Hydrol. Earth Syst. Sci., 22, 6005–6022, 2018
https://doi.org/10.5194/hess-22-6005-2018

Research article | 22 Nov 2018

Rainfall–runoff modelling using Long Short-Term Memory (LSTM

Frederik Kratzert[1,*], Daniel Klotz[1], Claire Brenner[1], Karsten Schulz[1], and Mathew Herrnegger[1]
[1]Institute of Water Management, Hydrology and Hydraulic Engineering, University of Natural Resources and Life Sciences, Vienna, 1190, Aus
[*] Invited contribution by Frederik Kratzert, recipient of the EGU Hydrological Sciences Outstanding Student Poster and PICO Award 2016.

Correspondence: Frederik Kratzert (f.kratzert@gmail.com)

Received: 04 May 2018 – Discussion started: 14 May 2018 – Revised: 25 Sep 2018 – Accepted: 13 Nov 2018 – Published: 22 N

Abstract

Rainfall–runoff modelling is one of the key challenges in the field of hydrology. Various approaches exist, ranging from physically conceptual to fully data-driven models. In this paper, we propose a novel data-driven approach, using the Long Short-Term Mem network, a special type of recurrent neural network. The advantage of the LSTM is its ability to learn long-term dependencies be provided input and output of the network, which are essential for modelling storage effects in e.g. catchments with snow influen 241 catchments of the freely available CAMELS data set to test our approach and also compare the results to the well-known Sa Moisture Accounting Model (SAC-SMA) coupled with the Snow-17 snow routine. We also show the potential of the LSTM as a reg hydrological model in which one model predicts the discharge for a variety of catchments. In our last experiment, we show the p transfer process understanding, learned at regional scale, to individual catchments and thereby increasing model performance w compared to a LSTM trained only on the data of single catchments. Using this approach, we were able to achieve better model p as the SAC-SMA + Snow-17, which underlines the potential of the LSTM for hydrological modelling applications.

Download & links

Article (PDF, 10256 KB)

How to cite. Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term (LSTM) networks, Hydrol. Earth Syst. Sci., 22, 6005–6022, https://doi.org/10.5194/hess-22-6005-2018, 2018.

1 Introduction

Rainfall–runoff modelling has a long history in hydrological sciences and the first attempts to predict the discharge as a function precipitation events using regression-type approaches date back 170 years (Beven, 2001; Mulvaney, 1850). Since then, modellin have been further developed by progressively incorporating physically based process understanding and concepts into the (math model formulations. These include explicitly addressing the spatial variability of processes, boundary conditions and physical pro catchments (Freeze and Harlan, 1969; Kirchner, 2006; Schulla, 2007). These developments are largely driven by the advanceme computer technology and the availability of (remote sensing) data at high spatial and temporal resolution (Hengl et al., 2017; Ko et al., 2010; Mu et al., 2011; Myneni et al., 2002; Rennó et al., 2008).

However, the development towards coupled, physically based and spatially explicit representations of hydrological processes at t scale has come at the price of high computational costs and a high demand for necessary (meteorological) input data (Wood et a Therefore, physically based models are still rarely used in operational rainfall–runoff forecasting. In addition, the current data se parameterization of these kind of models, e.g. the 3-D information on the physical characteristics of the sub-surface, are mostly for small, experimental watersheds, limiting the model's applicability for larger river basins in an operational context. The high c costs further limit their application, especially if uncertainty estimations and multiple model runs within an ensemble forecasting hydrological problems. Shi et al. (2015) investigated a deep learning approach for precipitation nowcasting. Tao et al. (2016) use neural network for bias correction of satellite precipitation products. Fang et al. (2017) investigated the use of deep learning mo soil moisture in the context of NASA's Soil Moisture Active Passive (SMAP) satellite mission. Assem et al. (2017) compared the p a deep learning approach for water flow level and flow predictions for the Shannon River in Ireland with multiple baseline models reported that the deep learning approach outperforms all baseline models consistently. More recently, D. Zhang et al. (2018) co performance of different neural network architectures for simulating and predicting the water levels of a combined sewer structu Drammen (Norway), based on online data from rain gauges and water-level sensors. They confirmed that LSTM (as well as anot neural network architecture with cell memory) are better suited for for multi-step-ahead predictions than traditional architecture explicit cell memory. J. Zhang et al. (2018) used an LSTM for predicting water tables in agricultural areas. Among other things, t compared the resulting simulation from the LSTM-based approach with that of a traditional neural network and found that the fo outperforms the latter. In general, the potential use and benefits of DL approaches in the field of hydrology and water sciences h recently come into the focus of discussion (Marçais and de Dreuzy, 2017; Shen, 2018; Shen et al., 2018). In this context we wo mention Shen (2018) more explicitly, since he provides an ambitious argument for the potential of DL in earth sciences/hydrolog so he also provides an overview of various applications of DL in earth sciences. Of special interest for the present case is his poi might also provide an avenue for discovering emergent behaviours of hydrological phenomena.

Regardless of the hydrological modelling approach applied, any model will be typically calibrated for specific catchments for whic time series of meteorological and hydrological data are available. The calibration procedure is required because models are only simplifications of real catchment hydrology and model parameters have to effectively represent non-resolved processes and any subgrid-scale heterogeneity in catchment characteristics (e.g. soil hydraulic properties) (Beven, 1995; Merz et al., 2006). The tra of model parameters (regionalization) from catchments where meteorological and runoff data are available to ungauged or data- is one of the ongoing challenges in hydrology (Buytaert and Beven, 2009; He et al., 2011; Samaniego et al., 2010).

The aim of this study is to explore the potential of the LSTM architecture (in the adapted version proposed by Gers et al., 2000) the rainfall–runoff behaviour of a large number of differently complex catchments at the daily timescale. Additionally, we want to potential of LSTMs for regionalizing the rainfall–runoff response by training a single model for a multitude of catchments. In orde a more general conclusion about the suitability of our modelling approach, we test this approach on a large number of catchmen CAMELS data set (Addor et al., 2017b; Newman et al., 2014). This data set is freely available and includes meteorological forcin observed discharge for 671 catchments across the contiguous United States. For each basin, the CAMELS data set also includes t simulated discharge from the Sacramento Soil Moisture Accounting Model (Burnash et al., 1973) coupled with the Snow-17 snow (Anderson, 1973). In our study, we use these simulations as a benchmark, to compare our model results with an established mo approach.

The paper is structured in the following way: in Sect. 2, we will briefly describe the LSTM network architecture and the data set u followed by an introduction into three different experiments: in the first experiment, we test the general ability of the LSTM to m runoff processes for a large number of individual catchments. The second experiment investigates the capability of LSTMs for reg modelling, and the last tests whether the regional models can help to enhance the simulation performance for individual catchme Section 3 presents and discusses the results of our experiments, before we end our paper with a conclusion and outlook for futu

2 Methods and database

2.1 Long Short-Term Memory network

In this section, we introduce the LSTM architecture in more detail, using the notation of Graves et al. (2013). Beside a technical the network internals, we added a "hydrological interpretation of the LSTM" in Sect. 3.5 in order to bridge differences between th hydrological and deep learning research communities.

The LSTM architecture is a special kind of recurrent neural network (RNN), designed to overcome the weakness of the traditional long-term dependencies. Bengio et al. (1994) have shown that the traditional RNN can hardly remember sequences with a length For daily streamflow modelling, this would imply that we could only use the last 10 days of meteorological data as input to predi streamflow of the next day. This period is too short considering the memory of catchments including groundwater, snow or even storages, with lag times between precipitation and discharge up to several years.

To explain how the RNN and the LSTM work, we unfold the recurrence of the network into a directed acyclic graph (see Fig. 1). T our case discharge) for a specific time step is predicted from the input $x=[\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n]$ consisting of the last n consecutive time st independent variables (in our case daily precipitation, min/max temperature, solar radiation and vapour pressure) and is process sequentially. In each time step t $(1 \leq t \leq n)$, the current input $\boldsymbol{x}_t$ is processed in the recurrent cells of each layer in the networ

Figure 2 (a) The internal operation of a traditional RNN cell: $\boldsymbol{h}_t$ stands for hidden state and $\boldsymbol{x}_t$ for the input at time step t. **(b)** internals of a LSTM cell, where $\boldsymbol{f}$ stands for the forget gate (Eq. 2), $\boldsymbol{i}$ for the input gate (Eqs. 3–4), and $\boldsymbol{o}$ for the output gate (E $\boldsymbol{c}_t$ denotes the cell state at time step t and $\boldsymbol{h}_t$ the hidden state.

Download

In a traditional RNN cell, only one internal state $\boldsymbol{h}_t$ exists (see Fig. 2a), which is recomputed in every time step by the following e

$$\boldsymbol{h}_t = g(\mathbf{W}\boldsymbol{x}_t + \mathbf{U}\boldsymbol{h}_{t-1} + \boldsymbol{b}),$$

where $g(\cdot)$ is the activation function (typically the hyperbolic tangent), $\mathbf{W}$ and $\mathbf{U}$ are the adjustable weight matrices of the hidden the input $\boldsymbol{x}$, and $\boldsymbol{b}$ is an adjustable bias vector. In the first time step, the hidden state is initialized as a vector of zeros and its le user-defined hyperparameter of the network.

In comparison, the LSTM has (i) an additional cell state or cell memory $\boldsymbol{c}_t$ in which information can be stored, and (ii) gates (thre letters in Fig. 2b) that control the information flow within the LSTM cell (Hochreiter and Schmidhuber, 1997). The first gate is the introduced by Gers et al. (2000). It controls which elements of the cell state vector $\boldsymbol{c}_{t-1}$ will be forgotten (to which degree):

$$\boldsymbol{f}_t = \sigma(\mathbf{W}_\mathrm{f}\boldsymbol{x}_t + \mathbf{U}_\mathrm{f}\boldsymbol{h}_{t-1} + \boldsymbol{b}_\mathrm{f}),$$

where $\boldsymbol{f}_t$ is a resulting vector with values in the range (0, 1), $\sigma(\cdot)$ represents the logistic sigmoid function and $\mathbf{W}_\mathrm{f}$, $\mathbf{U}_\mathrm{f}$ and $\boldsymbol{b}_\mathrm{f}$ defin learnable parameters for the forget gate, i.e. two adjustable weight matrices and a bias vector. As for the traditional RNN, the hi is initialized in the first time step by a vector of zeros with a user-defined length.

In the next step, a potential update vector for the cell state is computed from the current input ($\boldsymbol{x}_t$) and the last hidden state ($\boldsymbol{h}_t$ the following equation:

$$\tilde{\boldsymbol{c}}_t = \tanh(\mathbf{W}_{\tilde{c}}\boldsymbol{x}_t + \mathbf{U}_{\tilde{c}}\boldsymbol{h}_{t-1} + \boldsymbol{b}_{\tilde{c}}),$$

where $\tilde{\boldsymbol{c}}_t$ is a vector with values in the range (−1, 1), $\tanh(\cdot)$ is the hyperbolic tangent and $\mathbf{W}_{\tilde{c}}$, $\mathbf{U}_{\tilde{c}}$ and $\boldsymbol{b}_{\tilde{c}}$ are another set of parameters.

Additionally, the second gate is compute, the input gate, defining which (and to what degree) information of $\tilde{\boldsymbol{c}}_t$ is used to update state in the current time step:

$$\boldsymbol{i}_t = \sigma(\mathbf{W}_\mathrm{i}\boldsymbol{x}_t + \mathbf{U}_\mathrm{i}\boldsymbol{h}_{t-1} + \boldsymbol{b}_\mathrm{i}),$$

where $\boldsymbol{i}_t$ is a vector with values in the range (0, 1), and $\mathbf{W}_\mathrm{i}$, $\mathbf{U}_\mathrm{i}$ and $\boldsymbol{b}_\mathrm{i}$ are a set of learnable parameters, defined for the input gat

With the results of Eqs. (2)–(4) the cell state $\boldsymbol{c}_t$ is updated by the following equation:

$$\boldsymbol{c}_t = \boldsymbol{f}_t \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t,$$

where $\odot$ denotes element-wise multiplication. Because the vectors $\boldsymbol{f}_t$ and $\boldsymbol{i}_t$ have both entries in the range (0, 1), Eq. (5) can be the way that it defines, which information stored in $\boldsymbol{c}_{t-1}$ will be forgotten (values of $\boldsymbol{f}_t$ of approx. 0) and which will be kept (value approx. 1). Similarly, $\boldsymbol{i}_t$ decides which new information stored in $\tilde{\boldsymbol{c}}_t$ will be added to the cell state (values of $\boldsymbol{i}_t$ of approx. 1) and ignored (values of $\boldsymbol{i}_t$ of approx. 0). Like the hidden state vector, the cell state is initialized by a vector of zeros in the first time st corresponds to the length of the hidden state vector.

The third and last gate is the output gate, which controls the information of the cell state $\boldsymbol{c}_t$ that flows into the new hidden state output gate is calculated by the following equation:

$$\boldsymbol{o}_t = \sigma(\mathbf{W}_\mathrm{o}\boldsymbol{x}_t + \mathbf{U}_\mathrm{o}\boldsymbol{h}_{t-1} + \boldsymbol{b}_\mathrm{o}),$$

where $\boldsymbol{o}_t$ is a vector with values in the range (0, 1), and $\mathbf{W}_\mathrm{o}$, $\mathbf{U}_\mathrm{o}$ and $\boldsymbol{b}_\mathrm{o}$ are a set of learnable parameters, defined for the output this vector, the new hidden state $\boldsymbol{h}_t$ is calculated by combining the results of Eqs. (5) and (6):

$$\boldsymbol{h}_t = \tanh(\boldsymbol{c}_t) \odot \boldsymbol{o}_t.$$

It is in particular the cell state ($\boldsymbol{c}_t$) that allows for an effective learning of long-term dependencies. Due to its very simple linear i with the remaining LSTM cell, it can store information unchanged over a long period of time steps. During training, this characte

Input: $x = [\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n], \boldsymbol{x}_t \in \mathbb{R}^m$

Given parameters: $\mathbf{W}_\mathrm{f}, \mathbf{U}_\mathrm{f}, \boldsymbol{b}_\mathrm{f}, \mathbf{W}_{\tilde{c}}, \mathbf{U}_{\tilde{c}}, \boldsymbol{b}_{\tilde{c}}, \mathbf{W}_\mathrm{i}, \mathbf{U}_\mathrm{i}, \boldsymbol{b}_\mathrm{i}, \mathbf{W}_\mathrm{o}, \mathbf{U}_\mathrm{o}, \boldsymbol{b}_\mathrm{o}$

Initialize $\boldsymbol{h}_0, \boldsymbol{c}_0 = \vec{0}$ of length *p*

for t=1, …, n **do**

  Calculate $\boldsymbol{f}_t$ (Eq. 2), $\tilde{\boldsymbol{c}}_t$ (Eq. 3), $\boldsymbol{i}_t$ (Eq. 4)

  Update cell state $\boldsymbol{c}_t$ (Eq. 5)

  Calculate $\boldsymbol{o}_t$ (Eq. 6), $\boldsymbol{h}_t$ (Eq. 7)

end for

Output: $h = [\boldsymbol{h}_1, \ldots, \boldsymbol{h}_n], \boldsymbol{h}_t \in \mathbb{R}^p$

2.2 The calibration procedure

In traditional hydrological models, the calibration involves a defined number of iteration steps of simulating the entire calibration given set of model parameters and evaluating the model performance with some objective criteria. The model parameters are, r the applied optimization technique (global and/or local), perturbed in such a way that the maximum (or minimum) of an objectiv found. Regarding the training of a LSTM, the adaptable (or *learnable*) parameters of the network, the weights and biases, are als depending on a given loss function of an iteration step. In this study we used the mean-squared error (MSE) as an objective crit

In contrast to most hydrological models, the neural network exhibits the property of differentiability of the network equations. Th gradient of the loss function with respect to any network parameter can always be calculated explicitly. This property is used in t back-propagation step in which the network parameters are adapted to minimize the overall loss. For a detailed description see Goodfellow et al. (2016).

A schematic illustration of one iteration step in the LSTM training/calibration is is provided in Fig. 3. One iteration step during the LSTMs usually works with a subset (called *batch* or *mini-batch*) of the available training data. The number of samples per batch i hyperparameter, which in our case was defined to be 512. Each of these samples consists of one discharge value of a given day meteorological input of the *n* preceding days. In every iteration step, the loss function is calculated as the average of the MSE of and observed runoff of these 512 samples. Since the discharge of a specific time step is only a function of the meteorological inp last *n* days, the samples within a batch can consist of random time steps (depicted in Fig. 3 by the different colours), which mus necessarily be ordered chronologically. For faster convergence, it is even advantageous to have random samples in one batch (L et al., 2012). This procedure is different from traditional hydrological model calibration, where usually all the information of the data is processed in each iteration step, since all simulated and observed runoff pairs are used in the model evaluation.

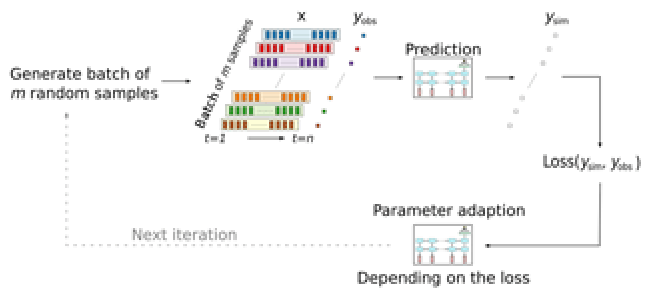


Figure 3 Illustration of one iteration step in the training process of the LSTM. A random batch of input data *x* consisting of *m* independent training samples (depicted by the colours) is used in each step. Each training sample consists of *n* days of look-ba and one target value (y_{obs}) to predict. The loss is computed from the observed discharge and the network's predictions y_{sim} an update the network parameters.

Download

For efficient learning, all input features (the meteorological variables) as well as the output (the discharge) data are normalized l the mean and dividing by the standard deviation (LeCun et al., 2012; Minns and Hall, 1996). The mean and standard deviation u normalization are calculated from the calibration period only. To receive the final discharge prediction, the output of the network retransformed using the normalization parameters from the calibration period (Fig. 4 shows the retransformed model outputs).

2.3 Open-source software

Our research heavily relies on open source software. The programming language of choice is Python 3.6 (van Rossum, 1995). Th use for preprocessing our data and for data management in general are Numpy (Van Der Walt et al., 2011), Pandas (McKinney, Scikit-Learn (Pedregosa et al., 2011). The Deep-Learning frameworks we use are TensorFlow (Abadi et al., 2016) and Keras (Ch All figures are made using Matplotlib (Hunter, 2007).

Table 1 Overview of the HUCs considered in this study and some region statistics averaged over all basins in that region. For variable mean and standard deviation is reported.

HUC	Region name	No. of basins	Mean precipitation (mm day^{-1})	Mean aridity[1] (–)	Mean altitude (m)	Mean snow frac.[2] (–)	Mean seasonality[3] (–)
01	New England	27	3.61 ± 0.26	0.60 ± 0.03	316 ± 182	0.24 ± 0.06	0.10 ± 0.08
03	South Atlantic-Gulf	92	3.79 ± 0.49	0.87 ± 0.14	189 ± 179	0.02 ± 0.02	0.12 ± 0.26
11	Arkansas-White-Red	31	2.86 ± 0.89	1.18 ± 0.50	613 ± 713	0.08 ± 0.13	0.25 ± 0.29
17	Pacific Northwest	91	5.22 ± 2.03	0.59 ± 0.40	1077 ± 589	0.33 ± 0.23	-0.72 ± 0.17

[1] PET/P; see Addor et al. (2017a). [2] Fraction of precipitation falling on days with temperatures below 0 °C. [3] Positive values ind precipitation peaks in summer, negative values that precipitation peaks in the winter month, and values close to 0 that the pre is uniform throughout the year (see Addor et al., 2017a).

Download Print Version
|
Download XLSX

2.4 The CAMELS data set

The underlying data for our study is the CAMELS data set (Addor et al., 2017b; Newman et al., 2014). The acronym stands for "C Attributes for Large-Sample Studies" and it is a freely available data set of 671 catchments with minimal human disturbances ac contiguous United States (CONUS). The data set contains catchment aggregated (lumped) meteorological forcing data and obser discharge at the daily timescale starting (for most catchments) from 1980. The meteorological data are calculated from three dif gridded data sources (Daymet, Thornton et al., 2012; Maurer, Maurer et al., 2002; and NLDAS, Xia et al., 2012) and consists of precipitation, shortwave downward radiation, maximum and minimum temperature, snow-water equivalent and humidity. We us Daymet data, since it has the highest spatial resolution (1 km grid compared to 12 km grid for Maurer and NLDAS) as a basis for the catchment averages and all available meteorological input variables with exception of the snow-water equivalent and the day

The 671 catchments in the data set are grouped into 18 hydrological units (HUCs) following the U.S. Geological Survey's HUC ma et al., 1987). These groups correspond to geographic areas that represent the drainage area of either a major river or the combi area of a series of rivers.

In our study, we used 4 out of the 18 hydrological units with their 241 catchments (see Fig. 5 and Table 1) in order to cover a w different hydrological conditions on one hand and to limit the computational costs on the other hand. The New England region in east contains 27 more or less homogeneous basins (e.g. in terms of snow influence or aridity). The Arkansas-White-Red region i of CONUS has a comparable number of basins, namely 32, but is completely different otherwise. Within this region, attributes e. mean annual precipitation have a high variance and strong gradient from east to west (see Fig. 5). Also comparable in size but w hydro-climatic conditions are the South Atlantic-Gulf region (92 basins) and the Pacific Northwest region (91 basins). The latter s the Pacific coast till the Rocky Mountains and also exhibits a high variance of attributes across the basins, comparable to the Ark Red region. For example, there are very humid catchments with more than 3000 mm yr^{-1} precipitation close to the Pacific coast (aridity index 2.17, mean annual precipitation 500 mm yr^{-1}) basins in the south-east of this region. The relatively flat South Atla region contains more homogeneous basins, but in contrast to the New England region is not influenced by snow.

2.5 Experimental design

Throughout all of our experiments, we used a two-layer LSTM network, with each layer having a cell/hidden state length of 20. T
the resulting shapes of all model parameters from Eqs. (2) to (8). Between the layers, we added dropout, a technique to prevent
from overfitting (Srivastava et al., 2014). Dropout sets a certain percentage (10 % in our case) of random neurons to zero durin
order to force the network into a more robust feature learning. Another hyperparameter is the length of the input sequence, whi
corresponds to the number of days of meteorological input data provided to the network for the prediction of the next discharge
decided to keep this value constant at 365 days for this study in order to capture at least the dynamics of a full annual cycle.

The specific design of the network architecture, i.e. the number of layers, cell/hidden state length, dropout rate and input seque
were found through a number of experiments in several seasonal-influenced catchments in Austria. In these experiments, differe
architectures (e.g. one or two LSTM layers or 5, 10, 15, or 20 cell/hidden units) were varied manually. The architecture used in t
proved to work well for these catchments (in comparison to a calibrated hydrological model we had available from previous studi
Herrnegger et al., 2018) and was therefore chosen to be applied here without further tuning. A systematic sensitivity analysis of
different hyper-parameters was however not done and is something to do in the future.

Table 2 Shapes of learnable parameters of all layers.

Layer	Parameter	Shape
1st LSTM layer	$\mathbf{W}_\mathrm{f}, \mathbf{W}_{\tilde{c}}, \mathbf{W}_\mathrm{i}, \mathbf{W}_\mathrm{o}$	[20, 5]
	$\mathbf{U}_\mathrm{f}, \mathbf{U}_{\tilde{c}}, \mathbf{U}_\mathrm{i}, \mathbf{U}_\mathrm{o}$	[20, 20]
	$\boldsymbol{b}_\mathrm{f}, \boldsymbol{b}_{\tilde{c}}, \boldsymbol{b}_\mathrm{i}, \boldsymbol{b}_\mathrm{o}$	[20]
2nd LSTM layer	$\mathbf{W}_\mathrm{f}, \mathbf{W}_{\tilde{c}}, \mathbf{W}_\mathrm{i}, \mathbf{W}_\mathrm{o}$	[20, 20]
	$\mathbf{U}_\mathrm{f}, \mathbf{U}_{\tilde{c}}, \mathbf{U}_\mathrm{i}, \mathbf{U}_\mathrm{o}$	[20, 20]
	$\boldsymbol{b}_\mathrm{f}, \boldsymbol{b}_{\tilde{c}}, \boldsymbol{b}_\mathrm{i}, \boldsymbol{b}_\mathrm{o}$	[20]
Dense layer	$\mathbf{W}_\mathrm{d}$	[20, 1]
	$\boldsymbol{b}_\mathrm{d}$	[1]

Download Print Version
|
Download XLSX

We want to mention here that our calibration scheme (see description in the three experiments below) is not the standard way f
and selecting data-driven models, especially neural networks. As of today, a widespread calibration strategy for DL models is to
data into three parts, referred to as training, validation and test data (see Goodfellow et al., 2016). The first two splits are used
parametrization of the networks and the remainder of the data to diagnose the actual performance. We decided to not implemen
splitting strategy, because we are limited to the periods Newman et al. (2015) used so that our models are comparable with thei
Theoretically, it would be possible to split the 15-year calibration period of Newman et al. (2015) further into a training and valic
However, this would lead to (a) a much shorter period of data that is used for the actual weight updates or (b) a high risk of ove
short validation period, depending on how this 15-year period is divided. In addition to that, LSTMs with a low number of hidden
quite sensitive to the initialization of their weights. It is thus common practice to repeat the calibration task several times with d
random seeds to select the best performing realization of the model (Bengio, 2012). For the present purpose we decided not to i
these strategies, since it would make it more difficult or even impossible to compare the LSTM approach to the SAC-SMA + Snow
reference model. The goal of this study is therefore not to find the best per-catchment model, but rather to investigate the gene
of LSTMs for the task of rainfall–runoff modelling. However, we think that the sample size of 241 catchment is large enough to in
the (average) properties of the LSTM-based approach.

2.5.1 Experiment 1: one model for each catchment

With the first experiment, we test the general ability of our LSTM network to model rainfall–runoff processes. Here, we train one
separately for each of the 241 catchments. To avoid the effect of overfitting of the network on the training data, we identified th
epochs (for a definition of an epoch, see Sect. 2.2) in a preliminary step, which yielded, on average, the highest Nash–Sutcliffe e
(NSE) across all basins for an independent validation period. For this preliminary experiment, we used the first 14 years of the 1
calibration period as training data and the last, fifteenth, year as the independent validation period. With the 14 years of data, w
model for in total 200 epochs for each catchment and evaluated each model after each epoch with the validation data. Across all
the highest mean NSE was achieved after 50 epochs in this preliminary experiment. Thus, for the final training of the LSTM with

same batch size as in Experiment 1 is used (see Sect. 2.2 for an explanation of the connection of number of iterations, number o samples and number of epochs). Thus, for the final training, we train one LSTM for each of the four used HUCs for 20 epochs wit 15-year long calibration period.

2.5.3 Experiment 3: fine-tuning the regional model for each catchment

In the third experiment, we want to test whether the more general knowledge of the regional model (Experiment 2) can help to performance of the LSTM in a single catchment. In the field of DL this is a common approach called fine-tuning (Razavian et al., Yosinski et al., 2014), where a model is first trained on a huge data set to learn general patterns and relationships between (met input data and (streamflow) output data (this is referred to as *pre-training*). Then, the pre-trained network is further trained for number of epochs with the data of a specific catchment alone to adapt the more generally learned processes to a specific catchm speaking, the LSTM first learns the general behaviour of the runoff generating processes from a large data set, and is in a secon adapted in order to account for the specific behaviour of a given catchment (e.g. the scaling of the runoff response in a specific c

In this study, the regional models of Experiment 2 serve as pre-trained models. Therefore, depending on the affiliation of a catch certain HUC, the specific regional model for this HUC is taken as a starting point for the fine-tuning. With the initial LSTM weights regional model, the training is continued only with the training data of a specific catchment for a few epochs (ranging from 0 to 2 median 10). Thus, similar to Experiment 1, we finally have 241 different models, one for each of the 241 catchments. Different f previous experiments, we do not use a global number of epochs for fine-tuning. Instead, we used the 14-year/1-year split to det optimal number of epochs for each catchment individually. The reason is that the regional model fits individual catchments withi differently well. Therefore, the number of epochs the LSTM needs to adapt to a certain catchment before it starts to overfit is dif each catchment.

2.6 Evaluation metrics

The metrics for model evaluation are the Nash–Sutcliffe efficiency (Nash and Sutcliffe, 1970) and the three decompositions follo et al. (2009). These are the correlation coefficient of the observed and simulated discharge (r), the variance bias (α) and the tot bias (β). While all of these measures evaluate the performance over the entire time series, we also use three different signatures duration curve (FDC) that evaluate the performance of specific ranges of discharge. Following Yilmaz et al. (2008), we calculate the 2 % flows, the peak flows (FHV), the bias of the slope of the middle section of the FDC (FMS) and the bias of the bottom 30 (FLV).

Because our modelling approach needs 365 days of meteorological data as input for predicting one time step of discharge, we ca the first year of the calibration period. To be able to compare our models to the SAC-SMA + Snow-17 benchmark model, we reco metrics for the benchmark model for the same simulation periods.

3 Results and discussion

We start presenting our results by showing an illustrative comparison of the modelling capabilities of traditional RNNs and the LS highlight the problems of RNNs to learn long-term dependencies and its deficits for the task of rainfall–runoff modelling. This is f the analysis of the results of Experiment 1, for which we trained one network separately for each basin and compare the results SMA + Snow-17 benchmark model. Then we investigate the potential of LSTMs to learn hydrological behaviour at the regional sca context, we compare the performance of the regional models from Experiment 2 against the models of Experiment 1 and discuss strengths and weaknesses. Lastly, we examine whether our fine-tuning approach enhances the predictive power of our models i individual catchments. In all cases, the analysis is based on the data of the 241 catchments of the calibration (the first 15 years) validation (all remaining years available) periods.

3.1 The effect of (not) learning long-term dependencies

As stated in Sect. 2.1, the traditional RNN can only learn dependencies of 10 or less time steps. The reason for this is the so-cal or exploding gradients" phenomenon (see Bengio et al., 1994, and Hochreiter and Schmidhuber, 1997), which manifests itself in signal during the backward pass of the network training that either diminishes towards zero or grows against infinity, preventing learning of long-term dependencies. However, from the perspective of hydrological modelling, a catchment contains various pro dependencies well above 10 days (which corresponds to 10 time steps in the case of daily streamflow modelling), e.g. snow accu during winter and snowmelt during spring and summer. Traditional hydrological models need to reproduce these processes corre to be able to make accurate streamflow predictions. This is in principle not the case for data-driven approaches.

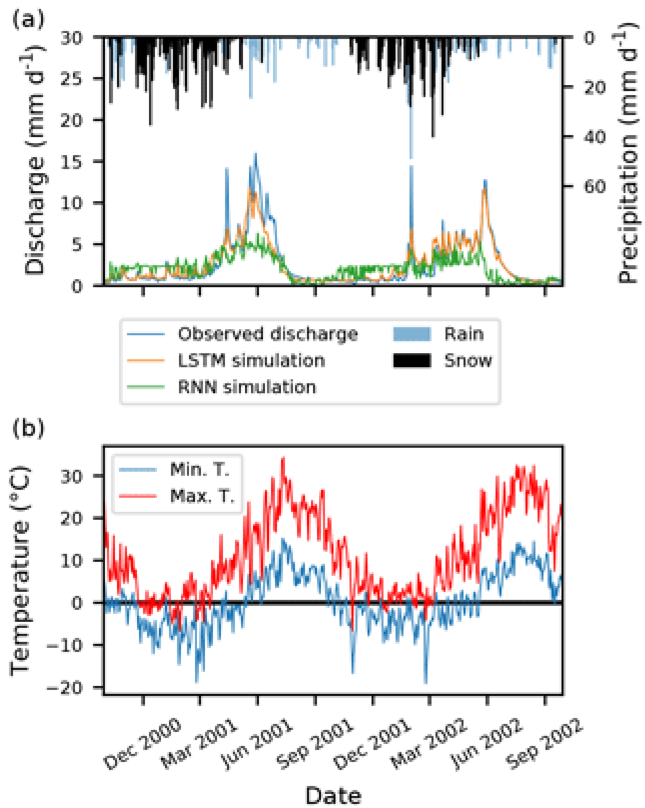


Figure 6 **(a)** Two years of observed as well as the simulated discharge of the LSTM and RNN from the validation period of basin 13340600. The precipitation is plotted from top to bottom and days with minimum temperature below zero are marked a (black bars). **(b)** The corresponding daily maximum and minimum temperature.

Download

In contrast, the LSTM seems to have (i) no or fewer problems with predicting the correct amount of discharge during the snowm and (ii) the predicted hydrograph is much smoother and fits the general trends of the hydrograph much better. Note that both n trained with the exact same data and have the same data available for predicting a single day of discharge.

Here we have only shown a single example for a snow-influenced basin. We also compared the modelling behaviour in one of the catchments of the Arkansas-White-Red region, and found that the trends and conclusion were similar. Although only based on a illustrative example that shows the problems of RNNs with long-term dependencies, we can conclude that traditional RNNs shoul if (e.g. daily) discharge is predicted only from meteorological observations.

3.2 Using LSTMs as hydrological models

Figure 7a shows the spatial distribution of the LSTM performances for Experiment 1 in the validation period. In over 50 % of the an NSE of 0.65 or above is found, with a mean NSE of 0.63 over all catchments. We can see that the LSTM performs better in ca with snow influence (New England and Pacific Northwest regions) and catchments with higher mean annual precipitation (also th England and Pacific Northwest regions, but also basins in the western part of the Arkansas-White-Red region; see Fig. 5a for pre distribution). The performance deteriorates in the more arid catchments, which are located in the western part of the Arkansas-V region, where no discharge is observed for longer periods of the year (see Fig. 5b). Having a constant value of discharge (zero ir for a high percentage of the training samples seems to be difficult information for the LSTM to learn and to reproduce this hydrol behaviour. However, if we compare the results for these basins to the benchmark model (Fig. 7b), we see that for most of these catchments the LSTM outperforms the latter, meaning that the benchmark model did not yield satisfactory results for these catch either. In general, the visualization of the differences in the NSE shows that the LSTM performs slightly better in the northern, m influenced catchments, while the SAC-SMA + Snow-17 performs better in the catchments in the south-east. This clearly shows th using LSTMs, since the snow accumulation and snowmelt processes are correctly reproduced, despite their inherent complexity. suggest that the model learns these long-term dependencies, i.e. the time lag between precipitation falling as snow during the w and runoff generation in spring with warmer temperatures. The median value of the NSE differences is −0.03, which means that benchmark model slightly outperforms the LSTM. Based on the mean NSE value (0.58 for the benchmark model, compared to 0. LSTM of this Experiment), the LSTM outperforms the benchmark results.

Figure 7 Panel **(a)** shows the NSE of the validation period of the models from Experiment 1 and panel **(b)** the difference of th
between the LSTM and the benchmark model (blue colours (>0) indicate that the LSTM performs better than the benchmark m
(<0) the other way around). The colour maps are limited to [0, 1] for the NSE and [−0.4, 0.4] for the NSE differences for bett
visualization.

Download

In Fig. 8, we present the cumulative density functions (CDF) for various metrics for the calibration and validation period. We see
LSTM and the benchmark model work comparably well for all but the FLV (bias of the bottom 30 % low flows) metric. The undere
the peak flow in both models could be expected when using the MSE as the objective function for calibration (Gupta et al., 2009)
the LSTM underestimates the peaks more strongly compared to the benchmark model (Fig. 8d). In contrast, the middle section o
better represented in the LSTM (Fig. 8e). Regarding the performance in terms of the NSE, the LSTM shows fewer negative outlie
seems to be more robust. The poorest model performance in the validation period is an NSE of −0.42 compared to −20.68 of the
Snow-17. Figure 8f shows large differences between the LSTM and the SAC-SMA + Snow-17 model regarding the FLV metric. The
sensitive to the one single minimum flow in the time series, since it compares the area between the FDC and this minimum valu
space of the observed and simulated discharge. The discharge from the LSTM model, which has no exponential outflow function
traditional hydrological models, can easily drop to diminutive numbers or even zero, to which we limited our model output. A rat
solution for this issue is to introduce just one additional parameter and to limit the simulated discharge not to zero, but to the m
observed flow from the calibration period. Figure 9 shows the effect of this approach on the CDF of the FLV. We can see that this
solution leads to better FLV values compared to the benchmark model. Other metrics, such as the NSE, are almost unaffected by
since these low-flow values only marginally influence the resulting NSE values (not shown here).

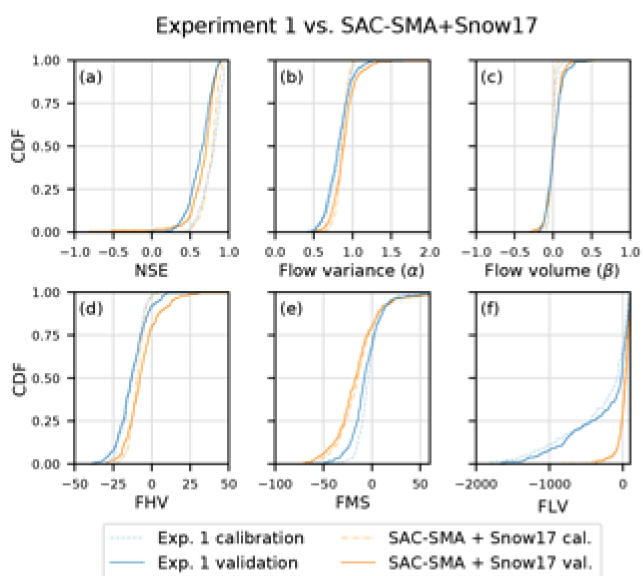
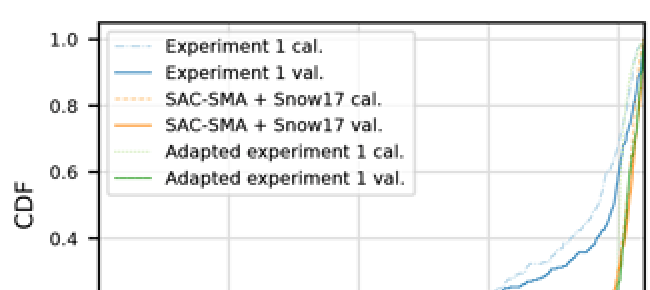


Figure 8 Cumulative density functions for various metrics of the calibration and validation period of Experiment 1 compared to
benchmark model. FHV is the bias of the top 2 % flows, the peak flows, FMS is the slope of the middle section of the flow dura
and FLV is the bias of the bottom 30 % low flows.

Download

From the CDF of the NSE in Fig 8a, we can also observe a trend towards higher values in the calibration compared to the validat
both modelling approaches. This is a sign of overfitting, and in the case of the LSTM, could be tackled by a smaller network size,
regularization or more data. However, we want to highlight again that achieving the best model performance possible was not th
study, but rather testing the general ability of the LSTM to reproduce runoff processes.



3.3 LSTMs as regional hydrological models

We now analyse the results of the four regional models that we trained for the four investigated HUCs in Experiment 2.

Figure 10 shows the difference in the NSE between the model outputs from Experiments 1 and 2. For some basins, the regional r
perform significantly worse (dark red) than the individually trained models from Experiment 1. However, from the histograms of
differences we can see that the median is almost zero, meaning that in 50 % of the basins the regional model performs better th
specifically trained for a single basin. Especially in the New England region the regional model performed better for almost all ba
for two in the far north-east). In general, for all HUCs and catchments, the median difference is −0.001.

From Fig. 11 it is evident that the increased data size of the regional modelling approach (Experiment 2) helps to attenuate the d
performance between the calibration and validation periods, which could be observed in Experiment 1 probably as a result of ove
the CDF of the NSE (Fig. 11a) we can see that Experiment 2 performed worse for approximately 20 % of the basins, while being
or even slightly better for the remaining watersheds. We can also observe that the regional models show a more balanced under
estimation, while the models from Experiment 1 as well as the benchmark model tend to underestimate the discharge (see Fig. 1
the flow variance, the top 2 % flow bias or the bias of the middle flows). This is not too surprising, since we train one model on a
different basins with different discharge characteristics, where the model minimizes the error between simulated and observed d
all basins at the same time. On average, the regional model will therefore equally over- and under-estimate the observed discha

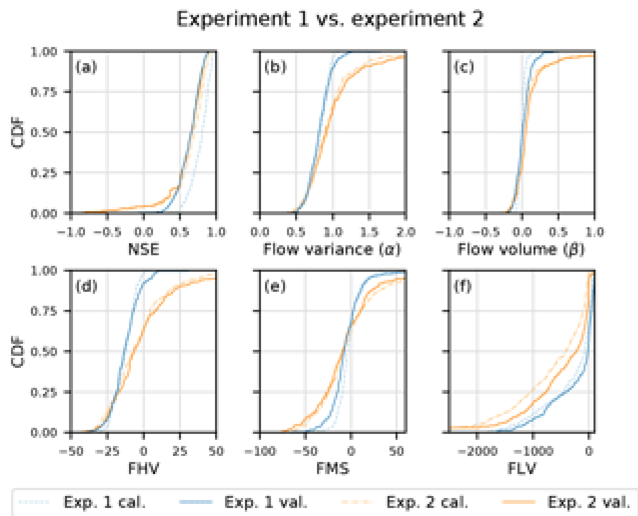


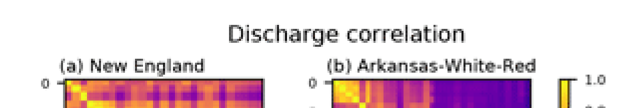
Figure 11 Cumulative density functions for several metrics of the calibration and validation period of the models from Experim
compared to the regional models from Experiment 2. FHV is the bias of the top 2 % flows, the peak flows, FMS is the slope of t
section of the flow duration curve and FLV is the bias of the bottom 30 % low flows.

Download

The comparison of the performances of Experiment 1 and 2 shows no clear consistent pattern for the investigated HUCs, but rev
toward higher NSE values in the New England region and to lower NSE values in the Arkansas-White-Red region. The reason for
differences might become clearer once we look at the correlation in the observed discharge time series of the basins within both
Fig. 12). We can see that in the New England region (where the regional model performed better for most of the catchments cor
individual models of Experiment 1) many basins have a strong correlation in their discharge time series. Conversely, for Arkansa
region the overall image of the correlation plot is much different. While some basins exist in the eastern part of the HUC with dis
correlation, especially the basins in the western, more arid part have no inter-correlation at all. The results suggest that a single
calibrated LSTM could generally be better in predicting the discharge of a group of basins compared to many LSTMs trained sepa
each of the basins within the group especially when the group's basins exhibit a strong correlation in their discharge behaviour.

3.4 The effect of fine-tuning

In this section, we analyse the effect of fine-tuning the regional model for a few number of epochs to a specific catchment.



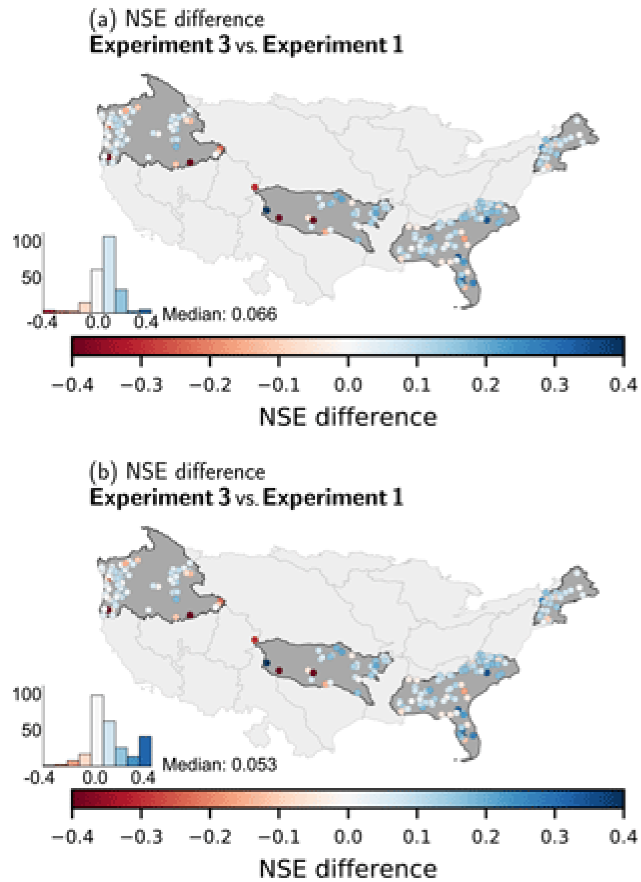


Figure 13 Panel **(a)** shows the difference of the NSE in the validation period of Experiment 3 compared to the models of Expe and panel **(b)** in comparison to the models of Experiment 2. Blue colours (>0) indicate in both cases that the fine-tuned mode Experiment 3 perform better and red colours (<0) the opposite. The NSE differences are capped at [−0.4, 0.4] for better visua

Download

3.5 A hydrological interpretation of the LSTM

To round off the discussion of this manuscript, we want to come back to the LSTM and try to explain it again in comparison to th of a classical hydrological model. Similar to continuous hydrological models, the LSTM processes the input data time step after ti every time step, the input data (here meteorological forcing data) are used to update a number of values in the LSTM internal ce comparison to traditional hydrological models, the cell states can be interpreted as storages that are often used for e.g. snow ac soil water content, or groundwater storage. Updating the internal cell states (or storages) is regulated through a number of so-c one that regulates the depletion of the storages, a second that regulates the increase in the storages and a third that regulates t the storages. Each of these gates comes with a set of adjustable parameters that are adapted during a calibration period (referre *training*). During the validation period, updates of the cell states depend only on the input at a specific time step and the states time step (given the *learned* parameters of the calibration period).

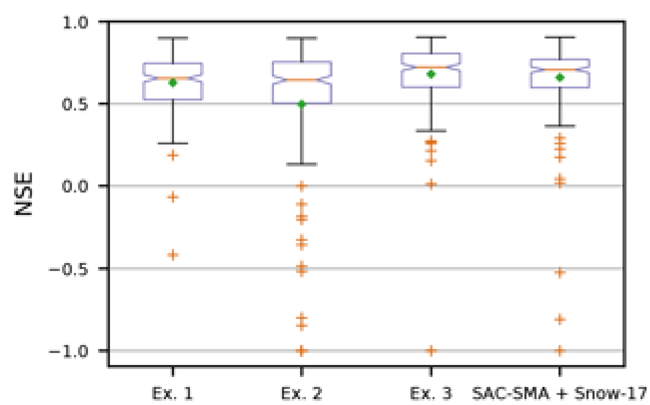


Figure 14 Boxplot of the NSE of the validation period for our three Experiments and the benchmark model. The NSE is cappe

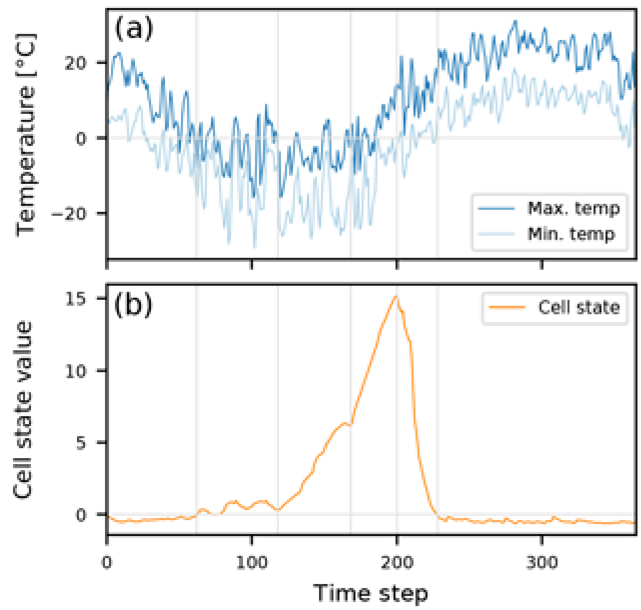


Figure 15 Evolution of a specific cell state in the LSTM **(b)** compared to the daily min and max temperature, with accumulatio
and depletion in spring **(a)**. The vertical grey lines are included for better guidance.

Download

4 Summary and conclusion

This contribution investigated the potential of using Long Short-Term Memory networks (LSTMs) for simulating runoff from mete
observations. LSTMs are a special type of recurrent neural networks with an internal memory that has the ability to learn and sto
dependencies of the input–output relationship. Within three experiments, we explored possible applications of LSTMs and demon
they are able to simulate the runoff with competitive performance compared to a baseline hydrological model (here the SAC-SMA
model). In the first experiment we looked at classical single basin modelling, in a second experiment we trained one model for a
each of the regions we investigated, and in a third experiment we showed that using a pre-trained model helps to increase the m
performance in single basins. Additionally, we showed an illustrative example why traditional RNNs should be avoided in favour o
the task is to predict runoff from meteorological observations.

The goal of this study was to explore the potential of the method and not to obtain the best possible realization of the LSTM mod
catchment (see Sect. 2.5). It is therefore very likely that better performing LSTMs can be found by an exhaustive (catchment-wi
hyperparameter search. However, with our simple calibration approach, we were already able to obtain comparable (or even slig
model performances compared to the well-established SAC-SMA + Snow-17 model.

In summary, the major findings of the present study are the following.

a. LSTMs are able to predict runoff from meteorological observations with accuracies comparable to the well-established S
SMA + Snow-17 model.

b. The 15 years of daily data used for calibration seem to constitute a lower bound of data requirements.

c. Pre-trained knowledge can be transferred into different catchments, which might be a possible approach for reducing th
demand and/or regionalization applications, as well as for prediction in ungauged basins or basins with few observation

The data intensive nature of the LSTMs (as for any deep learning model) is a potential barrier for applying them in data-scarce p
for the usage within a single basin with limited data). We do believe that the use of “pre-trained LSTMs” (as explored in Experim
promising way to reduce the large data demand for an individual basin. However, further research is needed to verify this hypot
Ultimately, however, LSTMs will always strongly rely on the available data for calibration. Thus, even if less data are needed, it c
a disadvantage in comparison to physically based models, which – at least in theory – are not reliant on calibration and can thus
with ease to new situations or catchments. However, more and more large-sample data sets are emerging which will catalyse fut
applications of LSTMs. In this context, it is also imaginable that adding physical catchment properties as an additional input laye
LSTM may enhance the predictive power and ability of LSTMs to work as regional models and to make predictions in ungauged b

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S. I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., M Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vi Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Sys available at: https://www.tensorflow.org/ (last access: 21 November 2018), 2016. a

Abrahart, R. J., Anctil, F., Coulibaly, P., Dawson, C. W., Mount, N. J., See, L. M., Shamseldin, A. Y., Solomatine, D. P., Toth, E., a R. L.: Two decades of anarchy? Emerging themes and outstanding challenges for neural network river forecasting, Prog. Phys. G 480–513, 2012. a

Adams, T. E. and Pagaon, T. C. (Eds.): Flood Forecasting: A Global Perspective, Academic Press, Boston, MA, USA, 2016. a

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P.: The CAMELS data set: catchment attributes and meteorology for large studies, Hydrol. Earth Syst. Sci., 21, 5293–5313, https://doi.org/10.5194/hess-21-5293-2017, 2017a. a, b, c

Addor, N., Newman, A. J., Mizukami, N., and Clark, M. P.: Catchment attributes for large-sample studies, UCAR/NCAR, Boulder, https://doi.org/10.5065/D6G73C3Q, 2017b. a, b

Anderson, E. A.: National Weather Service River Forecast System - Snow Accumulation and Ablation Model, Tech. Rep. Novembe Department of Commerce, Silver Spring, USA, 1973. a

ASCE Task Committee on Application of Artificial Neural Networks: Artificial Neural Networks in Hydrology. II: Hydrologic Applica Hydrol. Eng., 52, 124–137, 2000. a

Assem, H., Ghariba, S., Makrai, G., Johnston, P., Gill, L., and Pilla, F.: Urban Water Flow and Water Level Prediction Based on De in: ECML PKDD 2017: Machine Learning and Knowledge Discovery in Databases, 317–329, Springer, Cham., 2017. a

Bengio, Y.: Practical recommendations for gradient-based training of deep architectures, in: Neural networks: Tricks of the trade Springer, Berlin, Heidelberg, 2012. a

Bengio, Y., Simard, P., and Frasconi, P.: Learning long-term dependencies with gradient descent is difficult, IEEE T. Neural Netw 166, 1994. a, b

Beven, K.: Linking parameters across scales: subgrid parameterizations and scale dependent hydrological models, Hydrol. Proces 525, 1995. a

Beven, K.: Rainfall-Runoff Modelling: The Primer, John Wiley & Sons, Chichester, UK, 2001. a

Blöschl, G., Sivapalan, M., Wagener, T., Viglione, A., and Savenije, H. (Eds.): Runoff Prediction in Ungauged Basins: Synthesis A Processes, Places and Scales, Cambridge University Press, UK, 465 pp., 2013. a

Burnash, R. J. C., Ferral, R. L., and McGuire, R. A.: A generalised streamflow simulation system conceptual modelling for digital Tech. rep., US Department of Commerce National Weather Service and State of California Department of Water Resources, Sacr USA, 1973. a

Buytaert, W. and Beven, K.: Regionalization as a learning process, Water Resour. Res., 45, 1–13, 2009. a

Carriere, P., Mohaghegh, S., and Gaskar, R.: Performance of a Virtual Runoff Hydrographic System, Water Resources Planning a Management, 122, 120–125, 1996. a

Chollet, F.: Keras, available at: https://github.com/fchollet/keras (last access: 1 April 2018), 2015. a

Clark, M. P., Bierkens, M. F. P., Samaniego, L., Woods, R. A., Uijlenhoet, R., Bennett, K. E., Pauwels, V. R. N., Cai, X., Wood, A. Peters-Lidard, C. D.: The evolution of process-based hydrologic models: historical challenges and the collective quest for physica Hydrol. Earth Syst. Sci., 21, 3427–3440, https://doi.org/10.5194/hess-21-3427-2017, 2017. a

Daniell, T. M.: Neural networks. Applications in hydrology and water resources engineering, in: Proceedings of the International and Water Resource Symposium, vol. 3, 797–802, Institution of Engineers, Perth, Australia, 1991. a

Duan, Q., Gupta, V. K., and Sorooshian, S.: Shuffled complex evolution approach for effective and efficient global minimization, Theory App., 76, 501–521, 1993. a

Fang, K., Shen, C., Kifer, D., and Yang, X.: Prolongation of SMAP to Spatiotemporally Seamless Coverage of Continental U.S. Us Learning Neural Network, Geophys. Res. Lett., 44, 11030–11039, 2017. a

Farabet, C., Couprie, C., Najman, L., and Lecun, Y.: Learning Hierarchical Features for Scene Labeling, IEEE T. Pattern Anal., 35

Hestness, J., Narang, S., Ardalani, N., Diamos, G. F., Jun, H., Kianinejad, H., Patwary, M. M. A., Yang, Y., and Zhou, Y.: Deep Le Scaling is Predictable, Empirically, available at: https://arxiv.org/abs/1712.00409 (last access: 21 November 2018), 2017. a

Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A.-R., Jaitly, N., Senior, A., Vanhoucke, V., Nguyen, P., Sainath, T. N., and K Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups, IEEE Signal Pro 82–97, 2012. a

Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, Neural Comput., 9, 1735–1780, 1997. a, b, c, d

Hsu, K., Gupta, H. V., and Soroochian, S.: Application of a recurrent neural network to rainfall-runoff modeling, Proc., Aesthetics Constructed Environment, ASCE, New York, 68–73, 1997. a, b

Hunter, J. D.: Matplotlib: A 2D graphics environment, Comput. Sci. Eng., 9, 90–95, 2007. a

Kirchner, J. W.: Getting the right answers for the right reasons: Linking measurements, analyses, and models to advance the sci hydrology, Water Resour. Res., 42, 1–5, 2006. a

Kollet, S. J., Maxwell, R. M., Woodward, C. S., Smith, S., Vanderborght, J., Vereecken, H., and Simmer, C.: Proof of concept of r hydrologic simulations at hydrologic resolution utilizing massively parallel computer resources, Water Resour. Res., 46, 1–7, 201

Krizhevsky, A., Sutskever, I., and Hinton, G. E.: ImageNet Classification with Deep Convolutional Neural Networks, Adv. Neur. In 1105, 2012. a

Kumar, D. N., Raju, K. S., and Sathish, T.: River Flow Forecasting using Recurrent NeuralNetworks, Water Resour. Manag., 18, 1 2004. a

LeCun, Y. A., Bottou, L., Orr, G. B., and Müller, K. R.: Efficient backprop, Springer, Berlin, Heidelberg, Germany, 2012. a, b

Lindström, G., Pers, C., Rosberg, J., Strömqvist, J., and Arheimer, B.: Development and testing of the HYPE (Hydrological Predic Environment) water quality model for different spatial scales, Hydrol. Res., 41, 295–319, 2010. a

Marçais, J. and de Dreuzy, J. R.: Prospective Interest of Deep Learning for Hydrological Inference, Groundwater, 55, 688–692, 2

Maurer, E. P., Wood, A. W., Adam, J. C., Lettenmaier, D. P., and Nijssen, B.: A long-term hydrologically based dataset of land su and states for the conterminous United States, J. Climate, 15, 3237–3251, 2002. a

McKinney, W.: Data Structures for Statistical Computing in Python, Proceedings of the 9th Python in Science Conference, Locatio Texas, USA, 1697900, 51–56, available at: http://conference.scipy.org/proceedings/scipy2010/mckinney.html (last access: 1 Ap 2010. a

Merz, R., Blöschl, G., and Parajka, J.: Regionalisation methods in rainfall-runoff modelling using large samples, Large Sample Ba Experiments for Hydrological Model Parameterization: Results of the Model Parameter Experiment–MOPEX, IAHS Publ., 307, 117 a

Minns, A. W. and Hall, M. J.: Artificial neural networks as rainfall-runoff models, Hydrolog. Sci. J., 41, 399–417, 1996. a

Mu, Q., Zhao, M., and Running, S. W.: Improvements to a MODIS global terrestrial evapotranspiration algorithm, Remote Sens. 115, 1781–1800, 2011. a

Mulvaney, T. J.: On the use of self-registering rain and flood gauges in making observations of the relations of rainfall and of floo in a given catchment, in: Proceedings Institution of Civil Engineers, Dublin, Vol. 4, 18–31, 1850. a

Myneni, R. B., Hoffman, S., Knyazikhin, Y., Privette, J. L., Glassy, J., Tian, Y., Wang, Y., Song, X., Zhang, Y., Smith, G. R., Lotsc M., Morisette, J. T., Votava, P., Nemani, R. R., and Running, S. W.: Global products of vegetation leaf area and fraction absorbed year one of MODIS data, Remote Sens. Environ., 83, 214–231, 2002. a

Nash, J. E. and Sutcliffe, J. V.: River Flow Forecasting Through Conceptual Models Part I-a Discussion of Principles, J. Hydrol., 10 1970. a

Newman, A., Sampson, K., Clark, M., Bock, A., Viger, R., and Blodgett, D.: A large-sample watershed-scale hydrometeorological the contiguous USA, UCAR/NCAR, Boulder, CO, USA, https://doi.org/10.5065/D6MW2F4D, 2014. a, b

Newman, A. J., Clark, M. P., Sampson, K., Wood, A., Hay, L. E., Bock, A., Viger, R. J., Blodgett, D., Brekke, L., Arnold, J. R., Hop Duan, Q.: Development of a large-sample watershed-scale hydrometeorological data set for the contiguous USA: data set charac assessment of regional variability in hydrologic model performance, Hydrol. Earth Syst. Sci., 19, 209–223, https://doi.org/10.51 209-2015, 2015. a, b, c, d

Shen, C., Laloy, E., Elshorbagy, A., Albert, A., Bales, J., Chang, F.-J., Ganguly, S., Hsu, K.-L., Kifer, D., Fang, Z., Fang, K., Li, D
Tsai, W.-P.: HESS Opinions: Incubating deep-learning-powered hydrologic science advances as a community, Hydrol. Earth Syst
5639–5656, https://doi.org/10.5194/hess-22-5639-2018, 2018. a

Shi, X., Chen, Z., Wang, H., Yeung, D.-Y., Wong, W.-K., and Woo, W.-C.: Convolutional LSTM network: A machine learning appr
precipitation nowcasting, Adv. Neur. In., 28, 802–810, 2015. a

Sivapalan, M.: Prediction in ungauged basins: a grand challenge for theoretical hydrology, Hydrol. Process., 17, 3163–3170, 200

Solomatine, D., See, L. M., and Abrahart, R. J.: Data-driven modelling: concepts, approaches and experiences, in: Practical hydr
17–30, Springer, Berlin, Heidelberg, 2009. a

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Netwo
Overfitting, J. Mach. Learn. Res., 15, 1929–1958, 2014. a

Stanzel, P., Kahl, B., Haberl, U., Herrnegger, M., and Nachtnebel, H. P.: Continuous hydrological modelling in the context of real
forecasting in alpine Danube tributary catchments, IOP C. Ser. Earth Env., 4, 012005, https://doi.org/10.1088/1755-1307/4/1/0
2008. a

Sutskever, I., Vinyals, O., and Le, Q. V.: Sequence to sequence learning with neural networks, in: Advances in neural informatio
systems, 3104–3112, 2014. a

Tao, Y., Gao, X., Hsu, K., Sorooshian, S., and Ihler, A.: A Deep Neural Network Modeling Framework to Reduce Bias in Satellite I
Products, J. Hydrometeorol., 17, 931–945, 2016. a

Thielen, J., Bartholmes, J., Ramos, M.-H., and de Roo, A.: The European Flood Alert System – Part 1: Concept and development
Earth Syst. Sci., 13, 125–140, https://doi.org/10.5194/hess-13-125-2009, 2009. a

Thornton, P. E., Thornton, M. M., Mayer, B. W., Wilhelmi, N., Wei, Y., Devarakonda, R., and Cook, R.: Daymet: Daily surface wea
km grid for North America, 1980–2008, Oak Ridge National Laboratory (ORNL) Distributed Active Archive Center for Biogeochem
(DAAC), Oak Ridge, Tennessee, USA, 2012. a

Tompson, J., Jain, A., LeCun, Y., and Bregler, C.: Joint Training of a Convolutional Network and a Graphical Model for Human Pos
in: Proceedings of Advances in Neural Information Processing Systems, 27, 1799–1807, 2014. a

Van Der Walt, S., Colbert, S. C., and Varoquaux, G.: The NumPy array: A structure for efficient numerical computation, Comput.
13, 22–30, 2011. a

van Rossum, G.: Python tutorial, Technical Report CS-R9526, Tech. rep., Centrum voor Wiskunde en Informatica (CWI), Amster
Netherlands, 1995. a

Wesemann, J., Herrnegger, M., and Schulz, K.: Hydrological modelling in the anthroposphere: predicting local runoff in a heavily
high-alpine catchment, J. Mt. Sci., 15, 921–938, 2018. a

Wood, E. F., Roundy, J. K., Troy, T. J., van Beek, L. P. H., Bierkens, M. F. P., Blyth, E., de Roo, A., Döll, P., Ek, M., Famiglietti, J.
van de Giesen, N., Houser, P., Jaffé, P. R., Kollet, S., Lehner, B., Lettenmaier, D. P., Peters-Lidard, C., Sivapalan, M., Sheffield,
and Whitehead, P.: Hyperresolution global land surface modeling: Meeting a grand challenge for monitoring Earth's terrestrial wa
Resour. Res., 47, W05301, https://doi.org/10.1029/2010WR010090, 2011. a

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., Luo, L., Alonge, C., Wei, H., Meng, J., and Livneh, B.: Contine
water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2)
1. Intercomparison and application of model products, J. Geophys. Res.-Atmos., 117, D03109, https://doi.org/10.1029/2011JD0
a

Yilmaz, K. K., Gupta, H. V., and Wagener, T.: A process-based diagnostic approach to model evaluation: Application to the NWS
hydrologic model, Water Resour. Res., 44, 1–18, 2008. a

Yosinski, J., Clune, J., Bengio, Y., and Lipson, H.: How transferable are features in deep neural networks?, Adv. Neur. In., 27, 1-

Young, P. C. and Beven, K. J.: Data-based mechanistic modelling and the rainfall-flow non-linearity, Environmetrics, 5, 335–363

Zhang, D., Lindholm, G., and Ratnaweera, H.: Use long short-term memory to enhance Internet of Things for combined sewer o
monitoring, J. Hydrol., 556, 409–418, 2018. a

Zhang, J., Zhu, Y., Zhang, X., Ye, M., and Yang, J.: Developing a Long Short-Term Memory (LSTM) based model for predicting w
depth in agricultural areas, J. Hydrol., 561, 918–929, 2018. a