



From weather data to river runoff: using spatiotemporal convolutional networks for discharge forecasting

Florian Börgel, Sven Karsten, Karoline Rummel, and Ulf Gräwe

Leibniz Institute for Baltic Sea Research Warnemünde, Rostock, Germany

Correspondence: Florian Börgel (florian.boergel@io-warnemuende.de)

Received: 28 August 2024 – Discussion started: 11 October 2024

Revised: 21 January 2025 – Accepted: 2 February 2025 – Published: 27 March 2025

Abstract. The quality of river runoff determines the quality of regional climate projections for coastal oceans or other estuaries. This study presents a novel approach to river runoff forecasting using convolutional long short-term memory (ConvLSTM) networks. Our method accurately predicts daily runoff for 97 rivers within the Baltic Sea catchment by modeling runoff as a spatiotemporal sequence defined by atmospheric forcing. The ConvLSTM model predicts river runoff with an accuracy of $\pm 5\%$ when compared to the hydrological model. Compared to more complex process-based hydrological models, ConvLSTM networks offer fast processing times and easy integration into climate models, demonstrating their potential as a powerful tool for climate simulation and water resource management.

1 Introduction

River runoff is a key component of the global water cycle as it comprises about one-third of the precipitation over land areas (Hagemann et al., 2020), making accurate runoff forecasting essential for effective water resource management, particularly over extended periods (Fang et al., 2019; Tan et al., 2018). In addition to short-term forecasting, long-term projections of river runoff are vital for climate change studies, projecting flooding and droughts over global and river basins (Cook et al., 2020). These studies calculate river runoff using a land model incorporating a hydrological model within a coupled Earth system model (ESM) (Wang et al., 2022). In the absence of a fully coupled ESM, river runoff as input for ocean models can be created using hydrological models such as Hydrological Discharge (HD) (Hagemann et al., 2020) or HYdrological Predictions for the Environment (HYPE)

(Lindström et al., 2010). Hydrological models represent a process-based approach where the water balance is calculated using hydrological processes (e.g., snow, glaciers, soil moisture, or groundwater contribution). These models are complex forecasting tools that are widely utilized, such as high-resolution multi-basin models applied across Europe (Hundecha et al., 2016).

The second approach to projecting river runoff employs data-driven models, such as when calculating river runoff as the difference between precipitation and evaporation over a catchment area, with an integrated statistical correction (Meier et al., 2012). With the recent rise in machine learning in climate research, various data-driven model architectures have been explored for river runoff forecasting. Common approaches include feed-forward artificial neural networks; support vector machines; adaptive neuro-fuzzy inference systems; and, notably, long short-term memory (LSTM) neural networks. LSTM networks have gained traction for long-term hydrological forecasting due to their excellent performance (Humphrey et al., 2016; Huang et al., 2014; Ashrafi et al., 2017; Liu et al., 2020; Fang and Shao, 2022; Kratzert et al., 2018). LSTM networks, first introduced by Hochreiter and Schmidhuber (1997), are an evolution of classical recurrent neural networks (Sherstinsky, 2020). A significant advantage of an LSTM network's architecture is the memory cell's ability to retain gradients. This mechanism addresses the vanishing gradient problem, where, as input sequences elongate, the influence of initial stages becomes harder to capture, causing gradients of early input points to approach zero. LSTM networks have shown stability and efficacy in sequence-to-sequence predictions. However, a limitation of LSTM networks is their inability to effectively capture two-dimensional structures, which is an area where convolutional

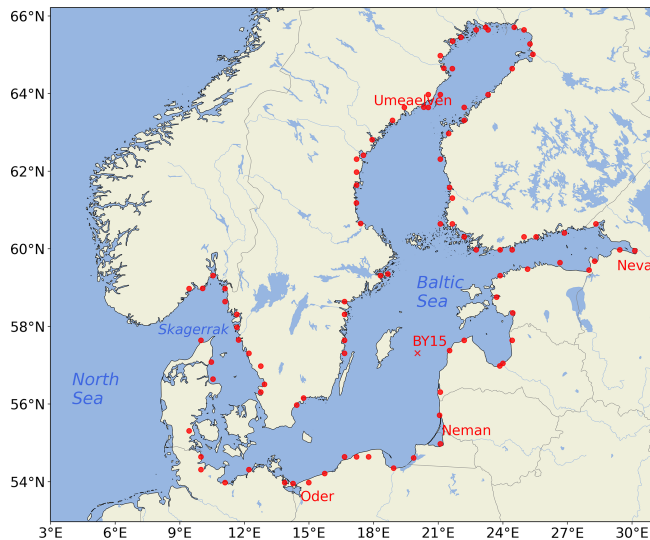


Figure 1. Map of the Baltic Sea region. The red dots represent the locations of the major rivers that flow into the Baltic Sea, as represented in the E-HYPE hydrological model. The annotation BY15 marks the chosen validation station situated in the central Baltic Sea, which is used for validating the regional ocean model. The four remaining annotations (red) indicate the positions of the rivers that will be evaluated in detail.

neural networks (CNNs) excel (Höhlein et al., 2020). Recognizing this limitation, Shi et al. (2015) proposed a convolutional LSTM (ConvLSTM) architecture that combines the strengths of both LSTM networks and CNNs. In practical applications, the combination of LSTM networks and CNNs in the form of ConvLSTM models allowed for improvement of the accuracy of precipitation nowcasting (Shi et al., 2015), flood forecasting (Moishin et al., 2021), and river runoff forecasting (Ha et al., 2021; Zhu et al., 2023).

We use the Baltic Sea catchment as an example to illustrate our approach. Although the methodology we propose is universally applicable across various geographical regions, the Baltic Sea is a challenging region due to its unique hydrological characteristics, which are nearly decoupled from the open ocean (see Fig. 1) (Meier and Döscher, 2002). Freshwater enters the Baltic Sea through river runoff or positive net precipitation (precipitation minus evaporation) over the sea surface. The net precipitation accounts for 11 % and the river input for 89 % of the total freshwater input (Meier and Döscher, 2002). Consequently, the Baltic's sea surface salinity (SSS) is largely driven by freshwater supply from rivers.

SSS is an essential variable for the marine ecosystem in the Baltic Sea, as most species are adapted to either marine or freshwater conditions. Therefore, biases in the SSS's spatial and temporal variability significantly impact primary production and fish biomass (Kniebusch et al., 2019). Accurate modeling of the Baltic Sea relies heavily on the quality of the river input data used in the simulations. Analyzing nearly 100 years of observations, Winsor et al. (2001)

found that variations in freshwater storage are closely correlated with accumulated changes in river runoff. From 1902 to 1998, the average freshwater inflow into the Baltic amounted to $16\,115\text{ m}^3\text{ s}^{-1}$, with contributions from river runoff ($14\,085\text{ m}^3\text{ s}^{-1}$) and net precipitation over the Baltic Sea ($2030\text{ m}^3\text{ s}^{-1}$) (Meier and Kauker, 2003). This freshwater inflow results in a residence time of about 35 years for freshwater in the Baltic Sea (Omstedt and Hansson, 2006; Winsor et al., 2001; Meier and Kauker, 2003).

In this work, we demonstrate that ConvLSTM networks are a reliable method for predicting multiple rivers simultaneously, using only atmospheric forcing as input data, which allows us to emulate a hydrological model. The main focus of this work is on presenting a ConvLSTM architecture capable of predicting daily river runoff for 97 rivers across the Baltic Sea catchment. To train the network, we use data from the E-HYPE model (Väli et al., 2019) as reference output data and data from the UERRA project (Uncertainties in Ensembles of Regional Reanalyses, <http://www.uerra.eu/>, last access: 25 March 2025) as atmospheric forcing (Sect. 3). The quality of the model is evaluated in Sect. 4. The obtained results are discussed further and evaluated in Sect. 5.

2 Implemented model architecture

2.1 The main idea

We assume that the runoff at a specific point in time t for all N_r considered rivers collected in the vector $\mathbf{R}^t \in \mathbb{R}^{N_r}$ can be accurately approximated by a functional $\mathbf{M}(\{X^{tk}[x, y, \tau]\})$ of $k = 1, \dots, N_k$ atmospheric fields $X^{tk}[x, y, \tau]$ which are known for the past $\tau = 1, \dots, N_\tau$ time instances. This relationship is expressed as

$$\mathbf{R}^t = \mathbf{M}(\{X_k^t[x, y, \tau]\}). \quad (1)$$

The atmospheric fields are evaluated over a spatial domain $x = 1, \dots, N_x$ and $y = 1, \dots, N_y$, which is sufficiently large to capture all significant local and non-local contributions of the atmospheric fields to the river runoff. Typically, such mapping is realized using a hydrological model that simulates all relevant physical processes, transforming variables like precipitation and evaporation into river runoff. This process relies heavily on domain knowledge to tune all parameters to reasonable values. As an alternative, combining a ConvLSTM model with a subsequent fully connected (FC) neural network can adequately represent this functional. This approach eliminates the need for detailed knowledge of the involved processes and their modeling. Instead, these features can be “learned” by the network in an automated manner; i.e., all free parameters are optimized such that the network's output reproduces the data of the hydrological model with given atmospheric input fields. Our proposed network architecture is visualized in Fig. 2 and is described in detail

in the following sections. To provide an overview, we will discuss the main components of this architecture one by one.

2.2 The ConvLSTM network

2.2.1 The LSTM approach

Before turning directly to the ConvLSTM, the simpler architecture of the plain LSTM model is examined. This serves as a foundation for understanding the more complex ConvLSTM. The LSTM network, a specialized form of a recurrent neural network (RNN), is specifically designed to model temporal sequences $X^t[1], \dots, X^t[\tau], \dots, X^t[N_\tau]$ of N_τ input quantities $X^t[\tau] = (X_k^t[\tau]) \in \mathbb{R}^{N_k}$. This sequence is taken from a dataset given in the form of a time series $\{X^t\}$, with the endpoint coinciding with the specific element in the time series connected to time t , i.e., $X^t[N_\tau] \equiv X^t$; see Fig. 2. Here, N_k represents the number of input “channels”, which can correspond to different measurable quantities. The LSTM’s unique design allows it to adeptly handle long-range dependencies, setting it apart from traditional RNNs in terms of accuracy (see Fig. 3).

The critical component of the LSTM’s innovation is its cell state, $C^t[\tau] = (C_h^t[\tau]) \in \mathbb{R}^{N_h}$, which stores state information also referred to as long-term memory. This state information complements the so-called hidden state vector $H^t[\tau] = (H_h^t[\tau]) \in \mathbb{R}^{N_h}$, which is also known from simpler neural network architectures. In the case of the LSTM, the hidden state vector plays the role of short-term memory. The cell state and the hidden state are vectors where each element is associated with one of the N_h hidden layers labeled by h . These internal, artificial degrees of freedom enable the high adaptability of neural networks. The two state vectors are determined through several self-parameterized gates, all in the same vector space as $C^t[\tau]$; see Fig. 3 for a visualization.

In particular, the forget gate $F^t[\tau]$ defines the portion of the previous (long-term memory) cell state $C^t[\tau - 1]$ that should be kept (see the dashed cyan box therein). The input gate $I^t[\tau]$ controls the contribution of the current input used to update the long-term memory $C^t[\tau]$ (magenta and yellow boxes). The output gate, $O^t[\tau]$, then determines how much this updated long-term memory contributes to the new (short-term memory) hidden state $H^t[\tau]$ (dashed black box).

For a fixed point τ in the sequence, the action of an LSTM cell, i.e., the connection between the input $X^t[\tau]$, the various gates, and the state vectors, is given mathematically as follows. First, the elements of the input sequence together with the hidden state are mapped onto auxiliary gate vectors, collectively denoted by $g^t[\tau] = (g_h^t[\tau]) \in \mathbb{R}^{N_h}$, via

$$g_h^t[\tau] = \mathcal{M}_{hk}^g X_k^t[\tau] + \mathcal{N}_{hh'}^g H_{h'}^t[\tau - 1] + \mathcal{B}_h^g, \quad (2)$$

where $g \in i, f, o, c$ stands for the input, forget, output, and cell state gates, respectively, and Einstein’s summation convention is employed; i.e., indices that appear twice are summed over. The calligraphic symbols $\mathcal{M}_{hk}^g, \mathcal{N}_{hh'}^g$, and \mathcal{B}_h^g

are the free parameters of the network that are optimized for the given problem during the training, which is at the heart of any machine learning approach. The matrix $\mathcal{M}^g = (\mathcal{M}_{hk}^g) \in \mathbb{R}^{N_h \times N_k}$ can be interpreted as a Markovian-like contribution of the current input $X^t[\tau]$ to the gates, whereas $\mathcal{N}^g = (\mathcal{N}_{hh'}^g) \in \mathbb{R}^{N_h \times N_h}$ scales a non-Markovian part determined by the hidden state of the last sequence point $\tau - 1$. The vector $\mathcal{B}^g = (\mathcal{B}_h^g) \in \mathbb{R}^{N_h}$ is a learnable bias. It should be stressed that these parameters do not depend on t or τ and are thus optimized once for the complete dataset $\{X^t\}$.

Note that this mapping is sometimes extended by a contribution to $g_h^t[\tau]$ from the past cell state $C^t[\tau - 1]$. Nevertheless, this “peeping” mechanism is not considered further in this work. For the sake of brevity, we can write the mapping more compactly in matrix vector form as

$$g^t[\tau] = \mathcal{M}^g X^t[\tau] + \mathcal{N}^g H^t[\tau - 1] + \mathcal{B}^g. \quad (3)$$

Second, the actual gate vectors are computed by the core equations of the LSTM as proposed by Hochreiter and Schmidhuber (1997):

$$\begin{aligned} I^t[\tau] &= \sigma(i^t[\tau]), \\ F^t[\tau] &= \sigma(f^t[\tau]), \\ O^t[\tau] &= \sigma(o^t[\tau]), \\ C^t[\tau] &= F^t[\tau] \circ C^t[\tau - 1] + I^t[\tau] \circ \tanh(c^t[\tau]), \\ H^t[\tau] &= O^t[\tau] \circ \tanh(C^t[\tau]), \end{aligned} \quad (4)$$

where σ denotes the logistic sigmoid function, \tanh is the hyperbolic tangent, and \circ stands for the Hadamard product (all applied in an element-wise fashion to the vectors). In the last two equations, the roles of the input, forget, and output gates as described above become apparent.

The third step in a single-layer LSTM (as employed for the work presented here) is to provide the outputs of the current LSTM cell, i.e., $H^t[\tau]$ and $C^t[\tau]$, to the subsequent LSTM cell that processes the next element $X^t[\tau + 1]$ of the input sequence.

The full action of the LSTM network up to the end of the sequence can be written as a nested function call:

$$\begin{aligned} & (H^t[N_\tau], C^t[N_\tau]) \\ &= L(X^t[N_\tau], L(X^t[N_\tau - 1], \dots \\ & L(X^t[1], (H^t[0], C^t[0])) \dots)), \end{aligned} \quad (5)$$

where $L(X^t[\tau], (H^t[\tau - 1], C^t[\tau - 1]))$ represents Eqs. (3) and (4). For the present work, the initial conditions are chosen as $H^t[0] = C^t[0] = 0$, which means that there is no memory longer than N_τ time steps.

The final outputs of the ConvLSTM chain, $H^t[N_\tau]$ and $C^t[N_\tau]$, encode information on the entire input sequence, ending at time t . This information must be decoded via an appropriate subsequent network, as described in Sect. 2.4.

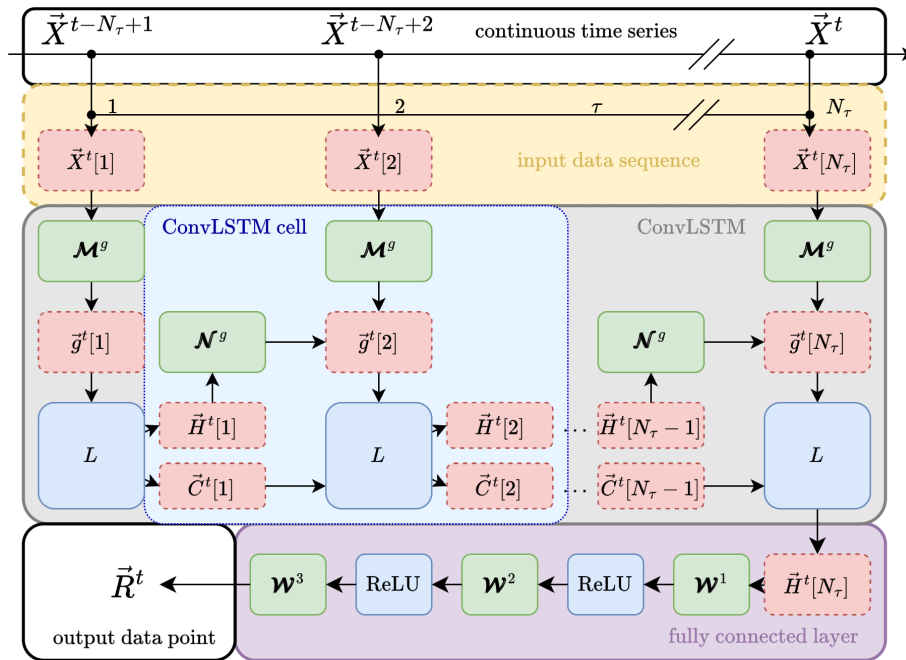


Figure 2. Combined ConvLSTM and FC network architecture. The starting point is the continuous time series of input data $\{\vec{X}^t\}$ (upper white block). From this series, a contiguous sequence of N_τ elements (yellow block) is used to feed a chain of N_τ connected ConvLSTM cells (light-blue block) building the ConvLSTM network (grey block). The input sequence is mapped via weighting matrices \mathcal{M}^g (green blocks) onto gate vectors $\vec{g}^t[\tau]$. The gate vectors are then used to update the cell state $\vec{C}^t[\tau-1]$ and the hidden state $\vec{H}^t[\tau-1]$ of the last ConvLSTM cell to the current values $\vec{C}^t[\tau]$ and $\vec{H}^t[\tau]$, respectively. The update is performed with the LSTM core equations collectively described by the mapping L ; see Eq. (4). The weighting matrices \mathcal{N}^g (green blocks) control how much of the last hidden state enters the updated state. The final output of the ConvLSTM $\vec{H}^t[\tau]$ is then propagated to a FC network, which itself is a chain of three FC layers consisting of weighting matrices \mathcal{W} and connected via ReLU functions; see Sect. 2.4. The final result is the river runoff \vec{R}^t for all rivers considered at the current time instance t (white block in the lower left corner). Note that all the bias vectors are omitted for the sake of clarity. See the text for more information.

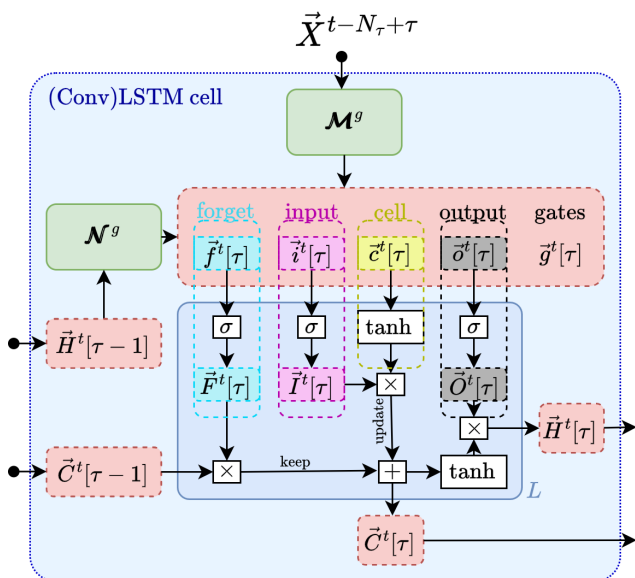


Figure 3. Inner structure of a long short-term memory cell. See Fig. 2 and the text for information.

2.3 Combining the LSTM with spatial convolution

Although the plain LSTM performs well in handling temporal sequences of point-like quantities, it is not designed to recognize spatial features in sequences of two-dimensional maps as atmosphere–ocean interface fields. To address this limitation, we employ a ConvLSTM architecture as described below.

In this type of network, the elements of the input sequence are given as spatially varying fields $\vec{X}^t[\tau] = (X_k^t[x, y, \tau]) \in \mathbb{R}^{N_k \times (N_x \times N_y)}$, where $x \in [1, N_x]$ and $y \in [1, N_y]$ run over the horizontal and vertical dimensions of the map. To enable the “learning” of spatial patterns, the free parameters of the network are replaced with two-dimensional convolution kernels $\mathcal{M}^g = (\mathcal{M}_{hk}^g[\xi, \eta]) \in \mathbb{R}^{(N_h \times N_k) \times (N_\xi \times N_\eta)}$ and $\mathcal{N}^g = (\mathcal{N}_{hh'}^g[\xi, \eta]) \in \mathbb{R}^{(N_h \times N_h) \times (N_\xi \times N_\eta)}$. The width and height of the kernels are given by N_ξ and N_η , respectively, and $\xi \in [-(N_\xi - 1)/2, (N_\xi - 1)/2]$, $\eta \in [-(N_\eta - 1)/2, (N_\eta - 1)/2]$. Without loss of generality, we assume odd numbers for the kernel sizes.

A convolution with these kernels then gives the mapping from the input quantities to the gates:

$$g_h^t[x, y, \tau] = \mathcal{M}_{hh}^g[\xi, \eta] X_k^t[x - \xi, y - \eta, \tau] + \mathcal{N}_{hh'}^g[\xi, \eta] H_{h'}^t[x - \xi, y - \eta, \tau - 1] + \mathcal{B}_h^g, \quad (6)$$

again with Einstein's convention imposed.

It immediately becomes apparent that, in the case of the ConvLSTM, the gate and state vectors must become vector fields ($\in \mathbb{R}^{N_h \times (N_x \times N_y)}$) as well. We can write this mapping in the same way as Eq. (3) but by replacing the standard matrix vector multiplication with a convolution (denoted with $*$), i.e.,

$$g^t[\tau] = \mathcal{M}^g * X^t[\tau] + \mathcal{N}^g * H^t[\tau - 1] + \mathcal{B}^g. \quad (7)$$

The subsequent processing of $g^t[\tau]$ remains symbolically the same as presented in Eq. (4) but with all appearing quantities now meaning vector fields instead of simple vectors.

In summary, the ConvLSTM is designed to process tasks that require a combined understanding of spatial patterns and temporal sequences in data. It merges the image-processing capabilities of CNNs with the time series modeling of LSTM networks.

2.4 Fully connected layer

As stated in Sect. 2.2.1, the final outputs $H^t[N_\tau]$ and $C^t[N_\tau]$ of the ConvLSTM encode information on the full input sequence. To contract this information to obtain the runoff vector R^t representing the N_r rivers, we propose subjecting the final short-term memory (i.e., the hidden state $H^t[N_\tau]$) to an additional FC network.

In particular, the dimensionality of the vector field $H^t[N_\tau]$ is reduced sequentially by three nested FC layers, each connected to the other by the rectified linear unit (ReLU); see Fig. 2. Integrating out artificial degrees of freedom in a step-wise fashion has turned out to be beneficial.

The runoff of the r th river is then obtained via (using Einstein's convention)

$$R_r^t = \mathcal{W}_{rb}^3 \text{ReLU}(\mathcal{W}_{ba}^2 \text{ReLU}(\mathcal{W}_{ah}^1[x, y] H_h^t[x, y, N_\tau] + \mathcal{B}_a^1) + \mathcal{B}_b^2) + \mathcal{B}_r^3, \quad (8)$$

where $a = 1, \dots, N_a$, $b = 1, \dots, N_b$, and the hyperparameters N_a and N_b are chosen such that $N_h \cdot N_x \cdot N_y > N_a > N_b > N_r$ in order to achieve the aforementioned step-by-step reduction in dimensionality. The weights \mathcal{W} and biases \mathcal{B} stand for parameters that are optimized during the training of the network.

In matrix vector notation, this can be compressed to

$$R^t = \mathcal{W}^3 \text{ReLU}(\mathcal{W}^2 \text{ReLU}(\mathcal{W}^1 H^t[N_\tau] + \mathcal{B}^1) + \mathcal{B}^2) + \mathcal{B}^3. \quad (9)$$

Combining Eq. (9) with Eq. (5) finally provides an explicit formula for the initial assumption of modeling the runoff for

time t as a function of a sequence of atmospheric fields; i.e.,

$$\begin{aligned} R^t &= M(\{X_k^t[x, y, \tau]\}) \\ &= \mathcal{W}^3 \text{ReLU}(\mathcal{W}^2 \text{ReLU}(\mathcal{W}^1 L_H(X^t[N_\tau], \\ &\quad L(X^t[N_\tau - 1], \dots, L(X^t[1], (0, 0)) \dots)) + \mathcal{B}^1) \\ &\quad + \mathcal{B}^2) + \mathcal{B}^3, \end{aligned} \quad (10)$$

where L_H means that only the hidden state vector of the final ConvLSTM call is forwarded to the FC layer.

3 Technical details

3.1 Runoff data used for training

The non-stationary daily runoff data covering the period 1979 to 2011 are based on an E-HYPE hindcast simulation that was forced by a regional downscaling of ERA-Interim (Dee et al., 2011) with RCA3 (Samuelsson et al., 2011) and implemented in NEMO-Nordic (Hordoir et al., 2019) as a mass flux. The BMIP project (Gröger et al., 2022) played a crucial role in addressing the lack of consistent river discharge data for the entire study period (1961–2018). On this point, no comparable long-term dataset with daily resolution was available. In other studies multiple datasets have been merged but offer only monthly resolution (see, e.g., Fig. 3, Meier et al., 2019). Hence, a new homogeneous runoff dataset was created. The 1961–1978 runoff data are based on Bergström and Carlsson (1994), with values interpolated from monthly to daily scales. The 2012–2018 data are derived from an E-HYPE forecast product. To ensure consistency for the analysis, the periods before (1961 to 1978) and after (2012 to 2018) have been neglected. Notably, the Neva River is an exception, as its discharge data originate from observational records (1961–2016) provided by the Russian State Hydrological Institute rather than E-HYPE hindcasts.

It should be noted that, for this study, we used an intermediate dataset of river runoff developed during BMIP that was employed to run the ocean model. In this dataset, some rivers had not yet been merged, resulting in discrepancies between the totals of freshwater input locations of 97 in this study and 91 rivers in the final version of Väli et al. (2019). The quality of the runoff was extensively evaluated. The dataset was found to closely align with historical observations for various rivers and with the Bergström and Carlsson (1994) dataset, showing a difference of under 1 % for the total Baltic Sea runoff (Väli et al., 2019). For more information, see Gröger et al. (2022) and Väli et al. (2019).

3.2 Atmospheric forcing

The UERRA-HARMONIE regional reanalysis dataset was developed as part of the FP7 UERRA project (<http://www.uerra.eu/>). The UERRA-HARMONIE system represents a comprehensive, high-resolution reanalysis

covering a wide range of essential climate variables. This dataset includes data on air temperature, pressure, humidity, wind speed and direction, cloud cover, precipitation, albedo, surface heat fluxes, and radiation fluxes from January 1961 to July 2019. With a horizontal resolution of 11 km and analyses conducted at 00:00, 06:00, 12:00, and 18:00 UTC, it also provides hourly resolution forecast model data. UERRA-HARMONIE is accessible through the Copernicus Climate Data Store (CDS, <https://cds.climate.copernicus.eu/#!/home>, last access: 25 March 2025) initially produced during the UERRA project and later changed to the Copernicus Climate Change Service (C3S, <https://climate.copernicus.eu/copernicus-regional-reanalysis-europe>, last access: 25 March 2025). For the training of the neural network, the hourly data were remapped to daily values.

Lastly, it should be noted that UERRA is not the atmospheric dataset that was used to drive the original E-HYPE model.

3.3 Ocean model

We use a coupled three-dimensional ocean model, called the Modular Ocean Model (MOM) (Griffies, 2012), to simulate the Baltic Sea. It has a horizontal resolution of 3 nautical miles, roughly corresponding to 5.556 km, and 152 vertical z^* levels with a first layer thickness of 0.5 m and a total depth of 264 m. This model uses a finite-difference method to solve the full set of primitive equations to calculate the motion of water and the transport of heat and salt. The K-profile parameterization (KPP) was used as a turbulence closure scheme. The model’s western boundary opens into the Skagerrak and connects the Baltic Sea to the North Sea. A more detailed description of the setup can be found in Gröger et al. (2022).

3.4 Neural network hyperparameters

Our architecture is implemented as a sequential model, which allows for testing of multiple ConvLSTM layers – a concatenation of multiple ConvLSTM cells. The following ConvLSTM output is then mapped by three fully connected linear layers, where the final layers map the output to 97 rivers. We used a custom loss function similar to a mean squared error loss that penalizes outliers more severely. For the training, we use the AdamW optimizer, which is an improved version of Adam that decouples weight decay (regularization) from the gradient update, likely leading to better generalization. The optimizer is configured with a learning rate and a weight decay to prevent overfitting. To adapt the learning rate during training, we employ the ReduceLROnPlateau scheduler. This scheduler monitors the validation mean squared error (MSE) and reduces the learning rate by a factor of 10 if no improvement is observed over 10 epochs. This dynamic adjustment helped the model converge more efficiently and avoid getting stuck in local minima.

Table 1. Final set of parameters for the ConvLSTM model.

Parameter name	Parameter size
Channel size	4
No. of hidden layers	9
No. of time steps	30
Convolutional kernel size	(7,7)
No. of ConvLSTM layers	1
Batch size	50

The best set of hyperparameters has been defined by iterating over a pre-defined selection of possible parameters. The set of hyperparameters that has been chosen for the present study is given in Table 1. The model’s performance can be described as relatively robust when changing the set of hyperparameters (see Fig. A3). Interestingly, smaller input sizes of 10 d also perform really well. However, we still decided to use longer timescales, as we assume that larger input sizes increase the stability of the model needed for long-term climate simulations.

River runoff is influenced not only by the current day’s atmospheric conditions, but also by the cumulative and lagged effects of prior days’ weather patterns. The choice of atmospheric fields was based on the assumption that the runoff should be mainly given by the net precipitation (precipitation minus evaporation). The evaporation flux is often calculated as a function of wind speed, the air’s humidity and density, and the involved turbulent exchange coefficients (Karsten et al., 2024), where the air temperature influences the latter two. Hence, as input, the model receives $N_\tau = 30$ d of the atmospheric surface fields temperature, precipitation, specific humidity, and wind speed, with a daily resolution to predict daily river runoff R (see Fig. 4). This window size allows the model to “remember” key atmospheric conditions leading up to a given day, enabling it to accurately predict runoff.

4 Results

4.1 ConvLSTM model evaluation

The model was trained and evaluated with daily data from 1979 to 2011, as this period represents the only period of E-HYPE without further bias correction applied to the runoff to match the observations. The complete dataset was divided into randomly chosen splits of 80 % training data, 10 % validation data to evaluate the model’s performance during training, and 10 % test data which are finally used to assess the model’s performance after training. The model was trained for 400 epochs, and the model weights with the lowest mean squared validation error have been stored. The model’s accuracy for the combined daily predicted runoff from all 97 rivers flowing into the Baltic Sea is displayed in Fig. 5. For

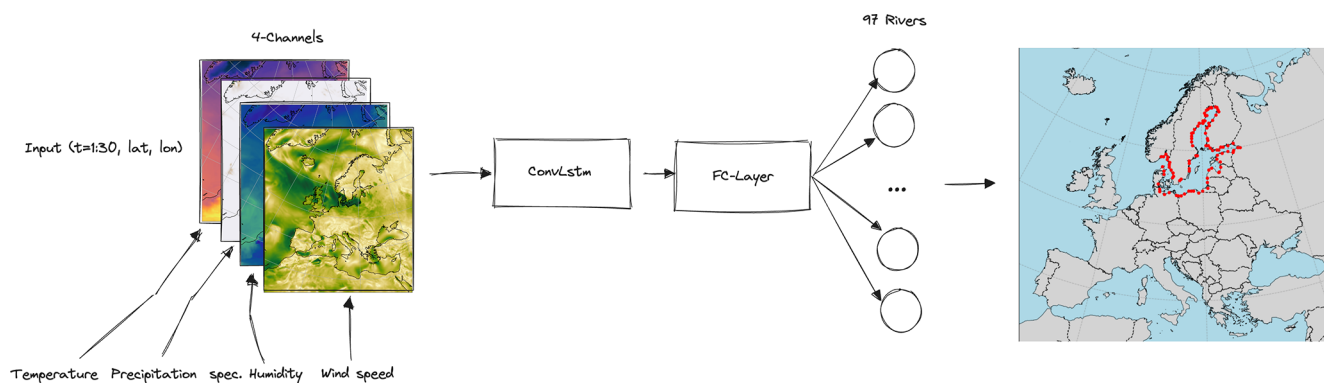


Figure 4. Schematic structure of the ConvLSTM implementation for river runoff forecasting.

this evaluation, the model's output is compared to the test data, which the model has not seen during the training phase.

Panel (a) illustrates the relative prediction error in relation to the E-HYPE data, indicating that, on daily timescales, the model can predict river runoff with an accuracy of $\pm 5\%$. The overall correlation is 0.997, with the resulting error metrics yielding a root mean square error (RMSE) of $323.99 \text{ m}^3 \text{ s}^{-1}$ and a mean absolute error (MAE) of $249.51 \text{ m}^3 \text{ s}^{-1}$. While the model's performance is already satisfactory, the discrepancies between the actual values and the predictions can partly be attributed to the use of a different atmospheric dataset than the one originally used to drive the E-HYPE model. However, by applying a rolling mean with a 5 d window, the prediction error is reduced to less than 1 %, which is acceptable for climate modeling purposes. Panel (b) displays the distribution of residuals as a density plot. The figure shows that the distribution of residuals follows a Gaussian shape. The bell-shaped curve is approximately centered around zero, indicating that the model does not exhibit a systematic bias and meaning that it does not consistently overestimate or underestimate the river runoff values. Most residuals lie within a narrow range around zero, suggesting that large prediction errors are relatively rare.

For individual rivers, the distribution of the residuals still follows a Gaussian shape. However, on daily timescales the errors are larger, reaching $\pm 30\%$ during individual peaks (Fig. A4 in the Appendix).

In the following, the model's performance in reproducing the total river runoff and the discharges of four individual rivers into the Baltic Sea is addressed. Using the test dataset, Fig. 6 shows the predicted river runoff and E-HYPE data.

Panel (a) illustrates the total river runoff into the Baltic Sea, with both the predicted runoff (ConvLSTM) and the river runoff of the hydrological model E-HYPE smoothed using a 5 d rolling mean. The predicted river runoff closely matches the original data, demonstrating the model's accuracy in predicting the overall river runoff into the Baltic Sea. Panel (b) focuses on the Neva River (see Fig. 1), one of the largest rivers flowing into the Baltic Sea. The residual plot

illustrates the prediction errors relative to the E-HYPE river runoff data over time. The ConvLSTM model predicts the Neva River runoff within a $\pm 2.5\%$ range.

Panels (c)–(e) show the prediction residuals for the Oder River, the Ume River, and the Neman River (see Fig. 1). Compared to the Neva River, the prediction errors are larger, which may be attributed to the training dataset, as the runoff for the Neva River is based on measurements, whereas that of the other rivers is based solely on E-HYPE. Still, all of the results lie within the error margin of E-HYPE itself compared to the observations (Fig. A1), with the average error on daily timescales for individual rivers mostly under 10 %, showcasing the model's ability to forecast runoff for this river accurately.

The residuals were calculated as the relative difference between the predicted runoff and runoff data used for training and were finally normalized by the runoff data used for training.

An overview of the individual error metrics is given in Table A1.

4.2 Application of the ConvLSTM in combination with an ocean model

Lastly, we evaluate the performance of the ConvLSTM by incorporating the predicted river runoff as hydrological forcing into MOM5. This provides a robust validation of the runoff model against more complex real-world conditions and ensures that the predictions accurately reflect the impact of the river discharge on the ocean dynamics. This, in turn, confirms that the ConvLSTM captures the temporal and spatial variability of river runoff and that the residuals shown in Fig. 5 are indeed insignificant when it comes to realistic applications.

Figure 7 shows the salinity comparison between E-HYPE and the predicted river runoff at BY15 – a central station in the Baltic Sea east of Gotland. The ocean model simulation using the river runoff predicted by the ConvLSTM closely mirrors the control simulation that is forced with the E-HYPE runoff. Panel (a) shows the surface salinity, repre-

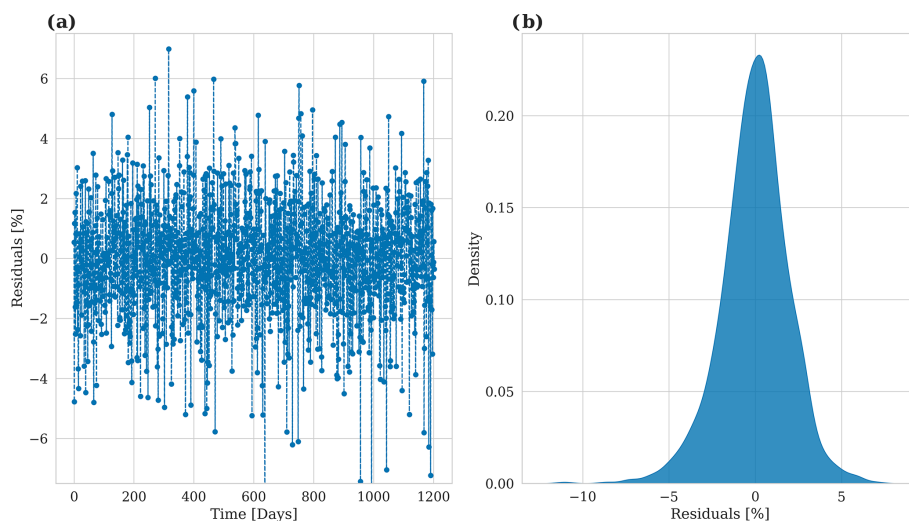


Figure 5. Model accuracy for predicted river runoff. **(a)** Relative prediction error over time in relation to the E-HYPE data. **(b)** Density plot of the residuals.

sending the high-frequency variations in salinity heavily affected by river runoff. The predicted salinity using ConvLSTM river runoff matches the control simulation well, capturing the short-term fluctuations effectively. Panel (b) shows the bottom salinity, representing low-frequency variations in the Baltic Sea, which is also reproduced well with the ConvLSTM predictions. It should be noted that the discrepancies between the simulated salinity and the observed values at BY15 are not directly linked to the performance of the ConvLSTM river runoff model. Instead, they are attributed to MOM5's representation of physical processes, particularly the treatment of mixing, advection, and stratification in the Baltic Sea. Several factors may contribute to these discrepancies. The Baltic Sea is known for its strong vertical stratification due to the input of freshwater from rivers. MOM5 uses the KPP scheme for turbulence, which may not perfectly resolve small-scale mixing processes and vertical salinity gradients. This can result in an overestimation of salinity variability at the surface. Moreover, while MOM5 captures the large-scale dynamics of the Baltic Sea, the lateral transport of saltwater from the Skagerrak into the central Baltic Sea may not be represented perfectly. This can introduce variability in surface and bottom salinity that is not observed in reality. However, all in all, the long-term trends and larger salinity changes are captured accurately, indicating the model's robustness in predicting high-frequency and low-frequency variations.

The final evaluation of the ConvLSTM model focuses on the spatial accuracy of river runoff predictions as visualized in Fig. 8. Panel (a) exhibits the vertically averaged salinity from 1981 to 2011 in the reference simulation. It highlights the Baltic Sea's strong horizontal gradients and complex topographic features, as evidenced by salinity variations in deeper waters captured by the vertical integration.

In panel (b), these reference results are compared to the ConvLSTM simulation by showing the percentage difference in vertically averaged salinity. Overall, the differences remain below 1 %, except in the Gulf of Riga (22–24° E, 56.5–58.5° N for orientation), where the Daugava River dominates the runoff. The difference is approximately 1 %.

5 Conclusions

With the increasing demand from decision-makers for regional climate projections to quantify regional climate change impacts, the availability of precise hydrological forecasting becomes invaluable. In this work, we describe the implementation of a ConvLSTM network for predicting river runoff in a regional climate model, highlighting its potential to enhance river runoff forecasting across different coastal seas. Our model not only reproduces the total river runoff entering the Baltic Sea but also performs exceptionally well for individual rivers.

All of the results lie within the error margin of the hydrological model itself when compared to the observations, with the average error on daily timescales for individual rivers mostly under 10 %. Hence, our results confirm the excellent performance of LSTM networks in predicting river runoff (Humphrey et al., 2016; Huang et al., 2014; Ashrafi et al., 2017; Liu et al., 2020; Fang and Shao, 2022; Kratzert et al., 2018). In addition, our results align well with the observed performance of ConvLSTM networks in similar applications for predicting individual rivers (Ha et al., 2021), basin-wide runoff (Zhu et al., 2023), and precipitation nowcasting (Shi et al., 2015). In our study, we further extend the use of ConvLSTM networks by predicting multiple ($n = 97$) rivers at once while maintaining high accuracy for the entire Baltic Sea as well as individual rivers. Moreover, the predicted river

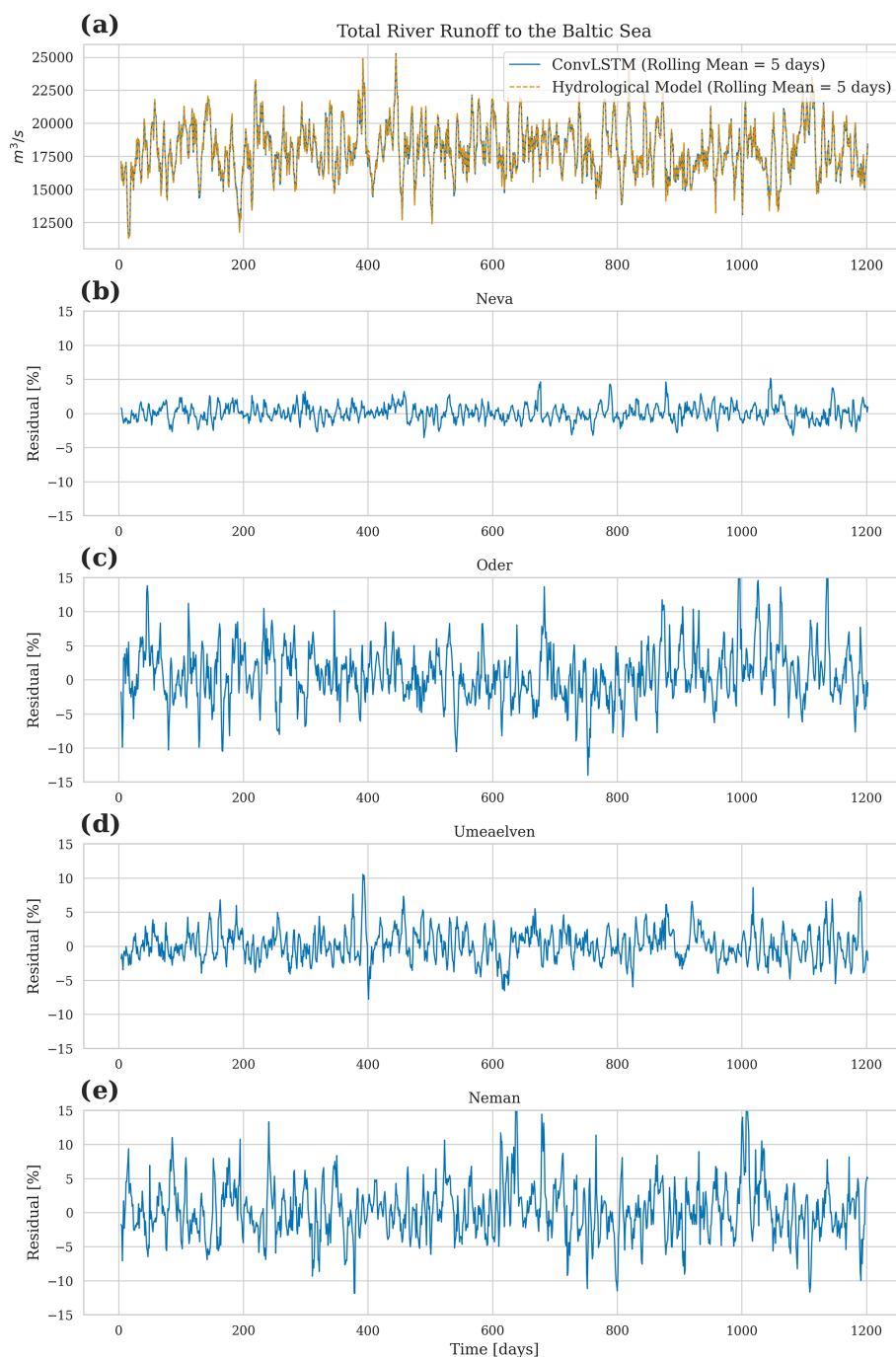


Figure 6. Model performance in predicting river runoff: (a) total river runoff to the Baltic Sea with a 5 d rolling mean for both the predicted and original data of the E-HYPE hydrological model. (b–e) Residuals of runoff prediction for individual rivers showing the prediction error over time. The residuals were calculated as the relative difference between the predicted and observed values, normalized by the observed values.

runoff proved robust when using the river runoff in a comprehensive ocean model setup of the Baltic Sea. Extending the simulation beyond the known period also provided robust results (Fig. A2).

The transition from traditional hydrological models to machine learning approaches, such as the implementation of the ConvLSTM model, offers significant advantages as the model can be integrated seamlessly into regional climate models, allowing for real-time computation of river runoff

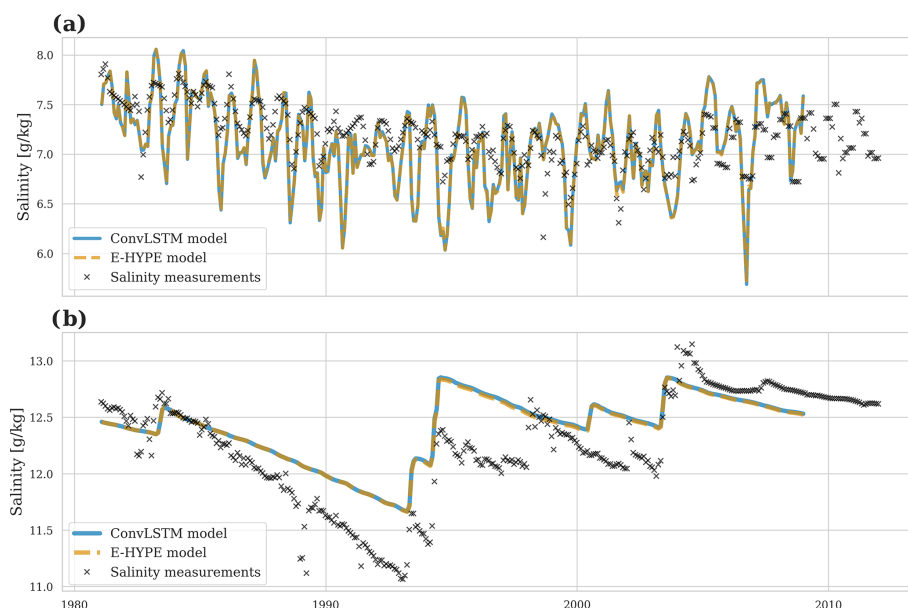


Figure 7. Salinity comparison at BY15 in the Baltic Sea: (a) surface salinity showing high-frequency variations and (b) bottom salinity showing low-frequency trends. The ConvLSTM predictions follow the E-HYPE data closely, demonstrating the model's accuracy in reproducing salinity levels affected by river runoff.

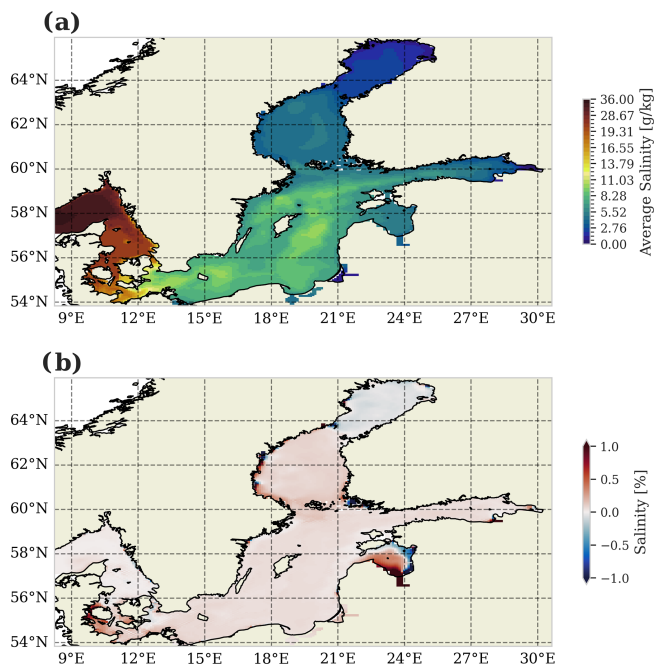


Figure 8. Spatial accuracy of river runoff predictions using the ConvLSTM model. (a) Vertically averaged salinity for the period 1981 to 2011 in the reference simulation, highlighting the sharp horizontal gradients and complex topographic features in the Baltic Sea. (b) Percentage difference in vertically averaged salinity between the reference simulation and the ConvLSTM simulation.

while making climate projections. While the initial training of the model requires substantial computational resources, this remains significantly less intensive than running comprehensive hydrological models. Furthermore, once trained, the ConvLSTM model is computationally efficient during inference, ensuring enhanced forecasting capabilities without significantly increasing computational demands. The achieved speedup (depending on the complexity of the hydrological model) is within the range of 30 to 90 times faster.

Nevertheless, the quality of the ConvLSTM model still depends on the performance of the hydrological model, which provides a comprehensive, homogeneous dataset that is essential for effective training. While different in their architectures, both the hydrological model and the ConvLSTM model provide a mapping from atmospheric variables to river runoff. This training approach contrasts with using measurement data for training, which is significantly more challenging due to the data sparsity over larger regions and potentially varying measurement techniques. Thus, rather than rendering traditional methods obsolete, the integration of machine learning models builds upon and enhances the foundational data provided by hydrological model methods.

The robust performance of the ConvLSTM model in simulating river runoff and its possible effective integration into coupled regional climate models, as in the IOW ESM (Karsten et al., 2024), pave the way for a multitude of new storyline simulations. Importantly, this can now be achieved without any expert domain knowledge on hydrological modeling. Hence, we can now explore various “what if” scenarios more reliably, under the assumption that the model weights

attained during the training are robust. Such scenario testing is crucial for crafting effective water resource management strategies and adapting to a changing climate, and it therefore represents a significant step forward in our ability to understand and predict the complex dynamics of river systems and their impact on regional climate systems.

Appendix A

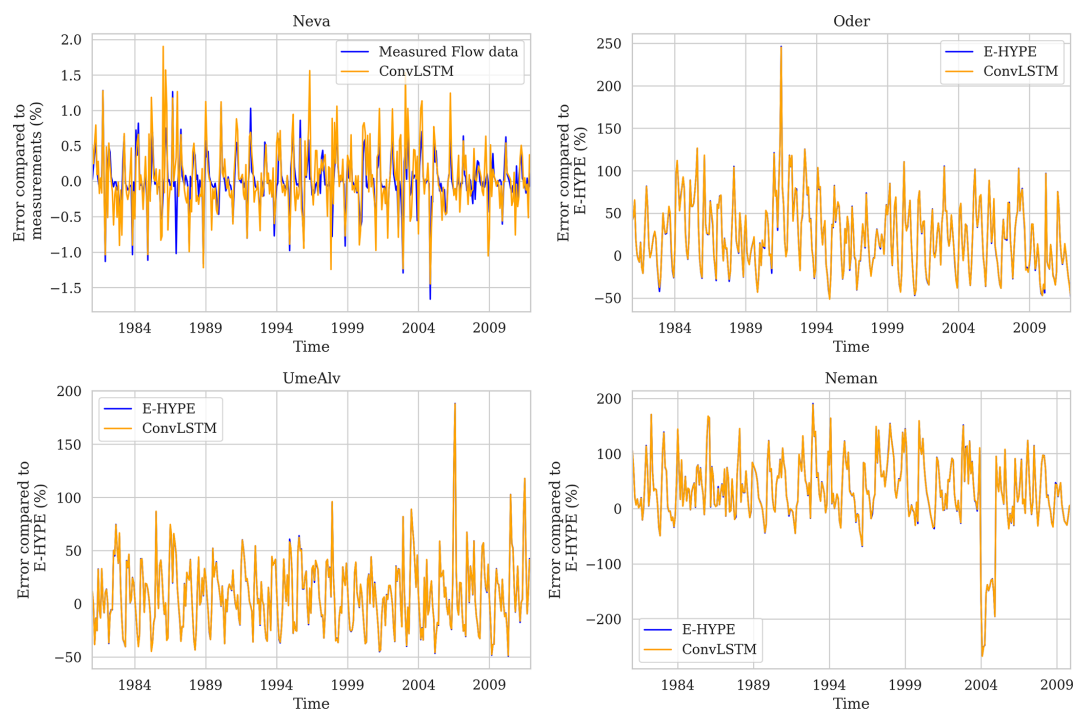


Figure A1. Residual for the hydrological model E-HYPE as well as for the prediction by the ConvLSTM model. The residuals were calculated as the relative differences between the predicted and observed values, normalized by the observed values.

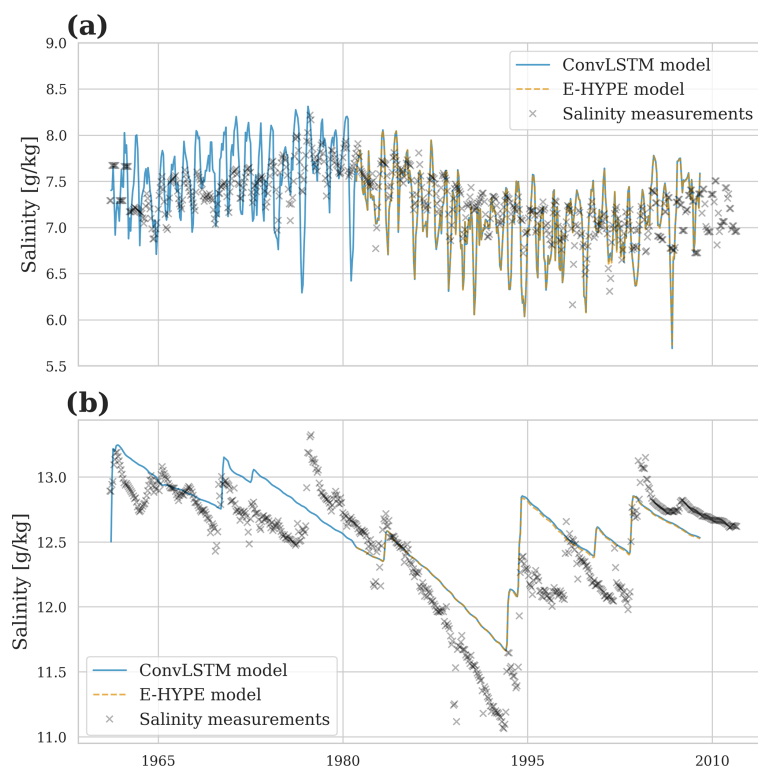


Figure A2. Extended regional ocean model simulation for the period 1961–2009. Station BY15 is validated.

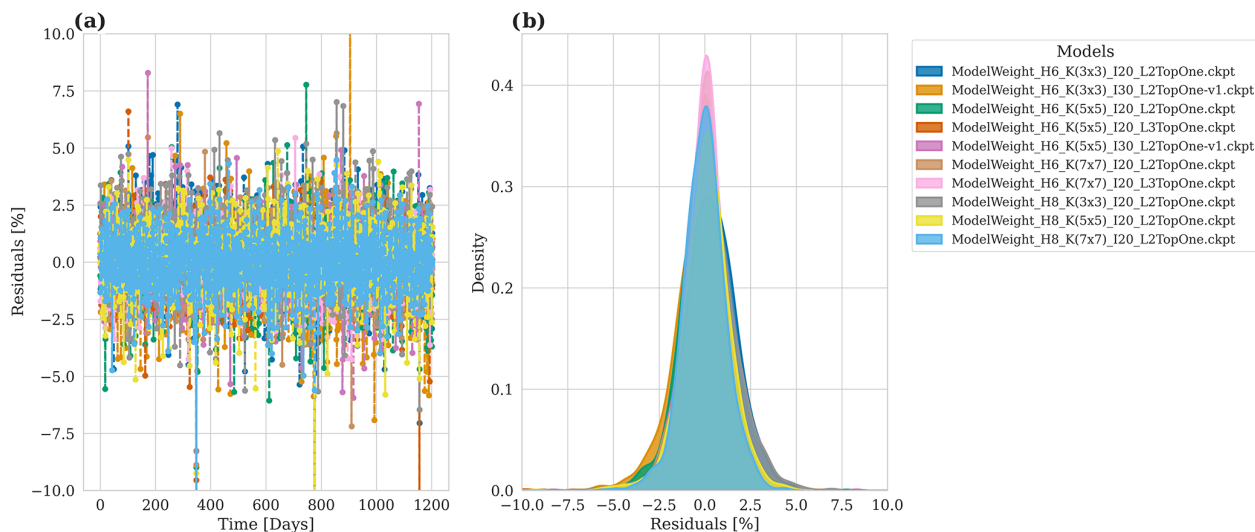


Figure A3. Model accuracy for predicted river runoff for 10 different hyperparameters. (a) Relative prediction error over time in relation to the E-HYPE data. (b) Density plot of the residuals.

Table A1. Model performance in predicting river runoff, together with the error metrics.

River	RMSE ($\text{m}^3 \text{s}^{-1}$)	MAE ($\text{m}^3 \text{s}^{-1}$)	Correlation
All rivers	323.99	249.51	0.997
Neva	65.41	48.36	0.995
Oder	49.93	38.79	0.994
Ume	23.73	17.92	0.994
Neman	55.65	41.63	0.996

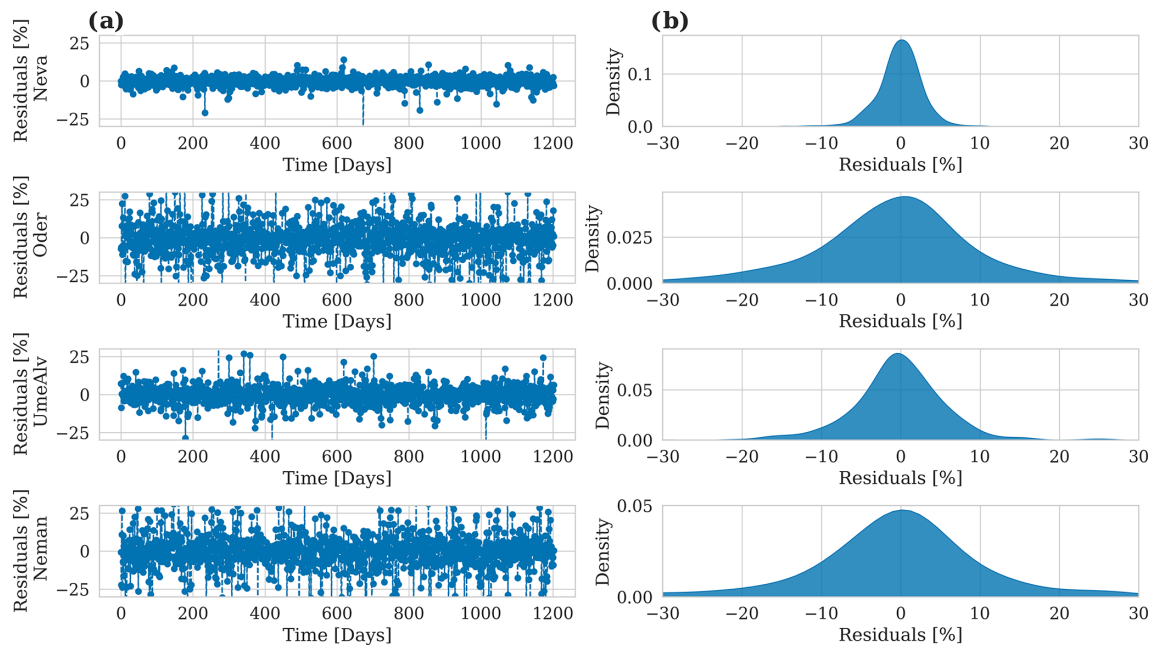


Figure A4. Model accuracy for predicted river runoff for individual rivers. (a) Relative prediction error over time in relation to the E-HYPE data. (b) Density plots of the residuals.

Code and data availability. The ocean model data and atmospheric data used in this study are described and/or published in Gröger et al. (2022) (<https://doi.org/10.5194/gmd-15-8613-2022>). All additional data are properly cited and accessible. The ocean model data needed to reproduce the results of this study are accessible at <https://doi.org/10.5281/zenodo.13365099> (Börgel, 2024a). The code of the ConvLSTM can be accessed at <https://doi.org/10.5281/zenodo.13910136> (Börgel, 2024b). All scripts to reproduce the figures can be found at <https://doi.org/10.5281/zenodo.13910136> (Börgel, 2024b). The source code of the ocean model is available at <https://github.com/mom-ocean/MOM5> (Griffies, 2012).

Author contributions. FB: writing – review and editing, writing – original draft, visualization, resources, methodology, investigation, formal analysis, conceptualization. SK: writing – review and editing, methodology. KR: visualization, writing – review and editing, methodology. UG: writing – review and editing, conceptualization.

Competing interests. The contact author has declared that none of the authors has any competing interests.

Disclaimer. Publisher’s note: Copernicus Publications remains neutral with regard to jurisdictional claims made in the text, published maps, institutional affiliations, or any other geographical representation in this paper. While Copernicus Publications makes every effort to include appropriate place names, the final responsibility lies with the authors.

Acknowledgements. The research presented in this study is part of the Baltic Earth program (Earth System Science for the Baltic Sea region; see <https://www.baltic.earth>, last access: 25 March 2025). The authors gratefully acknowledge the computing time granted by the Resource Allocation Board and provided on the supercomputers at NHR@ZIB as part of the NHR infrastructure.

Review statement. This paper was edited by David Topping and reviewed by two anonymous referees.

References

- Ashrafi, M., Chua, L. H. C., Quek, C., and Qin, X.: A fully-online Neuro-Fuzzy model for flow forecasting in basins with limited data, *J. Hydrol.*, 545, 424–435, 2017.
- Bergström, S. and Carlsson, B.: RIVER RUNOFF TO THE BALTIC SEA – 1950–1990, *Ambio*, 23, 280–287, 1994.
- Börgel, F.: Supporting data ocean model GMD submission: From Weather Data to River Runoff: Leveraging Spatiotemporal Convolutional Networks for Comprehensive Discharge Forecasting, Zenodo [data set], <https://doi.org/10.5281/zenodo.13365099>, 2024a.
- Börgel, F.: florianboergel/runoff_prediction: gmd (Version v1), Zenodo [code], <https://doi.org/10.5281/zenodo.13910136>, 2024b.
- Cook, B. I., Mankin, J. S., Marvel, K., Williams, A. P., Smerdon, J. E., and Anchukaitis, K. J.: Twenty-First Century Drought Projections in the CMIP6 Forcing Scenarios, *Earth's Future*, 8, e2019EF001461, <https://doi.org/10.1029/2019EF001461>, 2020.
- Dee, D. P., Uppala, S. M., Simmons, A. J., Berrisford, P., Poli, P., Kobayashi, S., Andrae, U., Balmaseda, M. A., Balsamo, G., Bauer, P., Bechtold, P., Beljaars, A. C. M., van de Berg, L., Bidlot, J., Bormann, N., Delsol, C., Dragani, R., Fuentes, M., Geer, A. J., Haimberger, L., Healy, S. B., Hersbach, H., Hólm, E. V., Isaksen, L., Kållberg, P., Köhler, M., Matricardi, M., McNally, A. P., Monge-Sanz, B. M., Morcrette, J.-J., Park, B.-K., Peubey, C., de Rosnay, P., Tavolato, C., Thépaut, J.-N., and Vitart, F.: The ERA-Interim reanalysis: Configuration and performance of the data assimilation system, *Q. J. Roy. Meteor. Soc.*, 137, 553–597, 2011.
- Fang, L. and Shao, D.: Application of long short-term memory (LSTM) on the prediction of rainfall-runoff in karst area, *Front. Phys.*, 9, 790687, <https://doi.org/10.3389/fphy.2021.790687>, 2022.
- Fang, W., Huang, S., Ren, K., Huang, Q., Huang, G., Cheng, G., and Li, K.: Examining the Applicability of Different Sampling Techniques in the Development of Decomposition-Based Streamflow Forecasting Models, *J. Hydrol.*, 568, 534–550, <https://doi.org/10.1016/j.jhydrol.2018.11.020>, 2019.
- Griffies, S.: Elements of the Modular Ocean Model (MOM) (2012 release with updates) GFDL Ocean Group Technical Report No. 7 Stephen M. Griffies NOAA/Geophysical Fluid Dynamics Laboratory 632 + xiii pages with this https://mom-ocean.github.io/assets/pdfs/MOM5_manual.pdf (last access: 25 March 2025), 2012 (code available at: <https://github.com/mom-ocean/MOM5>, last access: 25 March 2025).
- Gröger, M., Placke, M., Meier, H. E. M., Börgel, F., Brunnabend, S.-E., Dutheil, C., Gräwe, U., Hieronymus, M., Neumann, T., Radtke, H., Schimanke, S., Su, J., and Väli, G.: The Baltic Sea Model Intercomparison Project (BMIP) – a platform for model development, evaluation, and uncertainty assessment, *Geosci. Model Dev.*, 15, 8613–8638, <https://doi.org/10.5194/gmd-15-8613-2022>, 2022.
- Ha, S., Liu, D., and Mu, L.: Prediction of Yangtze River Streamflow Based on Deep Learning Neural Network with El Niño–Southern Oscillation, *Sci. Rep.*, 11, 11738, <https://doi.org/10.1038/s41598-021-90964-3>, 2021.
- Hagemann, S., Stacke, T., and Ho-Hagemann, H. T. M.: High Resolution Discharge Simulations Over Europe and the Baltic Sea Catchment, *Front. Earth Sci.*, 8, 12, <https://doi.org/10.3389/feart.2020.00012>, 2020.
- Hochreiter, S. and Schmidhuber, J.: Long short-term memory, *Neural Comput.*, 9, 1735–1780, 1997.
- Hordoir, R., Axell, L., Höglund, A., Dieterich, C., Fransner, F., Gröger, M., Liu, Y., Pemberton, P., Schimanke, S., Andersson, H., Ljungemyr, P., Nygren, P., Falahat, S., Nord, A., Jönsson, A., Lake, I., Döös, K., Hieronymus, M., Dietze, H., Löptien, U., Kuznetsov, I., Westerlund, A., Tuomi, L., and Haapala, J.: Nemo-Nordic 1.0: a NEMO-based ocean model for the Baltic and North seas – research and operational applications, *Geosci. Model Dev.*, 12, 363–386, <https://doi.org/10.5194/gmd-12-363-2019>, 2019.
- Huang, S., Chang, J., Huang, Q., and Chen, Y.: Monthly streamflow prediction using modified EMD-based support vector machine, *J. Hydrol.*, 511, 764–775, 2014.
- Humphrey, G. B., Gibbs, M. S., Dandy, G. C., and Maier, H. R.: A hybrid approach to monthly streamflow forecasting: integrating hydrological model outputs into a Bayesian artificial neural network, *J. Hydrol.*, 540, 623–640, 2016.
- Hundecha, Y., Arheimer, B., Donnelly, C., and Pechlivanidis, I.: A regional parameter estimation scheme for a pan-European multi-basin model, *J. Hydrol.*, 6, 90–111, <https://doi.org/10.1016/j.ejrh.2016.04.002>, 2016.
- Höhlein, K., Kern, M., Hewson, T., and Westermann, R.: A Comparative Study of Convolutional Neural Network Models for Wind Field Downscaling, *Meteorol. Appl.*, 27, e1961, <https://doi.org/10.1002/met.1961>, 2020.
- Karsten, S., Radtke, H., Gröger, M., Ho-Hagemann, H. T. M., Mashayekh, H., Neumann, T., and Meier, H. E. M.: Flux coupling approach on an exchange grid for the IOW Earth System Model (version 1.04.00) of the Baltic Sea region, *Geosci. Model Dev.*, 17, 1689–1708, <https://doi.org/10.5194/gmd-17-1689-2024>, 2024.
- Kniebusch, M., Meier, H. M., and Radtke, H.: Changing Salinity Gradients in the Baltic Sea As a Consequence of Altered Freshwater Budgets, *Geophys. Res. Lett.*, 46, 9739–9747, <https://doi.org/10.1029/2019GL083902>, 2019.
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, *Hydrol. Earth Syst. Sci.*, 22, 6005–6022, <https://doi.org/10.5194/hess-22-6005-2018>, 2018.
- Lindström, G., Pers, C., Rosberg, J., Strömqvist, J., and Arheimer, B.: Development and testing of the HYPE (Hydrological Predictions for the Environment) water quality model for different spatial scales, *Hydrol. Res.*, 41, 295–319, 2010.
- Liu, D., Jiang, W., Mu, L., and Wang, S.: Streamflow prediction using deep learning neural network: case study of Yangtze River, *IEEE Access*, 8, 90069–90086, 2020.
- Meier, H., Eilola, K., Almroth-Rosell, E., Schimanke, S., Kniebusch, M., Höglund, A., Pemberton, P., Liu, Y., Väli, G., and Saraiva, S.: Disentangling the impact of nutrient load and climate changes on Baltic Sea hypoxia and eutrophication since 1850, *Clim. Dynam.*, 53, 1145–1166, <https://doi.org/10.1007/s00382-018-4296-y>, 2019.

- Meier, H. E. M. and Kauker, F.: Modeling decadal variability of the Baltic Sea: 2. Role of freshwater inflow and large-scale atmospheric circulation for salinity, *J. Geophys. Res.-Oceans*, 108, 3368, <https://doi.org/10.1029/2003JC001799>, 2003.
- Meier, H. E. M., Hordoir, R., Andersson, H. C., Dieterich, C., Eilola, K., Gustafsson, B. G., Höglund, A., and Schimanke, S.: Modeling the Combined Impact of Changing Climate and Changing Nutrient Loads on the Baltic Sea Environment in an Ensemble of Transient Simulations for 1961–2099, *Clim. Dynam.*, 39, 2421–2441, <https://doi.org/10.1007/s00382-012-1339-7>, 2012.
- Meier, H. M. and Döscher, R.: Simulated water and heat cycles of the Baltic Sea using a 3D coupled atmosphere–ice–ocean model, *Boreal Environ. Res.*, 7, 327–334, 2002.
- Moishin, M., Deo, R. C., Prasad, R., Raj, N., and Abdulla, S.: Designing deep-based learning flood forecast model with ConvLSTM hybrid algorithm, *IEEE Access*, 9, 50982–50993, 2021.
- Omstedt, A. and Hansson, D.: The Baltic Sea ocean climate system memory and response to changes in the water and heat balance components, *Cont. Shelf Res.*, 26, 236–251, <https://doi.org/10.1016/j.csr.2005.11.003>, 2006.
- Samuelsson, P., Jones, C. G., Willén, U., Ullerstig, A., Golivik, S., Hansson, U., Jansson, C., Kjellström, E., Nikulin, G., and Wyser, K.: The Rossby Centre Regional Climate Model RCA3: Model Description and Performance, *Tellus A*, 63, 4–23, <https://doi.org/10.1111/j.1600-0870.2010.00478.x>, 2011.
- Sherstinsky, A.: Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network, *Physica D*, 404, 132306, <https://doi.org/10.1016/j.physd.2019.132306>, 2020.
- Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-C.: Convolutional LSTM network: A machine learning approach for precipitation nowcasting, in: *Advances in Neural Information Processing Systems*, edited by: Cortes, C., Lawrence, N., Lee, D., Sugiyama, M., and Garnett, R., Vol. 28, Curran Associates, Inc., 2015.
- Tan, Q.-F., Lei, X.-H., Wang, X., Wang, H., Wen, X., Ji, Y., and Kang, A.-Q.: An Adaptive Middle and Long-Term Runoff Forecast Model Using EEMD-ANN Hybrid Approach, *J. Hydrol.*, 567, 767–780, <https://doi.org/10.1016/j.jhydrol.2018.01.015>, 2018.
- Väli, G., Meier, H. M., Placke, M., and Dieterich, C.: River runoff forcing for ocean modeling within the Baltic Sea Model Intercomparison Project, *Meereswiss. Ber.*, 113, <https://doi.org/10.12754/msr-2019-0113>, 2019.
- Wang, A., Miao, Y., Kong, X., and Wu, H.: Future Changes in Global Runoff and Runoff Coefficient From CMIP6 Multi-Model Simulation Under SSP1-2.6 and SSP5-8.5 Scenarios, *Earth's Future*, 10, e2022EF002910, <https://doi.org/10.1029/2022EF002910>, 2022.
- Winsor, P., Rodhe, J., and Omstedt, A.: Baltic Sea Ocean Climate: An Analysis of 100 Yr of Hydrographic Data with Focus on the Freshwater Budget, *Clim. Res.*, 18, 5–15, <https://doi.org/10.3354/cr018005>, 2001.
- Zhu, S., Wei, J., Zhang, H., Xu, Y., and Qin, H.: Spatiotemporal deep learning rainfall-runoff forecasting combined with remote sensing precipitation products in large scale basins, *J. Hydrol.*, 616, 128727, <https://doi.org/10.1016/j.jhydrol.2022.128727>, 2023.