Response to reviewer 2: Thomas Lees

Link to review

Comments/Text of reviewer posted in **black**; our answers are posted in **blue**.

1. OVERVIEW

This paper seeks to answer the question: "Can a single LSTM model be used to produce accurate and consistent discharge simulations at daily timescales and sub-daily timescales?". The major finding was that yes, you can use a single LSTM to produce daily and hourly predictions. Furthermore, compared with more traditional hydrological models, the MTS-LSTM shows a much smaller performance deterioration when comparing daily simulations (better) to hourly simulations (only slightly worse).

The novel contributions of this paper are threefold:

1. The development of a new multi-timescale LSTM (MTS-LSTM) that produces discharge simulations at both daily and sub-daily timescales (including the flexibility to include arbitrary timescales).

2. The manipulation of the loss function to explicitly account for prior knowledge about the translation between daily and sub-daily timescales. Related to the “hierarchical” nature of these timescales.

3. The benchmarking of a suite of LSTM-based models against the operationally used NOAA National Water Model (NWM).

Research into LSTM based rainfall-runoff modelling has, thus far, mainly focused on simulations at daily timescales. This paper provides a welcome addition to the literature, since sub-daily trends can be important for flood impacts and for water resource managers. The authors focus on producing discharge simulations at a daily timescale and an hourly timescale, although they also show results for 3-hourly and 6-hourly timescales (see Table 2 p9, Table 7 p16).

In order to explore the LSTM architectures that can produce discharge simulations at multiple timescales, the authors suggest three possible avenues (more are included in Appendix B):

- Multiple LSTMs with different timescales, an hourly LSTM and a daily LSTM (naive).
- A “shared” multiple-timescale LSTM (sMTS-LSTM), which overcomes the problems of overly long input sequences, causing long training and inference times for the naive model.
- The MTS-LSTM, which overcomes the problems of the sMTS-LSTM being unable to include different input data for the different timescales.

Both the sMTS-LSTM and the MTS-LSTM are novel contributions to both hydrological modelling, and as far as I am aware, machine learning more generally. My main comment about the paper is that the difference between the sMTS-LSTM and the MTS LSTM could be made clearer.

The authors describe four experiments to demonstrate the usefulness of their newly developed models:

1. Benchmark the MTS models (sMTS-LSTM & MTS-LSTM) against traditional hydrological models (NOAA NWM) and the naive LSTM (which although “naive” is still the most difficult benchmark to compete with). This comparison is thorough and explores accuracy across the hydrograph (see Table 4, Figure 3, Figure 4).

2. Explore the consistency of the MTS models hourly discharge predictions when aggregated to the models daily discharge predictions. The regularisation of the loss function improved the consistency of the sMTS-LSTM.

3. Compare the computational efficiency of the 3 LSTM-based models. The MTS LSTM was the most computationally efficient.

4. Test whether including the same information from different timescales improves model accuracy. The extra information improved forecast accuracy over a range of performance metrics (Table 6).

Overall, these experiments are well thought through and they meet the aims of HESS. The research advances hydrological modelling by:

- benchmarking data-driven models (LSTMs) on an hourly timescale
- developing novel model architectures that show state-of-the-art performance
- demonstrate a next step for LSTM-based models to be used in operational forecasting settings
- demonstrate the flexibility of manipulating the loss function in data-driven models to meet different requirements (e.g. timescale consistency).

Furthermore, the availability of the code via the neuralhydrology repository, with an accompanying notebook makes it possible to view the author's assumptions and reproduce the figures in the paper.

2. SPECIFIC COMMENTS

I was grateful for the following:

- Figure 2 (P7) is extremely helpful and very professionally made. This is extremely helpful when trying to parse the novel model architecture (MTS-LSTM) proposed by the authors.
- The overt structure outlined on P3 L64-78 is a very helpful signpost to the reader.

- The regularization used to ensure timescale consistency (Sect 2.3.2) is novel and interesting for the target audience of HESS, hydrologists and earth scientists. It confirms the view that the loss function offers huge flexibility to modellers to improve their models for specific use-cases.
- Equation 1 (P8 L180), the annotations to this equation are extremely helpful.
- Table 3 and Table 4 demonstrate an extremely thorough comparison of the models for various metrics and hydrological signatures. This could be used as an example for future benchmarking experiments as an extremely thorough inter comparison, exploring the various facets of the hydrograph.
- Appendix B is a very worthwhile addition, since these negative results can help the field from repeating these results, especially because they turned out to work less well than the model architectures included in the main text. It also outlines the thoroughness of the authors experiments.
- The inclusion of the data and a Jupyter Notebook for readers to reproduce the results is to be applauded. The notebook is well written and the community will be grateful for the time and effort that the authors have put into making their code available and their experiments reproducible. Thank you.

We would like to thank Thomas Lees for the detailed and thoughtful review. Based on the comments, we have updated our manuscript in several places, most notably to include a more detailed description of the different multi-timescale LSTM variants. We will address each comment individually below (our responses are colored in blue).

2.1. COMMENTS

2.1.1. P3 L80-87 Are you still using the CAMELS observed discharge or do you now exclusively use the USGS Water Information System REST API values for both hourly and daily evaluation?

We only use discharge from the REST API in this study. We will add a sentence that clarifies this in the revised manuscript.

2.1.2. P3 L81-82 Just to confirm, this is still a "predict timestep including all input data up to time t" rather than a forecast. This is confirmed on P17 L306 but might be worth also including that information here.

Yes, that is correct: we have a setup for a simulation model. We predict the average daily/hourly discharge at timestep t, using inputs that include the meteorological information of the same timestep t. Since L81-82 are part of the Data section, however, we argue against mentioning this there.

2.1.3. P4 L101-104 In Section 2.2.1 you describe that you use the NWM v2 Reanalysis product. You describe that this is an hourly product. Do you therefore calculate a daily average of these results to compare against the daily simulations?

Correct. We'll add a sentence to clarify that the lower-resolution predictions are averaged hourly predictions.

2.1.4. P6 L131-155 I am still not fully clear on the difference between the sMTS-LSTM and the MTS-LSTM. Can we work to make this slightly clearer in Section 2.3.

This relates to Jens Kiesel's comments (questions 2.3.5, 2.3.6 and 2.3.7). We'll briefly answer the specific questions here; for the discussion on a clarified description of sMTS-LSTM we refer to our answer there.

- Do the sMTS-LSTM and the MTS-LSTM receive the same input data?
  Yes, with the exception that the input of sMTS-LSTM additionally contains a flag that identifies the timescale (one-hot encoding).
- Do both the sMTS-LSTM and MTS-LSTM require two forward passes (L140)?
  Not quite (though this may to some degree be a matter of interpretation). The sMTS-LSTM does a second forward pass for the timesteps where daily and hourly values overlap. In the MTS-LSTM, the overlapping time steps are handled by different LSTMs, so there is only one forward pass which involves "splitting" the processing across two LSTMs.
- It seems that the MTS-LSTM "splits the LSTM into two branches" (L148), which is described as unique to the MTS-LSTM, but then Figure 2 suggests that the sMTS-LSTM also does this splitting but the fully connected layers (FC_c, FC_h) are simply identity functions.
  It is a question of interpretation whether an sMTS-LSTM consists of multiple identical branches or of a single LSTM that is used for multiple input resolutions. We think that the former interpretation nicely highlights the similarity to MTS-LSTM, while the second one might better highlight the differences.
- Does the one hot encoding (L141) mean that the LSTM weights are copied in both branches but then zeroed if we are looking at either the hourly or the daily data?
  If so then why can we not use different input datasets in the sMTS-LSTM as we can in the MTS-LSTM?

  The one-hot encoding is related to the inputs (they are essentially additional inputs that are created to differentiate between hourly (e.g., input value 0) or daily (e.g., input value 1). We do not touch the weights at all. We think this also explains the second question. Since the input dimensions do not change (they are the number of meteorological and static inputs + the number of timescales embedded into one-hot encoding), we cannot use different forcings in the sMTS-LSTM model. See also our answer to Jens Kiesel's questions 2.3.6 and 2.3.7..

There are various solutions. One could: include a table explaining the differences explicitly; include the sMTS-LSTM as its own diagram in Figure 2; or spend more time in Section 2.3 clearly outlining the differences between the two architectures.

As said in the answer to Jens Kiesel's questions 2.3.5/2.3.6, we will add a more detailed explanation of the differences between the two variants in the revised manuscript.

2.1.5. P6 L154-156 "This architecture makes it clear why we call the other variant "shared" MTS-LSTM: Effectively, the sMTS-LSTM is an ablation of the MTS LSTM. Both variants have the same architecture, but the weights of the sMTS LSTM are shared across all per-timescale branches and its state transfer layers are identity operations." I am not what this sentence means. I think it could potentially be clearer for a hydrological audience. My understanding is that an "ablation" means that the sMTS-LSTM is missing something that the MTS-LSTM has, but if they have the same architecture then I am not certain what is missing? From reading and re-reading the difference is something to do with the fully connected layer but I am just a little bit confused about the difference between these two models.

The thing that the MTS-LSTM has and the sMTS-LSTM does not is the flexibility to use a different LSTM in the hourly vs. the daily branch. The fully-connected layer is not that important in this context---it is only necessary to make the dimensions of the daily and the hourly states match.

Looking at it conversely, it is maybe clearer that MTS-LSTM is a generalization of sMTS-LSTM: Consider an MTS-LSTM that uses the same hidden size in all branches. This model could learn to use identity matrices as fully-connected layers and equal weights for all LSTM branches, which would make it an sMTS-LSTM (save for the one-hot encoding).

We will rephrase our explanations in the manuscript, together with the improved explanation of MTS-LSTM vs. sMTS-LSTM, to make this more clear.

2.1.6. P7 L158-169 Related to the misunderstanding of the difference between the sMTS-LSTM and the MTS-LSTM, I am not certain what it means to include multiple datasets and why this could not be done for the sMTS-LTSM. I know that in the paper: "A note on leveraging synergy in multiple meteorological datasets with deep learning for rainfall-runoff modeling", some of the authors have shown that the LSTM produces more accurate discharge simulations with multiple sources of rainfall information. Is that what is being done in this experiment?

Partly, yes. There are two possibilities:
(1) using multiple data products that all have the same temporal resolution (this is what's being done in the paper you refer to),
(2) using multiple data products with different temporal resolutions.
Both options can be used with MTS-LSTM. With sMTS-LSTM, the data used at the different time scales must have the same dimensionality (because they're processed by the same LSTM, see question 2.1.4), so option (2) (per-timescale products that may have different amounts of variables) does not work. Option (1) is possible with both architectures.

We will clarify this in the revised manuscript.

Furthermore, if the sMTS-LSTM has the same architecture as the MTS-LSTM (as outlined in the caption to Figure 2), then why can't the sMTS also include new information to the hourly branch? (I am assuming here that the only difference is that the FC_h and FC_c are identity functions rather than linear functions as in the MTS).

The brief explanation is that the input dimensionality would in general not match (see also answer above). Since the weights of the LSTM across all timescales are the same, the number of inputs has to be the same as well (and actually not only the number of inputs, but it should be the *same* inputs). We will rephrase our explanation in the revised manuscript to clarify this.

2.1.7. P7 Related to the comment above. "In the other, we additionally ingested the corresponding day's Daymet and Maurer forcings at each hour." Is this data at a daily resolution?

Yes. We will add a brief explanation that clarifies this. To summarize, for every hour of a particular day, we concatenate the daily forcings of the same day as additional inputs.

If so, does this mean that you are copying the daily inputs 24 times as input for each hour? So if we have hourly NLDAS, You are including Daymet for Day 1 24 times? NLDAS1 + Daymet 1, . . ., NLDAS24 + Daymet1. Apologies if I have misunderstood.

Correct (see also the answer to the previous part of this question). We will add a brief explanation that clarifies this.

2.1.8. P7 L167 "... In the other, we additionally ..." I think it would make sense to explicitly write that you are using the NLDAS forcings AND the Daymet/Maurer forcings. Perhaps something like: "In the other, we ingest the NLDAS forcings as well as the corresponding day's Daymet ..."

We will rephrase the sentence to be more explicit.

2.1.9. P13 Table 4: You write in the Table caption "Bold values highlight results that are not significantly different from the best model in the respective metric or signature (α = 0.001)". I am sure I have misunderstood, but when I look at the Hydrologic Signatures, for example Daily, Q mean. Both the Naive (0.986) and NWM (0.972) results are highlighted. However, Both the sMTS-LSTM (0.985) and the MTS LSTM (0.984) have values closer to the best model. Is this an artefact of the aggregation? Where the mean is hiding the distribution of Pearson Correlation scores across multiple basins? If so that is fine I just wanted to ensure that this was not a mistake.

The highlighting is correct, and it is indeed an artefact of the aggregation. For signatures (not metrics), the table shows the Pearson correlation with observed values. The bold font highlights models for which the results were not significantly different to those of the model with the highest Pearson correlation. The significance test (Wilcoxon) has the null hypothesis that the differences between two populations are symmetric around zero. Since

Pearson correlation doesn’t operate on these differences, there exist cases where the test is significant but the Pearson correlation is close to that of the reference group. Had we chosen the Spearman correlation (which is based on ranks), sMTS-LSTM would have had a slightly higher correlation coefficient than Naive.

2.1.10. P14 Table 5 Why do we only see results for the sMTS-LSTM. I believe you have written that it is the “best benchmark model”, but is there any other reason to include/exclude the MTS-LSTM? If the experiment was already run it might be an idea to include it, but it is not necessary.

We only reported results for sMTS-LSTM since it was the best model in the above benchmarking. While we have not done it so far, the experiment would certainly be possible for MTS-LSTM, too.

3. FORMATTING

I am not certain of the procedure here but I am drawing to your attention in case it is useful.

3.1. P2 L33: "... (e.g., Schmidhuber (1991), Mozer (1991))." to "... (e.g., Schmidhuber 1991, Mozer 1991)"

3.2. P8 L173-174: "... (e.g., computer vision, Zamir et al. (2020))" to "... (e.g., computer vision, Zamir et al. 2020)"

Thank you for pointing this out. We will change the references to the correct format.

4. SUGGESTIONS

I believe that you are using the terms "look-back window" and "input sequence" interchangeably. Is it perhaps worth using one term consistently through the paper?

- P5 L115 "... look-back windows of 365 days ..."
- P6 L128 "... input sequence of 4320 hours (180 days) ..."
- P6 L137 "... input sequence of T_D time-steps ..."
- P6 L143 "... has access to a large look-back window ..."
- P8 L190 "... achieve a sufficiently long look-back window ..."

We agree that it makes sense to work on a more consistent use of the two terms. The terms are almost identical, but there are slight differences: A long input sequence does not have to mean that the model looks far into the past (if the resolution is high). Also, look-back seems like a nice way to refer to input sequences regardless of their timescale, whereas we usually associate input sequences with a fixed timescale (e.g., hourly). We will try to follow this distinction in the revised manuscript.