Leveraging LSTM Embeddings for River Water Temperature Modeling

Benjamin Fankhauser[1][0000−0002−7982−2669], Vidushi Bigler[2][0000−0001−6043−8264], and Kaspar Riesen[1][0000−0002−9145−3157]

[1] Institute of Computer Science, University of Bern, Bern, Switzerland
{benjamin.fankhauser,kaspar.riesen}@unibe.ch
[2] Institute for Optimisation and Data Analysis, Bern University of Applied Sciences, Biel, Switzerland
vidushi.bigler@bfh.ch

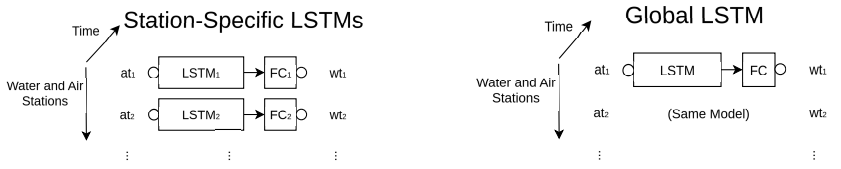
Abstract. River water temperature modeling is a major task in climate research. State-of-the-art methods for water temperature modeling deploy a transductive design, which makes it difficult to generalize to unseen water stations during test time. In the present paper, we isolate one of the common building blocks – a central LSTM, trained for each water station – and propose an embedding scheme in order to increase both the prediction accuracy and the amount of shared parameters and thus the generalization. The proposed embeddings are learned during training time. In an empirical evaluation we show that our method is able to reduce the RMSE by about five percentage points compared to the state-of-the-art reference method while decreasing the tuneable parameters by several orders of magnitude. We also provide a sample analysis of the embedding space of the catchment area of one specific river. Looking at the results of this qualitative analysis, we come to the conclusion that deploying an embedding in water temperature models is not only convincing to decrease the RMSE of water temperature predictions, but also enables better explainable deep learning models. Moreover, the proposed embedding technique opens up various unexplored applications in water temperature research.

Keywords: Water Temperature · Embedding · LSTM · Recurrent Neural Network

1 Introduction

Rivers and their tributaries can be found everywhere in habitable areas. River water temperature has a large impact on various biological processes in our ecosystems [1, 2], and thus on flora and fauna, agriculture, or our drinking water. Therefore, water temperature in rivers, and in particular its control and prediction, plays an important role in future climate change.

In general, water temperature is influenced by solar radiation heating the river bed, atmospheric exchange at the surface, and friction. Other important factors are snow melt, rain, ground water inflow or waste water inflows of large



(a) (b)

Fig. 1: (a) Transductive station-specific LSTMs, where every water station has its own tuned neural network, which implies that the water station has to be present during training. (b) Inductive global LSTM where all network weights are shared and generalization to unseen water stations is possible. (Note that at_k and wt_k denote the air and water temperature time series from the k-th water station and its corresponding weather station, respectively.)

cities. Air temperature and – if available – also discharge are usually used as exogenous variables to model water temperature [3].

In order to monitor water temperature increase due to climate change, the *Federal Office for the Environment of Switzerland* (FOEN) is running a water temperature monitoring for more than 40 years (with a doubling of the number of water stations since 2010). The water stations are equipped with a water temperature sensor and they monitor the discharge as well. The *Swiss Meteorological Institute* (MeteoSwiss) provides air temperature and other meteorological measurements from weather stations in the vicinity of such water stations. Recently, the data stemming from both water and weather stations was augmented with data from a *geographic information system* (GIS) in order to model the river segments between such stations as a graph and published under the name *Swiss River Network* [4].

For modeling water temperature based on air temperature, various methods have been proposed in the literature. Physically inspired methods derive models with few parameters, which are in turn tuned using statistical methods [5]. Other recent methods are based on deep learning such as, for instance, the *long short-term memory* (LSTM). The LSTM is a *recurrent neural network* (RNN) which has been successfully applied to many time series problems [6] (including the task of water temperature modeling [7–9]). More sophisticated neural network architectures also take into account neighboring water stations [10–12], modeling the river network as a graph, with nodes representing the water stations and edges representing the river sections.

The above mentioned deep learning methods perform well and provide accurate predictions of the water temperature for practical purposes[3]. However, most of these models are constructed in a way, where water temperature is only considered at single points in the river. In graph learning terminology, they are transductive learning methods, where all nodes have to be available at training

[3] In practice, an RMSE under one degree Celsius is considered a "good model".

time, providing no generalization to unseen nodes at test time [13] (see left side of Fig. 1). This limitation is problematic, as local characteristics of a water station can indeed be learned, but the model does not necessarily generalize to other river segments (with different topological or meteorological characteristics).

As a straightforward way to implement a deep learning method in an inductive way, one could apply one single LSTM to all water stations (see right side of Fig. 1). This method is designed to learn a model which can estimate the water temperature in the entire Swiss River Network, based on the air temperature in the water station's vicinity. However, in preliminary experiments we observe that this global model can not reach the same accuracy as the transductive methods, which train one station-specific LSTM for every single water station. In the present work we bridge the performance gap between the transductive and inductive methods by proposing an embedding based method for water temperature prediction. For the proposed method, we use a single global LSTM for the whole Swiss River Network and move the transductive property into a low dimensional embedding. Due to the small size of the embedding space, the global network is required to generalize to different river sections.

The remainder of this paper is structured the following way. In Section 2, we describe the problem of water temperature modeling more formally and describe the proposed method in detail. In Section 3, we thoroughly assess the proposed method with an empirical evaluation. As an additional outcome, we discuss the emerged embedding space as well. Finally, we draw conclusions and propose future research activities in Section 4.

2 Concatenation Embeddings in an LSTM

2.1 Problem Definition

The problem of water temperature modeling based on air temperature is defined as follows. For a given time series of an air temperature $a_k^{(1)}, ..., a_k^{(T)}$, the task is to find a model f_k which predicts the water temperature $\hat{y}_k^{(t)}$ at time step t (with $1 \leq t \leq T$) at the water station k. Formally, we seek a function f_k

$$f_k(a_k^{(1)}, ..., a_k^{(t)}) = \hat{y}_k^{(t)}, \quad \forall t : 1 \leq t \leq T \tag{1}$$

which can predict the actual water temperature $y_k^{(t)}$ of the k-th station at time t as accurately as possible.

2.2 Concatenation Based LSTM Embedding

The proposed method deploys a station-specific n-dimensional embedding $e_k \in \mathbb{R}^n$ in an LSTM. The embedding $e_k \in \mathbb{R}^n$ denotes the embedding for the k-th water station and can be represented as a point in the embedding space $\mathbb{R}^n$. The embeddings for each station are learned using a gradient descent technique during the training phase and represent the only transductive part of our model. In Fig. 2 the proposed architecture is illustrated.

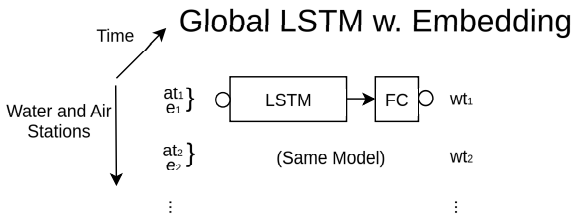


Fig. 2: The proposed LSTM with concatenation based embeddings. The LSTM is shared among all water stations but can encode station-specific characteristics in the embedding e_k.

In order to deploy the embedding to an LSTM, we concatenate the embedding e_k to the input $a_k^{(t)}$ at any time step t. Formally, the input $x^{(t)}$ to the LSTM is defined by

$$x^{(t)} := a_k^{(t)} || e_k \ , \tag{2}$$

where $||$ denotes concatenation.

At first glance, this behavior might appear cumbersome, as e_k is the same static value at any time step. Yet, it turns out that this method allows the embedding to influence each gate of the LSTM. By factoring out the multiplications with the embedding, this method corresponds to the following LSTM instance:

$$\begin{aligned}
\text{concat_lstm}(a_k^{(t)}, e_k) := LSTM(a_k^{(t)} || e_k) : \\
f^{(t)} &= \sigma(\boldsymbol{W}_f a_k^{(t)} + \boldsymbol{V}_f e_k + \boldsymbol{U}_f h^{(t-1)} + \boldsymbol{b}_f) \\
i^{(t)} &= \sigma(\boldsymbol{W}_i a_k^{(t)} + \boldsymbol{V}_i e_k + \boldsymbol{U}_i h^{(t-1)} + \boldsymbol{b}_i) \\
o^{(t)} &= \sigma(\boldsymbol{W}_o a_k^{(t)} + \boldsymbol{V}_o e_k + \boldsymbol{U}_o h^{(t-1)} + \boldsymbol{b}_o) \\
\tilde{c}^{(t)} &= \theta(\boldsymbol{W}_c a_k^{(t)} + \boldsymbol{V}_c e_k + \boldsymbol{U}_c h^{(t-1)} + \boldsymbol{b}_c) \\
c^{(t)} &= f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot \tilde{c}^{(t)} \\
h^{(t)} &= o^{(t)} \odot \theta(c^{(t)}) \\
\hat{y}^{(t)} &= MLP(h^{(t)})
\end{aligned} \tag{3}$$

Where $\sigma(z)$ denotes the sigmoid activation function and $\theta(z)$ the tanh function. The variables h and c are propagated through time. Bold symbols denote the learnable network weights. The concatenation embedding introduces four matrix multiplications with $\boldsymbol{V}_f$, $\boldsymbol{V}_i$, $\boldsymbol{V}_o$, $\boldsymbol{V}_c$. All matrices are globally trained weights and only e_k depends on the water station k. This static product is then added to the corresponding gate and can (theoretically) influence each gate in each hidden dimension.

The last line of Expression 3 is the final step to project the hidden state to a one dimensional water temperature prediction $\hat{y}^{(t)}$ at time step t. In our model we use a multi layer perceptron (MLP) to accomplish this final prediction.

3 Experimental Evaluation

3.1 Experimental Setup

For our experiment we use the Swiss River Network dataset G_{2010} [4]. As there are fewer weather stations than water stations, we select one water station per weather station (to mitigate an unfair disadvantage in the global reference system, where the same air temperature time series would be mapped to different water temperatures). In total the resulting dataset consists of 42 water stations and corresponding weather stations acquired during the years 2010 to 2021. The temporal resolution are daily averages.

Data from years 2010 to 2018 is used as training set, with a 90/10% validation split. The validation data is only used for model selection. Data from the two years 2019 and 2020 is used to define the hold-out test set and we only report numbers on these two years.

To measure and compare our predictions, we compute the widely used *Root Mean Squared Error* (RMSE), and we also report the *Mean Average Error* (MAE) and the *Nash-Sutcliffe model Efficiency Coefficient* (NSE) as defined in Table 1. While values closer to 0 for the two errors indicate good prediction quality, values of the NSE closer to 1 indicate a better model.

Table 1: The metrics used for evaluation. $y^{(t)}$ is the ground truth value and $\hat{y}^{(t)}$ our prediction.

RMSE	MAE	NSE
$\sqrt{\frac{1}{T}\sum_{t=1}^{T}(y^{(t)}-\hat{y}^{(t)})^2}$	$\frac{1}{T}\sum_{t=1}^{T}\lvert y^{(t)}-\hat{y}^{(t)}\rvert$	$1-\frac{\sum_{t=1}^{T}(y^{(t)}-\hat{y}^{(t)})^2}{\sum_{t=1}^{T}(y^{(t)}-\bar{y})^2}$

3.2 Training and Hyperparameter Tuning

In order to find the best performing embedding, we compute the gradients of the embedding e_k with respect to the training loss and simultaneously adjust the embedding to the weights of the LSTM during the training phase. We use a grid search on the hidden dimension of the LSTM, the amount of stacked LSTMs, the learning rate for the Adam optimizer [14], as well as the amount of an L2 regularization. We then freeze the network parameters and the embedding and select the model with the lowest validation loss. Note that we fix the dimension of the embedding space to $n = 5$ for our evaluations (in Section 3.6 we analyze the embedding space in more detail).

3.3 Reference Systems

Current state-of-the-art methods build on top of a central LSTM and add neighboring water stations to their model [4, 10]. Our goal is to improve this central

LSTM. Therefore, we compare our novel model against the two following LSTM based reference methods without the use of neighboring water stations[4] (outlined in Fig. 1).

1. **Station-Specific LSTMs**: The station-specific LSTM reference system deploys a full LSTM_k for each available water station k (as introduced in [8]). We deploy a separate grid search for each trained LSTM_k.

2. **Global LSTM**: The global LSTM reference system is one single LSTM, as described in Section 2.2 without the introduced embedding. The same parameters are used and tuned for all available water stations simultaneously and thus we refer to it as global LSTM.

3.4 Growth of Tuneable Parameters

The global LSTM is a *Vanilla LSTM* and has

$$P_g = 4d(hi + hh + h)$$

tuneable parameters, where i refers to the input size, d is the number of stacked LSTMs and h refers to the size of the hidden dimension for the four gates. Using K station-specific LSTMs leads to

$$P_s = K * 4d(hi + hh + h) \tag{4}$$

tuneable parameters.

The proposed method introduces additional parameters to the *Vanilla LSTM*, and uses tuneable parameters to store the embeddings. Therefore our method leads to an LSTM with

$$P_{our} = 4d(hi + hn + hh + h) + Kn \tag{5}$$

tuneable parameters.

We analyze the growth of the tuneable parameters by computing the partial derivative with respect to the variable in question. With respect to the embedding size n, the following increase in tuneable parameters of our model results:

$$\frac{P_{our}}{\partial n} = 4dh + K \ .$$

This means that one additional dimension in the embedding space adds an additional column to the four weight matrices in the LSTM gates (d-times) plus one tuneable parameter for each of the K water stations. As our embedding space n is usually small, this growth is almost negligible.

[4] In future work, more sophisticated methods can then replace their central LSTM with our proposed method.

Table 2: The average metrics reported on the hold-out test set over 42 water stations.

	Method	Metric		
		RMSE	MAE	NSE
Reference	Station-Specific LSTMs	0.80	0.62	0.95
	Global LSTM	1.54	1.29	0.79
Ours	Global LSTM w. Embedding	**0.76**	**0.59**	**0.96**

The number of water stations K is given by the application and is not subject to our control. The partial derivatives with respect to K of our method P_{our} and the station-specific reference system P_s are

$$\frac{P_{our}}{\partial K} = n \quad \text{and} \tag{6}$$

$$\frac{P_s}{\partial K} = 4dh(i + h + 1), \quad \text{respectively.} \tag{7}$$

Hence, we conclude that when the embedding size n is small, our method provides a parameter growth (Eq. 6) that is orders of magnitude smaller compared to the station-specific parameter growth (Eq. 7).

3.5 Empirical Results

The test results are obtained after an exhaustive grid search and controlled model convergence. We report the metrics on the hold-out test set, after the validation set was used for model selection. Table 2 shows the averages of the reported metrics (RMSE, MAE, and NSE) measured over all 42 test stations.

We observe that the station-specific LSTMs – as expected and already known [4, 8] – provide accurate predictions with an RMSE well below 1.0. At the same time, it is clear that the naïve approach of training a global LSTM for all stations does not work sufficiently well and provides values that cannot be used in practice. The proposed method not only closes the performance gap between the station-specific and the global reference systems, but decreases the RMSE even further. That is, compared to the station-specific reference system our model improves the RMSE by about 5 percentage points (improvements on the previous state of the art are observed in the other two metrics as well).

Fig. 3 shows the distribution of the RMSEs of all 42 stations stemming from the hold-out test set as box plots. This illustration shows quite nicely that there are large differences in the RMSE among the stations (regardless of which model is used). For example, we observe that in all three systems there are stations that have an RMSE above 1. However, for the station-specific and our novel model

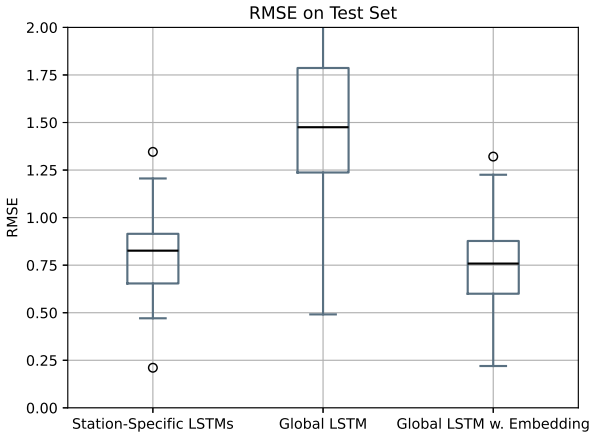


Fig. 3: RMSE on the test set, illustrated with boxplots of the same 42 water stations. Some stations are more difficult to model than others.

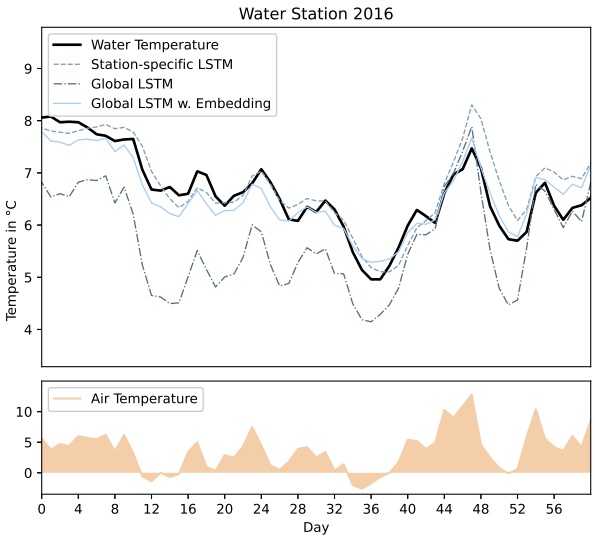


Fig. 4: Sample result at one specific water station of the Swiss River Network over a randomly selected test set period of 60 days. The three thin lines of the plot refer to the predicted water temperatures of the three methods. The thick black line refers to the ground truth water temperature. The lower graph, with the same x-axis, shows the air temperature over the same time period.

Table 3: The number of tuneable parameters of the selected models in the empirical evaluation to serve 42 water stations.

	Method	Parameters
Reference	Station-Specific LSTMs	723,394
	Global LSTM	1,513
Ours	Global LSTM w. Embedding	**1,307**

using an embedding, the RMSE values for the clear majority of stations are below this 1 degree threshold (while the global LSTM dramatically loses performance compared to the station-specific methods).

The lack of performance of the global LSTM is also visible in the illustration of Fig. 4. In this qualitative analysis we randomly select a test period of 60 days at one specific water station. We choose a station in the flatlands, where the river already accumulated to a bigger stream, and thus, the water temperature is less sensitive to small changes in the air temperature. In Fig. 4 it is clearly visible that the station-specific and embedding based methods can adapt to this behavior. The global LSTM, on the other hand, tends to exaggerate influences of air temperature. For example, around days 12, 36, and 52, the global LSTM exaggerates the effect of a drop in air temperature. Moreover, it is also visible that the global LSTM is generally further away from the ground truth water temperature than the other two models.

In Section 3.4, we discuss that our method has a more beneficial parameter growth in comparison to the station-specific reference system when deploying it to multiple water stations. In Table 3, we sum up the amount of tuneable parameters of the best performing models. While station-specific models require a total of more than 700K parameters to be optimized, we only observe about 1,300 parameters in our novel approach (including the embedding).

3.6 Analysis of the Embedding Space

One follow up question to our method is the size of the embedding space n. In an additional experiment, we re-run our grid search and evaluate various embedding sizes ($n \in \{1, 2, 3, 5, 10, \ldots, 30\}$). Fig. 5 shows the RMSE on the y-axis with respect to the embedding size n (on the x-axis). The RMSE plateaus at an embedding dimension of $n = 10$, but even a very low dimensional embedding space with $n = 1$ or $n = 2$ is beneficial compared to the station-specific baseline model.

We also deploy a two dimensional embedding space and train a new model on all water stations of the Rhine and its tributaries in Switzerland (the Rhine is selected for this analysis as it has the largest catchment area). This allows us

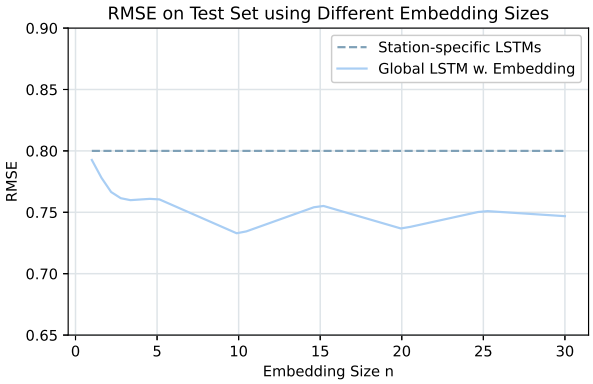


Fig. 5: RMSE with respect to the embedding size n compared to the station-specific LSTMs reference system.

to visually analyze spatial properties of all stations in the complete catchment area in a two-dimensional space (see Fig. 6).

The embedding space is unrestricted and the optimization scheme can place embeddings arbitrary. However, four natural clusters (A, B, C, and D) are visually identifiable in the embedding in Fig. 6:

- Cluster A: Stations of this cluster refer to the outflow of the alpine lakes Walensee (2104), Brienzersee (2457) and Thunersee (2030).
- Cluster B: This cluster contains mostly big rivers in the flatlands, like the Rhine in Basel (2091), or the Aare in Brugg (2016).
- Cluster C: This cluster is composed of stations near the outflow of the flatland lakes Bielersee (2029) or Bodensee (2288).
- Cluster D: This cluster is a collection of various tributaries in hilly, prealpine landscapes relatively far away from each other (like the Gürbe at Belp (2159), Murg at Wängi (2126), or Linth at Mollis (2372)).

Further away from the origin, we observe various alpine water stations (mainly leaf nodes), that do not belong to a distinct cluster.

4 Conclusion and Future Work

Many ecosystems depend on the well-being of rivers. Hence, monitoring river water temperature plays an important role in research of future climate change. After revisiting state-of-the-art methods for water temperature prediction, we observe that a vast majority of these methods (e.g. [4, 7, 8]) have a transductive design, meaning that all water stations have to be available during training time. These designs do not share parameters and the method has to learn basic water temperature characteristics at every station from scratch (leading to large amounts of tuneable parameters). We propose to address this issue with an

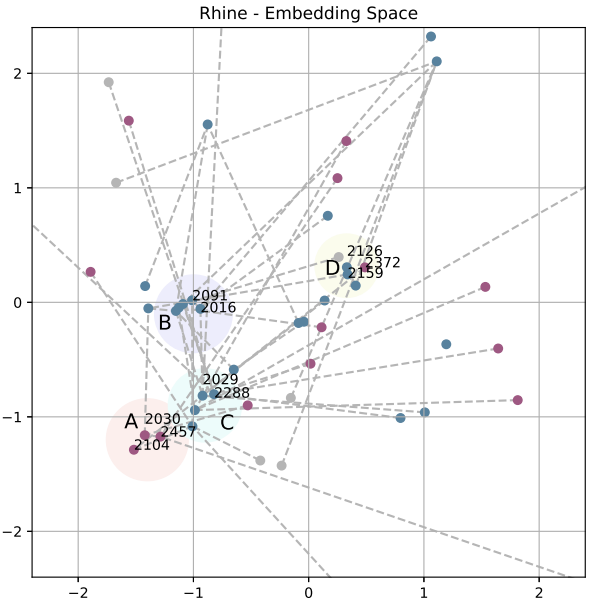


Fig. 6: Embedding space visualization of the Rhine catchment area. Dotted edges represent neighboring stations with respect to the water flow [4]. Thus, leaf nodes denote the first water station after the source of the stream. Blue dots denote water stations in the flatland while red dots are water stations from alpine regions. Four clusters (A, B, C, and D) emerge from the data.

embedding scheme. This embedding is designed so that it can be seamlessly integrated into more sophisticated state-of-the-art methods.

In an empirical evaluation, we demonstrate that our method outperforms the station-specific LSTMs while using two orders of magnitude less tuneable parameters. Moreover, by changing the dimensions of the embedding space, our method allows for flexible tuning between shared parameters and station-specific characteristics. Further, we show that the method is not sensitive to the size of the embedding space, and various configurations lead to good results. Last but not least, we are also able to detect naturally emerging clusters in the embedding space.

We see several possible future research activities. For instance, we can embed even more data or enforce constraints in the embedding space. Another idea is to use embeddings for few shot learning. In this setting, only a few months of water temperature data needs to be available. This could be particularly interesting as we cannot wait until we have measured a decade's worth of water temperatures before developing new models to account for ongoing climate change.

Acknowledgments. This project is supported by the Swiss National Science Foundation (SNSF) Grant Nr. PT00P2 206252. Data are kindly provided by the Federal

Office for the Environment and MeteoSwiss. Calculations were performed on UBELIX (https://www.id.unibe.ch/hpc), the HPC cluster at the University of Bern.

References

1. Lessard, J.L., Hayes, D.B.: Effects of elevated water temperature on fish and macroinvertebrate communities below small dams. River research and applications 19(7), 721–732 (2003)
2. Caissie, D.: The thermal regime of rivers: a review. Freshwater biology 51(8), 1389–1406 (2006)
3. Piccolroaz, S., Calamita, E., Majone, B., Gallice, A., Siviglia, A., Toffolon, M.: Prediction of river water temperature: a comparison between a new family of hybrid models and statistical approaches. Hydrological Processes 30(21), 3901–3917 (2016)
4. Fankhauser, B., Bigler, V., Riesen, K.: Graph-based deep learning on the swiss river network. In: International Workshop on Graph-Based Representations in Pattern Recognition. pp. 172–181. Springer (2023)
5. Toffolon, M., Piccolroaz, S.: A hybrid model for river water temperature as a function of air temperature and discharge. Environmental Research Letters 10(11), 114011 (2015)
6. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
7. Piotrowski, A.P., Napiorkowski, M.J., Napiorkowski, J.J., Osuch, M.: Comparing various artificial neural network types for water temperature prediction in rivers. Journal of Hydrology 529, 302–315 (2015)
8. Qiu, R., Wang, Y., Rhoads, B., Wang, D., Qiu, W., Tao, Y., Wu, J.: River water temperature forecasting using a deep learning method. Journal of Hydrology 595, 126016 (2021)
9. Moshe, Z., Metzger, A., Elidan, G., Kratzert, F., Nevo, S., El-Yaniv, R.: Hydronets: Leveraging river structure for hydrologic modeling. arXiv preprint arXiv:2007.00595 (2020)
10. Jia, X., Zwart, J., Sadler, J., Appling, A., Oliver, S., Markstrom, S., Willard, J., Xu, S., Steinbach, M., Read, J., et al.: Physics-guided recurrent graph model for predicting flow and temperature in river networks. In: Proceedings of the 2021 SIAM International Conference on Data Mining (SDM). pp. 612–620. SIAM (2021)
11. Chen, S., Zwart, J.A., Jia, X.: Physics-guided graph meta learning for predicting water temperature and streamflow in stream networks. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2752–2761 (2022)
12. Fankhauser, B., Bigler, V., Riesen, K.: Impute water temperature in the swiss river network using lstms. In: International Conference on Pattern Recognition Applications and Methods. pp. 732–738. Scitepress (2024)
13. Hamilton, W.L.: Graph representation learning, chap. 5.1.1. Morgan & Claypool Publishers (2020)
14. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)