

2.1 *Statistics-based sample generation*

The distribution of training samples in neural network models should evenly cover various situations that may occur in practical applications. Typhoon induced storm surges play an important role in the vulnerability assessment of coastal regions. In the absence of typhoon processes, the SL is relatively low, especially in Southeast China, where it is less affected by extratropical cyclones. If all time series data within valid time are organised as model inputs, the proportion of tidal variations caused by typhoons is very small, and the training dataset will have high bias (as high-value data accounts for a relatively small amount), which may cause low model performance in predicting storm surge levels in typhoon processes. However, if we only use data from a period of time before and after typhoons passing in model training, then we cannot make accurate predictions under the premise that we cannot determine whether the typhoon will transit. A compromise method is to delineate an ROI around the site and select a certain period after the typhoon enters the region as the sample, regardless of whether the typhoon will pass across the site. For example, if the total number of typhoons entering the ROI is n and the selected period length is t, then the

length of the generated sample is $n \times t$. In this way, sample generation translates into the following two issues: (1) the size and location of the ROI and (2) the length of time period (LTP) after typhoons entering the ROI.

In view of this, the following solutions are proposed on the basis of data statistics:

(1) With the site as the centre, demarcating r rectangular ROIs of different sizes at regular intervals, determine and count the number of typhoons crossing the i^{th} (i=1, 2, 3, ..., r) ROI, recorded as t_i.

(2) Calculate the time period of j^{th} (j=1, 2, 3, ..., t) typhoon crosses the i^{th} ROI (the time from when a typhoon is generated in the region or enters the region from outside until it leaves the region or decays in the region). For each hour in the period, the time sequence is recorded as k (k=1, 2, 3, ..., t_i); the SL at the time corresponding to each sequence number is recorded as S_{ijk}.

(3) For the i^{th} ROI, calculate the mean absolute surge level (MASL) at all corresponding hours of the k^{th} sequence number $\text{MASL}_{ik} = \dfrac{\sum_{j=1}^{j=t} \left| S_{ijk} \right|}{N_{ik}}$, where N_{ik} is the total number of records at the k^{th} hour after all passing typhoons enter the i^{th} ROI, and the time-sharing MASL curves are plotted.

(4) Determine the LTP under the condition of covering 95% of the number of records, i.e., accumulate the number of typhoon track nodes every 3 h from the first hour. When the accumulated number of nodes exceeds 95% of the total number of nodes at hour T, select T as the value of LTP.

(5) Select the appropriate ROI according to the MASL curves; the MASL curve of appropriate size ROI should slowly rise after passing through a low value area and then maintain a high level of oscillation before time T. Observing the time-sharing MASL curve, if the curve immediately rises from the first hour or there is a clear peak between hour 1 and T, it indicates that the ROI is too small; thus, select the qualified ROI with the smallest area as the desired region.

The use of entire hourly time series data as inputs to the model is difficult owing to the required computational resources. This method can exclude part of the data according to the unified standard and also can incorporate most of the data affected by typhoons into the sample. It reduces the sample size, saves computational resources, and improves data quality. It should be noted that this method is a rough method for screening sample data, and its purpose is not to accurately exclude all samples that are not affected by typhoons. This study aims to explore the application of CNN and LSTM in storm surge forecasting; sample generation is only the basic data processing work, not the main research direction here. After sample generation, as long as the typhoon enters the selected region, regardless of whether it will cross the vicinity of the site, the trained models based on the generated sample will have a certain prediction ability to a period of time from that moment.

2.2 *CNN and LSTM*

CNNs are a class of deep, feed-forward artificial neural networks that employ a variety of multilayer perceptrons trained according to the error back propagation algorithm. CNNs contain one-dimensional convolution or multi-dimensional convolution. For time series modelling, a one-dimensional CNN (conv1D) seems more suitable. The inputs of conv1D are 1D series containing a few observations in a certain channel. When the data fragment is not highly location correlated, this method has the ability to identify simple patterns in data and then generate more complex patterns at a more advanced level. Therefore, conv1D can successfully obtain effective features from the shorter segments of the overall dataset, which can be used for the analysis and processing of observed data.

The key for the application of a CNN to time series data is its convolutional layer, where a filter is applied to the data across all time stamps. Conv1D can be performed as a filter dedicated to processing sequence data. For a single-layer conv1D, the output vector $\boldsymbol{O}: O_1, \cdots, O_j, \cdots, O_{\left[\frac{a-b+1}{t}\right]}$ is obtained as follows:

$$O_j = \begin{cases} \sum_{i=1}^{b} I_{j+i-1} \cdot k_i, & j = 1, \\ \sum_{i=1}^{b} I_{j+i+t-2} \cdot k_i, & 1 < j \leqslant \left[\dfrac{a-b+1}{t}\right], \end{cases} \tag{1}$$

where $\boldsymbol{I}: [I_1, \cdots, I_i, \cdots, I_a]$ is an input vector with size a, t is stride, and b is the size of the convolution kernel k, $i = \{1, 2, 3, ..., b\}$. Figure 1 illustrates the conv1D strategy. After O_1 is obtained according Eq. (1), the calculation window slides back to I_{t+1} to calculate O_2. The next output is obtained by this rule until the highest t value is reached.

Figure 1 shows a two-layer conv1D network. $\left[X_{i1}^{l_1}, X_{i2}^{l_1}, \cdots, X_{iN_1}^{l_1}\right]$ is obtained after the first layer of convolution for the intercepted signal S_i:

$$X_{ik}^{l_1} = b_{ik}^{l_1} + \text{conv1}D(S_i, W_{ik}^{l_1}), \tag{2}$$

where W and b are the weight matrix and bias respectively, N_1 is the number of the first convolution kernels, $k = \{1, 2, 3, ..., N_1\}$, l_1 indicates the first CNN layer, $\boldsymbol{X}$ is the output vector after convolution. Features generated by a filter usually go through an activation function. In the model, the most widely used "Relu" is applied as the activation function to increase the nonlinear characteristics. Therefore, the output of the single CNN layer group is

$$x_{ik}^{l_j} = \text{Relu}\left(X_{ik}^{l_j}\right), \; j = 1, 2. \tag{3}$$

After the computation of the second CNN layer, the head and tail of $x_{i1}^{l_2}, x_{i2}^{l_2}, \cdots, x_{iN2}^{l_2}$ are connected to obtain the fully connected layer. Then, the fully connected layer, middle layer, and output layer constitute a traditional neural network. Regression activation is adopted to obtain a quantitative forecast value.

Meanwhile, LSTM, an elegant variant of the recurrent neural network (RNN), was proposed to solve the vanishing gradient problem by adding input gates, output gates, and forget gates to the RNN (Hochreiter and Schmidhuber, 1997). It has been used in many applications to analyse time series data. The connections between LSTM units allow information to cycle through a loop across adjacent time steps. The key to LSTM is a cell that can remember or forget information selectively by maintaining a memory state. Information of high significance is retained and back propagated, while irrelevant information forgotten and was discarded.

The structure of LSTM is illustrated in Fig. 2. The memory cell of LSTM looks like a conveyor belt, where the data streaming through are carefully regulated by structures called gates, including a forget gate, input gate, and output gate. The gates are ways for optional inlet of information. They are composed of a sig-

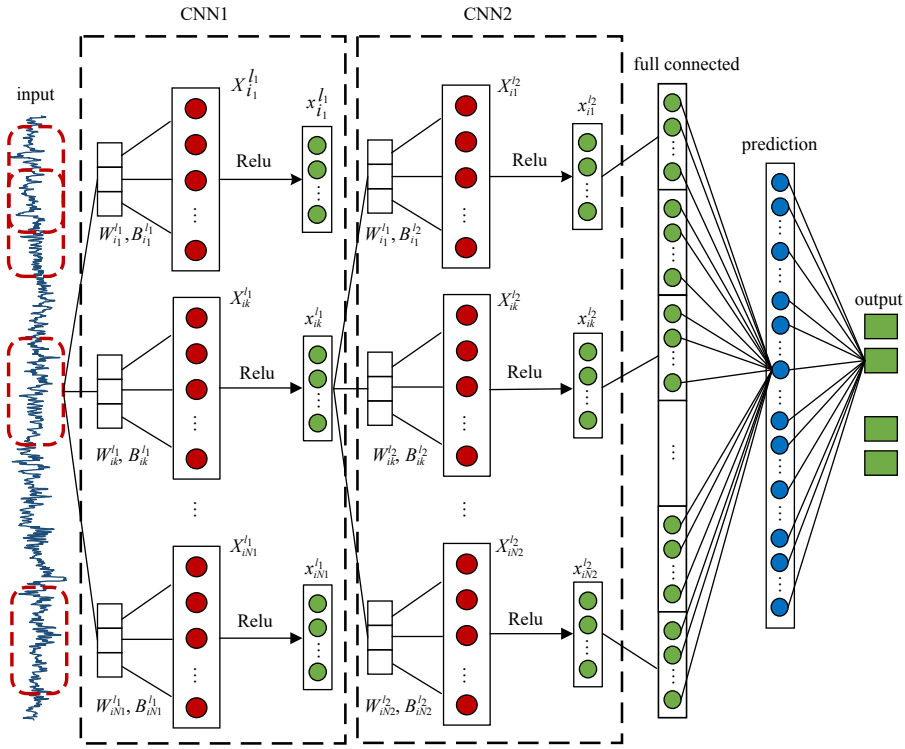

Fig. 1. Structural illustration of conv1D.

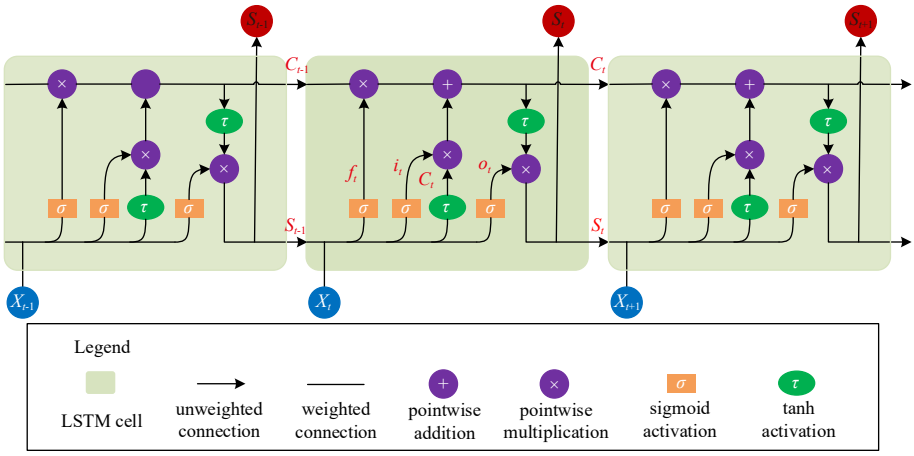

Fig. 2. Structural illustration of LSTM.

moid neural net layer and a pointwise multiplication operation. The working mechanism of the gates and information flow can be expressed using the following equations:

$$\begin{cases} f_t = \sigma\left(X_t U^f + S_{t-1} W^f + b_f\right), \\ i_t = \sigma\left(X_t U^i + S_{t-1} W^i + b_i\right), \\ \tilde{C}_t = \tanh\left(X_t U^c + S_{t-1} W^c + b_c\right), \\ C_t = C_{t-1} \otimes f_t \oplus i_t \otimes \tilde{C}_t, \\ o_t = \sigma\left(X_t U^o + S_{t-1} W^o + b_o\right), \\ S_t = o_t \cdot \tanh\left(C_t\right), \end{cases} \tag{4}$$

where f_t, i_t, and o_t are the forget gate, input gate, and output gate of sigmoid functions σ, respectively. Their values are between 0 and 1 and control the information that is forgotten in the old cell state C_{t-1}, the information that is stored ($\tilde{C}_t$) in the new cell state C_t, and the information that is output from the cell, respectively. $\otimes$ denotes Hardamard product, and $\oplus$ is the concatenation operator. X_t is an input at time step t, and S_t is hidden state at time step t. U^f, U^i, U^c, and U^o are input weights of new input X_t; W^f, W^i, W^c, and W^o are recurrent weights of output S_{t-1} from the previous cell; and b_f, b_i, b_c and b_o are corresponding biases.

2.3 *Combinations of CNN and LSTM*

There are two ways to combine CNN and LSTM sequentially by using a different model as the upper layers: LSTM–CNN and CNN–LSTM. This study applied both combinations to predict short-term SL. In the LSTM–CNN model, temporal features are first trained by several LSTMs, and the obtained result is utilised as the input of CNNs. In the CNN–LSTM model, CNNs compose an input layer that accepts observed variables as inputs and an

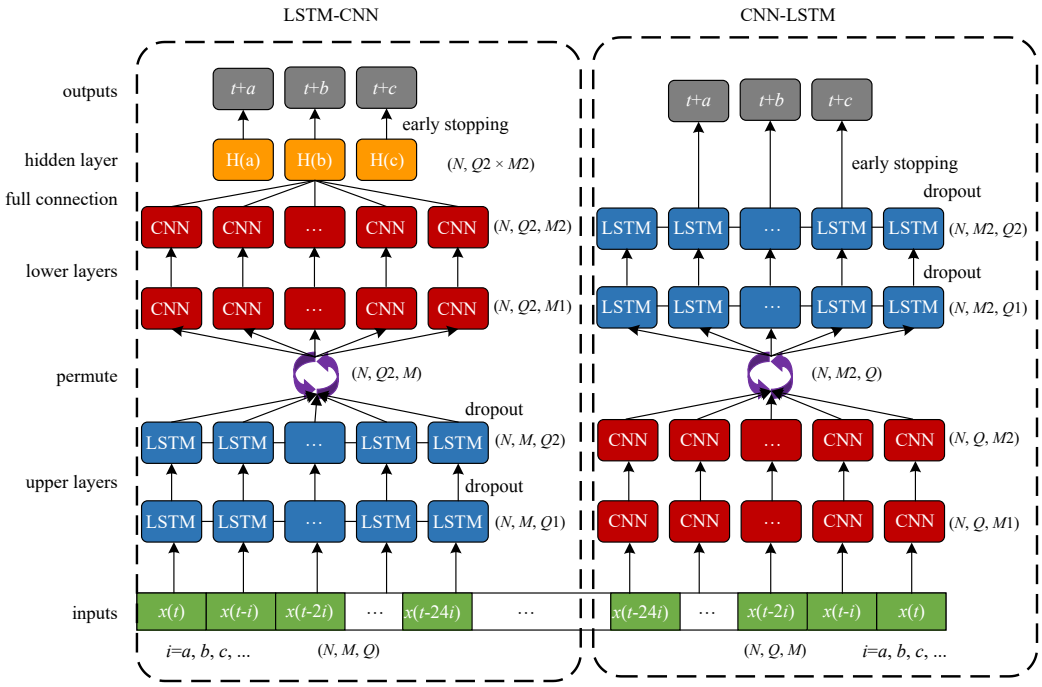

Fig. 3. Frameworks of LSTM–CNN and CNN–LSTM.

output layer that extract features to LSTMs.

A graphical illustration of the integrated models is shown in Fig. 3. Both models contain an input layer, two CNNs, two LSTMs, and an output layer. Compared with CNN–LSTM, LSTM–CNN has an extra fully connected layer and hidden layer. As the input data consists of multivariate time series, we define two time series datasets as tensors of shape (N, M, Q) and (N, Q, M) for LSTM–CNN and CNN–LSTM, respectively, where N is the number of samples in the dataset, Q is the maximum number of time steps amongst all variables, and M is the number of variables processed per time step. $M1$ and $M2$ are the numbers of filters in the CNN layers, and $Q1$ and $Q2$ are output dimensions of the LSTM layers. Because LSTM is capable of learning long-term dependency and CNN can extract time-invariant features, the last two dimensions of the tensors are permuted after the computation of upper layers in both models. The details of model construction in this study, including the organisation of input data and parameter optimisation, are discussed in Section 3.3.

If the combined networks use too many weights and are trained too strictly, training results will be of high accuracy, but they might involve a large number of less meaningful computations for non-trained data. Otherwise, if they use too few weights, they will not be able to learn features of inputs efficiently. Here, we use the regularisation method of dropout and early stopping before the performance function reaches a pre-defined threshold (for example 10^{-4} cm) of the mean absolute error between the observed and predicted SL.

2.4 *Evaluation of model performance*

Three indexes were used to evaluate the performance of different methods for multi-step short-time SL prediction, namely, mean absolute error (MAE), root mean square error (RMSE), and correlation coefficient (CC), which are defined as follows:

$$\mathrm{MAE} = \frac{1}{n}\sum_{1}^{n}|y_i - y'_i|, \tag{5}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{1}^{n} d_i^2}{n}}, \tag{6}$$

$$\mathrm{CC} = \frac{\sum_{1}^{n}(y_i - \bar{y})(y'_i - \bar{y'})}{\sqrt{\sum_{1}^{n}(y_i - \bar{y})^2}\sqrt{\sum_{1}^{n}(y'_i - \bar{y'})^2}}, \tag{7}$$

where d_i is the error vector calculated by the difference between the observed SL value y_i and the predicted SL value y'_i. $\bar{y}$ and $\bar{y'}$ are the mean values of observed SL values and predicted SL values, respectively. n is the number of test samples. The respective indexes are calculated for the predictions on each horizon (1 h, 2 h, 4 h, and 6 h) separately.

3 Case study

3.1 *Observation data*

Xiuying Station was selected as a case study to verify the feasibility and superiority of CNN and LSTM applications for SL prediction. Xiuying Station (20°6′N, 110°10′E) is located in the north of Hainan Island, China, on the south coast of Qiongzhou Strait, facing Leizhou Peninsula of Guangdong Province across the straits. Affected by typhoons, the coast along Qiongzhou Strait to Beibu Gulf is one of the areas with the most severe storm surge disasters. Since 2008, Xiuying Station has encountered 4–5 typhoons per year on average within a range of 200 km. Haikou Port, where Xiuying Station is located, is the busiest port in Hainan Province, so storm surge disasters cause serious economic losses. In 2014, Typhoon Rammasun (#1410) passed Qiongzhou Strait at a maximum wind speed of 55 m/s from the southeast direction, causing the maximum storm surge of Xiuying Station to increase by 215 cm at 10:00 am on July 18, 2014, and the highest SL rose to 343 cm (the top-level warning tidal line is 286 cm). Ac-

cording to statistics from Hainan provincial government, Rammasun caused a direct economic loss of 11.95 billion yuan.

We collected hourly observed sea water level and astronomical tide of Xiuying Station from years 2008 to 2018, the monitoring of which is unified based on the *1985 National Elevation Benchmark of China*. The hourly SL values were obtained by subtracting astronomical tide from sea water level. Owing to the lack of wind speed and wind direction observations at Xiuying Station, we downloaded the reanalysis data of hourly 10 m wind in a (1/4)° resolution grid during these 11 years from the European Centre for the Medium-Range Weather Forecasts (ECMWF) website (https://apps.ecmwf.int/datasets/). The hourly 10 m U and V wind component time series data were extracted from the grid where Xiuying Station is located and served as observation values. All observation datasets were thoroughly checked for continuity, and any missing or default values over an interval of up to 4 h were filled in using values obtained via linear regression, whereas missing or default data values over interval durations exceeding 4 h were omitted. In addition, all track information of north-western Pacific typhoons during these 11 years was also downloaded from Shanghai Typhoon Institute of China Meteorological Administration website (http://tcdata.typhoon.org.cn/). By statistically analysing the times when typhoons entered and stayed away from the sea areas near Xiuying Station (as described in Section 2.1), the sample data were screened to appropriate time periods to reduce the data quantity and optimise data composition.

Figure 4 shows two typhoons: the 23rd typhoon Khanum (1723) in 2017 and 26th typhoon Mangkhut (1826) in 2018. In Section 4.2, these two typhoons are selected as two typical cases for evaluating the predictions of CNN and LSTM models. The years for the test dataset of this study are 2017 and 2018. Khanum and Mangkhut are the two typhoons that caused the highest storm surge at Xiuying Station in these 2 years, causing the highest 76 cm and 72 cm SLs, respectively. The figure shows the transit time, path trajectory, and intensity of the two typhoons. When the intensity is typhoon (TY), severe typhoon (STY), or super typhoon (SuperTY), each path node is 3 h apart from the previous node; at other intensities, it is 6 h apart.

3.2 *Sample data selection*

The sample data selection for model training is based on the method and rules described in Section 2.1.

Firstly, 6 ROIs of different sizes were delineated every 2 latitudes and longitudes, as shown in Table 1. Because typhoons mostly entered the regions from the east and south, inland deep west of 102°E and north of 26°N had few typhoons arrive and were far away from Xiuying Station, so no more regional division was made. Secondly, based on the typhoon tracks from 2008 to 2018, the time-sharing node information of each typhoon entering the ROI and SL values at the corresponding times was determined and recorded. Then, the time-sharing number of nodes and MASLs was computed and counted, and the cumulative percentage of time-sharing typhoon track nodes and MASL curves is plotted in Fig. 5.

Finally, the LTP was calculated and the appropriate ROI was selected. Covering 95% of typhoon track nodes, the values of T were 61, 102, 117, 135, 155, and 161 for regions #1 to #6, respectively. Observing the time-sharing MASL curves, from the 1st hour to the Tth, there were obvious peaks in ROI#1 at the 12th hour, ROI#2 at the 20th, and ROI#3 at the 30th and 66th. In ROI#4, ROI#5, and ROI#6, after a period of low values, the curves then basically maintained high oscillation. According to the principle of minimum area, ROI#4 was selected as the case area. That is, when a typhoon entered ROI#4, 135 hours after that moment was selected as the initial forecast hours of training models.

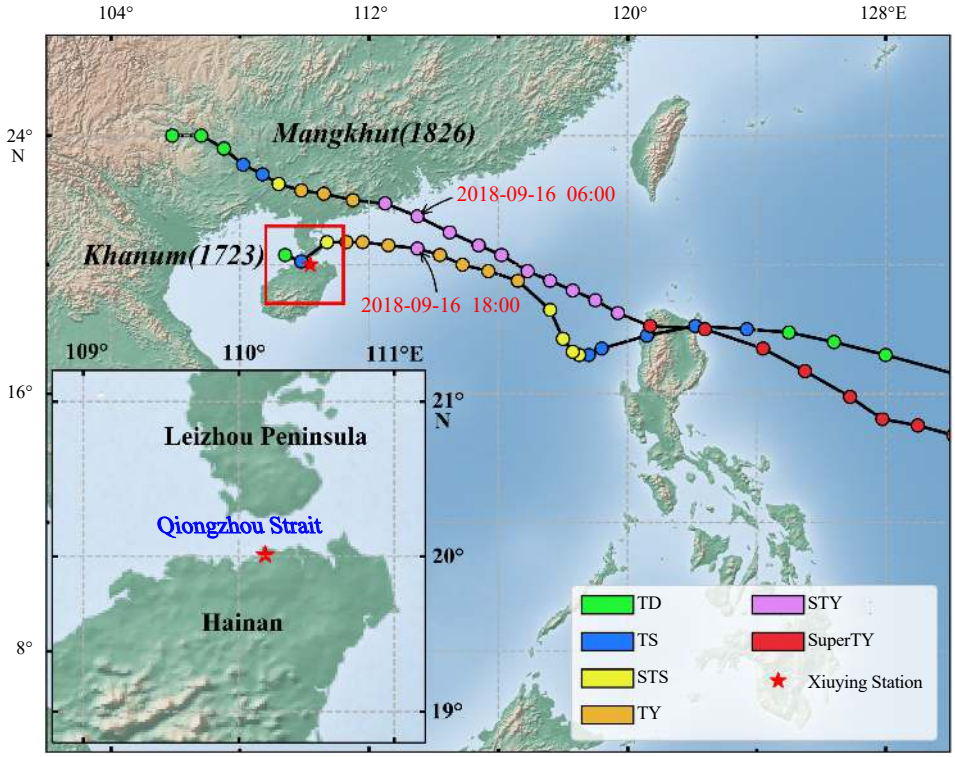

Fig. 4. Location of the Xiuying Station and tracks of typhoons Khanum and Mangkhut. TD is tropical depression, TS is tropical storm, STS is severe tropical storm, TY is typhoon, STY is severe typhoon, and Super TY is super typhoon.

Table 1. Delineation of 6 ROIs around Xiuying Station and corresponding information about typhoons crossing each region from 2008 to 2018

ID	ROI	Count of crossing typhoon	Total track nodes	Maximum crossing time/h	Mean absolute surge level/cm
1	18°–22°N, 108°–112°E	48	1 135	113	31.995 3
2	16°–24°N, 106°–114°E	76	3 471	188	26.297 2
3	14°–26°N, 104°–116°E	88	5 392	197	25.181 2
4	12°–26°N, 102°–118°E	112	7 801	205	24.065 6
5	10°–26°N, 102°–120°E	133	10 022	259	23.055 5
6	8°–26°N, 102°–122°E	137	12 015	296	22.267 9

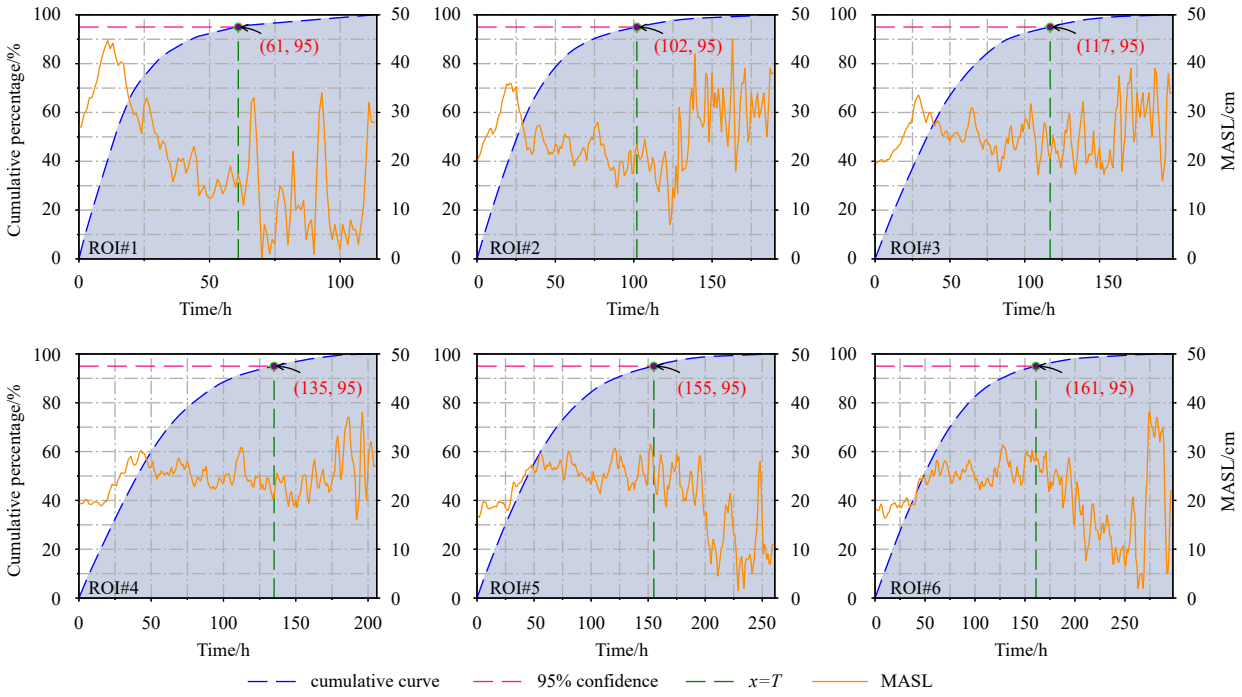

Fig. 5. Time-sharing mean absolute surge level (MASL) curves and cumulative percentages of typhoon track nodes.

A total of 112 typhoons crossed ROI#4 (as shown in Table 1). Considering the fact that no data gaps exist, the dataset should have a total of 15 120 records. According to this principle, the above selected hours were used as the initial forecasting time to sample the dataset. The data between 2008 and 2016 were divided into a training dataset and validation dataset at a ratio of 8:2, and data from 2017 to 2018 were organised as the testing dataset. The training dataset was used to adjust model parameters, such as the weights and bias of the neural network. The validation dataset was used to verify whether a model is undertrained or overfitting. The test dataset was used to evaluate model performance.

After sampling, the training and validation dataset had 11 489 samples, and the testing dataset had 3 191 samples. The details of the sample dataset and its comparison with the original dataset are given in Table 2.

As presented in Table 2, the amount of generated dataset after sampling has been significantly reduced, while the extreme value characteristics of the original data are basically retained. Compared with the original dataset, it has higher mean absolute values of SL, U10, and V10, indicating that the sample data have a higher proportion of positive high values and negative low values. The higher kurtosis indicates that the "hot tails" feature is more noticeable in the probability density distribution, and the

Table 2. Details of sample data and its comparison with original data

Index	Sample data						Original data					
	Training and validation			Testing			2008–2016			2017–2018		
	SL /cm	U10 /(m·s^{-1})	V10 /(m·s^{-1})	SL /cm	U10 /(m·s^{-1})	V10 /(m·s^{-1})	SL /cm	U10 /(m·s^{-1})	V10 /(m·s^{-1})	SL /cm	U10 /(m·s^{-1})	V10 /(m·s^{-1})
Samples	11 489			3 191			75 588			17 166		
Max	215	16.0	18.4	76	11.3	7.9	215	16.0	18.4	76	11.3	7.9
Min	–67	–17.3	–17.8	–27	–12.9	–10.8	–67	–17.3	–17.8	–39	–12.9	–10.8
MAV	21.02	3.73	2.78	23.25	3.66	2.46	17.84	3.25	2.53	19.68	3.40	2.41
Kurtosis	11.98	0.43	1.16	1.12	0.18	–0.41	5.88	1.07	–0.05	0.48	1.77	–0.59

Note: U10, 10 m U wind component; V10, 10 m V wind component; MAV, mean absolute value; SL, surge level.

probability of an extreme value distribution at both ends is greater. Comparing the training dataset and testing dataset, the former has a larger threshold and higher kurtosis, indicating that Xiuying Station has a low frequency of strong storm surges during 2017 and 2018, and more SL values are distributed around the mean value. The differences between the training dataset and testing dataset pose a certain challenge to the applicability of prediction models.

3.3 *Model construction*

One problem with the construction of time series prediction models is the determination of the number of feature inputs. After data pre-processing and sample data generation, the initial forecasting hours were extracted, and input features were defined as sequences of previous observations before these hours. Deeper neural networks with longer input sequences intend to obtain more accurate prediction results. In multi-step time series forecasting, some sequence-to-sequence models have utilised long-term sequence data to predict multiple values at once. In this study, we did not adopt this concept because the missing of observation data are frequent, especially when a typhoon occurs. If the length of the input sequence is L, the observation missing of 1 h will cause L hours to be unpredictable. Based on the trial-and-error approach as well as extant studies performed by Nitsure et al. (2014) and El-Diasty et al. (2018), this study intended to use 24 as the number of feature inputs. The input sequence is organised as follows:

$$S_{t+i} = f(S_t, S_{t-i}, S_{t-2i}, S_{t-3i}, \cdots, S_{t-23i}), \tag{8}$$

where f is neural network model; S_{t+i} is the predicted value at time $t+i$; S_t is the observed value at time t, including SL, U10, and V10; t is the initial forecasting hour; and i is the forecasting horizon.

Another problem of model construction is model selections and parameter settings. The aim of this study was the application of CNN and LSTM in SL prediction, so CNN, LSTM, LSTM–CNN, and CNN–LSTM models were constructed. To analyse and compare the model results, SVR and MLP models were selected as references. The SVR model, a regression application of the support vector machine (SVM) by introducing an alternative loss function that is modified to include a distance measure, is a superior model that has been verified by numerous practical implementations. MLP is a core method in the deep leaning framework. By adding multiple hidden layers, MLP can deeply tap the potential connections between the input layer and output layer

Table 3. Hyper-parameter details of the six models

Model	Hyper-parameter	Value	Reason
SVR	shape input layer nodes	(None, 72)	shape of feature inputs
	kernel function	RBF	a competitive kernel function
	parameters adjustment method	GridSearchCV	an exhaustive search method in "sklearn"
MLP CNN LSTM	shape of input layer	(None, 72) for MLP, (None, 24, 3) for CNN, (None, 3, 24) for LSTM	shape of feature inputs
	number of 1st hidden layer nodes	128	common value [16, 32, 64, 128, 256]
	number of 2nd hidden layer nodes	32	common value [8, 16, 32, 64, 128]
	size of batch	256	common value [64, 128, 256]
	early stopping patience	5	a common value
	early stopping minimum delta	1×10^{-4}	minimum gap of loss in "Keras"
	early stopping loop count	7 for 1 h model, 5 for others	common value [1, 3, 5, 7]
LSTM–CNN	shape of input layer	(None, 3, 24)	shape of feature inputs
	number of 1st LSTM layer nodes	128	same as LSTM
	number of 2nd LSTM layer nodes	32	same as LSTM
	number of 1st CNN layer nodes	128	same as CNN
	number of 2nd CNN layer nodes	32	same as CNN
	size of batch	256	same as CNN
	early stopping patience	5	same as CNN
	early stopping minimum delta	1×10^{-4}	same as CNN
	early stopping loop count	7 for 1 h model, 5 for others	common value [1, 3, 5, 7]
CNN–LSTM	shape of input layer	(None, 24, 3)	shape of feature inputs
	number of 1st CNN layer nodes	128	same as CNN
	number of 2nd CNN layer nodes	32	same as CNN
	number of 1st LSTM layer nodes	128	same as LSTM
	number of 2nd LSTM layer nodes	32	same as LSTM
	size of batch	256	same as LSTM
	early stopping patience	5	same as LSTM
	early stopping minimum delta	1×10^{-4}	same as LSTM
	early stopping loop count	7 for 1 h model, 1 for 6 h model, 5 for others	common value [1, 3, 5, 7]

to make better predictions. The hyper-parameters of all models are listed in Table 3.

The settings of all model parameters were predominantly determined by trial-and-error among common values. Table 3 lists the values of hyper-parameters for different models and the reasons why they have been selected. For hyper-parameters having multiple available common values, the prediction errors were calculated for all possible combinations using training data, and the hyper-parameters leading to the lowest prediction errors were selected to build models. The same parameters are equal in these models to ensure fairness of comparison, for example, the number of CNN layer nodes in CNN model equals to those in LSTM-CNN and CNN-LSTM models. In addition to the parameters listed in Table 3, there are still others that must be illustrated in the training stage. For example, the epoch number for training is 200; all optimizers in neural networks are "Adam", and activations are "Relu". No CNN layers, including those in the LSTM–CNN and CNN–LSTM models, have max-pooling or dropout layer behind them, kernel size is (2, 1), and stride is 1. All LSTM layers, including the layers in LSTM–CNN and CNN–LSTM models, are followed by a recurrent dropout layer. In LSTM–CNN and CNN–LSTM, the input dimensions were permuted before the concatenation of the two models. To avoid over-fitting, the early stopping method was adopted at the minimum delta of 1×10^{-4}.

Based on the solution of the above two problems, the input data were organized to meet the needs of multi-step forecasting, and the six prediction models were constructed for comparison. Figure 6 shows the data organization and network structures of models. The original inputs of all six models are sorted out to three-dimensional tensors of shape (None, 24, 3), where the first dimension is the number of samples, the second dimension is the length of time sequences, and the third dimension is the number of variables (SL, U10, V10). For each step (i=1, 2, 4, 6) of forecasting, different sample datasets are generated, and different prediction models are trained, respectively. In addition, the network structures of SVR, MLP, CNN, LSTM, LSTM-CNN and

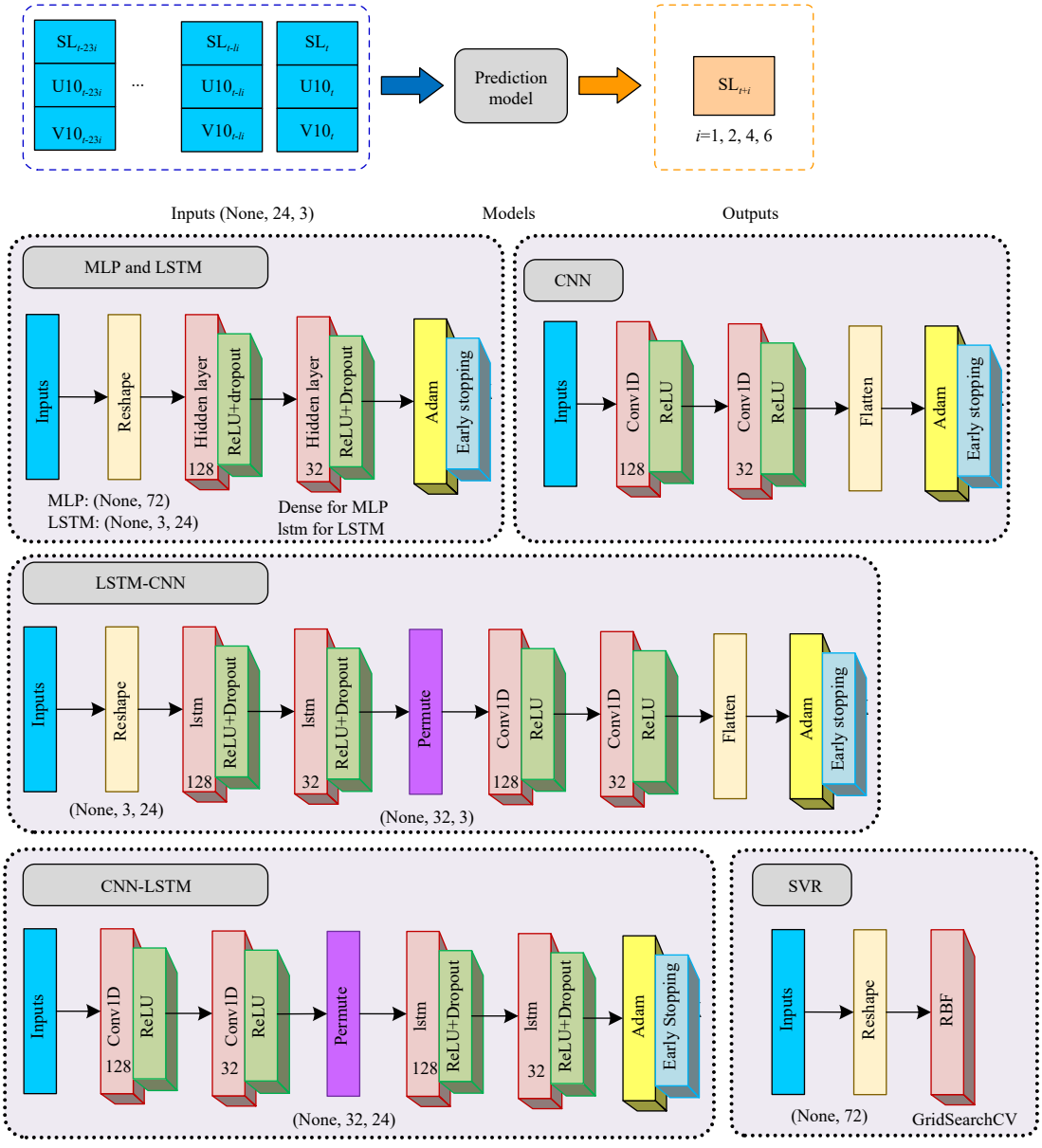

Fig. 6. Data organization for multi-step predictions and architectures of selected models.

CNN-LSTM are depicted in detail in Fig. 6, among which MLP and LSTM are presented together.

All models are implemented in Python, utilizing the Python libraries sklearn and Keras. As the results of each run of SVR are the same, the model runs were performed only once. MLP, CNN, LSTM, LSTM–CNN, and CNN–LSTM were iterated to optimize weights, and there are random numbers in the models. The results of each run are different, so these five models ran 10 times and the averages were regarded as the final results.

4 Results and discussion

Table 4 summarises the results of the six competitive algorithms. As can be observed from the 1– 6 h ahead forecasts, CNN and LSTM performed better than SVR and MLP, and combinations of the CNN and LSTM networks achieved more accurate results than the individual CNN and LSTM. The best MAE results in the six models are indicated by the grey-filled cells. CNN–LSTM maintained the best performance at all four forecast horizons, but SVR, the MAE of which increased from 5.90 cm to 9.51 cm, achieved the worst results. The MAE values at the four steps using CNN decreased by 15.98%, 6.62%, 6.20%, and 7.41% against MLP, respectively; using CNN–LSTM, they decreased by 18.01%, 11.48%, 10.82%, and 12.35%. Comparing the individual CNN and LSTM models with their combined models, the average MAE values of LSTM–CNN and CNN–LSTM were 2.65 cm, 4.33 cm, 6.14 cm, and 6.83 cm at four respective steps, all of which are approximately 3% below those of CNN and LSTM.

To display the prediction results of different models more clearly, the SL prediction performances of 1 hour-ahead SVR, MLP, LSTM, and CNN–LSTM on testing data are plotted in the form of scatter charts (Fig. 7). The scatter plots have observed values on the *x*-axis and predicted values on the *y*-axis. The data

Table 4. Experimental results of six algorithms on testing data at multi-hour steps

Steps	Index	SVR	MLP	CNN	LSTM	LSTM–CNN	CNN–LSTM
1 h	CC	0.850 5	0.954 3	0.965 5	0.962 9	0.965 8	0.966 1
	MAE/cm	5.895 8	3.208 3	2.695 6	2.795 7	2.668 2	2.630 3
	RMSE/cm	7.544 2	4.212 4	3.531 1	3.622 0	3.478 4	3.463 1
2 h	CC	0.771 6	0.883 0	0.900 2	0.902 9	0.904 6	0.908 6
	MAE/cm	6.869 2	4.821 1	4.502 1	4.455 6	4.383 2	4.267 6
	RMSE/cm	8.868 5	6.291 9	5.832 4	5.767 6	5.722 3	5.612 5
4 h	CC	0.671 5	0.733 4	0.780 7	0.789 2	0.794 2	0.799 1
	MAE/cm	7.644 9	6.831 0	6.407 1	6.368 3	6.191 9	6.091 8
	RMSE/cm	10.047 3	9.021 9	8.335 1	8.196 2	8.063 4	7.982 1
6 h	CC	0.460 8	0.658 8	0.705 6	0.718 3	0.720 4	0.736 1
	MAE/cm	9.507 3	7.687 7	7.118 4	6.935 3	6.921 1	6.738 0
	RMSE/cm	12.060 8	9.813 2	9.132 7	8.945 3	8.914 7	8.709 5

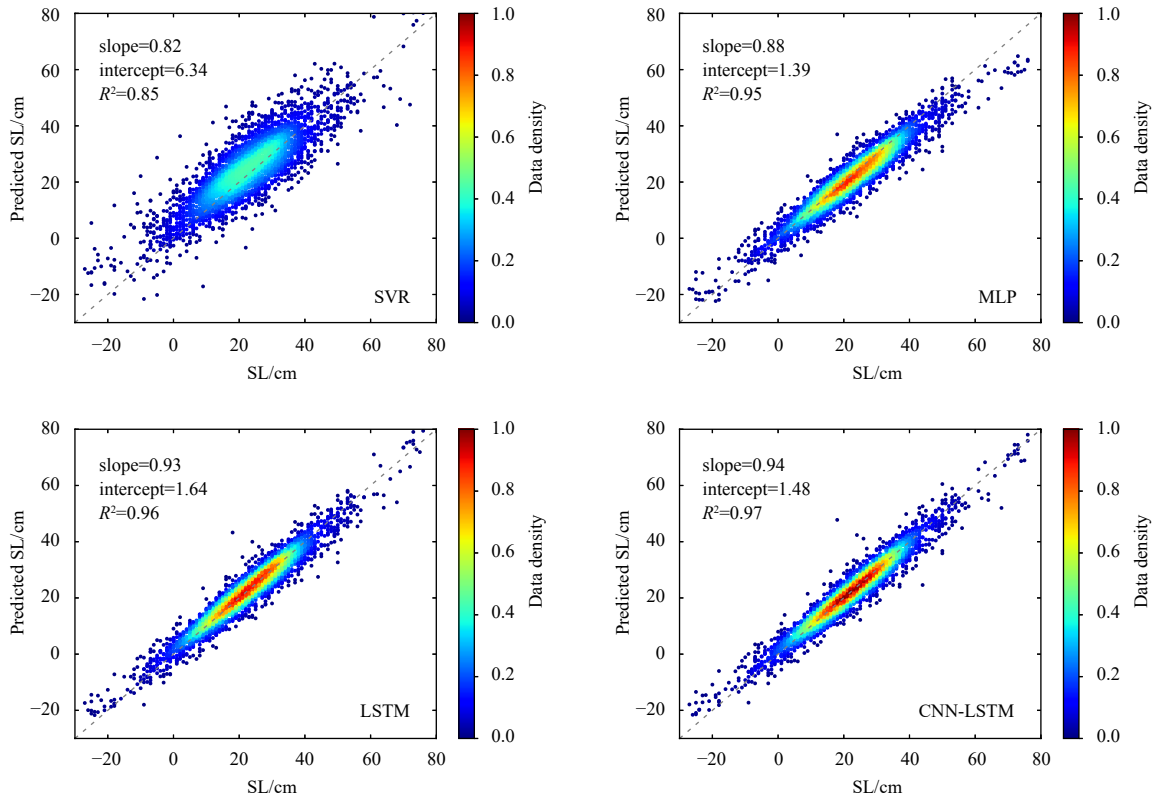

Fig. 7. Scatter plots of observed SLs and 1 h ahead predicted SLs of SVR, MLP, LSTM, and CNN–LSTM.

pairs are colour coded by data density, which was calculated by kernel density estimation using Gaussian functions, and all results are normalised in the range [0, 1] using unified extremum. The standard lines ($y = x$) are represented by dashed lines for better observation of the distribution of scatters.

Comparing the scatter plots of the four models, the CNN–LSTM model showed the best agreement with the observed SL with a slope of 0.94, intercept of 1.48, and R^2 of 0.97. The distribution of the SVR scatter is the most dispersed. For MLP, the scatter distribution is relatively concentrated, but the fitted line deviates too much from the standard line (slope of 0.88). In particular, the predicted values are lower in the high-value area and higher in the low-value area. The slopes of the LSTM and CNN–LSTM fitted lines are closer to 1, and the R^2 values are higher. Comparing LSTM with CNN–LSTM, CNN–LSTM achieved a slimmer distribution form, higher data density, and the scatter is closer to the standard line in the high-value area; therefore, its prediction is more reasonable.

As shown in Table 3 and Fig. 7, CNN, LSTM, and their combined models have evident advantages in prediction accuracy over SVR and MLP, and the combined models are superior to individual CNN and LSTM, but these advantages are not significant. To effectively understand the forecast details of the models, line graphs of error distributions were drawn. Firstly, the observation values were arranged in ascending order, and the number of testing samples was equally divided into 10 blocks. Then, the MAE of each model for each block was calculated to obtain 1 h, 2 h, 4 h, and 6 h forecasting error distributions of MLP, CNN, LSTM, LSTM–CNN, and CNN–LSTM in ascending SL interval ranges, as shown in Fig. 8. Figure 8 shows that from 1 h to 6 h time steps, the forecasting errors for ascending data blocks roughly present U-shaped distributions, indicating that all models have higher forecasting errors on negative low and positive high SLs but have low errors near the mean. Secondly, the higher the forecasting horizons, the more obvious the U-shaped characteristics. With larger forecasting steps, the error growth at the U-shaped bottom is not large, but the errors increase obviously at both ends, demonstrating that the increases of model errors mainly occur in the prediction of data at both ends. For example, in the prediction of CNN, the errors of 4th–6th blocks at 2 h and 6 h ahead steps are approximately 4 cm and 5 cm, respectively, while the error increases from 6 cm to 13 cm for the 10th block. Finally, compared with the CNN and LSTM models, although the global MAEs of LSTM–CNN and CNN–LSTM are lower, the errors at the U-shaped bottom are higher in the 1 h, 4 h, and 6 h ahead predictions. Therefore, the main advantages of the combined models are mainly concentrated in the prediction of high and low SLs, which is largely affected by severe meteorological activities, such as typhoons.

Typhoon Khanum and Mangkhut were selected as cases to study the prediction accuracy of these models when typhoons occur. The tracks of Khanum and Mangkhut area shown in Fig. 4. Figures 9 and 10 are the results of different models predicting the 1 h and 2 h ahead SL hourly variations during these two typhoon periods, which were chosen as 12:00 on October 12th to 12:00 on October 17th, 2017 and 00:00 on September 15th to 00:00 on

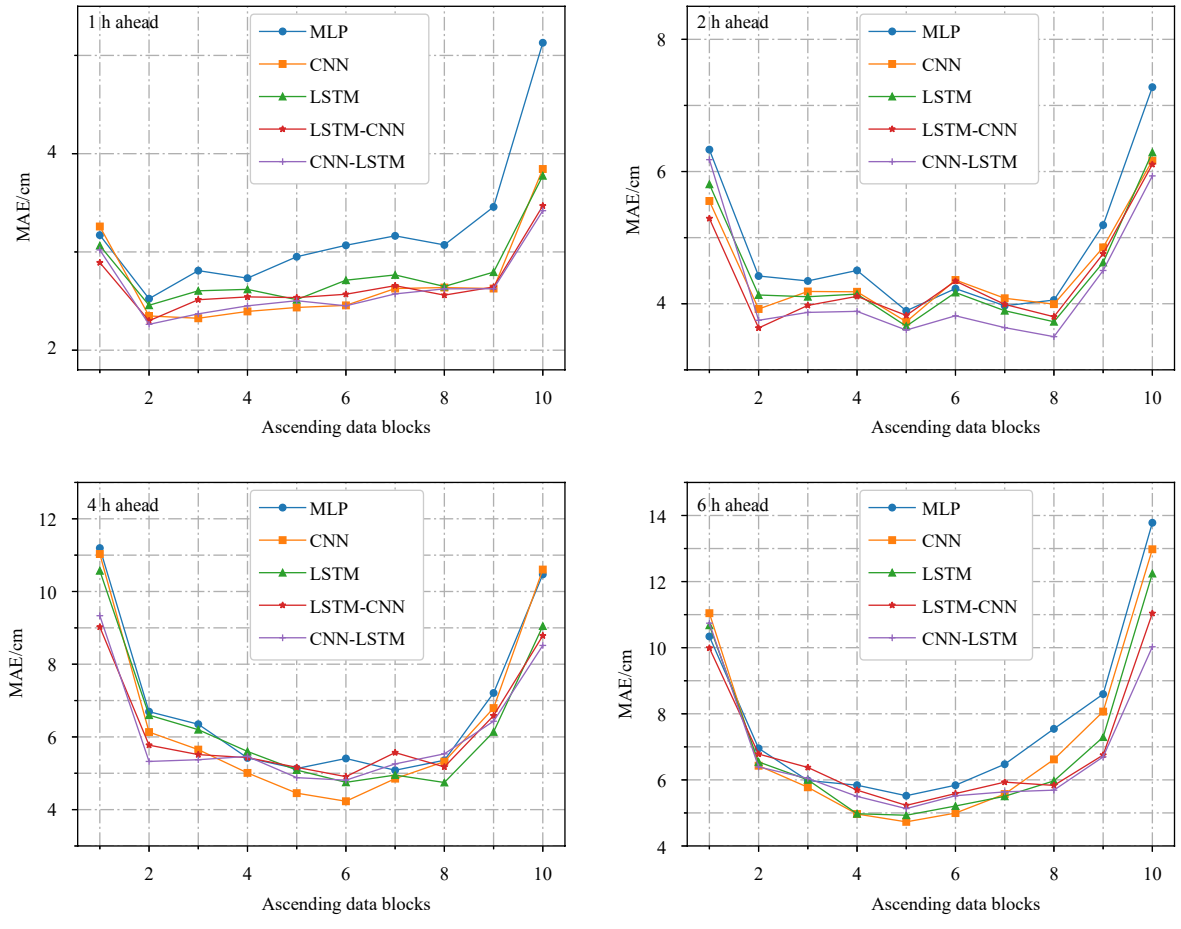

Fig. 8. Error distributions with ascending data blocks at 1 h, 2 h, 4 h, and 6 h ahead steps.

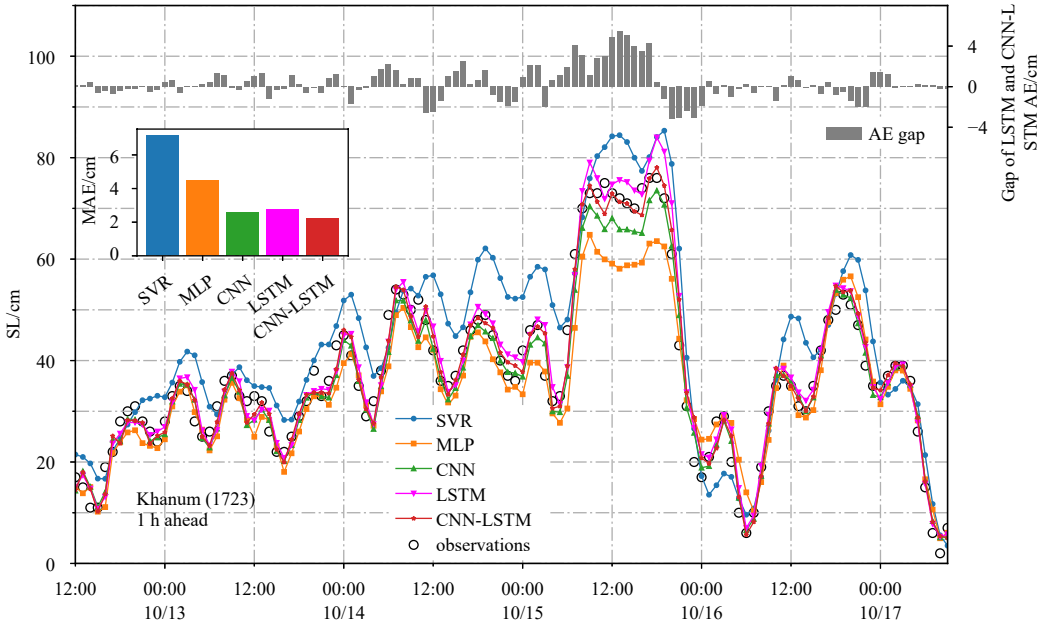

Fig. 9. 1 h ahead predicted hourly SLs of SVR, MLP, CNN, LSTM and CNN–LSTM.

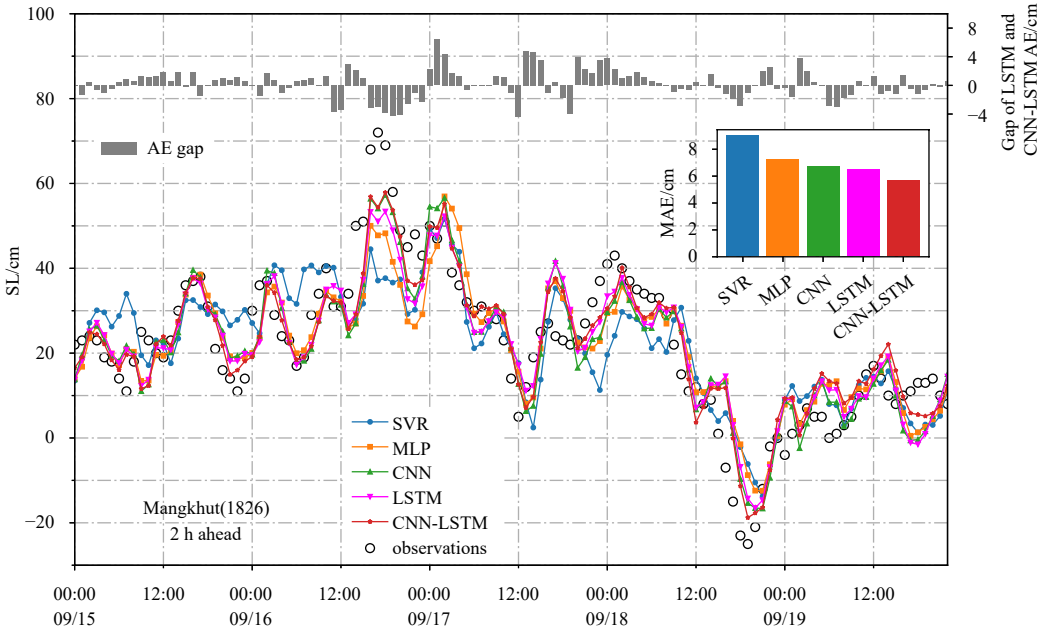

Fig. 10. 2 h ahead predicted hourly SLs of SVR, MLP, CNN, LSTM, and CNN–LSTM.

September 20th, 2018. To express the differences among these models clearly, representative results from only 5 models' experiments are selected for display, including SVR, MLP, CNN, LSTM and CNN–LSTM. LSTM-CNN was not chosen in the two figures, for the reason that LSTM-CNN had smaller MAEs when compared with CNN or LSTM, but its prediction errors were similar (only a little bit larger) to CNN-LSTM (as shown in Table 4).

As shown in Figs 9 and 10, CNN–LSTM achieved the best results, followed by CNN and LSTM, while SVR was the worst. In Fig. 9, except for SVR, the SL predictions of the other three models can fit the observations under 50 cm well, but MLP, CNN and LSTM tend to underestimate at peaks and overestimate at troughs. The MAE histogram in the figure clearly shows the differences between the forecast errors of the four models. To intuitively observe the difference between the predictions of CNN and CNN–LSTM, a grey bar chart at the top of the figure is used to display the hourly gap between the absolute errors (AEs) of CNN and CNN–LSTM ($AE_{\mathrm{CNN}} - AE_{\mathrm{CNN-LSTM}}$). Evidently, most of the gaps are greater than 0, and the positive values are higher, which indicates that CNN–LSTM is definitely superior to CNN. More importantly, CNN–LSTM enhances the forecast quality near the peaks, where the prediction task is most challenging and meaningful. A similar conclusion can be drawn from Fig. 10, except that the prediction errors of all models are larger.

Figures 9 and 10 show the 1 h and 2 h ahead hourly predictions of the four models, but they do not explain their comprehensive performances for all forecasting horizons. Therefore, the MAEs predicted by the five models (except SVR) at 1 h, 2 h, 4 h,

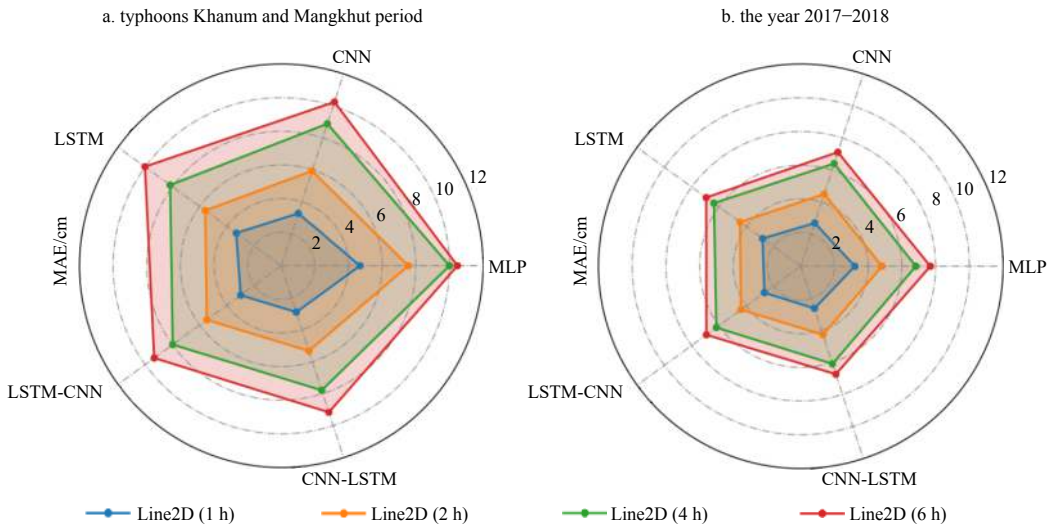

Fig. 11. Radar charts of different modes for multi-step ahead SL predictions for the period of typhoons Khanum and Mangkhut and the year from 2017 to 2018.

and 6 h ahead are plotted as a radar chart, as shown in Fig. 10. The entire forecast results from 2017 to 2018 are also drawn for comparison.

Figure 11 illustrates that the prediction errors of all models during typhoons at different steps are greater than the overall errors from 2017 to 2018, but CNN and LSTM, and especially their combinations, show greater superiority. During the period of typhoons Khanum and Mangkhut, the average MAEs of CNN and LSTM were 29.72%, 23.80%, 14.81%, and 3.29% smaller than those of MLP at the four respective forecasting horizons, while the corresponding reductions of the combined models were 37.21%, 28.54%, 21.20%, and 11.71%. Meanwhile, the forecasting advantages of the combined models during typhoons are more obvious than usual. For example, in the period of 2017 and 2018, the MAEs of CNN at the four steps were 2.80 cm, 4.46 cm, 6.41 cm, and 7.69 cm, respectively, and the errors were reduced by 2.42%, 5.21%, 4.92%, and 5.34% in CNN–LSTM. In the period of the two typhoons, the MAEs of CNN were 3.27 cm, 5.93 cm, 8.88 cm, and 10.24 cm, and CNN–LSTM outperformed CNN by 11.39%, 10.13%, 12.29%, and 10.54%, all of which are significant improvements.

In addition, some limitations of our study should be mentioned. As the forecasting step number grows, the errors during typhoon periods increase more than usual. The MAEs of CNN and LSTM in Fig. 11a at the 6 h ahead step are close to that of MLP, which are both above 10 cm, but the average observed SL during the period was only 33.08 cm. Meanwhile, in Fig. 11b, the errors of all models at 6 h ahead are similar to those at 4 h. This does not mean that the 6 h ahead predictions are accurate, but it may reflect a problem that the prediction errors are close to a reasonable upper limit. If the forecasting horizon were extended to 12 h or 24 h, the prediction results would be even less credible. Therefore, the methods proposed in this paper may not achieve satisfactory accuracy in mid- to long-term SL forecasting. There could be two reasons for this. (1) The input features are organised as $[S_t, S_{t-i}, S_{t-2i}, S_{t-3i}, \cdots, S_{t-23i}]$ (S is observed time series, t is initial forecasting hour, and i is forecast horizon), and the bigger is i, the greater the difference among the input features, which may no longer effectively reflect the variation laws of storm surges. (2) The sample data only include the time series observed SL and wind data of the predicted site and cannot represent the impact of the surrounding environment on the changes of tidal level on a spatial scale. Nevertheless, the combinations of CNN and LSTM could still show their advantages in predictions.

5 Conclusions

In this study, we applied CNN, LSTM, and their sequential combinations in short-term multi-step ahead storm surge level predictions using observed SL and wind time series data as input. The major objectives of this study included the development and evaluation of methods for improving SL forecasting accuracy via comparison against SVR and MLP methods. The Xiuying Station data were used to implement and validate the proposed models. Firstly, a method of sample generation was proposed based on data statistics to improve the quality of sample data and reduce the amount of training data. Secondly, the sample data were organised as model inputs by using sliding time windows with different step sizes, and all data were standardised and divided into training, validation, and testing datasets before model training. Finally, the architecture of each model was determined based on a trial-and-error approach and experiences gained from previous studies.

Experimental results demonstrate that the CNN and LSTM models evidently outperform the SVR and MLP models in terms of forecasting accuracy. Moreover, the combinations of CNN and LSTM enhance the performances of single CNN and LSTM mostly by 4%–6%. The error distribution analysis found that the forecasting quality improvement of combined models mainly concentrates on positive high and negative low SLs, where the prediction task is most challenging and meaningful. From a case study of two typhoons that caused the most severe storm surges in the testing period, the combined models outperform individual models by over 10% at all four forecasting steps.

The models proposed in this paper can be implemented in real time and can be applied to forecasting SLs for other tide-gauge stations with tide and wind observations in situations that do not require physical and dynamical modelling. However, there are still some limitations of our study, as mentioned at the end of Section 4. Future works will mainly be conducted from the following two aspects: (1) applying more hybrid models, such as

bidirectional LSTM, self-attention-based LSTM, and their combinations with CNN, to achieve higher prediction accuracy, and (2) introducing the observed data of adjacent stations or grid data into the models considering that sea water level, wind, and other observable elements have continuity in geographic space.

Acknowledgements

We express appreciations to colleagues in the laboratory for their constructive suggestions. We also thank the anonymous reviewers for their constructive comments.

References

Araya I A, Valle C, Allende H. 2020. A Multi-Scale Model based on the Long Short-Term Memory for day ahead hourly wind speed forecasting. Pattern Recognition Letters, 136: 333–340, doi: 10.1016/j.patrec.2019.10.011

Balas C E, Koç M L, Tür R. 2010. Artificial neural networks based on principal component analysis, fuzzy systems and fuzzy neural networks for preliminary design of rubble mound breakwaters. Applied Ocean Research, 32(4): 425–433, doi: 10.1016/j.apor.2010.09.005

Chang H K, Lin L C. 2006. Multi-point tidal prediction using artificial neural network with tide-generating forces. Coastal Engineering, 53(10): 857–864, doi: 10.1016/j.coastaleng.2006.05.001

Chen Yong, Zhang Shuai, Zhang Wenyu, et al. 2019. Multifactor spatio-temporal correlation model based on a combination of convolutional neural network and long short-term memory neural network for wind speed forecasting. Energy Conversion and Management, 185(4): 783–799

Cox D T, Tissot P, Michaud P. 2002. Water level observations and short-term predictions including meteorological events for entrance of Galveston bay, Texas. Journal of Waterway, Port, Coastal, and Ocean Engineering, 128(1): 21–29, doi: 10.1061/(ASCE)0733-950X(2002)128:1(21)

Dong Na, Chang Jianfang, Wu Aiguo, et al. 2020. A novel convolutional neural network framework based solar irradiance prediction method. International Journal of Electrical Power & Energy Systems, 114(3): 105411

El-Diasty M, Al-Harbi S, Pagiatakis S. 2018. Hybrid harmonic analysis and wavelet network model for sea water level prediction. Applied Ocean Research, 70(2): 14–21

Hochreiter S, Schmidhuber J. 1997. Long Short-term Memory. Neural Computation, 9(8): 1735–1780, doi: 10.1162/neco.1997.9.8.1735

Imani M, Kao H C, Lan W H, et al. 2018. Daily sea level prediction at Chiayi coast, Taiwan using extreme learning machine and relevance vector machine. Global and Planetary Change, 161: 211–221, doi: 10.1016/j.gloplacha.2017.12.018

Ishida K, Tsujimoto G, Ercan A, et al. 2020. Hourly-scale coastal sea level modeling in a changing climate using long short-term memory neural network. Science of the Total Environment, 720: 137613, doi: 10.1016/j.scitotenv.2020.137613

Kao I F, Zhou Yanlai, Chang L C, et al. 2020. Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting. Journal of Hydrology, 583: 124631, doi: 10.1016/j.jhydrol.2020.124631

Karimi S, Kisi O, Shiri J, et al. 2013. Neuro-fuzzy and neural network techniques for forecasting sea level in Darwin Harbor, Australia. Computers & Geosciences, 52: 50–59

Kim S, Matsumi Y, Pan Shunqi, et al. 2016. A real-time forecast model using artificial neural network for after-runner storm surges on the Tottori coast, Japan. Ocean Engineering, 122: 44–53, doi: 10.1016/j.oceaneng.2016.06.017

Kim S W, Melby J A, Nadal-Caraballo N C, et al. 2015. A time-dependent surrogate model for storm surge prediction based on an artificial neural network using high-fidelity synthetic hurricane modeling. Natural Hazards, 76: 565–585, doi: 10.1007/s11069-014-1508-6

Kim S, Pan Shunqi, Mase H. 2019. Artificial neural network-based storm surge forecast model: Practical application to Sakai Minato, Japan. Applied Ocean Research, 91: 101871, doi: 10.1016/j.apor.2019.101871

Lee T L. 2006. Neural network prediction of a storm surge. Ocean Engineering, 33(3–4): 483–494, doi: 10.1016/j.oceaneng.2005.04.012

Lee T L. 2008. Back-propagation neural network for the prediction of the short-term storm surge in Taichung harbor, Taiwan. Engineering Applications of Artificial Intelligence, 21(1): 63–72, doi: 10.1016/j.engappai.2007.03.002

Lee T L. 2009. Predictions of typhoon storm surge in Taiwan using artificial neural networks. Advances in Engineering Software, 40(11): 1200–1206, doi: 10.1016/j.advengsoft.2007.06.005

Li Youru, Zhu Zhenfeng, Kong Deqiang, et al. 2019. EA-LSTM: Evolutionary attention-based LSTM for time series prediction. Knowledge-Based Systems, 181: 104785, doi: 10.1016/j.knosys.2019.05.028

Liang S X, Li M C, Sun Z C. 2008. Prediction models for tidal level including strong meteorologic effects using a neural network. Ocean Engineering, 35(7): 666–675, doi: 10.1016/j.oceaneng.2007.12.006

Liu Hui, Mi Xiwei, Li Yanfei. 2018. Smart deep learning based wind speed prediction model using wavelet packet decomposition, convolutional neural network and convolutional long short term memory network. Energy Conversion and Management, 166(3): 120–131

Mel R, Viero D P, Carniello L, et al. 2014. Simplified methods for real-time prediction of storm surge uncertainty: The city of Venice case study. Advances in Water Resources, 71: 177–185, doi: 10.1016/j.advwatres.2014.06.014

Mok K M, Lai U H, Hoi K I. 2016. Development of an adaptive Kalman filter-based storm tide forecasting model. Journal of Hydrodynamics, 28(6): 1029–1036, doi: 10.1016/S1001-6058(16)60707-2

Nitsure S P, Londhe S N, Khare K C. 2014. Prediction of sea water levels using wind information and soft computing techniques. Applied Ocean Research, 47: 344–351, doi: 10.1016/j.apor.2014.07.003

Oh S L, Ng E Y K, Tan R S, et al. 2018. Automated diagnosis of arrhythmia using combination of CNN and LSTM techniques with variable length heart beats. Computers in Biology and Medicine, 102: 278–287, doi: 10.1016/j.compbiomed.2018.06.002

Pak U, Ma Jun, Ryu U, et al. 2020. Deep learning-based PM2.5 prediction considering the spatiotemporal correlations: A case study of Beijing, China. Science of the Total Environment, 699: 133561, doi: 10.1016/j.scitotenv.2019.07.367

Petersen N C, Rodrigues F, Pereira F C. 2019. Multi-output bus travel time prediction with convolutional LSTM neural network. Expert Systems with Applications, 120: 426–435, doi: 10.1016/j.eswa.2018.11.028

Qing Xiangyun, Niu Yugang. 2018. Hourly day-ahead solar irradiance prediction using weather forecasts by LSTM. Energy, 148: 461–468, doi: 10.1016/j.energy.2018.01.177

Rajasekaran S, Gayathri S, Lee T L. 2008. Support vector regression methodology for storm surge predictions. Ocean Engineering, 35(16): 1578–1587, doi: 10.1016/j.oceaneng.2008.08.004

Sadaei H J, de Lima e Silva P C, Guimarães F G, et al. 2019. Short-term load forecasting by using a combined method of convolutional neural networks and fuzzy time series. Energy, 175: 365–377, doi: 10.1016/j.energy.2019.03.081

Sagheer A, Kotb M. 2019. Time series forecasting of petroleum production using deep LSTM recurrent networks. Neurocomputing, 323: 203–213, doi: 10.1016/j.neucom.2018.09.082

Shi Xingjian, Chen Zhourong, Wang Hao, et al. 2015. Convolutional LSTM Network: a machine learning approach for precipitation nowcasting. In: Proceedings of the 28th International Conference on Neural Information Processing Systems. Cambridge, MA, United States: MIT Press, 802–810

Song Xuanyi, Liu Yuetian, Xue Liang, et al. 2020. Time-series well performance prediction based on Long Short-Term Memory

(LSTM) neural network model. Journal of Petroleum Science and Engineering, 186: 106682, doi: 10.1016/j.petrol.2019.106682

Sztobryn M. 2003. Forecast of storm surge by means of artificial neural network. Journal of Sea Research, 49(4): 317–322, doi: 10.1016/S1385-1101(03)00024-8

Wang Kang, Li Kenli, Zhou Liqian, et al. 2019. Multiple convolutional neural networks for multivariate time series prediction. Neurocomputing, 360: 107–119, doi: 10.1016/j.neucom.2019.05.023

Wang Bao, Wang Bin, Wu Wenzhou, et al. 2020. Sea-water-level prediction via combined wavelet decomposition, neuro-fuzzy and neural networks using SLA and wind information. Acta Oceanologica Sinica, 39(5): 157–167, doi: 10.1007/s13131-020-1569-1

Xiao Changjiang, Chen Nengcheng, Hu Chuli, et al. 2019. Short and mid-term sea surface temperature prediction using time-series satellite data and LSTM-AdaBoost combination approach. Remote Sensing of Environment, 233: 111358, doi: 10.1016/j.rse.2019.111358

Yin Jianchuan, Wang Nini. 2016. An online sequential extreme learning machine for tidal prediction based on improved Gath-Geva fuzzy segmentation. Neurocomputing, 174: 85–98, doi: 10.1016/j.neucom.2015.02.094

You S H, Seo J W. 2009. Storm surge prediction using an artificial neural network model and cluster analysis. Natural Hazards, 51(1): 97–114, doi: 10.1007/s11069-009-9396-x

Zhang Zeguo, Yin Jianchuan, Wang Nini, et al. 2017. A precise tidal prediction mechanism based on the combination of harmonic analysis and adaptive network-based fuzzy inference system model. Acta Oceanologica Sinica, 36: 94–105

Zhao Xinyu, Jiang Na, Liu Jinfu, et al. 2020. Short-term average wind speed and turbulent standard deviation forecasts based on one-dimensional convolutional neural network and the integrate method for probabilistic framework. Energy Conversion and Management, 203: 112239, doi: 10.1016/j.enconman.2019.112239