Spatio-Temporal Graph Neural Networks for 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 – a time series problem where spatial relations matter – is important to understand our environment. Currently two research directions are present to tackle this problem, viz. Recurrent Neural Networks, namely long short-term memory (LSTM), and Graph-based approaches, which exploit the natural tree structure of rivers. In the present paper, we extend a state-of-the-art LSTM method for water temperature modeling with a Graph Convolutional Network and a Graph Isomorphism Network. This novel combination results in a spatio-temporal neural network which can be applied for node predictions in any graph having nodes with unique identifiers. In the present paper, we apply the novel procedure on the Swiss River Network (a data set with decades of measurements of river temperature and atmospheric variables in Switzerland). In an experimental evaluation we show that the proposed method is robust in convergence and improves the state-of-the-art result by several percentage points in terms of Root Mean Squared Error.

Keywords: River Water Temperature · LSTM · Recurrent Neural Network · GCN · GIN · Graph Neural Network · Spatio Temporal Model

1 Introduction

River water temperature is an important variable in our ecosystem, as a large number of ecological processes are heavily dependent by it. Mainly higher river water temperatures are critical as the hydrological ecosystems are sensitive to raising water temperatures [1]. In the worst case, such a rise results in the extinction of species as well as the deterioration of water quality [2]. The present paper researches novel approaches for water temperature modeling.

An accurate and elaborated modeling of the river water temperature has two benefits. First, with future air temperature projections, one can model river water temperature more precisely and in turn detect river sections with critical water temperature more precisely. Second, having more sophisticated models allows us to adapt them to more complex problems, such as, for instance, the investigation of the effects of anthropological buildings on the river infrastructure [3,4].

River water temperature modeling is an interesting real world application for machine learning as there are several non linear effects contributing together. Solar radiation, for instance, is absorbed by the river bed or particles in the water, which has a major influence on the temperature. The river bed itself creates friction (which also leads to heat). Snow melt, ground water inflow, city sewage, or rain water also influence the river water temperature [5].

The *Federal Office for the Environment of Switzerland* (FOEN) is running a water temperature monitoring for several decades in many rivers of Switzerland. Additionally, the *Swiss Meteorological Institute* (MeteoSwiss) measures air temperature and other atmospheric variables in the vicinity of these water stations. Recently, data from these water and weather stations in combination with their connectivity has been published as the *Swiss River Network* data set [6]. This data set is unique, yet somehow related to other data sets, e.g. data for rain-fall run-off models [7], or weather parameter prediction [8].

For modeling the water temperature, the use of the air temperature as well as the discharge as predictor variables has been proposed. For instance, physically inspired methods, like Air2Stream [9], show success based on statistical models using these variables. More recently, the application of long short-term memory (LSTM) [10,11] has been proposed to further improve the prediction precision. LSTMs are a special type of Recurrent Neural Network (RNN) and are used to model time series [12]. Also other deep learning architectures, some taking the neighboring relation of water stations into account, provide competitive results [13,14]. In order to model neighboring water stations, the use of graph structures, where the nodes represent water stations and the edges represent relations between stations based on the river sections, has also been proposed [15].

In the present work, we propose to combine one of the latest RNN methods for water temperature modeling [16] with Graph Neural Networks (GNNs). The proposed procedure is particularly designed for node predictions in a node-with-id network, namely a graph where each node has a unique identifier and is thus permutation invariant by definition.

The remainder of this paper is structured as follows. In Section 2, we formally introduce the problem of temperature modeling. Additionally, we briefly review the latest RNN methods for water temperature modeling and the GNNs actually used in our procedure. In Section 3, we introduce the proposed spatio-temporal neural network suited for a node-with-id network like the Swiss River Network. Section 4 contains a thorough evaluation of the proposed method. Finally, we draw conclusions and propose future research directions in Section 5.

2 Related Work

In this work we focus on water temperature modeling based on air temperature. Formally, for a given time series of T air temperatures $at_k^{(1)}, ..., at_k^{(T)}$ at a specific water station indexed k, the goal is to model f_k with

$$f_k(at_k^{(1)}, ..., at_k^{(t)}) = \hat{wt}_k^{(t)}, \quad \forall t \in \{1, \ldots, T\}$$

so that $\hat{wt}_k^{(t)}$ is the predicted water temperature at time step t (with $1 \leq t \leq T$) at water station k.

The method proposed in this paper combines two state-of-the-art directions in water temperature modeling, namely RNNs and GNNs which are reviewed in the next two subsections.

2.1 LSTM Based Water Temperature Modeling

LSTM (a special type of an RNN) has been deployed to many time series problems [12]. The *Station-Specific LSTM* [6,10] for water temperature modeling, for instance, uses one vanilla LSTM for each available water station in order to model f_k. The drawback of this approach is that it results in many models to converge and maintain.

Water temperature modeling LSTMs have recently been extended with an embedding per water station [16], termed LSTM-E. Formally, the LSTM-E method uses the following algorithm:

$$\text{LSTM-E}(at_k^{(t)}, e_k) := LSTM(at_k^{(t)} || e_k):$$

$$\begin{aligned}
f^{(t)} &= \sigma(\boldsymbol{W}_f a_k^{(t)} + \boldsymbol{V}_f e_k + \boldsymbol{U}_f h^{(t-1)} + \boldsymbol{b}_f) \\
i^{(t)} &= \sigma(\boldsymbol{W}_i a_k^{(t)} + \boldsymbol{V}_i e_k + \boldsymbol{U}_i h^{(t-1)} + \boldsymbol{b}_i) \\
o^{(t)} &= \sigma(\boldsymbol{W}_o a_k^{(t)} + \boldsymbol{V}_o e_k + \boldsymbol{U}_o h^{(t-1)} + \boldsymbol{b}_o) \\
\tilde{c}^{(t)} &= \theta(\boldsymbol{W}_c a_k^{(t)} + \boldsymbol{V}_c e_k + \boldsymbol{U}_c h^{(t-1)} + \boldsymbol{b}_c)
\end{aligned}$$

$$\begin{aligned}
c^{(t)} &= f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot \tilde{c}^{(t)} \\
h^{(t)} &= o^{(t)} \odot \theta(c^{(t)}) \\
\hat{wt}^{(t)} &= MLP(h^{(t)})
\end{aligned}$$

As in the vanilla LSTM, $\sigma(z)$ denotes the sigmoid activation function and $\theta(z)$ is the tanh function. The two variables h and c are the hidden state and propagated through time. The bold symbols are the learnable network weights, which are shared among all water stations.

Roughly speaking, the LSTM-E corresponds to one single global LSTM where every water station k has an assigned embedding e_k which is in turn learned during training time. It has been shown that this method not only improves the predictive performance but also lowers the trainable parameters by several orders of magnitude compared to station-specific LSTMs [16].

2.2 Graph Neural Networks (GNNs)

In several contributions, it has been shown that combining information of neighboring water stations is beneficial to hydrological models [13,15,17]. In order to make use of the spatial connectivity between water stations, we use the graph introduced in the Swiss River Network [6] which actually corresponds to a node-with-id network.

Formally, a node-with-id network is a graph $G = (V, E)$ where V are the nodes and E the edges. Additionally, there is an identifier function $id : V \rightarrow \mathbb{N}$, which assigns a unique and ordinal identifier to each node on the network. This introduces a deterministic ordering of nodes in the graph. Note, however, that the graph still has an irregular structure, as the degree of each node can vary. Thus, GNNs with message passing mechanisms are suited to this data structure.

The generalized message passing algorithm expects a feature vector $\boldsymbol{x}_i$ at each node $v_i \in V$. In a first step, this feature vector is transformed by a function $g(\boldsymbol{x}_i)$ and then transmitted along the edges of node v_i. In a second step, each node v_i receives $|\mathcal{N}(v_i)|$ incoming messages and aggregates them to $\boldsymbol{z}_i$ (where $\mathcal{N}(v_i)$ refers to the set of neighboring nodes of v_i). In a third step, each node updates its own feature vector using a function $h(\boldsymbol{x}_i, \boldsymbol{z}_i)$, resulting in a new state at each node. The complete message passing process can be applied M-times (increasing the receptive field of each node). By adding an activation function in between executions of message passing we finally obtain a GNN [18].

More formally, Eq. 1 shows the generalized message passing framework for the update step of node v_i with its neighbor nodes $v_j \in \mathcal{N}(v_i)$. The aggregation function *aggr* is usually implemented as mean or sum (yet, other methods are available).

$$\boldsymbol{x}_i^{(m)} = h(\boldsymbol{x}_i^{(m-1)}, aggr_{v_j \in \mathcal{N}(v_i)}(g(\boldsymbol{x}_j^{(m-1)}, \boldsymbol{x}_i^{(m-1)}))) \quad (1)$$

In Eq. 1 we use index m to indicate the m-th execution of message passing. In the present work, we use two state-of-the-art implementations of the message passing framework, viz. *Graph Convolutional Networks* (GCN) [18] and *Graph Isomorphic Networks* (GIN) [19].

– The GCN [18] uses the following implementation of the message passing algorithm (with an edge weight of 1):

$$\boldsymbol{x}_i^{(m)} = \underbrace{\Theta^T}_{h} \underbrace{\sum_{v_j \in \mathcal{N}(v_i) \cup \{v_i\}}}_{aggr} \underbrace{\frac{1}{\sqrt{\hat{d}_j \hat{d}_i}} \boldsymbol{x}_j^{(m-1)}}_{g}$$

$$\hat{d}_i = 1 + deg(v_i)$$

where Θ are the learnable filter weights and the aggregation corresponds to the adjacency matrix with inserted self loops ($deg(v_i)$ refers to the degree of node v_i).

– The GIN [19] improves the expressiveness of GCN by approximating an injective multi set aggregation function using MLPs. In its reduced form it uses the following implementations of the message passing algorithm:

$$\boldsymbol{x}_i^{(m)} = \underbrace{MLP}_{h}\left((1+\epsilon)\boldsymbol{x}_i^{(m-1)} + \underbrace{\sum_{v_j \in \mathcal{N}(v_i)}}_{aggr} \underbrace{\boldsymbol{x}_j^{(m-1)}}_{g}\right)$$

As in the literature proposed we use an MLP during the node update step and ϵ is a free parameter.

3 Spatio-Temporal Nodes-with-Id Network

Major contribution of this paper is that we combine the RNN model LSTM-E (detailed in Section 2.1) and both GNN architectures GCN and GIN (detailed in Section 2.2) for water temperature modeling. In particular, we propose the following architecture that describes the method at water station k at time step t in the time series for $m \in \{1, \ldots, M\}$ message passing steps. (See also Fig. 1 which illustrates the general architecture of the proposed spatio-temporal network).

$$\boldsymbol{x}_k^{(t,0)} = \text{LSTM-E}(at_k^{(t)}, e_k) \qquad \text{, stacked } D\text{-times} \qquad (2)$$

$$\boldsymbol{x}_k^{(t,m)} = \text{GNN}(X^{(t,m-1)}, E) \qquad \text{, repeated } M\text{-times} \qquad (3)$$

$$\hat{wt}_k^{(t)} = \text{MLP}(\boldsymbol{x}_k^{(t,M)}) \qquad (4)$$

First (in Eq. 2), we apply the LSTM-E Method at each node of the network in order to transform the air temperature into a hidden state which has high predictive power. In Eq. 2 at_k is the air temperature at water station k and e_k refers to the station specific embedding. In a second step (in Eq. 3), we apply the message passing algorithm, so that the network can take information of its surrounding nodes into account and further refine the hidden state. This message passing step takes as input all the hidden states $\boldsymbol{x}_i \in X$ of all water stations and edges E of the Swiss River Network graph. In a last step (in Eq. 4), the hidden state is projected to a water temperature using linear transformations.

Note that the LSTM-E, GNN, MLP are all global models with shared parameters among all water stations, and only the embedding e_k is station-specific.

4 Experimental Evaluation

4.1 Experimental Setup

For our evaluation, we use the catchment area of the river Rhine of the Swiss River Network dataset G_{2010} [6]. This results in one connected component with

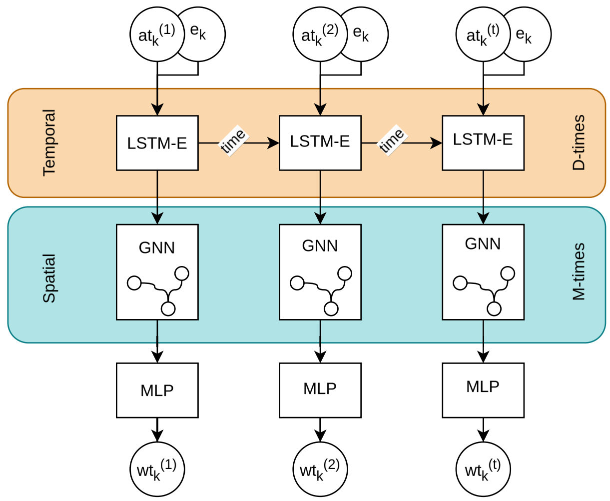

Fig. 1: The proposed method at water station k within a node-with-id network. The air temperature at_k is transformed using the LSTM-E Method [16], an embedding based LSTM. The hidden state is then collected in the graph structure of the Swiss River Network [6]. The GNN refines the hidden state using message passing among neighboring nodes. At the latest stage, an MLP predicts the final water temperature $\hat{wt}_k^{(t)}$ at time step t.

50 water stations and corresponding air temperature from the years 2010 to the end of 2020. We use daily averaged temperatures.[3]

Data from the years 2010 to 2018 are used as training set. From these years we use a 90/10% training / validation split. The validation set is used for model selection only, as we run an exhaustive grid search over the hyperparameters. Data from the two years 2019 and 2020 are used as test set and are not considered during training time. In this work we only report metrics obtained on the test set.

We compare the proposed method to two state-of-the-art systems, viz. the station-specific LSTM [10] and the embedding based LSTM [16].

- The *Station-Specific LSTM* uses one vanilla LSTM for each water station [10]. Each station-specific LSTM is trained in isolation and a grid search is run to find the best performing hyperparameters for each water station [6].
- The *Embedding LSTM* corresponds to the state-of-the-art RNN LSTM-E and does not use neighboring information, yet learned embeddings per station [16].

To compare and asses the predictive performance, we report the widely used *Root Mean Squared Error* (RMSE), the *Mean Average Error* (MAE) and the

[3] Code and data is made available under `https://swiss-river-network.github.io` .

Nash-Sutcliffe model Efficiency Coefficient (NSE) as defined in Table 1. For both RMSE and MAE, values closer to 0 and for the NSE values closer to 1 indicate better model performance.

Table 1: The three metrics used for evaluation of the methods ($wt^{(t)}$ is the ground truth value for the water temperature and $\hat{wt}^{(t)}$ refers to our prediction).

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

4.2 Training and Hyperparameter Tuning

As proposed in the LSTM-E method [16], we optimize the embeddings e_k during training time. We use a grid search on the learning rate, the embedding dimension, the amount of stacked LSTMs D, the amount of message passing steps M, the width of the hidden space used in the LSTM as well the GNN. Table 2 shows the search space of the grid search for all five parameters. As gradient descent based optimizer we use the Adam optimizer [20] and model selection is based on the lowest RMSE on the validation set.

4.3 Empirical Results

We report the metrics (RMSE, MAE, and NSE) on the hold-out test set in Table 3. The test set contains 50 water stations and we report the average values of each metric over all 50 stations. To encounter random artifacts, we rerun the grid search several times and report the standard deviation in brackets.

The proposed method that combines the LSTM-E with GNNs shows an improvement in all metrics for both instances of the GNN. If we compare our approach with the station-specific LSTM, we observe an improvement in the RMSE and MAE of around 10 percent (while the NSE is only slightly improved). When comparing with the Embedding LSTM, we observe only slight improvements (for the RMSE we see an absolute improvement of 0.01 and 0.02 for GCN and GIN, respectively) and for the MAE an absolute improvement of 0.01 is observed for both models). What is noticeable, however, is the significant decrease in the standard deviation for the GIN model. Compared to the Embedding LSTM, these deviations are approximately seven times, five times and 20 times smaller than for the Embedding LSTM (according to RMSE, MAE and NSE), which corresponds to a massive improvement in the robustness of the model.

Table 2: Space of the grid search among all embedding and graph based methods.

Hyperparameter	**Values**	
	Min	Max
Learning rate	0.005	0.01
Embedding dimensions	5	10
Stacked LSTMs D	1	2
Message passing steps M	1	2
Hidden space width	8	16

Table 3: The average metrics reported on the hold-out test set over 50 water stations. Brackets indicate the standard deviation after multiple runs of the grid search. For each metric the best result is marked in bold.

	Method	**Metric**		
		RMSE	MAE	NSE
Reference	Station-Specific LSTMs [10]	0.80	0.62	0.95
Reference	Embedding LSTM [16]	0.74 (± 0.007)	0.57 (± 0.005)	0.96 (± 0.002)
Ours	Spatio Temporal (GCN)	0.73 (± 0.012)	**0.56** (± 0.009)	**0.97** (± 0.001)
Ours	Spatio Temporal (GIN)	**0.72** (± 0.001)	**0.56** (± 0.001)	**0.97** (± 0.00001)

5 Conclusion

Water temperature of rivers plays an important role in future climate change and thus quite an effort is made in monitoring and prediction. However, further research is needed due to two reasons. First, to improve the predictive performance of the current models, and second to flexibly adapt the models to future tasks. In the present paper, we propose to use a state-of-the-art RNN in water temperature modeling and extend it with a graph based network using the message passing mechanism. This leads to a spatio-temporal neural network.

In an empirical evaluation we demonstrate that the use of the spatial information improves the modeling performance in three different metrics, setting a new state-of-the-art baseline in water temperature modeling. We also observe a beneficial improvement in robustness of model convergence.

Major goal of this paper is to improve the accuracy of water temperature modeling. Yet, working with graphs enables more flexible approaches than collecting data for a decade until an isolated LSTM converges. Moreover, graph modeling opens a variety of other research directions. The proposed framework might be suited, for instance, to model dynamic changes on the underlying river network (for example, if a new water station is built or a short-term measurement is made). Some parts of Switzerland do have more water stations, this means we can also nest finer resolutions into the current graph, or explore the relation between neighboring stations in more detail. Last but not least, the proposed method makes use of a node-with-id network, and thus it easily generalizes to other real world applications like transportation, computer, or social media networks.

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. Dahlke, F.T., Wohlrab, S., Butzin, M., Pörtner, H.O.: Thermal bottlenecks in the life cycle define climate vulnerability of fish. Science 369(6499), 65–70 (2020)
2. Caissie, D.: The thermal regime of rivers: a review. Freshwater biology 51(8), 1389–1406 (2006)
3. Reid, P.C., Hari, R.E., Beaugrand, G., Livingstone, D.M., Marty, C., Straile, D., Barichivich, J., Goberville, E., Adrian, R., Aono, Y., et al.: Global impacts of the 1980s regime shift. Global change biology 22(2), 682–703 (2016)
4. Woolway, R.I., Dokulil, M.T., Marszelewski, W., Schmid, M., Bouffard, D., Merchant, C.J.: Warming of central european lakes and their response to the 1980s climate regime shift. Climatic Change 142, 505–520 (2017)
5. 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)
6. 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)
7. Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M.: Rainfall–runoff modelling using long short-term memory (lstm) networks. Hydrology and Earth System Sciences 22(11), 6005–6022 (2018)
8. Zanetta, F., Nerini, D., Beucler, T., Liniger, M.A.: Physics-constrained deep learning postprocessing of temperature and humidity. Artificial Intelligence for the Earth Systems 2(4), e220089 (2023)
9. 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)
10. 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)
11. 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)

12. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural computation 9(8), 1735–1780 (1997)
13. 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)
14. 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)
15. Zhao, Q., Zhu, Y., Shu, K., Wan, D., Yu, Y., Zhou, X., Liu, H.: Joint spatial and temporal modeling for hydrological prediction. Ieee Access 8, 78492–78503 (2020)
16. Fankhauser, B., Bigler, V., Riesen, K.: Leveraging LSTM embeddings for river water temperature modeling (in review)
17. 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)
18. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net (2017), `https://openreview.net/forum?id=SJU4ayYgl`
19. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net (2019), `https://openreview.net/forum?id=ryGs6iA5Km`
20. Kingma, D.P., Ba, J.: Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)