General note: I was asked by the editor to review this manuscript, although groundwater hydrology is not my area of expertise. However, machine learning is and therefore most of my review will be around the methods and experimental setting used in this manuscript.

This manuscript presents an approach for filling gaps in time series of ground water well measurements. Specifically, the authors compare two different methods (LSTM-based and ARIMA) for different gap lengths for six different wells.
Although I generally welcome publications that try to make use of deep learning based methods for various applications in earth science, I see various major concerns with the manuscript at hand. Overall, it seems like the authors are not too familiar with the methods they apply (especially the LSTM-based model) and many decisions made seem questionable and lack any justification or explanation. Because of these concerns, I'm not sure if I can recommend this manuscript for publication. If it should be published at all, major revisions are required.

Major Concerns:

1. **Model architecture**: Coming from the field of machine learning, I was surprised by the creativity of the authors in finding their model architecture. To be honest, I have never seen such a combination of LSTM layers, dense layers and convolutional layers for a time series task and I wonder if the authors know what they are doing. Here is a list of sub points to this major comment:
    a. First: Did you perform any hyperparameter search at all to find this architecture? If yes, please give details on the model configurations (in terms of layers) you tried, if not, why not? To propose such an excotic architecture, it is required to see quantitative evidence that this is required and not a much simple LSTM-based model would be better (e.g. single LSTM layer with single dense + dropout layer)
    b. Why do you stack 3 LSTM layers? In theory, a single LSTM layer is turing-complete. Besides probably natural language processing, where the training data consists of million/billion of samples, there is almost always no need to use more than a single LSTM layer. Additionally, since you have very limited training data (2 years of hourly data are just 17520 data points), the size of your LSTMs seem to be exorbitantly large. Especially with 3 LSTM layers.
    c. Why the combination of convolutional layers and dense layers after the LSTM? Probably the standard is to have a single dense layer that uses the hidden output of the LSTM to map to your desired target shape. Why do you think so much complexity is needed after the LSTM, since the LSTM should capture the complex temporal dependencies already?
    d. Why do you have the convolutional layer at all? If I understand your setting correctly, the convolutional layer can again look at the entire sequence (M x 64, with M the input sequence length). Why is this necessary? The task of the LSTM is to summarize the input sequence and store all the information necessary for

predicting the M+1 time step (first step of your N time step long gap) in it’s cell state.

   e. Another point related to the convolutional layer. I see that the filter size was solely chosen to be able to map from a sequence length of M to an output of N (filter size M-N+1). However, are the authors aware of what that means? For example, for predicting the first of the N time steps, the convolutional filter will only look at the first M-N+1 input sequence elements, effectively ignoring what has happened at the time steps preceding the current time step. Why do you want this? It makes absolutely no sense to not include the most informative information (the previous time steps) necessary to predict the next time step.

2. **Related work:** Since (correct me if I’m wrong) this is not a forecast task, but just filling gaps in historic data records, I wonder if the authors have done some research, which approaches are currently used in the field of deep learning, before proposing their own method. E.g. for gap filling in historic time series, Bi-directional LSTMs are commonly used over normal LSTMs, since they do two sided gap filling (closer to interpolation), compared to the standard LSTM, which basically extrapolates into the future. I would also advise to add some related work section of LSTM-based gap filling into the introduction.

3. **Training setup:** There are various points around the model training setup that I see problematic. Some of them might overlap to other points mentioned above or below.
   a. Input features for any neural network should be normalized to zero mean, unit variance and not to the range of 0 to 1. This will basically bias your network during the start of the training in a wrong way. Maybe as some intuition: Most (all?) activation functions are centered around zero, e.g. the sigmoid function in all gates of the LSTM. With randomly initialized weights (which are normally initialized around 0), using your normalization would bias the entire network to always have pre-activations of larger than zero, and thus sigmoid values close to one. However, what you want is in expectancy to be undecided in the beginning (pre-activation of 0, equals to sigmoid of 0.5). Long story short, you should re-run all experiments with different normalizations, at least for the LSTM.
   b. Results of neural networks are generally affected by some stochasticity, because of the random weight initialization and the randomness of stochastic gradient descent. This requires almost always to train multiple models for the exact same setting with different random initialization (seeds) and to report the average model performance and variations across those repetitions. Otherwise, results might not be reproducible, since you might only be lucky (or unlucky) with your single initialization.
   c. In general, you have very few data points for such a large deep learning model, as already stated above. You could either think of ways, how to combine the data of all wells in a single model, or reduce your model size drastically, which is what I would propose here.

d. I found it very hard to follow your training and testing setup, until late in the paper. E.g. around the number of possible model configurations, and total train-test combinations. I would advise to a sentence at the very beginning of the methods like “We train one model for a single well and evaluate this model on the same well and all other wells.”

e. Furthermore, why are models tested out-of-sample, meaning being trained on different wells than evaluated? Is there any idea behind it? Is the idea to learn a model that should be able to fill gaps in time series of any well at any location? If yes, you should probably re-think your entire training setup. If not, I don’t see the need for this evaluation, since this is also not done for the ARIMA model.

4. **LSTM vs ARIMA comparison:**
    a. Why did you perform Hyperparameter search for the ARIMA method and not for the LSTM-based model?
    b. Why is ARIMA not tested on wells that are not the training well, while the LSTM is?
    c. P12 L6f: How was the best model decided? On training or test period? As of P13 Line 2f it seems like you picked the best model based on the test period results. If this is true, your results are biased and do not represent the true expected results of your methods. You either chose the best model by the training period, or better, have a third independent period (called validation split in machine learning) and pick your model based on the performance in this third data split, which is neither used for training nor for the final model evaluation.
5. **SpC:** Later in the results section, you state that only SpC is of interest and no results for any of the other two variables are presented in this manuscript. This is totally okay, but my question is, why then do you model all three variables? Why not train the model using three inputs (temp, level and SpC) and predict only SpC?
6. **P 11 L 20:** “We also observe that models with a daily 24-hour input window outperform other models with longer input windows as shown in Figure 6 (c).” This statement, figure 6(c) and thus your conclusion in the following sentences and the rest of the paper are misleading. It is completely logical, that the averaged MAPE over all settings for the input sequence length of 24h is the lowest, since this only includes models, where you predicted N=1h, 6h, 12h or 24h (as of table 1: N <= M). And as you have seen from all other experiments, filling only small gaps is easier for any model than filling large gaps. So the fact that the 24h input sequence has the smallest error is not due to the 24h input sequence, but due to the short output sequence for M=24h inputs. I would bet that if you train a model with input length 168h and only evaluate for 1h, 6h, 12h and 24h performance should be similar/better than for a 24h input window. It is probably better to remove figure 6(c) or rethink how you can fairly compare the average results over different input sequence length, since the different input sequence length also mean you evaluate them for different gap filling length.

Minor Comments:

- Title: At no point of this manuscript I see the term “spatio-temporal” justified. You are only filling temporal gaps in time gaps of a single well, without any spatial input information (e.g. the input features of the neighboring wells). So I would strongly advise to change all occurrences of the spatio-temporal framing to temporal only or clearly justify what in your work is the spatial component.
- P3 L4: Connor et al. (1994) is not the citation you should cite here for the RNN. Jordan (1986) would be more appropriate. Also the blog post from Olah (2015) is probably misleading here.
- P3 L11 Ma et al (2015) is definitely not the correct reference here and you should cite the original LSTM paper by Hochreiter & Schmidhuber (1997).
- P3 L11f. Beside text prediction, text translation, speech recognition and image captioning, LSTMs have also already been applied to earth science and even in hydrology, which might be also/more relevant to mention here.
- P 4 L 2 “select” -> “selected”
- P5 L15ff: In this entire discussion you mention “highly correlated” (L19), “lower correlations” (L20), “correlates well” (L20) and many more of these statements. Such statements usually required some quantitative measures (e.g. correlation coefficient). Otherwise, what is a high correlation and what low?
- P5 L27 here you state you only investigate 24-, 48-, 72-h gaps. In table 1 you have much longer periods listed as well as in figure 6, while then in figure 7 again only 24, 48, 72. This is a bit inconsistent.
- P5 L23 delete “clearly”
- P6 L3 What you mean is not a dropout layer, but the combination of a dense layer with additional dropout. Two consecutive dropout layer would mean simply applying dropout again to the result of your previous dropout output. Correctly it would state “followed by dense layer with dropout”.
- P6 L3f: “This model architecture is generally described as a stacked LSTM model, given that the LSTM layers are "stacked" on top of each other.” This is a tautology. Maybe simply remove this sentence or rephrase it.
- P7 L7 “select” -> “selected”
- P7 L17 This is not called a “sigmoid neural net layer”. You could say “A linear layer with sigmoid activation function”. At least call it “neural network” not “neural net”.
- P7 L17: The pointwise multiplication is not part of the gate it-self, but how the gate is combined with the cell state.
- P7 L18ff and Fig5: all gates (f,i,o) and the cell and hidden state are vectors and should be written in lower, bold, italics letter and not capital letters.
- P7 L 23ff: “Finally, an output gate (O t) decides what to output based on the input and previous memory state. The sigmoid layer of the output gate decides what parts of the memory state will be output...” The second sentence is basically a repetition of the first. Consider rephrasing.

- Table 1: Any particular reason, why you excluded 96h from the list of possible output window length, since otherwise possible input and output window length seems to be equal?
- P10 L 22 How are the terms (P, D, Q)m combined into equation 2. This needs more explanation.
- P11 L 19: In your setting, you always extrapolate. So this statement is not correct.
- P11 L 32: delete “very”
- LSTM results in general: It would be good to see only insample results at some point. How good does the LSTM perform for the same well it was trained for (as average over the 6 wells or for each well independently).
- Figure 7: Missing the information that results are only for SpC.
- The point above applies to the entire section here.
- P12 L15f: “It is noted that the optimal…” I would be cautious with such statements, unless you perform similar hyperparameter search for LSTMs as you did for ARIMA.
- P13 L 8f I do not see this in Figure 8. For me, there is no visible difference (or very hard to detect) in the Arima and LSTM error at any special frequencies. Maybe a better visualization or some quantitative measures would help.
- Figure 8. Why are the results now with the ARIMA model and 72 hour inputs and not 168 as in Figure 7?
- P14 L 1 Again, I don’t see the LSTM outperforming ARIMA from Figure 8 column 3. Not sure how these (also column 4) help here. Maybe it is due to my lack of understanding of the data itself, but I think some quantitative measures are better than these figures. (e.g. a table with some metrics)
- “In general, both LSTM and ARIMA are effective at capturing longer term variability, but LSTM is more effective at capturing high-frequency fluctuations and nonlinearities in the dataset.” I don’t see any (quantitative) evidence for such a statement.

- Conclusion: As of everything written above, I think the conclusions need to be entirely rewritten, including possible new results of different model configurations etc. I will not go into more detail here, since I raised many concerns above, that apply similarly to the same statements in the conclusion (e.g. LSTM and ARIMA comparisons etc). Furthermore, you miss to say for which variable you are doing gap filling (SpC only)

References:

Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, *9*(8), 1735-1780.

Jordan, M. I. (1986). Attractor dynamics and parallelism in a connectionist sequential machine. In Proceedings of Ninth Annual Conference of the Cognitive Science Society, Amherst, pages 531–546