Hydrol. Earth Syst. Sci., 25, 1671–1687, 2021
https://doi.org/10.5194/hess-25-1671-2021

Hydrology and Earth System Sciences
Open Access

Groundwater level forecasting with artificial neural networks: a comparison of long short-term memory (LSTM), convolutional neural networks (CNNs), and non-linear autoregressive networks with exogenous input (NARX)

Andreas Wunsch[1], **Tanja Liesch**[1], **and Stefan Broda**[2]

[1]Hydrogeology, Karlsruhe Institute of Technology (KIT), Institute of Applied Geosciences, Kaiserstr. 12, 76131 Karlsruhe, Germany
[2]Federal Institute for Geosciences and Natural Resources (BGR), Wilhelmstr. 25–30, 13593 Berlin, Germany

Correspondence: Andreas Wunsch (andreas.wunsch@kit.edu)

Received: 23 October 2020 – Discussion started: 23 November 2020
Revised: 1 February 2021 – Accepted: 2 March 2021 – Published: 1 April 2021

Abstract. It is now well established to use shallow artificial neural networks (ANNs) to obtain accurate and reliable groundwater level forecasts, which are an important tool for sustainable groundwater management. However, we observe an increasing shift from conventional shallow ANNs to state-of-the-art deep-learning (DL) techniques, but a direct comparison of the performance is often lacking. Although they have already clearly proven their suitability, shallow recurrent networks frequently seem to be excluded from the study design due to the euphoria about new DL techniques and its successes in various disciplines. Therefore, we aim to provide an overview on the predictive ability in terms of groundwater levels of shallow conventional recurrent ANNs, namely non-linear autoregressive networks with exogenous input (NARX) and popular state-of-the-art DL techniques such as long short-term memory (LSTM) and convolutional neural networks (CNNs). We compare the performance on both sequence-to-value (seq2val) and sequence-to-sequence (seq2seq) forecasting on a 4-year period while using only few, widely available and easy to measure meteorological input parameters, which makes our approach widely applicable. Further, we also investigate the data dependency in terms of time series length of the different ANN architectures. For seq2val forecasts, NARX models on average perform best; however, CNNs are much faster and only slightly worse in terms of accuracy. For seq2seq forecasts, mostly NARX outperform both DL models and even almost reach the speed of CNNs. However, NARX are the least robust against initialization effects, which nevertheless can be handled easily using ensemble forecasting. We showed that shallow neural networks, such as NARX, should not be neglected in comparison to DL techniques especially when only small amounts of training data are available, where they can clearly outperform LSTMs and CNNs; however, LSTMs and CNNs might perform substantially better with a larger dataset, where DL really can demonstrate its strengths, which is rarely available in the groundwater domain though.

1 Introduction

Groundwater is the only possibility for 2.5 billion people worldwide to cover their daily water needs (UNESCO, 2012). and at least half of the global population uses groundwater for drinking-water supplies (WWAP, 2015). Moreover, groundwater also constitutes for a substantial amount of global irrigation water (FAO, 2010), which altogether and among other factors such as population growth and climate change make it a vital future challenge to dramatically improve the way of using, managing, and sharing water (WWAP, 2015). Accurate and reliable groundwater level (GWL) forecasts are a key tool in this context, as they provide important information on the quantitative availability of groundwater and can thus form the basis for management decisions and strategies.

Published by Copernicus Publications on behalf of the European Geosciences Union.

Especially due to the success of deep-learning (DL) approaches in recent years and their more and more widespread application in our daily life, DL is starting to transform traditional industries and is also increasingly used across multiple scientific disciplines (Shen, 2018). This applies as well to water sciences, where machine learning methods in general are used in a variety of ways, as data-driven approaches offer the possibility to directly address questions on relationships between relevant input forcings and important system variables, such as run-off or groundwater level, without the need to build classical models and explicitly define physical relationships. This is especially handy because these classical models might sometimes be oversimplified or (in the case of numerical models) data hungry, difficult, time-consuming to set up and maintain, and therefore expensive. In particular artificial neural networks (ANNs) have been successfully applied to research related to a variety of surface-water (Maier et al., 2010) and groundwater-level (Rajaee et al., 2019) questions already; however, especially DL was used only gradually at first (Shen, 2018) but is just about to take off, which is reflected in the constantly increasing number of DL and water-resource-related publications (see e.g. Chen et al., 2020; Duan et al., 2020; Fang et al., 2019, 2020; Gauch et al., 2020, 2021; Klotz et al., 2020; Kraft et al., 2020; Kratzert et al., 2018, 2019a, b; Pan et al., 2020; Rahmani et al., 2021). In this work we explore and compare the abilities of non-linear autoregressive networks with exogenous input (NARX), which have been successfully applied multiple times to groundwater level forecasting in the past and to the currently popular DL approaches of long short-term memory (LSTM) and convolutional neural networks (CNNs).

During the last years several authors have shown the ability of NARX to successfully model and forecast groundwater levels (Alsumaiei, 2020; Chang et al., 2016; Di Nunno and Granata, 2020; Guzman et al., 2017, 2019; Hasda et al., 2020; Izady et al., 2013; Jeihouni et al., 2019; Jeong and Park, 2019; Wunsch et al., 2018; Zhang et al., 2019). Although LSTMs and CNNs are state-of-the-art DL techniques and commonly applied in many disciplines, they are not yet widely adopted in groundwater level prediction applications, except within the last 2 years. Thereby, LSTMs were used twice as often to predict groundwater levels (Afzaal et al., 2020; Bowes et al., 2019; Jeong et al., 2020; Jeong and Park, 2019; Müller et al., 2020; Supreetha et al., 2020; Zhang et al., 2018) compared to CNNs (Afzaal et al., 2020; Lähivaara et al., 2019; Müller et al., 2020). The main reason might be that the strength of CNNs is mainly the extraction of spatial information from image-like data, whereas LSTMs are especially suited to process sequential data, such as from time series. Overall, these studies show that LSTMs and CNNs are very well suited to forecast groundwater levels. Both Afzaal et al. (2020) and Müller et al. (2020) also directly compared the performance of LSTMs and CNNs, but no clear superiority of one over the other can be drawn from their results. Müller et al. (2020), who focus on hyperparameter optimization, draw the conclusion that CNN results are less robust compared to LSTM predictions; however, other analyses in their study also show better results of CNNs compared to LSTMs. Jeong and Park (2019) conducted a comparison of NARX and LSTM (and others) performance on groundwater level forecasting. They found both NARX and LSTM to be the best models in their overall comparison concerning the prediction accuracy; however, they used a deep NARX model with more than one hidden layer. To the best of the authors' knowledge, no direct comparison has yet been made of (shallow) NARX, LSTMs, and CNNs to predict groundwater levels.

In this study we aim to provide an overview on the predictive ability in terms of groundwater levels of shallow conventional recurrent ANNs, namely NARX and popular state-of-the-art DL techniques LSTM and (1D) CNNs. We compare the performance of both on single-value (sequence-to-value; also known as one-step-ahead, sequence-to-one, or many-to-one forecasting) and sequence (sequence-to-sequence) forecasting. We use data from 17 groundwater wells within the Upper Rhine Graben region in Germany and France, which was selected based on prior knowledge and representing the full bandwidth of groundwater dynamic types in the region. Further, we use only widely available and easy-to-measure meteorological input parameters (precipitation, temperature, and relative humidity), which makes our approach widely applicable. All models are optimized using Bayesian optimization models, which we extend to also solve the common input parameter selection problem by considering the inputs as optimizable parameters. Further, the data dependency of all models is explored in a simple experimental setup for which there are substantial differences in shallow- and deep-learning models in the need for training data, as one might suspect.

2 Methodology

2.1 Input parameters

In this study we only use the meteorological input variables precipitation (P), temperature (T), and relative humidity (rH), which in general are widely available and easy to measure. In principle, this makes this approach easily transferable and thus applicable almost everywhere. Precipitation may serve as a surrogate for groundwater recharge; temperature and relative humidity include the relationship with evapotranspiration and at the same time provide the network with information on seasonality due to the usually distinct annual cycle. As an additional synthetic input parameter, a sinusoidal signal fitted to the temperature curve (T_{sin}) can provide the model with noise-free information on seasonality, which often allows for significantly improved predictions to be made (Kong-A-Siou et al., 2014). Without doubt, the most important input parameter out of these is P, since groundwater recharge usually has the greatest influence on groundwater dynamics. Therefore, P is used always as input parameter; the suitability of the remaining parameters is checked and optimized for each time series and each model individually. The fundamental idea is that for wells with primarily natural groundwater dynamics, the relationship between groundwater levels and the important processes of groundwater recharge and evapotranspiration should be mapped via the meteorological parameters P, T, and rH. However, especially for wells with a dynamic influenced by other factors, this is usually only valid to a limited extent, since groundwater dynamics can depend on various additional factors such as groundwater extractions or surface water interactions. Due to a typically strong autocorrelation of groundwater level time series, a powerful predictor for the future groundwater level is the groundwater level in the past. Depending on the purpose and methodological setup, it does not always make sense to include this parameter; however, where meaningful we explored also past GWL as inputs.

2.2 Nonlinear autoregressive exogenous model (NARX)

Non-linear autoregressive models with exogenous input relate the current value of a time series to past values of the same time series, as well as to current and past values of additional exogenous time series. We implement this type of model as a recurrent neural network (RNN), which extends the well-known feed-forward multilayer perceptron structure (MLP) by a global feedback connection between output and input layers. One can therefore also refer to it as recurrent MLP. NARX are frequently applied for non-linear time series prediction and non-linear filtering tasks (Beale et al., 2016). Similar to other types of RNNs, NARX have also difficulties in capturing long-term dependencies due to the problem of vanishing and exploding gradients (Bengio et al., 1994), yet they can keep information up to 3 times longer than simple RNNs (Lin et al., 1996a, b), so they can converge more quickly and generalize better in comparison (Lin et al., 1998). Using the recurrent connection, future outputs are both regressed on independent inputs and on previous outputs (groundwater levels in our case), which is the standard configuration for multi-step prediction and also known as closed-loop configuration. However, NARX can also be trained by using the open-loop configuration, where the observed target is presented as an input instead of feeding back the estimated output. This configuration can make training more accurate and efficient, as well as computationally less expensive, because learning algorithms do not have to handle recurrent connections (Moghaddamnia et al., 2009). However, experience shows that both configurations can be adequate for training a NARX model, since open-loop training often results in more accurate performance in terms of mean errors, whereas closed-loop trained models often are better in capturing the general dynamics of a time series. NARX also contain a short-term memory, i.e. delay vectors for each input (and feedback), which allow for the availability of several input time steps simultaneously, depending on the length of the vector. Usually, delays are crucial for the performance of NARX models. Please note that some of our experiments include past GWLs for training (compare Sect. 2.1), which is also performed in closed-loop setup and thus uses both multiple observed past GWLs (according to the size of the input delay) as an input, as well as multiple simulated GWLs (according to the size of the feedback delay) via the feedback connection. In a way this mimics the open-loop setup; however, we still use the feedback connection and simply treat the past observed GWL as an additional input.

The given configuration describes sequence-to-value forecasting. To perform sequence-to-sequence forecasts, some modifications are necessary. As other ANNs, NARX are capable of performing forecasts of a complete sequence at once; i.e. one output neuron predicts a vector with multiple values. Technically it is necessary to use the same length for input and output sequences. To build and apply NARX models, we use MATLAB 2020a (Mathworks Inc., 2020) and its Deep Learning Toolbox.

2.3 Long short-term memory (LSTM)

Long short-term memory networks are recurrent neural networks which are widely applied to model sequential data like time series or natural language. As stated, RNNs suffer from the vanishing gradient problem during backpropagation, and in the case of simple RNNs, their memory barely includes the previous 10 time steps (Bengio et al., 1994). LSTMs, however, can remember long-term dependencies because they have been explicitly designed to overcome this problem (Hochreiter and Schmidhuber, 1997). Besides the hidden state of RNNs, LSTMs have a cell memory (or cell state) to store information and three gates to control the information flow (Hochreiter and Schmidhuber, 1997). The forget gate (Gers et al., 2000) controls which and how much information of the cell memory is forgotten, the input gate controls which inputs are used to update the cell memory, and the output gate controls which elements of the cell memory are used to update the hidden state of the LSTM cell. The cell memory enables the LSTM to handle long-term dependencies, because information can remain in the memory for many steps (Hochreiter and Schmidhuber, 1997). Several LSTM layers can be stacked on top of each other in a model; however, the last LSTM layer is followed by a traditional fully connected dense layer, which in our case is a single output neuron that outputs the groundwater level. To realize sequence forecasting, as many output neurons in the last dense layer as steps in the sequence are needed. For LSTMs, we rely on Python 3.8 (van Rossum, 1995) in combination with the libraries Numpy (van der Walt et al., 2011), Pandas (McKinney, 2010; Reback et al., 2020), scikit-learn (Pedregosa et al., 2011), and Matplotlib (Hunter, 2007). Further, we use the deep-learning frameworks TensorFlow (Abadi et al., 2015) and Keras (Chollet, 2015).

2.4 Convolutional neural networks (CNNs)

CNNs are neural networks established by LeCun et al. (2015) and are predominantly used for image recognition and classification. However, they also work well on signal processing tasks and are used for natural language processing for example. CNNs usually comprise three different layers. Convolutional layers, the first type, consist of filters and feature maps. The input to a filter is called receptive field and has a fixed size. Each filter is dragged over the entire previous layer resulting in an output, which is collected in the feature map. Convolutional layers are often followed by pooling layers that perform downsampling of the previous layers feature map; thus, information is consolidated by moving a receptive field over the feature map. Such fields apply simple operations like averaging or maximum selection. Similar to LSTM models, multiple convolutional and pooling layers in varying order can be stacked on top of each other in deeper models. The last layer is followed by a fully connected dense layer with one or several output neurons. To realize sequence forecasting, as many output neurons in the last dense layer as steps in the sequence are needed. For CNNs, we equally use Python 3.8 (van Rossum, 1995) in combination with the above-mentioned libraries and frameworks.

2.5 Model calibration and evaluation

In this study we use NARX models with one hidden layer, and we train them in a closed loop using the Levenberg–Marquardt algorithm, which is a fast and reliable second-order local method (Adamowski and Chan, 2011). We choose closed-loop configuration for training, because other hyperparameters (HPs) are optimized using a Bayesian model (see below), which seems to work properly only in closed-loop configuration, probably due to the artificially pushed training performance in open-loop configuration. Optimized HPs are the inputs T, T_{sin}, and rH (1/0, i.e. yes/no); size of the input delays (ID P, ID T, ID T_{sin}, ID rH); size of the feedback delay vector ($\boldsymbol{FD}$); and number of hidden neurons (hidden size). Delays (ID and $\boldsymbol{FD}$) can take values between 1 and 52 (which is 1 year of weekly data); the number of hidden neurons is optimized between 1 and 20. Strictly speaking, input selection is not a hyperparameter optimization problem; however, the algorithm can also be applied to select an appropriate set of inputs (Fig. 1). This assumption applies in our study also to LSTM and CNN models.

We choose our LSTM models to consist of one LSTM layer, followed by a fully connected dense layer with a single output neuron in the case of sequence-to-value forecasting. We use the Adam optimizer with an initial learning rate of 1×10^{-3} and apply gradient clipping to prevent gradients from exploding. Hyperparameters being optimized by a Bayesian model are the number of units within the LSTM layer (hidden size, 1 to 256), the batch size (1 to 256), and the sequence length (1 to 52). The latter can be interpreted more or less as equivalent to the delay size of the NARX models and is often referred to as the number of inputs (Fig. 1).

The CNN models we apply consist of one convolutional layer, a max-pooling layer, and two dense layers, where the second one consists only of one neuron in the case of sequence-to-value forecasting. The Adam optimizer is used with the same configuration as for the LSTM models. For all CNN models, we use a kernel size of 3 and optimize the batch size (1 to 256), sequence length (1 to 52), the number of filters (1 to 256) within the convolutional layer, and the number of neurons within the first dense layer (dense size, 1 to 256) according to a Bayesian optimization model (Fig. 1).

Hyperparameter optimization is conducted by applying Bayesian optimization using the Python implementation by Nogueira (2014). We apply 50 optimization steps as a minimum (25 random exploration steps followed by 25 Bayesian optimization steps). After that, the optimization stops as soon as no improvement has been recorded during 20 steps or after a maximum of 150 steps. For the NARX models, we use the MATLAB built-in Bayesian optimization, where the first 50 steps cannot be distinguished as explained; however, the rest applies accordingly. The acquisition function in all three cases is expected improvement, and the optimization target function we chose is the sum of Nash–Sutcliffe efficiency (NSE) and squared Pearson's correlation coefficient (R^2) (compare Eqs. 1 and 2), because these two criteria are very important and well-established criteria for assessing the forecast accuracy in water-related contexts.

All three model types use 30 as the maximum number of training epochs. To prevent overfitting, we apply early stopping with a patience of 5 steps. The testing or evaluation period in this study for all models are the years 2012 to 2015 inclusively. This period is exclusively used for testing the models. The data before 2012 are of varying length (hydrographs start between 1967 and 1994; see also Fig. 3), depending on the available data, and are split into three parts, namely 80 % for training, 10 % for early stopping, and 10 % for testing during HP optimization (denoted the opt-set) (Fig. 2). Thus, the target function of the HP-optimization procedure is only calculated on the opt-set.

All data are scaled between −1 and 1 and all models are initialized randomly and show therefore a dependency towards the random number generator seed. To minimize initialization influence, we repeat every optimization step five times and take the mean of the target function. For the final model evaluation in the test period (2012–2016), we use 10 pseudo-random initializations and calculate errors of the median forecast. For seq2seq forecasting, we always take the median performance over all forecasted sequences, which have a length of 3 months or 12 steps. This is a realistic length for direct sequence forecasting of groundwater levels, which also has some relevance in practice, because it (i) provides

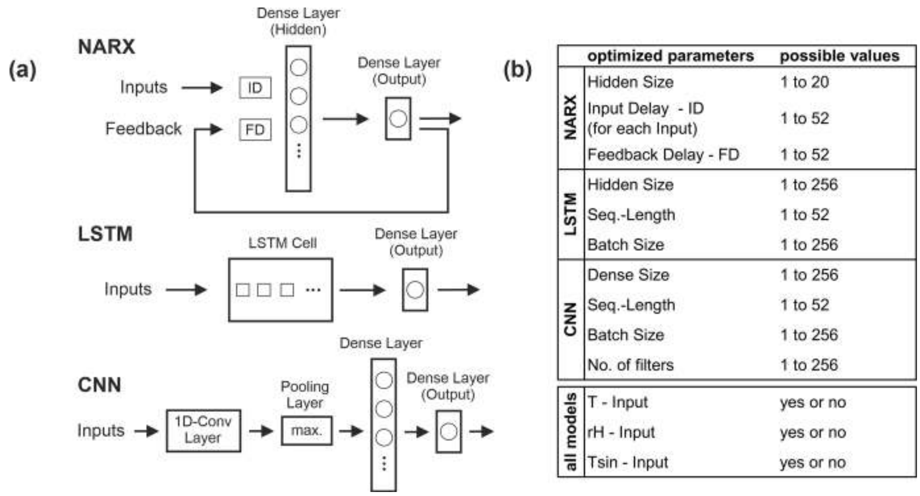

Figure 1. **(a)** Simplified schematic summary of the models and their structures used in this work. ID and FD are delays, circles in dense layers symbolize neurons, and squares within the LSTM cell represent the number of hidden units. **(b)** Hyperparameters (and inputs) of each model used to tune the models by using Bayesian optimization algorithm; the last column summarizes the optimization ranges for each parameter.

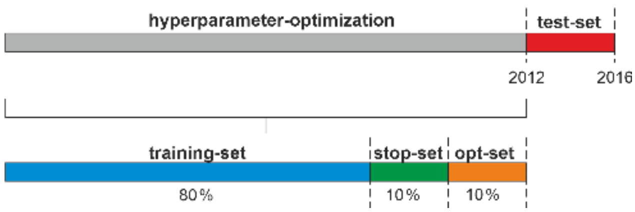

Figure 2. Data splitting scheme: each time series is split into four parts for training, early stopping, HP optimization, and testing. The latter is fixed to the period years 2012 to 2016 for all wells; the former three parts depend on the available time series length.

useful information for many decision-making applications (e.g. groundwater management) and (ii) is also an established time span in meteorological forecasting, known as seasonal forecasts. In principle, this also allows for a performance comparison of 12-step seq2seq forecasts with a potential 12-step seq2val forecast, based on operational meteorological forecasting, where the input uncertainty potentially lowers the groundwater level forecast performance. However, this is beyond the scope of this study, which focuses on neural network architecture comparison.

To judge forecast accuracy, we calculate several metrics: Nash–Sutcliffe efficiency (NSE), squared Pearson's correlation coefficient (R^2), absolute and relative root mean squared errors (RMSE and rRMSE, respectively), absolute and relative biases (Bias and rBias, respectively), and the persistency index (PI). For the following equations, it applies that o represents observed values, p represents predicted values, and n stands for the number of samples.

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{n} (o_i - p_i)^2}{\sum_{i=1}^{n} (o_i - \overline{o})^2} \tag{1}$$

Please note that in the denominator we use the mean observed values until the start of the test period (2012 in the case of our final model evaluation). This best represents the meaning of the NSE, which compares the model performance to the mean values of all known values at the time of the start of the forecast.

$$R^2 = \left(\frac{\sum_{i=1}^{n} (o_i - \overline{o})(p_i - \overline{p})}{\sqrt{\sum_{i=1}^{n} (o_i - \overline{o})^2} \sqrt{\sum_{i=1}^{n} (p_i - \overline{p})^2}} \right)^2 \tag{2}$$

In our case, we use the squared Pearson correlation coefficient R^2 as a general coefficient of determination, since it compares the linear relation between simulated and observed GWLs.

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (o_i - p_i)^2}, \tag{3}$$

$$\mathrm{rRMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^{n} \left(\frac{o_i - p_i}{o_{\max} - o_{\min}} \right)^2}, \tag{4}$$

$$\mathrm{Bias} = \frac{1}{n}\sum_{i=1}^{n}(o_i - p_i)\,, \tag{5}$$

$$\mathrm{rBias} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{o_i - p_i}{o_{\max} - o_{\min}}\right), \tag{6}$$

$$\mathrm{PI} = 1 - \frac{\sum_{i=1}^{n}(o_i - p_i)^2}{\sum_{i=1}^{n}(o_i - o_{\mathrm{last}})^2}. \tag{7}$$

Please note that RMSE and Bias are useful to compare performances for a specific time series among different models; however, only rRMSE and rBias are meaningful to compare model performance between different time series. The persistency index (PI) basically compares the performance to a naïve model that uses the last known observed groundwater level at the time the prediction starts. This is particularly important to judge the performance when past groundwater levels ($\mathrm{GWL}_{(t-1)}$) are used as inputs, because especially in this case the model should outperform a naïve forecast (PI > 0).

2.6 Data dependency

The data dependency of empirical models is a classical research question (Jakeman and Hornberger, 1993), often focusing on the number of parameters but also concerning the length of available data records. Data scarcity is also an important topic in machine learning in general, especially in deep learning and the focus of recent research (e.g. Gauch et al., 2021). One can therefore expect to find performance differences between both shallow and deep models used in this study. We hence performed experiments to explore the need for training data for each of the model types. For this, we started with a reduced training record length of only 2 years before testing the performance on the fixed test set of 4 years (2012–2016). In the following we gradually lengthened the training record until the maximum available length for each well and tracked the error measure changes. This experiment aims to give an impression of how much data might be needed to achieve satisfying forecasting performance and if there are substantial differences between the models; however, it lies out of the scope of interest to answer this very complex question in a general way for each of the modelling approaches.

2.7 Computational aspects

We used different computational setups to build and apply the three model types. We built the NARX models in MATLAB and performed the calculations on the CPU (AMD-Ryzen 9 3900X). The use of a GPU instead of a CPU is not possible for NARX models in our case because of the Levenberg–Marquardt training algorithm, which is not suitable for GPU computation. Both LSTMs and CNNs, however, can be calculated on a GPU, which in the case of LSTMs is the preferred option. For CNNs, we observed a substantially faster calculation (factor 2 to 3) on the CPU and therefore favoured this option. Both LSTMs and CNNs were built and applied using Python 3.8, and the GPU we used for LSTMs was a Nvidia GeForce RTX 2070 Super.

3 Data and study area

In this study we examine the groundwater level forecasting performance at 17 groundwater wells within the Upper Rhine Graben (URG) area (Fig. 3), which is the largest groundwater resource in central Europe (LUBW, 2006). The aquifers of the URG cover 80 % of the drinking water demand of the region as well as the demand for agricultural irrigation and industrial purposes (Région Alsace – Strasbourg, 1999). The wells are selected from a larger dataset from the region with more than 1800 hydrographs. Based on prior knowledge, the wells of this study represent the full bandwidth of groundwater dynamics occurring in the dataset. The whole dataset mainly consists of shallow wells from the uppermost aquifer within the Quaternary sand/gravel sediments of the URG. Mean GWL depths are lower than 5 m b.g.l. for 70 % of the data, rising to a maximum of about 20–30 m towards the Graben edges. The considered aquifers show generally high storage coefficients and high hydraulic conductivities of the order of 1×10^{-4} to $1 \times 10^{-3}\,\mathrm{m\,s^{-1}}$ (LUBW, 2006). In some areas, e.g. the northern URG, strong anthropogenic influences exist, due to intensive groundwater abstractions and management efforts. A list of all examined wells with additional information on identifiers and coordinates can be found in the Supplement (Table S1). All groundwater data are available for free via the web services of the local authorities (HLNUG, 2019; LUBW, 2018; MUEEF, 2018). The shortest time series starts in 1994 and the longest in 1967; however, most hydrographs (12) start between 1980 and 1983 (Fig. 3). Meteorological input data were derived from the HYRAS dataset (Frick et al., 2014; Rauthe et al., 2013), which can be obtained free of charge for non-commercial purposes on request from the German Meteorological Service (DWD, 2021). In this study we exclusively consider weekly time steps for both groundwater and meteorological data.

4 Results and discussion

4.1 Sequence-to-value (seq2val) forecasting performance

Figure 4 summarizes and compares the overall seq2val forecasting accuracy of the three model types for all 17 wells. Figure 4a shows the performance when only meteorological inputs are used; the models in Fig. 4b are additionally provided with GWL_{t-1} as an input. Because the GWL of the last step has to be known, the latter configuration has only

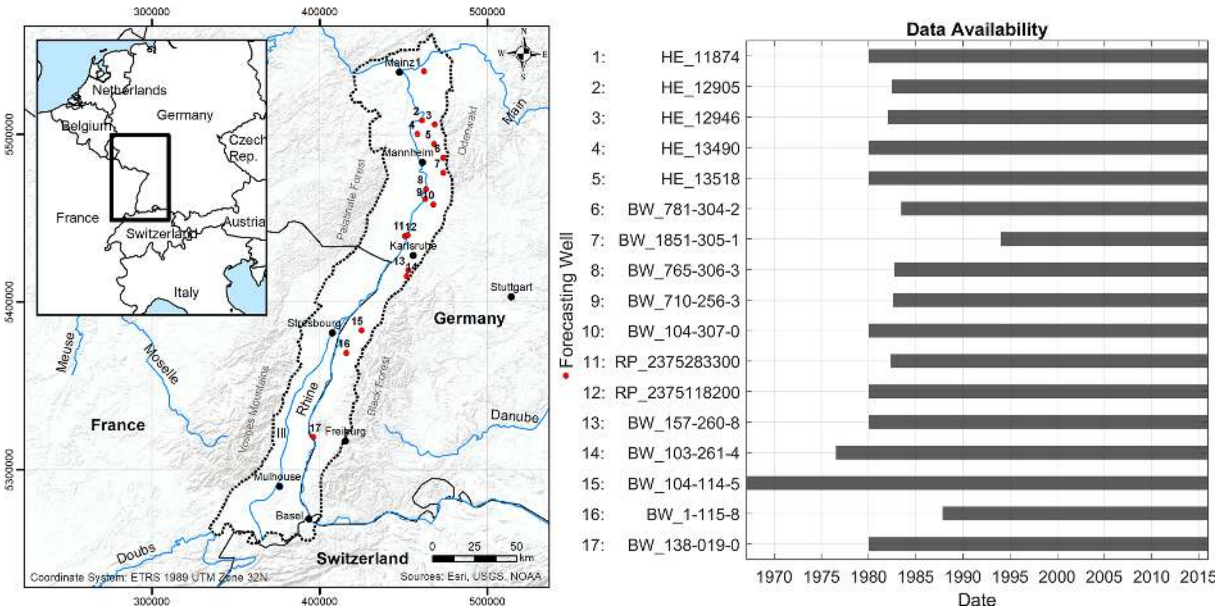

Figure 3. Study area and positions of examined wells (left), as well as respective time series length for each of the wells (right).

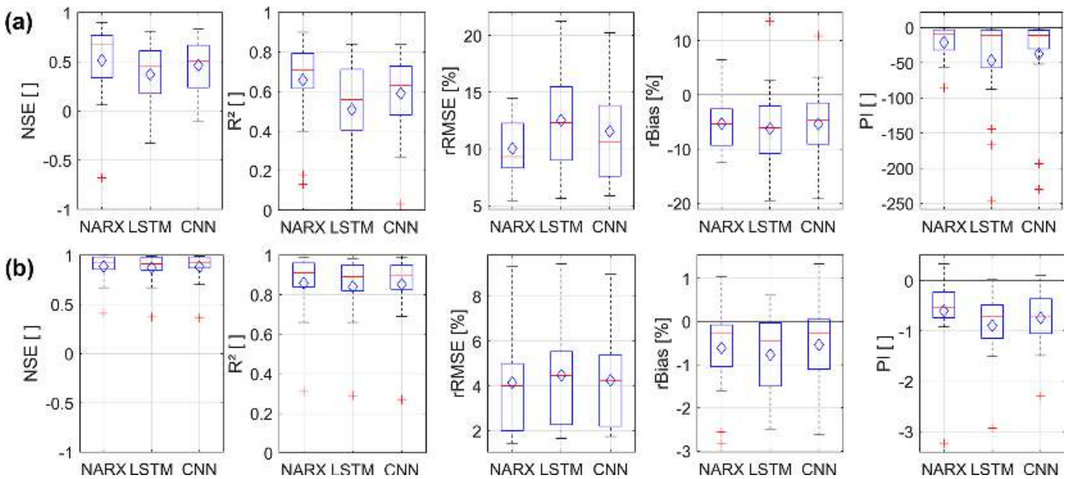

Figure 4. Boxplots showing the seq2val forecast accuracy of NARX, LSTM, and CNN models within the test period (2012–2016) for all considered 17 hydrographs. The diamond symbols indicate the arithmetic mean; **(a)** only meteorological inputs; **(b)** GWL_{t-1} as additional input.

limited value for most applications since only one-step-ahead forecasts are possible in a real-world scenario. However, the inputs of the former configuration are usually available as forecasts themselves for different time horizons. Figure 4a shows that on average NARX models perform best, followed by CNN models; LSTMs achieve the least accurate results. This is consistent for all error measures except rBias, where CNN models show slightly less bias than NARX. However, all models suffer from significant negative bias values of the same order of magnitude, meaning that GWLs are systematically underestimated. Providing information about past groundwater levels up to $t-1$ (GWL_{t-1}) improves the performance of all three models significantly (Fig. 4b). Additionally, performance differences between the models vanish and remain only visible as slight tendencies. This is not surprising, as the past groundwater level is usually a good or even the best predictor of the future GWL, at least for one-step-ahead forecasting, and all models are able to use this information. The general superiority of NARX in the case of Fig. 4a is therefore also expected, as a feedback connection within the model already provides information on past groundwater levels, even though it includes also a certain forecasting error. However, providing GWL_{t-1} as input to a seq2val model (Fig. 4b) basically means providing the naïve model itself, which needs to be outperformed in the case of the PI metric (compare Sect. 2.5). PI values below zero therefore basically mean that the output is worse than the input, which is, apart from the limited benefit for real applications mentioned above, why we refrain from further discussion of the models shown in Fig. 4b.

For our analysis, we did not make a preselection of hydrographs that show predominantly natural groundwater dynamics and thus a comparatively strong relationship between the available input data and the groundwater level. Therefore, even though hydrographs possibly influenced by additional factors were examined, we can conclude that the forecasting approach in general works quite well, and we reach, for example, median NSE values of ≥ 0.5 for NARX and CNNs, while LSTMs show a median value only slightly lower. In terms of robustness against the initialization dependency of all models (ensemble variability), we clearly observe the highest dependency for NARX, followed by CNN and LSTM, while LSTMs on average perform slightly more robust than CNNs. Including GWL_{t-1} lowers the error variance of the ensemble members, which we used to judge robustness in this case, by several orders of magnitude for all models. NARX and LSTMs on average now show slightly lower ensemble variability than CNNs; however, all models are quite close. A corresponding figure was added (Fig. S69). Furthermore, we also added to the Supplement information on the results of the hyperparameter optimization (Tables S2–S4), a table with all error measure values of each considered hydrograph and model (Table S5), and (according to seq2val) forecasting plots (Figs. S1 to S34 in the Supplement).

Figure 5 shows exemplarily the forecasting performance of all three models for well BW_104-114-5, where all three models consistently achieved good results in terms of accuracy. The NARX model (a) outperforms both LSTM (b) and CNN (c) models and shows very high NSE and R^2 values between 0.8 and 0.9. The CNN model provides the second best forecast, which even very slightly shows less underestimation (Bias/rBias) of the GWLs than the NARX model. By comparing the graphs in (a) and (c), we assume that this is only true on average. The CNN model overestimates in 2012 and constantly underestimates the last third of the test period. The NARX model, however, is more consistent and therefore better. Concerning R^2 values, the LSTM basically keeps up with the CNN, and all other error measures show the still good (but in comparison worst) values. We notice that in accordance to our overall findings mentioned above, the LSTM shows the lowest ensemble variability and therefore the smallest initialization dependency. Taking a look at the selected inputs and hyperparameters, we notice that relative humidity (rH) does not seem to provide important information and was therefore never selected as an input. Further, the input sequence length of both LSTM and CNN is equally 35 steps (weeks). In the NARX model there is no direct correspondence, but a similar value is shown by the parameter FD and thus the number of past predicted GWL values available via the feedback connection.

In contrast to the above-mentioned well, hardly any systematic pattern can be derived from the choice of input parameters across all wells that even might have physical implications for each site. Rather, it is noticeable that certain model types seem to prefer also certain inputs. For example, temperature is only selected as input in 5 out of 17 cases for LSTM models and in 2 out of 17 cases for CNN models. Furthermore, rH is always selected for LSTM models except for two times. In the case of NARX models, there seems to be a lack of systematic behaviour. For more details please see Tables S2–S4.

Our approach assumes a groundwater dynamic mainly dominated by meteorological factors. We can assume that all three model types are basically capable of modelling groundwater levels very accurately if all relevant input data can be identified. To exemplarily show the influence of additional input variables, which, however, are usually not available as input for a forecast or even have insufficient historical data, Fig. 6 illustrates the significantly improved performance after including the Rhine water level (W), which is a large streamflow within the study area, using the example of the NARX model for well BW_710-256-3, which indeed is located close to the river. Besides improved performance, we also observe lower variability of the ensemble member results and thus lower dependency to the model initialization, which corresponds also to other time series, where we often find a smaller influence, the more relevant the input data are. This also confirms that low accuracy is probably due to insufficient input data on a case-by-case basis and not necessarily because of an inadequate modelling approach. Similarly, this applies also to other wells in our dataset that show unsatisfying forecasting performance. Examples of this are wells in the northern part of the study area (e.g. most wells starting with HE_...), for which our approach is generally more challenging due to strong groundwater extraction activities in this area, as well as well BW_138-019-0, which is close to the Rhine and probably under the influence of a large ship lock nearby. Additionally, this well is within a flood retention area that is spatially coupled to the ship lock.

4.2 Sequence-to-sequence (seq2seq) forecasting performance

Sequence-to-sequence forecasting is especially interesting for short- and mid-term forecasts, because the input variables only have to be available until the start of the forecast. Figure 7 summarizes and compares the overall seq2seq forecasting accuracy of the three model types for all 17 wells. Figure 7a shows the performance when only meteorological inputs are used; the models in Fig. 7b are additionally provided with GWL_{t-1} as an input. Similarly to the seq2val forecasts (Fig. 4), past GWLs seem to be especially important for LSTM and CNN models, where this additional input vari-

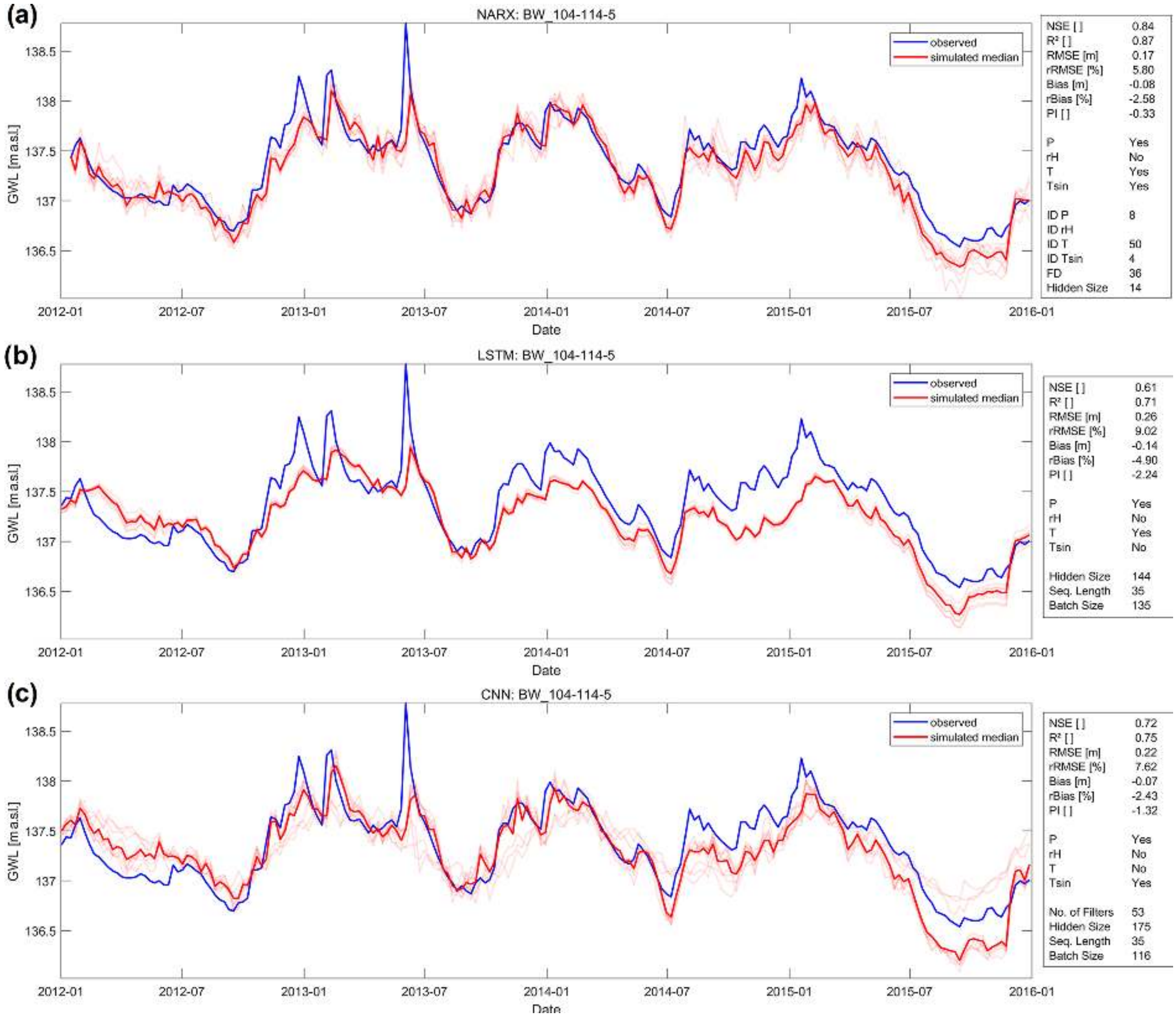

Figure 5. Forecasts of **(a)** a NARX, **(b)** a LSTM and **(c)** a CNN model for well BW_104-114-5 during the test period 2012–2016.

able causes substantial performance improvement. Without past GWLs, NARX seem to be clearly superior due to their inherent global feedback connection. However, NARX show almost equal performance values in both scenarios (Fig. 7a and b). In contrast to the seq2val forecasts, NARX systematically show lower R^2 values than LSTM and CNN models for seq2seq forecasts. For all other error measures, the accuracy of NARX models outperforms LSTMs and CNNs in a direct comparison for the vast majority of all time series. While LSTMs and CNNs show lower performance for sequence-to-sequence forecasting compared to sequence-to-value forecasting, NARX seq2seq models even outperform NARX seq2val models (except for R^2). This is quite counterintuitive as one would expect it to be more difficult to forecast a whole sequence than a single value. All in all, the scenario including past GWLs (Fig. 7b) seems to be the preferable one for all three models and shows promising results for real-world applications. Detailed results on all seq2seq models can be found Table S6 and Figs. S35 to S68.

Figure 8 summarizes exemplarily for well HE_11874 the sequence-to-sequence forecasting performance for NARX (a, b), LSTMs (c, d), CNNs (e, f), only with meteorological input variables (a, c, e), and with an additional past GWL input (b, d, f). These confirm that GWL_{t-1} substantially improves the performance of LSTMs and CNNs; however, NARX forecasts in this case only improve very slightly. Especially for LSTMs and CNNs, it is easily visible that the sequence forecasts of the better models (d,f) mostly estimate the intensity of a future groundwater level change too conservatively; thus, both increases and decreases are predicted too weak. This is a commonly known issue with ANNs, as extreme values are typically underrepresented in the distribution of the training data (e.g. Sudheer et al., 2003). We further notice that the robustness of LSTMs and CNNs in terms of initialization dependency and thus the ensemble variabil-

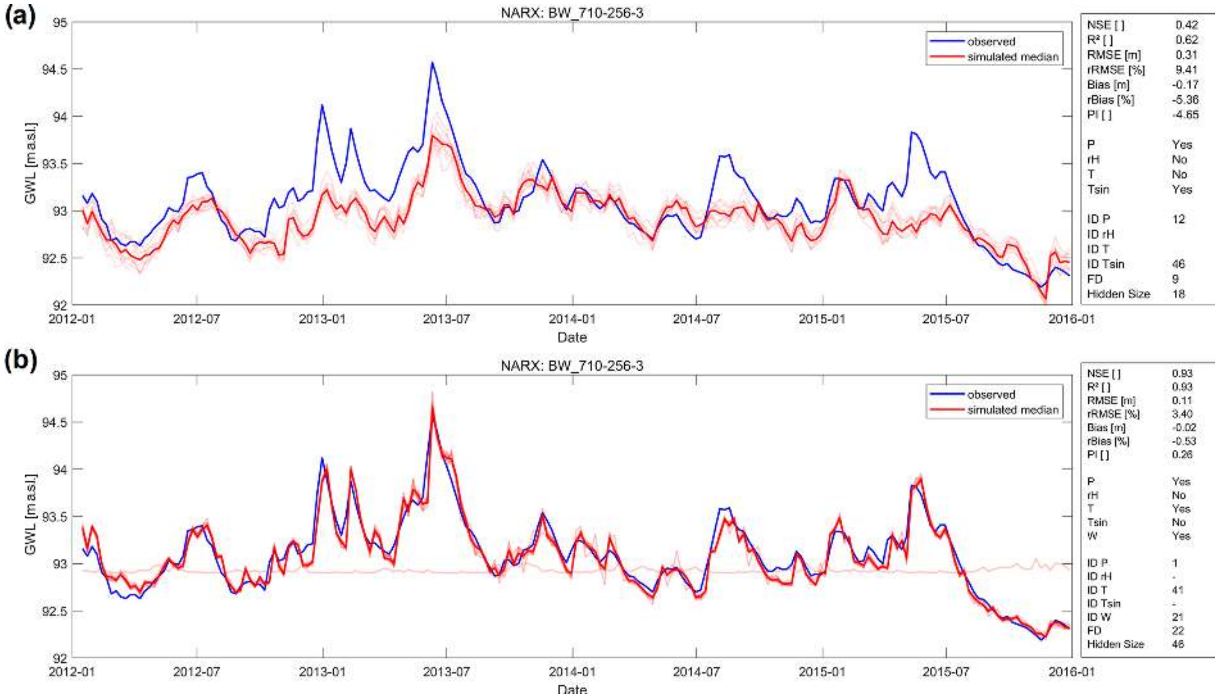

Figure 6. Forecasting performance exemplarily shown for NARX model of well BW_710-256-3 **(a)** based on meteorological input variables and **(b)** improved performance after including the Rhine water level (W) as input variable.

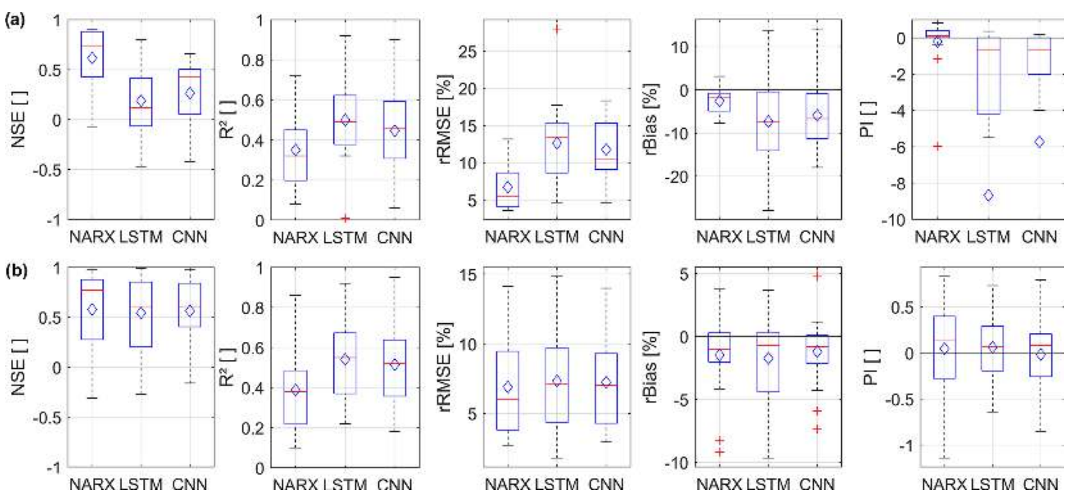

Figure 7. Boxplots showing the seq2seq forecast accuracy of NARX, LSTM, and CNN models within the test period (2012–2016) for all considered 17 hydrographs. The diamond symbols indicate the arithmetic mean; **(a)** only meteorological inputs; **(b)** GWL_{t-1} as additional input.

ity significantly improves when past GWLs are provided as inputs (Fig. 8). This is also supported by analysing the ensemble member error variances and also true for all other time series in the dataset as well (Fig. S69). Just like for seq2val forecasts, NARX usually show a significantly lower robustness in terms of initialization dependency; however, the median ensemble performance nevertheless is of high accuracy. All models, but especially NARX models, therefore should not be evaluated without including an initialization ensemble. The initialization dependency of LSTMs and CNNs is significantly lower, with LSTMs being even more robust than CNNs.

The extraordinary performance of the NARX models, especially in the case of well HE_11874 (Fig. 8) is surprising, because the performance substantially outperforms the seq2val NARX without GWL_{t-1} input (e.g. NSE: 0.35,

R^2: 0.75); however, the seq2val NARX model with GWL_{t-1} inputs also showed high accuracy (e.g. NSE: 0.99, R^2: 0.99). It is also interesting to note that the sequence predictions of the NARX models overlap exactly, and the individual sequences are therefore no longer visible. One reason for this different behaviour compared to the LSTM and CNN models is probably that the technical approach for seq2seq forecasting differs for these models. While LSTMs and CNNs use multiple output neurons to predict multiple time steps, this approach for us did not yield meaningful results for a NARX model, probably because of feedback connection issues. Instead we used one NARX output neuron to predict a multi-element vector at once.

4.3 Hyperparameter optimization and computational aspects

During the HP optimization, depending on the forecasting approach (seq2val/seq2seq) and available inputs (with or without GWL_{t-1}), there were noticeable differences with regard to the number of iterations required and the associated time needed (Fig. 9). The best parameter combination, especially for CNN and LSTM networks, was often found in 33 steps or fewer, i.e. after 25 obligatory random exploration steps in only 8 Bayesian steps. Please note that prior to the analysis we chose to at least perform 50 optimization steps, which explains the distribution in the "total iterations" column. In column two ("best iteration") we can observe similar behaviour of CNNs and LSTMs, while NARX are always somehow different to these two. We suspect that this is rather an influence of the software or the optimization algorithm, since especially model types implemented in Python show an identical behaviour. However, in the majority of cases the best iteration was found in less than 33 steps; the minimum as well as the maximum number of iteration steps were therefore obviously sufficient. It is interesting that for CNNs and LSTM the number of steps is similar throughout the experiments, whereas for NARX the inclusion of GWL_{t-1} as input caused an increase in iterations. Columns three to five in Fig. 9 show substantial differences concerning the calculation speed of the three model types. CNNs outperform all other models systematically; however, concerning the sequence-to-sequence forecasts, NARX models can almost keep up. We also observe that LSTMs seem to slow down when including GWL_{t-1} as input or when performing seq2seq forecasts, the opposite happens in the case of NARX models, which speed up in these cases. This also means that even though NARX models need more optimization iterations until the assumed optimum than LSTMs, in terms of time they outperform them due to shorter duration per iteration (column 3). Please note that it is out of the scope of this work to provide detailed assessments of the calculation speed under benchmark conditions, but we do share practical insights for fellow hydrogeologists.

4.4 Influence of training data length

In the following section we explore similarities and differences of NARX, LSTMs, and CNNs in terms of the influence of training data length. It is commonly known that data-driven approaches profit from additional data; however, how much data are necessary to build models that are able to perform reasonable calculations still remains an open question. This is because the answer is highly dependent on the application case, data properties (e.g. distribution), and model properties, as model depth can sometimes exponentially decrease the need for training data (Goodfellow et al., 2016). Therefore, this question cannot be entirely answered by a simple analysis like we perform here. Nevertheless, we still want to give an impression of how much data might be approximately needed in the case of groundwater level data in porous aquifers and if the models substantially differ in their need for training data. For our analysis, we always consider the forecasting accuracy during the 4-year testing period (2012–2016) and systematically expand the training data basis year by year, starting in 2010, thus with only clearly insufficient 2 years of training data. We focus on sequence-to-value forecasting due to the easier interpretability of the results, and we always consider the median performance of 10 different model initializations for evaluation. Figure 10 summarizes the performance and the improvement that comes with additional training data; all values are normalized per well to make them comparable. Please note that all models at least show 28 years of training data (until 1982), and only three models exceed 30 years of training data (1980); thus, the number of samples represented by the boxplots decreases significantly after 30 years. Figure 10 summarizes as well models with and without GWL_{t-1} inputs, because no significantly different behaviour was observed for each group. Please find corresponding figures for each group in Figs. S70 and S71.

As expected, we observe significant improvements with additional training data. NARX models seem to improve more or less continuously and also work better with few data, whereas for LSTMs and CNNs some kind of threshold is visible (about 10 years, thus approx. 500 samples), where the performance significantly increases and rapidly approaches the optimum. It should be noted, though, that this can probably not be transferred to other time steps; that is, in the case of daily values, 500 d will most certainly not be enough, since only one full yearly cycle is included. We explored the reason for this threshold and observed that when stopping the training 5 years earlier (2007), the threshold now occurs correspondingly 5 years earlier (Fig. S72). Additionally, we found that several standard statistic values such as mean; median; variance; overall maximum; and the percentiles at 25, 75, and 97.5 show similar thresholds (Fig. S73). Thus, the early years of the 2000s seem to be especially relevant for our test period. This is a highly dataset-specific observation that cannot be generalized; however, this also shows that it is vital

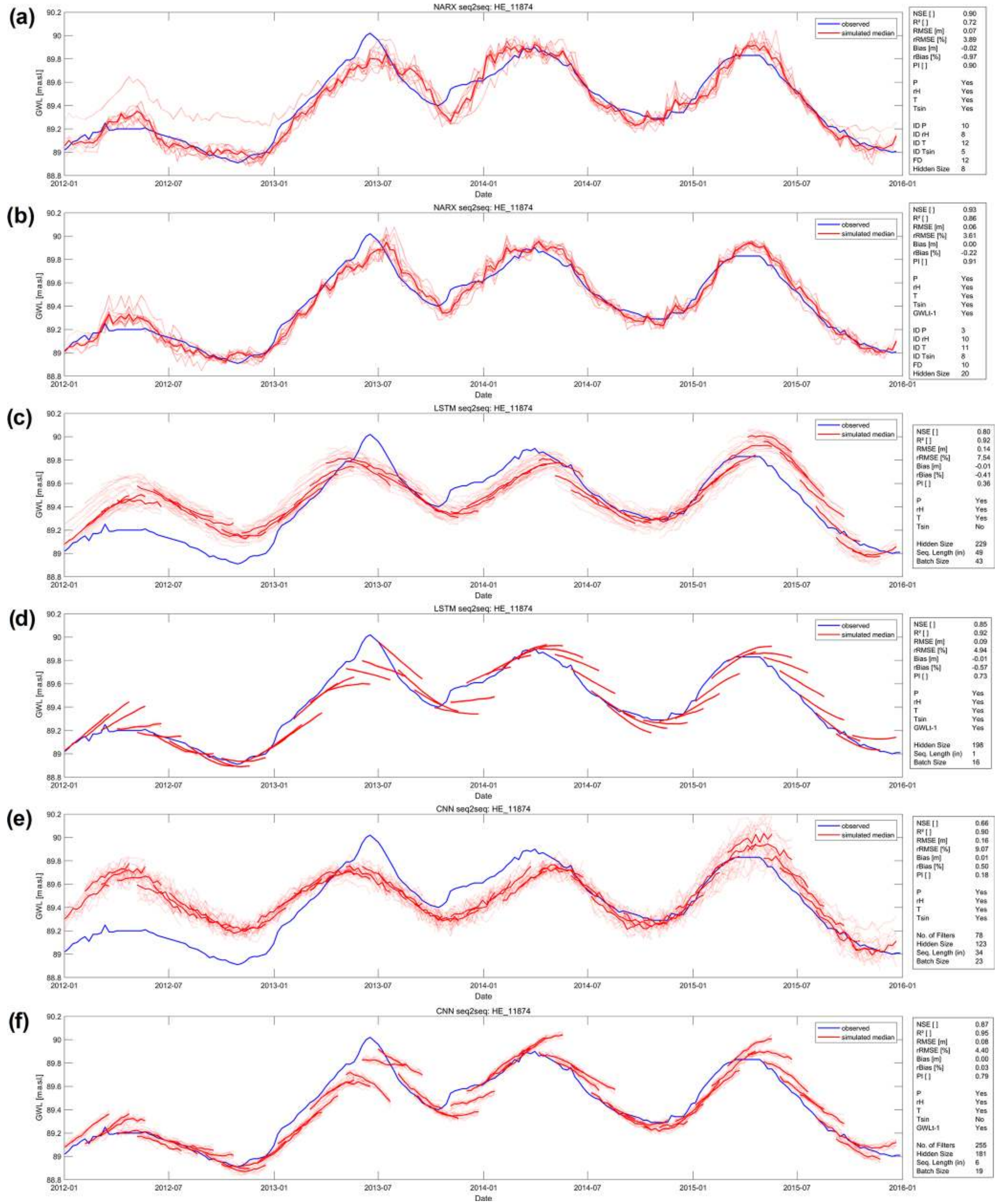

Figure 8. Forecasts of **(a, b)** a NARX, **(c, d)** a LSTM, and **(e, f)** a CNN model for well BW_104-114-5 during the test period 2012–2016. Models in **(a, c, e)** use only meteorological input variables, and models in **(b, d, f)** use also past GWL observations.

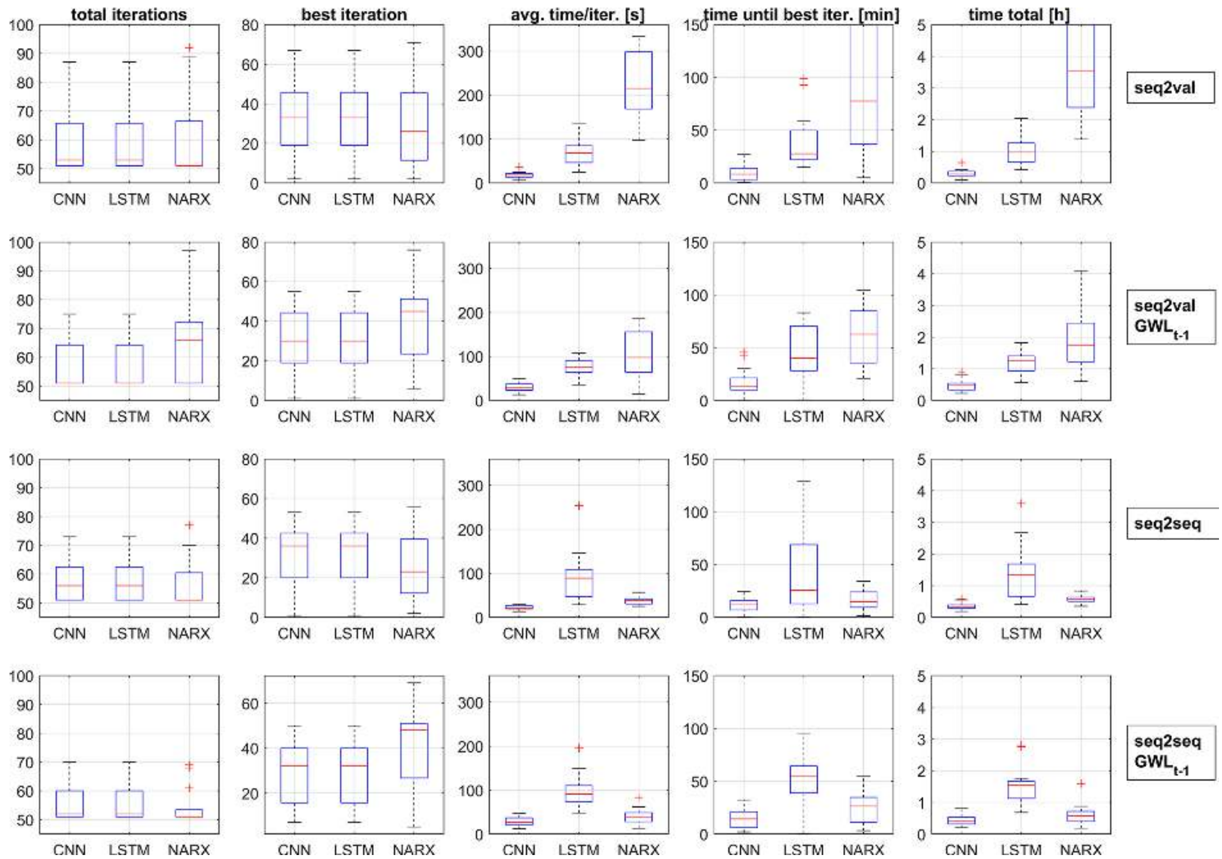

Figure 9. Comparison of the performed HP optimizations (columns 1 and 2); their calculation time per iteration in seconds (column 3), until the optimum was found (minutes) (column 4); and the total time spent on optimization in hours (column 5).

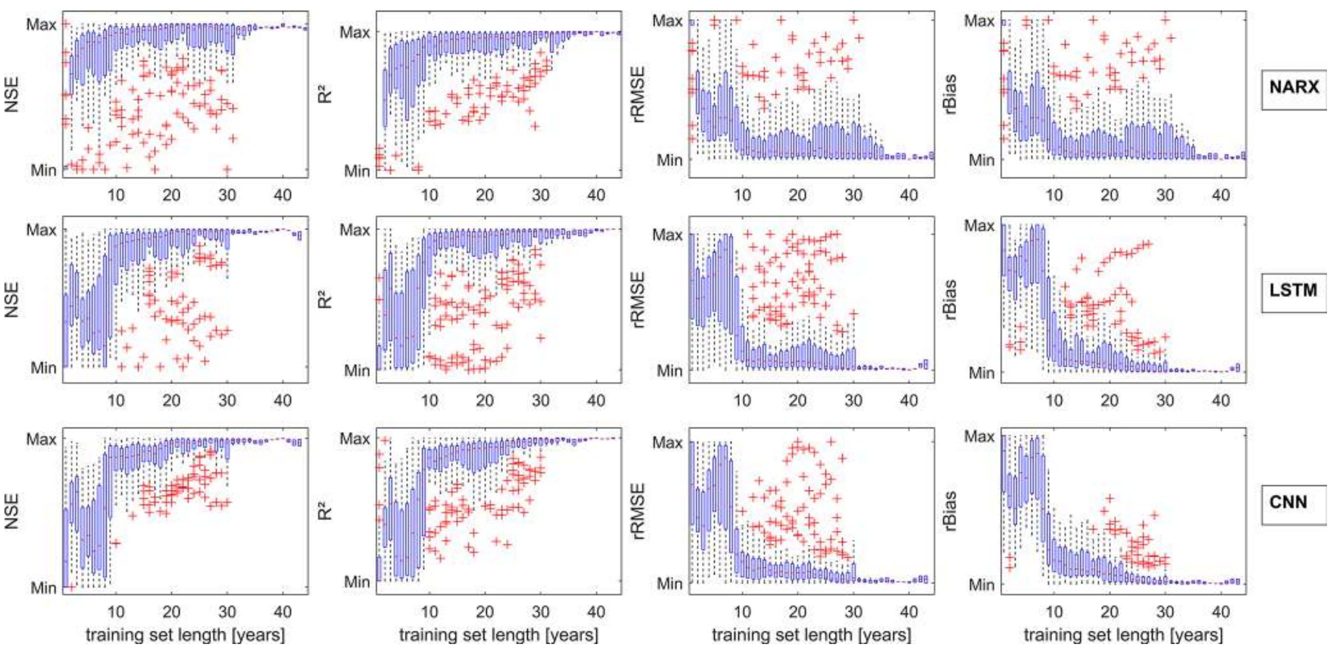

Figure 10. Influence of training data length on model performance.

to include relevant training data, which is, however, not very easy to identify. Nevertheless as a rule of thumb the chance of using the right data increases with the amount of available data. These findings are supported by the observation that not every additional year improves the accuracy; only the overall trend is positive. This seems plausible, because especially when conditions change over time, the models can also learn behaviour that is no longer valid and which possibly decreases future forecast performance. One should therefore not only include as much data as possible but also carefully evaluate and also possibly shorten the training database if necessary.

5 Conclusions

In this study we evaluate and compare the groundwater level forecasting accuracy of NARX, CNN and LSTM models. We examine sequence-to-value and sequence-to-sequence forecasting scenarios. We can conclude that in the case of seq2val forecasts all models are able to produce satisfying results, and NARX models on average perform best, while LSTMs perform the worst. Since CNNs are much faster in calculation speed than NARX and only slightly behind in terms of accuracy, they might be the favourable option if time is an issue. If accuracy is especially important, one should stick with NARX models. LSTMs, however, are most robust against initialization effects, especially compared to NARX. Including past groundwater levels as inputs strongly improves CNN and LSTM seq2val forecast accuracy. However, all three models mostly cannot beat the naïve model in this scenario and are therefore of no value.

Especially when no input data are available in short- and mid-term forecasting applications, sequence-to-sequence forecasting is of special interest. Again, past groundwater levels as input significantly improved CNN and LSTM performance, while NARX performed almost similar in both scenarios. Overall, NARX models show the best performance (except R^2 values) in the vast majority of all cases. In addition to the fast calculation of NARX in this case, which almost keeps up with CNN speed, they are clearly preferable. However, NARX models are least robust against initialization effects, which nevertheless are easy to handle by implementing a forecasting ensemble.

We further analysed what data might be needed or sufficient to reach acceptable results. As expected, we found that in principle the longer the training data, the better; however, a noteworthy threshold seems to exist for about 10 years of weekly training data, below which the performance becomes significantly worse. This applies especially for LSTM and CNN models but was also found to probably be highly dataset specific. Overall, NARX seem to perform better in comparison to CNN and LSTM models, when only few training data are available.

The results are surprising in a way that LSTMs are widely known to perform especially well on sequential data and are therefore also more commonly applied. In this work they were outperformed by CNNs and NARX models. We showed that for this specific application (i) CNNs might be the better choice due to significantly faster calculation and mostly similar performance, and (ii) even though DL approaches are currently often preferred over traditional (shallow) neural networks such as NARX, the latter should not be neglected in the selection processes especially when there is few training data available. Particularly NARX sequence-to-sequence forecasting seems to be promising for short- and mid-term forecasts. However, we do not want to ignore the fact that LSTMs and CNNs might perform substantially better with a larger dataset, which better fulfils common definitions of DL applications and where deeper networks can demonstrate their strengths, such as automated feature extraction. Since such data are usually not available in groundwater level prediction tasks yet, for the moment this remains in theory.

Code and data availability. All groundwater data are available for free via the web services of the local authorities (HLNUG, 2019; LUBW, 2018; MUEEF, 2018). Meteorological input data was derived from the HYRAS dataset (Frick et al., 2014; Rauthe et al., 2013), which can be obtained free of charge for non-commercial purposes on request from the German Meteorological Service (DWD, 2021). Our Python and MATLAB code files are available on GitHub (Wunsch, 2020).

Supplement. The supplement related to this article is available online at: https://doi.org/10.5194/hess-25-1671-2021-supplement.

Author contributions. AW conceptualized the study, wrote the code, validated and visualized the results, and wrote the original paper draft. All three authors contributed to the methodology and performed review and editing tasks. TL contributed to the conceptualization and validation as well and further supervised the work.

Competing interests. The authors declare that they have no conflict of interest.

Financial support. The article processing charges for this open-access publication were covered by a Research Centre of the Helmholtz Association.

Review statement. This paper was edited by Mauro Giudici and reviewed by Daniel Klotz and one anonymous referee.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mane, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viegas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X.: TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems, p. 19, available at: https://www.tensorflow.org/ (last access: 30 March 2021), 2015.

Adamowski, J. and Chan, H. F.: A Wavelet Neural Network Conjunction Model for Groundwater Level Forecasting, J. Hydrol., 407, 28–40, https://doi.org/10.1016/j.jhydrol.2011.06.013, 2011.

Afzaal, H., Farooque, A. A., Abbas, F., Acharya, B., and Esau, T.: Groundwater Estimation from Major Physical Hydrology Components Using Artificial Neural Networks and Deep Learning, Water, 12, 5, https://doi.org/10.3390/w12010005, 2020.

Alsumaiei, A. A.: A Nonlinear Autoregressive Modeling Approach for Forecasting Groundwater Level Fluctuation in Urban Aquifers, Water, 12, 820, https://doi.org/10.3390/w12030820, 2020.

Beale, H. M., Hagan, M. T., and Demuth, H. B.: Neural Network Toolbox™ User's Guide: Revised for Version 9.1 (Release 2016b), The MathWorks, Inc., available at: https://de.mathworks.com/help/releases/R2016b/nnet/index.html (last access: 30 March 2021), 2016.

Bengio, Y., Simard, P., and Frasconi, P.: Learning Long-Term Dependencies with Gradient Descent Is Difficult, IEEE Trans. Neural Netw., 5, 157–166, https://doi.org/10.1109/72.279181, 1994.

Bowes, B. D., Sadler, J. M., Morsy, M. M., Behl, M., and Goodall, J. L.: Forecasting Groundwater Table in a Flood Prone Coastal City with Long Short-Term Memory and Recurrent Neural Networks, Water, 11, 1098, https://doi.org/10.3390/w11051098, 2019.

Chang, F.-J., Chang, L.-C., Huang, C.-W., and Kao, I.-F.: Prediction of Monthly Regional Groundwater Levels through Hybrid Soft-Computing Techniques, J. Hydrol., 541, 965–976, https://doi.org/10.1016/j.jhydrol.2016.08.006, 2016.

Chen, Y., Kang, Y., Chen, Y., and Wang, Z.: Probabilistic Forecasting with Temporal Convolutional Neural Network, Neurocomputing, 399, 491–501, https://doi.org/10.1016/j.neucom.2020.03.011, 2020.

Chollet, F.: Keras, Keras, GitHub, available at: https://github.com/fchollet/keras (last access: 30 March 2021), 2015.

Di Nunno, F. and Granata, F.: Groundwater Level Prediction in Apulia Region (Southern Italy) Using NARX Neural Network, Environ. Res., 190, 110062, https://doi.org/10.1016/j.envres.2020.110062, 2020.

Duan, S., Ullrich, P., and Shu, L.: Using Convolutional Neural Networks for Streamflow Projection in California, Front. Water, 2, 28, https://doi.org/10.3389/frwa.2020.00028, 2020.

DWD: HYRAS – Hydrologische Rasterdatensätze, available at: https://www.dwd.de/DE/leistungen/hyras/hyras.html, last access: 30 March 2021.

Fang, K., Pan, M., and Shen, C.: The Value of SMAP for Long-Term Soil Moisture Estimation With the Help of Deep Learning, IEEE Trans. Geosci. Remote, 57, 2221–2233, https://doi.org/10.1109/TGRS.2018.2872131, 2019.

Fang, K., Kifer, D., Lawson, K., and Shen, C.: Evaluating the Potential and Challenges of an Uncertainty Quantification Method for Long Short-Term Memory Models for Soil Moisture Predictions, Water Resour. Res., 56, e2020WR028095, https://doi.org/10.1029/2020WR028095, 2020.

FAO: The Wealth of Waste: The Economics of Wastewater Use in Agriculture, no. 35 in FAO Water Reports, Food and Agriculture Organization of the United Nations, Rome, 2010.

Frick, C., Steiner, H., Mazurkiewicz, A., Riediger, U., Rauthe, M., Reich, T., and Gratzki, A.: Central European High-Resolution Gridded Daily Data Sets (HYRAS): Mean Temperature and Relative Humidity, Meteorol. Z., 23, 15–32, https://doi.org/10.1127/0941-2948/2014/0560, 2014.

Gauch, M., Kratzert, F., Klotz, D., Nearing, G., Lin, J., and Hochreiter, S.: Rainfall–Runoff Prediction at Multiple Timescales with a Single Long Short-Term Memory Network, Hydrol. Earth Syst. Sci. Discuss. [preprint], https://doi.org/10.5194/hess-2020-540, in review, 2020.

Gauch, M., Mai, J., and Lin, J.: The Proper Care and Feeding of CAMELS: How Limited Training Data Affects Streamflow Prediction, Environ. Model. Softw., 135, 104926, https://doi.org/10.1016/j.envsoft.2020.104926, 2021.

Gers, F. A., Schmidhuber, J., and Cummins, F.: Learning to Forget: Continual Prediction with LSTM, Neural Comput., 12, 2451–2471, https://doi.org/10.1162/089976600300015015, 2000.

Goodfellow, I., Bengio, Y., and Courville, A.: Deep Learning, Adaptive Computation and Machine Learning, The MIT Press, Cambridge, Massachusetts, 2016.

Guzman, S. M., Paz, J. O., and Tagert, M. L. M.: The Use of NARX Neural Networks to Forecast Daily Groundwater Levels, Water Resour. Manage., 31, 1591–1603, https://doi.org/10.1007/s11269-017-1598-5, 2017.

Guzman, S. M., Paz, J. O., Tagert, M. L. M., and Mercer, A. E.: Evaluation of Seasonally Classified Inputs for the Prediction of Daily Groundwater Levels: NARX Networks Vs Support Vector Machines, Environ. Model. Assess., 24, 223–234, https://doi.org/10.1007/s10666-018-9639-x, 2019.

Hasda, R., Rahaman, M. F., Jahan, C. S., Molla, K. I., and Mazumder, Q. H.: Climatic Data Analysis for Groundwater Level Simulation in Drought Prone Barind Tract, Bangladesh: Modelling Approach Using Artificial Neural Network, Groundwater Sustain. Dev., 10, 100361, https://doi.org/10.1016/j.gsd.2020.100361, 2020.

HLNUG: GruSchu, available at: http://gruschu.hessen.de (last access: 30 March 2021), 2019.

Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, Neural Comput., 9, 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735, 1997.

Hunter, J. D.: Matplotlib: A 2D Graphics Environment, Comput. Sci. Eng., 9, 90–95, https://doi.org/10.1109/MCSE.2007.55, 2007.

Izady, A., Davary, K., Alizadeh, A., Moghaddamnia, A., Ziaei, A. N., and Hasheminia, S. M.: Application of NN-ARX Model to Predict Groundwater Levels in the Neishaboor Plain, Iran, Water Resour. Manage., 27, 4773–4794, https://doi.org/10.1007/s11269-013-0432-y, 2013.

Jakeman, A. J. and Hornberger, G. M.: How Much Complexity Is Warranted in a Rainfall-Runoff Model?, Water Resour. Res., 29, 2637–2649, https://doi.org/10.1029/93WR00877, 1993.

Jeihouni, E., Mohammadi, M., Eslamian, S., and Zareian, M. J.: Potential Impacts of Climate Change on Groundwater Level through Hybrid Soft-Computing Methods: A Case Study–Shabestar Plain, Iran, Environ. Monit. Assess., 191, 620, https://doi.org/10.1007/s10661-019-7784-6, 2019.

Jeong, J. and Park, E.: Comparative Applications of Data-Driven Models Representing Water Table Fluctuations, J. Hydrol., 572, 261–273, https://doi.org/10.1016/j.jhydrol.2019.02.051, 2019.

Jeong, J., Park, E., Chen, H., Kim, K.-Y., Shik Han, W., and Suk, H.: Estimation of Groundwater Level Based on the Robust Training of Recurrent Neural Networks Using Corrupted Data, J. Hydrol., 582, 124512, https://doi.org/10.1016/j.jhydrol.2019.124512, 2020.

Klotz, D., Kratzert, F., Gauch, M., Sampson, A., Klambauer, G., Hochreiter, S., and Nearing, G.: Uncertainty Estimation with Deep Learning for Rainfall-Runoff Modelling, arXiv: preprint, https://doi.org/10.31223/X5JS4T, 2020.

Kong-A-Siou, L., Fleury, P., Johannet, A., Borrell Estupina, V., Pistre, S., and Dörfliger, N.: Performance and Complementarity of Two Systemic Models (Reservoir and Neural Networks) Used to Simulate Spring Discharge and Piezometry for a Karst Aquifer, J. Hydrol., 519, 3178–3192, https://doi.org/10.1016/j.jhydrol.2014.10.041, 2014.

Kraft, B., Jung, M., Körner, M., and Reichstein, M.: Hybrid Modeling: Fusion of a Deep Learning Approach and a Physics-Based Model for Global Hydrological Modeling, Int. Arch. Photogramm. Remote Sens. Spatial Inf. Sci., XLIII-B2-2020, 1537–1544, https://doi.org/10.5194/isprs-archives-XLIII-B2-2020-1537-2020, 2020.

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.

Kratzert, F., Herrnegger, M., Klotz, D., Hochreiter, S., and Klambauer, G.: NeuralHydrology – Interpreting LSTMs in Hydrology, arXiv: preprint, arXiv:1903.07903 [physics, stat], 2019a.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G.: Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets, Hydrol. Earth Syst. Sci., 23, 5089–5110, https://doi.org/10.5194/hess-23-5089-2019, 2019b.

Lähivaara, T., Malehmir, A., Pasanen, A., Kärkkäinen, L., Huttunen, J. M. J., and Hesthaven, J. S.: Estimation of Groundwater Storage from Seismic Data Using Deep Learning, Geophys. Prospect., 67, 2115–2126, https://doi.org/10.1111/1365-2478.12831, 2019.

LeCun, Y., Bengio, Y., and Hinton, G.: Deep Learning, Nature, 521, 436–444, https://doi.org/10.1038/nature14539, 2015.

Lin, T., Horne, B. G., Tiño, P., and Giles, C. L.: Learning Long-Term Dependencies in NARX Recurrent Neural Networks, IEEE T. Neural Netw., 7, 1329–1338, 1996a.

Lin, T., Horne, B. G., Tiño, P., and Giles, C. L.: Learning Long-Term Dependencies Is Not as Difficult with NARX Networks, in: Advances in Neural Information Processing Systems, MIT Press, Denver, Colorado, 577–583, 1996b.

Lin, T., Horne, B. G., and Giles, C. L.: How Embedded Memory in Recurrent Neural Network Architectures Helps Learning Long-Term Temporal Dependencies, Neural Networks, 11, 861–868, https://doi.org/10.1016/S0893-6080(98)00018-5, 1998.

LUBW: Hydrogeologischer Bau Und Hydraulische Eigenschaften – 9INTERREG III A-Projekt MoNit "Modellierung Der Grundwasserbelastung Durch Nitrat Im Oberrheingraben"/Structure Hydrogéologique et Caractéristiques Hydrauliques – 9INTERREG III A: MoNit "Modélisation de La Pollution Des Eaux Souterraines Par Les Nitrates Dans La Vallée Du Rhin Supérieur", Tech. rep., LUBW, available at: https://www4.lubw.baden-wuerttemberg.de/servlet/is/18633/monit_hydrogeologischer_bau.pdf?command=downloadContent&filename=monit_hydrogeologischer_bau.pdf (last access: 11 June 2019), 2006.

LUBW: UDO – Umwelt-Daten und – Karten Online, available at: https://udo.lubw.baden-wuerttemberg.de/public/ (last access: 31 March 2021), 2018.

Maier, H. R., Jain, A., Dandy, G. C., and Sudheer, K.: Methods Used for the Development of Neural Networks for the Prediction of Water Resource Variables in River Systems: Current Status and Future Directions, Environ. Model. Softw., 25, 891–909, https://doi.org/10.1016/j.envsoft.2010.02.003, 2010.

Mathworks Inc.: Matlab 2020a, The MathWorks Inc., 2020.

McKinney, W.: Data Structures for Statistical Computing in Python, in: Python in Science Conference, Austin, Texas, 56–61, https://doi.org/10.25080/Majora-92bf1922-00a, 2010.

Moghaddamnia, A., Remesan, R., Kashani, M. H., Mohammadi, M., Han, D., and Piri, J.: Comparison of LLR, MLP, Elman, NNARX and ANFIS Models – with a Case Study in Solar Radiation Estimation, J. Atmos. Sol.-Terr. Phy., 71, 975–982, https://doi.org/10.1016/j.jastp.2009.04.009, 2009.

MUEEF: Geoportal Wasser, available at: http://geoportal-wasser.rlp.de/servlet/is/8183/ (last access: 31 March 2021), 2018.

Müller, J., Park, J., Sahu, R., Varadharajan, C., Arora, B., Faybishenko, B., and Agarwal, D.: Surrogate Optimization of Deep Neural Networks for Groundwater Predictions, J. Glob. Optim., https://doi.org/0.1007/s10898-020-00912-0, in press, 2020.

Nogueira, F.: Bayesian Optimization: Open Source Constrained Global Optimization Tool for Python, GitHub, available at: https://github.com/fmfn/BayesianOptimization (last access: 15 April 2020), 2014.

Pan, M., Zhou, H., Cao, J., Liu, Y., Hao, J., Li, S., and Chen, C.-H.: Water Level Prediction Model Based on GRU and CNN, IEEE Access, 8, 60090–60100, https://doi.org/10.1109/ACCESS.2020.2982433, 2020.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., and Cournapeau, D.: Scikit-Learn: Machine Learning in Python, J. Mach. Learn. Res., 12, 2825–2830, 2011.

Rahmani, F., Lawson, K., Ouyang, W., Appling, A., Oliver, S., and Shen, C.: Exploring the Exceptional Performance of a Deep Learning Stream Temperature Model and the Value of Streamflow Data, Environ. Res. Lett., 16, 024025, https://doi.org/10.1088/1748-9326/abd501, 2021.

Rajaee, T., Ebrahimi, H., and Nourani, V.: A review of the artificial intelligence methods in groundwater level modeling, 572, 336–351, https://doi.org/10.1016/j.jhydrol.2018.12.037, 2019.

Rauthe, M., Steiner, H., Riediger, U., Mazurkiewicz, A., and Gratzki, A.: A Central European Precipitation Climatology –

Part I: Generation and Validation of a High-Resolution Gridded Daily Data Set (HYRAS), Meteorol. Z., 22, 235–256, https://doi.org/10.1127/0941-2948/2013/0436, 2013.

Reback, J., McKinney, W., Jbrockmendel, Bossche, J. V. D., Augspurger, T., Cloud, P., Gfyoung, Sinhrks, Klein, A., Roeschke, M., Hawkins, S., Tratner, J., She, C., Ayd, W., Petersen, T., Garcia, M., Schendel, J., Hayden, A., MomIsBestFriend, Jancauskas, V., Battiston, P., Seabold, S., Chris-B1, H-Vetinari, Hoyer, S., Overmeire, W., Alimcmaster, Dong, K., Whelan, C., and Mehyar, M.: Pandas-Dev/Pandas: Pandas 1.0.3, Zenodo, https://doi.org/10.5281/ZENODO.3509134, 2020.

Région Alsace – Strasbourg: Bestandsaufnahme Der Grundwasserqualität Im Oberrheingraben/Inventaire de La Qualité Des Eaux Souterraines Dans La Vallée Du Rhin Supérieur, available at: https://www.ermes-rhin.eu/uploads/pdf/acces-libres/Resultats-INV1997.pdf (last access: 31 March 2021), 1999.

Shen, C.: A Transdisciplinary Review of Deep Learning Research and Its Relevance for Water Resources Scientists, Water Resour. Res., 54, 8558–8593, https://doi.org/10.1029/2018WR022643, 2018.

Sudheer, K. P., Nayak, P. C., and Ramasastri, K. S.: Improving Peak Flow Estimates in Artificial Neural Network River Flow Models, Hydrol. Process., 17, 677–686, https://doi.org/10.1002/hyp.5103, 2003.

Supreetha, B. S., Shenoy, N., and Nayak, P.: Lion Algorithm-Optimized Long Short-Term Memory Network for Groundwater Level Forecasting in Udupi District, India, Appl. Comput. Intel. Soft Comput.g, 2020, 1–8, https://doi.org/10.1155/2020/8685724, 2020.

UNESCO: World's Groundwater Resources Are Suffering from Poor Governance, available at: http://www.unesco.org/new/en/media-services/single-view/news/worlds_groundwater_resources_are_suffering_from_poor_gove/ (last access: 31 March 2021), 2012.

van der Walt, S., Colbert, S. C., and Varoquaux, G.: The NumPy Array: A Structure for Efficient Numerical Computation, Comput. Sci. Eng., 13, 22–30, https://doi.org/10.1109/MCSE.2011.37, 2011.

van Rossum, G.: Python Tutorial, Centrum voor Wiskunde en Informatica Amsterdam, Amsterdam, 1995.

Wunsch, A.: AndreasWunsch/Groundwater-Level-Forecasting-with-ANNs-A-Comparison-of-LSTM-CNN-and-NARX, GitHub repository, Zenodo, https://doi.org/10.5281/zenodo.4018722, 2020.

Wunsch, A., Liesch, T., and Broda, S.: Forecasting Groundwater Levels Using Nonlinear Autoregressive Networks with Exogenous Input (NARX), J. Hydrol., 567, 743–758, https://doi.org/10.1016/j.jhydrol.2018.01.045, 2018.

WWAP: Water for a Sustainable World, no. 6.2015 in The United Nations World Water Development Report, United Nations World Water Assessment Programme – UNESCO, Paris, France, 2015.

Zhang, J., Zhu, Y., Zhang, X., Ye, M., and Yang, J.: Developing a Long Short-Term Memory (LSTM) Based Model for Predicting Water Table Depth in Agricultural Areas, J. Hydrol., 561, 918–929, https://doi.org/10.1016/j.jhydrol.2018.04.065, 2018.

Zhang, J., Zhang, X., Niu, J., Hu, B. X., Soltanian, M. R., Qiu, H., and Yang, L.: Prediction of Groundwater Level in Seashore Reclaimed Land Using Wavelet and Artificial Neural Network-Based Hybrid Model, J. Hydrol., 577, 123948, https://doi.org/10.1016/j.jhydrol.2019.123948, 2019.