

Water Resources Research®

RESEARCH ARTICLE

10.1029/2022WR032602

Special Section:

Modeling, simulation, and big data techniques in subsurface fluid flow and transport

Key Points:

- The Transformer architecture was applied in karst hydrology for the first time, showing high performance for discharge forecasting
- Monte Carlo dropout revealed that the prediction intervals are smallest and cover the measured discharges best in winter and autumn
- The high temporal resolution of the input data sets improved the forecasting performance

Supporting Information:

Supporting Information may be found in the online version of this article.

Correspondence to:

J. Derx,
derx@hydro.tuwien.ac.at

Citation:

Pöhl, A., Blaschke, A. P., Komma, J., Farnleitner, A. H., & Derx, J. (2024). Transformer versus LSTM: A comparison of deep learning models for karst spring discharge forecasting. *Water Resources Research*, 60, e2022WR032602. <https://doi.org/10.1029/2022WR032602>

Received 15 APR 2022

Accepted 20 MAR 2024

Transformer Versus LSTM: A Comparison of Deep Learning Models for Karst Spring Discharge Forecasting

Anna Pöhl^{1,2} , Alfred Paul Blaschke^{1,2}, Jürgen Komma¹, Andreas H. Farnleitner^{2,3,4}, and Julia Derx^{1,2} 

¹Institute of Hydraulic Engineering and Water Resources Management 222/2, TU Wien, Vienna, Austria, ²Interuniversity Cooperation Centre Water and Health, Vienna, Austria, ³Institute of Chemical, Environmental and Bioscience Engineering, Research Group Microbiology and Molecular Diagnostics 166/5/3, TU Wien, Vienna, Austria, ⁴Division Water Quality and Health, Department for Pharmacology, Physiology and Microbiology, Karl Landsteiner University for Health Sciences, Krems, Austria

Abstract Karst springs are essential drinking water resources, however, modeling them poses challenges due to complex subsurface flow processes. Deep learning models can capture complex relationships due to their ability to learn non-linear patterns. This study evaluates the performance of the Transformer in forecasting spring discharges for up to 4 days. We compare it to the Long Short-Term Memory (LSTM) Neural Network and a common baseline model on a well-studied Austrian karst spring (LKAS2) with an extensive hourly database. We evaluated the models for two further karst springs with diverse discharge characteristics for comparing the performances based on four metrics. In the discharge-based scenario, the Transformer performed significantly better than the LSTM for the spring with the longest response times (9% mean difference across metrics), while it performed poorer for the spring with the shortest response time (4% difference). Moreover, the Transformer better predicted the shape of the discharge during snowmelt. Both models performed well across all lead times and springs with 0.64–0.92 for the Nash–Sutcliffe efficiency and 10.8%–28.7% for the symmetric mean absolute percentage error for the LKAS2 spring. The temporal information, rainfall and electrical conductivity were the controlling input variables for the non-discharge based scenario. The uncertainty analysis revealed that the prediction intervals are smallest in winter and autumn and highest during snowmelt. Our results thus suggest that the Transformer is a promising model to support the drinking water abstraction management, and can have advantages due to its attention mechanism particularly for longer response times.

1. Introduction

Karst aquifers provide 10% of the population with drinking water (Olarinoye et al., 2020). Due to its distinct hydrogeological characteristics, including sinking streams, caves, enclosed depressions, and networks of conduits and fractures, it is difficult to accurately predict the discharges of karst springs (Ford & Williams, 2007). Advances in this field of research are needed for supporting sustainable drinking water supply management.

Different karst discharge modeling approaches exist such as hydrogeological, pipe flow, and data-driven models (Jeannin et al., 2021). Hydrogeological (fully distributed) models are based on the solution of process-based governing equations, for example, laminar, turbulent, unsaturated, and saturated flow. Pipe flow models are based on a conceptualization of different geological units of the aquifer such as the epikarst, conduits, and the permeable matrix (Çallı et al., 2022; Mazzilli et al., 2017). Data-driven model approaches include machine learning models, which are statistical models that learn certain tasks through training with data (Rahbar et al., 2022).

Within the group of machine learning models, deep learning models have been shown to work particularly well for groundwater related problems, outperforming conceptual and physically based model approaches, as demonstrated for example, by Husic et al. (2022). The state-of-the-art deep learning approaches applied in hydrology so far include the multi-layer perceptron (MLP), convolutional neural networks (CNN), and recurrent neural networks (RNN), such as LSTMs (Rahbar et al., 2022; Wunsch et al., 2021, 2022). LSTMs are neural networks that were developed to overcome shortcomings of previous RNNs, which have difficulties learning long-term dependencies. LSTMs use a cell state that is updated through gates, which control information flow (Hochreiter & Schmidhuber, 1997). The Transformer is a deep learning model for predicting sequential data that was introduced by Vaswani et al. (2017). It circumvents problems associated with the sequential nature of RNNs by using the so-called “attention mechanism.” Attention allows for building relations between every pair of input feature values in

© 2024. The Authors.

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

the input sequence and not only to the temporally close data points. Transformers have been built for and are very successful in natural language processing (Brown et al., 2020; Devlin et al., 2019) and are now being adapted for a wide range of problems. For example, in speech recognition, a Transformer was shown to outperform state-of-the-art LSTMs (Zeyer et al., 2019). In recent years, efforts have been made to implement and improve Transformers for time series forecasting (Li et al., 2019; Mohammdi Farsani & Pazouki, 2021). In the field of hydrology, the Transformer is widely unknown, and, to the best of our knowledge, has not been explored in karst hydrology yet. We hypothesize that it might work particularly well and potentially outperform other deep learning models in forecasting non-linear hydrological processes.

Machine learning and especially deep learning models have been applied in hydrology for modeling surface water and groundwater (C. Chen et al., 2020; Fang et al., 2021; Xiang et al., 2020). The primary focus so far has been on rainfall–runoff modeling and flood forecasting (Kratzert et al., 2018; Lioung & Chandrasekaran, 2007). On top of that, groundwater level forecasting has been explored in multiple studies (Daliakopoulos et al., 2005; Nayak et al., 2006; Wunsch et al., 2021). For the prediction of karst discharges, machine learning models have only recently been used (An et al., 2020; Cheng et al., 2021; Jeannin et al., 2021; Rahbar et al., 2022; Xu et al., 2022). For instance, Rahbar et al. (2022) used several shallow and deep learning approaches, for example, an MLP combined with a hybrid gamma test-genetic algorithm approach, achieving high performances in the discharge forecasts for some of the investigated karst springs. Generally, deep learning approaches (e.g., MLP and Long Short-Term Memory (LSTM)) showed high performances for the prediction of karst spring discharges (Cheng et al., 2021; Rahbar et al., 2022). Most of the machine learning applications for karst discharge predictions, however, were based on only a few feature variables at low temporal resolutions (Cheng et al., 2021; Rahbar et al., 2022). Incorporating online measurements of meteorological and physicochemical water quality variables at high temporal resolution offers opportunities for improving the karst spring discharge forecasts. The aim of this paper is to test the performance and evaluate the model uncertainty of the Transformer and compare it to an LSTM for forecasting karst discharges. The primary focus is on the well studied limestone karst aquifer spring LKAS2 in Austria (Farnleitner et al., 2005; Reischer et al., 2008; Stadler et al., 2010), providing an excellent data basis including several variables at high temporal resolution. For cross-comparison of the model performances for different discharge characteristics, we include two further springs, the higher dynamic Aubach spring and the lower dynamic Ursprung spring, for both of which fewer variables were available. We use an hourly temporal frequency of the input and output variables and a forecast horizon of one to 4 days ahead (i.e., lead time of one to 4 days). This time horizon was chosen because it overlaps with the demands of the local water industry and preliminary experiments showed acceptable results up to a lead time of 4 days, in contrast to longer lead times. This was also shown by Lambrakis et al. (2000) who tested lead times of up to 16 days and concluded that lead times over 4 days result in high errors due to the chaotic nature of the karst system dynamics. For comparison, we selected the LSTM as one of the state-of-the-art deep learning neural network approaches applied earlier for karst spring discharges (An et al., 2020), and a common baseline model. We hypothesize that the Transformer outperforms the LSTM because of its advantages in comparison to the RNN mentioned above. The secondary aim was to explore the potential of various input variables for improving the spring discharge forecasting performance, including hourly meteorological, as well as spring discharge, and online physicochemical water quality variables. To meet our aims and test our hypothesis, we used 7-year data sets from the LKAS2, the Aubach and the Ursprung karst springs. We performed a feature selection to obtain the ideal input variables for the respective models. For both deep learning models, we obtained an optimized model architecture by automatic hyperparameter tuning with regards to the number of layers, units, and regularization measures. Finally, we compared the mean model performances of all models over each day of lead time based on the Nash–Sutcliffe efficiency (NSE), the mean absolute error (MAE), the root mean square error (RMSE), and the symmetric Mean Absolute Percentage Error (sMAPE). For evaluating the model uncertainty, we used a state-of-the-art Bayesian method, called Monte Carlo (MC) dropout that aids in determining prediction intervals (Gal & Ghahramani, 2016). We used the percentage of measured discharge in the prediction intervals and the prediction interval width to quantify the uncertainty of the predictions.

2. Materials and Methods

2.1. Study Area and Data

Karst aquifers are of essential importance for the water supply of many European regions. Cities such as Vienna (AT), Rome (IT), and Grenoble (FR) rely predominantly on karst aquifers for drinking water (Z. Chen et al., 2017; Goldscheider, 2005).

Table 1*Notation and Units for the Online Near-Real Time Data Used in the Experiments*

Feature	Abbreviation	Feature class	Unit
Discharge	Q	Hydrological	m ³ /s
Precipitation station 1	rain1	Meteorological	mm/hr
Precipitation station 2	rain2	Meteorological	mm/hr
Air temperature station 1	AT1	Meteorological	°C
Air temperature station 2	AT2	Meteorological	°C
Snow height station 1	snow1	Meteorological	cm
Snow height station 2	snow2	Meteorological	cm
Turbidity	turb	Physicochem	FTU
Electrical conductivity	EC	Physicochem	S/m
Spectral absorption coefficient at 254 nm	SAC	Physicochem	1/cm
Water temperature	WT	Physicochem	°C
Encoded “day within the year”	day	Numerical	—
Encoded “month within the year”	mth	Numerical	—

LKAS2. The limestone karst aquifer spring LKAS2 is located in the Northern Calcareous Alps in Austria. LKAS2 contributes substantially to the Austrian urban water supply, and its discharge and water quality characteristics have been extensively studied previously (Farnleitner et al., 2005; Reischer et al., 2008; Reszler et al., 2018; Stadler et al., 2010). The altitudes of the catchment reach up to 2,277 m above sea level (a.s.l.). The catchment area is approximately 70 km², the mean altitude of the catchment is 1,380 m a.s.l. and the spring is accessible in a valley at 600 m a.s.l (Farnleitner et al., 2005; Reischer et al., 2008). The average discharge is 5880 l/s, with a ratio Q_{\max}/Q_{\min} of 57.5 during 2012–2019. The entire plateau in the catchment and most slopes are well karstified. The dominant subsurface karst features are dolines. Furthermore, multiple polje-like features exist. Polygenetic glaciokarstic depressions with a diameter of more than 500 m and depths of 60 m are found. More than 1,000 caves have been mapped within the catchment area and observations indicate that water rapidly passes the vadose zone. Springs, streams, and ponors are rarely found in the catchment (Plan et al., 2010). The vegetation consists of pastures, forests, and natural calcareous alpine swards with krummholz (Farnleitner et al., 2005). For forecasting karst discharge at LKAS2, we used hydrological, physicochemical, and meteorological data that is, the spring discharge, the turbidity, the spectral absorption coefficient at 254 nm (SAC), the water temperature, the electrical conductivity (EC), the air temperature, precipitation, and snow height (Table 1). The EC, water temperature, and water pressure were measured at the karst spring online and near-real time through a data collection system called GEALOG-S from Logotronic (Vienna, Austria). Discharges were calculated from the water pressures and available rating curves. Turbidity and the SAC was measured via the spectro:lyser (V2), which is a spectrometer for online water quality monitoring. The spectrometer uses UV-visible-spectroscopy for measuring the intensity of light that is transmitted through an in situ sample, along with a reference measurement of the surrounding light for calibration. Multiple variables can be measured with the spectrometer through transmitting light beams with different wavelengths. For more details on the measurement of discharges and physicochemical variables, we refer to Farnleitner et al. (2005). For more details on UV-visible-spectroscopy, we refer to van den Broeke et al. (2008). The physicochemical variables are well established for karst systems. They were found to correlate moderately with discharge at LKAS2 (correlation coefficients of -0.70 for EC, 0.67 for turbidity, and 0.66 for SAC, p -value < 0.05, Reischer et al., 2008), but also in other karst catchments (e.g., Chang et al., 2022). Meteorological variables were available from two stations in the catchment area at altitudes of 1,350 and 1,520 m a.s.l. All data was available from 2012 to 2019 at hourly timesteps. The percentage of missing values per variable was 3.6% for Q, 3.6% for WT, 3.1% for EC, 0.2% for SAC, 0.2% for turb, 0.4% for rain1, 0.7% for AT1, 0.3% for snow1, 1.3% for rain2, 0.1% for AT2, and 0% for snow2 (for notations, see Table 1).

Aubach. The Aubach spring is located at the northern edge of the Alps in Western Austria. It drains the Gottesacker plateau along with multiple other springs. The Aubach spring is located at 1,080 m a.s.l. The highest point in the catchment is at 2,033 m a.s.l. The plateau predominantly consists of bare karst. Almost the entire

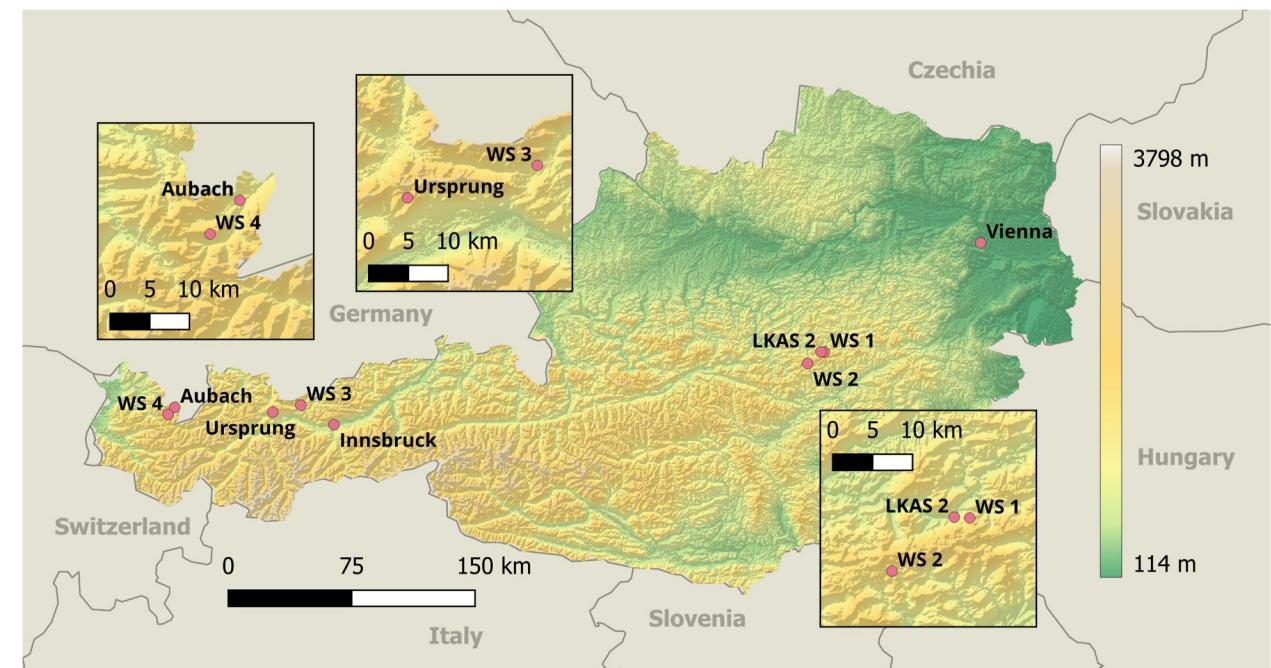


Figure 1. Topographical map of the three investigated Austrian alpine springs LKAS2, Ursprung spring and Aubach spring including the weather stations (WS 1–4).

plateau is a single karrenfield. The dominant rock type is highly karstified limestone, called Schrattenkalk with a thickness of approximately 100 m. The underlying confining layer consists of sequences of mudrock, marl, and limestone (“Drusbergschichten”). The catchment is traversed by many faults and fractures. The tectonic disruption of the Schrattenkalk is the reason for the extreme karstification. In addition to karrenfields, dolines, and ponors are frequent features. The Aubach spring has multiple openings. At higher discharges the more elevated springs become active. The spring is intermittent with completely dry periods in winter (Goldscheider, 2005). The average and maximum discharges are 876 and 10,693 l/s (2012–2019), respectively, demonstrating strong dynamics. More details about the Aubach spring and the Gottesacker karst systems can be found in Goldscheider (2005). To simulate discharges of the Aubach spring we used spring discharge, rainfall, snow height, EC, and water temperature as input. For the Aubach spring the percentage of missing values was 2.8% for Q, 0% for rainfall, 0% for snow height, 13% for EC, and 14.5% for WT (for notations see Table 1).

Ursprung. The Ursprung spring is also located in the Northern Alps in Western Austria 57 km away from the Aubach spring. It drains the Mieming mountain range in the South. This mountain range is built of Wetterstein limestone. The Ursprung spring is located at 1,590 m a.s.l. The highest point in the catchment is at 2,768 m a.s.l. The average discharge is 117 l/s, and the ratio Q_{\max}/Q_{\min} is 21, which shows that the Ursprung spring is the least dynamic one of the three studied springs. The hydrogeology of the catchment area has not been studied and thus limited information is available. The tectonics of the Mieming mountains are investigated in Ortner and Kilian (2022). For the Ursprung spring, the same variables as for the Aubach spring were available as input and the percentage of missing values was 0.5% for Q and 0% for rainfall, snow height, EC and WT (for notations, see Table 1). The data gaps were filled through linear interpolation for all springs and variables. A topographical map of the springs is shown in Figure 1.

The model input variables are defined here as “features.” Multiple features can be created from one variable, for example, by calculating lagged versions of the variable. To account for seasonal effects, we implemented additional features representing the temporal information for the models. Please refer to Section 2.4.1 for more details, such as on the description of the features “day within a year” and “month within a year.”

We separated the data into three chronological time periods in order to tune our models without using the data set for final testing. The training data set has a range of five years (2012–2017), and the validation (2017–2018) and test data set (2018–2019) have a range of one year each.

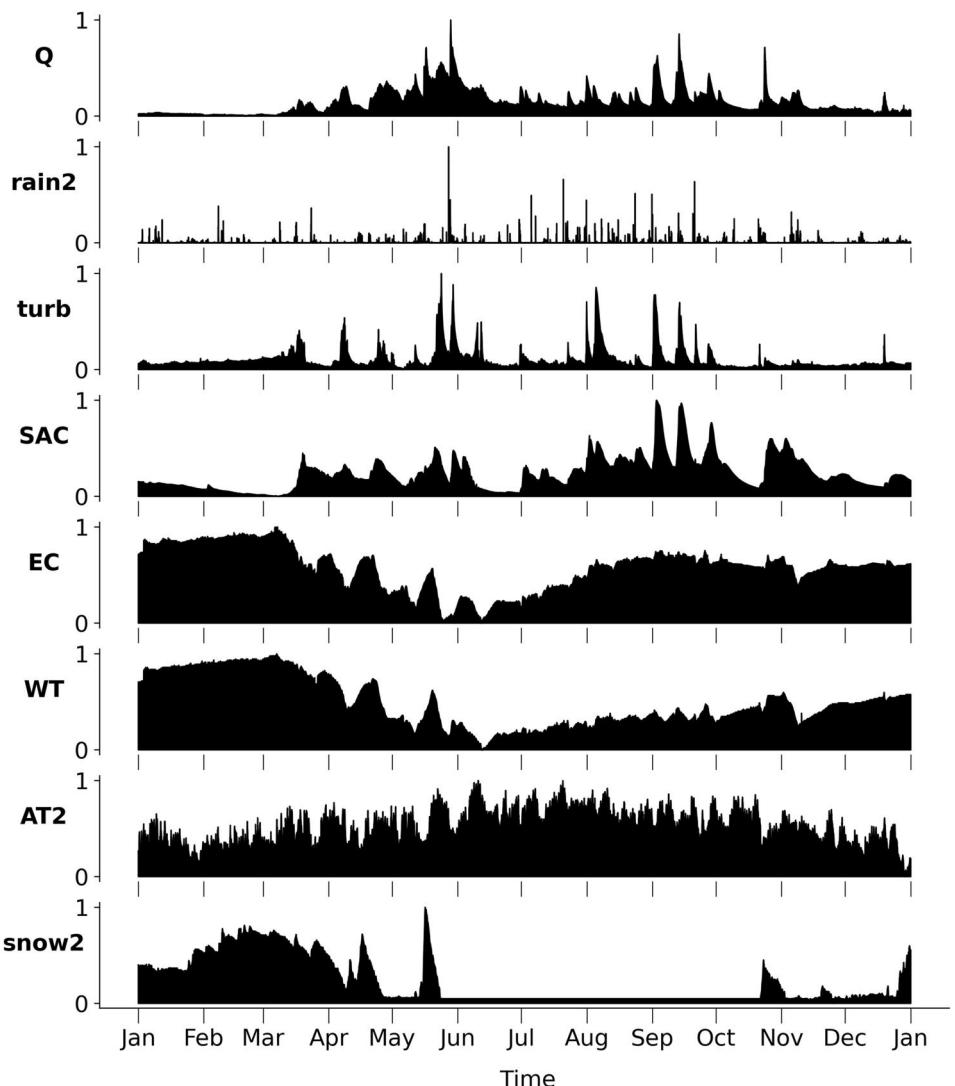


Figure 2. Normalized data from LKAS2 for 2014 (see Table 1 for feature notations and units).

The seasons were defined according to the meteorological seasons on the Northern Hemisphere, that is, with spring beginning on March 1, summer on June 1, autumn on September 1, and winter on December 1. In the catchment there is no permanent snow cover, as also shown at the meteorological station 2 (Figure 2). The highest discharges generally occur during the snowmelt period in spring. Most of the rainfall occurs in summer and autumn. Figure 2 further shows that turbidity and SAC peaks are related to discharge peaks. On average the temporal shifts between turbidity and discharge peaks are 11 ± 10 hr and between SAK and discharge they are 27 ± 5 hr, differing for each event, based on the data between 2012 and 2019.

2.2. Deep Learning and Baseline Models

Artificial neural networks (ANNs) are mathematical models that are inspired by biological neural networks in human brains (Müller et al., 1995). ANNs consist of neurons that are connected through weighted links, similar to axons. In those networks, layers consisting of neurons are created. The first layer is the input layer, which inserts data into the network. It is followed by a variable number of hidden layers. In each of the hidden layers, neurons are connected to the neurons in the layer before and after. Finally, the output layer is shaped according to the desired task. A specialized form of ANN are RNN, which feed their output back as input, allowing them to work with sequences of arbitrary length. They are commonly used in time series modeling (Coulibaly et al., 2001).

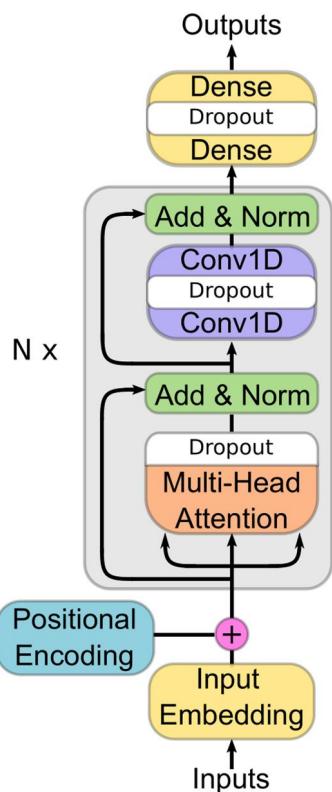


Figure 3. Transformer architecture with encoder, adapted from Vaswani et al. (2017).

of this, we used a dense layer for embedding to increase the dimension of our inputs. Positional encoding, implemented through sine and cosine functions, gives sequential information to the network and is added to the embedded data. This is done because the Transformer does not process each input step in the input window sequentially but pays attention to selected parts of the sequence. This allows the Transformer to obtain information about the temporal order of the input sequence.

Scaled Dot-Product Attention. The idea behind attention is to allow the model to calculate the importance of each of the steps in the input window for the current output. The input for the “scaled dot-product attention” consists of vectors called keys, queries (dimension d_k), and values. The dot product of the query with each of the keys is calculated to produce information about the respective relations. The result is scaled by $1/\sqrt{d_k}$ and Softmax (normalized exponential function) is applied, which transforms the result into a [0,1] interval. The scaling is incorporated to prevent non-useful Softmax results, that could occur from a large dot product. Finally, we multiply the result with the value vector. Thus, we receive an attention vector that includes information about the relations between different samples. In practice, the queries, keys and values are stacked into matrices Q , K , and V , and the scaled-dot product output accounts to

$$\text{Attention}(Q, K, V) = \text{Softmax}\left(\frac{QK^T}{\sqrt{d_k}}\right)V$$

Multi-Head Attention. Multi-head attention refers to computing the scaled dot-product attention multiple times with different K , Q , and V dimensions. The results are concatenated and projected through a linear layer. This mechanism provides more attention information than compared to only using a single attention head.

Residual Connections and Layer Normalization. Residual connections are shortcuts in the model where the information can skip certain blocks. In the gray encoder block in Figure 3, the black arrows that skip blocks are

Neural networks are trained for specific tasks through learning to detect patterns from a large amount of input data. Usually, a data set is iterated over multiple times during training, which is described by the number of epochs. The training itself is done through so-called “backpropagation,” a process where the error between predicted and measured output is propagated back through the layers, and weights are updated. Weights that contributed most to the error are changed the most. For a detailed introduction to neural networks, we refer to Calin (2020) and Lee et al. (2021).

2.2.1. Transformer

Transformers are neural networks for sequence-to-sequence modeling, originally implemented for natural language processing. Transformers process information globally over the input sequence, as opposed to sequentially, as in the case of RNNs. Thus, the vanishing information problem is not present in this architecture and parallel computation can be used to increase training speed. To learn relevant information from the input sequences, an attention mechanism is applied, which we will explain in the paragraph titled “Scaled dot-product attention.”

The Transformer was constructed according to Vaswani et al. (2017). In preliminary tests, we compared an encoder-decoder structure with an encoder-only/decoder-only structure for our karst discharge forecasting data set and found that the encoder-only architecture worked best for our problem. For this reason only the encode block of the original Transformer is used and depicted in Figure 3. We will now explain the implemented Transformer structure using the notation from Vaswani et al. (2017).

Input Embedding and Positional Encoding. In its original application for text data, Transformer embeddings were introduced to transform words to high-dimensional vectors. In our case, we have vectors to begin with. Despite

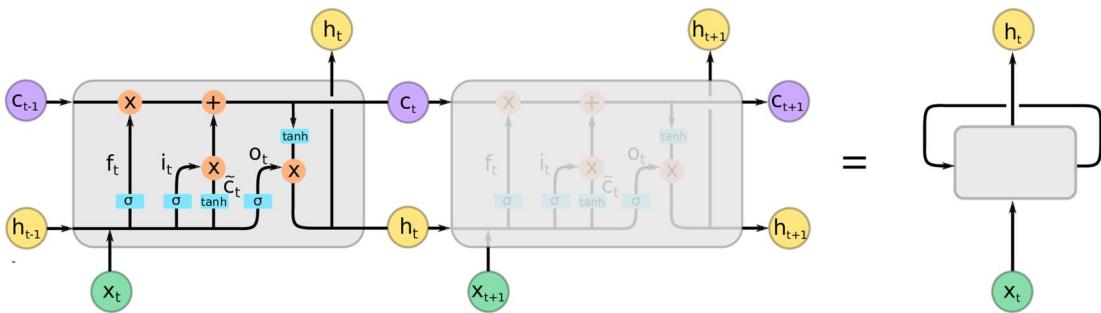


Figure 4. Long Short-Term Memory structure, x_t describes the input and h_t the hidden state, which is equal to the output at time t , adapted from Althoff et al. (2021).

residual connections. This reduces information losses throughout the network. Next, layer normalization is performed. This means that for every sample the mean and standard deviations of the output of the previous layer are calculated and used to normalize the output, which is then fed into the next layer.

Convolutional Layers and Model Output. Following the multi-head attention and normalization, two one dimensional convolutional layers are applied. They combine information for each input window step of the previous block through learned convolutional kernels. Again, residual connections are used to maintain ideal information flow. Dropout layers are both placed in the multi-head attention block, the convolutional block as well as the final dense block, which provides the hourly model output.

2.2.2. Long Short-Term Memory (LSTM) Neural Network

The LSTM by Hochreiter and Schmidhuber (1997) aims to reduce the vanishing information problem of conventional RNN by using the concept of a cell state that is updated and acts as memory storage. Furthermore, gates are introduced that help to add and remove information from the cell state. Each LSTM cell has three gates that control the information flow through the network: the forget gate f_t , the input gate i_t , and the output gate o_t .

An overview of the structure of an LSTM cell is given in Figure 4. The black arrows depict the data flow through the network. The joining lines indicate concatenation and splitting of lines indicates duplication. As an example for concatenation, the input x_t and hidden state h_{t-1} are stacked $[h_{t-1}, x_t]$ and flow into the forget gate f_t (Althoff et al., 2021).

Note that one LSTM layer consists of only one LSTM cell, where the cell and hidden state (c_t and h_t) are recurrently fed back into the cell, as can be seen in Figure 4. During training, its set of weight matrices (W_i, W_f, W_c, W_o), bias vectors (b_i, b_f, b_c, b_o), cell state c_t and hidden state h_t are updated at each time step of the input sequence. During inference, only the cell state c_t the hidden state h_t are changed. The dimension of the hidden state, which is equivalent to the dimension of the cell state, is a hyperparameter that can be chosen for each LSTM layer. It can be interpreted as the model's memory size. Note that the length of the input sequence has no influence on the number of parameters within an LSTM layer. For more details we refer to Althoff et al. (2021) and Hochreiter and Schmidhuber (1997). Both the LSTM and Transformer implemented in this study are designed for time series modeling. Their architecture, including the cell state of the LSTM and the attention mechanism of the Transformer are used to capture temporal dependencies within the dynamic karst system. Through the feature selection, the testing of input sequence lengths and frequencies, the models are refined to simulate the karst discharge. Finally, through extensive hyperparameter tuning, the optimal number of layers, units, dropout rate are found to effectively learn the connections between features and the karst spring discharge.

2.2.3. Baseline Model

We chose a common baseline model for comparison of the Transformer and LSTM results. The simple moving average method uses the mean discharge data over the input sample window for forecasting the 96 hr of the output window.

2.3. Experimental Setup

We implemented two different scenarios for modeling discharge using the Transformer and LSTM. For scenario Q+, we used the historical discharge, along with additional input features to forecast discharge. For scenario noQ, we excluded discharge from the input features. We performed a feature selection to obtain the optimal, (additional) features (Section 2.4.1). The model hyperparameters were then tuned (Section 2.4.2). For evaluating the model performances, we used the predictions on the test data set (Section 2.5).

2.3.1. Metrics

Model performance was measured by the following metrics, where N indicated the number of samples, Q_{obs}^t referred to the observed discharge at time t , Q_{mod}^t indicated the simulated discharge at time t , and $Q_{\text{obs}}^{\text{avg}}$ represented the average observed discharge. The NSE (Nash & Sutcliffe, 1970) is defined as

$$\text{NSE} = 1 - \frac{\sum_{t=1}^N (Q_{\text{obs}}^t - Q_{\text{mod}}^t)^2}{\sum_{t=1}^N (Q_{\text{obs}}^t - Q_{\text{obs}}^{\text{avg}})^2}$$

and is a measure of the ratio between the Mean Square Error (MSE) and the variance. The range of the NSE is from $-\infty$ to one, and a value of one indicates optimal model performance. The MAE returns the mean of the absolute difference between the observed and simulated data over all samples.

$$\text{MAE} = \frac{1}{N} \sum_{t=1}^N |Q_{\text{obs}}^t - Q_{\text{mod}}^t|$$

Similarly, the RMSE is the root over the mean of the squared differences between observed and simulated data. The MAE and RMSE are scale-dependent measures and the RMSE is more sensitive to outliers. Both metrics range from 0 and ∞ , and smaller values indicate good performance.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{t=1}^N (Q_{\text{obs}}^t - Q_{\text{mod}}^t)^2}$$

The sMAPE measures the model accuracy as percentage. The symmetry was included because the MAPE, which is similar to MAE but divides each subtraction in the sum by the absolute value of the observed value, has a bias towards larger values (Armstrong, 1985).

$$\text{sMAPE} = \frac{100}{N} \sum_{t=1}^N \frac{|Q_{\text{obs}}^t - Q_{\text{mod}}^t|}{(|Q_{\text{obs}}^t| + |Q_{\text{mod}}^t|)/2}$$

For evaluating the performance of the uncertainty intervals, we used the metric Percentage Of Coverage (POC) of observations within the prediction interval (Althoff et al., 2021; Zhang et al., 2009), counting how often the observed value lays in the prediction interval and returning the percentage of this count. The POC is defined as

$$\text{POC} = \frac{\sum_{t=1}^N I(Q_{\text{obs}}^t \in [q_{t,\text{low}}, q_{t,\text{up}}])}{N},$$

where I is the indicator function and $[q_{t,\text{low}}, q_{t,\text{up}}]$ is the prediction interval. As an additional metric for evaluating the model uncertainty, we used the measure of Average Interval Width (AW) (Althoff et al., 2021), defined as

$$AW = \frac{\sum_{t=1}^N (q_{t,\text{low}} - q_{t,\text{up}})}{N}.$$

2.4. Data Preprocessing and Model Calibration

2.4.1. Data Preprocessing

First, the training data is normalized per feature to values between 0 and 1. We saved the scaling factors to perform normalization on the validation and test data during inference on a sample-by-sample basis.

For the feature selection, we considered all features shown in Table 1, as well as the snow height difference between two consecutive time steps. Additionally, lagged meteorological data was used as input features. The time lag was chosen through maximizing correlations between discharge and lagged meteorological data.

The features “day within a year” (day) and “month within a year” (mth) provide information on the sample date to the model. The periodic nature was modeled by a cyclical implementation that uniquely characterizes the temporal information with sine and cosine functions. The term “cyclical” indicates that the months are not used as their respective month number (M) 1–12, because in this case, distances such as that between December and January would be different from the distance between January and February. For the information “month within a year” the cyclical encoding is defined as

$$\cos(2\pi M/12) \text{ and } \sin(2\pi M/12).$$

To select the ideal features we used the common method of forward selection as briefly summarized here. We begin by training different models using only one feature each and evaluate the models, based on the average sMAPE over lead times of one to 4 days. Next, we test if the best one-feature model can be improved by adding another feature. This stepwise process continues until there is no further improvement by adding a new feature.

The feature selection algorithm was performed separately for the scenarios Q+ and noQ for the Transformer and LSTM and for each spring.

2.4.2. Model Training, Regularization, and Tuning

The models were trained with the state-of-the-art optimization algorithm “Adam.” We selected the MAE as loss function. The batch size was set to 24 and the epochs were set to 100. Early stopping was implemented as a regularization measure, which stops the training if there is no improvement over a certain number of epochs. This threshold is called “patience” and was set to five after experimental trials. Dropout was also used for regularization and uncertainty quantification (Section 2.5.1). A comparison of a Transformer with and without dropout is shown in the Figure S7 in Supporting Information S1 highlighting the need for dropout for regularization.

For tuning, we used the hyperband algorithm implemented by KerasTuner from the open-source deep learning library, Keras. For the LSTM, we considered the number of layers, the number of units within these layers, the dropout rate, and the recurrent dropout rate as hyperparameters during tuning. For the Transformer, we considered the head size, the number of attention heads, the number of convolution filters, the number of encoder blocks, the dropout rate, the number of units in the final dense layer (dense units) and their dropout rate (dense dropout) as hyperparameters for tuning.

2.5. Model Comparison and Uncertainty Evaluation

2.5.1. Monte Carlo (MC) Dropout

MC dropout, or dropout ensemble, is a technique for obtaining prediction intervals as model output, as opposed to point estimates. Dropout is a technique to avoid overfitting. It operates by randomly deactivating (dropping) a certain percentage of neurons (the dropout rate) during training. Thus, the network becomes more robust, and improved regularization and model performances have been documented (Srivastava et al., 2014). In the past, dropout was only used during training; thus, for inference, all neurons in the network were active. MC dropout refers to dropout being active at the inference state as well. This leads to an interval estimate, also referred to as

“sister” forecasts when the forecasts are run multiple times, as opposed to a point estimate obtained with the standard dropout technique (Althoff et al., 2021). The model generates different forecasts for the same samples, because at each run different neurons are deactivated through random dropout sampling. Gal and Ghahramani (2016) showed that dropout in neural networks can be interpreted as a Bayesian approximation of a Gaussian process. The mean resulting from MC dropout samples is defined as the forecast, and the standard deviation as a measure of the uncertainty of the prediction.

2.5.2. Model Comparison

For each of the four models (Transformer Q+, LSTM Q+, Transformer noQ, LSTM noQ), 100 forecasting runs were performed for every sample in the test set. The resulting prediction intervals were used to quantify and compare the model performances and uncertainties.

2.5.3. Final Prediction Intervals and Uncertainty Evaluation for Q+ Scenario

Please note that for the evaluation of the model uncertainties a comparison of the standard deviations is not meaningful because these are related to the dropout rate, that is, a higher dropout rate leads to a larger standard deviation. Using the same dropout rate for all models generally allows a better comparison of the standard deviations (Althoff et al., 2021). In our case, however, the models strongly differed in their structure, one being an RNN and the other one being an attention-based model; thus, the model tuning resulted in different dropout rates. Setting all dropout rates to the same setting would decrease the individual model performance. We therefore decided to tune our models by including the dropout rate as hyperparameter, and develop a further process to ensure comparability.

For that, we scaled the prediction interval over the entire length of the validation data set, to obtain 80% POC on average, which was our selected optimization objective. This was done separately for the Transformer and the LSTM. We saved the scaling factors obtained from the validation data set to perform the uncertainty evaluation on the independent test data set.

2.6. Software

The models were implemented in Python (3.8.8). The software libraries tensorflow (2.8.0), keras (2.8.0), numpy (1.20.0), pandas (1.2.4), matplotlib (3.3.4), and inkscape (1.2.1) were used to generate all simulations and all figures, except Figure 1, which was created in QGIS (3.30.1). The code is available upon request.

3. Results

3.1. Different Spring Dynamics

The three chosen springs exhibit a wide spectrum of response times to rainfall events. The Pearson correlation between discharge and lagged rainfall data shows that the most dynamic spring is the Aubach spring, where the maximum correlation is found at a 4-hr lag (Figure S1 in Supporting Information S1). It is followed by the LKAS2 spring with a maximal correlation at 14 hr for the rain2 (13 hr for rain1). Finally, the Ursprung spring is the least responsive spring with a 33 hr time lag. These results correspond with the information about the spring discharge dynamics through max/min discharge ratios given in the study area section. The frequency of the model input and output was set to 1 hour. For LKAS2 we investigated frequencies between 1 hour and 1 day and found that lowering the frequencies reduces the performances of the models.

3.2. Feature Selection

The feature selection is performed by training the models using fixed hyperparameters, which are listed in the supplementary A in Supporting Information S1. In Figure 5 the feature selection result is shown on the validation data, since the test data information must not influence the model calibration in any way and thus the test data was not used for selecting the features.

For the LKAS2 spring, the feature selection for the Q+ scenario shows that discharge was a strong predictor and led to a good model performance on its own (Figure 5, top), especially for a lead time of one day. Only the rainfall data from station 2 improved the models for the Q+ scenario (NSE 2.3%, MAE 4.0%, RMSE 3.5%,

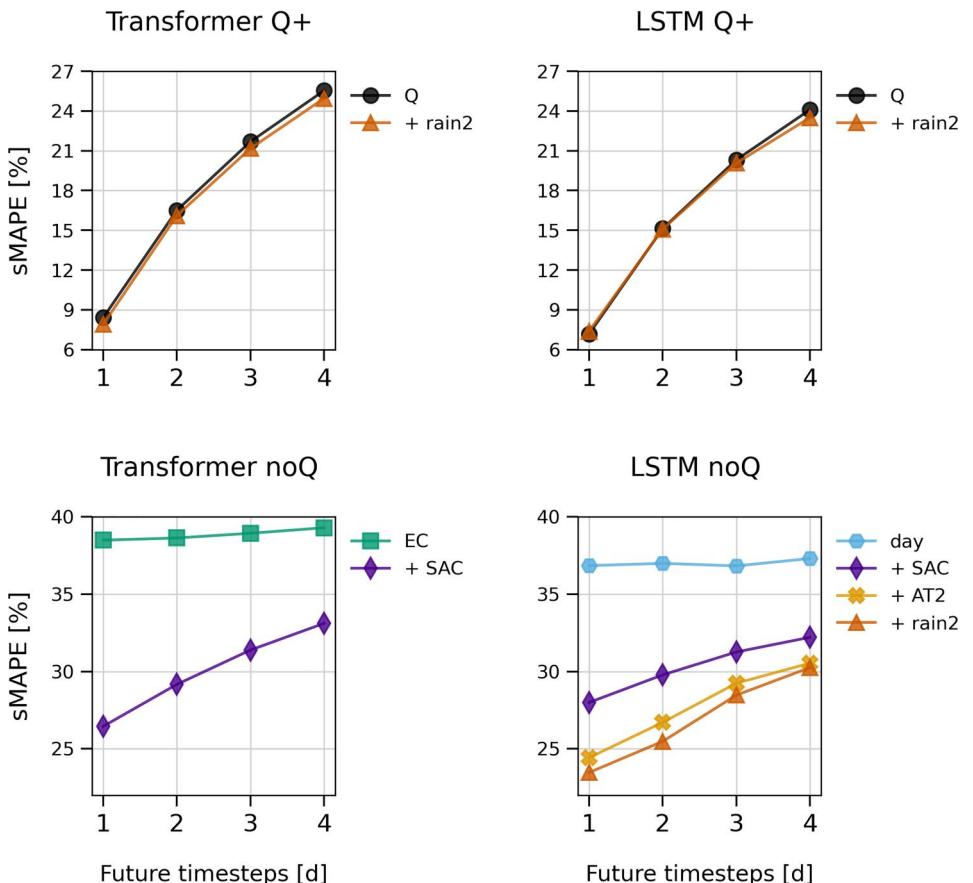


Figure 5. Symmetric Mean Absolute Percentage Error for Long Short-Term Memory feature selection results (right), Transformer feature selection results (left) for the two scenarios Q+ and noQ for LKAS2.

sMAPE 2.8% improvements for the Transformer, the improvements of the LSTM are comparable). Before hyperparameter tuning, we received model performances with an sMAPE < 10%, and an NSE > 0.9 for the one-day-ahead forecasts. The errors increased with increasing lead time, since short-term information was not available to the models for lead times of 2–4 days. Yet, both models achieved adequate performances for lead times of up to 4 days, with an sMAPE of <27% (Figure 5, top) and an NSE > 0.65 (results not shown in Figure).

For the noQ scenario, discharge is not used as input, and the results differ in terms of ideal features (Figure 5, bottom). The models show a substantially lower performance than in the Q+ scenario. For the noQ scenario simulations with the Transformer the EC was the strongest predictor, followed by the SAC. The strongest predictor for the LSTM is the encoded day feature. Air temperature, SAC and the rainfall also aided in improving the model. We found that some variables such as water temperature, did not improve the model performances even though they correlated moderately with discharge ($r = 0.5\text{--}0.7$, $p < 0.05$). On the contrary, air temperature and rainfall, with weak to no correlation with discharge, improved the results. This can be attributed to the non-linear relationship between the variables.

For the scenario Q+ and the Aubach spring the LSTM's selected features are Q and WT, and the Transformer's features are Q, month and WT. For the Ursprung spring and the LSTM Q and EC were the optimal features and for the Transformer Q and WT. For the scenario noQ and the Aubach spring the final best feature combination for LSTM is day, WT, EC and rainfall. For the Transformer it is EC, day and rainfall. For the Ursprung spring and the LSTM the best features are day, WT, EC and for the Transformer day and rainfall.

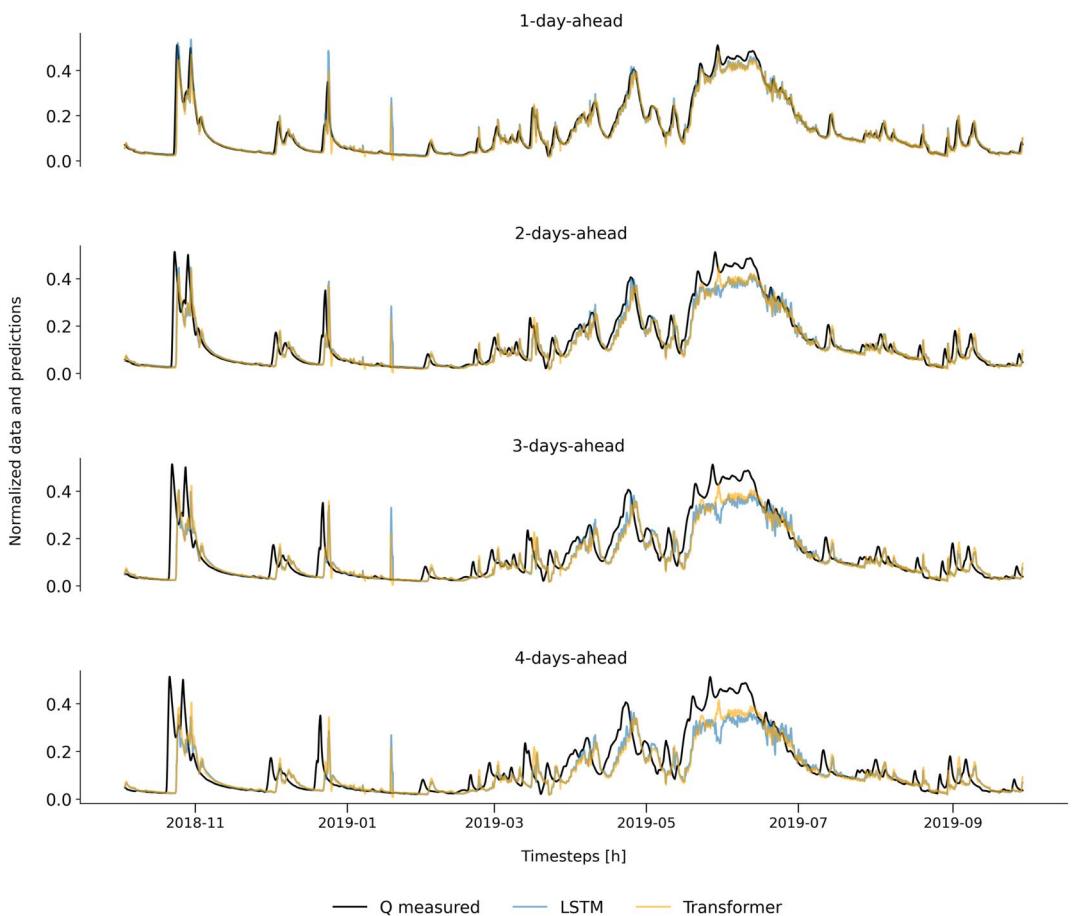


Figure 6. Normalized measured and predicted LKAS2 spring discharges (mean over 100 trials) of the Transformer and Long Short-Term Memory for 1–4 days of lead time for Q+ scenario.

3.3. Tuning Results and Model Comparison

The tuning results and ranges for all springs and models are shown in the supplementary section A in Supporting Information S1. For lead times of one day, the forecasted discharge agreed well with the measured discharge (Figure 6). During several peaks in December, March, and September, both models overestimated, and during the snowmelt period, both models underestimated the spring discharge. The predicted discharge peak in January can be explained by a rainfall event, that did not lead to a measured discharge response, potentially due to the present snow cover. For lead times of 2–4 days, the model performances gradually deteriorated. There was an increasing delay in the simulated onset and offset of peaks, as well as increasing errors during spring, with lead times. The Transformer better predicted the shape of the hydrograph than the LSTM for 2–4 days of lead time. For instance, the LSTM predicted a downward peak in June 2019, resulting in large errors in contrast to the Transformer.

The results in Table 2 show that a lower spring dynamic is generally associated with a better model performance, that is, all models performed best for the Ursprung spring, followed by the LKAS2 and Aubach spring. For the LKAS2 spring and the Q+ scenario the results show that the Transformer performed comparably well as the LSTM (<8% difference of NSE, MAE, RMSE, and sMAPE, Table 2). Both models resulted in error values of $\text{NSE} > 0.9$ and $\text{sMAPE} < 11\%$ for one-day-ahead forecasting. The Transformer resulted in 2%–21% better performance than the baseline model depending on the metric and lead time. For the Aubach spring in the Q+ scenario, the LSTM performance was 1%–11% better than the Transformer, and by 8%–87% better than the baseline model according to all metrics. The sMAPE is not given, due to the intermittent spring type resulting in a division by zero. For the Ursprung spring in the Q+ scenario, the Transformer achieved 1%–25% smaller errors than the LSTM, and 4%–51% smaller errors than the baseline model according to all metrics. To investigate whether the differences in the

Table 2*Performance Metrics for the Tuned Transformers and LSTMs, As Well As for the Baseline Model for Scenarios Q+*

Scenario Q+		LKAS 2			Aubach			Ursprung		
Metrics		Transformer	LSTM	Baseline	Transformer	LSTM	Baseline	Transformer	LSTM	Baseline
1-day-ahead	NSE	0.92	0.93	0.88	0.63	0.67	0.43	0.99	0.98	0.95
	MAE	0.63	0.58	0.80	0.27	0.24	0.42	17.1	22.8	33.4
	RMSE	1.30	1.22	1.57	0.64	0.60	0.82	36.2	44.5	73.3
	sMAPE	10.8	10.0	13.4	—	—	—	3.9	4.9	6.2
2-days-ahead	NSE	0.80	0.80	0.76	0.40	0.43	0.28	0.95	0.94	0.91
	MAE	1.16	1.15	1.29	0.38	0.36	0.50	30.7	36.1	45.8
	RMSE	2.12	2.12	2.32	0.84	0.82	0.92	71.3	78.2	96.8
	sMAPE	19.4	18.9	21.6	—	—	—	6.2	7.2	8.5
3-days-ahead	NSE	0.70	0.69	0.67	0.31	0.33	0.19	0.91	0.90	0.87
	MAE	1.49	1.50	1.60	0.43	0.42	0.54	42.4	46.7	55.7
	RMSE	2.60	2.63	2.72	0.90	0.89	0.97	97.1	101	115
	sMAPE	24.9	24.4	26.8	—	—	—	8.3	9.1	10.4
4-days-ahead	NSE	0.64	0.62	0.62	0.27	0.28	0.15	0.87	0.86	0.83
	MAE	1.71	1.73	1.79	0.46	0.45	0.55	51.2	54.7	63.5
	RMSE	2.86	2.92	2.93	0.93	0.92	1.00	115	118	131
	sMAPE	28.7	28.2	30.3	—	—	—	10.0	10.6	11.8

Note. Bold values indicate the best performance metrics per scenario. Units of the errors MAE and RMSE are given in (m^3/s) for the LKAS2 and Aubach spring. For the Ursprung spring the units are given in (m^3/hr) due to its small discharge.

model performances are significant, we performed the Welch's *t*-test, which was chosen due to the normally distributed data with unequal variances. The results show that there is a significant difference between the LSTM and Transformer model performance for every metric ($p < 0.01$) for LKAS2, in detail the NSE is significantly higher for the Transformer and the other metrics are significantly lower for the LSTM. Analogously, a significant difference was found when investigating the results for the Ursprung and Aubach spring ($p < 0.01$), that is, all metrics of the Ursprung spring are significantly better for the Transformer than for the LSTM and vice versa for the Aubach spring. The discharge forecasting models show an extensive performance difference between the two scenarios for all springs. In the scenario noQ, the Transformer and LSTM performance was 30%–220% poorer than for the scenario Q+ depending on the metric, lead time and spring.

We also evaluated the forecasts during low flow (lowest 20% of discharge) and high flow (highest 20% of discharge). The results are similar for the three springs. The errors are smallest for low flows and increase with flow rate. For the high flow the results are in accordance with the results from the entire subset. We found that for the Ursprung spring, the LSTM underestimated the high flows, whereas the Transformer's forecasts were better. In contrast, for the Aubach spring, the LSTM forecasts were better for high flows, where the Transformer underestimated the discharges. Generally, different models performed best depending on lead time and metric for the LKAS2 spring. During low flows, where discharge is almost constant, the deep learning models did not show an advantage over the baseline model. The NSE is mostly higher in the low and high flow than over the entire set, particularly for the high flows. This is likely due to the fact that the variance in the data set, which is used in the denominator of the NSE, is higher for the high flow than low flows. Thus, the ratio between the residual variance and the variance in the data is smaller for the high flows leading to a higher NSE. The low and high flow model performances, as well as the results for the noQ scenario are given in the Table S4 in Supporting Information S1, respectively Table S3 in Supporting Information S1.

3.4. Uncertainty Evaluation for Scenario Q+

For the uncertainty evaluation, we selected the scenario Q+, because the model performances were significantly better than for the noQ scenario, and thus more useful. We created prediction intervals based on the trained model (Section 2.5.3). We optimized the prediction intervals to achieve 80% of POC for each day of lead time.

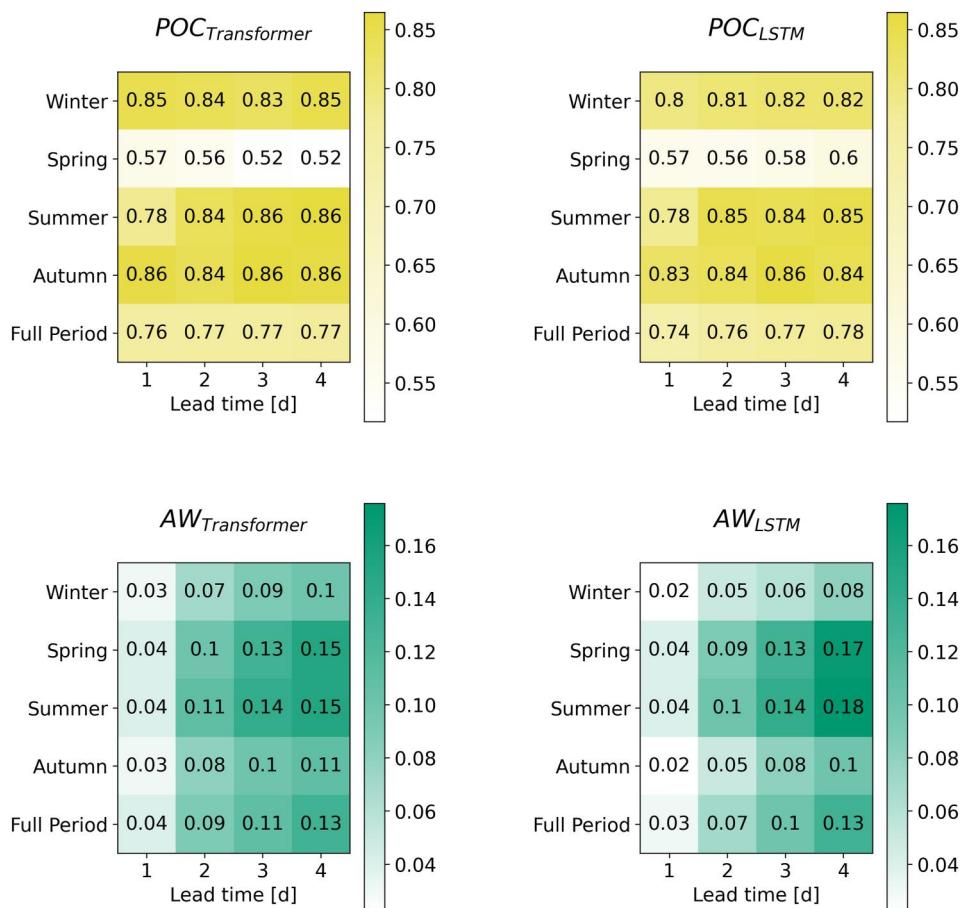


Figure 7. Long Short-Term Memory and Transformer seasonal prediction intervals for scenario Q+ for LKAS2 spring, evaluated by the percentage of coverage (POC (%)) and average width (AW (-)) of normalized simulated discharges.

Figure 7 shows the POC and AW of the prediction intervals for the full and seasonal periods. The 80% prediction intervals show the increase in width with increasing lead time and that the uncertainties are particularly high in spring and early summer when discharges are high due to snowmelt. For the one to four-day-ahead forecasts, we achieved a POC of 76%–77% for the Transformer and of 74%–78% for the LSTM. The results show comparable uncertainties for the Transformer and the LSTM throughout the year for lead times of one day. The AW is generally large in spring and summer, and smaller in autumn and winter, in comparison. Reliable prediction intervals and low model uncertainties, as shown by high POC and small AW values, are mainly achieved in autumn and winter. The POC and AW of the prediction intervals for the Ursprung and Aubach spring show similar results and are visualized in the Figures S2 and S4 in Supporting Information S1. Moreover, the prediction intervals for lead times of 1–4 days are shown in the Figures S3, S5, and S6 in Supporting Information S1 for all three springs.

4. Discussion

4.1. Transformer Forecasting Performance for Studied Karst System, Potential Improvements, and General Applicability

The primary aim of this paper was to evaluate the performance of the Transformer in forecasting karst spring discharge and how it compares to the LSTM as one of the best performing state-of-the-art deep learning models to date (Han et al., 2021). The good performance metrics shown in our study supported our hypothesis that the Transformer is a promising method for forecasting discharges of karst systems. In comparison to the LSTM, we found slight differences in the performance metrics, depending on the metric and lead time and spring.

The model performances (LSTM, Transformer and baseline) overall show that lower spring dynamics are associated with a higher model performance. This was also shown through a linear regression between the coefficient of variation of the discharge, used to quantify the spring dynamic, and the model performance metrics. We found a strong relationship, albeit weakly significant, likely due to the low sample size (R^2 : 0.96–0.99, P : 0.05–0.23). Our results indicate that the Transformer seems to have an advantage over the LSTM for karst springs with longer response times. Therefore, it might perform particularly well also for other hydrological applications with longer response times to hydrological forcing such as for forecasting groundwater quantity or quality. This could be due to the Transformer's attention mechanism, which allows incorporating information far back in time. This advantage is shown particularly for the Ursprung spring, which has the longest response time in our study, and where the Transformer outperformed the LSTM by 1%–25% (mean difference 9%). While for the LKAS2 spring the model performance of the Transformer was similar to the LSTM (<8% for all metrics, mean difference 3%), the Transformer better predicted the discharge shape during snowmelt. We analyzed the two discharge peaks at the end of May, where the Transformer's performance was much better than the LSTM's (Figure 6). We found that these rainfall events lasted approximately one day and that prior to the discharge peak, there were up to 6 hours of no rainfall. The performance difference is thus likely due the Transformer's attention mechanism, that could link the antecedent rainfall to the discharge increase. In contrast, the LSTM, for which the recent input is weighted higher than the antecedent due to its architecture, did not forecast a discharge increase. For the highly dynamic Aubach spring the LSTM outperformed the Transformer by 1%–11% (mean difference 4%). While the Transformer and the LSTM generally show a similar performance and are suitable for forecasting spring discharge (mean difference of 5% over all performance metrics and springs) in the Q+ scenario, the Transformer was previously not recognized as a useful candidate in this field and should be included in the list of ML model candidates in future studies. Finally, the results of the low and high flow analysis revealed that the best performing models over the entire year also performed best during high flows. Their high performance in contrast to the baseline underlines their usefulness for forecasting hydrological extremes.

In the noQ scenario, model performances were 30%–220% poorer depending on the spring, metric and lead time, showing that the relations of meteorological and water quality data with discharge were not sufficiently well captured. The performance could be improved for example, by making use of modeled water storages or spatial information from conceptual rainfall-runoff models as input.

To discuss the potential model improvements, we use the results of the model uncertainty and its seasonal variation. During autumn and winter, the LSTM and Transformer for the scenario Q+ produced forecasts with low uncertainty, as shown by a high POC and a small AW. In contrast during snowmelt in spring, both models did not perform well. This becomes apparent when visually comparing the simulated and observed hydrographs (Figure 6), and by the large AW and the low POC of the prediction intervals ranging from 52% to 60% (Figure 7). The larger AW of the prediction intervals in spring and summer is also likely due to the 2.4–3.1 times higher standard deviation of the discharge than in winter. Spatially distributed meteorological input features such as from satellite or radar data could potentially improve the forecast of high discharges in spring. Such data could be used as input for a convolutional neural network coupled with a Transformer, as CNN using spatial input data were already successfully applied for forecasting karst spring discharge (Wunsch et al., 2022). Alternatively, the machine learning models could be coupled with physically based snow models, as demonstrated by Xu et al. (2022). Moreover, the differences in seasonal model uncertainty could also motivate the implementation of separate models for different seasons, which would involve a separate feature selection for different seasons. In this way, the snowmelt information in spring might be learned by the seasonal model and improve the model's ability to predict discharge more adequately.

4.2. Potential of Input Features for Improving the Spring Discharge Forecasting Performance

The secondary aim of this paper was to explore the potential of various input features for improving the spring discharge forecasting performance, including hourly meteorological, spring discharge, and water quality data. The possibility to include physicochemical variables as input variables in deep learning methods is one main advantage over other hydrological modeling concepts, where this is not possible. The model results indicate that the selection of the optimum input features depends on the specific on-site characteristics such as the rock type, number and location of caves, aquifer structures, and meteorological conditions, and there is no optimum combination that can be generally applied for karst systems. To interpret the physical meaning of the input feature selection, we further discuss their relation to karst spring discharge responses. Rainfall impacted the model

performance in four out of six models in the noQ scenario, suggesting that most models could establish a relation of rainfall and discharge. For the LKAS2 spring in the Q+ scenario, the precipitation improved the LSTM and the Transformer performance only to a small extent. Snow, which was tested as input feature for all model combinations did not lead to an improved model performance. This low contribution is likely due to the low representativeness of the weather stations for the precipitation in the entire catchment of LKAS2, which was also confirmed by a hydrological HBV model in this catchment. With regards to snow fall, these are also complex, long-term processes that cannot be captured by the models operating on a short time window of hourly input data. Air temperature (AT2) improved the LSTM noQ model for LKAS2. When evaluating its contribution, we found that it improved the model particularly in spring (without AT2: NSE 0.24, with AT2: NSE 0.49), which is likely due to its impact on snowmelt and discharge during this season. While the impact of the meteorological variables on the karst spring discharges are more obvious due to their forcing characteristics, the impact of physicochemical parameters on the spring discharges is less clear. At least one physicochemical variable was selected as input feature for each of the six model constellations in the noQ scenario. The SAC was a major contributor to both models in the noQ scenario for LKAS2, underlining its strong relation to discharge. The SAC is a proxy-indicator for dissolved organic substances in water. Particularly, humic substances found in soil can be measured via spectroscopy at this wavelength (Stadler et al., 2010). The organic substances in spring water are mostly surface-associated, and generally increase during events, which explains its strong impact on the modeled discharge. The turbidity relates to the number of suspended solids in water. Depending on the velocity of the internal streams, these particles can settle and build up sediments inside the caves or remain suspended in the aquifer. During events these sediments can be resuspended in the aquifer. The relation between turbidity and discharge thus depends on the flow velocities of internal streams and the time between the events, which might explain why the turbidity was not a good predictor variable. The EC of the water is mainly determined by the dissolution of ions forming carbonate rock and the dilution of the dissolved ions with rainwater. EC has generally such a strong correlation with discharge, that it can be used as proxy for discharge, as shown for example, by Chang et al. (2022) for a small karst catchment in China (Chang et al., 2022). In our study it improved four of the six models in the noQ scenario when used as input feature, particularly for the Aubach spring which has a short response time. Finally, water temperature also improved two of the six models in the noQ scenario, in particular at the Aubach spring, where the water temperature dropped quickly during snowmelt. Generally, the more dynamic Aubach spring exhibits wider water temperature ranges (4°C) than the LKAS2 (0.8°C) and the Ursprung spring (0.4°C). An inverse relation between water temperature and discharge and a positive relation between EC and water temperature was also found for example, in the Jadro karst spring in Croatia (Bonacci & Roje-Bonacci, 2023).

Our experiments with meteorological features shifted in time did not result in improved model performances. A reason for this could be our evaluation method. As we averaged the model performances over lead times of one to 4 days during model optimization, it is possible that the shift improved the model for a lead time of one day but decreased the performance for two to 4 days of lead time. In future studies, one could thus reevaluate the shifting experiment on a daily basis and adapt the modeling process to create x models for one to x day lead times. The feature selection and shifts could thus be individually optimized for each day of lead time to improve the performance of the respective models.

4.3. Implications for Water Management

Discharge forecasts are essential for water supply management to achieve an optimal mix of potable water from different karst springs and an optimal distribution in the urban water infrastructure. The Transformer and LSTM performed very well for one-day-ahead forecasts and performed acceptable for up to 3 days of lead time, based on the performance metrics and the targeted forecasting horizon communicated by local water authorities. The requirements may differ from site to site, as for different water infrastructures the storage space, and the population size and density differ. Based on our results, we generally recommend including the discharges as input features for data-driven forecasting models. For the case when there are gaps in discharge observations, however, we showed that a large variety of online physicochemical and meteorological features is of great advantage to obtain useful forecasts. To ensure robust forecasts of spring discharges, continuous measurements of various water quality and meteorological features are thus advantageous to support water management. Both the LSTM and Transformer provide reliable forecasts of spring discharges during winter and autumn even for lead times up to 4 days according to our simulations. Future research is needed for improving the spring discharge forecasts during snow melting periods.

5. Conclusion

In this paper, we tested for the first time the Transformer for forecasting the discharges of karst springs, which are important drinking water resources worldwide. As test cases we used three karst springs in Austria covering a wide range of hydrological and hydrogeological conditions. The generated spring discharge forecasting models were built on an extensive, high-frequency data set of hydrological, meteorological, and water quality features.

The Transformer showed generally a high performance in forecasting karst discharges. It outperformed the state-of-the-art LSTM for the low dynamic Ursprung spring with a long response time (9% better performance metrics, NSE ranging from 0.87 to 0.99, and sMAPE from 3.9% to 10% for lead times of 1–4 days). Otherwise, the Transformer performed comparably well to the LSTM for the LKAS2 spring with a medium response time (NSE from 0.64 to 0.92 and sMAPE from 10.8% to 28.7% for lead times of 1–4 days). In spring, however we found multiple long rainfall events, for which only the Transformer was able to forecast the discharge response, likely due to its attention mechanism. Only for the highly dynamic Aubach spring with a short response time the LSTM outperformed the Transformer (4% better performance metrics, NSE from 0.28 to 0.67). The model uncertainties were lowest in winter and autumn according to the analysis of the prediction intervals (POC, AW). The best forecasting performance was achieved when including the spring discharge as an input variable (scenario Q+). The performance was 30%–220% poorer when using solely meteorological and water quality features (scenario noQ). The study showed that the Transformer is a promising deep learning approach to forecast karst spring discharge with distinct advantages compared to the LSTM and should thus be considered as a potential candidate in future model comparisons.

Data Availability Statement

The LKAS2 data supporting this research was provided by local authorities. The data are under restrictions that include a required NDA, and are not accessible to the public or research community because the area belongs to a water supply region in Austria. The data for the Aubach and Ursprung spring was provided by the Federal Ministry of the Republic of Austria for Water Management—Department of Water Balance (data status July 2023).

Acknowledgments

This work was supported by a research cooperation between Vienna Water (City of Vienna) and the Interuniversity Cooperation Centre for Water & Health (ICC Water & Health) in the frame of the Vienna Water Resource Systems project (ViWa 2020+), by the Austrian Science Fund (FWF) BACTOTRANS project [10.55776/P35733], and the Austrian Science Fund (FWF) as part of the Vienna Doctoral program on water resources systems [10.55776/W1219]. For open access purposes, the author has applied a CC BY public copyright license to any author accepted manuscript version arising from this submission. The TU Wien funded the personal costs of Anna Pöhlz in the frame of the ViWa 2020+ project. This is a joint project of the ICC Water & Health (www.waterandhealth.at).

References

- Althoff, D., Rodrigues, L. N., & Bazame, H. C. (2021). Uncertainty quantification for hydrological models based on neural networks: The dropout ensemble. *Stochastic Environmental Research and Risk Assessment*, 35(5), 1051–1067. <https://doi.org/10.1007/s00477-021-01980-8>
- An, L., Hao, Y., Yeh, T. C. J., Liu, Y., Liu, W., & Zhang, B. (2020). Simulation of karst spring discharge using a combination of time–frequency analysis methods and long short-term memory neural networks. *Journal of Hydrology*, 589, 125320. <https://doi.org/10.1016/j.jhydrol.2020.125320>
- Armstrong, J. S. (1985). *Long-range forecasting: From crystal ball to computer*. John Wiley & Sons, Inc.
- Bonacci, O., & Roje-Bonacci, T. (2023). Water temperature and electrical conductivity as an indicator of karst aquifer: The case of Jadro Spring (Croatia). *Carbonates and Evaporites*, 38(3), 55. <https://doi.org/10.1007/s13146-023-00881-x>
- Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., et al. (2020). Language models are few-shot learners. [Preprint]. *arXiv:2005.14165 [cs]*. Retrieved from <http://arxiv.org/abs/2005.14165>
- Calin, O. (2020). *Neural networks, deep learning architectures: A mathematical approach* (pp. 167–198). Springer International Publishing (Springer Series in the Data Sciences). https://doi.org/10.1007/978-3-030-36721-3_6
- Çallı, S. S., Çallı, K. Ö., Tuğrul Yılmaz, M., & Çelik, M. (2022). Contribution of the satellite-data driven snow routine to a karst hydrological model. *Journal of Hydrology*, 607, 127511. <https://doi.org/10.1016/j.jhydrol.2022.127511>
- Chang, Y., Mewes, B., & Hartmann, A. (2022). Using LSTM to monitor continuous discharge indirectly with electrical conductivity observations. *Hydrology and Earth System Sciences Discussions*, 1–19. <https://doi.org/10.5194/hess-2022-77>
- Chen, C., He, W., Zhou, H., Xue, Y., & Zhu, M. (2020). A comparative study among machine learning and numerical models for simulating groundwater dynamics in the Heihe River Basin, northwestern China. *Scientific Reports*, 10(1), 3904. <https://doi.org/10.1038/s41598-020-60698-9>
- Chen, Z., Auler, A. S., Bakalowicz, M., Drew, D., Griger, F., Hartmann, J., et al. (2017). The world karst aquifer mapping project: Concept, mapping procedure and map of Europe. *Hydrogeology Journal*, 25(3), 771–785. <https://doi.org/10.1007/s10400-016-1519-3>
- Cheng, S., Qiao, X., Shi, Y., & Wang, D. (2021). Machine learning for predicting discharge fluctuation of a karst spring in North China. *Acta Geophysica*, 69(1), 257–270. <https://doi.org/10.1007/s11600-020-00522-0>
- Coulibaly, P., Anctil, F., & Bobée, B. (2001). Multivariate reservoir inflow forecasting using temporal neural networks. *Journal of Hydrologic Engineering*, 6(5), 367–376. [https://doi.org/10.1061/\(ASCE\)1084-0699\(2001\)6:5\(367\)](https://doi.org/10.1061/(ASCE)1084-0699(2001)6:5(367))
- Daliakopoulos, I. N., Coulibaly, P., & Tsanis, I. K. (2005). Groundwater level forecasting using artificial neural networks. *Journal of Hydrology*, 309(1), 229–240. <https://doi.org/10.1016/j.jhydrol.2004.12.001>
- Devlin, J., Chang, M.-W., Le, K., & Toutanova, K. (2019). BERT: Pre-training of deep bidirectional Transformers for language understanding [Preprint]. *arXiv:1810.04805 [cs]*. Retrieved from <http://arxiv.org/abs/1810.04805>
- Fang, Z., Wang, Y., Peng, L., & Hong, H. (2021). Predicting flood susceptibility using LSTM neural networks. *Journal of Hydrology*, 594, 125734. <https://doi.org/10.1016/j.jhydrol.2020.125734>

- Farnleitner, A., Wilhartz, I., Ryzinska, G., Kirschner, A. K. T., Stadler, H., Burtscher, M. M., et al. (2005). Bacterial dynamics in spring water of two contrasting alpine karst aquifers indicate autochthonous microbial endokarst communities. *Environmental Microbiology*, 7(8), 1248–1259. <https://doi.org/10.1111/j.1462-2920.2005.00810.x>
- Ford, D., & Williams, P. (2007). Karst hydrogeology and geomorphology.
- Gal, Y., & Ghahramani, Z. (2016). Dropout as a Bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd international conference on machine learning. International conference on machine learning* (pp. 1050–1059). PMLR. Retrieved from <https://proceedings.mlr.press/v48/gal16.html>
- Goldscheider, N. (2005). Fold structure and underground drainage pattern in the alpine karst system Hochifen-Gottesacker. *Eclogae Geologicae Helvetiae*, 98, 1–17. <https://doi.org/10.1007/s00115-005-1143-z>
- Han, Z., Zhao, J., Leung, H., Ma, K. F., & Wang, W. (2021). A review of deep learning models for time series prediction. *IEEE Sensors Journal*, 21(6), 7833–7848. <https://doi.org/10.1109/JSEN.2019.2923982>
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, 9(8), 1735–1780. <https://doi.org/10.1162/neco.1997.9.8.1735>
- Husic, A., Al-Aamery, N., & Fox, J. F. (2022). Simulating hydrologic pathway contributions in fluvial and karst settings: An evaluation of conceptual, physically-based, and deep learning modeling approaches. *Journal of Hydrology X*, 17, 100134. <https://doi.org/10.1016/j.hydroa.2022.100134>
- Jeannin, P.-Y., Artigue, G., Butscher, C., Chang, Y., Charlier, J. B., Duran, L., et al. (2021). Karst modelling challenge 1: Results of hydrological modelling. *Journal of Hydrology*, 600, 126508. <https://doi.org/10.1016/j.jhydrol.2021.126508>
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall-runoff modelling using Long Short-Term Memory (LSTM) networks. *Hydrology and Earth System Sciences*, 22(11), 6005–6022. <https://doi.org/10.5194/hess-22-6005-2018>
- Lambrakis, N., Andreou, A. S., Polydoropoulos, P., Georgopoulos, E., & Bountis, T. (2000). Nonlinear analysis and forecasting of a brackish karstic spring. *Water Resources Research*, 36(4), 875–884. <https://doi.org/10.1029/1999WR900353>
- Lee, T., Singh, V. P., & Cho, K. H. (2021). Deep learning for hydrometeorology and environmental science.
- Li, S., Jin, X., Xuan, Y., Zhou, X., Chen, W., Wang, Y. X., & Yan, X. (2019). Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting. In *Advances in neural information processing systems*. Curran Associates, Inc. Retrieved from <https://proceedings.neurips.cc/paper/2019/hash/6775a0635c302542da2c32aa19d86be0-Abstract.html>
- Liong, S.-Y., & Chandrasekaran, S. (2007). Flood stage forecasting with support vector machines. *JAWRA Journal of the American Water Resources Association*, 38(1), 173–186. <https://doi.org/10.1111/j.1752-1688.2002.tb01544.x>
- Mazzilli, N., Guinot, V., Jourde, H., Lecoq, N., Labat, D., Arfib, B., et al. (2017). KarstMod: A modelling platform for rainfall - Discharge analysis and modelling dedicated to karst systems. *Environmental Modelling & Software*, 122, 103927. <https://doi.org/10.1016/j.envsoft.2017.03.015>
- Mohammadi Farsani, R., & Pazouki, E. (2021). A transformer self-attention model for time series forecasting. *Journal of Electrical and Computer Engineering Innovations*, 9(1), 1–10. <https://doi.org/10.22061/jecei.2020.7426.391>
- Müller, B., Reinhardt, J., & Strickland, M. T. (1995). The structure of the central nervous system. In B. Müller, J. Reinhardt, & M. T. Strickland (Eds.), *Neural networks: An introduction* (pp. 3–12). Springer. https://doi.org/10.1007/978-3-642-57760-4_1
- Nash, J. E., & Sutcliffe, J. V. (1970). River flow forecasting through conceptual models part I — A discussion of principles. *Journal of Hydrology*, 10(3), 282–290. [https://doi.org/10.1016/0022-1694\(70\)90255-6](https://doi.org/10.1016/0022-1694(70)90255-6)
- Nayak, P. C., Rao, Y. R. S., & Sudheer, K. P. (2006). Groundwater level forecasting in a shallow aquifer using artificial neural network approach. *Water Resources Management*, 20(1), 77–90. <https://doi.org/10.1007/s11269-006-4007-z>
- Olarinoye, T., Gleeson, T., Marx, V., Seeger, S., Adinehvand, R., Allocata, V., et al. (2020). Global karst springs hydrograph dataset for research and management of the world's fastest-flowing groundwater. *Scientific Data*, 7(1), 59. <https://doi.org/10.1038/s41597-019-0346-5>
- Ortner, H., & Kilian, S. (2022). Thrust tectonics in the Wetterstein and Mieming mountains, and a new tectonic subdivision of the Northern Calcareous Alps of Western Austria and Southern Germany. *International Journal of Earth Sciences*, 111(2), 543–571. <https://doi.org/10.1007/s00531-021-02128-3>
- Plan, L., Kuschnig, G., & Stadler, H. (2010). Case study: Kläffer spring—The major spring of the Vienna water supply (Austria). In *Groundwater hydrology of springs* (pp. 411–427). Elsevier.
- Rahbar, A., Mirarabi, A., Nakhaei, M., Talkhabri, M., & Jamali, M. (2022). A comparative analysis of data-driven models (SVR, ANFIS, and ANNs) for daily karst spring discharge prediction. *Water Resources Management*, 36(2), 589–609. <https://doi.org/10.1007/s11269-021-03041-9>
- Reischer, G. H., Haider, J. M., Sommer, R., Stadler, H., Keiblanger, K. M., Hornek, R., et al. (2008). Quantitative microbial faecal source tracking with sampling guided by hydrological catchment dynamics. *Environmental Microbiology*, 10(10), 2598–2608. <https://doi.org/10.1111/j.1462-2920.2008.01682.x>
- Reszler, C., Komma, J., Stadler, H., Strobl, E., & Blöschl, G. (2018). A propensity index for surface runoff on a karst plateau. *Hydrology and Earth System Sciences*, 22(12), 6147–6161. <https://doi.org/10.5194/hess-22-6147-2018>
- Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15, 1929–1958. Retrieved from https://www.jmlr.org/papers/volume15/srivastava14a/srivastava14a.pdf?utm_content=buffer79b43&utm_medium=social&utm_source=twitter.com&utm_campaign=buffer
- Stadler, H., Klock, E., Skritek, P., Mach, R. L., Zerobin, W., & Farmleitner, A. H. (2010). The spectral absorption coefficient at 254 nm as a real-time early warning proxy for detecting faecal pollution events at alpine karst water resources. *Water Science and Technology*, 62(8), 1898–1906. <https://doi.org/10.2166/wst.2010.500>
- van den Broeke, J., Picher, W., Zerobin, W., & Hofstädter, F. (2008). Use of in-situ UV/Vis spectrometry in water monitoring in Vienna. Retrieved from https://www.s-can.at/wp_contents/uploads/2021/09/p_2008_01.pdf
- Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., et al. (2017). Attention is all you need. In *Advances in neural information processing systems*. Curran Associates, Inc. Retrieved from <https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fb053c1c4a845aa-Abstract.html>
- Wunsch, A., Liesch, T., & Broda, S. (2021). 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). *Hydrology and Earth System Sciences*, 25(3), 1671–1687. <https://doi.org/10.5194/hess-25-1671-2021>
- Wunsch, A., Liesch, T., Cinkus, G., Ravbar, N., Chen, Z., Mazzilli, N., et al. (2022). Karst spring discharge modeling based on deep learning using spatially distributed input data. *Hydrology and Earth System Sciences*, 26(9), 2405–2430. <https://doi.org/10.5194/hess-26-2405-2022>
- Xiang, Z., Yan, J., & Demir, I. (2020). A rainfall-runoff model with LSTM-based sequence-to-sequence learning. *Water Resources Research*, 56(1), e2019WR025326. <https://doi.org/10.1029/2019WR025326>

- Xu, T., Longyang, Q., Tyson, C., Zeng, R., & Neilson, B. T. (2022). Hybrid physically based and deep learning modeling of a snow dominated, mountainous, karst watershed. *Water Resources Research*, 58(3), e2021WR030993. <https://doi.org/10.1029/2021WR030993>
- Zeyer, A., Bahar, P., Irie, K., Schlüter, R., & Ney, H. (2019). A comparison of transformer and LSTM encoder decoder models for ASR. In *2019 IEEE automatic speech recognition and understanding workshop (ASRU)*. *2019 IEEE automatic speech recognition and understanding workshop (ASRU)* (pp. 8–15). <https://doi.org/10.1109/ASRU46091.2019.9004025>
- Zhang, X., Liang, F., Srinivasan, R., & Van Liew, M. (2009). Estimating uncertainty of streamflow simulation using Bayesian neural networks. *Water Resources Research*, 45(2), W02403. <https://doi.org/10.1029/2008WR007030>