SPECIFIC COMMENTS

1. Abstract: define terms CNN, LSTM, etc. at first use in the abstract.

We explained them in the abstract.

2. Introduction: First, why are deep learning methods needed for this problem more so than traditional machine learning methods (e.g., 3-layer multilayer perceptron, decision trees, support vector machine).

We added that: "There are some machine learning methods available for TSC such as K nearest neighbour and support vector machine. However, the focus of this research is on the deep learning models that have greatly impacted sequence classification problems and they can also be used for multivariate TSC with good performance. Deep learning methods are able to consider two-dimensionality in multivariate time-series and their deeper architecture could further improve the classification especially for complex problems, which is why their results are more accurate and robust than other methods (Wu et al., 2018a, April). However, they are more time consuming and difficult to interpret."

And also later we added in the text that: "Deep learning models perform well for the problems with complex-nonlinear dependencies, time dependencies, and multivariate inputs."

Second, there are numerous applications of CNN, LSTM, and their hybrid versions applied in hydrology (Althoff et al., 2021; Apaydin et al., 2020; Barzegar et al., 2021, 2020; Kratzert et al., 2018; Wunsch et al., 2020; Zhang et al., 2018). It would be good to mention the application of such models in hydrology to show that their use is well established within this domain and to highlight that none of these methods have been explored for ice jam prediction.

We added that: "There are numerous applications of CNN, LSTM, and their hybrid versions applied in hydrology (Althoff et al., 2021; Apaydin et al., 2020; Barzegar et al., 2021, 2020; Kratzert et al., 2018; Wunsch et al., 2020; Zhang et al., 2018). Although deep learning methods seem to be promising to address the requirements of ice-jam predictions, none of these methods yet have been explored for ice jam prediction."

3. L183: '…and multiplying it by tanh…' Is this fully correct? tanh is an activation function and therefore needs an input to evaluate, would it not be more correct to write '…and multiplying it by tanh(something)'?

That is right. It is tanh (cell state).

4. L217-219: do you mean that earlier experimentation showed that MinMaxScaler lead to the most accurate results?

My experiments showed that. We changed the phrase:" The results show that MinMaxScaler (Eq. (1)) leads to the most accurate results."

5. Figures: the vast majority of figures in this paper are taken from other sources. While the figures appear to be properly cited it may be worthwhile to consider creating some new figures specific to the dataset and models employed in this work.

I created and added figures 4, 5 and 6.

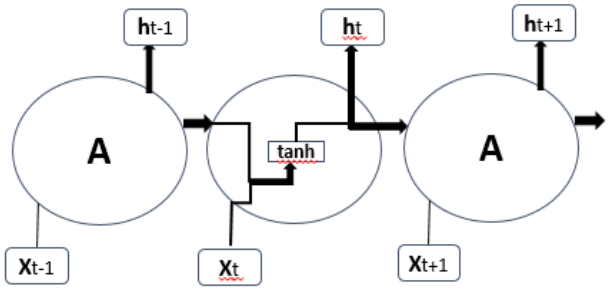

Figure 4. An RNN with a single tanh layer, where A is a chunk of the neural network, x is input data, and h is output data.

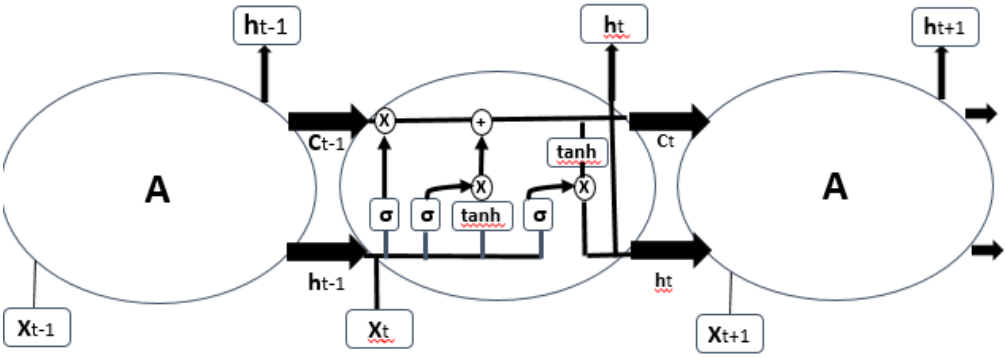

Figure 5. Structure of LSTM block with four interacting layers.

Figure 6. A convolution layer structure including two sets of filters.

6. The authors refer to 'loss' (i.e., a loss function) without defining it until section 2.5.1.5, where they then switch to the term 'cost function'. It would be good for the authors to: a) clarify early on that a loss function for neural networks is similar to an objective function for process-based hydrological models (to

make this term more approachable for a wide audience) and b) use consistent terminology (i.e., choose loss or cost function).

We selected the term loss function to be used in the text.

We added the phrase “The loss function is similar to an objective function for process-based hydrological models”.

7. Eqs. 2 and 3: specifically mention which equation pertains to the L1 and L2 regularization. The authors should also explicitly state the cost function (see also comment 6 above to ensure consistent terminology is adopted) or at least point to where it is discussed in more detail within the text.

We had explicitly stated that “The two main methods used to calculate the size of the weights are L1 (Eq. 2) and L2 or weight decay (Eq. 3)”.

We added that “Neural networks solve an optimization problem that requires a loss function to calculate the model error.”

We also explained that in more detail in “Network optimization” section in the Appendix: “Training CNN involves global optimization by defining a loss function to be minimized over training. For the classification task, the loss function of the models is calculated using categorical cross-entropy between network outputs and targets (Eq. (5)), where L is the loss, p is the prediction (probability), t is the target, and c is the number of classes. Then, the mean of the loss is computed over each mini-batch.

$$L = -\sum_{i=1}^{c=2} t_i \log(p_i)$$

8. Section 2.5.2: what do you use for identifying the optimal architecture? Grid-search, random-search, Bayesian optimization, …?

We were aware of other methods for hyperparameter tuning but we selected a manual trial and error method, as grid search experiments suffer from poor coverage in dimensions (Bergstra and Bengio, 2012) and manual experiments are much easier and more interpretable in investigating the effect of one hyperparameter of interest.

9. Eqs. 4-6: why are RELU and Sigmoid in italic font but not tanh? I believe the Sigmoid function is referred to earlier as σ, consistency should be maintained here (and elsewhere in the text, e.g., see comment 6 above).

WE edited that. I used σ in the figures and explained that in the corresponding line. In text it is mentioned as sigmoid.

10. L330: I do not think ‘mini-batch’ has been discussed yet, nor should it be assumed that most readers will be familiar with the term. What is a ‘mini-batch’ and what is its purpose?

We added: “Mini-batches split the training data into small batches which is used during each iteration one after the other to calculate model error and update model parameters. It is computationally efficient not having all training data in memory and model implementations, since batch size significantly influences the training time (Fawaz et al., 2019, July). Mini-batches cause the model to update more frequently resulting in a more robust convergence, and avoiding local minima.”

11. 2.5.3: see comments 6 and 7 above.

Ok. We edited that.

12. Much of sections 3.1 – 3.2 (and their sub-sections) do not belong in a Results and Discussion section, these sections are more related to Methodology or Model Development. Some sections such as 3.1.8 include results, but the majority of these sub-sections in 3.1 and 3.2 do not.

You are right. We moved them to materials and method section.

13. L517-518: the authors should include at least a single reference that corroborate this statement.

There is no reference for that. We changed my statement to “It is not clear that whether the order of input variables in the input file might influence multivariate TSC or not when using 2-D filters and 2-D max-pooling layers.

14. L547-551: no new information should be provided in the Conclusion section, why is project ‘DAVE’ not mentioned earlier in the paper? It would seem best to mention this information in the introduction of the paper to better motivate its goals and objectives.

We removed that from conclusion and added that at the end of the Introduction where we explained the objective of this work.

15. The authors spend most of the paper describing all the components of the different models and spend little time focussing on the significance of the results. The authors may wish to only describe the main components of the models and shift non-essential information to a supplementary material file.

A) We removed all subsections of the section “Overcome overfitting” and tried to make this part shorter. We also moved some parts from Results to “Overcome overfitting” section. We moved some explanations of methods that are used to the Appendix. There are now 14 lines in “Overcome overfitting” section in the text.”

We moved subsections “Activation function”, “Learning rate”, “Padding”, “Activation functions in CN layers”, “Dense layer”, “Network optimization”, and “Update expression” to the Appendix.

We added Table 3 in the text.

Table 1. Common values and selected values for different parameters of the models.

Parameter	Common values	Selected value
Mini-batch size	16, 32, 64	16
Number of convolution filters	32, 64, 128	128
Filter size	3, 5, 7	(5,1) and (5,3)
Number of LSTM units	32, 64, 128	128
Number of dense layer units	16, 32, 128, 256	32
Momentum in SGD	0.5, 0.99, 0.9	0.9

B) In terms of significance of the results, we added a section to the Discussion "Model transferability":

"The transferability of a model between river basins is highly desirable but has not yet been achieved because most river ice-jam models are site specific (Mahabir et al., 2007). The developed models in this study can be used to predict future ice jams some days before the event not only for Quebec but also for eastern parts of Ontario and western New Brunswick. For other locations, the developed models can be transferred via retraining and a small amount of fine-tuning using labeled instances, rather than building from scratch. It is because the logic in the model may be transferable to another site with small modifications. To transfer a model from one river basin to another, historic records of ice jams and equivalent hydro-meteorological variables (e.g., precipitation, temperature, and snow depth) as inputs to the model must be available at each site."

16. Why did the authors not compare these much more complicated models (with, as noted in 3.3.2, 100's of thousands of parameters!) to a simple 3-layered multilayer perceptron model, or a decision tree model (e.g., random forests or eXtreme Gradient Boosting), or a support vector machine (which is ideal for small datasets)? Based on the type of dataset (small and structured) it seems the previously mentioned methods might be more appropriate, may result in better performance with much less complexity (e.g., fewer parameters and hyper-parameters), and have lower training times (e.g., likely minutes rather than hours). Without lack of a meaningful benchmark it is difficult to justify the use of these very complicated models (CNN, LSTM, CN-LSTM) that took nearly the whole paper to describe.

We developed machine learning models and compared their performance with developed deep learning models in the text. The validation and test F1 scores of Machine learning models are presented in Tables 10 and 12, showing that deep learning performed much better than machine learning in ice-jam prediction.

Table 10. F1 scores of validation for SVM, DT, and KNN and MLP models with 100 random train-validation splits.

Models	F1 score		
	mean	max	min
SVM	0.76	0.82	0.69
DT	0.74	0.80	0.67
KNN	0.75	0.84	0.68

MLP	0.75	0.83	0.68

Table 12. Test F1 scores for SVM, DT, and KNN and MLP models.

Models	F1 score
SVM	0.75
DT	0.71
KNN	0.78
MLP	0.70

17. There is little emphasis placed on exploring why certain models performed better than others and how this relates to the physical system under study. It's great if a new model provides high accuracy for modelling a physical system but understanding why the model may work better than others is also important to explore.

A) We added that: "There are some machine learning methods available for TSC such as K nearest neighbour and support vector machine. However, the focus of this research is on the deep learning models that have greatly impacted sequence classification problems and they can also be used for multivariate TSC with good performance. Deep learning methods are able to consider two-dimensionality in multivariate time-series and their deeper architecture could further improve the classification especially for complex problems, which is why their results are more accurate and robust than other methods (Wu et al., 2018a, April). However, they are more time consuming and difficult to interpret."

And also later we added in the text that: "Deep learning models perform well for the problems with complex-nonlinear dependencies, time dependencies, and multivariate inputs."

B) A) In terms of relation to the physical system, we added a section "Discussion on the interpretability of deep learning models" and later explained that we will cover this issue in our future work:

"Even though the developed deep learning models performed pretty well in predicting ice jams in Quebec, the interpretability of the results with respect to the physical processes of the ice jam is still essential. It is because although deep learning models have achieved superior performance in various tasks, these really complicated models with a large number of parameters might exhibit unexpected behaviours (Samek et al., 2017 & Zhang et al., 2021). This is because the real-world environment is still much more complex. Furthermore, the models may learn some spurious correlations in the data and make correct predictions with the 'wrong' reason (Samek and Müller, 2019). Hence, interpretability is especially important in some real-world applications like flood and ice-jam predictions where an error may cause catastrophic results. Also, interpretability can be used to extract novel domain knowledge and hidden laws of nature in the research fields with limited domain knowledge (Alipanahi et al., 2015) like ice-jam prediction.

However, the nested non-linear structure and the "black box" nature of deep neural networks make interpretability of their underlying mechanisms and their decisions a significant challenge (Montavon et al., 2018, Zhang et al., 2021 and Wojtas and Chen, 2020). That is why, interpretability of deep neural

networks still remains a young and emerging field of research. Nevertheless, there are various methods available to facilitate understanding of decisions made by a deep learning model such as feature importance ranking, sensitivity analysis, layer-wise relevance propagation, and the global surrogate model. However, the interpretability of developed deep learning models for ice-jam prediction is beyond the scope of this study and it will be investigated in our future works."

TECHNICAL CORRECTIONS

• L12 (and elsewhere in the text, e.g., L13): remove 'the' before 'ice-jam'. Done

• L19: 'validation and generalization **sets**'? Why not use test set instead of generalization set, as it is more common in the ML community? Done

• L35: remove 'to' before 'jam'. Done

• L49: 'carefully' instead of 'wisely'. Done

• L50: include 'classifier' after 'kNN'. Done

• I will mostly stop providing editorial remarks at this point…the paper should be carefully edited.

• L57: '…that use multiple layers where nonlinear transformation is used to extract…' Done

• L219: I suppose the brackets around the scaled variable should be black instead of red. Done

• L266: 'covariate **shift**'? "In deep learning, the distribution of the input of each layer will be changed by updates to all the preceding layers (i.e., internal covariate shift)."

However, I moved this part to Appendix.

• L310: is 'drown' the right word to use here? Changed to "gained"

• L336: 'over-training'. Done

REFERENCES

Alipanahi, B., Delong, A., Weirauch, M. T., & Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nature biotechnology, 33(8), 831-838.

Althoff, D., Rodrigues, L.N., Bazame, H.C., 2021. Uncertainty quantification for hydrological models based on neural networks: the dropout ensemble. Stoch. Environ. Res. Risk Assess. 35, 1051–1067. https://doi.org/10.1007/s00477-021-01980-8

Apaydin, H., Feizi, H., Sattari, M.T., Colak, M.S., Shamshirband, S., Chau, K.-W., 2020. Comparative Analysis of Recurrent Neural Network Architectures for Reservoir Inflow Forecasting. Water 12. https://doi.org/10.3390/w12051500

Barzegar, R., Aalami, M.T., Adamowski, J., 2020. Short-term water quality variable prediction using a hybrid CNN--LSTM deep learning model. Stoch. Environ. Res. Risk Assess. 1–19.

Barzegar, R., Taghi Aalami, M., Adamowski, J., 2021. Coupling a Hybrid CNN-LSTM Deep Learning Model with a Boundary Corrected Maximal Overlap Discrete Wavelet Transform for Multiscale Lake Water Level Forecasting. J. Hydrol. 126196. https://doi.org/https://doi.org/10.1016/j.jhydrol.2021.126196

Bergstra, J., & Bengio, Y. (2012). Random search for hyper-parameter optimization. Journal of machine learning research, 13(2).

Fawaz, H. I., Forestier, G., Weber, J., Idoumghar, L., & Muller, P. A. (2019, July). Deep neural network ensembles for time series classification. In 2019 International Joint Conference on Neural Networks (IJCNN) (pp. 1-6). IEEE

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., Herrnegger, M., 2018. 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

Mahabir, C., Hicks, F. E., & Fayek, A. R. (2007). Transferability of a neuro-fuzzy river ice jam flood forecasting model. Cold Regions Science and Technology, 48(3), 188-201.

Montavon, G., Samek, W., & Müller, K. R. (2018). Methods for interpreting and understanding deep neural networks. Digital Signal Processing, 73, 1-15.

Samek, W., & Müller, K. R. (2019). Towards explainable artificial intelligence. In Explainable AI: interpreting, explaining and visualizing deep learning (pp. 5-22). Springer, Cham.

Samek, W., Wiegand, T., & Müller, K. R. (2017). Explainable artificial intelligence: Understanding, visualizing and interpreting deep learning models. arXiv preprint arXiv:1708.08296.

Wojtas, M., & Chen, K. (2020). Feature importance ranking for deep learning. arXiv preprint arXiv:2010.08973.

Wu, J., Yao, L., & Liu, B. (2018a, April). An overview on feature-based classification algorithms for multivariate time series. In 2018 IEEE 3rd International Conference on Cloud Computing and Big Data Analysis (ICCCBDA) (pp. 32-38). IEEE.

Wunsch, A., Liesch, T., Broda, S., 2020. Groundwater Level Forecasting with Artificial Neural Networks: A Comparison of LSTM, CNN and NARX. Hydrol. Earth Syst. Sci. Discuss. 1–23. https://doi.org/10.5194/hess-2020-552

Zhang, D., Lin, J., Peng, Q., Wang, D., Yang, T., Sorooshian, S., Liu, X., Zhuang, J., 2018. Modeling and simulating of reservoir operation using the artificial neural network, support vector regression, deep learning algorithm. J. Hydrol. 565, 720–736. https://doi.org/10.1016/j.jhydrol.2018.08.050

Zhang, Y., Tiňo, P., Leonardis, A., & Tang, K. (2021). A survey on neural network interpretability. IEEE Transactions on Emerging Topics in Computational Intelligence.