Response to referee comments

Reviewer 3

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.

Response: Thank you for reviewing and providing the summary. We agree with the reviewer that it is important to demonstrate that we know what we are doing, and we appreciate the reviewer's careful attention to make sure we did our due diligence. The comments are addressed point-by-point.

Reviewer Comment 3.1 — 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 exotic 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)

  Response: We performed hyperparameter searches on: Number of LSTM layers, number of units per LSTM layer, number (and size of) dense layers, activation functions. This was performed for data on one well (399-1-1) with a smaller subset of input and output prediction windows, experimenting with different architecture configurations.

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.

  Response: In response to using multiple LSTM layers, there has been research looking at the benefits of using multiple RNNs/LSTMs in a model in comparison to a single RNN/LSTM [Graves et al., 2013, Pascanu et al., 2013]. Likewise, there has been work in using multiple LSTMs for action recognition [Zhu et al., 2016], traffic prediction [Du et al., 2017], and vulnerable road users location predictions [Saleh et al., 2017]. As such, we wanted to investigate the potential benefits of using multiple LSTM layers in our problem domain. We will add a paragraph to our manuscript discussing previous uses of stacked LSTMs and some comparisons of single versus multiple in different domains to give context on why we are interested in this model architecture and update our references with the cited articles.

Old paragraph: We have tested the effects of training data on model performance using 2,4 and 6 years data, and found that 4 years of training data led to similar performance to 6 years of training data. Therefore, we are confident that we have enough data to support the selected architecture. We will provide results from a single LSTM layer in supplemental material for comparison.

New paragraph: There has been research looking at the benefits of using multiple RNNs/LSTMs in a model in comparison to a single RNN/LSTM [Graves et al., 2013, Pascanu et al., 2013]. Likewise, there has been work in using multiple LSTMs for action recognition [Zhu et al., 2016], traffic prediction [Du et al., 2017], and vulnerable road users location predictions [Saleh et al., 2017]. As such, we investigate the potential benefits of using multiple LSTM layers in the problem domain of hydrological networks.

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?

Response: As stated in our response in 3.1a, we performed some hyperparameter searches, experimenting with different architecture configurations which led us to use convolutional and dense layers. We acknowledge that more information on the extensive analysis and experimentation we have performed would be useful in further justifying the choice of model architecture, so we will provide those details in supplemental material.

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.

Response: Yes, the intent was to map from a sequence length of M to an output of N. The reviewer is correct that the convolutional filter does limit the model in ignoring the most recent time steps. As stated in our response to comment 3.1c, we felt our exploration of architectures, including using convolutional layers, resulted in a good architecture. Furthermore, the time steps immediately preceding the current time are not necessarily the most informative information in the presence of dynamical behavior. However, in response to the reviewers concern, we will train the models without the convolutional layer (filter size M-N+1) and compare the results against the original architecture. We will ensure all of the input sequence will be used in predicting each future time step. A proposed architecture is to first change the last LSTM layer to return the last hidden states (i.e, output size 128). This modifies the M x 128 dense layer to be a one-dimensional 128 size layer. Then, we will remove the M x 64 dense layer and the convolutional layer (now defunct) and change the final dense layer to be size N, whose output will be the N predicted SpC values. The architecture can be modified to return the predicted N values for all three measurements by using three independent dense layers of size N instead of one, which will be concatenated at the end into a N x 3 output. Below are two images of the proposed architectures, one predicting only SpC and one predicting all three measurements.

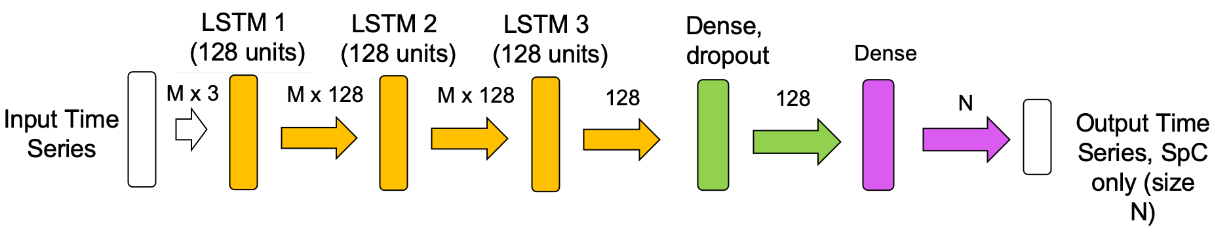

Figure 1: Modified model architecture without convolutional layer, only predicting SpC measurement

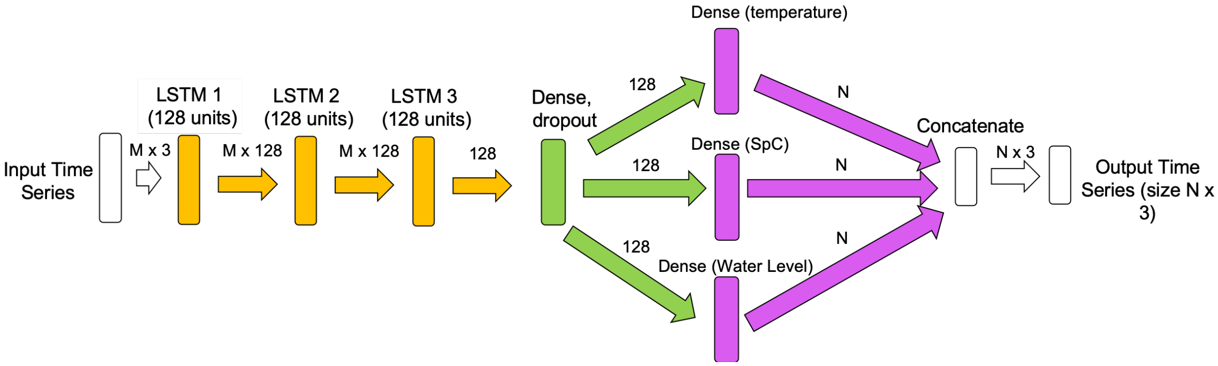

Figure 2: Modified model architecture without convolutional layer, predicting temperature, SpC, and water level measurements

Reviewer Comment 3.2 — 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.

Response: The reviewer is correct that the goal is to test gap filling in historical records. For our work, we treat the gaps as a forecasting problem which means we use the historical data as input to predict the values during gap period. Bi-directional architectures have been used for gap-filling. A bi-directional LSTM is another model type applicable to the work, but we felt that a more thorough explanation of an LSTM would be prudent to explore before jumping to that architecture. We will add a paragraph describing deep learning techniques that have been applied to gap filling, including LSTMs.

New Description: There have been several applications of deep learning techniques to fill gaps in time-series data. Ustoorikar and Deo [2008] used genetic programming to fill in gaps of ocean wave heights. Khalil et al. [2001] estimated missing values in monthly streamflow time-series data using neural networks. Berglund et al. [2015] used RNNs and bi-directional RNNs to infer missing values in high-dimensional binary time-series data.

Reviewer Comment 3.3 — 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.

Response: Thank you for the comment. We will re-run the experiments using the zero mean, unit variance normalization technique and compare those results against the original normalization.

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.

Response: Thank you for the comment. We will re-run the experiments with different initialization seeds.

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.

Response: We have the ability to combine the data of input from neighboring wells (up to five more) for the large deep learning model, which for 4 years would be approximately 210240 data points. Similar to the reviewer's comments for 3.1a, we can also perform more extensive experimentation on a smaller model (single LSTM layer with single dense + dropout layer).

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.”

Response: Thank you for the great suggestion. We will add a sentence to the beginning to further clarify our training and testing setup.

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.

Response: The intent on evaluating models on wells different from the training well was to analyze how well the model does on data from a well it has not seen. However, as noted by the reviewer, this evaluation was not done for the ARIMA model. As such, we will remove this evaluation in order to make the paper more straight forward and less confusing in its comparison of LSTMs and ARIMA. Furthermore, we will redo figure 6 without the additional analysis and remove figure 6f.

Reviewer Comment 3.4 — LSTM vs ARIMA comparison:

a Why did you perform Hyperparameter search for the ARIMA method and not for the LSTM-based model?

Response: A hyperparameter search for the ARIMA approach is performed by using the “auto.arima” function in R automatically. We also performed a hyperparameter search on the architecture of the LSTM models. This includes: the number of LSTM layers, the number of units per LSTM layer, andthe number (and size of) dense layers, and activation functions.

b Why is ARIMA not tested on wells that are not the training well, while the LSTM is?

Response: ARIMA is not tested on other wells since the ARIMA model is built dynamically based on the 168 historical hours for each well. We believe the information carried by the ARIMA model is not enough to train other well. Also according to the comment 3e, we will remove the model evaluation on testing (which includes figure 6(f) on the non-training wells to reduce the confusion.

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.

Response: The best model for each well was decided on the test period (data from 2011). We have two more years of data (2017-2018) for the wells. We will update the analysis to use data from 2011 as the validation split and the data from 2017-2018 for the final model evaluation.

Reviewer Comment 3.5 — 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?

Response: Other similar analyses for groundwater table and temperature are done but not shown here because SpC is our primary interest for this study. The reviewer is right that we don't have to predict all three variables. Although retaining all three variables in the model affords the flexibility to fill gaps in the other two variables, it would be too much for this paper to cover. Therefore, we will follow your suggestion to keep only SpC as our predicted output variable.

Reviewer Comment 3.6 — 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.

Response: Thank you for the feedback. We re-did the analysis done in figure 6c, but limiting the predict output to 1h, 6h, 12h, and 24h. As the reviewer noted, the averaged MAPE for the model with input length of 168h, 7.330167, is similar to that of the 24h input length models (6.394719). This provides a more fair analysis on the model performance based on input length, given the number of model performances averaged per input window is the same. We will redo figure 6(c) to only include models whose predicted outputs are 1h, 6h, 12h, and 24h. Additionally, we will update our analysis of figure 6(c). The updated figure also includes the removal of figure 6(f), as stated in our response to comment 3.4b. The updated figure and caption is shown below.

Old sentence: 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 likely results from an optimal number of memory units for capturing daily and subdaily memories.

New sentence: We compare model performance by input window size, but limiting to models whose predict output is 1h, 6h, 12h, or 24h. As show in Figure 6 (c), models with a daily 24-hour input window have the best performance. However, there is a large amount of overlap of the 95% confidence interval for each input window.

Minor Comments:

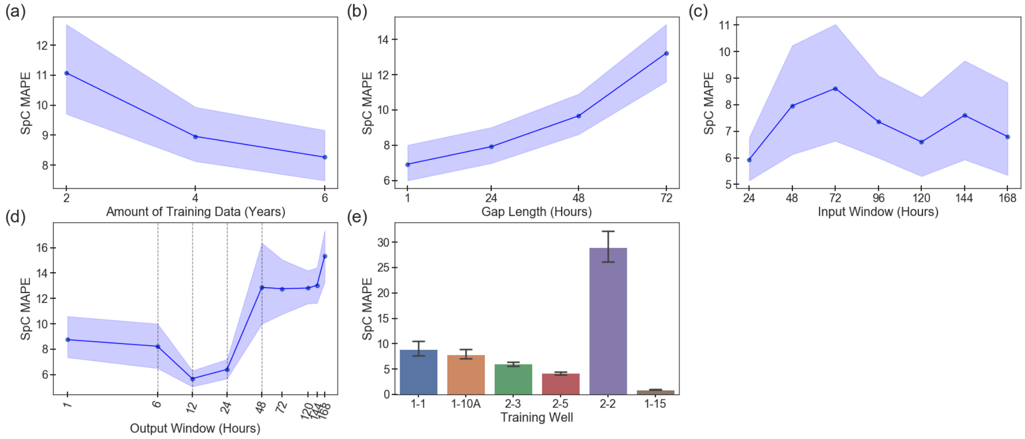

Figure 6: Gap filling performance for SpC evaluated against multiple model configuration parameters (a-d) or grouped by training wells (e). (a) average MAPE vs. number of years of training data; (b) average MAPE vs. gap lengths; (c) average MAPE vs. input window size M; (d) average MAPE vs. output window size N, but only for models whose predicted output is 1h, 6h, 12h, and 24h; (e) average MAPE aggregated by wells used to train the models. 95% confidence intervals of the averaged MAPE value are shown in shaded area in plots (a) -(d) and as the error bars in (e)

Reviewer Comment 3.7 — 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.

Response: We agree with reviewer and have changed all instances.

Reviewer Comment 3.8 — 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.

Response: We thank the reviewer for the comment. We will remove the citation of Olah (2015) and update our citation for the RNN to Jordan (1986).

Reviewer Comment 3.9 — 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).

Response: We thank the reviewer for the comment. We will update the citation and references accordingly.

Reviewer Comment 3.10 — 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.

Response: On P2 Line-35, we cite papers using DL in geophysical domain. However, as implied by the reviewer, we will add a brief description of LSTMs applied to earth science and hydrology (this is the same update in response to reviewer comment 2.10)

Old section (P3 L13): This makes LSTMs well suited for the problem at hand, particularly for data where multiple timescales of variability can affect responses [Liu et al., 2016, Song et al., 2017].

New section (P3 L13): This makes LSTMs well suited for the problem at hand, particularly for data where multiple timescales of variability can affect responses [Liu et al., 2016, Song et al., 2017]. Furthermore, LSTMs have been applied to earth science and hydrology domains. Kratzert et al. [2018] used LSTMs to predict rainfall-runoff from meteorological observations. Zhang et al. [2018] used LSTMs for predicting and monitoring sewer overflow. Additionally, Fang et al. [2017] used LSTMs to predict soil moisture with high fidelity.

Reviewer Comment 3.11 — P 4 L 2 "select" -"selected"

Response: Thanks for the catch. It will be modified in the manuscript.

Reviewer Comment 3.12 — P5 L15: 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?

Response: The correlation intensities are shown in wavelet power spectrum (WPS) figures using squared wavelet coefficients which yield information of the correlation between the signal at certain scale at particular location. A larger amplitude in WPS (e.g., the log10(WPS) is larger than 0.2) indicates a higher correlation which could be represented using the color codes in the figures.

Reviewer Comment 3.13 — 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.

Response: Thanks for the comment. We will add more explanation that the periods listed in table 1 and figure 6 are the trained prediction windows for the models. The 24, 48, and 72 hour gaps in figure 7 and stated in P5 L27 are for testing the model performance.

Reviewer Comment 3.14 — P5 L23 delete "clearly"

Response: Agreed. It will be deleted in the manuscript.

Reviewer Comment 3.15 — 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".

Response: Thank you for the comment. We will update the sentence to correctly describe the model.

Old sentence: The DNN architecture is shown in Figure 4, which contains three LSTM layers, followed by two consecutive dropout layers, a convolutional layer, and a final output dense layer

New sentence: The DNN architecture is shown in Figure 4, which contains three LSTM layers, followed by two dense layer with dropout, a convolutional layer, and a final output dense layer

Reviewer Comment 3.16 — 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.

Response: Thank you for the comment. We will remove the sentence.

Reviewer Comment 3.17 — P7 L7 "select" - "selected"

Response: Thanks for the catch. It will be modified in the manuscript.

Reviewer Comment 3.18 — 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".

Response: Thank you for the comment. We will update the sentence to say "A linear layer with a sigmoid activation function".

Old sentence: Each gate is composed of a sigmoid neural net layer and a pointwise multiplication operation.

New sentence: Each gate is composed of a linear layer with a sigmoid activation function.

Reviewer Comment 3.19 — P7 L17: The pointwise multiplication is not part of the gate it-self, but how the gate is combined with the cell state.

Response: Thank you for the comment. We will update the sentence to distinguish the multiplication from the gate. See the above response (3.18) for the updated sentence

Reviewer Comment 3.20 — P7 L18 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

Response: Thank you for the comment. We will update the gate letters accordingly.

Reviewer Comment 3.21 — P7 L 23: "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.

Response: Thank you for the comment. The instruction of output gate will be rephrased in the manuscript.

Reviewer Comment 3.22 — 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?

Response: There is no particular reason why 96h is excluded.We are trying to find the best parameters in training model and a lot of combinations need to be tested. Even without the 96hr output window, we still could obtain the same conclusion from Figure 6d since the MAPE keeps increasing when the output window is greater than 24hr.

Reviewer Comment 3.23 — P10 L 22 How are the terms (P, D, Q)m combined into equation 2. This needs more explanation.

Response: The equation 2 only contains non-seasonal terms (p, d, q) which are number of autoregressive terms, the number of nonseasonal differences and the number of moving-average terms. The terms (P, D, Q) are three additional numbers to represent the seasonal part of an ARIMA.

Reviewer Comment 3.24 — P11 L 19: In your setting, you always extrapolate. So this statement is not correct.

Response: Thank you for the comment. We will correct the statement in the manuscript.

Reviewer Comment 3.25 — P11 L 32: delete "very"

Response: Yes. It is deleted in the manuscript.

Reviewer Comment 3.26 — 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).

Response: From the reviewers comment on 3.3e, we will redo the LSTM analysis to only include the test results for the models on the same well it was trained for to be in line with the training/testing performed with the ARIMA analysis.

Reviewer Comment 3.27 — Figure 7: Missing the information that results are only for SpC.

Response: Thank you for the comment. We will update the figure to explicitly state the results are for SpC only

Reviewer Comment 3.28 — The point above applies to the entire section here.

Response: Thank you for the comment. We will update the section to explicitly state the results are for SpC only

Reviewer Comment 3.29 — P12 L15: "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.

Response: We will update the sentence to limit the scope to our experimental runs.

Old sentence: It is noted that the optimal input window size M for the LSTM models is smaller than that required by the ARIMA method for all the wells tested, indicating that LSTM models can rely on less input information than the ARIMA models to produce predictions of comparable accuracy.

New sentence: In the experimental runs for the LSTM models, the input window size M is smaller than that required by the ARIMA method for the wells tested. This indicates the LSTM models can rely on less input information than the ARIMA models to produce predictions of comparable accuracy.

Reviewer Comment 3.30 — 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.

Response: We can provide a zoomed-in plot of the ARIMA and LSTM predictions for well 1-1 in figure 8 (January 2011 to March 2011) that shows this behaviour and provide quantitative measurements

Reviewer Comment 3.31 — Figure 8. Why are the results now with the ARIMA model and 72 hour inputs and not 168 as in Figure 7?

Response: The intent of figure 8 was to compare the performance of the ARIMA model and LSTM model when only given 72 hours of previous data for filling in gaps of 24 hours. We will update figure 8 to be for the ARIMA models trained with 168 inputs. Additionally, we will update figure 8 to be for the LSTM models mentioned in figure 7, in order to remove testing models on wells it was not trained on.

Reviewer Comment 3.32 — 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)

Response: Thank you for the comment. We can add a table of the mean relative error and MAPE for the LSTM and ARIMA models for each well in Figure 8, column 1 and 2.

Reviewer Comment 3.33 — "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.

Response: As mentioned in the previous comment, we can add a table showing the MAPE and mean relative error of the LSTMs and ARIMA models as well as adding an additional figure showing the LSTM capturing the high-frequency fluctuations.

Reviewer Comment 3.34 — 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)

Response: Thank you for the comment. We will redo our conclusions section based on the additional analysis we will do. As previously stated, we will explicitly mention we are gap filling for the SpC measurement only.

References

Mathias Berglund, Tapani Raiko, Mikko Honkala, Leo Kärkkäinen, Akos Vetek, and Juha T Karhunen. Bidirectional recurrent neural networks as generative models. In C. Cortes, N. D.

Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 856–864. Curran Associates, Inc., 2015. URL http://papers.nips.cc/paper/5651-bidirectional-recurrent-neural-networks-as-generative-models.pdf.

X. Du, H. Zhang, H. V. Nguyen, and Z. Han. Stacked lstm deep learning model for traffic prediction in vehicle-to-vehicle communication. In *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*, pages 1–5, Sep. 2017. doi: 10.1109/VTCFall.2017.8288312.

Kuai Fang, Chaopeng Shen, Daniel Kifer, and Xiao Yang. Prolongation of smap to spatiotemporally seamless coverage of continental u.s. using a deep learning neural network. *Geophysical Research Letters*, 44(21):11,030–11,039, 2017. doi: 10.1002/2017GL075619. URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2017GL075619.

A. Graves, A. Mohamed, and G. Hinton. Speech recognition with deep recurrent neural networks. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 6645–6649, May 2013. doi: 10.1109/ICASSP.2013.6638947.

Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8): 1735–1780, November 1997. ISSN 0899-7667. doi: 10.1162/neco.1997.9.8.1735. URL https://doi.org/10.1162/neco.1997.9.8.1735.

M. JORDAN. Attractor dynamics and parallelism in a connectionist sequential machine. *Proc. of the Eighth Annual Conference of the Cognitive Science Society (Erlbaum, Hillsdale, NJ), 1986*, 1986. URL https://ci.nii.ac.jp/naid/10018634949/en/.

M Khalil, U.S Panu, and W.C Lennox. Groups and neural networks based streamflow data infilling procedures. *Journal of Hydrology*, 241(3):153 – 176, 2001. ISSN 0022-1694. doi: https://doi.org/10.1016/S0022-1694(00)00332-2. URL http://www.sciencedirect.com/science/article/pii/S0022169400003322.

F. Kratzert, D. Klotz, C. Brenner, K. Schulz, and M. Herrnegger. Rainfall–runoff modelling using long short-term memory (lstm) networks. *Hydrology and Earth System Sciences*, 22(11):6005–6022, 2018. doi: 10.5194/hess-22-6005-2018. URL https://www.hydrol-earth-syst-sci.net/22/6005/2018/.

Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision*, pages 816–833. Springer, 2016.

Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to construct deep recurrent neural networks, 2013.

K. Saleh, M. Hossny, and S. Nahavandi. Intent prediction of vulnerable road users from motion trajectories using stacked lstm network. In *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, pages 327–332, Oct 2017. doi: 10.1109/ITSC.2017.8317941.

Sijie Song, Cuiling Lan, Junliang Xing, Wenjun Zeng, and Jiaying Liu. An end-to-end spatio-temporal attention model for human action recognition from skeleton data. In *AAAI*, volume 1, pages 4263–4270, 2017.

Ketaki Ustoorikar and M.C. Deo. Filling up gaps in wave data with genetic programming. *Marine Structures*, 21(2):177 – 195, 2008. ISSN 0951-8339. doi: https://doi.org/10.1016/j.marstruc.2007.12.001. URL http://www.sciencedirect.com/science/article/pii/S0951833907000676.

Duo Zhang, Geir Lindholm, and Harsha Ratnaweera. Use long short-term memory to enhance internet of things for combined sewer overflow monitoring. *Journal of Hydrology*, 556:409 – 418, 2018. ISSN 0022-1694. doi: https://doi.org/10.1016/j.jhydrol.2017.11.018. URL http://www.sciencedirect.com/science/article/pii/S0022169417307722.

Wentao Zhu, Cuiling Lan, Junliang Xing, Wenjun Zeng, Yanghao Li, Li Shen, and Xiaohui Xie. Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks, 2016. URL https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11989/12149.