Hydrol. Earth Syst. Sci., 25, 2951–2977, 2021
https://doi.org/10.5194/hess-25-2951-2021

Hydrology and Earth System Sciences
Open Access
EGU

Machine-learning methods for stream water temperature prediction

Moritz Feigl[1,★], **Katharina Lebiedzinski**[1,★], **Mathew Herrnegger**[1], **and Karsten Schulz**[1]

[1]Institute for Hydrology and Water Management, University of Natural Resources and Life Sciences, Vienna, Austria
★These authors contributed equally to this work.

Correspondence: Moritz Feigl (moritz.feigl@boku.ac.at)

Received: 21 December 2020 – Discussion started: 14 January 2021
Revised: 15 April 2021 – Accepted: 27 April 2021 – Published: 31 May 2021

Abstract. Water temperature in rivers is a crucial environmental factor with the ability to alter hydro-ecological as well as socio-economic conditions within a catchment. The development of modelling concepts for predicting river water temperature is and will be essential for effective integrated water management and the development of adaptation strategies to future global changes (e.g. climate change). This study tests the performance of six different machine-learning models: step-wise linear regression, random forest, eXtreme Gradient Boosting (XGBoost), feed-forward neural networks (FNNs), and two types of recurrent neural networks (RNNs). All models are applied using different data inputs for daily water temperature prediction in 10 Austrian catchments ranging from 200 to 96 000 km^2 and exhibiting a wide range of physiographic characteristics. The evaluated input data sets include combinations of daily means of air temperature, runoff, precipitation and global radiation. Bayesian optimization is applied to optimize the hyperparameters of all applied machine-learning models. To make the results comparable to previous studies, two widely used benchmark models are applied additionally: linear regression and air2stream.

With a mean root mean squared error (RMSE) of 0.55 °C, the tested models could significantly improve water temperature prediction compared to linear regression (1.55 °C) and air2stream (0.98 °C). In general, the results show a very similar performance of the tested machine-learning models, with a median RMSE difference of 0.08 °C between the models. From the six tested machine-learning models both FNNs and XGBoost performed best in 4 of the 10 catchments. RNNs are the best-performing models in the largest catchment, indicating that RNNs mainly perform well when processes with long-term dependencies are important. Furthermore, a wide range of performance was observed for different hyperparameter sets for the tested models, showing the importance of hyperparameter optimization. Especially the FNN model results showed an extremely large RMSE standard deviation of 1.60 °C due to the chosen hyperparameters.

This study evaluates different sets of input variables, machine-learning models and training characteristics for daily stream water temperature prediction, acting as a basis for future development of regional multi-catchment water temperature prediction models. All preprocessing steps and models are implemented in the open-source R package wateRtemp to provide easy access to these modelling approaches and facilitate further research.

1 Introduction

Water temperature in rivers should not be considered only a physical property, since it is a crucial environmental factor and a substantial key element for water quality and aquatic habitats. In particular, it influences riverine species by governing e.g. metabolism (Álvarez and Nicieza, 2005), distribution (Boisneau et al., 2008), abundance (Wenger et al., 2011), community composition (Dallas, 2008) and growth (Imholt et al., 2010); thus, aquatic organisms have a specific range of river temperature they are able to tolerate (Caissie, 2006). Due to the impact of water temperature on chemical processes (Hannah et al., 2008) and other physical properties such as density, vapour pressure and viscosity (Stevens et al., 1975), stream temperature indirectly influences key ecosystem processes such as primary production, decomposition and nutrient cycling within rivers (Friberg et al., 2009). These parameters and processes affect the level of dissolved oxygen (Sand-Jensen and Pedersen, 2005) and, of course, have a major influence on water quality (Beaufort et al., 2016).

Besides its ecological importance, river temperature is also of socio-economic interest for electric power and industry (cooling), drinking water production (hygiene, bacterial pollution) and fisheries (fish growth, survival and demographic characteristics) (Hannah and Garner, 2015). Hence, a changing river temperature can strongly alter the hydro-ecological and socio-economic conditions within the river and its neighbouring region. Assessing alterations of this sensitive variable and its drivers is essential for managing impacts and enabling prevention measurements.

Direct temperature measurements are often scarce and rarely available. For successful integrated water management, it will be essential to derive how river temperature will be developing in the future, in particular when considering relevant global change processes (e.g. climate change), but also on shorter timescales. The forecast, for example, of river temperature with a lead time of a few days can substantially improve or even allow the operation of thermal power plants. Two aspects are important: the efficiency of cooling depends on the actual water temperature. On the other hand, legal constraints regarding maximum allowed river temperatures due to ecological reasons can be exceeded when warmed-up water is directed into the river after the power plant. This is especially relevant during low-flow conditions in hot summers. Knowledge of the expected water temperature in the next few days is therefore an advantage. An important step in this context is the development of appropriate modelling concepts to predict river water temperature to describe thermal regimes and to investigate the thermal development of a river.

In the past, various models were developed to investigate thermal heterogeneity at different temporal and spatial scales, the nature of past availability and likely future trends (Laizé et al., 2014; Webb et al., 2008). In general, water temperature in rivers is modelled by process-based models, statistical/machine-learning models or a combination of both approaches. Process-based models represent physical processes controlling river temperature. According to Dugdale et al. (2017), these models are based on two key steps: first, calculating energy fluxes to or from the river and then determining the temperature change in a second step. Calculating the energy fluxes means solving the energy balance equation for a river reach by considering the heat fluxes at the air–water and riverbed–water interfaces (Beaufort et al., 2016). These demanding energy budget components are derived either by field measurements or by approximations (Caissie and Luce, 2017; Dugdale et al., 2017; Webb and Zhang, 1997), highlighting the complexity and parametrization of this kind of model. Although it is not feasible to monitor these components over long periods or at all points along a river network and contributing catchments (Johnson et al., 2014), they provide clear benefits: (i) give insights into the drivers of river water temperature and (ii) inform about metrics, which can be used in larger statistical models and (iii) different impact scenarios (Dugdale et al., 2017). These arguments are also the reasons why data-intensive process-based models are widely used despite their high complexity.

Statistical and machine-learning models are grouped into parametric approaches, including regression (e.g. Mohseni and Stefan, 1999) and stochastic models (e.g. Ahmadi-Nedushan et al., 2007) and non-parametric approaches based on computational algorithms like neural networks or k-nearest neighbours (Benyahya et al., 2007). In contrast to process-based models, statistical models cannot inform about energy transfer mechanisms within a river (Dugdale et al., 2017). However, unlike process-based models, they do not require a large number of input variables, which are unavailable in many cases. Non-parametric statistical models have gained attention in the past few years. Especially machine-learning techniques have been proofed to be useful tools in river temperature modelling already (Zhu and Piotrowski, 2020).

For this study we chose a set of state-of-the-art machine-learning models that showed promising results for water temperature prediction or in similar time-series prediction tasks. The six chosen models are step-wise linear regression, random forest, eXtreme Gradient Boosting (XGBoost), feed-forward neural networks (FNNs) and two types of recurrent neural networks (RNNs). Step-wise linear regression models combine an iterative variable selection procedure with linear regression models. The main advantage of step-wise linear regression is the possibility of a variable selection procedure that also includes all variable interaction terms, which is only possible due to the short run times when fitting the model. The main disadvantages are the linear regression specific assumptions (e.g. linearity, independence of regressors, normality, homoscedasticity) that might not hold for a given problem, which consequently could lead to a reduced model performance. To our knowledge only one previous study by Neumann et al. (2003) already applied this method for predicting daily maximum river water temperature.

The random forest model (RF) (Breiman, 2001) is an ensemble-learning model that averages the results of multiple regression trees. Since they consist of a ensemble of regression trees that are trained on random subsamples of the data, RF models are able to model linear and non-linear dependencies and are robust to outliers. RF models are fast and easy to use, as they do not need extensive hyperparameter tuning (Fernández-Delgado et al., 2014). This could also be a disadvantage as it is also difficult to further improve RF models by hyperparameter optimization (Bentéjac et al., 2021). To date, only one previous study by Heddam et al. (2020) applied RF for predicting lake surface temperatures. Zhu et al. (2019d) used bootstrap-aggregated decision trees, which are similar but do not include the random variable sampling for splitting the tree nodes, which is an important characteristic of the RF model.

XGBoost (Chen and Guestrin, 2016) is also a regression tree-based ensemble-learning model. However, instead of averaging multiple individual trees, XGBoost builds comple-

mentary trees for prediction, which allows for very different functional relationships compared to random forests. Differently from RF models, XGBoost depends on multiple hyperparameters, which makes it harder and more computationally expensive to apply (Bentéjac et al., 2021). XGBoost showed excellent performances in a range of machine-learning competitions (Nielsen, 2016) and also in hydrological time-series applications (e.g. Ni et al., 2020; Gauch et al., 2019; Ibrahem Ahmed Osman et al., 2021), which makes it a promising candidate model for this study. To the authors' knowledge, XGBoost has not been applied for river water temperature predictions yet. However, results from short-term water quality parameter predictions, which also include water temperature, show promising performances (Lu and Ma, 2020; Joslyn, 2018).

FNNs (White and Rosenblatt, 1963) are the first and simplest type of neural networks. FNNs have already been applied in numerous stream water temperature prediction studies, which range from simple one hidden layer models (e.g. Risley et al., 2003; Bélanger et al., 2005; Chenard and Caissie, 2008; McKenna et al., 2010; Hadzima-Nyarko et al., 2014; Rabi et al., 2015; Zhu et al., 2018; Temizyurek and Dadaser-Celik, 2018) to multiple hidden-layer models with a hyperparameter optimization (Sahoo et al., 2009) and to more complex FNN (hybrid) architectures and ensembles of FNNs (e.g. DeWeber and Wagner, 2014; Piotrowski et al., 2015; Abba et al., 2017; Graf et al., 2019; Zhu et al., 2019a, b). While FNNs are very flexible and a one-layer FNN can theoretically approximate any functions (Pinkus, 1999), they are prone to overfitting and dependent on input data scaling and an adequate choice of hyperparameters.

In contrast to FNNs, recurrent neural networks (RNNs) are networks developed specifically to process sequences of inputs. This is achieved by introducing internal hidden states allowing one to model long-term dependencies in data at the cost of higher computational complexity (Hochreiter and Schmidhuber, 1997) and longer run times. Furthermore, gaps in the observation time series can reduce the number of usable data points significantly, as a certain number of previous time steps are needed for prediction. While there are many different types of RNNs, we focused on the two most widely known, the long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and the gated recurrent unit (GRU) (Cho et al., 2014). To the authors' knowledge, RNNs have been used in one study by Stajkowski et al. (2020), in which a LSTM in combination with a genetic algorithm hyperparameter optimization was used to forecast hourly urban river water temperature. However, LSTMs have recently been applied in a wide range of hydrological studies and showed promising results for time-series prediction tasks (e.g. Kratzert et al., 2018, 2019; Xiang et al., 2020; Li et al., 2020).

To make findings comparable with other studies investigating this approach, we apply two benchmark models as the baseline: linear regression and air2stream (Toffolon and Piccolroaz, 2015). Linear regression models are widely used for river water temperature studies. While earlier studies used mainly air temperature as a regressor to predict river water temperature (e.g. Smith, 1981; Crisp and Howson, 1982; Mackey and Berrie, 1991; Stefan and Preud'homme, 1993), more recent publications use a wider range of input variables or some modification to the standard linear regression model (e.g. Caldwell et al., 2013; Li et al., 2014; Segura et al., 2015; Arismendi et al., 2014; Naresh and Rehana, 2017; Jackson et al., 2018; Trinh et al., 2019; Piotrowski and Napiorkowski, 2019). air2stream is a hybrid model for predicting river water temperature, which combines a physically based structure with a stochastic parameter calibration. It was already applied in multiple studies over a range of catchments and generally had an improved performance compared to linear regression and other machine-learning models (e.g. Piccolroaz et al., 2016; Yang and Peterson, 2017; Piotrowski and Napiorkowski, 2018; Zhu et al., 2019d; Piotrowski and Napiorkowski, 2019; Tavares et al., 2020).

Most studies mainly use air temperature and discharge as inputs for water temperature prediction (e.g. Piccolroaz et al., 2016; Naresh and Rehana, 2017; Sohrabi et al., 2017), while others use additional information from precipitation (e.g. Caldwell et al., 2013) and/or solar radiation (e.g. Sahoo et al., 2009). Additionally, air temperature can either be included as mean, maximum or minimum daily temperature (e.g. Piotrowski et al., 2015). To further investigate which meteorological and hydrological inputs are important and necessary for water temperature prediction, we here use multiple sets of input data and compare their outcome. Especially knowing how simple models with few data inputs perform in comparison with more complex input combinations can give insight into how to plan applications of water temperature modelling for a range of purposes.

Machine-learning models are generally parameterized by a set of hyperparameters that have to be chosen by the user to maximize performance of the model. The term "hyperparameters" refers to any model parameter that is chosen before training the model (e.g. neural network structure). Depending on the model, hyperparameters can have a large impact on model performance (Claesen and De Moor, 2015) but are still most often chosen by rules of thumb (Hinton et al., 2012; Hsu et al., 2003) or by testing sets of hyperparameters on a predefined grid (Pedregosa et al., 2011). In this study we apply a hyperparameter optimization using the Bayesian optimization method (Kushner, 1964; Zhilinskas, 1975; Močkus, 1975; Močkus et al., 1978; Močkus, 1989) to minimize the possibility of using unsuitable hyperparameters for the applied models and to investigate the spread in performance depending on the chosen hyperparameters.

This publication presents a thorough investigation of models, input data and model training characteristics for daily stream water temperature prediction. It consists of the application of six types of machine-learning models on a range of different catchments using multiple sets of data inputs.

The present work's originality includes (i) application of a range of ML models for water temperature prediction, (ii) the use of different climatic variables and combinations of these as model inputs, and (iii) the use of Bayesian optimization to objectively estimate hyperparameters of the applied ML models. The resulting performance of all models is compared to two widely applied benchmark models to make the presented results comparable. Finally, all methods and models are incorporated into an open-source R library to make theses approaches available for researchers and industries.

2 Methods

2.1 Study sites and data

In Austria there are 210 river water temperature measurement stations available, sometimes with 30+ years of data. This large number of available data in Austria are highly advantageous for developing new modelling concepts. Additionally, a wide range of catchments with different physiographic properties are available, ranging from high-alpine, glacier-dominated catchments to lowland rivers, with meandering characteristics.

For this study, 10 catchments with a wide range of physiographic characteristics, human impacts (e.g. hydropower, river regulation) and available observation period length were selected. Including study sites with diverse properties allows for validation of the applicability and performance of the introduced modelling approach. The catchments are situated in Austria, Switzerland and Germany, with outlets located in the Austrian Alps or adjacent flatlands. All catchments and gauging stations are shown in Fig. 1, and their main characteristics are summarized in Table 1.

The gauging stations are operated by the Austrian Hydrographical Service (HZB) and measure discharge (Q) in 15 min intervals and water temperature (T_w) in a range of different time intervals (daily mean – 1 min). The temperature sensors are situated in a way that complete mixing can be assumed, e.g. after a bottom ramp. Consequently, the measured water temperature should reflect the water temperature of the given cross section.

The meteorological data used in this study are daily mean air temperature (T_a), daily max air temperature (T_{max}), daily min air temperature (T_{min}), precipitation sum (P) and global radiation (GL). T_a, T_{max}, T_{min} and P were available from the SPARTACUS project (Hiebl and Frei, 2016, 2018) on a 1×1 km grid from 1961 onward. The SPARTACUS data were generated by using observations and external drift kriging to create continuous maps. GL data were available from the INCA analysis (Integrated Nowcasting through Comprehensive Analysis) (Haiden et al., 2011, 2014) from 2007 onward. The INCA analysis used numerical weather simulations in combination with observations and topographic information to provide meteorological analysis and nowcasting fields of several meteorological parameters on a 1×1 km grid in 15–60 min time steps. For the presented study, the 15 min INCA GL analysis fields were aggregated to daily means. The catchment means of all variables are shown in Table 1. By using high-resolution spatially distributed meteorological data as the basis for our inputs, we aim to better represent the main drivers of water temperature changes in the catchments. Similar data sets are available for other parts of the world, e.g. globally (Hersbach et al., 2020), for North America (Thornton et al., 2020; Werner et al., 2019), for Europe (Brinckmann et al., 2016; Razafimaharo et al., 2020) and for China (He et al., 2020).

2.2 Data preprocessing

The applied data preprocessing consists of aggregation of gridded data, feature engineering (i.e. deriving new features from existing inputs) and splitting the data into multiple sets of input variables. Since river water temperature is largely controlled by processes within the catchment, variables with an integral effect on water temperature over the catchment (i.e. T_a, T_{max}, T_{min}, P and GL) are aggregated to catchment means.

Computing additional features from a given data set (i.e. feature engineering) and therefore having additional data representation can significantly improve the performance of machine-learning models (Bengio et al., 2013). Previous studies have shown that especially time information is important for water temperature prediction. This includes time expressed as day of year (e.g. Hadzima-Nyarko et al., 2014; Li et al., 2014; Jackson et al., 2018; Zhu et al., 2018, 2019c, d), the content of the Gregorian calendar (i.e. year, month, day) (Zhu et al., 2019b), or expressed as the declination of the Sun (Piotrowski et al., 2015), which is a function of the day of the year. Nevertheless, using cyclical features like day of the year as an integer variable will most likely reduce model performance, since days 1 and 365 are as close together as 1 and 2. To translate time information into a more suitable format, we chose to transform months and days of months into trapezoidal fuzzy sets, called fuzzy months. Similar to dummy encoding, the values of a fuzzy month are between 0 and 1. They are equal to 1 on the 15th of the corresponding month and linearly decreasing each day until they are zero on the 15th day of the previous and following months. Therefore, the values of two adjacent months will be around 0.5 at the turn of the month. By encoding the categorical variable "month" into these 12 new fuzzy variables, it should be possible to represent time of the year influence more smoothly, as no jumps in monthly influence are possible. Initial test showed that the advantage of this representation exceeds the disadvantage of using 12 variables instead of 1 or 2. A similar approach for encoding time variables was already applied by Shank et al. (2008).

Besides time variables, a previous study by Webb et al. (2003) showed that lag information is significantly associ-

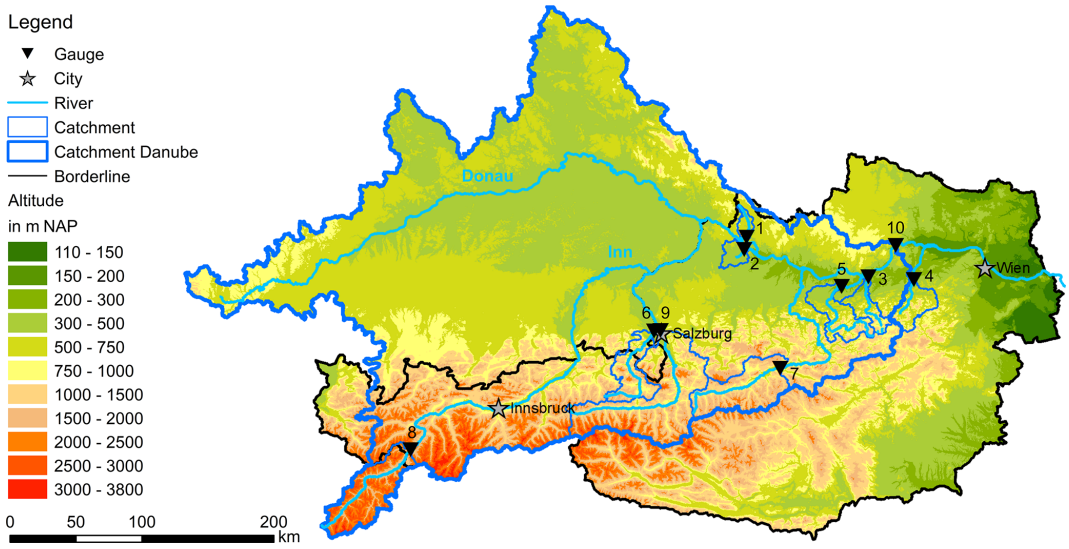

Figure 1. Study sites in Austria, Germany and Switzerland. All gauging station IDs refer to the IDs in Table 1. Delineation sources: Bayrisches Landesamt für Umwelt; HAO: Hydrological Atlas of Austria digHAO (BMLFUW, 2007).

Table 1. Overview of study catchment characteristics, including means of meteorological values of catchment means (T_w, Q, T_a, P, GL), catchment areas (Area), mean catchment elevations (Elevation), catchment glacier and perpetual snow cover (Glacier), available data time periods (Time period) and number of years with data (Years). IDs refer to the IDs used in Fig. 1. The percentage of glacier and perpetual snow cover was computed from the CORINE Land Cover data 2012 and the mean catchment elevation from the EU-DEM v1.1 digital elevation model with 25 × 25 m resolution.

ID	Catchment	Gauging station	Time period	Years	Area (km^2)	Elevation (m NAP)	Glacier (%)	T_w (°C)	Q (m^3/s)	T_a (°C)	P (mm)	GL (W/m^2)
1	Kleine Mühl	Obermühl	2002–2015	14.0	200.2	602	0	8.87	3.12	8.71	2.73	135
2	Aschach	Kropfmühle	2004–2015	11.9	312.2	435	0	10.78	3.80	9.57	2.50	136
3	Erlauf	Niederndorf	1980–2015	35.3	604.9	661	0	9.42	15.27	7.99	3.59	127
4	Traisen	Windpassing	1998–2015	17.7	733.3	697	0	9.83	14.88	8.47	3.33	131
5	Ybbs	Greimpersdorf	1981–2015	34.7	1 116.6	691	0	9.87	31.50	7.97	3.77	127
6	Saalach	Siezenheim	2000–2015	16.0	1 139.1	1196	0	8.50	39.04	6.72	4.60	135
7	Enns	Liezen	2006–2015	10.0	2 116.2	1419	0	1.19	67.56	5.62	3.60	137
8	Inn	Kajetansbrücke	1997–2015	18.8	2 162.0	2244	2.8	6.00	59.26	0.12	2.56	153
9	Salzach	Salzburg	1977–2015	39.0	4 425.7	1475	1.4	7.63	178.11	5.22	4.16	136
10	Danube	Kienstock	2005–2015	11.0	95 970.0	827	0.4	10.77	1 798.31	10.05	2.13	131

ated with water temperature and can improve model performance. The lag period of 4 d was chosen based on an initial data analysis that included (i) assessing partial autocorrelation plots of water temperatures, (ii) testing for significance of lags in linear regression models, and (iii) checking variable importance of lags in a random forest model. Therefore, to allow for information of previous days to be used by the models, the lags of all variables for the 4 previous days are computed and used as additional features.

Using these input variables, six experiments with different sets of inputs considering different levels of data availability are defined. The variable compositions of all experiments are shown in Table 2. All features include four lags, and each experiment also includes fuzzy months as inputs. Experiment 0 (T_{mean}) acts as another simple benchmark in which only daily mean air temperature and fuzzy months are used for predictions. Experiment 1 (T) will be able to show the benefit of including T_{max} and T_{min}. Experiments 2–4 consist of combinations of experiment 1 with precipitation and discharge data. Experiments 5–6 include combinations with GL and therefore include only data of the time period 2007–2015 in which GL data were available.

2.3 Benchmark models

Two widely applied models for stream water temperature prediction are used as a benchmark for all models tested in this study: multiple linear regression (LM) models and air2stream (Toffolon and Piccolroaz, 2015). By including these two models, it will be possible to compare this study's

Table 2. Overview of available meteorological and hydrological variables and the composition of the different input data set experiments. If an input variable is used in a data set, the lags for the 4 previous days are included as well. Additionally to the shown variables, all experiments use fuzzy months as input.

Experiment	T_a	T_{max}	T_{min}	P	Q	GL
0 (T_{mean})	X					
1 (T)	X	X	X			
2 (TP)	X	X	X	X		
3 (TQ)	X	X	X		X	
4 (TQP)	X	X	X	X	X	
5 (TPGL)	X	X	X	X		X
6 (TQPGL)	X	X	X	X	X	X

results to a wider range of previous studies, which investigated models for stream water temperature prediction.

2.3.1 Linear regression

Linear regression models are widely used for river water temperature studies. Earlier studies used mainly air temperature as a regressor to predict river water temperature (e.g. Smith, 1981; Crisp and Howson, 1982; Mackey and Berrie, 1991; Stefan and Preud'homme, 1993). More recent publications use a wider range of input variables or some modification to the standard linear regression model (e.g. Caldwell et al., 2013; Li et al., 2014; Segura et al., 2015; Arismendi et al., 2014; Naresh and Rehana, 2017; Jackson et al., 2018; Trinh et al., 2019; Piotrowski and Napiorkowski, 2019).

The ordinary least-square linear regression model is defined as

$$\boldsymbol{Y} = \beta \mathbf{X} + \boldsymbol{\epsilon}, \tag{1}$$

where $\boldsymbol{Y}$ denotes the vector of the dependent variable (river water temperature), $\mathbf{X}$ denotes the matrix of independent variables (e.g. daily mean air temperature, global radiation), β denotes the vector of model coefficients and $\boldsymbol{\epsilon}$ denotes the error term. $\boldsymbol{\epsilon}$ is assumed to be normal distributed with a diagonal covariance matrix. The estimates for the model coefficients and the dependent variable, which minimize the sum of squared errors, are given by

$$\hat{\boldsymbol{Y}} = \hat{\beta} \mathbf{X}, \tag{2}$$

$$\hat{\beta} = (\mathbf{X}'\mathbf{X})^{-1}\mathbf{X}'\boldsymbol{Y}, \tag{3}$$

where $\hat{\boldsymbol{Y}}$ and $\hat{\beta}$ represent estimated values. The linear regression model applied in this study includes an intercept and the variables T_a and Q as independent variables to predict T_w.

2.3.2 air2stream

air2stream (Toffolon and Piccolroaz, 2015) is a hybrid model for predicting river water temperature, which combines a physically based structure with a stochastic parameter calibration. It was already applied in multiple studies over a range of catchments and generally had an improved performance compared to linear regression models (e.g. Piccolroaz et al., 2016; Yang and Peterson, 2017; Piotrowski and Napiorkowski, 2018; Zhu et al., 2019d; Piotrowski and Napiorkowski, 2019; Tavares et al., 2020). air2stream uses the inputs T_a and Q and was derived from simplified physical relationships expressed as ordinary differential equations for heat-budged processes. Due to this simplification, it may be applied like a data-driven model, which depends on parameter calibration. The eight-parameter version of air2stream is defined as

$$\frac{\mathrm{d}T_w}{\mathrm{d}t} = \frac{1}{\theta^{a_4}}[a_1 + a_2 T_a - a_3 T_w + \theta\left(a_5 + a_6 \cos\left(2\pi\left(\frac{t}{t_y} - a_7\right)\right) - a_8 T_w\right)], \tag{4}$$

where t is the time in days, t_y is the number of days per year, $\bar{Q}$ is the mean discharge, $\theta = Q/\bar{Q}$ is the dimensionless discharge and $a_{1,\ldots,8}$ are the model parameters. This differential equation is numerically integrated at each time step using the Crank–Nicolson numerical scheme (Crank and Nicolson, 1947) and the model parameters are calibrated using particle swarm optimization (Kennedy and Eberhart, 1995).

2.4 Machine-learning models

In this study we compare six different machine-learning models: step-wise linear regression (step-LM), RF, XGBoost, FNNs and two RNNs – the long short-term network (RNN-LSTM) and the gated recurrent unit (RNN-GRU). An overview and simple depiction of the models are shown in Fig. 2.

2.4.1 Step-wise linear regression

Step-wise linear regression models combine an iterative variable selection procedure with linear regression models. The step-wise variable selection starts at an initial model (e.g. all variables) and removes or adds at each iteration based on a prespecified criterion. We applied the step-wise variable selection starting with an initial model including all variables and using the Akaike information criterion (AIC) (Akaike H, 1973). The AIC for a linear regression model is given by

$$\mathrm{AIC} = n \times \ln\left(\frac{\sum_{i=1}^{n}(\boldsymbol{Y}_i - \hat{\boldsymbol{Y}}_i)^2}{n}\right) + 2k, \tag{5}$$

where n is the number of samples, ln() the natural logarithm, $\boldsymbol{Y}$ and $\hat{\mathbf{Y}}$ the observed and predicted water temperatures and k the number of selected input variables. The step-wise variable selection is iteratively applied until AIC is at a minimum. Additionally to the variables given in Sect. 2.2, interaction terms between T_a, Q, GL and P are included.

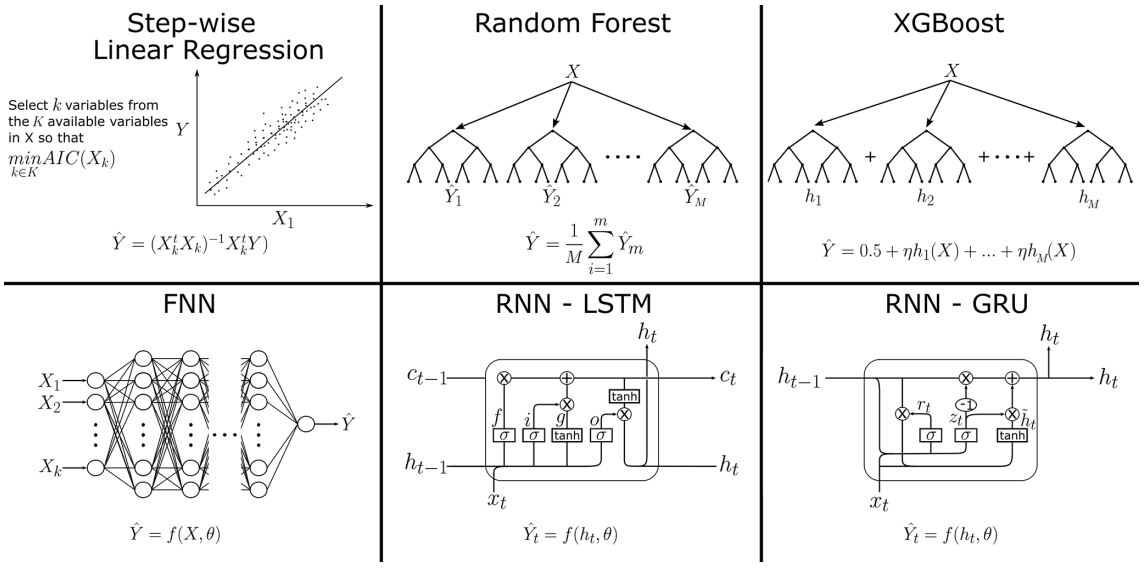

Figure 2. Overview of the applied models with $\hat{Y}$ denoting estimated water temperatures and $\mathbf{X}$ the matrix of observed variables. $\hat{Y}_{1,...,M}$ are the predictions from individual RF trees. $h_{1,\ ...,M}$ are the predicted residuals from individual XGBoost trees. $f(X, \theta)$ denotes a mapping from a FNN with the parameters θ. For a given time step, h_t denotes the hidden internal state of a RNN cell, c_t the internal cell state of a LSTM cell and $f(h_t, \theta)$ the mapping from a RNN with the parameters θ. RNNs consist of a cascade of cells, each feeding their internal states into the next cell, finally resulting in a single feed-forward layer estimating $\hat{Y}$ from h_t.

2.4.2 Random forest

The RF model (Breiman, 2001) is an ensemble-learning model based on the idea of bagging (bootstrap aggregating) (Breiman, 1996). Bagging predictors average multiple model predictions, where each model is trained on a bootstrapped sample instead of the full observed sample. This randomness introduced by bootstrapping increases the model's ability to generalize and to produce stable prediction results.

RF models are bagging predictors which use classification and regression trees (CARTs) as a base learner. RF CARTs recursively apply binary splits to the data to minimize entropy in the tree nodes. This is done until each node reaches a minimum node size or a previously defined maximum tree depth is reached. Breiman (2001) showed that adding further randomness to the bagging method improves prediction accuracy. In random forests this is achieved by only selecting a random subset of available variables for the split at each node. The estimate for the dependent variable is given by

$$\hat{\boldsymbol{Y}} = \frac{1}{M} \sum_{m=1}^{M} f_i(\mathbf{X}), \tag{6}$$

where f_m denotes a single fitted CART, M the number of used CARTs, $\mathbf{X}$ the matrix of regressors and $\hat{\boldsymbol{Y}}$ the vector of estimated water temperature. A simplified depiction of the RF algorithm is shown in Fig. 2. RF has two important hyperparameters: the number of predictors sampled at each node (mtry) and the minimum size of nodes (min node size). The number of trees was chosen to be constant with 500 trees.

2.4.3 XGBoost

XGBoost (Chen and Guestrin, 2016) is a tree-boosting algorithm that was developed based on the already existing concept of boosting, which was further enhanced to increase efficiency, scalability and reduced overfitting. Similarly to bagging, boosting methods combine the prediction of an ensemble of weak learners to improve prediction accuracy. However, while bagging ensemble members are trained in parallel, boosting iteratively trains new ensemble members and adds them to the existing ensemble. Boosting was first introduced by Schapire (1990) and then widely applied after the introduction of the Adaboost algorithm (Freund and Schapire, 1995). Friedman (2001) further enhanced boosting by adding gradient decent optimization for the boosting iterations. This resulted in the development of gradient tree boosting (Friedman, 2002), which uses CART as weak learners.

XGBoost is an implementation of gradient tree boosting with further enhancements in the form of added stochasticity and regularization. The XGBoost estimated for the independent variable is given by

$$\hat{\boldsymbol{Y}} = 0.5 + \sum_{m=1}^{M} \eta f_m(\mathbf{X}), \tag{7}$$

where $f_1, \ldots f_M$ is a sequence of CARTs, $\eta \in [0, 1]$ is the learning rate, M is the number of used CARTs, $\mathbf{X}$ is the matrix of input features and $\hat{\boldsymbol{Y}}$ is the vector of estimated water temperatures. The mth tree is trained to predict the residu-

als of a model of the form given in Eq. (7), which uses the previous $m-1$ CARTs. The loss function used to train each tree includes a regularization term to prevent overfitting. Additionally, overfitting is reduced by only allowing a random subset of samples and variables to be used for constructing trees and tree nodes at each iteration. A simplified depiction of the XGBoost algorithm is shown in Fig. 2.

XGBoost has multiple important hyperparameters that have to be chosen before fitting the model: the maximum number of iterations (nrounds), the learning rate (η), the maximum depth of a tree (max depth), the minimum sum of instance weight needed in a child (min node size), the ratio of random subsamples used for growing a tree (subsample) and the random fraction of variables used for growing a tree (colsample bytree).

2.4.4 Feed-forward neural networks

FNNs (White and Rosenblatt, 1963) are the first and simplest type of neural networks. FNNs consist of multiple layers of nodes, where each node is connected to all nodes of the previous and following layers. A node applies linear and non-linear (activation) functions to its input to produce an output. The general structure of a FNN is shown in Fig. 2.

Piotrowski et al. (2020) showed that adding dropout (Hinton et al., 2012; Srivastava et al., 2014; Baldi and Sadowski, 2014) to FNNs for stream water temperature prediction improved performance of single-layer FNNs. Dropout refers to randomly dropping nodes from a layer during training, which can prevent overfitting and potentially improve generalization. We added a dropout to every FNN layer and defined the dropout rate as a hyperparameter, which can be zero and therefore also allow for model structures without dropout.

While the parameters (θ) of the linear function get optimized using backpropagation (Rumelhart et al., 1986), FNNs have multiple hyperparameters that need to be predefined before training. These hyperparameters include the activation functions, the number of layers, the number of nodes per layer and the dropout ratio. After initial tests, in which a large set of different activation functions was applied, we chose the scaled exponential linear unit (SELU) activation function (Klambauer et al., 2017) for all nodes in the network. SELU includes a normalization, which enhances convergence and avoids both vanishing and exploding gradients during backpropagation. The other hyperparameters are optimized as described in Sect. 2.5.

The hyperparameter optimization approach presented here differs from previous studies, which generally assume a set of a fixed number of layers and/or nodes per layer that were derived by a trial-and-error approach (e.g. Bélanger et al., 2005; Hadzima-Nyarko et al., 2014; Piotrowski et al., 2015; Zhu et al., 2018, 2019d).

2.4.5 Recurrent neural networks

In contrast to FNNs, RNNs are able to process sequences of inputs. This is achieved by having internal (hidden) states. While there are many different types of RNNs, we focused on the two most widely known, the long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) and the gated recurrent unit (GRU) (Cho et al., 2014). Each layer of an RNN consists of a sequence of cells that share a common set of weights. The cells of both LSTM and GRU are shown in Fig. 2 and are described in Appendices A1 and A2. A single RNN cell consists of multiple gates, which refers to the nodes of a cell where non-linear transformations are applied to the inputs and states. The main difference between LSTM and GRU cells is their number of gates and internal states, where LSTMs are more complex (two internal states and three gates) than GRUs (one internal state and two gates). While in some cases GRUs outperform LSTMs, there is no clear rule of when to use one or the other (Yazidi et al., 2020). Each RNN contains a FNN layer with a single node at its end, which is used to compute the predicted values from the hidden states of the last time step ($\mathbf{h}_T$). Both types of RNNs have the same set of hyperparameters that need to be specified before training the model: the number of used RNN layers, the number of units per layer, the numbers of time steps, the dropout ratio, and the batch size.

Due to their internal states and the usage of multiple time steps for prediction, it can be assumed that RNNs do not need time information (here in the form of fuzzy months) for predicting water temperature data. To test this assumption, both RNN variants are also trained without fuzzy months to check the influence of these additional variables on model performance. Being able to achieve equally good results without fuzzy months would reduce training time considerably due to decreasing the input data by 12 dimensions (columns).

2.5 Bayesian hyperparameter optimization

Choosing adequate hyperparameters for a machine-learning model can have a large impact on its performance. Therefore, it is necessary to apply some sort of optimization procedure. While it might be possible to apply a grid search over the range of all possible parameter value combinations for a small set of hyperparameters, it is usually not feasible due to available computational resources. For that reason, we chose to optimize the hyperparameters of nearly all machine-learning models in this study with the Bayesian optimization method. Only random forest with three hyperparameters is optimized using a grid search. Step-wise linear regression does not have hyperparameters that need optimization.

Bayesian optimization is a global optimization method for blackbox functions (i.e. lacks known structure and is derivative-free) that is often applied in cases where the objective function is computationally expensive to evaluate. It originates from work by Kushner (1964), Zhilinskas (1975),

Močkus (1975), Močkus et al. (1978), and Močkus (1989) and was later popularized by Jones et al. (1998). It became especially well known for being suitable for optimizing machine-learning hyperparameters after a study by Snoek et al. (2012).

Bayesian optimization consists of two parts: a method for statistical inference and an acquisition function for deciding the next sample point. The method for statistical inference is usually a Gaussian process (GP) which provides an estimated posterior distribution at each iteration that is an estimate for the function that should be optimized. The acquisition function is used to find the next point to evaluate during each optimization step and was chosen to be the upper confidence bound (UCB) (Srinivas et al., 2009) in this study. In summary, Bayesian optimization constructs a surrogate model at each iteration during optimization to choose a suitable next point. The hyperparameters of all optimized models and their chosen bounds are given in Appendix A3.

2.6 Evaluation metrics

The objective function for all models and the hyperparameter optimization is the mean squared error (MSE):

$$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\mathbf{y}_i - \hat{\mathbf{y}}_i)^2, \tag{8}$$

where n is the number of samples (d) and y_i the observed and $\hat{y}_i$ the predicted water temperatures. To compare the performance of different models, the root mean squared error RMSE and the mean absolute error MAE are used:

$$\mathrm{RMSE} = \sqrt{\mathrm{MSE}}, \tag{9}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\mathbf{y}_i - \hat{\mathbf{y}}_i|. \tag{10}$$

2.7 Experimental setup

To be able to objectively compare all applied models, the available data sets are split into two parts: the first 80 % of the time series were used for training/validation and the last 20 % were used for testing. We deliberately did not choose a random split, because predicting water temperatures for a future time period is a more adequate test for models. This is especially relevant for water temperature, which is characterized by non-stationarity due to climate change (Van Vliet et al., 2013). The training/validation and test time series are compared to assess the difference of water temperature distribution of all catchments.

The step-wise linear regression model, RF and XGBoost are optimized using cross-validation (CV). Two kinds of CV are applied: a five times repeated 10-fold CV and a time-series CV. While the 10-fold CV splits the data randomly, the time-series CV gradually adds data to an initial part of the time series while evaluating the performance of each step. The time-series CV started with an initial window of 730 d for training the following 90 d for validation. The training set is increased by 90 d at each different cross-validation set until the full time series except for the last 90 d was used. Therefore, instead of 10 folds, the number of folds for the time-series CV depends on the time-series length.

Due to computational and time constraints, hyperparameter optimization for all neural networks was done by using a training/validation split with 60 % data for training and 20 % data for validation. This allows model validation performance estimation by training a model once, while a 5 times repeated 10-fold CV would require training a model 50 times. Furthermore, the training/validation split is the standard way of training neural networks for real-world applications.

Bayesian hyperparameter optimization consists of 20 random parameter samples and 40 iterations of optimization. The data inputs for all neural networks were standardized by subtracting the mean and dividing by the standard deviation of the training data. The optimized neural network hyperparameter sets are used to create five independently trained models, from which an ensemble for prediction is created by taking the average of all five prediction results. Using ensembles of networks is a way to significantly increase a neural network's ability to generalize and is an often-applied approach which was first introduced by the work of Hansen and Salamon (1990). In addition, early stopping with patience = 5 was applied to all neural networks to avoid overfitting.

The best-performing model for each model type and experiment is chosen using the validation RMSE. Test RMSE and MAE results are only compared after choosing the models with minimum validation RMSE. Consequently, it might be possible that some models have a superior test performance but are not chosen as the best-performing model for a specific model type and/or experiment. This should reflect a real-world application, where test data act as a previously unknown future time series.

Table 3 gives an overview of all time periods and the hyperparameter optimization details. All models are trained using the training/validation period data and either applied CV or a training/validation split. Models with hyperparameters are trained multiple times during hyperparameter optimization. The fully trained models are then applied in the test time period to produce comparable out-of-sample results. The eight air2stream hyperparameters are optimized using the particle swarm optimization with 500 iterations, 500 particles, cognitive and social learning factors set to 2 and inertia max and min set to 0.9 and 0.4. All models were run on the Vienna Scientific Cluster, where each run had access to two Intel Xeon E5-2650v2, 2.6 GHz, eight-core CPUs and 65 GB RAM.

Table 3. Overview of the different modelling time periods and hyperparameter optimization details, including information about cross-validation (CV), the number of hyperparameters (Hyperparameters) and the number of iterations of the Bayesian hyperparameter optimization (Iterations).

Catchment	Training/validation period	Test period	Model	CV	Hyperparameters	Iterations
Kleine Mühl	2002–2012	2013–2015	LM	no	0	0
Aschach	2004–2012	2013–2015	air2stream	no	8	500
Erlauf	1980–2007	2008–2015	step-LM	yes	0	0
Traisen	1998–2011	2012–2015	RF	yes	2	60
Ybbs	1981–2007	2008–2015	XGBoost	yes	6	60
Saalach	2000–2011	2012–2015	FNN	no	4	60
Enns	2006–2013	2014–2015	RNN-GRU	no	5	60
Inn	1997–2011	2012–2015	RNN-LSTM	no	5	60
Salzach	1977–2007	2008–2015				
Danube	2005–2012	2013–2015				

2.8 Statistical tests

The Kruskal–Wallis test (Kruskal and Wallis, 1952) was used to test for differences in overall model performances, different training/model characteristics and different data inputs. Dunn's test for multiple comparison (Dunn, 1964) was used for pair-wise comparisons between model performances. To investigate the association of model types, experiments and catchments with test RMSE, an ordinary least-square linear regression model was used. Level of significance was set to $p = 0.05$ for all statistical tests.

2.9 Open-source R package

All preprocessing steps and models were implemented in the open-source R package *wateRtemp*, which is available under https://www.github.com/MoritzFeigl/wateRtemp (last access: 25 April 2021) or from Feigl (2021a). This provides easily applicable modelling tools for the water temperature community and allows all results of this study to be replicated. All programming was done in R (R Core Team, 2020), where the model development relied heavily on Caret (Kuhn, 2020), xgboost (Chen et al., 2020) and TensorFlow (Allaire and Tang, 2020) and the visualizations on ggplot2 (Wickham, 2016).

3 Results

3.1 Time period characteristics

Due to climate change, both air temperatures and water temperatures are steadily increasing (Mohseni and Stefan, 1999; Pedersen and Sand-Jensen, 2007; Harvey et al., 2011; Kędra, 2020). This is clearly visible when comparing the change in number of extreme warm days and the increase in mean water temperature in all studied catchments with time. For this we compared the training/validation and test time data in each catchment. Since test data consist of the last 20 % of the overall data, the exact length of these time series is dependent on the catchment but is always a subset of the years 2008–2015. We can observe an increase of 138 % of the median number of days with water temperature above the 90 % quantile between training/validation and test time period in all catchments. This increase ranges from 69 % or from 32 to 54 d, in the Danube catchment and up to 285 %, or from 26 to 100 d, in the Salzach catchment. This change is even more pronounced when comparing the last year of test data (2015) to all other available years, where the median number of days with water temperatures above the 90 % quantile (computed for the overall time series) of all catchments increases by 273 %. Figure 3 shows the corresponding boxplots of days with stream temperature above the 90 % quantile for each catchment in training/validation and in test time period. A similar pattern can be observed in the changes in mean yearly stream temperatures. The median increase in mean yearly water temperature of all catchments is 0.48 °C when comparing training/validation with test time period and 0.77 °C when comparing the last year of the test period (2015) with all other years. Since the test period is, as shown here regarding extremes, different from the training/validation period, the models are also, at least to some extent, tested on how they perform under instationary conditions. This is a test where environmental models often fail (e.g. Kling et al., 2015).

3.2 Overall performance comparison

Table 4 gives an overview of the best-applied machine-learning models and the two benchmark models LM and air2stream. This table compares the RMSE and MAE performances of all models; additional performance metrics are shown in Table A1. The mean test RMSE of LM is 1.55 °C with an overall range of [1.25, 2.15] °C, while air2stream has a mean test RMSE of 0.98 °C with an overall range of [0.74, 1.17] °C. The performance results for each catchment show that air2stream always outperformed LM and consequently

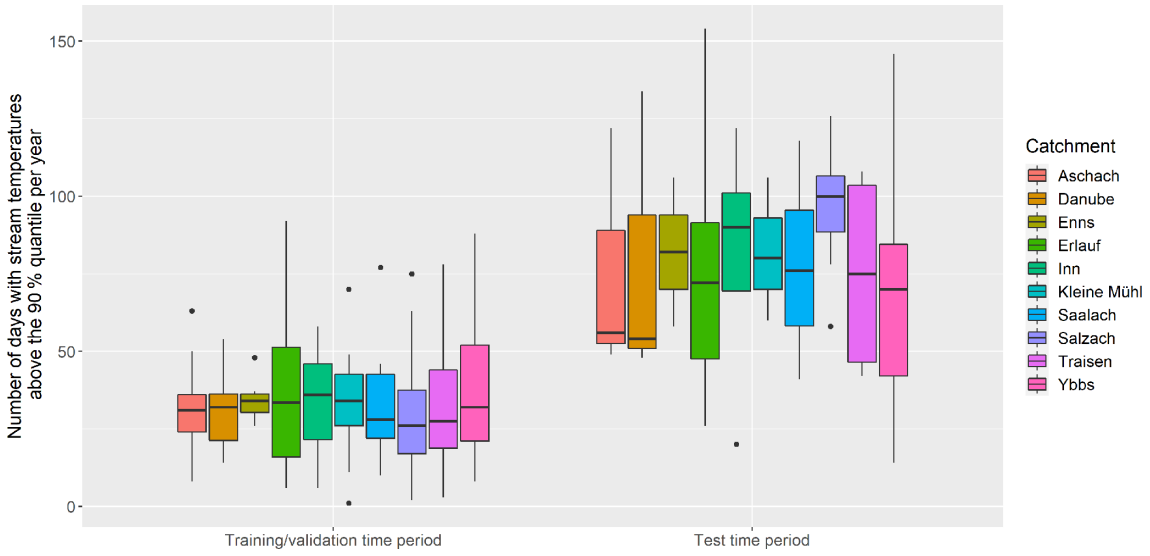

Figure 3. Boxplots showing the distribution of numbers of days with stream temperatures above the 90 % quantile per year for all study catchments for the training/validation and the test time period, where the test time period consists of the last 20 % of data in each catchment. The 90 % quantile values were estimated using the full time series for each catchment.

results in a significant lower test RMSE ($p < 0.001$). The mean test RMSE of the best machine-learning models per catchment is 0.55 °C with an overall range of [0.42, 0.82] °C and always outperformed the air2stream benchmark. Based on the RMSE means, the highest performing ML model is 64 % and 43 % better, compared to LM and air2stream. This results in a significantly lower test RMSE of the tested machine-learning models compared to the air2stream benchmark ($p < 0.001$).

Both XGBoost and FNN were found to be the best-performing model in 4 of 10 analysed catchments each. RF was the best-performing model in the Salzach catchment and RNN-LSTM in the Danube catchment. Step-LM and RNN-GRU did not outperform the other models in any of the study catchments. Experiment 3, which only includes air temperature and discharge input features, resulted in the best-performing model in four catchments. Experiment 6, which included all available input features, also produced the best-performing model in four catchments. Experiment 4, which includes air temperature, discharge and precipitation input features, performed best in two catchments.

Figure 4 shows the results of all models, catchments and experiment combinations. The boxplots in Fig. 4a show the range of model performances depending on the model type. Kruskal–Wallis test results show no significant difference ($p = 0.11$) of the test RMSE of different model types. Figure 4b shows boxplots of model performance for all experiments. Kruskal–Wallis test results show a highly significant difference of test RMSE of the different experiments ($p < 10^{-14}$). The results in Fig. 4b show an increase in median performance with an increasing number of input features until experiment 4 (TQP). When adding global radiation as an additional input parameter, the median performance does not increase further. This could be explained by a reduced time-series length of experiments 5 (TPGL) and 6 (TQPGL), since global radiation was only available from 2007 on. A comparison between experiments with equal time-series lengths (experiments 0–4 and experiments 5–6) also indicates that runoff information improves the modelling performance.

Figure 4c illustrates the RMSE performance results for each catchment shown as boxplots. A corresponding figure of the MAE results is shown in Fig. A1. The boxplots are overlayed with scatter-plot points adding an overview of the individual performance of each model and experiment combination. To account for a better visibility, the scatter-plot points are shifted in horizontal direction randomly. The difference in performance between catchments is clearly visible and ranges from a median RMSE of around 0.93 °C in catchments Kleine Mühl and Aschach down to a median RMSE of 0.58 °C in the Inn catchment.

Figure 4c also includes the air2stream benchmark performance shown as a grey line for each catchment. Nearly all tested experiments and model combinations showed improved performance compared to the air2stream benchmark. Only in five catchments could we observe models in combination with experiments 0, 1, and 5 and one time with experiment 6 that predicted worse than air2stream. There are surprisingly few models considering the fact that experiments 0, 1, 5 and 6 are heavily constrained due to the amount of information that is available for prediction. Experiments 0 and 1, which only use air temperature, are still able to improve predictions compared to air2stream for all model types in seven catchments. Similarly, experiments 5 and 6 with only 6 years

Table 4. Overview of model performance of the best machine-learning model for each catchment and the two reference models. The best-performing model results in each catchment are shown in bold font. The best machine-learning model for each catchment was chosen by comparing validation RMSE values, while test RMSE and test MAE values were never part of any selection or training procedure. The shown values all refer to the test time period.

	Best ML model results				LM		air2stream	
Catchment	Model	Experiment	RMSE (°C)	MAE (°C)	RMSE (°C)	MAE (°C	RMSE (°C)	MAE (°C)
Kleine Mühl	XGBoost	4 (TQP)	**0.740**	**0.578**	1.744	1.377	0.908	0.714
Aschach	XGBoost	6 (TQPGL)	**0.815**	**0.675**	1.777	1.408	1.147	0.882
Erlauf	XGBoost	6 (TQPGL)	**0.530**	**0.419**	1.354	1.057	0.911	0.726
Traisen	FNN	3 (TQ)	**0.526**	**0.392**	1.254	0.970	0.948	0.747
Ybbs	RF	3 (TQ)	**0.576**	**0.454**	1.787	1.415	0.948	0.756
Saalach	XGBoost	6 (TQPGL)	**0.527**	**0.420**	1.297	1.062	0.802	0.646
Enns	FNN	6 (TQPGL)	**0.454**	**0.347**	1.425	1.166	1.168	0.671
Inn	FNN	3 (TQ)	**0.422**	**0.329**	1.376	0.098	1.097	0.949
Salzach	FNN	4 (TQP)	**0.430**	**0.338**	1.327	1.077	0.743	0.595
Danube	RNN-LSTM	3 (TQ)	**0.521**	**0.415**	2.145	1.721	1.099	0.910
		Mean:	**0.554**	**0.437**	1.549	1.235	0.977	0.760

of training data are able to improve predictions compared to air2stream for all model types in five catchments.

From the results in Fig. 4a, b, c it seems likely that performance is in general influenced by the combination of model, data inputs (experiment) and catchment, while the influence of different experiments and catchments is larger than the influence of model types on test RMSE. The linear regression model for test RMSE with catchment, experiment and model type as regressors is able to explain most of the test RMSE variance with a coefficient of determination of $R^2 = 0.988$. Furthermore, it resulted in significant association of all catchments ($p < 10^{-15}$), all experiments ($p < 0.005$) and the FNN model type ($p < 0.001$). The estimated coefficient of the FNN is -0.05, giving evidence of a prediction improvement when applying the FNN model. All other model types do not show a significant association. However, this might be due to a lack of statistical power, as the estimated coefficients of the model types (mean: -0.01, range: $[-0.05, 0.02]$) are generally small compared to catchment coefficients (mean: 0.86, range: [0.69, 1.06]) and experiment coefficients (mean: -0.12, range: $[-0.2, -0.04]$). Overall, the influence of the catchment is higher than the influence of model type and experiment, which is clearly shown with their around 1 order of magnitude larger coefficients.

Multiple experiments often result in very similar RMSE values for a single model type. Furthermore, the best-performing experiments of different model types are always very close in performance. This results in a median test RMSE difference of the best experiments of different model types of 0.08 °C and a median test RMSE difference of the best-performing model and the second best model of another model type of 0.035 °C. On the other hand, the median difference between the tested machine-learning model RMSE and the air2stream RMSE is -0.39 °C.

The relationship between mean catchment elevation, glacier fraction and test RMSE was analysed with a linear model using mean catchment elevation, glacier fraction in percentage of the total catchment area, total catchment area and the experiments as independent variables and test RMSE as the dependent variable. This resulted in a significant association of elevation (p value $< 2 \times 10^{-16}$) with lower RMSE values and catchment area (p value $= 3.91 \times 10^{-4}$) and a significant association of glacier cover (p value $= 9.79 \times 10^{-5}$) with higher RMSE values. Applying the same model without using the data of the largest catchment, the Danube, resulted in a significant (p value $= 2.12 \times 10^{-11}$) association between catchment area and lower RMSE values, while the direction of the other associations stayed the same.

The run times for all applied ML models are summarized in Table 5. FNN and RF have the lowest median run times with comparatively narrow inter-quartile ranges (IQRs), so that most models take between 30 min and 1 h to train. XGBoost has a median run time of around 3 h (172.9 min) and also a comparatively low IQR with a span of 50 min. Step LM and both RNNs need much longer to train, with median run times of around 700 min. They also have a much larger variability in the needed run time, especially the step-LM model with an IQR of more than 1500 min. In contrast, the run time of the LM model is negligibly small (< 1 s), and air2stream is also considerably faster, with run times of < 2 min in all catchments.

3.3 Detailed analysis for a single catchment

To further investigate the difference in performance, the prediction results for the last year of the test data (2015) of

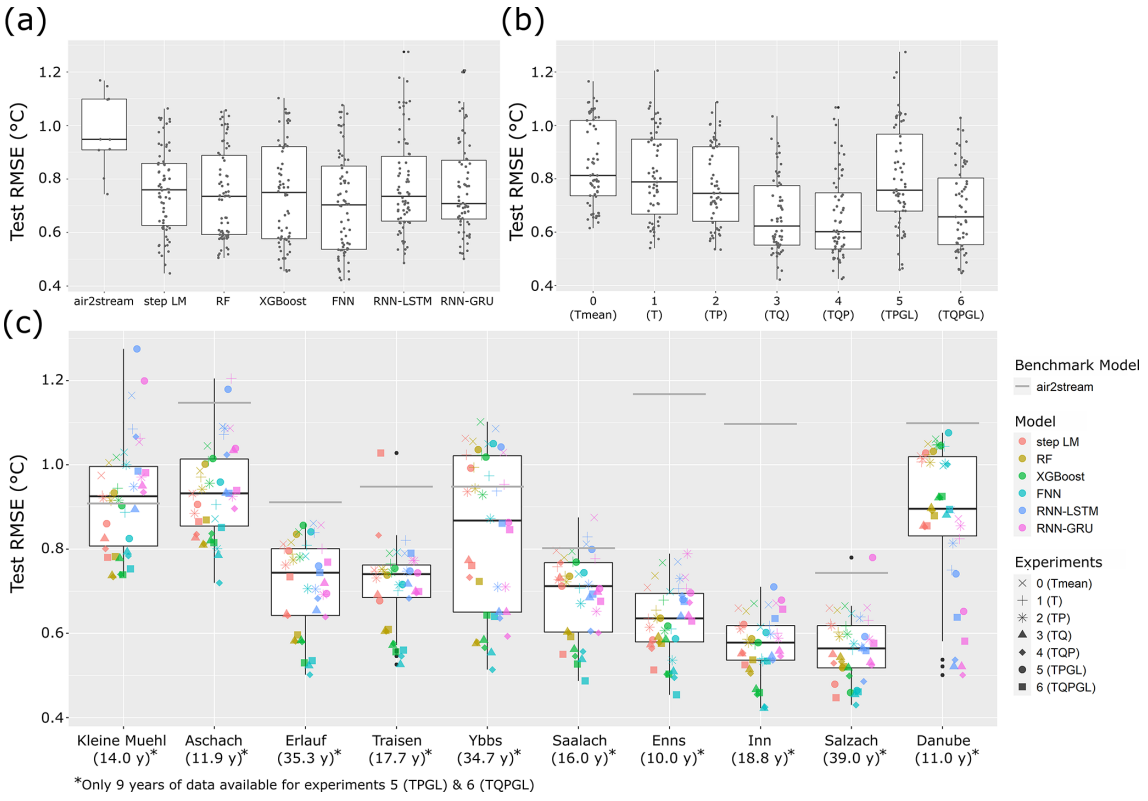

Figure 4. Boxplots of model performance comparing **(a)** the different machine-learning models, **(b)** the different experiments and **(c)** model performance in each catchment with additional scatter-plot overlay to show performance of individual combinations of catchments, models and experiments. The catchments in **(c)** are ordered by catchment size from smallest to largest with additional information of the available time-series length in parentheses below. The air2stream benchmark performance is shown as grey line for each catchment. Due to the much larger test RMSE values, LM performance is not shown to account for a better visibility.

Table 5. Run times of all applied ML models given as the median and inter-quartile ranges (IQR) of run times in minutes.

	Run times (min)	
Model	Median	IQR
Step-LM	698.9	158.8–1733.8
RF	54.3	44.3–74.6
XGBoost	172.9	153.6–204.0
FNN	30.8	28.5–41.5
RNN-LSTM	748.6	520.9–1111.6
RNN-GRU	767.8	583.9–1171.1

the Inn catchment are examined. The year 2015 was chosen for comparison, since it has an extraordinarily large number of days with high water temperatures and therefore can be used to give a robust estimate of model performance. It is a strong test under instationary conditions. The time period 1997–2014 has a median of 30 d per year with water temperatures over 11 °C, whereas 102 d with such high water temperature could be observed in the year 2015. Figure 5 shows the prediction results of each model (red lines) compared to the observation (blue line) and all other model predictions (grey lines) for the year 2015 and the corresponding RMSE and MAE result for that year.

The two benchmark models (LM and air2stream) show large differences between prediction and observations and show in general a very different behaviour than all tested machine-learning models. While the largest prediction errors of the tested machine-learning models occur during similar time periods, large deviations can be observed over the whole year in both benchmark models.

The largest prediction errors of all machine-learning models occur during warmer periods and peaks in the summer months and during periods of low water temperature in November–December. This is clearly visible in all tested models. Therefore, differences in RMSE and MAE mainly result from their performance during these periods and consequently can be quite large even though the actual numerical difference is rather small. This can be observed when comparing the results of best-performing model FNN and RNN-GRU in Fig. 5. Both models produce similar prediction results for the largest part of the year, but the FNN model is better able to predict the peaks with high water temperatures

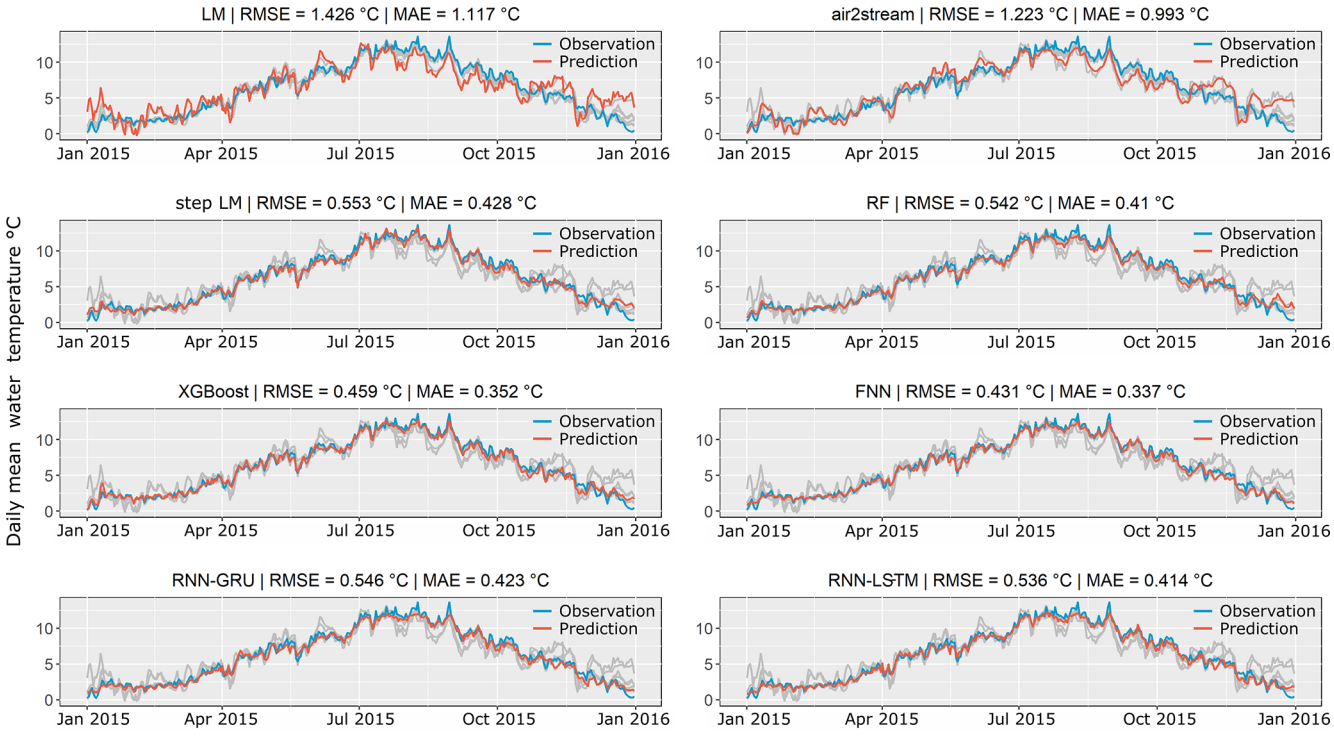

Figure 5. Comparison of the prediction of all tested model types for the Inn catchment for the year 2015. Data from 2015 were not used for training and validation. Prediction results for each model are shown with red lines, while the observations are shown in blue lines. The predictions of all other models are illustrated with grey lines.

in the summer months, which results in a RMSE and MAE difference of 0.115 and 0.086, respectively. Very small differences in RMSE and MAE as seen between the two best-performing models, FNN and XGBoost, result in only very subtle differences in the predicted time series. Very similar observations can be made when analysing the prediction results in the other catchments. The only exception can be observed in the largest catchment, the Danube (Fig. A2), where the time series is much smoother with relatively few peaks in water temperature. This results in the RNN models being the best-performing models with a large performance difference compared to all other models. The corresponding figures of all catchments except the Inn and Danube are provided in the Supplement.

3.4 Influence of time variables for RNNs, cross-validation methods

Removing time information in the form of fuzzy months from the training data of RNNs does not significantly change the catchment test RMSE ($p = 0.17$). However, the optimal number of time steps estimated by the hyperparameter optimization is significantly increased ($p = 0.02$). By removing time information from the inputs, the estimated time steps by Bayesian hyperparameter optimization are 37.78 d longer than when using time information as additional input. This significantly increases model training time ($p = 0.034$), with a mean difference of 132.45 min.

The different CV schemes applied to steps LM, RF and XGBoost showed no significant difference in performance ($p = 0.91$).

3.5 Influence of hyperparameters on model results

The influence of different sets of hyperparameters on model performance is shown in Fig. 6. This figure shows the validation RMSE for all parameter sets which were used during hyperparameter optimization. A large difference in the range of performance can be observed for different models. Validation RMSE means, standard deviations, and minimum and maximum of all models are shown in Table 6. The largest variability is apparent in the FNN results, with a validation RMSE standard deviation of $\sigma_{\mathrm{FNN}} = 1.60\,^{\circ}\mathrm{C}$ and an overall RMSE range of [0.41, 16.6] °C. This is followed by XGBoost, which has multiple outliers in each catchment that increase the performance spread, resulting in $\sigma_{\mathrm{XGBoost}} = 1.07\,^{\circ}\mathrm{C}$ and the RMSE range [0.40, 9.15] °C. Both RNNs show very similar performance distributions, with a RMSE range of around [0.45, 6.3] °C. Compared to all other tested models, the RF model has a much smaller spread in performance, resulting from different hyperparameter sets with $\sigma_{\mathrm{RF}} = 0.16\,^{\circ}\mathrm{C}$ and a resulting RMSE range of [0.45, 1.14] °C. Tables with all optimized hyperparameters are provided in the Supplement.

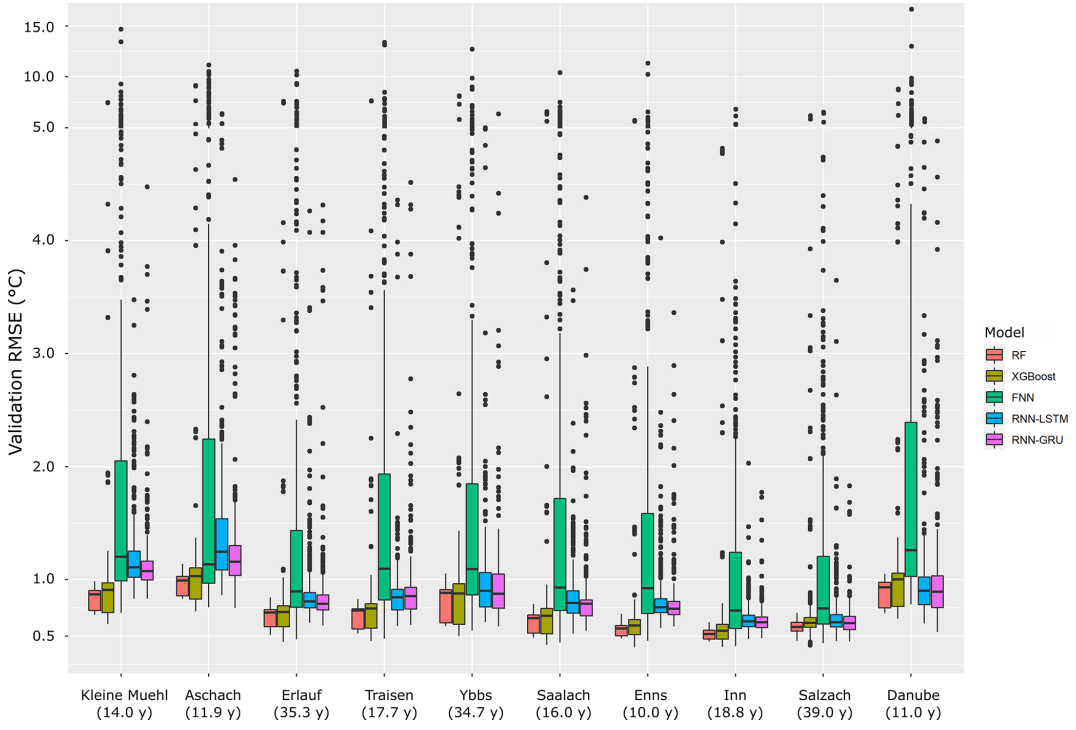

Figure 6. Boxplots showing the validation RMSE distribution for different hyperparameter sets for all model types, catchments and experiments. The catchments are ordered by catchment size from smallest (left) to largest (right), with additional information of the available time-series length in parentheses below.

Table 6. Validation RMSE means μ, standard deviations σ, and maxima and minima for all model types resulting from hyperparameter optimization.

	Validation RMSE (°C)			
Model	μ	σ	Min	Max
RF	0.70	0.16	0.45	1.14
XGBoost	0.95	1.07	0.40	9.15
FNN	1.70	1.60	0.41	16.6
RNN-LSTM	0.97	0.53	0.46	6.4
RNN-GRU	0.91	0.44	0.45	6.3

4 Discussion

In this study, we show the stream water temperature prediction performance of six machine-learning models with a range of input data sets in 10 catchments and compared them to two widely used benchmark models. The results show generally a very similar performance of the tested machine-learning models, with a median test RMSE difference of 0.08 °C between models. In contrast, the models had a significantly improved performance when compared to the air2stream benchmark model, with a mean test RMSE decrease of 0.42 °C (42 %). Results showed that nearly all of the test RMSE variance ($R^2 = 0.99$) can be explained by the catchment, the input data set and the model type. This also showed that the performance is significantly influenced by the type of input data, where more inputs generally performed better and, that of all models, only the FNN model had a significant association with lower test RMSE values. Furthermore, a wide range of performance was observed for the different hyperparameter sets for the tested models, with extremely large RMSE standard deviation (1.60 °C) observed in the FNN results.

Except for very few model types and experiment combinations, all tested machine-learning models showed an improved performance when compared to the two benchmark models. The difference between the benchmark and tested models was not only visible in the resulting test RMSE and MAE values, but also clearly visible in the range and time of occurrence of large prediction errors in the predicted time series (see Fig. 5). Given the range of estimated coefficients of the catchments ([0.69, 1.06]), data inputs ([−0.2, −0.04]) and model types ([−0.05, 0.02]) in the regression model for test RMSE, we can state that given an adequate model setup and selected hyperparameters, the influence of different data inputs and different catchments is much larger than the influence of the model types. However, there seems to be an advantage of using the FNN model, as it was the only model that had a significant association with lower RMSE values

and also the largest estimated coefficient of all model types (−0.05).

The result presented here shows that FNN and XGBoost perform best in 8 of 10 catchments and are therefore a first choice for water temperature prediction tasks. For modelling large catchments with comparable size to the Danube catchment (96 000 km^2), where long-term dependencies seem to be more relevant, RNNs are the best choice. Both RNN architectures, GRU and LSTM, produce very similar results in the Danube catchment, with a best test RMSE of approximately 0.52 °C. This is considerably lower than the median test RMSE of the other models (0.90 °C) and the air2stream benchmark (1.10 °C). The RF model has the lowest standard deviation in the resulting RMSE depending on the chosen hyperparameter (0.16 °C) and thus might be the most reasonable choice in situations with limited computational resources. More input data are generally better, but the combination of air temperature and discharge input data already produces prediction results with a median RMSE of 0.62 °C. This can be further enhanced by adding precipitation data, which decreases the median RMSE further to 0.60 °C. Adding GL data can potentially increase performance as well, as experiment 6 shows a similar performance range to experiment 3 while using only 6 years of training data. Results of experiment 2 (TP), which is most relevant for practical application as it uses inputs that are general available for most regions and from climate models, show a median test RMSE of 0.75 °C. This is only a 19 % reduction in RMSE performance compared to the experiment with the lowest median RMSE and an improvement of 21% compared to air2stream. Thus, application of this set of widely available data inputs is able to produce prediction performance, improving the current state of the art, and could be used as a basis for short-term forecasts and assessing near-future predictions (5–10 years) under climate change. The ability of ML approaches to simulate processes and signals from a system under prolonged climate change is important and a topic of future research.

The presented machine-learning approaches could considerably improve prediction results compared to the current state-of-the-art air2stream model. This stands in contrast to the findings of Zhu et al. (2019d), which assessed the performance of a suite of machine-learning models for daily stream water temperature. Zhu et al. (2019d) results showed that air2stream had an improved performance when compared to FNNs, Gaussian process regression and decision tree models in eight catchments using water temperature, discharge and day of year as model inputs. The air2stream results presented here have a test RMSE range of [0.74, 1.17] °C, which is comparable to results of Zhu et al. (2019d) with [0.64, 1.16] °C and also to other studies applying air2stream, e.g. Piotrowski and Napiorkowski (2018) with a range of [0.625, 1.31] °C. This leads us to the conclusion that our benchmark performance is in line with other air2stream applications and therefore provides a consistent reference, even though air2stream was originally set up for the use of point source data and not the catchment means that we used to make results comparable to the tested machine-learning models. Consequently, our presented approaches show a significant improvement compared to existing machine-learning daily stream water temperature prediction models, which can be attributed to the adequate representation of time (fuzzy months) as data input, the applied hyperparameter optimization, the choice of lagged time steps and the used input variables.

Due to the lack of physical restraints, statistical modelling approaches are often suspected of failing when extrapolating outside their training data range (Benyahya et al., 2007). However, machine-learning methods are more powerful and flexible than previous modelling approaches and are able to simultaneously use spatial and temporal information at different scales (Reichstein et al., 2019). This is especially important for climate change studies, where increasing air temperature might change the statistical relationships between meteorological drivers and stream water temperature. To investigate the extrapolation performance of the considered ML methods, we selected the much warmer recent years of the time series as a test period and analysed the year with the most frequent days of extreme temperatures in detail. All tested models where able to produce predictions with a performance close to the training performance in the test time period and in the year with the most temperature anomalies. These results show that it is still possible to produce robust prediction results at least for short time predictions (1–8 years) under a changing climate. Successful extrapolation for short-term periods suggests that mid- to long-term predictions might also produce reasonable results. However, this can only be evaluated based on future observations. It is clear that the ML approaches will fail in extrapolation, when catchment properties change with time. In the context of high-alpine, glacier-dominated catchments, for example, it can be assumed that the water temperature characteristics will change, when glaciers vanish. As a consequence, the underlying processes lead the water temperature in the stream change. These changes are not reflected in the ML approaches. It would need more physically or process-based approaches. For example, air2stream would not have an advantage in this respect. The current results suggest a strong influence of catchment properties on general ML model performance. While associations of performance with elevation, glacier cover and catchment area were apparent, we could not come to a strong conclusion, as even the direction of the relationship for one variable changed when removing one catchment from the analysis. We believe that there are a number of factors influencing these associations, and more in-depth investigations on a larger number of basins are needed to further understand the relationships between ML model performances and catchment properties and their implications.

Depending on the machine-learning model, our results varied considerably with the chosen hyperparameters. Espe-

cially the two best-performing models, XGBoost and FNNs, show an extreme variance in performance due to the chosen hyperparameters. This leads to the conclusion that flexibility might be necessary for a well-performing model but that it is also a possible source of error or reduced model performance. These findings highlight the importance of hyperparameter optimization of machine-learning models and might be a possible explanation of the fact that especially FNNs did not perform equally well in other studies. Most publications reporting findings regarding FNN performance for stream water temperature tested only a small set of FNN hyperparameter combinations, mostly chosen by trial and error (e.g. Piotrowski et al., 2015; Rabi et al., 2015; Abba et al., 2017; Zhu et al., 2018; Temizyurek and Dadaser-Celik, 2018; Zhu et al., 2019d). Our results show the extremely an large influence of hyperparameters, therefore rendering any trial-and-error approach insufficient and certainly non-optimal.

RNNs are successfully applied in current rainfall-runoff modelling studies (e.g. Kratzert et al., 2018, 2019; Xiang et al., 2020; Li et al., 2020), and are thus a promising candidate for stream water prediction. However, our results show a below average performance in most catchments when compared to the other tested machine-learning models. This is especially relevant, since compared to the other methods, RNNs use a range of previous time steps (optimized hyperparameter) for prediction, which contains much more information than the four previous time steps available for the other models. RNNs are the best-performing models in the largest catchment indicating that RNNs are especially strong when processes with long-term dependencies have to be described. These long-term dependencies result most likely from increased concentration times, which is generally dependent on catchment size (McGlynn et al., 2004). For all other catchments in this study, the 4 d lagged variables seem to be sufficient and RNNs are not able to predict the corresponding fast changes in water temperature. Our results also show the importance of using time information as input for RNNs. RNNs are generally able to learn the corresponding information from data, since there is no significant difference in performance for the RNNs with and without time information. However, RNNs optimized with time information inputs needed a significantly lower number of time steps for the same prediction performance, thus decreasing computation time and increasing the number of data points available for training.

This study has some limitations. Firstly, the selected catchments are all central European catchments with humid conditions. Testing these approaches on Mediterranean or more dynamic hydro-climatological conditions could potentially result in different importance of input variables (e.g. discharge in arid climates) and performance ranking of models. By selecting catchments with a wide range of physiographic characteristics this potential bias should be kept at a minimum. Furthermore, the performance of the air2stream benchmark is similar to the performance range of other studies, allowing for comparison. Secondly, we trained all models only for individual catchments and did not try to produce a global model that could predict water temperatures in multiple catchments, or even in a prediction of ungauged basin setting. While this is a relevant problem, we found it necessary to have a comprehensive evaluation of different data inputs, model types and training characteristics before combining all of this in a multi-catchment water temperature prediction model.

5 Conclusions

Current standard methods in daily stream water prediction are able to model 10 Austrian study catchments with a mean test RMSE of 1.55 °C (linear regression) and 0.98 °C (air2stream). We tested six machine-learning models with different data inputs and could produce predictions with a mean RMSE of 0.55 °C, an improvement of 64 % and 43 %. Of these tested models, the FNN model using air temperature, discharge and precipitation and, if available, radiation as inputs produces the best-performing models. With only 6 years of training data, state-of-the-art prediction model results can be achieved.

One major influence on performance are model hyperparameter. The variability in performance for different hyperparameters is much larger than for different model types or data inputs. Thus hyperparameter optimization is extremely important for a well-performing model. In situations where computing resources are limited and hyperparameter optimization is not possible, the RF model seems to be a reasonable choice for application, because it has the lowest variance in prediction RMSE resulting from the chosen hyperparameters.

RNNs with their internal states and ability to process long time series, are the best-performing model type for very large catchments. This is most likely a result from increased concentration times in the catchment. Consequently, estimating concentration times of a catchment for adequately choosing a model type or relevant lags of variables should be included in future research. Applying variable importance estimation methods are also another way to further enhance the understanding of the interactions between variables and model performance and could help deciding on the relevant number of variable lags. Applying these methods however, especially for neural networks, is out of scope for this study and will be part of future research.

The study catchments were chosen to have a wide range of physiographic characteristics but are all located in central Europe. Thus the range of characteristics is still limited and testing these model approaches in a wider range of catchments is still necessary and should also be included in future research. This will be especially important for developing multi-catchment water temperature prediction models for regional prediction, which is an important next step and

topic of current research. The development of regional models would also need to include comparison of cross-station scenarios and other tests for model transferability in time and space. The presented machine-learning methods, driven with observed meteorological inputs, seem to represent the system in an appropriate manner for applying them to predict river water temperature in changing conditions and may be promising for short time or real-time forecasting approaches. The resulting prediction uncertainties in such systems will be mainly related to uncertainties in the meteorological forecasts. By implementing all methods into the open-source R package wateRtemp, we hope to further contribute to reproducible research and make the presented methods available and easily applicable for management issues, scientists and industries and to facilitate research on these next steps.

Appendix A

A1 Long short-term memory cells

Given a sequence of inputs for T time steps $\boldsymbol{x_1}, \ldots, \boldsymbol{x}_T$, where each $\boldsymbol{x}_t \in \mathbb{R}^d$ is a vector of d input features, the forward pass of a single LSTM cell with h hidden units is given by the following equations:

$$\mathbf{f}_t = \sigma(\mathbf{W}_f \boldsymbol{x}_t + \mathbf{U}_f \mathbf{h}_{t-1} + \mathbf{b}_f), \tag{A1}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i \boldsymbol{x}_t + \mathbf{U}_i \mathbf{h}_{t-1} + \mathbf{b}_i), \tag{A2}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o \boldsymbol{x}_t + \mathbf{U}_o \mathbf{h}_{t-1} + \mathbf{b}_o), \tag{A3}$$

$$\tilde{\mathbf{c}_t} = \tanh(\mathbf{W}_g \boldsymbol{x}_t + \mathbf{U}_g \mathbf{h}_{t-1} + \mathbf{b}_g), \tag{A4}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}_t}, \tag{A5}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t), \tag{A6}$$

where $\mathbf{f}_t$, $\mathbf{i}_t$, and $\mathbf{o}_t \in \mathbb{R}^h$ are the forget gate, input gate and output gate, $\tilde{\mathbf{c}_t} \in \mathbb{R}^h$ is the cell input activation, $\mathbf{h}_t \in \mathbb{R}^h$ is the hidden state, $\mathbf{c}_t \in \mathbb{R}^h$ is the cell state and all $\mathbf{W} \in \mathbb{R}^{h \times d}$, $\mathbf{U} \in \mathbb{R}^{h \times h}$ and $\mathbf{b} \in \mathbb{R}^h$ are trainable weights. σ is the sigmoid function, tanh the hyperbolic tangent function and $\odot$ is element-wise multiplication.

The hidden state ($\mathbf{h}_t$) is computed from the current input ($\boldsymbol{x}_t$) and the previous hidden state ($\mathbf{h}_{t-1}$). The amount of information that is passed through the current cell is regulated by the input gate ($\mathbf{i}_t$) and the forget gate ($\mathbf{f}_t$). The cell state (c_t) regulates how much memory will be stored in the hidden state ($\mathbf{h}_t$). The output gate ($\mathbf{o}_t$) controls how much information is passed to the next cell.

A2 Gated recurrent unit cells

The GRU cell is similar to a LSTM cell but much simpler. It combines the forget and input gate into a single update gate and also merges the cell state and the hidden state. Given a sequence of inputs for T time steps $\boldsymbol{x_1}, \ldots, \boldsymbol{x}_T$, where each $\boldsymbol{x}_t \in \mathbb{R}^d$ is a vector of d input features, the forward pass of a single GRU cell with h hidden units is given by following equations:

$$\mathbf{z}_t = \sigma(\mathbf{W}_z \boldsymbol{x}_t + \mathbf{U}_z \mathbf{h}_{t-1} + \mathbf{b}_z), \tag{A7}$$

$$\mathbf{r}_t = \sigma(\mathbf{W}_r \boldsymbol{x}_t + \mathbf{U}_r \mathbf{h}_{t-1} + \mathbf{b}_r), \tag{A8}$$

$$\hat{\mathbf{h}}_t = \tanh(\mathbf{W}_h \mathbf{x}_t) + \mathbf{U}_h(\mathbf{r}_t \odot \mathbf{h}_{t-1} + \mathbf{b}_h), \tag{A9}$$

$$\mathbf{h}_t = (1 - \mathbf{z}_t) \odot \mathbf{h}_{t-1} + \mathbf{z}_t \odot \hat{\mathbf{h}}_t, \tag{A10}$$

where $\mathbf{z}_t \in \mathbb{R}^h$ is the update gate, $\mathbf{r}_t \in \mathbb{R}^h$ is the reset gate, $\hat{\mathbf{h}}_t \in \mathbb{R}^h$ is the candidate activation, $\mathbf{h}_t \in \mathbb{R}^d$ is the output and all $\mathbf{W} \in \mathbb{R}^{h \times d}$, $\mathbf{U} \in \mathbb{R}^{h \times h}$ and $\mathbf{b} \in \mathbb{R}^h$ are trainable weights. The reset gate ($\mathbf{r}_t$) determines how much information from the previous state will be forgotten when computing the candidate activation ($\hat{\mathbf{h}}_t$). The update gate is the amount of information used from the candidate activation ($\hat{\mathbf{h}}_t$) for computing the current output h_t.

A3 Model hyperparameter bounds

RF: min.node.size; 2–10, mtry: 3-(number of inputs -1) *XGBoost*: nrounds: 300–3000, eta: 0.001–0.3, max_depth: 3–12, min_child_weight: 1–10, subsample: 0.7–1, colsample_bytree: 0.7–1, gamma: 0–5 *FNN*: layers: 1–5, units: 5–200, dropout: 0–0.2, batch_size: 5–150, epochs: 100, early_stopping_patience: 5 *RNNs*: layers: 1–5, units: 5–300, dropout: 0–0.4, batch_size: 5–150, time steps: 5–200, epochs: 100, early_stopping_patience: 5.

Table A1. Overview of additional model quality criteria of the best machine-learning model for each catchment and the two reference models, consisting of the Nash–Sutcliffe model efficiency coefficient NSE, (Nash and Sutcliffe, 1970), the index of agreement d (Willmott, 1981) and the coefficient of determination R^2. The best machine-learning model for each catchment was chosen by comparing validation RMSE values. The shown values all refer to the test time period.

	Best ML model results					LM			air2stream		
Catchment	Model	Experiment	NSE	d	R^2	NSE	d	R^2	NSE	d	R^2
Kleine Mühl	XGBoost	4(TQP)	0.982	0.995	0.983	0.899	0.971	0.903	0.973	0.993	0.974
Aschach	XGBoost	6(TQPGL)	0.983	0.996	0.983	0.920	0.978	0.924	0.969	0.992	0.970
Erlauf	XGBoost	6(TQPGL)	0.985	0.996	0.986	0.884	0.968	0.900	0.959	0.989	0.960
Traisen	FNN	3(TQ)	0.985	0.996	0.985	0.912	0.977	0.915	0.951	0.988	0.955
Ybbs	RF	3(TQ)	0.989	0.997	0.989	0.889	0.971	0.890	0.968	0.992	0.969
Saalach	XGBoost	6(TQPGL)	0.977	0.994	0.979	0.864	0.961	0.883	0.951	0.988	0.955
Enns	FNN	6(TQPGL)	0.984	0.996	0.985	0.834	0.951	0.840	0.946	0.986	0.952
Inn	FNN	3(TQ)	0.984	0.996	0.984	0.829	0.949	0.830	0.882	0.968	0.882
Salzach	FNN	4(TQP)	0.986	0.996	0.986	0.862	0.961	0.864	0.957	0.989	0.963
Danube	RNN-LSTM	3(TQ)	0.986	0.996	0.989	0.842	0.955	0.843	0.961	0.990	0.968

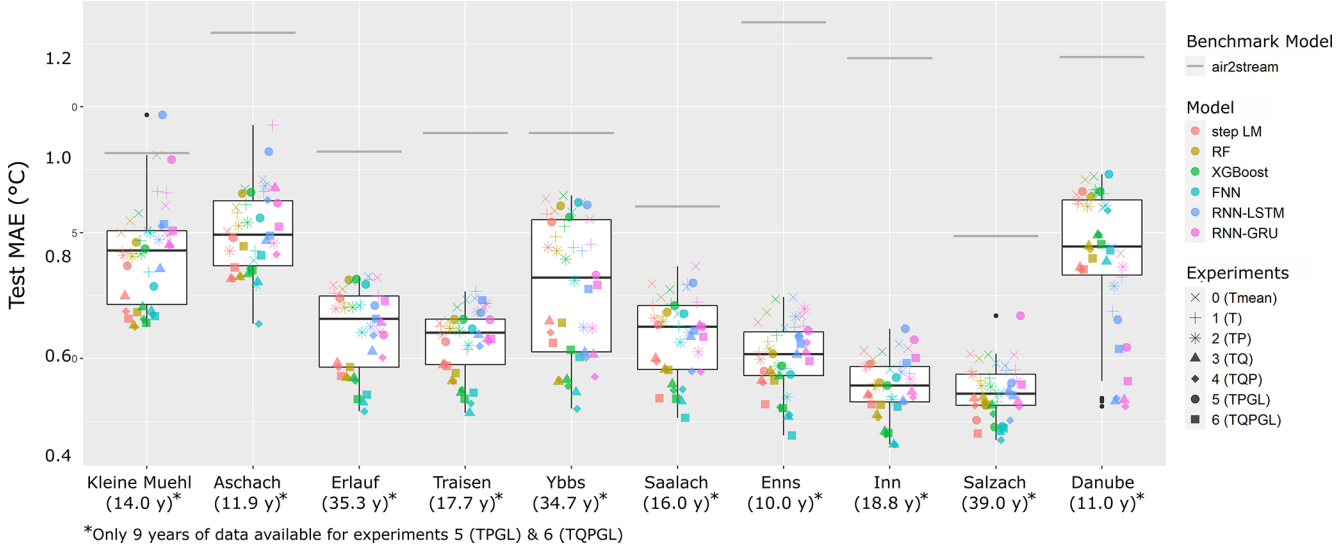

Figure A1. Boxplots of model performance comparing model MAE values in each catchment with additional scatter-plot overlay to show performance of individual combinations of catchments, models and experiments. The catchments are ordered by catchment size from smallest (left) to largest (right) with additional information of the available time-series length in parentheses below. The air2stream benchmark performance is illustrated as grey line for each catchment.

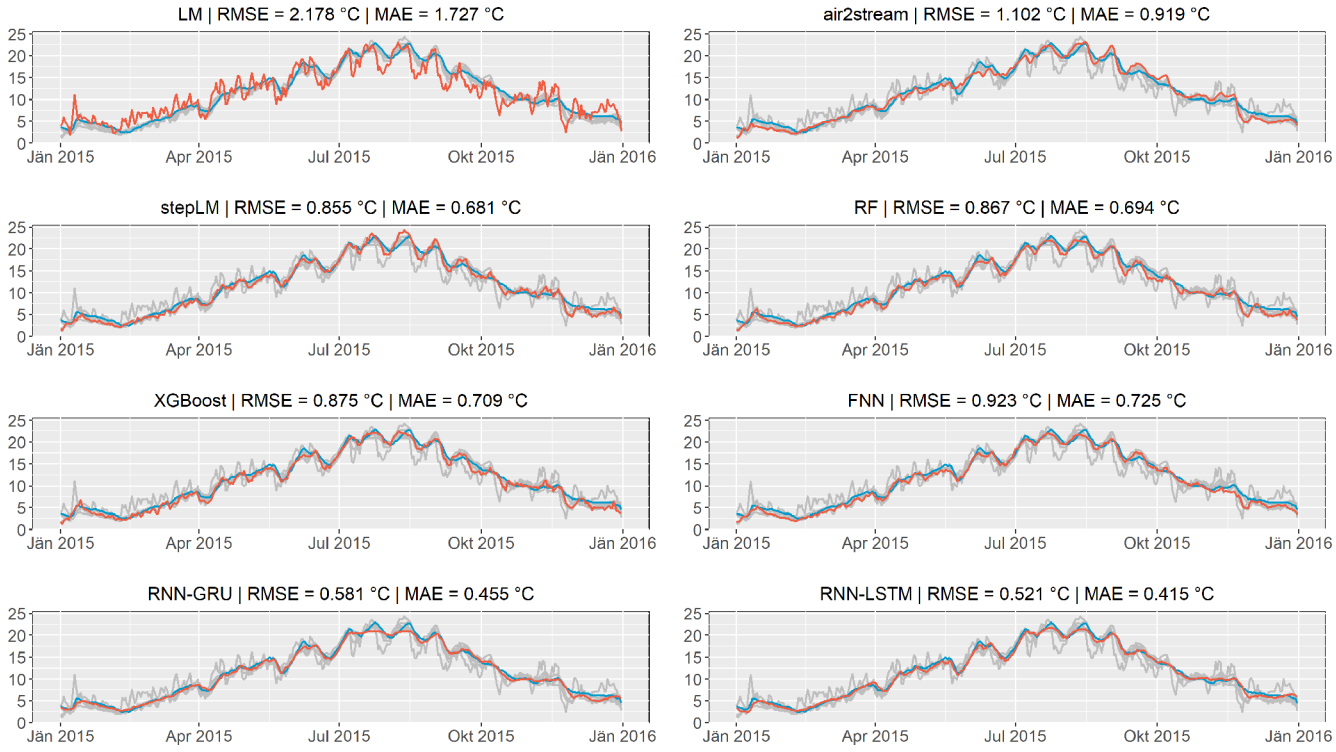

Figure A2. Comparison of the prediction of all tested model types for the Danube catchment for the year 2015. Prediction results for each model are shown with red lines, while the observations are shown with blue lines. The predictions of all other models are shown with grey lines.

Code and data availability. The R code used to generate all results for this publication can be found in Feigl (2021b). This includes the version of the wateRtemp R package providing all machine-learning methods and code that were used for producing the results of this paper. A maintained and continuously updated version of the wateRtemp package can be found at https://www.github.com/MoritzFeigl/wateRtemp (https://doi.org/10.5281/zenodo.4438575) or in Feigl (2021a).

We do not have permission for further distribution of the data used in this study. All input data can, however, be acquired from the rights holders of these data sets. The water temperature and discharge data used in this study can be requested from the Central Hydrographical Bureau (HZB) at https://www.ehyd.gv.at (Central Hydrographical Bureau, 2021). The rights for the meteorological data from the INCA and the SPARTACUS data sets belong to the Zentralanstalt für Meteorologie und Geodynamik (ZAMG) and can be acquired from https://www.zamg.ac.at (Zentralanstalt für Meteorologie und Geodynamik, 2021).

Supplement. The supplement related to this article is available online at: https://doi.org/10.5194/hess-25-2951-2021-supplement.

Author contributions. KL, MF and MH designed the study and acquired and processed the input data. MF and KL performed all analyses and prepared the figures. MF developed the software published with this work. MH and KS contributed to the methodological framework. MF prepared the paper with contributions from KL, MH and KS.

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

Acknowledgements. The computational results presented have been achieved using the Vienna Scientific Cluster (VSC). We also thank Ignacio Martin Santos for providing data from the upper Danube catchment and many valuable discussions about seasonal forecast and team spirit during the Covid-19 pandemic. Furthermore, we would like thank our reviewers, Salim Heddan and Adrien Michel, for their insightful comments and suggestions which helped to shape the manuscript into its current form.

Financial support. This research has been supported by the Austrian Science Fund (grant no. P 31213) and the Österreichische Akademie der Wissenschaften (Rechout and Poco-Flood).

Review statement. This paper was edited by Bettina Schaefli and reviewed by Adrien Michel and Salim Heddam.

References

Abba, S. I., Hadi, S. J., and Abdullahi, J.: River water modelling prediction using multi-linear regression, artificial neural network, and adaptive neuro-fuzzy inference system techniques, in: Procedia Computer Science, Elsevier B.V., Budapest, Hungary, 75–82, https://doi.org/10.1016/j.procs.2017.11.212, 2017.

Ahmadi-Nedushan, B., St-Hilaire, A., Ouarda, T. B. M. J., Bilodeau, L., Robichaud, É., Thiémonge, N., and Bobée, B.: Predicting river water temperatures using stochastic models: case study of the Moisie River (Québec, Canada), Hydrol. Process., 21, 21–34, https://doi.org/10.1002/hyp.6353, 2007.

Akaike, H.: Information theory as an extension of the likelihood principle., in: Second Akademiai International Symposium on Information Theory, edited by: Petrov, B. N. and Csaki, F., Kiado, Budapest, 267–281, 1973.

Allaire, J. J. and Tang, Y.: tensorflow: R Interface to "TensorFlow", available at: https://github.com/rstudio/tensorflow (last access: 13 Jauary 2021), 2020.

Álvarez, D. and Nicieza, A. G.: Compensatory response "defends" energy levels but not growth trajectories in brown trout, Salmo trutta L., P. Roy. Soc. B-Biol. Sci., 272, 601–607, https://doi.org/10.1098/rspb.2004.2991, 2005.

Arismendi, I., Safeeq, M., Dunham, J. B., and Johnson, S. L.: Can air temperature be used to project influences of climate change on stream temperature?, Environ. Res. Lett., 9, 084015, https://doi.org/10.1088/1748-9326/9/8/084015, 2014.

Baldi, P. and Sadowski, P.: The dropout learning algorithm, Artif. Intell., 210, 78–122, https://doi.org/10.1016/j.artint.2014.02.004, 2014.

Beaufort, A., Moatar, F., Curie, F., Ducharne, A., Bustillo, V., and Thiéry, D.: River Temperature Modelling by Strahler Order at the Regional Scale in the Loire River Basin, France, River Res. Appl., 32, 597–609, https://doi.org/10.1002/rra.2888, 2016.

Bélanger, M., El-Jabi, N., Caissie, D., Ashkar, F., and Ribi, J. M.: Water temperature prediction using neural networks and multiple linear regression, Revue des Sciences de l'Eau, 18, 403–421, https://doi.org/10.7202/705565ar, 2005.

Bengio, Y., Courville, A., and Vincent, P.: Representation learning: A review and new perspectives, IEEE T. Pattern Anal., 35, 1798–1828, https://doi.org/10.1109/TPAMI.2013.50, 2013.

Bentéjac, C., Csörgő, A., and Martínez-Muñoz, G.: A comparative analysis of gradient boosting algorithms, Artif. Intell. Rev., 54, 1937–1967, https://doi.org/10.1007/s10462-020-09896-5, 2021.

Benyahya, L., Caissie, D., St-Hilaire, A., Ouarda, T. B., and Bobée, B.: A Review of Statistical Water Temperature Models, Can. Water Resour. J., 32, 179–192, https://doi.org/10.4296/cwrj3203179, 2007.

BMLFUW: Hydrological Atlas of Austria, 3rd Edn., Bundesministerium für Land- und Forstwirtschaft, Umwelt und Wasserwirtschaft, Vienna, Austria, ISBN 3-85437-250-7, 2007.

Boisneau, C., Moatar, F., Bodin, M., and Boisneau, P.: Does global warming impact on migration patterns and recruitment of Allis shad (Alosa alosa L.) young of the year in the Loire River, France?, in: Fish and Diadromy in Europe (ecology, management, conservation), Springer, Dordrecht, the Netherlands, 179–186, https://doi.org/10.1007/978-1-4020-8548-2_14, 2008.

Breiman, L.: Bagging predictors, Mach. Learn., 24, 123–140, https://doi.org/10.1007/bf00058655, 1996.

Breiman, L.: Random forests, Mach. Learn., 45, 5–32, https://doi.org/10.1023/A:1010933404324, 2001.

Brinckmann, S., Krähenmann, S., and Bissolli, P.: High-resolution daily gridded data sets of air temperature and wind speed for Europe, Earth Syst. Sci. Data, 8, 491–516, https://doi.org/10.5194/essd-8-491-2016, 2016.

Caissie, D.: The thermal regime of rivers: A review, Freshwater Biol., 51, 1389–1406, https://doi.org/10.1111/j.1365-2427.2006.01597.x, 2006.

Caissie, D. and Luce, C. H.: Quantifying streambed advection and conduction heat fluxes, Water Resour. Res., 53, 1595–1624, https://doi.org/10.1002/2016WR019813, 2017.

Caldwell, R. J., Gangopadhyay, S., Bountry, J., Lai, Y., and Elsner, M. M.: Statistical modeling of daily and subdaily stream temperatures: Application to the Methow River Basin, Washington, Water Resour. Res., 49, 4346–4361, https://doi.org/10.1002/wrcr.20353, 2013.

Central Hydrographical Bureau: eHYD, available at: https://www.ehyd.gv.at/, last access: 26 May 2021.

Chen, T. and Guestrin, C.: XGBoost: A scalable tree boosting system, in: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, California, USA, 13–17 August 2016, 785–794, https://doi.org/10.1145/2939672.2939785, 2016.

Chen, T., He, T., Benesty, M., Khotilovich, V., Tang, Y., Cho, H., Chen, K., Mitchell, R., Cano, I., Zhou, T., Li, M., Xie, J., Lin, M., Geng, Y., and Li, Y.: xgboost: Extreme Gradient Boosting, available at: https://cran.r-project.org/package=xgboost (last access: 13 January 2021), 2020.

Chenard, J.-F. and Caissie, D.: Stream temperature modelling using artificial neural networks: application on Catamaran Brook, New Brunswick, Canada, Hydrol. Process., 22, 3361–3372, https://doi.org/10.1002/hyp.6928, 2008.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation, in: Proceedings of the EMNLP 2014 – 2014 Conference on Empirical Methods in Natural Language Processing, 25–29 October 2014, Doha, Qatar, 1724–1734, https://doi.org/10.3115/v1/d14-1179, 2014.

Claesen, M. and De Moor, B.: Hyperparameter Search in Machine Learning, arXiv [preprint], arXiv:1502.02127 (last access: 4 December 2020), 2015.

Crank, J. and Nicolson, P.: A practical method for numerical evaluation of solutions of partial differential equations of the heat-conduction type, Math. Proc. Cambridge, 43, 50–67, https://doi.org/10.1017/S0305004100023197, 1947.

Crisp, D. and Howson, G.: Effect of air temperature upon mean water temperature in streams in the north Pennines and English Lake District, Freshwater Biol., 12, 359–367, https://doi.org/10.1111/j.1365-2427.1982.tb00629.x, 1982.

Dallas, H.: Water temperature and riverine ecosystems: An overview of knowledge and approaches for assessing biotic responses, with special reference to South Africa, Water Sa, 34, 393–404, https://doi.org/10.4314/wsa.v34i3.180634, 2008.

DeWeber, J. T. and Wagner, T.: A regional neural network ensemble for predicting mean daily river water temperature, J. Hydrol., 517, 187–200, https://doi.org/10.1016/j.jhydrol.2014.05.035, 2014.

Dugdale, S. J., Hannah, D. M., and Malcolm, I. A.: River temperature modelling: A review of process-based approaches and future directions, Earth-Sci. Rev., 175, 97–113, https://doi.org/10.1016/j.earscirev.2017.10.009, 2017.

Dunn, O. J.: Multiple Comparisons Using Rank Sums, Technometrics, 6, 241–252, https://doi.org/10.1080/00401706.1964.10490181, 1964.

Feigl, M.: MoritzFeigl/wateRtemp: HESS submission (Version v0.2.0), Zenodo, https://doi.org/10.5281/zenodo.4438575, 2021a.

Feigl, M.: MoritzFeigl/ML_methods_for_stream_water_temperature_prediction: HESS paper (Version v1.0), Zenodo, https://doi.org/10.5281/zenodo.4438582, 2021b.

Fernández-Delgado, M., Cernadas, E., Barro, S., and Amorim, D.: Do we need hundreds of classifiers to solve real world classification problems?, J. Mach. Learn. Res., 15, 3133–3181, 2014.

Freund, Y. and Schapire, R. E.: A decision-theoretic generalization of on-line learning and an application to boosting, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), Springer Verlag, 23–37, https://doi.org/10.1007/3-540-59119-2_166, 1995.

Friberg, N., DybkjÆr, J. B., Olafsson, J. S., Gislason, G. M., Larsen, S. E., and Lauridsen, T. L.: Relationships between structure and function in streams contrasting in temperature, Freshwater Biol., 54, 2051–2068, https://doi.org/10.1111/j.1365-2427.2009.02234.x, 2009.

Friedman, J. H.: Greedy function approximation: A gradient boosting machine, Ann. Stat., 29, 1189–1232, https://doi.org/10.1214/aos/1013203451, 2001.

Friedman, J. H.: Stochastic gradient boosting, Comput. Stat. Data An., 38, 367–378, https://doi.org/10.1016/S0167-9473(01)00065-2, 2002.

Gauch, M., Tang, R., Mai, J., Tolson, B., Gharari, S., and Lin, J.: Machine Learning for Streamflow Prediction: Current Status and Future Prospects, 9–13 December 2019, San Francisco, USA, AGU Fall Meeting Abstracts, 2019, H33L–2127, 2019.

Graf, R., Zhu, S., and Sivakumar, B.: Forecasting river water temperature time series using a wavelet-neural network hybrid modelling approach, J. Hydrol., 578, 124115, https://doi.org/10.1016/j.jhydrol.2019.124115, 2019.

Hadzima-Nyarko, M., Rabi, A., and Šperac, M.: Implementation of Artificial Neural Networks in Modeling the Water-Air Temperature Relationship of the River Drava, Water Resour. Manag., 28, 1379–1394, https://doi.org/10.1007/s11269-014-0557-7, 2014.

Haiden, T., Kann, A., Wittmann, C., Pistotnik, G., Bica, B., and Gruber, C.: The integrated nowcasting through comprehensive analysis (INCA) system and its validation over the Eastern Alpine region, Weather Forecast., 26, 166–183, https://doi.org/10.1175/2010WAF2222451.1, 2011.

Haiden, T., Kann, A., and Pistotnik, G.: Nowcasting with INCA During SNOW-V10, Pure Appl. Geophys., 171, 231–242, https://doi.org/10.1007/s00024-012-0547-8, 2014.

Hannah, D. M. and Garner, G.: River water temperature in the United Kingdom, Prog. Phys. Geog., 39, 68–92, https://doi.org/10.1177/0309133314550669, 2015.

Hannah, D. M., Webb, B. W., and Nobilis, F.: River and stream temperature: dynamics, processes, models and implications, Hydrol. Process., 22, 899–901, https://doi.org/10.1002/hyp.6997, 2008.

Hansen, L. K. and Salamon, P.: Neural Network Ensembles, IEEE T. Pattern Anal., 12, 993–1001, https://doi.org/10.1109/34.58871, 1990.

Harvey, R., Lye, L., Khan, A., and Paterson, R.: The influence of air temperature on water temperature and the concentration of dissolved oxygen in Newfoundland Rivers, Can. Water Resour. J., 36, 171–192, https://doi.org/10.4296/cwrj3602849, 2011.

He, J., Yang, K., Tang, W., Lu, H., Qin, J., Chen, Y., and Li, X.: The first high-resolution meteorological forcing dataset for land process studies over China, Scientific Data, 7, 25, https://doi.org/10.1038/s41597-020-0369-y, 2020.

Heddam, S., Ptak, M., and Zhu, S.: Modelling of daily lake surface water temperature from air temperature: Extremely randomized trees (ERT) versus Air2Water, MARS, M5Tree, RF and MLPNN, J. Hydrol., 588, 125130, https://doi.org/10.1016/j.jhydrol.2020.125130, 2020.

Hersbach, H., Bell, B., Berrisford, P., Hirahara, S., Horányi, A., Muñoz-Sabater, J., Nicolas, J., Peubey, C., Radu, R., Schepers, D., Simmons, A., Soci, C., Abdalla, S., Abellan, X., Balsamo, G., Bechtold, P., Biavati, G., Bidlot, J., Bonavita, M., De Chiara, G., Dahlgren, P., Dee, D., Diamantakis, M., Dragani, R., Flemming, J., Forbes, R., Fuentes, M., Geer, A., Haimberger, L., Healy, S., Hogan, R. J., Hólm, E., Janisková, M., Keeley, S., Laloyaux, P., Lopez, P., Lupu, C., Radnoti, G., de Rosnay, P., Rozum, I., Vamborg, F., Villaume, S., and Thépaut, J. N.: The ERA5 global reanalysis, Q. J. Roy. Meteor. Soc., 146, 1999–2049, https://doi.org/10.1002/qj.3803, 2020.

Hiebl, J. and Frei, C.: Daily temperature grids for Austria since 1961 – concept, creation and applicability, Theor. Appl. Climatol., 124, 161–178, https://doi.org/10.1007/s00704-015-1411-4, 2016.

Hiebl, J. and Frei, C.: Daily precipitation grids for Austria since 1961 – development and evaluation of a spatial dataset for hydroclimatic monitoring and modelling, Theor. Appl. Climatol., 132, 327–345, https://doi.org/10.1007/s00704-017-2093-x, 2018.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R.: Improving neural networks by preventing co-adaptation of feature detectors, arXiv [preprint], arXiv:1207.0580 (last access: 7 August 2020), 2012.

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.

Hsu, C.-W., Chang, C.-C., and Lin, C.-J.: A Practical Guide to Support Vector Classification, Tech. Rep., Taipei, 2003.

Ibrahem Ahmed Osman, A., Najah Ahmed, A., Chow, M. F., Feng Huang, Y., and El-Shafie, A.: (2021). Extreme gradient boosting (xgboost) model to predict the groundwater levels in Selangor Malaysia, Ain Shams Engineering Journal, https://doi.org/10.1016/j.asej.2020.11.011, in press, 2021.

Imholt, C., Gibbins, C. N., Malcolm, I. A., Langan, S., and Soulsby, C.: Influence of riparian cover on stream temperatures and the growth of the mayfly Baetis rhodani in an upland stream, Aquat. Ecol., 44, 669–678, https://doi.org/10.1007/s10452-009-9305-0, 2010.

Jackson, F. L., Fryer, R. J., Hannah, D. M., Millar, C. P., and Malcolm, I. A.: A spatio-temporal statistical model of maximum daily river temperatures to inform the management of Scotland's Atlantic salmon rivers under climate change, Sci. Total Environ., 612, 1543–1558, https://doi.org/10.1016/j.scitotenv.2017.09.010, 2018.

Johnson, M. F., Wilby, R. L., and Toone, J. A.: Inferring air-water temperature relationships from river and catchment properties, Hydrol. Process., 28, 2912–2928, https://doi.org/10.1002/hyp.9842, 2014.

Jones, D. R., Schonlau, M., and Welch, W. J.: Efficient Global Optimization of Expensive Black-Box Functions, J. Global Optim., 13, 455–492, https://doi.org/10.1023/A:1008306431147, 1998.

Joslyn, K.: Water quality factor prediction using supervised machine learning REU Final Reports, 6, available at: https://archives.pdx.edu/ds/psu/26231 (last access: 26 May 2021), 2018.

Kędra, M.: Regional Response to Global Warming: Water Temperature Trends in Semi-Natural Mountain River Systems, Water, 12, 283, https://doi.org/10.3390/w12010283, 2020.

Kennedy, J. and Eberhart, R.: Particle swarm optimization, in: Proceedings of ICNN'95 – International Conference on Neural Networks, Perth, Australia, 27 November–1 December 1995, 1942–1948, https://doi.org/10.1109/ICNN.1995.488968, 1995.

Klambauer, G., Unterthiner, T., Mayr, A., and Hochreiter, S.: Self-normalizing neural networks, arXiv [preprint], arXiv:1706.02515, (last access: 3 August 2020), 2017.

Kling, H., Stanzel, P., Fuchs, M., and Nachtnebel, H.-P.: Performance of the COSERO precipitation–runoff model under non-stationary conditions in basins with different climates, Hydrolog. Sci. J., 60, 1374–1393, https://doi.org/10.1080/02626667.2014.959956, 2015.

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., 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, 2019.

Kruskal, W. H. and Wallis, W. A.: Use of Ranks in One-Criterion Variance Analysis, J. Am. Stat. Assoc., 47, 583–621, https://doi.org/10.1080/01621459.1952.10483441, 1952.

Kuhn, M.: caret: Classification and Regression Training, R package version 6.0-86, available at: https://CRAN.R-project.org/package=caret (last access: 13 Jaunary 2021), 2020.

Kushner, H. J.: A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise, J. Fluid. Eng.-T. ASME, 86, 97–106, https://doi.org/10.1115/1.3653121, 1964.

Laizé, C. L., Acreman, M. C., Schneider, C., Dunbar, M. J., Houghton-Carr, H. A., Flörke, M., and Hannah, D. M.: Projected flow alteration and ecological risk for pan-European rivers, River Res. Appl., 30, 299–314, https://doi.org/10.1002/rra.2645, 2014.

Li, H., Deng, X., Kim, D.-Y., and Smith, E. P.: Modeling maximum daily temperature using a varying coefficient regression model, Water Resour. Res., 50, 3073–3087, https://doi.org/10.1002/2013WR014243, 2014.

Li, W., Kiaghadi, A., and Dawson, C.: High temporal resolution rainfall–runoff modeling using long-short-term-memory (LSTM) networks, Neural Comput. Appl., 33, 1261–1278, https://doi.org/10.1007/s00521-020-05010-6, 2020.

Lu, H. and Ma, X.: Hybrid decision tree-based machine learning models for short-term water quality prediction, Chemosphere, 249, 126169, https://doi.org/10.1016/j.chemosphere.2020.126169, 2020.

Mackey, A. P. and Berrie, A. D.: The prediction of water temperatures in chalk streams from air temperatures, Hydrobiologia, 210, 183–189, https://doi.org/10.1007/BF00034676, 1991.

McGlynn, B. L., McDonnell, J. J., Seibert, J., and Kendall, C.: Scale effects on headwater catchment runoff timing, flow sources, and groundwater-streamflow relations, Water Resour. Res., 40, W07504, https://doi.org/10.1029/2003WR002494, 2004.

McKenna, J. E., Butryn, R. S., and McDonald, R. P.: Summer Stream Water Temperature Models for Great Lakes Streams: New York, T. Am. Fish. Soc., 139, 1399–1414, https://doi.org/10.1577/t09-153.1, 2010.

Močkus, J.: On Bayesian Methods for Seeking the Extremum, in: Optimization Techniques IFIP Technical Conference, Novosibirsk, 1–7 July 1974, 400–404, https://doi.org/10.1007/978-3-662-38527-2_55, 1975.

Močkus, J.: Bayesian Approach to Global Optimization, Mathematics and Its Applications Series, Springer Netherlands, Dordrecht, The Netherlands, 270 pp., https://doi.org/10.1007/978-94-009-0909-0, 1989.

Močkus, J., Tiesis, V., and Zilinskas, A.: The application of Bayesian methods for seeking the extremum, Towards global optimization, 2, 117–129, https://doi.org/10.1007/978-94-009-0909-0_8, 1978.

Mohseni, O. and Stefan, H. G.: Stream temperature/air temperature relationship: A physical interpretation, J. Hydrol., 218, 128–141, https://doi.org/10.1016/S0022-1694(99)00034-7, 1999.

Naresh, A. and Rehana, S.: Modeling Stream Water Temperature using Regression Analysis with Air Temperature and Streamflow over Krishna River, Rehana International Journal of Engineering Technology Science and Research, 4, 2394–3386, 2017.

Nash, J. and Sutcliffe, J.: River flow forecasting through conceptual models part I – A discussion of principles, J. Hydrol., 10, 282–290, https://doi.org/10.1016/0022-1694(70)90255-6, 1970.

Neumann, D. W., Rajagopalan, B., and Zagona, E. A.: Regression model for daily maximum stream temperature, J. Environ. Eng., 129, 667–674, https://doi.org/10.1061/(ASCE)0733-9372(2003)129:7(667), 2003.

Ni, L., Wang, D., Wu, J., Wang, Y., Tao, Y., Zhang, J., and Liu, J.: Streamflow forecasting using extreme gradient boosting model coupled with Gaussian mixture model, J. Hydrol., 586, 124901, https://doi.org/10.1016/j.jhydrol.2020.124901, 2020.

Nielsen, D.: Tree Boosting With XGBoost: Why does XGBoost win every machine learning competition?, Master's Thesis, Norwegian University of Science and Technology, Norway, 98 pp., 2016.

Pedersen, N. L. and Sand-Jensen, K.: Temperature in lowland Danish streams: contemporary patterns, empirical models and future scenarios, Hydrol. Process., 21, 348–358, https://doi.org/10.1002/hyp.6237, 2007.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R. Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E.: Scikit-learn: Machine Learning in Python, Tech. Rep., available at: https://hal.inria.fr/hal-00650905v2 (last access: 4 December 2020), 2011.

Piccolroaz, S., Calamita, E., Majone, B., Gallice, A., Siviglia, A., and Toffolon, M.: Prediction of river water temperature: a comparison between a new family of hybrid models and statistical approaches, Hydrol. Process., 30, 3901–3917, https://doi.org/10.1002/hyp.10913, 2016.

Pinkus, A.: Approximation theory of the MLP model in neural networks, Acta Numer., 8, 143–195, https://doi.org/10.1017/S0962492900002919, 1999.

Piotrowski, A. P. and Napiorkowski, J. J.: Performance of the air2stream model that relates air and stream water temperatures depends on the calibration method, J. Hydrol., 561, 395–412, https://doi.org/10.1016/j.jhydrol.2018.04.016, 2018.

Piotrowski, A. P. and Napiorkowski, J. J.: Simple modifications of the nonlinear regression stream temperature model for daily data, J. Hydrol., 572, 308–328, https://doi.org/10.1016/j.jhydrol.2019.02.035, 2019.

Piotrowski, A. P., Napiorkowski, M. J., Napiorkowski, J. J., and Osuch, M.: Comparing various artificial neural network types for water temperature prediction in rivers, J. Hydrol., 529, 302–315, https://doi.org/10.1016/j.jhydrol.2015.07.044, 2015.

Piotrowski, A. P., Napiorkowski, J. J., and Piotrowska, A. E.: Impact of deep learning-based dropout on shallow neural networks applied to stream temperature modelling, Earth-Sci. Rev., 201, 103076, https://doi.org/10.1016/j.earscirev.2019.103076, 2020.

R Core Team: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, available at: https://www.r-project.org/ (last access: 13 January 2021), 2020.

Rabi, A., Hadzima-Nyarko, M., and Šperac, M.: Modelling river temperature from air temperature: case of the River Drava (Croatia), Hydrolog. Sci. J., 60, 1490–1507, https://doi.org/10.1080/02626667.2014.914215, 2015.

Razafimaharo, C., Krähenmann, S., Höpp, S., Rauthe, M., and Deutschländer, T.: New high-resolution gridded dataset of daily mean, minimum, and maximum temperature and relative humidity for Central Europe (HYRAS), Theor. Appl. Climatol., 142, 1531–1553, https://doi.org/10.1007/s00704-020-03388-w, 2020.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat: Deep learning and process understanding for data-driven Earth system science, Nature, 566, 195–204, https://doi.org/10.1038/s41586-019-0912-1, 2019.

Risley, J. C., Roehl Jr., E. A., and Conrads, P. A.: Estimating Water Temperatures in Small Streams in Estimating Water Temperatures in Small Streams in Western Oregon Using Neural Network Models, Tech. Rep., USGS Water- Resources Investigation Report 02-4218, https://doi.org/10.3133/wri024218, 2003.

Rumelhart, D. E., Hinton, G. E., and Williams, R. J.: Learning representations by back-propagating errors, Nature, 323, 533–536, https://doi.org/10.1038/323533a0, 1986.

Sahoo, G. B., Schladow, S. G., and Reuter, J. E.: Forecasting stream water temperature using regression analysis, artificial neural network, and chaotic non-linear dynamic models, J. Hydrol., 378, 325–342, https://doi.org/10.1016/j.jhydrol.2009.09.037, 2009.

Sand-Jensen, K. and Pedersen, N. L.: Differences in temperature, organic carbon and oxygen consumption among lowland streams, Freshwater Biol., 50, 1927–1937, https://doi.org/10.1111/j.1365-2427.2005.01436.x, 2005.

Schapire, R. E.: The Strength of Weak Learnability, Mach. Learn., 5, 197–227, https://doi.org/10.1023/A:1022648800760, 1990.

Segura, C., Caldwell, P., Sun, G., Mcnulty, S., and Zhang, Y.: A model to predict stream water temperature across the conterminous USA, Hydrol. Process., 29, 2178–2195, https://doi.org/10.1002/hyp.10357, 2015.

Shank, D. B., Hoogenboom, G., and McClendon, R. W.: Dewpoint temperature prediction using artificial neural networks, J. Appl. Meteorol. Clim., 47, 1757–1769, https://doi.org/10.1175/2007JAMC1693.1, 2008.

Smith, K.: The prediction of river water temperatures, Hydrol. Sci. B., 26, 19–32, https://doi.org/10.1080/02626668109490859, 1981.

Snoek, J., Larochelle, H., and Adams, R. P. (2012). Practical bayesian optimization of machine learning algorithms, arXiv [preprint], arXiv:1206.2944 (last access: 6 August 2020), 2012.

Sohrabi, M. M., Benjankar, R., Tonina, D., Wenger, S. J., and Isaak, D. J.: Estimation of daily stream water temperatures with a Bayesian regression approach, Hydrol. Process., 31, 1719–1733, https://doi.org/10.1002/hyp.11139, 2017.

Srinivas, N., Krause, A., Kakade, S. M., and Seeger, M.: Gaussian Process Optimization in the Bandit Setting: No Regret and Experimental Design, IEEE T. Inform. Theory, 58, 3250–3265, https://doi.org/10.1109/TIT.2011.2182033, 2009.

Srivastava, N., Hinton, G., Krizhevsky, A., and Salakhutdinov, R.: Dropout: A Simple Way to Prevent Neural Networks from Overfitting, Tech. Rep., 15, 1929-−1958, available at: http://jmlr.org/papers/v15/srivastava14a.html (last access: 7 August 2020), 2014.

Stajkowski, S., Kumar, D., Samui, P., Bonakdari, H., and Gharabaghi, B.: Genetic-algorithm-optimized sequential model for water temperature prediction, Sustainability-Basel, 12, 5374, https://doi.org/10.3390/su12135374, 2020.

Stefan, H. G. and Preud'homme, E. B.: Stream temperature estimation from air temperature, J. Am. Water Resour. As., 29, 27–45, https://doi.org/10.1111/j.1752-1688.1993.tb01502.x, 1993.

Stevens, H., Ficke, J., and Smoot, G.: Techniques of water-resources investigations of the US Geological Survey, US Government Printing Office, Washington, 65 pp., 1975.

Tavares, M. H., Cunha, A. H. F., Motta-Marques, D., Ruhoff, A. L., Fragoso, C. R., Munar, A. M., and Bonnet, M. P.: Derivation of consistent, continuous daily river temperature data series by combining remote sensing and water temperature models, Remote Sens. Environ., 241, 111721, https://doi.org/10.1016/j.rse.2020.111721, 2020.

Temizyurek, M. and Dadaser-Celik, F.: Modelling the effects of meteorological parameters on water temperature using artificial neural networks, Water Sci. Technol., 77, 1724–1733, https://doi.org/10.2166/wst.2018.058, 2018.

Thornton, M. M., Shrestha, R., Wei, Y., Thornton, P. E., Kao, S., and Wilson, B. E.: Daymet: Daily Surface Weather Data on a 1-km Grid for North America, Version 4, ORNL DAAC, Oak Ridge, Tennessee, USA, https://doi.org/10.3334/ORNLDAAC/1840, 2020.

Toffolon, M. and Piccolroaz, S.: A hybrid model for river water temperature as a function of air temperature and discharge, Environ. Res. Lett., 10, 114011, https://doi.org/10.1088/1748-9326/10/11/114011, 2015.

Trinh, N. X., Trinh, T. Q., Phan, T. P., Thanh, T. N., and Thanh, B. N.: Water Temperature Prediction Models in Northern Coastal Area, Vietnam, Asian Review of Environmental and Earth Sciences, 6, 1–8, https://doi.org/10.20448/journal.506.2019.61.1.8, 2019.

Van Vliet, M. T., Franssen, W. H., Yearsley, J. R., Ludwig, F., Haddeland, I., Lettenmaier, D. P., and Kabat, P.: Global river discharge and water temperature under climate change, Global Environ. Chang., 23, 450–464, https://doi.org/10.1016/j.gloenvcha.2012.11.002, 2013.

Webb, B. W. and Zhang, Y.: Spatial and seasonal variability in the components of the river heat budget, Hydrol. Process., 11, 79–101, https://doi.org/10.1002/(sici)1099-1085(199701)11:1<79::aid-hyp404>3.0.co;2-n, 1997.

Webb, B. W., Clack, P. D., and Walling, D. E.: Water-air temperature relationships in a Devon river system and the role of flow, Hydrol. Process., 17, 3069–3084, https://doi.org/10.1002/hyp.1280, 2003.

Webb, B. W., Hannah, D. M., Moore, R. D., Brown, L. E., and Nobilis, F.: Recent advances in stream and river temperature research, 22, 902–918 https://doi.org/10.1002/hyp.6994, 2008.

Wenger, S. J., Isaak, D. J., Dunham, J. B., Fausch, K. D., Luce, C. H., Neville, H. M., Rieman, B. E., Young, M. K., Nagel, D. E., Horan, D. L., and Chandler, G. L.: Role of climate and invasive species in structuring trout distributions in the interior Columbia River Basin, USA, Can. J. Fish. Aquat. Sci., 68, 988–1008, https://doi.org/10.1139/f2011-034, 2011.

Werner, A. T., Schnorbus, M. A., Shrestha, R. R., Cannon, A. J., Zwiers, F. W., Dayon, G., and Anslow, F.: A long-term, temporally consistent, gridded daily meteorological dataset for northwestern North America, Scientific Data, 6, 180299, https://doi.org/10.1038/sdata.2018.299, 2019.

White, B. W. and Rosenblatt, F.: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms, Am. J. Psychol., 76, 705–707, https://doi.org/10.2307/1419730, 1963.

Wickham, H.: ggplot2: Elegant Graphics for Data Analysis, Springer-Verlag, New York, USA, available at: https://ggplot2.tidyverse.org (last access: 7 August 2020), 2016.

Willmott, C. J.: On the validation of models, Phys. Geogr., 2, 184–194, https://doi.org/10.1080/02723646.1981.10642213, 1981.

Xiang, Z., Yan, J., and Demir, I.: A Rainfall-Runoff Model With LSTM-Based Sequence-to-Sequence Learning, Water Resour. Res., 56, e2019WR02532, https://doi.org/10.1029/2019WR025326, 2020.

Yang, D. and Peterson, A.: River water temperature in relation to local air temperature in the Mackenzie and Yukon basins, Arctic, 70, 47–58, https://doi.org/10.14430/arctic4627, 2017.

Yazidi, A., Goyal, R., Paes, A., Gruber, N., De, N. G., and Jockisch, A.: Are GRU Cells More Specific and LSTM Cells More Sensitive in Motive Classification of Text?, Front. Artif. Intell., 3, 40, https://doi.org/10.3389/frai.2020.00040, 2020.

Zentralanstalt für Meteorologie und Geodynamik: ZAMG homepage, available at: https://www.zamg.ac.at, last access: 26 May 2021.

Zhilinskas, A. G.: Single-step Bayesian search method for an extremum of functions of a single variable, Cybernetics, 11, 160–166, https://doi.org/10.1007/BF01069961, 1975.

Zhu, S. and Piotrowski, A. P.: River/stream water temperature forecasting using artificial intelligence models: a systematic review, Acta Geophysica, 1–10, Springer, https://doi.org/10.1007/s11600-020-00480-7, 2020.

Zhu, S., Nyarko, E. K., and Hadzima-Nyarko, M.: Modelling daily water temperature from air temperature for the Missouri River, PeerJ, 6, e4894, https://doi.org/10.7717/peerj.4894, 2018.

Zhu, S., Hadzima-Nyarko, M., Gao, A., Wang, F., Wu, J., and Wu, S.: Two hybrid data-driven models for modeling water-air temperature relationship in rivers, Environ. Sci. Pollut. R., 26, 12622–12630, https://doi.org/10.1007/s11356-019-04716-y, 2019a.

Zhu, S., Heddam, S., Nyarko, E. K., Hadzima-Nyarko, M., Piccolroaz, S., and Wu, S.: Modeling daily water temperature for rivers: comparison between adaptive neuro-fuzzy inference systems and artificial neural networks models, Environ. Sci. Pollut. R., 26, 402–420, https://doi.org/10.1007/s11356-018-3650-2, 2019b.

Zhu, S., Heddam, S., Wu, S., Dai, J., and Jia, B.: Extreme learning machine-based prediction of daily water temperature for rivers, Environ. Earth Sci., 78, 202, https://doi.org/10.1007/s12665-019-8202-7, 2019c.

Zhu, S., Nyarko, E. K., Hadzima-Nyarko, M., Heddam, S., and Wu, S.: Assessing the performance of a suite of machine learning models for daily river water temperature prediction, PeerJ, 7, e7065, https://doi.org/10.7717/peerj.7065, 2019d.