Response to the short comments

Reviewer 1

Reviewer Comment 1.1 — This paper uses long short-term memory (LSTM) neural networks to fill in gaps in spatially distributed time-series data. The performance of the LSTM-based gap-filling method is compared to that of a traditional, popular gapfilling method: autoregressive integrated moving average (ARIMA). Overall, this paper is well written, structured and results seem sufficiently justified and useful. However, this paper is very technical and there is no physical insight beyond just feeding data into a standard code. I think this paper should be published as technical note (not as research article). Several aspects could be further improved in order to having it published in this journal.

Response: We thank the reviewer for overall positive assessment of our manuscript. While we emphasized the scientific importance of gap-filling the spatio-temporal data to capture and understand dynamic behaviors of complex systems, we also agree with the reviewer that our primary focus was to introduce a technical method that can fill data gaps by capturing the dynamic features using deep neural network that contains LSTM layers. We are resubmitting this manuscript as a technical note as suggested.

Reviewer Comment 1.2 — Would you guarantee the LSTM method in your paper can achieve the same excellent performance in other areas of the whole world? Is it possible that the good performance of the LSTM model is just applicable for the case given by the manuscript? The authors should include one more test for another area (maybe not in the text but in the supporting materials).

Response: There is unfortunately no guarantee for any model to have the same performance in other applications, which is the case for all data-driven and physics-based models. The performance of data-driven models can be optimized by iterating on various model configurations based on data types and characteristics of the relations between predictors and desired responses, as we have demonstrated in our study case when comparing ARIMA with LSTM-based DNNs. Sometimes the ARIMA works better, and sometimes the DNNs work better, and we believe that we have started down the path of predicting which model will work better for a given case. The LSTM-based DNN model we adopted in our study has the same chain-like nature as other recurrent neural networks, meaning that this architecture lends itself well to sequences, so it will often be a useful (if not the best) approach for dynamic system behaviors (Karim et al., 2017; Malhotra et al., 2015; Kratzert et al., 2018; Malhotra et al., 2016; Wang et al., 2017; Reddy and Prasad, 2018; Lipton et al., 2015). The optimal model configuration and performance we can achieve would be case by case, and our focus of this technical note is to introduce a general method that can be broadly applied to other systems and be evaluated similarly. We have emphasized this aspect of transferrability in our discussion and conclusion sections. We call on the community participation to test the transferrability of this method to other monitoring systems.

Reviewer Comment 1.3 — LSTM model is only compared to ARIMA. Why not compare LSTM with other widely-used methods (such as Kriging interpolation and Gaussian process)? Furthermore, are the authors familiar with DIEOF (Data Interpolating Empirical Orthogonal Functions) which are proposed by Beckers and Rixen (2003)? I think that DIEFOF is powerful and

useful for filling temporal and spatial gaps in geophysical datasets. Maybe the authors can compare LSTM with DIEOF.

Response: There are many interpolation approaches that are commonly used, including the EOF-based approaches and kriging (based on Gaussian processes) that the reviewer mentioned. We did discuss that kriging and the Gaussian processes are mostly used for spatial interpolation rather than spatio-temporal interpolation. We have added more discussions on the EOF related interpolation methods, such as least squares EOF (LSEOF), data interpolation EOF (DINEOF), and recursively subtracted EOF (REEOF), which are widely used to fill in missing data from geophysical fields such as clouds in sea surface temperature datasets or other satellite-based images with regular gridded domains. However, as discussed by Beckers and Rixen (2003), "For the method to have a chance to work, one needs, for each moment, at least a sufficiently large number of data points (otherwise one should drop the whole picture) and for each spatial point a sufficient amount of data in time (otherwise one should discard the point from the analysis)", which is a challenge for most of the monitoring networks that are sparsely distributed. Therefore, we keep using ARIMA as the benchmark since it is the most commonly used method in time series analysis (a conclusion based on reviewing the hydrological literature) and gap filling. While we acknowledge that we did not (and could not) explore every possible interpolation method, we feel that by choosing such a representative approach is appropriate for assessing the performance of our proposed method without loss of generality. We have also acknowledged this aspect in our introductions, discussion and conclusion sections.

New description of DINEOF has been added in manuscript at P3L1-P3L5: " Empirical Orthogonal Functions (EOF) related interpolation methods, such as least squares EOF(LSEOF), data interpolation EOF (DINEOF), and recursively subtracted EOF (REEOF), are widely used to fill in missing data from geophysical fields such as clouds in sea surface temperature datasets or other satellite-based images with regular gridded domains (Beckers and Rixen, 2003; Beckers et al., 2006; Alvera-Azcárate et al., 2016). However, the requirement of gridded data by the EOF methods limits their use in filling data gaps in irregularly spaced monitoring networks."

Reviewer Comment 1.4 — The present title "Using Deep Learning to Fill Spatio-Temporal Data Gaps in Hydrological Monitoring Networks" are inaccurate. I suggest new title like "Using Long Short Term Memory Neural Network Model to Fill Spatio-Temporal Data Gaps in Hydrological Monitoring Networks"

Response: Our neural network models contain both LSTM and CNN layers. Therefore, we modified our title to : "Technical note: Using Deep Neural Network Models to Fill Spatio-Temporal Data Gaps in Hydrological Monitoring Networks". Deep neural network is a broader term that allows flexible architecture we are using to include multiple types of layers.

References

Alvera-Azcárate, A., Barth, A., Parard, G., and Beckers, J.-M.: Analysis of SMOS sea surface salinity data using DINEOF, Remote Sensing of Environment, 180, 137 – 145, https://doi.org/https://doi.org/10.1016/j.rse.2016.02.044, URL `http://www.sciencedirect.com/science/article/pii/S0034425716300724`, special Issue: ESA's Soil Moisture and Ocean Salinity Mission - Achievements and Applications, 2016.

Beckers, J. M. and Rixen, M.: EOF Calculations and Data Filling from Incomplete Oceanographic Datasets*, Journal of Atmospheric and Oceanic Technology, 20, 1839–1856, https://doi.org/10.1175/1520-0426(2003)020⟨1839:ECADFF⟩2.0.CO;2, URL https://doi.org/10.1175/1520-0426(2003)020<1839:ECADFF>2.0.CO;2, 2003.

Beckers, J. M., Barth, A., and Alvera-Azcárate, A.: DINEOF reconstruction of clouded images including error maps. Application to the Sea-Surface Temperature around Corsican Island, Ocean Science Discussions (OSD), 2, https://doi.org/10.5194/os-2-183-2006, 2006.

Karim, F., Majumdar, S., Darabi, H., and Chen, S.: LSTM fully convolutional networks for time series classification, IEEE access, 6, 1662–1669, 2017.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrology and Earth System Sciences, 22, 6005–6022, https://doi.org/10.5194/hess-22-6005-2018, URL https://www.hydrol-earth-syst-sci.net/22/6005/2018/, 2018.

Lipton, Z. C., Kale, D. C., and Wetzel, R. C.: Phenotyping of clinical time series with LSTM recurrent neural networks, arXiv preprint arXiv:1510.07641, 2015.

Malhotra, P., Vig, L., Shroff, G., and Agarwal, P.: Long short term memory networks for anomaly detection in time series, in: Proceedings, vol. 89, Presses universitaires de Louvain, 2015.

Malhotra, P., Ramakrishnan, A., Anand, G., Vig, L., Agarwal, P., and Shroff, G.: LSTM-based encoder-decoder for multi-sensor anomaly detection, arXiv preprint arXiv:1607.00148, 2016.

Reddy, D. S. and Prasad, P. R. C.: Prediction of vegetation dynamics using NDVI time series data and LSTM, Modeling Earth Systems and Environment, 4, 409–419, 2018.

Wang, Q., Guo, Y., Yu, L., and Li, P.: Earthquake prediction based on spatio-temporal data mining: an LSTM network approach, IEEE Transactions on Emerging Topics in Computing, 2017.

Response to referee comments

Reviewer 2

General Remarks: The paper presents an interesting use of deep learning with LSTM Networks for infilling groundwater data. The article is timely and tries to make a comprehensive description and explanation of how the Deep learning technique is implemented using statistical and machine learning techniques. The paper is a welcome contribution to the field of groundwater and hydrological earth sciences. However, I cannot recommend publication in the present form due to the comments and questions raised. The paper needs major revision.

Response: We thank the reviewer for the positive assessment of our manuscript and the constructive comments. We have addressed all the individual comments to improve our manuscript for possible publication.

Reviewer Comment 2.1 — The paper states that long-term spatiotemporal changes in subsurface hydrological flow is usually quantified using a network of wells. However this paper does not deal with the long-term trend or analysis. Hourly data is hardly interpreted or used for the long term. Hourly information for sure contains noise that would be advisable to remove for the long term analysis.

Response: There seems to be some confusion between long-term changes and long-term trends. We have rephrased to clarify that monitoring networks capture the dynamic system behaviors over long time windows, which allows the discovery of signals over a spectrum of time scales as revealed by the spectral analysis we presented in the manuscript. While long-term trends, e.g., low-frequency variations, are usually smoother and could be captured by existing time series analysis method like ARIMA (see our comparison analyses between LSTM-based DNN and ARIMA gap filling results), our focus is on capturing high-frequency dynamics that are important signatures for understanding managed systems. We have clarified those points in our objectives and reiterate in conclusions to avoid confusion.

Reviewer Comment 2.2 — Observations are mentioned to be spatially sparse, and temporal gaps exist. Many papers have solved the same type of problem, without using the term spatiotemporal. Almost every course in hydrology deals in one chapter with the issue of using spatial correlation and temporal correlation to fill in data. So in this respect, the authors are invited to clearly indicate what innovation is brought by this work to spatiotemporal analysis.

Response: We agree with the reviewer that a lot has been done in hydrology for spatial and temporal analyses. However, there have been very few studies that address spatial and temporal correlations simultaneously due to the difficulty in parameterizing the spatial and temporal correlations all together. Deep neural networks provide an alternative way to represent such correlations without assuming the explicit form of correlations a priori, which is the innovation our work originally aimed to bring and demonstrate. However, we found there are multiple steps towards accomplishing that goal as it involves merging two types of deep neural networks to represent both the spatial and temporal correlations and evaluate various configurations thoroughly. In order to bring the spatial component together with the temporal correlations, also related to a comment raised by Reviewer #1, we added multi-well DNN models to take advantage of information from neighboring wells. The multi-well DNN models were found

to ourperform their single-well counterparts as shown in the newly added Section 4.3. We emphasized in the introduction and conclusion that the primary advantage of using DNN approach is to address both spatial and temporal correlations without assuming their explicit forms before hand.

Reviewer Comment 2.3 — Following point two, it is known that in most of the cases, aquifers with little or no human intervention have low variability. Conventional guidelines and measures in hydrogeological science are typically based on monthly data.

Response: It is true that some aquifers with no or little human intervention have low variability, for which monthly data could be sufficient to understand the system behavior. However, anthropogenic activities, in particular, dam operations, have increasingly impacted the river and aquifer systems by altering the exchange patterns between river water and groundwater, and the associated thermal and biogeochemical processes (Song et al., 2018; Shuai et al., 2019; Zachara et al., 2020). Due to significant, high-frequency (hourly) stage variations caused by dam regulations to meet power generation needs, it is insufficient to use monthly data to understand such systems as have been demonstrated in numerous studies performed at our study site. Our study site is representative of many dam-regulated gravel-bed rivers across the world. Therefore, our study could have broader impacts to many other systems. We have made this point clear in our study site description section.

We added a paragraph at P4 L30-P5 L6: "The understanding we developed from earlier studies is that the physical heterogeneity contributes to the different response behaviors at different locations while the river stage dynamics lead to multi-frequency dynamics in those responses. The seasonal and annual variations are driven by natural climatic forcing (Amaranto et al., 2019, 2018), whereas the higher-frequency (i.e., daily and sub-daily) fluctuations are primarily induced by operations of the upstream hydroelectric dam operations to meet various demands of human society (Song et al., 2018). Our system is representative of many dam-regulated gravel-bed rivers across the world, where dam operations as a typical anthropologenic activity have significantly altered the hydrologic exchanges between river water and groundwater, as well as the associated thermal and biogeochemical processes (Song et al., 2018; Shuai et al., 2019; Zachara et al., 2020). Note that the multi-frequency variations in data are characterizing the dynamic features of data, which could exist in both short-term and long-term time series data as a result of short-term or long-term monitoring effort."

Reviewer Comment 2.4 — In the present paper the idea of nonlinear dynamics is mentioned almost everywhere in the introduction and justification of the work. This is somewhat surprising and needs better justification, since groundwater dynamics, in many cases, can be represented with linear models. As it is concluded in this paper results, ARIMA can approximate the system quite well.

Response: This comment is related to the earlier comment 2.3 as high-frequency dynamics lead to higher level of nonlinearity in system responses, especially for the specific conductance that is a result of mixing of water from various sources. We have shown that a linear model like ARIMA was not able to capture such nonlinearity, while LSTM-based DNN performed better. We have explained this point in the revised manuscript. Please also refer to our response to comment 2.3 for the importance of capturing high-frequency dynamics for many dam-regulated systems, which was also better articulated in the revised manuscript.

Reviewer Comment 2.5 — The particular case study presented here shows a relative complex

dynamic nature indeed, but it seems it is due to human intervention (however I could be wrong). Can you comment on this and the uncertainties associated?

Response: The reviewer is partially right that human intervention contributes to the complexity of system behavior by creating high-frequency flow dynamics. However, the full complexity is a result of interactions between such human-induced variations and the natural heterogeneity of aquifer physical properties (Zachara et al., 2020). There is significant uncertainty associated with aquifer physical heterogeneity at our study site as revealed by previous studies. We have added these additional discussion about the system complexity in the revised manuscript.

a The human intervention might affect your calculation and therefore, extractions might not be following a random but more human induced behaviour. So data understanding or replicability used in one year might not be the same in another. It would be advisable first to check how much and when extraction took place. Is this data filled in for a long term analysis, or short-term? This question arises since the hourly step is used.

Response: Please refer to our response to Comment 2.1 for explaining our use of long-term data versus long-term trend analyses. The reviewer is right that high-frequency flow variations are mainly caused by the dam operations while the seasonal and interannual variabilities are controlled by climatic forcing like precipitation and melting of snow pack in the headwater systems. We have clarified the drivers of the high-frequency and low-frequency variations in the revision. We used multiple years of training data from dry, normal and wet hydrologic years to capture potential operational patterns associated with various conditions. LSTM units include a 'memory cell' that can maintain information in memory for long periods of time.

The model learns to extract the information from the input during the training process, typically by the end of 30 epochs we train the model for. The amount of information maintained by the LSTM is dependent on the size of the input time series. Future work is required to identify what information the LSTM model remembers in the cell, i.e., opening the black box of the DNN model. While this is a very important next step and an active research area, it is beyond the scope of this technical note to describe the relevant method and results imagining the level of effort that is required. After the data gaps are filled, the dataset can be used for both long-term or short-term analyses as we will have long-term dataset with high temporal resolution.

To illustrate what can be done to open the DNN box, we are providing an example where we examined the memory cells of a model with an input window size of 120 hours, output of 1 hour, trained on data from well 1-1. We investigated which of the inputs (SpC, water level, temperature) drives the state of the LSTM units, i.e., which input has the most influence on the LSTM units, for the validation period (year 2011). We divided the data into 8640 input samples of 120 hours each and ran each input through the trained model. We then extracted the hidden state of each unit in the three LSTM layers at each time sample and calculated the Pearson correlation coefficient between the hidden states of all the units and the normalized input. Next, we calculated the percentage of the 8640 samples for each input variable to have the largest positive and negative correlations with the hidden states of the LSTM units in each of the three layers. The results for the top positive and negative correlations are show in table 2.1, from which we see that the water level occurs the most frequent (almost 50% of cases) as the highest positive and negative correlation for the LSTM units in layers 1 and 2. SpC is consistently the second most frequent as the highest correlated in all the layers. SpC occurs most frequently (42.62%) as the highest

negatively correlated with the hidden states in the third LSTM layer. Thus, for this given model, the memory of the LSTM units are most strongly driven by the dynamics of the water level in the input, then followed by the dynamics of the SpC.

Layer 1			
Rank	**SpC (%)**	**Temperature (%)**	**Water Level (%)**
Highest Positive	32.6%	14.9%	52.4%
Highest Negative	32.4%	22.1%	45.6%

Layer 2			
Rank	**SpC (%)**	**Temperature (%)**	**Water Level (%)**
Highest Positive	28.5%	20.2%	51.3%
Highest Negative	29.6%	16.3%	54.2%

Layer 3			
Rank	**SpC (%)**	**Temperature (%)**	**Water Level (%)**
Highest Positive	28.2%	20.2%	51.6%
Highest Negative	42.6%	17.9%	39.4%

Table 2.1: Percent of input samples in 2011 (8640 in total) for each measurement to have the largest positive and negative Pearson correlation coefficient (R) with the hidden states of each LSTM layer

Taking a closer look, as seen in figure 2.1, the top positively correlated LSTM unit with SpC (R = 0.897) closely follows the dynamics of the SpC.

b If indeed human intervention influence the dynamics of the groundwater system, the logical approach would be to find a variable to represent direct or indirect measurement of extractions.

Response: Thanks for the suggestion. Please refer to our response to your earlier comment. We have looked into both memory cells and state cells to illustrate how and where the extraction occurs.

c It is suggested to read the paper by Amaranto et al. (2018) "Semi-seasonal groundwater forecast using multiple data-driven models in an irrigated cropland". J Hydroinformatics, 20 (6): 1227–1246. DOI: `https://doi.org/10.2166/hydro.2018.002` and - Amaranto et al. (2019). A spatially enhanced data driven multimodel to improve semiseasonal groundwater forecasts in the High Plains aquifer, USA. Water Resources Research, 55, 5941– 5961. `https://doi.org/10.1029/2018WR024301`

Response: Thank you for the paper suggestions. Both of the papers listed above use data-driven approaches to improve groundwater forecasts. The MuMoC framework select neighboring wells to assist groundwater predictions is of our interest. Although the authors used data with coarser temporal resolution (daily or monthly) to make monthly predictions, which is different from our purpose of filling short gaps (up to 3 days) for capturing high-frequency dynamics, the idea of using information from neighboring wells applies to our case. We have explored multi-well DNN

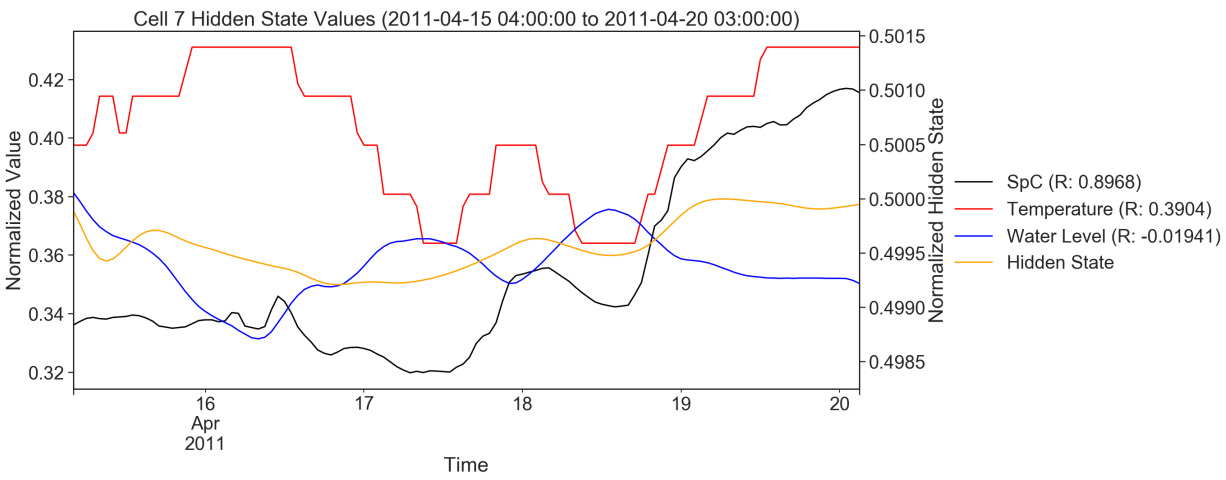

Figure 2.1: Well 1-1 normalized measurements over time (2011-04-15 04:00:00 to 2011-04-20 03:00:00), along with the hidden state of the top positvely correlated LSTM unit in the third layer

models to use information from neighboring wells, which led to improved accuracy in gap filling. Please refer to the new added section 4.3: Performance of multi-well DNN models. We have reviewed and discussed these two papers in our revision.

Reviewer Comment 2.6 — The regional aquifer and geology might play a more significant role in the study, since not only the river but the size and other interventions and hydrometeorological recharges might be correlated.

Response: We agree that the regional aquifer and geology play an important role as shown in previous studies performed by our colleagues (Chen et al., 2012, 2013; Zachara et al., 2020). The aquifer is composed of two distinct geologic formations, a highly permeable formation (Hanford formation, consisting of coarse gravelly sand and sandy gravel) underlain by a much less permeable formation (the Ringold Formation, consisting of silt and fine sand). The dominant hydrogeologic features of the aquifer are defined by the interface between the Hanford and Ringold formations and the heterogeneity within the Hanford formation. The understanding we developed from these earlier studies is that the physical heterogeneity contributes to the different response behaviours at different locations while the river stage dynamics lead to multi-frequency dynamics in those responses. We have added more information and re-organized the entire section 2: Study Site and Data Description to better describe our system to help readers understand. The recharge from rainfall is negligible due to the semi-arid climate.

Reviewer Comment 2.7 — The stations are so close, and the hourly variation appears to be periodic with an amplitude of 4 or 5cm, according to Figure 1 (and on other graphs). It is intriguing, the question I would have is what happens every hour? and if this hourly variation is noise on the measurement device or data? What is the precision of the measurement device? What is the volume of water extracted to reach the variation of 1 cm? Where the recharge water comes from(has this been studied in the past)? Is this 5 cm recharge volume feasible in one hour? Could

be the water from the river affecting your measurements (interflow)? It is advisable to present the time series of the river flow. It would be also useful to have a few hydrological balances (note that this is a hydrological journal). The problematic still can be questioned due to its apparent complex dynamics with the river and human intervention (not a typical, natural aquifer).

Response: The reviewer is right that the water table elevation difference is small due to the close distance between wells and the highly permeable aquifer material (hydraulic conductivity in the range of 4000-7000 m/d). The rapid change in groundwater table is at first caused by pressure wave propagation from the river stage variation, and then by recharge water coming from the river or displacement of groundwater from other parts of the aquifer depending on flow directions and locations of interest. The stainless-steel pressure transducer CS451 from campbell scientific (Scientific) was used for water level measurements. The measurement range of the pressure transducer in our study site is 0 to 10.2m with a standard accuracy of $\pm 0.1\%$, which leads to an accuracy of 1cm. In this case, the pressure changes with an amplitude of 4 5cm are the actual measurements and our consistent 10-year data has proven this point. In our revision, we have included the measurement accuracy information and numerous hydrologic modeling studies(Song et al., 2018; Shuai et al., 2019; Zachara et al., 2020) performed at the site to help readers better understand the flow conditions and where the recharging water comes from.

Reviewer Comment 2.8 — On the model setup, Please explain why you use Mx128.

Response: We use 128 (i.e. 128 units for each LSTM layer) because this number of units showed better performance after we experimented with different model architectures with different number of units. We have added this rationale to the manuscript at P10 L5-L6 "Each of the three LSTM layers has 128 units because this configuration outperformed others with more or fewer number of units".

Reviewer Comment 2.9 — Page 7, line 10, mentions the supplemental material, but I cannot find it in the paper.

Response: The supplemental materials can be found using this link
https://www.hydrol-earth-syst-sci-discuss.net/hess-2019-196/hess-2019-196-supplement.pdf

Reviewer Comment 2.10 — Important: choice of (a very complex model) LSTM has to be justified, since it seems AR-type models is enough. Frankly, I don't see the need for complex models like LSTM, but if you have arguments to defend your position, please present them to convince the readers.

Response: We are interested in using an LSTM-based DNN for this problem because DNNs have had success in predicting values in time-series data without assuming explicit temporal dependence forms. We added several examples of this on P3 L12. We aimed to explore whether an DNN model would provide improved performance over traditional methods (i.e. ARIMA). Our study demonstrated that more complex DNN models were able to capture high-frequency variations in system dynamics, for which a simpler ARIMA model failed to capture. The choice of the DNN architecture was a result of hyperparameter search based on the validation performance of the models. Related to our response to other relevant comments, we restructured the methodology section and added more reasoning for the choice of the DNN architecture and the advantages and necessity of using LSTM-based DNNs in

our revision. For example, We have expanded the introduction section to discuss recent applications of LSTMs to hydrology and earth sciences and the relevance to gap filling problems.

New section (P3 L21-P43): "There have been applications of RNNs and LSTMs emerging in hydrology. For example, Kratzert et al. (2018) used LSTMs to predict watershed runoff from meteorological observations, Zhang et al. (2018) used LSTMs for predicting sewer overflow events from rainfall intensity and sewer water level measurements, and Fang et al. (2017) used LSTMs to predict soil moisture with high fidelity. Compared to a single RNN/LSTM layer, more complex LSTM architectures such as stacked and bidirectional LSTMs, CNN-LSTM or convolutional LSTM have the potential to capture extra features (Graves et al., 2013; Pascanu et al., 2013) as shown in various applications, including action recognition (Zhu et al., 2016) and vulnerable road users location predictions (Saleh et al., 2017). A bidirectional RNN/LSTM works by duplicating the recurrent network into two networks: one responsible for fitting the positive time direction (i.e. the forward states) and the other responsible for the negative time direction (i.e the backwards state)(Schuster and Paliwal, 1997). In general, the input sequence is fed as-is to the forward state and a reversed copy of the input sequence is fed to the backwards state. The bidirectional LSTM can be used in history matching problems.

Our study aims to evaluate the potential of using LSTM layers within a DNN architecture to fill gaps in spatio-temporal environmental time series. We treat the gap filling as a forecasting problem, i.e., we use the historical data as input to predict the missing values in the data gaps. We demonstrate our method using a test case that focuses on understanding the interactions between a regulated river and contaminated groundwater aquifer. We adopt the stacked-LSTM combined with the convolutional layer as our DNN model to understand the interactions between a regulated river and contaminated groundwater aquifer. The DNN-based gap filling method is compared with traditional time series approaches (e.g., ARIMA) to identify situations in which DNNs outperform ARIMA as well as what the optimal configurations might be for this particular application."

Reviewer Comment 2.11 — On page 14, it states that other configurations of LSTM can be further explored; however, it is not clear why this was not done before. Not sure why the selected configuration was just tried to see if it works or not, without any analysis what is the best structure. This relates to comment 8 and 9.

Response: We acknowledge that we did not explain this point as well as we could have. 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. However, this was not an exhaustive search of all possible configurations. We have dedicated a subsection 3.1.1 on hyperparameter search to make this clear.

Reviewer Comment 2.12 — I am a bit in confusion how to interpret the statements made in conclusion. The ARIMA is not suited or less suited for filling high frequency (hourly, or short gaps) and more suitable for a long term period (24, 48 and 74 hours). It is suggested we need deep learning for filling high-frequency gaps (of one hour)?. Maybe is good to elaborate on the simplicity of what this translates to, I am not sure if the meaning is right.

Response: We acknowledge a potential source of confusion in terms of high-frequency fluctuations in system states versus short-term or long-term data records/gaps. In the revised manuscript, we made extra effort to distinguish those two concepts to avoid further confusion. In our study, we found ARIMA

to be suitable for time series with less dynamic behavior, while DNNs excel in capturing high-frequency dynamics (daily and subdaily) in time-series observations for various data gaps.

Reviewer Comment 2.13 — Not sure if there is an idea of how high is the overall error; in the figure 8, with well 1-15 it seems almost perfect representation (zero error in the validation data for many points). Also in the same well, it appears like high negative correlation up to 128 hours.

Response: Well 1-15 has the smallest error as been illustrated in Figures 8 and 9. The mean error shown in testing period in Figure 9 is 0.05%. Please note that the log10 scale of wavelet power spectrum is plotted in Figure 9, where the blue colors represent weak power rather than negative correlation.

References

Amaranto, A., Munoz-Arriola, F., Corzo, G., Solomatine, D. P., and Meyer, G.: Semi-seasonal groundwater forecast using multiple data-driven models in an irrigated cropland, Journal of Hydroinformatics, 20, 1227–1246, 2018.

Amaranto, A., Munoz-Arriola, F., Solomatine, D., and Corzo, G.: A spatially enhanced data-driven multimodel to improve semiseasonal groundwater forecasts in the High Plains aquifer, USA, Water Resources Research, 55, 5941–5961, 2019.

Chen, X., Murakami, H., Hahn, M. S., Hammond, G. E., Rockhold, M. L., Zachara, J. M., and Rubin, Y.: Three-dimensional Bayesian geostatistical aquifer characterization at the Hanford 300 Area using tracer test data, Water Resources Research, 48, https://doi.org/10.1029/2011WR010675, URL `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2011WR010675`, 2012.

Chen, X., Hammond, G. E., Murray, C. J., Rockhold, M. L., Vermeul, V. R., and Zachara, J. M.: Application of ensemble-based data assimilation techniques for aquifer characterization using tracer data at Hanford 300 area, Water Resources Research, 49, 7064–7076, https://doi.org/10.1002/2012WR013285, URL `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2012WR013285`, 2013.

Fang, K., Shen, C., Kifer, D., and Yang, X.: Prolongation of SMAP to Spatiotemporally Seamless Coverage of Continental U.S. Using a Deep Learning Neural Network, Geophysical Research Letters, 44, 11,030–11,039, https://doi.org/10.1002/2017GL075619, URL `https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2017GL075619`, 2017.

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

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrology and Earth System Sciences, 22, 6005–6022, https://doi.org/10.5194/hess-22-6005-2018, URL `https://www.hydrol-earth-syst-sci.net/22/6005/2018/`, 2018.

Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y.: How to Construct Deep Recurrent Neural Networks, 2013.

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

Schuster, M. and Paliwal, K. K.: Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing, 45, 2673–2681, 1997.

Scientific, C.: CS451:Stainless-Steel Pressure Transducer, URL https://www.campbellsci.com/cs451.

Shuai, P., Chen, X., Song, X., Hammond, G. E., Zachara, J., Royer, P., Ren, H., Perkins, W. A., Richmond, M. C., and Huang, M.: Dam Operations and Subsurface Hydrogeology Control Dynamics of Hydrologic Exchange Flows in a Regulated River Reach, Water Resources Research, 55, 2593–2612, https://doi.org/10.1029/2018WR024193, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018WR024193, 2019.

Song, X., Chen, X., Stegen, J., Hammond, G., Song, H.-S., Dai, H., Graham, E., and Zachara, J. M.: Drought Conditions Maximize the Impact of High-Frequency Flow Variations on Thermal Regimes and Biogeochemical Function in the Hyporheic Zone, Water Resources Research, 54, 7361–7382, https://doi.org/10.1029/2018WR022586, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018WR022586, 2018.

Zachara, J. M., Chen, X., Song, X., Shuai, P., Murray, C., and Resch, C. T.: Kilometer-scale hydrologic exchange flows in a gravel-bed river corridor and their implications to solute migration, Water Resources Research, n/a, e2019WR025 258, https://doi.org/10.1029/2019WR025258, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR025258, e2019WR025258 2019WR025258, 2020.

Zhang, D., Lindholm, G., and Ratnaweera, H.: Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring, Journal of Hydrology, 556, 409 – 418, https://doi.org/https://doi.org/10.1016/j.jhydrol.2017.11.018, URL http://www.sciencedirect.com/science/article/pii/S0022169417307722, 2018.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X.: Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks, URL https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11989/12149, 2016.

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. We have taken major revisions to address all the comments raised by the reviewer, as illustrated by the point-by-point response below.

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. We have dedicated a subsection 3.1.1 on hyperparameter search.

New description is at P9 L12 to P10 L6: We performed a hyperparameter search to explore different model architecture configurations, i.e., the number of LSTM layers, number of units per LSTM layer, number (and size of) dense layers, and activation functions. The search was performed on well 1-1 only due to computational cost. We chose the optimal DNN architecture using model performance on validation data set of well 1-1 (see Table 1) using MAPE defined in Eq. (1).

The final DNN architecture, as shown in Figure 4, contains three LSTM layers, followed by two dense layers with dropout, a convolutional layer, and a final output dense layer. Stacking three

layers of LSTM was found to yield better performance than a one- or two-layer architecture. Each of the three LSTM layers has 128 units because this configuration outperformed others with more or fewer number of units.

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 investigated the potential benefits of using multiple LSTM layers in our problem domain. We have added additional sentences in the revision to discuss 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 updated our references with the cited articles.

New paragraph (P3 L24-L27): Compared to a single RNN/LSTM layer, more complex LSTM architectures such as stacked and bidirectional LSTMs, CNN-LSTM or convolutional LSTM have the potential to capture extra features (Graves et al., 2013; Pascanu et al., 2013) as shown in various applications, including action recognition (Zhu et al., 2016) and vulnerable road users location predictions (Saleh et al., 2017). .

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 have provided 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 trained models with a single LSTM and dense layer and compared the results against the original architecture (see response in 3.3c).

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 and is another model type applicable to the work if we treat the gap filling as a history matching problem. As we stated in the conclusion section, the bi-directional LSTMs can be explored in future work to keep the scope of this study manageable. We have also added a paragraph describing multiple deep learning techniques that have been applied to gap filling in hydrology, including LSTMs. We have also added a brief description regarding bi-directional LSTMs.

New Description P3 L21-31: There have been applications of RNNs and LSTMs emerging in hydrology. For example, Kratzert et al. (2018) used LSTMs to predict watershed runoff from meteorological observations, Zhang et al. (2018) used LSTMs for predicting sewer overflow events from rainfall intensity and sewer water level measurements, and Fang et al. (2017) used LSTMs to predict soil moisture with high fidelity. Compared to a single RNN/LSTM layer, more complex LSTM architectures such as stacked and bidirectional LSTMs, CNN-LSTM or convolutional LSTM have the potential to capture extra features (Graves et al., 2013; Pascanu et al., 2013) as shown in various applications, including action recognition (Zhu et al., 2016) and vulnerable road users location predictions (Saleh et al., 2017). A bidirectional RNN/LSTM works by duplicating the recurrent network into two networks: one responsible for fitting the positive time direction (i.e. the forward states) and the other responsible for the negative time direction (i.e the backwards state) (Schuster and Paliwal, 1997). In general, the input sequence is fed as-is to the forward state and a reversed copy of the input sequence is fed to the backwards state. The bidirectional LSTM can be used in history matching problems..

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 have re-run the experiments using the zero mean, unit variance normalization. In comparing models trained with the original 0 to 1 scaling normalization technique against the zero mean normalization, the different normalization has a mixed result on model performance. As seen in figure 3.1 of this response, only models trained on wells 1-15 and 2-3 gain a notable improvement in performance, with 100% and 51.28% of model configurations tested on the gap lengths gaining an increase in performance, respectively. However, for wells 1-1, 1-10A, 2-2, and 2-5, only 21.15%, 39.74%, 37.82%, and 27.57% of configurations saw an improvement in performance. As such, while some model configurations do benefit (i.e. models trained on well 1-15 and 2-3), a majority of configurations tested on the four gap lengths do not improve in SpC MAPE performance when using zero mean normalization. We therefore kept our 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: We have re-run our experiment using three additional initialization seeds. Model results presented are now averaged over the initialization seeds.

New sentence P8 L18: Each model configuration was trained using four different initialization seeds and error metrics were averaged to determine the best configuration. .

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. We have used well 1-1 as example to show the comparison of model performance between single-well and multi-well models. We trained a multi-well model using 3 wells (1-1, 1-10A, and 1-16) and compared the performance to the single-well models trained on 1-1. With additional spatial information involved, we have added the extra discussion regarding to this effort under new subsection ("4.3 Performance of multi-well DNN models") on our manuscript.

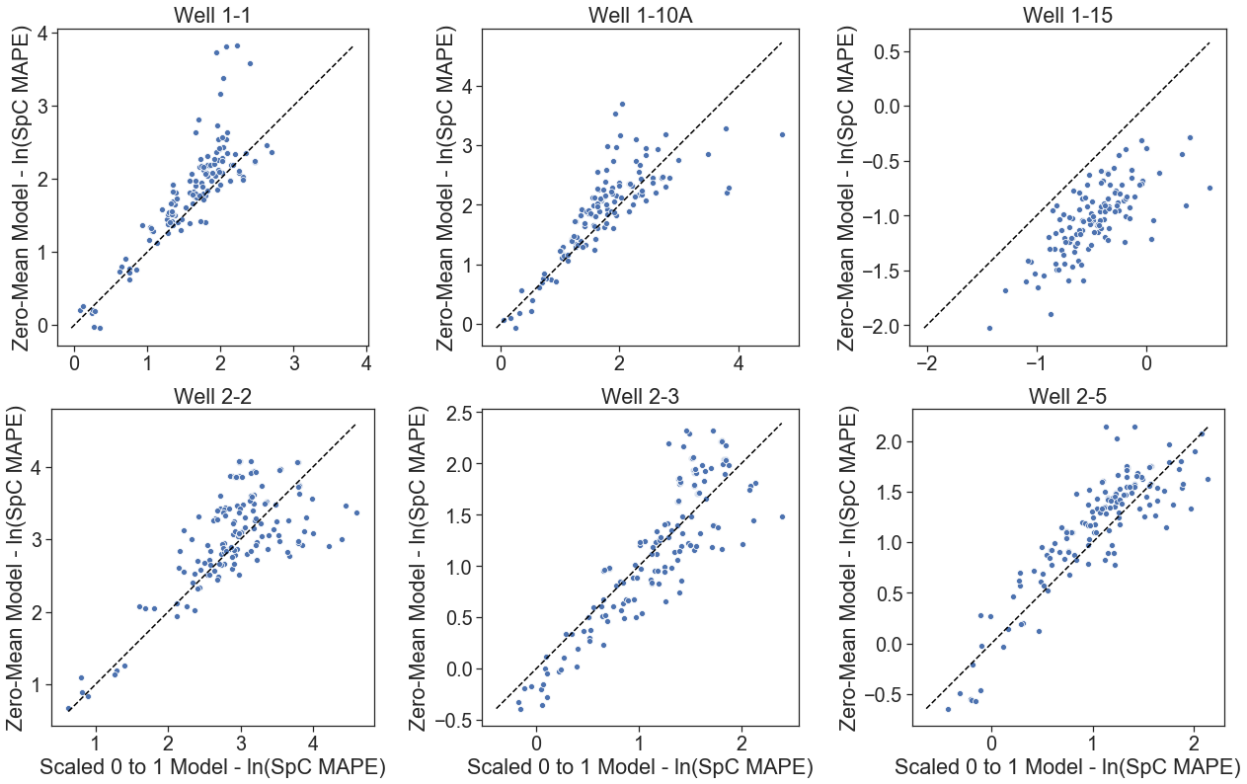

Figure 3.1: Comparison of models trained using scaling normalization on the data versus models trained using zero-mean, unit-variance normalization per well. Each data point represents the MAPE of a unique model configuration (size of input window, size of output window) tested on a gap length (1 hour, 24 hours, 48 hours, or 72 hours) for 2011 data. The x-axis of each plot is the natural log SpC MAPE of a unique model configuration trained on data normalized with scaling between 0 to 1. The y-axis is the natural log of the SpC MAPE of the same model configuration trained on data normalized via zero-mean. The dashed black line in each plot is the line $y = x$.

Similar to the reviewer's comments for 3.1a, we have also performed more extensive experimentation on smaller models (single LSTM layer with single dense + dropout layer) that only predicts SpC using the same model configurations as the original model (input window, output window, years of data), but limited to data from well 1-1. We compare the models against each other via gap filling on data from 2011. As seen in figure 3.2 of this response, the single LSTM performs better when filling in gaps of 1 hour for both normalization methods. However, our model architecture performs better in comparison for filling in gaps of 24, 48, and 72 hours. As such, the original model architecture is more robust in filling in larger gaps.

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 have added a sentence to the beginning to

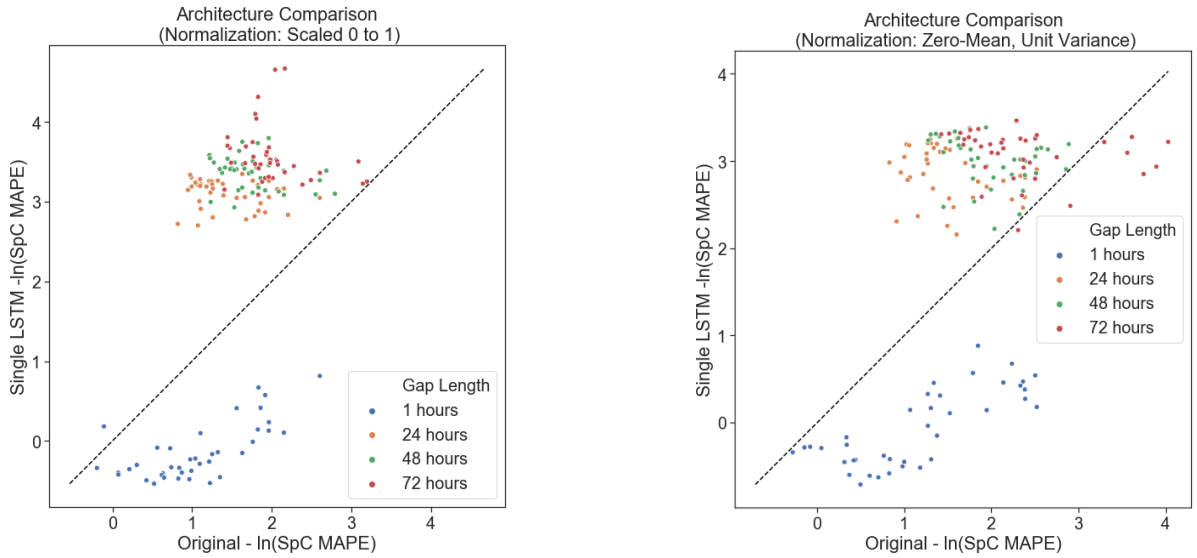

Figure 3.2: Comparison of models trained using the original model architecture versus a single LSTM layer with a single dense layer architecture. The left plot is for models trained on data normalized by scaling between 0 and 1. The right plot is for models trained on data normalized by zero-mean. Each data point represents the MAPE of a unique model configuration (size of input window, size of output window) tested on a gap length (1 hour, 24 hours, 48 hours, or 72 hours) for 2011 data. The x-axis of each plot is the natural log SpC MAPE of a unique model configuration using the original model architecture. The y-axis is the natural log of the SpC MAPE of the same model configuration using the single LSTM architecture. The dashed black line in each plot is the 1:1 line.

further clarify our training, valuation and testing setup. Also, table 1 had been updated to clarify the three independent time periods.

New description at P8 L10-L8: "After evaluating the gain in performance improvement by using increasingly more training data (details provided in the online supplemental materials), we concluded that 4 years of training data (2012-2015) was sufficient for all the models. Validation datasets were used to select the best model hyperparameters (section 3.1.1) and the optimal combination of M and N (section 3.1.2) for gap filling at each well. Another independent period was selected at each well, depending on data availability, to compare the gap filling performance using the DNN and ARIMA methods. The complete set of alternatives we considered for each DNN model configuration is shown in Table 1. Excluding combinations with $M < N$, 1080 unique models (180 models per well) were trained. We used an Adam optimizer (Kingma and Ba, 2014) for training and the mean-squared error as the loss function. The models were trained for 30 iterations (i.e., epochs) over the training data."

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 removed this evaluation in order to make the paper more straight forward and less confusing in its comparison of our DNN model and ARIMA. Furthermore, we have updated Figure 6 without the additional analysis and removed 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 DNN 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, as in the subsection 3.1.1.

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. The information carried by the ARIMA model is not enough to train other well. Also according to the comment 3e, we have removed 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 data period from 2011 (now labeled as the validation period in our paper). Based on the reviewer's suggestion, we have added a third independent time period (i.e. testing period) used to compare our DNN models to the ARIMA method. So now, we have three datasets: training, validation, and testing. The training period is used to fit the model (data from 2012-2015). The validation period (year 2011 for all wells) was used to determine the optimal model configuration. The testing period is the year 2016 for comparing the model performance against the ARIMA method for all wells except 2-5 and 1-15. The testing time periods for 2-5 and 1-15 are year 2008 and 2017 respectively because there was lack of SpC observations during 2016. Table 1 has been updated to clarify training, validation and testing period.

New description at P8 L12-L15: "Validation datasets were used to select the best model hyperparameters (section 3.1.1) and the optimal combination of M and N (section 3.1.2) for gap filling at each well. Another independent testing period was selected at each well, depending on data availability, to compare the gap filling performance using the DNN and ARIMA methods."

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: We performed other similar analyses for groundwater table and temperature, but they are not shown here because SpC is our primary interest for this study and also for space consideration. The reviewer is right that we don't have to predict all three variables. We experimented with a simpler architecture to predict SpC only. Those results are shown in response to comment 3.3c, which showed the simpler architecture performs better on smaller gaps, but our DNN model outperforms on larger hour gaps.

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 in Figure 6 by limiting the predicting output window sizes to 1h, 6h, 12h, and 24h for all input window sizes to provide a fair comparison. In comparing MAPEs across various input window sizes shown in Figure 6 (b), we observe that models with all input windows have comparable median MAPEs, with those of 24, 72, 144 and 168 hours leading to slightly smaller median MAPEs. We are now including all the models correspond to an input window size in boxplots rather than just showing a mean value as in the original manuscript. The input window size of 24 hours led to robust performance in terms of the fraction of small MAPEs, median and outliers on the large MAPE end. As the reviewer noted, the averaged MAPE for the model with input length of 168h, 7.33, is similar to that of the 24h input length models (6.39). In addition, we removed the original figure 6(a) for simplification and moved that analysis to the supplemental material, only showing results with 4 years of data. The updated figure and caption is shown in figure 3.3.

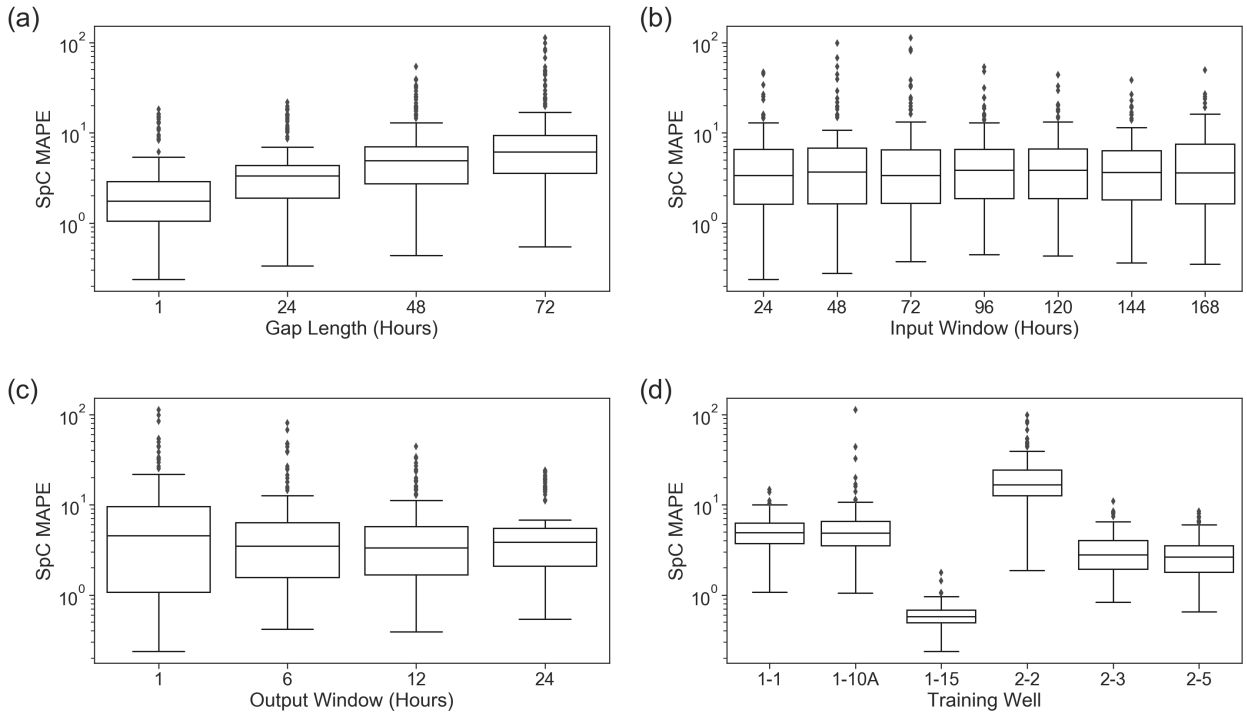

Figure 3.3: Updated Figure 6: Gap filling performance for SpC evaluated against the validation datasets under multiple model configuration parameters (a-c) or grouped by training wells (d). (a) distribution of SpC MAPE vs. tested gap lengths; (b) distribution of SpC MAPE vs. model input window size M; (c) distribution of SpC MAPE vs. model output window size N; (d) distribution of SpC MAPE aggregated by wells

New sentence P13 L12-L21: As shown in Figure 6(a), model performance deteriorates as the gap length increases, indicating that the DNN-based method tends to lose ground truth information from its input to inform prediction. In comparing MAPEs across various input window sizes shown in Figure 6 (b), we observe that models with all input windows have comparable median MAPEs, with those of 24, 72, 144 and 168 hours leading to slightly smaller median MAPEs. The 144- and 168-hour input windows

also yield lower third quartile of MAPE and fewer outliers on the larger MAPE end, indicating that the memory units in the LSTM layers are capturing important daily to weekly signatures (evident in WPS plots in Figure 2 for all wells except for Well 1-15) for some wells. As shown in 6 (c), daily and subdaily output windows yield comparable median MAPEs, with the 24-hour output window leading to smaller third quartile and fewer large MAPE outliers than its 1-, 6-, and 12-hour counterparts. Overall, an input window of 144 hours and an output windows of 24 hours appear to be a robust model configuration for all wells and gap lengths considered.

Minor Comments:

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 have trained multi-well DNN models by including multiple neighboring wells to address both spatial and temporal components in our study. The additional discussion related to this effort has been added to new subsection 4.3. Please also refer to our response to comment 3.3c.

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 have removed the citation of Olah (2015) and updated 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 have updated 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 have added a brief description of LSTMs applied to earth science and hydrology (this is the same update in response to reviewer comment 2.10)

New section (P3 L21-24): There have been applications of RNNs and LSTMs emerging in hydrology. For example, Kratzert et al. (2018) used LSTMs to predict watershed runoff from meteorological observations, Zhang et al. (2018) used LSTMs for predicting sewer overflow events from rainfall intensity and sewer water level measurements, and 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 has been modified in the revised 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: We have added the following sentences to clarify:

New sentences (P5 L13-L18):"A larger coherence at a given frequency indicates a stronger correlation at that frequency between the SpC at a well and the river stage. We consider these two variables highly correlated when the coherence is larger than 0.7 (shown in green to red colors in Coherence plots). We found that such high correlations exist at multiple frequencies, from subdaily to daily to yearly, at all the wells close to the river (e.g., 1-1, 1-10A, 2-2, and 2-3), while the higher correlation regimes in wells farther from the river (e.g., 1-15 and 2-5) are shifted towards longer periods at semi-annual and annual frequencies and less persistent in time."

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: We didn't include the comparison for gap length of 1 hour because both methods were highly accurate in filling such small data gaps. We added this explanation on P14 L1-2 to be consistent.

Reviewer Comment 3.14 — P5 L23 delete "clearly"

Response: Agreed. It has been 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: We have updated the sentence to correctly describe the model.

New sentence at P9 L3 to L4: The final DNN architecture, as shown in Figure 4, contains three LSTM layers, followed by two dense layers 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: We have removed the sentence.

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

Response: Thanks for the catch. It has been 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: We have been updated the sentence to say "A linear layer with a sigmoid activation function" in the manuscript.

New sentence (P11-L6): 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: We have updated the sentence to distinguish the multiplication from the gate. See the previous 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: We have updated 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: The second sentence in the instruction of output gate has been removed in the revised 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: We were not able to complete the training for all cases correspond to the output window of 96 hours within allocated computing hours. In the analyses with other output window sizes, we found the performance kept deteriorating when the output window size exceeded 24 hours. We also constrained all the analyses in the revised manuscript to not exceed 24 hours so that we have the same pool of models to compare across other parameters, see our response to comment 3.6. Therefore, our conclusions are not impacted by missing the 96hr output window, and we didn't take extra computational resources to finish training those cases.

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

Response: Equation 2 only contains non-seasonal terms (p, d, q), i.e., autoregressive terms, nonseasonal differences and moving-average terms. The model with seasonal terms are much more complicated, so we added the equations with seasonal terms in online supplemental material with a note added to the revised manuscript on P12 L5.

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

Response: The statement has been removed in revised manuscript.

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

Response: Yes. It has been 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 have redone the LSTM analysis to only include the test results for the models on the same well it was trained for, which is consistent 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: We have updated the figure caption to explicitly state the results are for SpC only

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

Response: We have updated the section to explicitly state that 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 have updated the sentence to limit the scope to our experimental runs.
New sentence (P13 L30 - P14 L1): We observe that DNN models require less or equal input information than that required by the ARIMA method for the wells tested.

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 revised the description to show a better linkage between the time when larger relative errors occur and the time window when higher-frequency variations show more energy in WPS plots. Please refer to P16 L17-27: "the time windows of high relative errors are found to approximately co-locate with the time when high-frequency (daily and subdaily) signals are gaining more power. The difference between the DNN and ARIMA models tend to be amplified during those time windows. Wells 1-1, 1-10A, and 2-2 share similar seasonal patterns in WPS, with the highest intensity bin above 1024 hours across February to July. Their average WPSs all show peaks around daily and subdaily frequencies. Well 2-3 has its greatest energy between 16 to 256 hours from January to July. Well 2-5 has low intensities of variability at daily and subdaily frequencies with the low-frequency variations (monthly and seasonal) dominating the Jan to March time frame. For well 1-15, one of its strongest intensities is above 2048 hours across the entire year, and the other strong intensities are narrow bands between

16 to 256 hours. In general, both DNN and ARIMA are effective at capturing low-frequency variability (monthly and seasonal). Although DNN is more effective at capturing high-frequency (daily and subdaily) fluctuations and nonlinearities in the datasets, it may also lead to overly dynamic predictions when the training data contain more significant high-frequency signatures than the system behavior to be predicted."

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:
Thank you for catching the mistake. The revised figure (Figure 9 in the revised manuscript) now shows the ARIMA models with the same parameters (input window size, output window size) that were used in the original Figure 7 (Figure 8 in the revised manuscript). These optimal ARIMA parameters are shown in the new Figure 7 to provide more information.

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: Please refer to our response to comment 3.30. Columns 3 and 4 were meant to show when the higher-frequency variations are gaining more power during the testing time period, which were found to colocate with the time window larger relative errors occur. Hope the more detailed explanation provided in the revise text P16 L17-27 helps with understanding. We haven't found a better way to quantitatively relate magnitude of error and the composition of flow variations.

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: Please refer to our responses to comments 3.30 and 3.32. The purpose of the original Figure 8 (Figure 9 in revised manuscript) is to show LSTM-based DNN was able to better match the observed behavior during time windows when high-frequency fluctuations are significant. Hope the revise text explains that better.

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: Agreed. We have updated our conclusions section based on the additional analysis we have done, including the multi-well model, zero-mean normalization and different seeding experiments. As previously stated, we have explicitly mentioned we are analyzing gap filling results for the SpC only.

References

Du, X., Zhang, H., Nguyen, H. V., and Han, Z.: Stacked LSTM Deep Learning Model for Traffic Prediction in Vehicle-to-Vehicle Communication, in: 2017 IEEE 86th Vehicular Technology Conference (VTC-Fall), pp. 1–5, https://doi.org/10.1109/VTCFall.2017.8288312, 2017.

Fang, K., Shen, C., Kifer, D., and Yang, X.: Prolongation of SMAP to Spatiotemporally Seamless Coverage of Continental U.S. Using a Deep Learning Neural Network, Geophysical Research Letters, 44, 11,030–11,039, https://doi.org/10.1002/2017GL075619, URL https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2017GL075619, 2017.

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

Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, Neural Comput., 9, 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735, URL https://doi.org/10.1162/neco.1997.9.8.1735, 1997.

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

Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, CoRR, abs/1412.6980, URL http://arxiv.org/abs/1412.6980, 2014.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrology and Earth System Sciences, 22, 6005–6022, https://doi.org/10.5194/hess-22-6005-2018, URL https://www.hydrol-earth-syst-sci.net/22/6005/2018/, 2018.

Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y.: How to Construct Deep Recurrent Neural Networks, 2013.

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

Schuster, M. and Paliwal, K. K.: Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing, 45, 2673–2681, 1997.

Zhang, D., Lindholm, G., and Ratnaweera, H.: Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring, Journal of Hydrology, 556, 409 – 418, https://doi.org/https://doi.org/10.1016/j.jhydrol.2017.11.018, URL http://www.sciencedirect.com/science/article/pii/S0022169417307722, 2018.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X.: Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks, URL https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11989/12149, 2016.

Technical note: Using Deep ~~Learning~~ Neural Network Models to Fill Spatio-Temporal Data Gaps in Hydrological Monitoring Networks

Huiying Ren[1], Erol Cromwell[2], Ben Kravitz[3,4], and Xingyuan Chen[4]

[1]Earth Systems Science Division, Pacific Northwest National Laboratory, WA, USA
[2]Advanced Computing, Mathematics, and Data Division, Pacific Northwest National Laboratory, WA, USA
[3]Department of Earth and Atmospheric Sciences, Indiana University, Bloomington, IN, USA
[4]Atmospheric Sciences and Global Change Division, Pacific Northwest National Laboratory, WA, USA

Correspondence: Xingyuan Chen (Xingyuan.Chen@pnnl.gov)

Abstract. ~~Long-term~~ The spatio-temporal ~~changes~~ dynamics over a long time window in subsurface hydrological flow are usually quantified through a network of wells; however, such observations often are spatially sparse and temporal gaps exist due to poor quality or instrument failure. In this study, we explore the ability of deep neural networks to fill in gaps in spatially distributed time-series data, especially for the datasets with high-frequency dynamics. We selected a location at the U.S. Department of Energy's Hanford site to demonstrate and evaluate the new method, using a 10-year spatio-temporal hydrological dataset of temperature, specific conductance, and groundwater table elevation from 42 wells that monitor the dynamic and heterogeneous hydrologic exchanges between the Columbia River and its adjacent groundwater aquifer. We employ a deep nueral network (DNN) architecture that contains stacked long short-term memory (LSTM)~~-based architecture, which is specially designed~~ , convolutional and dense layers to address both the spatial and temporal variations in the property fields.

The performance of ~~gap filling using an LSTM framework is evaluated using test datasets with synthetic data gaps created by assuming the observations were missing for a given time window (i.e., gap length), such that the mean absolute percentage error can be calculated against true observations. Such test datasets also allow us to examine how well the original nonlinear dynamics are captured in gap-filled time series beyond the error statistics. The performance of the LSTM-based gap-filling method is compared to that of a traditional , popular gap-filling method:~~ the DNN-based gap filling method was evaluated against a traditional autoregressive integrated moving average (ARIMA) ~~. Although ARIMA appears to perform slightly better than LSTM on average~~ method in terms of both error statistics and capturing nonlinear, dynamic patterns in wells that exhibit various dynamics signatures. Although the ARIMA models yield better error statistics, ~~LSTM is better able to capture nonlinear~~ they fail to capture abrupt changes or high-frequency (daily and subdaily) variations in system states that are typical characteristics of a complex dynamic system. The DNN-based models excel in capturing both high-frequency and low-frequency (monthly and seasonal) dynamics that are present in time series ~~. Thus, LSTMs show promising potential to outperform ARIMA~~ at all wells, although they may also lead to overly dynamic predictions as guided by the training data. The DNN is shown to improve the predictive ability by taking advantage of the spatial information from neighboring wells under highly challenging situations, such as multiple days of gaps in system states that vary at subdaily scales. The DNN-based models afford the great advantage of accounting for spatial and temporal correlations and nonlinearity in data without apriori assumptions. Although DNN models require substantial training data and computational resources and have

limited extrapolation power beyond the conditions represented in the training data, they showed promising potential for gap filling in highly dynamic time-series observations characterized by multiple dominant modes of variability. Capturing such dynamics is essential to generate the most valuable observations to advance our understanding of dynamic complex systems.

1 Introduction

Long-term hydrological monitoring using distributed well networks is of critical importance for many areas, including understanding how ecosystems respond to chronic or extreme perturbations, as well as informing policies and decisions related to natural resources and environmental issues (Wett et al., 2002; Taylor and Alley, 2002; Grant and Dietrich, 2017). One of the most common methods of collecting hydrological data in groundwater is through wells (Güler and Thyne, 2004; Strobl and Robillard, 2008; Lin et al., 2012); however, wells are necessarily sparse, leaving spatial gaps in the dataset. Moreover, most well data will also have temporal gaps due to instrument failure or poor quality of measurements for numerous reasons. These data gaps degrade the quality of the dataset and increase the uncertainty in the spatial and temporal patterns that are derived from them. Gap filling is necessary for developing understanding of the underlying system as well as for use in creating continuous, internally consistent boundary conditions for numerical models. Many natural systems exhibit nonlinear or/and ~~nonstationary~~ non-stationary behaviors due to evolving nonlinear dynamics, which makes it challenging to reproduce those complex patterns while filling in data gaps.

Various statistical methods have been developed to fill gaps in spatio-temporal datasets, with the most commonly used being the autoregressive integrated moving average (ARIMA) method (Han et al., 2010; Zhang, 2003). For any given spatial location, ARIMA uses temporal autocorrelation to predict unobserved data points in a time series. Spatio-temporal autocorrelations can be considered by using multivariate ARIMA and space-time autoregressive models (Kamarianakis and Prastacos, 2003; Wikle et al., 1998; Kamarianakis and Prastacos, 2005); however, ARIMA cannot capture nonlinear trends because it assumes a linear dependence between adjacent observations (Faruk, 2010; Valenzuela et al., 2008; Ho et al., 2002). In addition, all existing space-time ARIMA models assume fixed global autoregressive and moving average terms, which would fail to capture evolving dynamics in highly dynamic systems (Pfeifer and Deutrch, 1980; Griffith, 2010; Cheng et al., 2012, 2014). Spectral-based methods, such as singular spectrum analysis, maximum entropy method, and Lomb-Scargle periodogram, have been used to account for nonlinear trends while filling in gaps in spatio-temporal datasets (Ghil et al., 2002; Hocke and Kämpfer, 2008; Kondrashov and Ghil, 2006). However, these methods use a few optimal spatial or temporal modes occurring at low frequencies to predict the missing values, with the other higher frequency components discarded as noise, which may lead to reduced accuracy of the statistical models in fitting the observations and in predicting missing values (Kondrashov et al., 2010; Wang et al., 2012). Kriging and maximum likelihood estimation used in spatial and spatio-temporal gap filling often face computational challenges as they require computing the covariance matrix of the data vector, which can be quite large

(Katzfuss and Cressie, 2012; Eidsvik et al., 2014). Other nonlinear methods have been explored with some success, including expectation-maximization or Bayesian probabilistic inference including hierarchical models, Gaussian process, and Markov chain Monte Carlo; the spatial and temporal correlations are most effectively captured by using models that build dependencies in different stages or hierarchies (Calculli et al., 2015; Banerjee et al., 2014; Datta et al., 2016; Finley et al., 2013; Stroud et al., 2017). In general, the expectation-maximization algorithm and Bayesian-based methods are sensitive to the choice of initial values and prior distributions in parameter space (Katzfuss and Cressie, 2011, 2012). Moreover, the prior distributions with all the associated parameters in both the spatial and temporal domains need to be specified, which becomes increasingly difficult in more complex systems. Empirical Orthogonal Functions (EOF) related interpolation methods, such as least squares EOF (LSEOF), data interpolation EOF (DINEOF), and recursively subtracted EOF (REEOF), are widely used to fill in missing data from geophysical fields such as clouds in sea surface temperature datasets or other satellite-based images with regular gridded domains (Beckers and Rixen, 2003; Beckers et al., 2006; Alvera-Azcárate et al., 2016). However, the requirement of gridded data by the EOF methods limits their use in filling data gaps in irregularly spaced monitoring networks.

Deep neural networks (DNNs) (Schmidhuber, 2015) are data-driven tools that, in principle, could provide a powerful way of extracting the nonlinear spatio-temporal patterns hidden in the distributed time-series data without knowing their explicit forms (Längkvist et al., 2014). They are increasingly been used in geoscience domains to extract patterns and insights from the streams of geospatial data and to transform the understanding of complex systems (Reichstein et al., 2019; Shen, 2018; Sun, 2018; Sun et al., 2019; Gentine et al., 2018). The umbrella term of DNN contains numerous categories of architectures, depending on the problem at hand. For the analyses in this paper, which are focused on filling gaps in time-series data, a natural choice of architecture is recurrent neural networks (RNNs) ~~(Connor et al., 1994; Olah, 2015)~~(JORDAN, 1986). These networks take sequences (e.g., time series) as input and output single values or sequences that follow. They are designed to use information about previous events to make predictions about future events, essentially by letting the model "remember." However, for longer sequences of data, RNNs have been shown to lose memory from previously trained data, i.e., they "forget" (Hochreiter et al., 2001). This affects the performance of RNNs, particularly for data where the beginning of a sequence impacts the prediction, since this information becomes exponentially less impactful for the prediction as the size of the sequence increases. Long short-term memory (LSTM) networks are variations of RNNs that are explicitly designed to avoid this problem by using memory cells to retain information about relevant past events ~~(Ma et al., 2015)~~(Hochreiter and Schmidhuber, 1997). RNNs and LSTMs have been successfully applied to text prediction (Graves, 2013), text translation (Wu et al., 2016), speech recognition (Graves et al., 2013), and image captioning (You et al., 2016) ~~.~~(Specifics on LSTM architectures are described in Section 3.1.1~~.)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).~~). There have been applications of RNNs and LSTMs emerging in hydrology. For example, Kratzert et al. (2018) used LSTMs to predict watershed runoff from meteorological observations, Zhang et al. (2018) used LSTMs for predicting sewer overflow events from rainfall intensity and sewer water level measurements, and Fang et al. (2017) used LSTMs to predict soil moisture with high fidelity. Compared to a single RNN/LSTM layer, more complex LSTM architectures such as stacked and bidirectional LSTMs, CNN-LSTM or convolutional LSTM have the potential to capture extra features (Graves et al., 2013; Pascanu et al., 2013) as shown in various applications,

including action recognition (Zhu et al., 2016) and vulnerable road users location predictions (Saleh et al., 2017). A bidirectional RNN/LSTM works by duplicating the recurrent network into two networks: one responsible for fitting the positive time direction (i.e. the forward states) and the other responsible for the negative time direction (i.e the backwards state) (Schuster and Paliwal, 19 . In general, the input sequence is fed as-is to the forward state and a reversed copy of the input sequence is fed to the backwards state. The bidirectional LSTM can be used in history matching problems.

~~This~~

Our study aims to evaluate the potential of using LSTM layers within a DNN architecture ~~that utilizes LSTMs for filling gaps in a~~ to fill gaps in spatio-temporal environmental ~~dataset,~~ time series. We treat the gap filling as a forecasting problem, i.e., we use the historical data as input to predict the missing values in the data gaps. We demonstrate our method using a test case that focuses on understanding the interactions between a regulated river and contaminated groundwater aquifer. ~~The use of a DNN~~ We adopt the stacked-LSTM combined with the convolutional layer as our DNN model to understand the interactions between a regulated river and contaminated groundwater aquifer. The DNN-based gap filling method is compared with traditional time series approaches (e.g., ARIMA) to identify situations in which ~~a DNN outperforms more commonly used methods~~ DNNs outperform ARIMA as well as what the optimal configurations might be for this particular application.

2 Study Site and Data Description

A 10-year (2008–2018) ~~hourly~~ spatio-temporal dataset was collected from a network of groundwater wells that monitor temperature (Water conductivity and temperature probe CS547A by Campbell Scientific), specific conductance (SpC) (Water conductivity and temperature probeCS547A by Campbell Scientific), and water-table elevation (stainless-steel pressure transducer CS451 by the Campbell Scientific) at the 300 Area of the U.S. Department of Energy Hanford site, located in southeastern Washington State. The groundwater well network was originally built to monitor the attenuation of legacy contaminants. The ~~time series of river stage from 2008 to 2018 (Figure 1) shows large and dynamic fluctuations in river stage. These fluctuations are due not only to natural processes but also to regulation of the river by the upstream hydroelectric dam operations (Song et al., 2018), which on average vary ~0.5 m diurnally and up to ~2-3 m annually. The water elevation dynamics in each groundwater well is driven by river stage fluctuations, which in turn influence contaminant recharge to groundwater and lead to highly complex transport behaviors of the contaminants at the site (Arntzen et al., 2006; Zachara et al., 2016)~~groundwater aquifer at our study site is composed of two distinct geologic formations: a highly permeable formation (Hanford formation, consisting of coarse gravelly sand and sandy gravel) underlain by a much less permeable formation (the Ringold Formation, consisting of silt and fine sand). The dominant hydrogeologic features of the aquifer are defined by the interface between the Hanford and Ringold formations and the heterogeneity within the Hanford formation (Chen et al., 2012a, 2013a).

The intrusion of river water into the adjacent groundwater aquifer causes mixing of two water bodies with distinct geochemistry and stimulates biogeochemical reactions at the interface. The river water has lower SpC (0.1–0.2 mS/cm) than groundwater (averaging $\sim$ 0.4 mS/cm). Groundwater has a nearly constant temperature (16–17$^{\circ}C$) as opposed to seasonally varying river temperature (3–22$^{\circ}C$). The highly heterogeneous coarse-textured aquifer (Zachara et al., 2013) interacts with dy-

namic river stages to create complex river intrusion and retreat pathways and dynamics~~(Chen et al., 2013b, 2012b)~~. The time series of multi-year SpC and temperature observations at the ~~select~~ selected set of wells in the network have demonstrated these complicated processes of river water intrusion into our study site (Figure 1). ~~For wells that are farther inland (e.g., well 1-15), temperatures remain consistently within the groundwater temperature range and SpC has three noticeable dips (dropping from 0.5 to 0.4 mS/cm range), coinciding with the high river stages in years 2011, 2012, and 2017, which are featured with higher peak river stages than other years so the river water could intrude further into the groundwater aquifer. Wells close to the~~ Wells near the river shoreline (e.g., wells 1-1, 1-10A, 2-2, and 2-3) tend to be strongly affected by river water intrusion in spring and summer. As such, the dynamic patterns of SpC and temperature correspond well with river stage fluctuations, specifically that SpC decreases and temperature increases with increasing river stage. Fluctuations of SpC in well 2-2 appear to be stronger and at higher frequency than in other wells, likely indicating its higher connectivity with the river. For wells that are farther inland (e.g., well 1-15), on the other hand, temperatures remain consistently within the groundwater temperature range and SpC has three noticeable dips (dropping from 0.5 to 0.4 mS/cm range), coinciding with the high river stages in years 2011, 2012, and 2017, which are featured with higher peak river stages than other years so the river water was able to intrude further into the groundwater aquifer. Well 2-5 is located at an intermediate distance from the river compared to other wells shown in Figure 1, so the intrusion of river water is evident in most of the years except in low-flow years such as 2009 and 2015, during which both SpC and temperature remain nearly unchanged.

The understanding we developed from earlier studies is that the physical heterogeneity contributes to the different response behaviors at different locations while the river stage dynamics lead to multi-frequency dynamics in those responses. The seasonal and annual variations are driven by natural climatic forcing (Amaranto et al., 2019, 2018), whereas the higher-frequency (i.e., daily and sub-daily) fluctuations are primarily induced by operations of the upstream hydroelectric dam operations to meet various demands of human society (Song et al., 2018). Our system is representative of many dam-regulated gravel-bed rivers across the world, where dam operations as a typical anthropologenic activity have significantly altered the hydrologic exchanges between river water and groundwater, as well as the associated thermal and biogeochemical processes (Song et al., 2018; Shuai et al., 2019; . Note that the multi-frequency variations in data are characterizing the dynamic features of data, which could exist in both short-term and long-term time series data as a result of short-term or long-term monitoring effort.

To understand the ~~dynamic behaviors~~ multi-frequency variations of all the variables in each well at the study site, we perform spectral analysis on multi-year SpC observations at each selected well using a ~~continuous wavelet transform . The continuous wavelet transform~~ discrete wavelet transform (DWT). The DWT is widely used for time–frequency analysis of time series and relies on a ~~"mother wavelet,"~~ "mother wavelet", which is chosen to be the Morlet wavelet (Grossmann and Morlet, 1984) to deal with the time-varying frequency and amplitude in time-series data at this site (Stockwell et al., 1996; Grinsted et al., 2004). ~~The~~ We illustrate the Wavelet Power Spectrum (WPS) ~~for the multiyear SpC time series and the~~ in log scale and its normalized global power spectrum (average WPS over the time domain~~for each well) are displayed~~) for the multi-year SpC time series in the first two columns ~~in~~ of Figure 2. Data gaps are ~~indicated~~ shown as blank regions in Figure 2; examples include early year 2009 at well 1-1, the beginning of year 2011 at well 1-10A, and the later part of year 2012 at well 2-2. The amplitude of WPS represents the relative importance of variation at a given frequency compared to the variations at other frequencies across

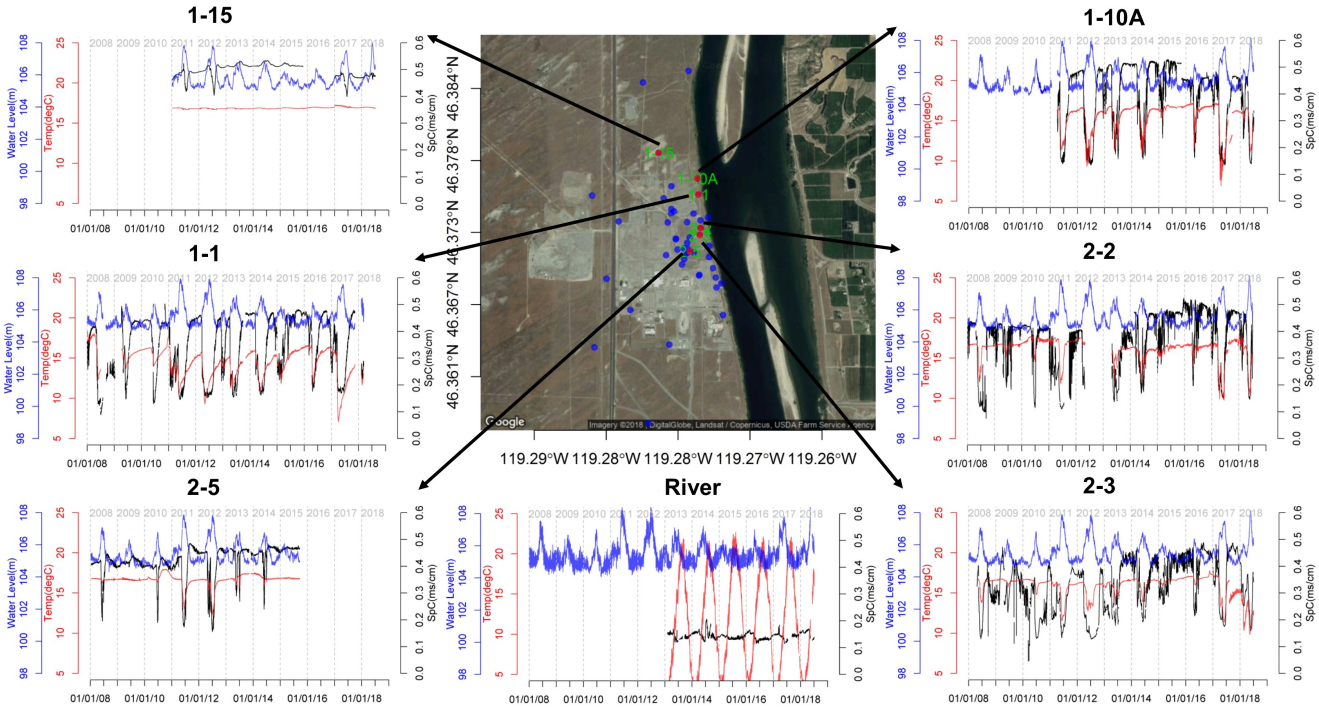

Figure 1. Groundwater monitoring well network at the 300 Area of the Hanford site and the monitoring data at select wells. Each well represented by a dot is instrumented to measure groundwater elevation, temperature, and SpC. The wells selected for this study are marked with red dots with well names. The three variables monitored in wells and in the river are shown in time-series plots with blue (water elevation), black (SpC), and red (temperature) lines.

the spectrum. At wells 1-1, 1-10A, 2-3, 2-5, and 2-2, the strong intensities of SpC signals appear at the half-year and yearly frequencies; however, well 1-15 has a different pattern in that most of its high intensities are below the 256-hour frequency. The averaged WPS more clearly shows the contrast in behaviors: wells 1-1, 1-10A, 2-3, and 2-5 have a dominant frequency at half a year; well 2-2 has multiple dominant frequencies at daily, monthly, and seasonal scales; while well 1-15 has similar intensities at half-year and hourly scales. Applying this information to the task at hand, we hypothesize that gap filling at well 2-2 could be more challenging due to the multiple sources of its dynamic behaviors, manifested as significant powers at multiple frequencies.

Since the dynamics of the system are driven by the river stage, we perform magnitude-squared wavelet coherence analysis via the Morlet wavelet to reveal dynamic correlations between the SpC and river stage time series (Grinsted et al., 2004; Vacha and Barunik, 2012). Wavelet coherence in the time-frequency domain is plotted in the third column in Figure 2 and the average coherence is plotted in the fourth column; statistically significant values at the 95th percent confidence interval are indicated with red points. ~~Red regions in the third column indicate that the two signals are highly correlated, while blue regions indicate lower correlations. SpC correlates well with river stagein every well at multiple time scales, with the exception of well 1-15.~~

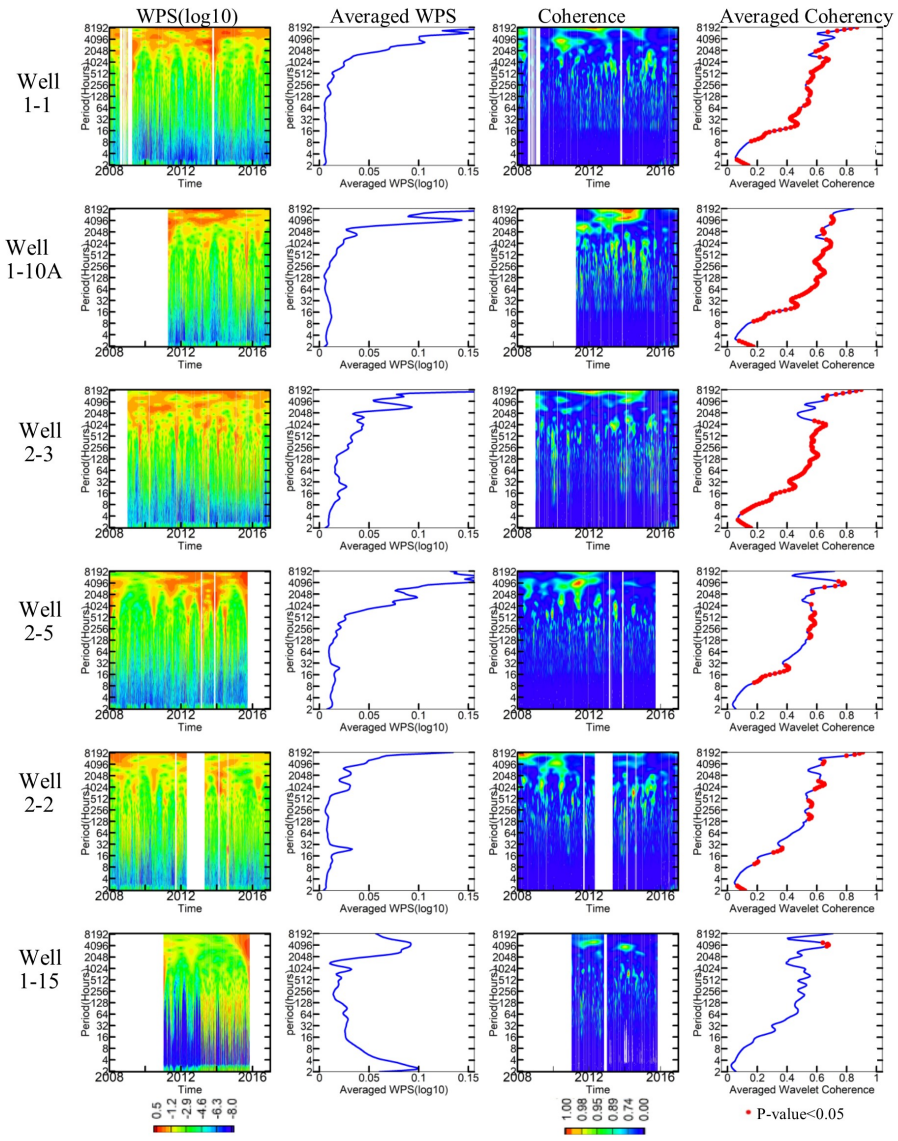

Figure 2. WPS analysis of SpC at each well from 2008 to 2018. The first column is the spectrogram (in log10 scale) of SpC in each well; the second column is the averaged WPS; the third column is the coherence between SpC in each well and the river stage; and the fourth column is the averaged coherence with $p < 0.05$ values indicated in red.

~~High correlations are found in~~ A larger coherence at a given frequency indicates a stronger correlation at that frequency between the SpC at a well and the river stage. We consider these two variables highly correlated when the coherence is larger than 0.7 (shown in green to red colors in Coherence plots). We found that such high correlations exist at multiple frequencies, from subdaily to daily to yearly, at all the wells close to the river (e.g., 1-1, 1-10A, 2-2, and 2-3)~~at half-year and yearly frequencies.~~

~~High correlations~~ , while the higher correlation regimes in wells farther from the river (e.g., 1-15 and 2-5) are shifted towards longer periods ~~and~~ at semi-annual and annual frequencies and less persistent in time.

~~WPS analysis of SpC at each well. The first column is the spectrogram of SpC in each well; the second column is the averaged WPS; the third column is the coherence between SpC in each well and the river stage; and the fourth column is the averaged coherence with $p < 0.05$ values indicated in red.~~

As can be ~~clearly~~ seen in Figure 2, many of the wells have long data gaps, which have unknown effects on our ability to estimate dynamics from the wavelet spectra. As such, gap filling is needed to infer observations and guide modeling of the underlying system. Figure 3 provides a summary of gap lengths for the overall network of monitoring wells. The majority of the gap lengths of all the three monitored variables are less than 50 hours. Therefore, in our investigations we explore the ability of the methods in filling gaps ~~at~~ of 24-, 48-, and 72-hour lengths using hourly data to capture the high-frequency fluctuations.

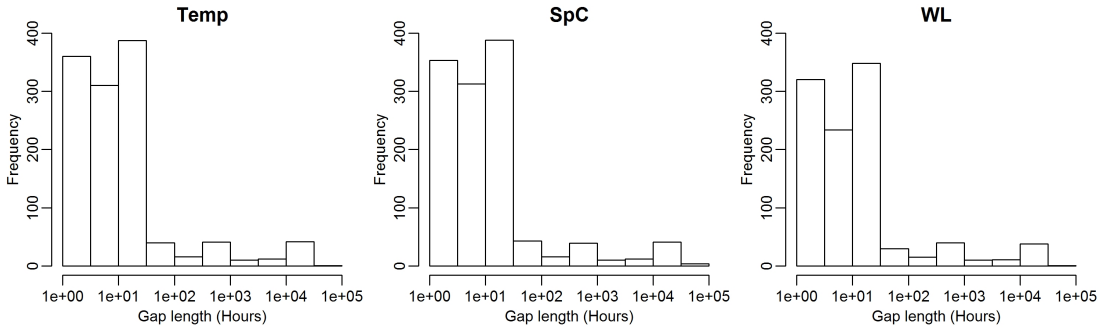

Figure 3. Histograms of gap lengths for each monitored variable, aggregated across all wells in the monitoring network during 2008-2018.

3 Gap-Filling Methods

In this section, we describe two methods we ~~use~~ implemented to fill gaps of various lengths in SpC measurements at ~~select~~ selected wells: a DNN model using several LSTM network layers and the traditional ARIMA model for comparison and assessing the strengths and limitations of the ~~LSTM-based~~ DNN model. In both ~~models~~methods, an input with M ~~data points~~ time steps (input window length) is provided to predict outputs of N time steps ~~that follow the input window~~ (output window length) ~~.~~ that follow the input window. The performance of both methods were evaluated using testing datasets for the selected wells.

3.1 ~~Stacked LSTM ArchitectureWe designed the~~ DNN Models for Gap Filling

We designed a DNN architecture to train models of ~~input size~~ an input size of M ~~and output size~~ time steps and an output size of N time steps to fill gaps of various lengths in groundwater well measurements. 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. This model architecture is generally described as a stacked LSTM model, given that the LSTM layers are "stacked" on~~

~~top of each other. Each~~ input and predicted output at a time step contain the following three ~~well measurements~~ measurements from a single well or multiple wells: water level (m), temperature ($^{\circ}C$), and SpC (mS/cm), leaving the model to generalize nonlinear connections among them. ~~Stacked LSTM layers take advantage of the temporal correlations of the measurements to improve model performance~~Assuming the observations from W ($W \geq 1$) wells are used to fill in data gaps, the input size of the model is then $M \times 3W$. Similarly, the model output can be those three variables in the next N time steps for one or more wells. Using multiple wells as input adds a spatial component to the model allows the DNN model to account for both the temporal and spatial correlations in the data to improve gap-filling performance. Wells were selected based on adequate data availability and their distances from the river. While the DNN model can be used to fill in gaps in all three variables, we focused our analyses on filling gaps in SpC because of its importance to reveal river water and groundwater mixing. Same set of analyses can be performed on water level and temperature.

We explored different DNN model architectures that contain a single or multiple LSTM layers for each desired combination of M and N at each well under various lengths of gaps with different amount of training data (2, 4 and 6 years). Training data for the DNN models were created by finding data segments of $M + N$ hours that have no missing values, i.e., no gaps in the data, for all three measurements over a specified monitoring window. The well data were then preprocessed by normalizing all measurements to fall between 0 and 1 using different scaling factors for each variable, as temperature measurements are on a scale of 10^1, SpC is on a scale of 10^{-1}, and water level is on a scale of 10^2. After evaluating the gain in performance improvement by using increasingly more training data (details provided in the online supplemental materials), we concluded that 4 years of training data (2012-2015) was sufficient for all the models. Validation datasets were used to select the best model hyperparameters (3.1.1) and the optimal combination of M and N (3.1.1) for gap filling at each well. Another independent testing period was selected at each well, depending on data availability, to compare the gap filling performance using the DNN and ARIMA methods. The complete set of alternatives we considered for each DNN model configuration is shown in Table 1. Excluding combinations with $M < N$, 1080 unique models (180 models per well) were trained. We used an Adam optimizer (Kingma and Ba, 2014) for training and the mean-squared error as the loss function. The models were trained for 30 iterations (i.e., epochs) over the training data. Each model configuration was trained using four different initialization seeds and error metrics were averaged to determine the best configuration.

In addition to the DNN models trained for the single-well setup, we also trained multi-well models that used observations from wells 1-1, 1-10A, and 1-16A to fill in data gaps for well 1-1. We explored the same set of configuration parameters shown in Table 1 for single-well models in multi-well models. We then compared the gap filling performance of the multi-well DNN with the single-well DNN model for well 1-1. The multi-well models were not explored for the other wells due to lack of neighboring wells in close proximity.

To evaluate the accuracy of the trained DNN models in filling SpC data gaps during the validation and testing processes, we assumed that synthetic gaps of various lengths (e.g., 1, 24, 48, and 72 hours, referred to as gap scenarios hereafter) exist in the validation or testing dataset of a well. Then a DNN model configured with input and output windows of M and N on a single or multiple wells is given the first M hours of data from the time series preceding the occurrence of a gap (assuming no missing values in these M hours) to fill in the first missing value in the gap by taking the first value of the predicted N

hours. This gap-filled value is then treated as if it was observed when repeating this procedure to fill in the gap of the next hour. This sliding window moves forward hour by hour until the entire gap of the data is filled. The accuracy of the gap-filling model is evaluated by calculating the mean absolute percentage error (MAPE; %) between the SpC values that are filled in (i.e., predicted) and that were observed:

$$MAPE = 100 \times \frac{1}{n} \sum_{t=1}^{n} \left| \frac{\text{Prediction} - \text{Observation}}{\text{Observation}} \right|, \quad (1)$$

where n is the number of data points being missing.

Table 1. Parameters used in training single-well DNN models[a].

Parameter	Values
Training wells	1-1, 1-10A, 1-15, 2-2, 2-3, 2-5
Synthetic gap length (hours)	1, 24, 48, 72
Model input window (M hours)	24, 48, 72, 96, 120, 144, 168
Model output window (N hours)	1, 6, 12, 24, 48, 72, 120, 144, 168
Training period	2012-2015
Validation period[b]	2011
Testing Period[c]	2008 for well 2-5; 2017 for well 1-15; 2016 for all other wells

[a] no models were trained for combinations with $M < N$. ~~Each of the~~

[b] used to select the best DNN model configurations and hyperparameters.

[c] used to evaluate performance of DNN vs ARIMA.

3.1.1 Hyperparameter search for the optimal DNN architecture

We performed a hyperparameter search to explore different model architecture configurations, i.e., the number of LSTM layers, number of units per LSTM layer, number (and size of) dense layers, and activation functions. The search was performed on well 1-1 only due to computational cost. We chose the optimal DNN architecture using model performance on validation data set of well 1-1 (see Table 1) using MAPE defined in Eq. (1).

The final DNN architecture, as shown in Figure 4, contains three LSTM layers, followed by two dense layers with dropout, a convolutional layer, and a final output dense layer. Stacking three layers of LSTM was found to yield better performance than a one- or two-layer architecture. Each of the three LSTM layers has 128 units ~~, with~~ because this configuration outperformed

others with more or fewer number of units. The output from the last LSTM layer ~~returning an~~ of size M x 128 ~~output, which~~ is fed into two consecutive dense layers ~~with dropout of 0.3, reducing the output from M x 128 to M x 64. A dense layer is a neural network~~ where every input neuron is connected to every output neuron with a weight matrix and bias vector. ~~Dropout is a regularization technique that randomly disables a select fraction of neurons during training to enhance robust model performance and prevent overfitting (Hinton et al., 2012).~~ The output from the second dense layer of size M x 64 is fed into a convolutional layer with 24 filters of size M-N+1, reducing the output size to N x 24. Finally, a dense layer is applied to yield a model output of our desired size, ~~N x 3.~~ $N \times 3$ for a single well or multiple of that when the model is designed to fill in data gaps in multiple wells. The detailed structures of the LSTM layers, dense layer, and convolutional layer are provided in the supplemental material. Dropout, i.e., randomly disables a selected fraction of neurons, was used in the dense layers as the regularization technique to enhance robust model performance and prevent overfitting (Hinton et al., 2012). We adopted a dropout rate of 0.3 after testing a set of alternatives (0, 0.1, 0.2, 0.3 and 0.4).

Each memory unit in the LSTM layer is further illustrated in Figure 5. The top panel shows generic representations of an RNN ~~(Olah, 2015)~~ in a looped (left) or ~~chain~~ chained (right) form, which allows information to be passed to the next successor and persist. While all RNNs have the form of a chain of repeating modules of neural network (i.e., boxes labeled as ~~A~~ A in Figure 5), the module being repeated can take different structural design to control the information flow, leading to different variants of RNN. Standard LSTMs use three gates, as shown in the bottom panel of Figure 5, to control the flow of information from one state to another and capture long-term dependencies. Each gate is composed of a ~~sigmoid neural net layer and a pointwise multiplication operation~~linear layer with a sigmoid activation function. A forget gate (~~F_t~~$\boldsymbol{f_t}$) decides what information to throw away from the previous memory state by using a sigmoid function that outputs a value between 0 and 1, where 0 represents completely forget the information and 1 represents completely keep the information. An input gate (~~I_t~~$\boldsymbol{i_t}$) decides which values from the new input to be used for updating the memory state. The input gate is combined with a vector of new candidate input values out of a tanh layer (~~generate~~ generates values between -1 and 1) through pointwise multiplication to ~~generate~~ yield information to be added to the current state. Finally, an output gate (~~O_t~~$\boldsymbol{o_t}$) decides what to output based on the input and the previous memory state. The previous hidden state and the current input are passed to a sigmoid layer of the output gate~~decides what parts of the memory state will be output~~, while the tanh layer scales the current memory state. ~~The~~ Then, pointwise multiplication of the outputs from the tanh and sigmoid layers leads to the output of this repeating module. For a more detailed description of the components of the LSTM unit, the reader is referred to ~~Olah (2015) and Ma et al. (2015)~~ Hochreiter and Schmidhuber (1997).

3.2 ~~Training the Stacked LSTM Models~~

~~Training data for the stacked LSTM models are created by finding data segments of $M + N$ hours that have no missing values, i.e., no gaps in the data, for all three measurements over a specified monitoring window. The well data are then preprocessed by normalizing all measurements to fall between 0 and 1 using different scaling factors for each variable, as temperature measurements are on a scale of 10^1, SpC is on a scale of 10^{-1}, and water level is on a scale of 10^2. The model is trained~~

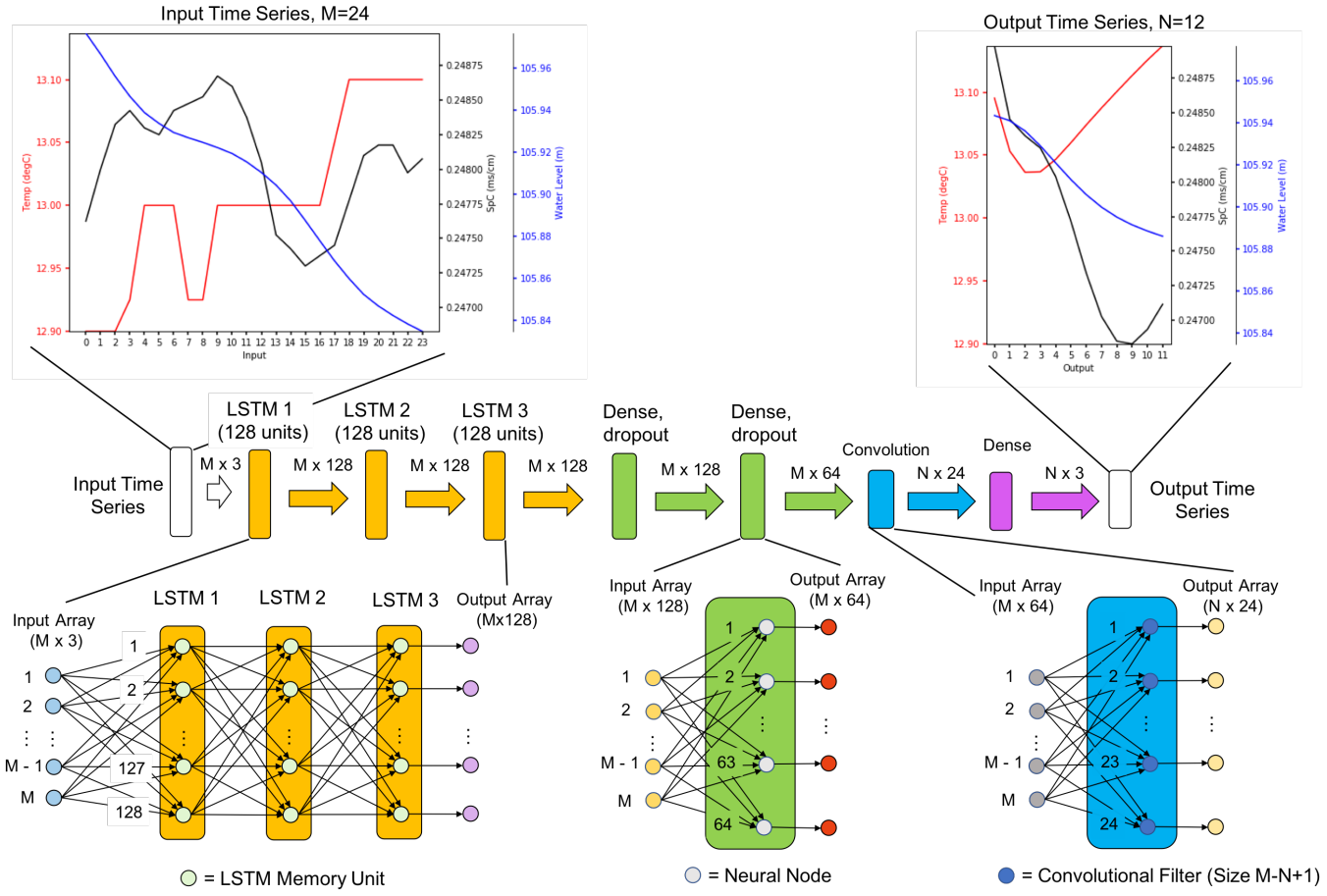

Figure 4. Architecture of the ~~stacked LSTM~~ DNN models, where M is the input window size and N is the output window size. Includes example input and output data with $M = 24$ and $N = 12$. For a more detailed diagram of the LSTM layers, dense layer, and convolutional layer, see Figures S1, S2, and S3, respectively, in the Supplemental Online Material.

~~for 30 iterations (i.e., epochs) over the training data. We use an Adam optimizer (Kingma and Ba, 2014) for training and the mean-squared error for the loss function. We train a stacked LSTM model for each desired combination of M,~~

3.1.1 Optimizing M and N for gap filling

Using the optimal DNN architecture, we further analyzed model MAPE metrics during the validation period with various combinations of M and N under each gap scenario for each well. The combination that yielded the lowest SpC MAPE were selected as the best configuration for a given gap length at each well~~given a certain amount of training data. The set of alternatives we consider for each model configuration parameter is shown in Table 1. Training wells were chosen based on adequate data availability and their distance from the river. Excluding combinations with $M < N$, a total of 810 unique models (135 model configurations for each of six wells) are trained.~~

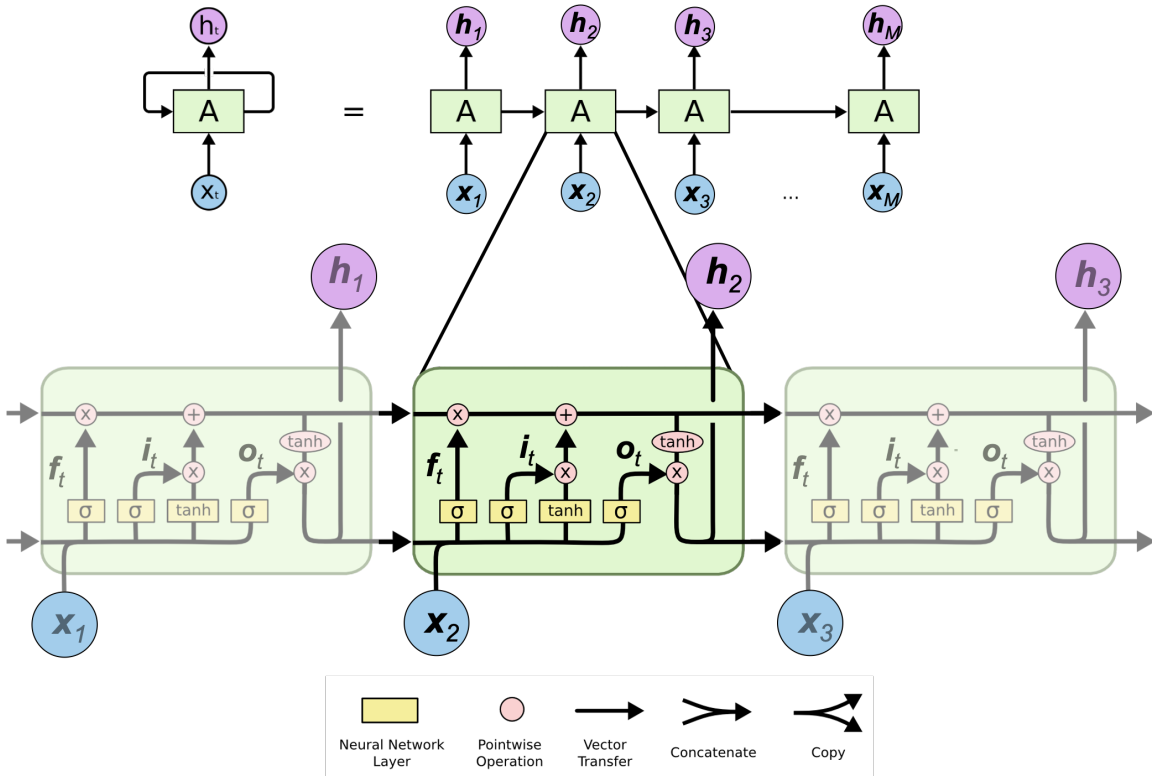

Figure 5. A diagram for network representing an LSTM unit. The top panel shows the looped and chain versions of a generic RNN, where ~~x_t~~ $\boldsymbol{x}_t$ is the input, ~~h_t~~ $\boldsymbol{h}_t$ is the output, and A is the repeating module of the LSTM unit. The bottom panel shows a diagram of the LSTM unit with the three main information gates: a forget gate (~~F_t~~ $\boldsymbol{f}_t$), an input gate (~~I_t~~ $\boldsymbol{i}_t$), and an output gate (~~O_t~~ $\boldsymbol{o}_t$). Images adapted from Olah (2015).

~~Parameters varied during LSTM model training. No models are trained where $M < N$. 24, 48, 72, 96, .~~ The best model configurations were then used to evaluate the LSTM-based DNN gap filling method against the ARIMA-based method (3.2) using relative errors (similar to MAPE by setting n=1 ~~, 6, 12, 24, 48, 72, 1-1, 1-10A, 2-2 2 years (2012-2013), 4 years (2012-2015)~~ in Eq. (1~~year (2011) 120, 144, 168 (hours) 120, 144, 168 (hours) 2-3, 2-5, 1-15 6 year (2010, 2012-2016)~~

3.2 ~~Testing the Stacked LSTM Models for Gap Filling~~

) and remove the absolute value operation) calculated for the testing period as listed in table 1. The test period varied among the wells due to availability of continuous data required for testing.

~~To evaluate the accuracy of the trained stacked LSTM models in filling in measurement gaps, we assume synthetic gaps of various lengths (e. g., 1, 24, 48, and 72 hours, referred to as gap scenarios hereafter) exist in a dataset containing all three variables monitored in year 2011 at each testing well (same as the training wells shown in Table 1). Given a model configuration (M, N, and training period), an LSTM model is trained for the six training wells , which is then tested on filling in synthetic data gaps of various lengths in all the wells. There are 36 training-testing well pairs in total. During testing, each model is~~

~~given the first M hours of data from the time series preceding the occurrence of a gap (assuming no missing values in these M hours). Then the model is used to fill in the first missing value in the gap by taking the first value of the predicted N hours from the model. This gap-filled value is then treated as if it was observed when repeating this procedure to fill in the gap of the next hour. The input data window keeps sliding in this way, hour by hour, until the model has filled the entire gap. The accuracy of the gap-filling model is evaluated by calculating the mean absolute percentage error (MAPE; %) between the values that are filled in (i.e., predicted) and the true values:~~

$$MAPE = 100 \times \frac{1}{n} \sum_{t=1}^{n} \left| \frac{\text{Prediction} - \text{Observation}}{\text{Observation}} \right|,$$

~~where n is the number of data points being missing.~~

3.2 ARIMA ~~model~~Models for Gap Filling

ARIMA is one of the most general classes of models for extrapolating time series to produce forecasts and we used it as a baseline to compare and assess the ~~LSTM-based~~ DNN gap-filling method. ARIMA is applicable to nonstationary processes in that the dataset can be made stationary by differencing if necessary. Differencing, autogressive, and moving average components make up a nonseasonal ARIMA(p,d,q) model given by:

$$Y_t = c + \phi_1 Y_{t-1}^d + \phi_p Y_{t-p}^d + \ldots + \theta_1 e_{t-1} + \theta_q e_{t-q} + e_t, \tag{2}$$

where ϕs and θs are polynomials of orders p and q, respectively, each containing no roots inside the unit circle. es are the error terms, $Y_\cdot^d$ is $Y_\cdot$ differenced d times, and c is a constant. Note that only non-seasonal terms (p,d,q) are included in Equ. (2). Seasonal structure can be added with parameters $(P,D,Q)_m$ to the base ARIMA model to become ARIMA$(p,d,q)(P,D,Q)_m$, including a periodic component containing m periods. $c \neq 0$ implies a polynomial of order $d+D$ in the forecast function. The detailed mathematical equations for seasonal ARIMA have been added in online supplemental materials.

The main task in ARIMA-based forecasting is to select appropriate model orders, i.e., the values of p,q,d,P,Q,D. If d and D are known, we can select the orders p,q,P,Q via an information criterion such as the Akaike Information Criterion (AIC):

$$AIC = -2\log(L) + 2(p+q+P+Q+k), \tag{3}$$

where $k=1$ if $c \neq 0$ and 0 otherwise, and L is the maximized likelihood of the model fitted to the differenced data. The best fitted parameters of the ARIMA model can be determined by minimizing the AIC.

Similar to the ~~LSTM-based~~ DNN-based gap filling, an ARIMA model is built for each combination of input and output window sizes for each well using the `auto.arima` function from the R package called `forecast` (Hyndman et al., 2007). The length of output N in ARIMA corresponds to the gap lengths. Each trained ARIMA model ~~is only~~ was tested on the well that is used for training the model, ~~not tested on any other wells (as is~~ as was done in the ~~LSTM approach)~~DNN approach. Accuracy of the ARIMA-based gap filling ~~can also be evaluated on the same synthetic gaps as in the LSTM approach~~ was evaluated using the same MAPE metric shown in Eq. (1) during the testing period listed in Table 1 and compared with the DNN-based methods.

4 Results and Discussion

4.1 Performance of ~~LSTM in filling gaps~~single-well DNN models

~~We evaluate~~ We evaluated the accuracy of ~~LSTM~~ DNN models in filling gaps of various lengths of the SpC measurements in the year 2011 following the steps described in section ~~3.3. MAPEs are summarized for different LSTM~~ 3.1.1. MAPEs were summarized in boxplots for different DNN model configuration parameters~~and for testing and training wells~~, as shown in Figure 6~~for SpC. Other similar analyses for groundwater table and temperature are done but not shown here because SpC is our primary interest for this study. Each MAPE shown in the plots represents an average of~~ . Each MAPE boxplot was drawn from a group of models with one parameter (corresponding to each x-axis) fixed at the given value while all the other parameters, including ~~training-testing well pairs~~ the training wells and gap scenarios, cycle through their possible combinations. ~~As can be seen in Figure 6 (a), using more training data improves the model performance on average, consistent with a standard observation in machine learning applications that model performance is highly dependent on the amount of training data available. When the amount of training data increases from 2 to 4 and 6 years, the MAPE drops slightly with consistent variability across all combinations. We therefore conclude that 2 years of data is sufficient to train the LSTM models we need for gap filling, although 4 years of training data would be better~~Although we attempted to train models with output window sizes greater than 24 hours, these models performed noticeably worse than those with output windows less than or equal to 24 hours (results shown in the online supplemental materials). Thus, our analyses here focus on models with output window less than or equal to 24 hours.

~~In~~ As shown in Figure 6 (~~b~~a), model performance deteriorates as the gap length increases~~. This is because the performance of LSTMs tends to degrade as it loses~~ , indicating that the DNN-based method tends to lose ground truth information from its input to ~~predict, i. e., the model begins to transition from interpolation to extrapolation. We also observe that models with a daily 24-hour input window outperform other models with longer input windows as~~ inform prediction. In comparing MAPEs across various input window sizes shown in Figure 6 (b), we observe that models with all input windows have comparable median MAPEs, with those of 24, 72, 144 and 168 hours leading to slightly smaller median MAPEs. The 144- and 168-hour input windows also yield lower third quartile of MAPE and fewer outliers on the larger MAPE end, indicating that the memory units in the LSTM layers are capturing important daily to weekly signatures (evident in WPS plots in Figure 2 for all wells except for Well 1-15) for some wells. As shown in 6 (c)~~. This likely results from an optimal number of memory units for capturing daily and subdaily memories. The output window lengths are shown in log scale in Figure 6 (d)to allow sufficient separation between the smaller time windows (i.e., 1, 6, and 12 hours). Daily and~~ daily and subdaily output windows yield comparable ~~performances in gap filling the SpC time series~~median MAPEs, with the ~~12-hour output window slightly outperforming~~ 24-hour output window leading to smaller third quartile and fewer large MAPE outliers than its 1-, 6-, and ~~24-hour counterparts. There is a significant performance deterioration when the output window increases from 24 hours to 48 hours and beyond.~~ 12-hour counterparts. Overall, an input window of ~~24~~ 144 hours and an output windows of ~~12~~ 24 hours appear to be a robust model configuration for all wells ~~considered.~~ and gap lengths considered.

We also tested how models trained on one well perform in filling gaps in other wells (defined as testing wells), given their vast differences in dynamic signatures. The performance of models trained on each well is single-well DNN models varied among the wells as shown in Figure 6 (e), from which we observe that models trained on wells d). The DNN models for well 1-15 , 2-3,1-10A, and 2-5 performed comparably in filling in gaps in the other wells, with the models of well 1-15 leading with a small margin. Models trained on well lead the performance with the smallest MAPEs, while those for well 2-2 yielded the largest error when tested to fill gaps in the 2011 testing data of all wells including itself. When the performanceis grouped by wells being tested, as shown in Figure 6 (f), all models can perform very well in filling in gaps for well 1-15 and reasonably well for yield the worst performance. The DNN models for wells 1-1,1-10A, 2-3 , and 2-5 , while all models appear to have difficulty in filling gaps for well 2-2. When selecting the optimal LSTM configuration of 24-hour input window and 12-hour output window using 4 years of training data, the models trained on well 2-3 perform the best in filling gaps of various lengths for all the 6 wells. performed comparably overall, with more large MAPE outliers for well 1-10A.

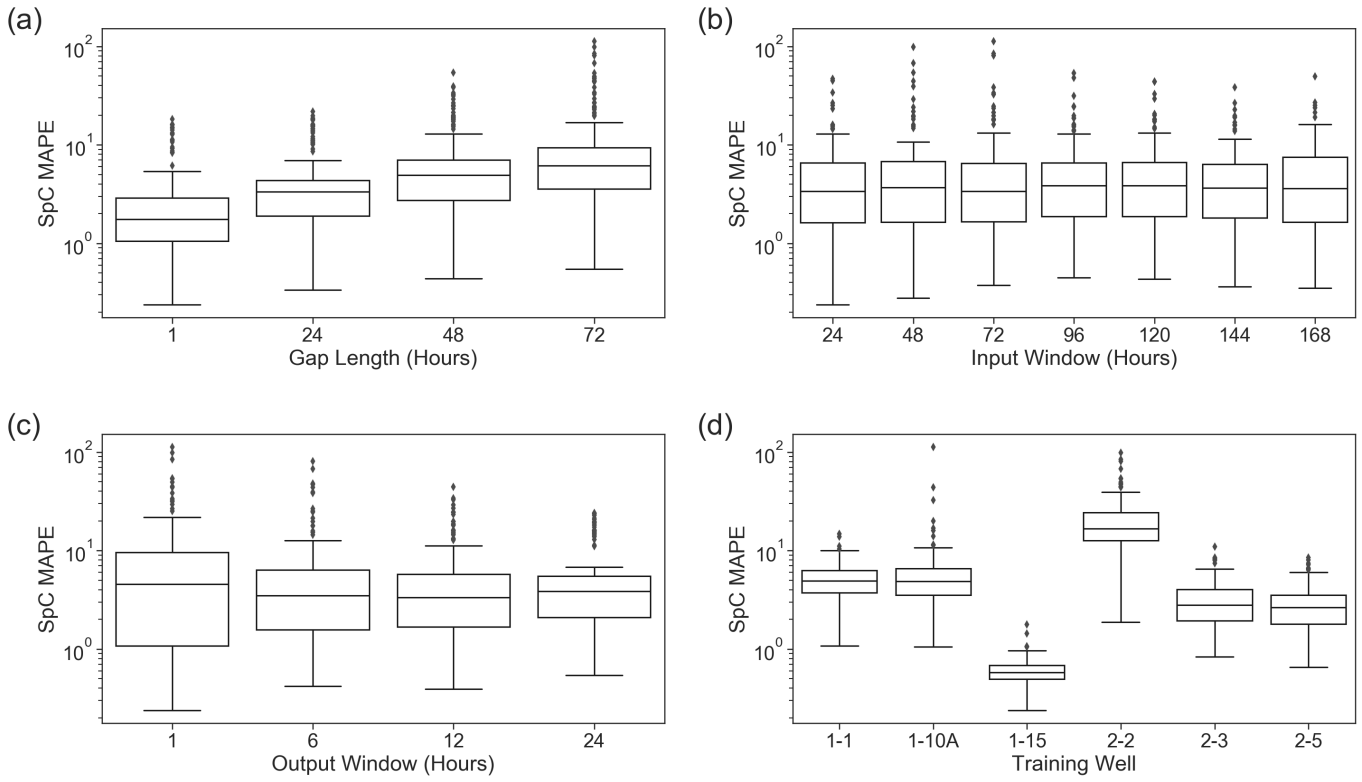

Figure 6. Gap filling performance for SpC evaluated against the validation datasets under multiple model configuration parameters (a-da-c) or grouped by training and testing wells (e and fd). (a) average MAPE vs. number distribution of years of training data, (b) average SpC MAPE vs. tested gap lengths, ; (eb) average distribution of SpC MAPE vs. model input window size M, ; (dc) average distribution of SpC MAPE vs. model output window size N, ; (ed) average distribution of SpC MAPE aggregated by wellsused to train the models, and (f) average MAPE aggregated by wells being tested on.95% confidence intervals of the averaged MAPE value are shown in shaded area in plots (a) -(d) and as the error bars in (e) and (f).

4.2 ~~ARIMA~~ Single-well DNN and ~~LSTM~~ ARIMA comparisons

~~Both ARIMA and LSTM approaches are tested in filling gaps of various lengths. Figure 8 shows the interquartile ranges of~~ The single-well DNN-based gap filling approach was compared to the ARIMA approach using relative errors calculated for each data point that ~~is~~ was assumed to be missing in the testing data by setting n=1 in Eq. (1) ~~, each bounded by its 25th to 75th percentiles under different gap lengths. The relative errors shown in Figure 8 are the results using~~ for MAPE. Best model configurations determined on the validation dataset (i.e., data from year 2011), as described in sections 3.1.1 and 3.2, were used in comparing the two approaches. Figure 7 illustrates the input and output windows selected as the best model configurations ~~trained for each well by the two approaches respectively. The gray lines representing~~ for DNN and ARIMA methods. The output window N of an ARIMA model is the same as the length of the gap it is built to fill. We observe that DNN models require less or equal input information than that required by the ARIMA method for the wells tested. None of the optimal output window sizes exceeds 24 hours for the DNN models. We only compared two methods for gap lengths of 24, 48 and 72 hours because both methods were highly accurate in filling small gaps such as one hour.

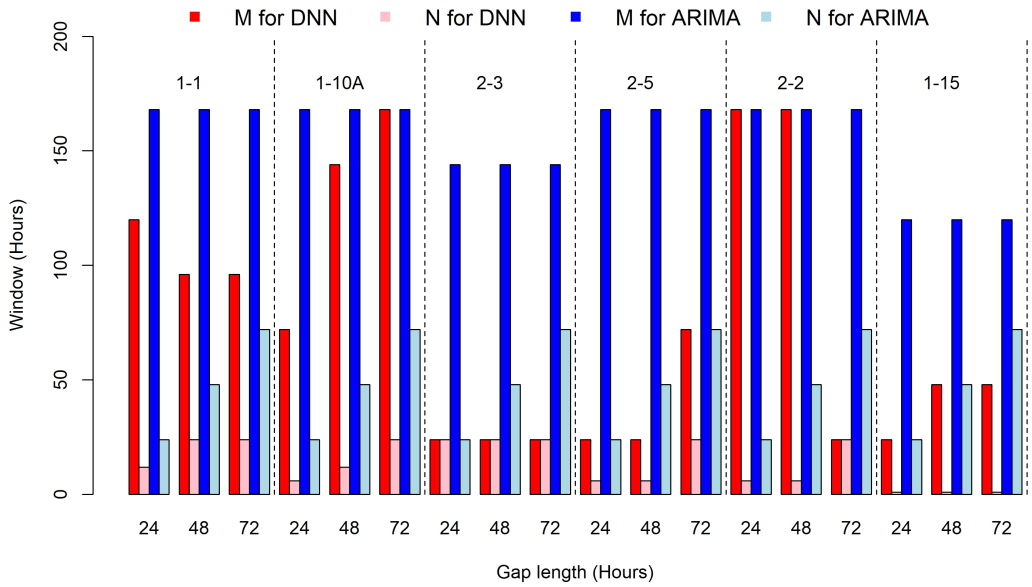

Figure 7. Best input and output windows for DNN and ARIMA models for filling gaps of various lengths at each well.

Figure 8 shows the interquartile ranges of relative errors under different gap lengths for all individual wells, each bounded by its 25th to 75th percentiles. Relative errors were used to show overestimations or underestimations by both approaches. The horizontal dotted lines represent the ±5% relative error range ~~for each model correspond to~~ that are typical measurement errors of the SpC sensors deployed at the site. ~~Most of the relative errors yielded from both gap-filling methods are within ±5% measurement error except for well 2-2 with 48- and 72-hour gaps.~~ The ARIMA models tend to perform better than the

~~LSTM~~ DNN models in terms of error statistics. For both approaches, the relative errors increase as the gap length increases as expected, especially so for well 1-1 when the gap lengths are 48 and 72 hours. While all the relative errors yielded from the ARIMA method are within the ±5% measurement error range, there are a few cases using DNN leading to relative errors outside the typical observational error range with longer gaps for wells 1-1, 1-10A and 2-2. The relative errors in the ARIMA models tend to distribute symmetrically on both sides of 0%, whereas errors in the ~~LSTM models~~ DNN models appear to skew toward the negative side for ~~wells 1-1 and 2-2 and towards the positive side for wells 1-10A and 2-3. Also for both approaches, the smallest and largest errors occur at wells 1-15 and 2-2, respectively.~~ all wells except for well 2-5. For well 1-15, the relative errors for all three gap lengths are very close to 0. ~~Well~~ Wells 1-1 and 2-2 ~~has the largest~~ have larger relative errors over the testing ~~window. 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. We also note that the optimal LSTM model configuration selected for each well based on its testing performance is different from that selected based on testing performance averaged across a range of training-testing well pairs.~~ period using both approaches.

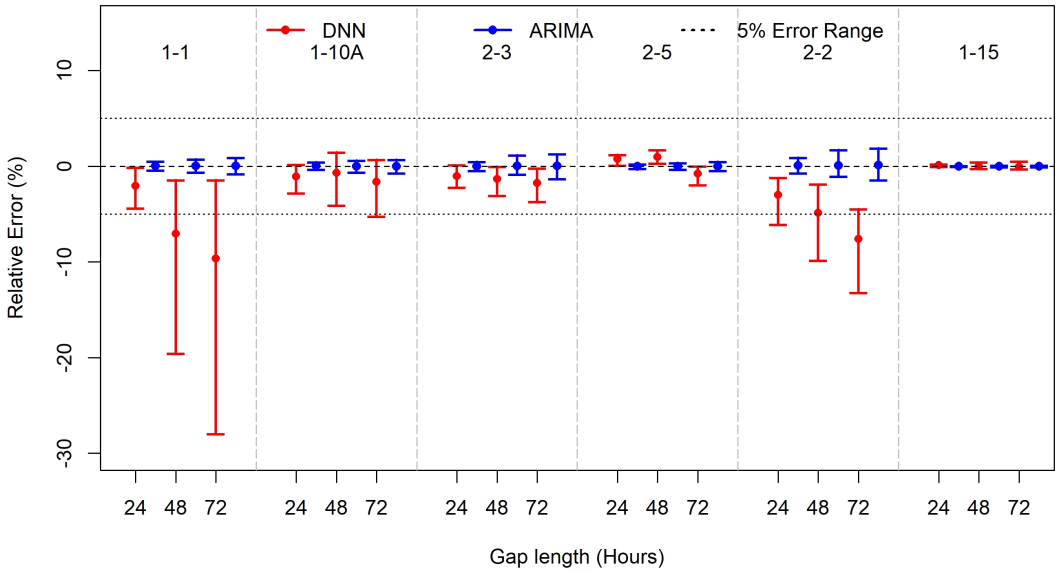

Figure 8. Summary of SpC relative errors for filling gaps of various lengths (i.e., 24, 48, and 72 hours) for best ~~LSTM~~ DNN and ARIMA models tested for each well.~~Their corresponding model input and output configuration (M and N for LSTM and and M for ARIMA) are shown for each well along the horizontal axis.~~

In addition to the error statistics, it is also important to examine how well a gap-filling method ~~can capture~~ captures the desired dynamic patterns in the gap-filled time series. Therefore, the SpC time series reproduced by the gap-filling methods ~~for the testing dataset~~ during the testing period (2016 for wells 1-1, 1-10A, 2-2, 2-3; 2017 for well 1-15; 2008 for well 2-5) with 24-hour synthetic gaps are evaluated against the real time series. Model configurations are the same as those used in error

statistics comparison (Figure 7). As shown in the first and second columns of Figure 9, the ARIMA approach (column 1) can capture the smooth changes in the observations but not abrupt changes that occur over a short time window (i.e., at higher frequency); these occur in all wells except 1-15. The spikes in errors during those rapid changes were not captured in Figure 7 as they are outside of the interquartile ranges of the relative errors. This is an indication that ARIMA fails to capture ~~higher frequency~~ high-frequency (daily and subdaily) dynamics and nonlinear trends despite having smaller ~~errors on average. The LSTM~~ error quartiles. The DNN approach, on the other hand, is able to better ~~resolve nonlinearity, nonstationary, and highly dynamic temporal patterns in time series, despite not having as small relative errors as the ARIMA approach. This holds for nearly all wells, including~~ capture such dynamics in some wells (e.g., wells 1-15, 2-3 and 2-5). However, the DNN approach appears to overestimate the high-frequency (daily and subdaily) fluctuations in some wells near the river (i.e., wells 1-1, 1-10A, and 2-2), which contributed to less desirable relative errors distributed between the first and third quartiles (7). This is likely caused by the variability in dynamics signatures among the training, validation and test periods. For well 1-15, which ~~has~~ exhibits less dynamic behavior ~~. Both~~ in SpC, both gap filling methods ~~exhibit difficulties in filling gaps for well 2-2~~ perform well in terms of both the relative errors and capturing ~~real dynamic patterns, especially during January, October, and November when the SpC appeared to be highly dynamic .~~ the dynamic patterns.

~~To~~

To further investigate how the relative performance of ~~both~~ the two gap-filling methods depends on the inherent dynamics in each time series, ~~wavelet analyses results~~ spectral analyses for the testing SpC ~~dataset are extracted from the~~ datasets were performed using the same wavelet decomposition method for the multi-year analyses (shown earlier in Figure ~~2~~2). As shown in Figure 9~~for the testing window of year 2011~~, the time windows of high relative errors are found to approximately co-locate with the time when high-frequency (daily and subdaily) signals are gaining more power. The ~~LSTM models tend to outperform the ARIMA models~~ difference between the DNN and ARIMA models tend to be amplified during those time windows. Wells 1-1, 1-10A, ~~2-3, and 2-5~~ and 2-2 share similar seasonal patterns in WPS, with the highest intensity bin above 1024 hours ~~. Among these four wells, well 2-5~~ across February to July. Their average WPSs all show peaks around daily and subdaily frequencies. Well 2-3 has its greatest ~~intensity above 2048 hours across the entire year~~energy between 16 to 256 hours from January to July. Well 2-5 has low intensities of variability at daily and subdaily frequencies with the low-frequency variations (monthly and seasonal) dominating the Jan to March time frame. For well 1-15, ~~the strongest intensities tend to group into three bands: one at~~ one of its strongest intensities is above 2048 hours across the entire year, ~~one between 256 to 2048 hoursfrom January to March, and one occurring below 128 hours in June~~and the other strong intensities are narrow bands between 16 to 256 hours. In general, both ~~LSTM~~ DNN and ARIMA are effective at capturing ~~longer term variability , but LSTM~~ low-frequency variability (monthly and seasonal). Although DNN is more effective at capturing high-frequency (daily and subdaily) fluctuations and nonlinearities in the ~~dataset~~datasets, it may also lead to overly dynamic predictions when the training data contain more significant high-frequency signatures than the system behavior to be predicted.

~~In terms of computational cost , ARIMA~~

There is also significant difference in computational cost between the DNN and ARIMA methods for gap filling. ARIMA requires very little computational resources: the `auto.arima` function in R requires approximately 40 seconds for fit and

validate a model using one year of data on a personal computer with a 3.00 GHz CPU. Conversely, training and ~~testing a single LSTM~~ validating a single DNN model takes approximately 20-30 minutes on dual NVIDIA P100 12GB PCI-e based GPUs.

~~Columns~~

4.3 Performance of multi-well DNN models

We evaluated the predictive ability of the multi-well DNN models in filling gaps of various lengths in the SpC data at well 1-1 by comparing the performance against their single-well counterparts. Well 1-1 was chosen because of data availability in nearby wells (wells 1-10A and 1-16A). Moreover, both the ARIMA and single-well DNN methods had difficulty in capturing its dynamic patterns as discussed in Section 4.2. Similar to the single-well DNN model for well 1-1, the multi-well DNN models also predict the three variables for the well 1-1 only. We adopted the same DNN architecture from the single-well models and trained the same set of alternatives considering input window sizes and output window sizes for various gap lengths as listed in Table 1. Only output window sizes smaller than 24 hours were considered as learnt from the single-well models. The same training and validation periods were adopted to select the optimal combination of M and N. Results were summarized in Figure 10, where each boxplot was generated in the same manner as in Figure 6.

Compared to the single-well DNN models, the multi-well DNN models significantly improve the gap filling accuracy at well 1-1 with longer gaps (48 and 72 hours) while perform comparably with smaller gaps (i.e., 1 and ~~2 show time series of ARIMA~~ 24 hours), as shown in Figure 10 (a). The multi-well DNN models reduce the fraction of larger MAPEs under all the input window sizes (figure 10 (b)) and all output windows (figure 10 (c)). Further one to one comparisons between different M and ~~LSTM~~ N combinations are provided in figure 11 under different gap scenarios. Natural log scales were on both axes for better separations in data points. All the points below the 1:1 line represent cases where a multi-well DNN outperforms a single-well DNN. The percentage of points below the 1:1 line increases with the gap lengths: 17.9%, 35.7%, 64.3%, and 82.1% for gaps of 1, 24, 48, and 72 hours, respectively. Therefore, including spatial information from neighbouring wells could potentially increase the chance of successes in filling gaps under more challenging circumstances, such as more complex dynamic patterns and longer data gaps.

5 Conclusion

In this study, we implemented ~~an LSTM~~ a DNN-based gap filling method to account for spatio-temporal correlations in ~~a~~ monitoring data. We extensively evaluate the new method in filling data gaps in SpC measurements that are often used to indicate groundwater and river water interactions along river corridors. We optimized a DNN architecture that contains stacked LSTM, convolutional, and dense layers to take advantage of a 10-year spatially distributed multi-variable time series dataset collected by a groundwater monitoring well network ~~. We evaluated the performance of the LSTM-based gap filling method by creating synthetic data gapsof various lengths (24, 48 and 72 hours) so that the accuracy of the filled data could be quantified in terms of error statistics and how well the original nonlinear dynamics are captured. The performance of the LSTM-based~~ for filling SpC data gaps. A primary advantage of using DNN is the ability to incorporate spatio-temporal correlations and

nonlinearity in model states without assuming an explicit form of correlations or nonlinear functions in advancing system states as a priori. We compared the performance of single-well DNN-based gap-filling method ~~is compared to that of a traditional , popular~~ with a traditional gap-filling method, ARIMA~~.~~, to evaluate how well a DNN can capture multi-frequency dynamics. We also trained DNN models that take input from multiple wells to predict responses at one well. The multi-well DNN models were compared with single-well models to assess the improvement in gap filling performance by including additional spatial correlation from neighboring wells.

In general, both ~~ARIMA and LSTM are~~ DNN and ARIMA were highly accurate in filling small data gaps (i.e., 1 hour). They were reasonably effective at filling in gaps of 24, 48, and 72 hours. The relative errors ~~are~~ were mostly within the range of instrument measurement error. ~~The models both capture~~ Both models captured the long-term trends in data ~~,~~(i.e., low-frequency variations at the monthly or seasonal time scales), except during some time windows with highly dynamic fluctuations. ~~ARIMA is~~ The ARIMA method was found to be suitable for time series with less dynamic behavior. ~~LSTMs~~ DNNs excel in dealing with high-frequency dynamics (daily and subdaily) or nonlinearities, although they ~~do~~ require more training data and computational ~~power.~~ resources. The DNN approach also appeared to overestimate the high-frequency (daily and subdaily) fluctuations in some wells near the river (i.e., wells 1-1, 1-10A, and 2-2), which was likely caused by the variability in dynamics signatures among the training, validation and test periods. Availability of sufficient training data is critical for the success of ~~LSTM~~ DNN-based methods, as is with any DNN-based learning ~~methods. In the gap-filling use case studied here, 2 years of training data yielded similar results compared with 4 and 6 years of training data. A general guideline is to have training data that covers various scenarios of inter-annual, seasonal, daily and even sub-daily dynamics, which is best assessed by understanding the nature of physical processes and drivers underlying the time series data~~method. Extrapolating the DNN models to conditions beyond those in the training data remains as a major challenge.

Wavelet analysis could provide useful insights to the dynamic signatures of the data and the change in composition of their important frequencies over time, which can serve as a prior basis for selecting an appropriate gap-filling method. For example, the ARIMA method would work well if the dynamics are dominated by seasonal cycles, while more sophisticated approaches like ~~LSTMs~~ DNN-based methods could work better if there is evidence of weekly, daily and subdaily fluctuations. ~~There may also be challenging situations for LSTMs, such as the highly dynamic time windows in our case study. The capability of learning over long sequences differentiates LSTM from other RNNs that do not have such memory.~~ Depending on the mixture of ~~long- and short-term~~ high- and low-frequency variability inherent in the time series, different ~~LSTM~~ DNN architecture and configurations can be ~~further~~ explored and evaluated ~~to~~ through hyperparameter searches with respect to LSTM layers, dense layers and activation functions to achieve better performance in capturing more complex dynamics. ~~Capturing such dynamics~~ We also demonstrated that incorporating spatial information from neighboring stations in DNN could contribute to performance improvement under challenging scenarios with dynamic system behaviours with longer data gaps (multiple days). The optimal DNN model configuration and performance that could be achieved would vary case by case. The bidirectional LSTM can be explored to formulate the gap filling as a history matching problem and evaluate the value of observed data in future time window relative to missing data for filling the data gaps. While we introduced a new method that can be broadly applied to fill in gaps in irregularly spaced network for monitoring groundwater and surface water interactions, the

transferrability of this method to other monitoring systems could be evaluated more extensively by community participation. Capturing spatio-temporal dynamics in system states is essential for generating the most valuable insights to advance our understanding of dynamic complex systems. ~~Future research could involve time series from multiple locations to explicitly account for spatial correlations.~~

Code and data availability. The well observations have been made accessible at https://sbrsfa.velo.pnnl.gov/datasets/?UUID=14febd81-05b6-47fb-be52 -439c4382decd

Author contributions. HR and EC developed scripts and performed the analyses. BK contributed on interpretation of the results. XC conceived and designed the study. All authors contributed to writing the manuscript.

Competing interests. The authors declare that they have no conflicts of interest.

Acknowledgements. This research was supported by the U.S. Department of Energy (DOE), Office of Biological and Environmental Research (BER), as part of BER's Subsurface Biogeochemical Research Program (SBR). A portion of methodology development was supported by the Laboratory Directed Research and Development Program at Pacific Northwest National Laboratory, a multiprogram national laboratory operated by Battelle for DOE under contract DE-AC05-76RL01830. This research was performed using PNNL Institutional Computing at Pacific Northwest National Laboratory. This research was also supported in part by the Indiana University Environmental Resilience Institute and the *Prepared for Environmental Change* grand challenge initiative.

References

Alvera-Azcárate, A., Barth, A., Parard, G., and Beckers, J.-M.: Analysis of SMOS sea surface salinity data using DINEOF, Remote sensing of environment, 180, 137–145, 2016.

Amaranto, A., Munoz-Arriola, F., Corzo, G., Solomatine, D. P., and Meyer, G.: Semi-seasonal groundwater forecast using multiple data-driven models in an irrigated cropland, Journal of Hydroinformatics, 20, 1227–1246, 2018.

Amaranto, A., Munoz-Arriola, F., Solomatine, D., and Corzo, G.: A spatially enhanced data-driven multimodel to improve semiseasonal groundwater forecasts in the High Plains aquifer, USA, Water Resources Research, 55, 5941–5961, 2019.

Arntzen, E. V., Geist, D. R., and Dresel, P. E.: Effects of fluctuating river flow on groundwater/surface water mixing in the hyporheic zone of a regulated, large cobble bed river, River Research and Applications, 22, 937–946, 2006.

Banerjee, S., Carlin, B. P., and Gelfand, A. E.: Hierarchical modeling and analysis for spatial data, CRC press, 2014.

Beckers, J.-M. and Rixen, M.: EOF calculations and data filling from incomplete oceanographic datasets, Journal of Atmospheric and oceanic technology, 20, 1839–1856, 2003.

Beckers, J.-M., Barth, A., and Alvera-Azcárate, A.: DINEOF reconstruction of clouded images including error maps? application to the Sea-Surface Temperature around Corsican Island, 2006.

Calculli, C., Fassò, A., Finazzi, F., Pollice, A., and Turnone, A.: Maximum likelihood estimation of the multivariate hidden dynamic geostatistical model with application to air quality in Apulia, Italy, Environmetrics, 26, 406–417, 2015.

Chen, X., Murakami, H., Hahn, M. S., Hammond, G. E., Rockhold, M. L., Zachara, J. M., and Rubin, Y.: Three-dimensional Bayesian geostatistical aquifer characterization at the Hanford 300 Area using tracer test data, Water Resources Research, 48, https://doi.org/10.1029/2011WR010675, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2011WR010675, 2012a.

Chen, X., Murakami, H., Hahn, M. S., Hammond, G. E., Rockhold, M. L., Zachara, J. M., and Rubin, Y.: Three-dimensional Bayesian geostatistical aquifer characterization at the Hanford 300 Area using tracer test data, Water Resources Research, 48, 2012b.

Chen, X., Hammond, G. E., Murray, C. J., Rockhold, M. L., Vermeul, V. R., and Zachara, J. M.: Application of ensemble-based data assimilation techniques for aquifer characterization using tracer data at Hanford 300 area, Water Resources Research, 49, 7064–7076, https://doi.org/10.1002/2012WR013285, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2012WR013285, 2013a.

Chen, X., Hammond, G. E., Murray, C. J., Rockhold, M. L., Vermeul, V. R., and Zachara, J. M.: Application of ensemble-based data assimilation techniques for aquifer characterization using tracer data at Hanford 300 area, Water Resources Research, 49, 7064–7076, 2013b.

Cheng, T., Haworth, J., and Wang, J.: Spatio-temporal autocorrelation of road network data, Journal of Geographical Systems, 14, 389–413, https://doi.org/10.1007/s10109-011-0149-5, https://doi.org/10.1007/s10109-011-0149-5, 2012.

Cheng, T., Haworth, J., Anbaroglu, B., Tanaksaranond, G., and Wang, J.: Spatiotemporal data mining, in: Handbook of regional science, pp. 1173–1193, Springer, 2014.

Connor, J. T., Martin, R. D., and Atlas, L. E.: Recurrent neural networks and robust time series prediction, IEEE transactions on neural networks, 5, 240–254, 1994.

Datta, A., Banerjee, S., Finley, A. O., and Gelfand, A. E.: Hierarchical nearest-neighbor Gaussian process models for large geostatistical datasets, Journal of the American Statistical Association, 111, 800–812, 2016.

Eidsvik, J., Shaby, B. A., Reich, B. J., Wheeler, M., and Niemi, J.: Estimation and prediction in spatial models with block composite likelihoods, Journal of Computational and Graphical Statistics, 23, 295–315, 2014.

Fang, K., Shen, C., Kifer, D., and Yang, X.: Prolongation of SMAP to Spatiotemporally Seamless Coverage of Continental U.S. Using a Deep Learning Neural Network, Geophysical Research Letters, 44, 11,030–11,039, https://doi.org/10.1002/2017GL075619, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1002/2017GL075619, 2017.

Faruk, D. Ö.: A hybrid neural network and ARIMA model for water quality time series prediction, Engineering Applications of Artificial Intelligence, 23, 586–594, 2010.

Finley, A. O., Banerjee, S., and Gelfand, A. E.: spBayes for large univariate and multivariate point-referenced spatio-temporal data models, arXiv preprint arXiv:1310.8192, 2013.

Gentine, P., Pritchard, M., Rasp, S., Reinaudi, G., and Yacalis, G.: Could Machine Learning Break the Convection Parameterization Deadlock?, Geophysical Research Letters, 45, 5742–5751, https://doi.org/10.1029/2018GL078202, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018GL078202, 2018.

Ghil, M., Allen, M., Dettinger, M., Ide, K., Kondrashov, D., Mann, M., Robertson, A. W., Saunders, A., Tian, Y., Varadi, F., et al.: Advanced spectral methods for climatic time series, Reviews of geophysics, 40, 3–1, 2002.

Grant, G. E. and Dietrich, W. E.: The frontier beneath our feet, Water Resources Research, 53, 2605–2609, 2017.

Graves, A.: Generating sequences with recurrent neural networks, arXiv preprint arXiv:1308.0850, 2013.

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

Graves, A., Mohamed, A.-r., and Hinton, G.: Speech recognition with deep recurrent neural networks, in: Acoustics, speech and signal processing (icassp), 2013 ieee international conference on, pp. 6645–6649, IEEE, 2013.

Griffith, D. A.: Modeling spatio-temporal relationships: retrospect and prospect, Journal of Geographical Systems, 12, 111–123, 2010.

Grinsted, A., Moore, J. C., and Jevrejeva, S.: Application of the cross wavelet transform and wavelet coherence to geophysical time series, Nonlinear processes in geophysics, 11, 561–566, 2004.

Grossmann, A. and Morlet, J.: Decomposition of Hardy functions into square integrable wavelets of constant shape, SIAM journal on mathematical analysis, 15, 723–736, 1984.

Güler, C. and Thyne, G. D.: Hydrologic and geologic factors controlling surface and groundwater chemistry in Indian Wells-Owens Valley area, southeastern California, USA, Journal of Hydrology, 285, 177–198, 2004.

Han, P., Wang, P. X., Zhang, S. Y., and Zhu, D. H.: Drought forecasting based on the remote sensing data using ARIMA models, Mathematical and Computer Modelling, 51, 1398–1403, 2010.

Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R.: Improving neural networks by preventing co-adaptation of feature detectors, arXiv preprint arXiv:1207.0580, 2012.

Ho, S., Xie, M., and Goh, T.: A comparative study of neural network and Box-Jenkins ARIMA modeling in time series prediction, Computers & Industrial Engineering, 42, 371–375, 2002.

Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, Neural Comput., 9, 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735, https://doi.org/10.1162/neco.1997.9.8.1735, 1997.

Hochreiter, S., Bengio, Y., Frasconi, P., Schmidhuber, J., et al.: Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

Hocke, K. and Kämpfer, N.: Gap filling and noise reduction of unevenly sampled data by means of the Lomb-Scargle periodogram, Atmospheric Chemistry and Physics Discussions, 8, 4603–4623, 2008.

Hyndman, R. J., Khandakar, Y., et al.: Automatic time series for forecasting: the forecast package for R, 6/07, Monash University, Department of Econometrics and Business Statistics, 2007.

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

Kamarianakis, Y. and Prastacos, P.: Forecasting traffic flow conditions in an urban network: Comparison of multivariate and univariate approaches, Transportation Research Record: Journal of the Transportation Research Board, pp. 74–84, 2003.

Kamarianakis, Y. and Prastacos, P.: Space–time modeling of traffic flow, Computers & Geosciences, 31, 119–133, 2005.

Katzfuss, M. and Cressie, N.: Spatio-temporal smoothing and EM estimation for massive remote-sensing data sets, Journal of Time Series Analysis, 32, 430–446, 2011.

Katzfuss, M. and Cressie, N.: Bayesian hierarchical spatio-temporal smoothing for very large datasets, Environmetrics, 23, 94–107, 2012.

Kingma, D. P. and Ba, J.: Adam: A Method for Stochastic Optimization, CoRR, abs/1412.6980, http://arxiv.org/abs/1412.6980, 2014.

Kondrashov, D. and Ghil, M.: Spatio-temporal filling of missing points in geophysical data sets, Nonlinear Processes in Geophysics, 13, 151–159, 2006.

Kondrashov, D., Shprits, Y., and Ghil, M.: Gap filling of solar wind data by singular spectrum analysis, Geophysical research letters, 37, 2010.

Kratzert, F., Klotz, D., Brenner, C., Schulz, K., and Herrnegger, M.: Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks, Hydrology and Earth System Sciences, 22, 6005–6022, https://doi.org/10.5194/hess-22-6005-2018, https://www.hydrol-earth-syst-sci.net/22/6005/2018/, 2018.

Lin, C. Y., Abdullah, M. H., Praveena, S. M., Yahaya, A. H. B., and Musta, B.: Delineation of temporal variability and governing factors influencing the spatial variability of shallow groundwater chemistry in a tropical sedimentary island, Journal of hydrology, 432, 26–42, 2012.

Liu, J., Shahroudy, A., Xu, D., and Wang, G.: Spatio-temporal lstm with trust gates for 3d human action recognition, in: European Conference on Computer Vision, pp. 816–833, Springer, 2016.

Längkvist, M., Karlsson, L., and Loutfi, A.: A review of unsupervised feature learning and deep learning for time-series modeling, Pattern Recognition Letters, 42, 11 – 24, https://doi.org/https://doi.org/10.1016/j.patrec.2014.01.008, http://www.sciencedirect.com/science/article/pii/S0167865514000221, 2014.

Ma, X., Tao, Z., Wang, Y., Yu, H., and Wang, Y.: Long short-term memory neural network for traffic speed prediction using remote microwave sensor data, Transportation Research Part C: Emerging Technologies, 54, 187–197, 2015.

Olah, C.: Understanding LSTM Networks, https://colah.github.io/posts/2015-08-Understanding-LSTMs/, 2015.

Pascanu, R., Gulcehre, C., Cho, K., and Bengio, Y.: How to Construct Deep Recurrent Neural Networks, 2013.

Pfeifer, P. E. and Deutrch, S. J.: A three-stage iterative procedure for space-time modeling phillip, Technometrics, 22, 35–47, 1980.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., and Prabhat: Deep learning and process understanding for data-driven Earth system science, Nature, 566, 195–204, https://doi.org/10.1038/s41586-019-0912-1, http://www.nature.com/articles/s41586-019-0912-1, 2019.

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

Schmidhuber, J.: Deep learning in neural networks: An overview, Neural Networks, 61, 85 – 117, https://doi.org/https://doi.org/10.1016/j.neunet.2014.09.003, http://www.sciencedirect.com/science/article/pii/S0893608014002135, 2015.

Schuster, M. and Paliwal, K. K.: Bidirectional recurrent neural networks, IEEE Transactions on Signal Processing, 45, 2673–2681, 1997.

Shen, C.: A Transdisciplinary Review of Deep Learning Research and Its Relevance for Water Resources Scientists, Water Resources Research, 54, 8558–8593, https://doi.org/10.1029/2018WR022643, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018WR022643, 2018.

Shuai, P., Chen, X., Song, X., Hammond, G. E., Zachara, J., Royer, P., Ren, H., Perkins, W. A., Richmond, M. C., and Huang, M.: Dam Operations and Subsurface Hydrogeology Control Dynamics of Hydrologic Exchange Flows in a Regulated River Reach, Water Resources Research, 55, 2593–2612, https://doi.org/10.1029/2018WR024193, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018WR024193, 2019.

Song, S., Lan, C., Xing, J., Zeng, W., and Liu, J.: An End-to-End Spatio-Temporal Attention Model for Human Action Recognition from Skeleton Data., in: AAAI, vol. 1, pp. 4263–4270, 2017.

Song, X., Chen, X., Stegen, J., Hammond, G., Song, H.-S., Dai, H., Graham, E., and Zachara, J. M.: Drought Conditions Maximize the Impact of High-Frequency Flow Variations on Thermal Regimes and Biogeochemical Function in the Hyporheic Zone, Water Resources Research, 2018.

Stockwell, R. G., Mansinha, L., and Lowe, R.: Localization of the complex spectrum: the S transform, IEEE transactions on signal processing, 44, 998–1001, 1996.

Strobl, R. O. and Robillard, P. D.: Network design for water quality monitoring of surface freshwaters: A review, Journal of environmental management, 87, 639–648, 2008.

Stroud, J. R., Stein, M. L., and Lysen, S.: Bayesian and maximum likelihood estimation for Gaussian processes on an incomplete lattice, Journal of computational and Graphical Statistics, 26, 108–120, 2017.

Sun, A. Y.: Discovering State-Parameter Mappings in Subsurface Models Using Generative Adversarial Networks, Geophysical Research Letters, 45, 11,137–11,146, https://doi.org/10.1029/2018GL080404, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018GL080404, 2018.

Sun, A. Y., Scanlon, B. R., Zhang, Z., Walling, D., Bhanja, S. N., Mukherjee, A., and Zhong, Z.: Combining Physically Based Modeling and Deep Learning for Fusing GRACE Satellite Data: Can We Learn From Mismatch?, Water Resources Research, 55, 1179–1195, https://doi.org/10.1029/2018WR023333, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2018WR023333, 2019.

Taylor, C. J. and Alley, W. M.: Ground-water-level monitoring and the importance of long-term water-level data, 1217-2002, US Geological Survey, 2002.

Vacha, L. and Barunik, J.: Co-movement of energy commodities revisited: Evidence from wavelet coherence analysis, Energy Economics, 34, 241–247, 2012.

Valenzuela, O., Rojas, I., Rojas, F., Pomares, H., Herrera, L. J., Guillén, A., Marquez, L., and Pasadas, M.: Hybridization of intelligent techniques and ARIMA models for time series prediction, Fuzzy sets and systems, 159, 821–845, 2008.

Wang, G., Garcia, D., Liu, Y., De Jeu, R., and Dolman, A. J.: A three-dimensional gap filling method for large geophysical datasets: Application to global satellite soil moisture observations, Environmental Modelling & Software, 30, 139–142, 2012.

Wett, B., Jarosch, H., and Ingerle, K.: Flood induced infiltration affecting a bank filtrate well at the River Enns, Austria, Journal of Hydrology, 266, 222–234, 2002.

Wikle, C. K., Berliner, L. M., and Cressie, N.: Hierarchical Bayesian space-time models, Environmental and Ecological Statistics, 5, 117–154, 1998.

Wu, Y., Schuster, M., Chen, Z., Le, Q. V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Łukasz Kaiser, Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M., and Dean, J.: Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation, CoRR, abs/1609.08144, http://arxiv.org/abs/1609.08144, 2016.

You, Q., Jin, H., Wang, Z., Fang, C., and Luo, J.: Image captioning with semantic attention, in: Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 4651–4659, 2016.

Zachara, J. M., Long, P. E., Bargar, J., Davis, J. A., Fox, P., Fredrickson, J. K., Freshley, M. D., Konopka, A. E., Liu, C., McKinley, J. P., et al.: Persistence of uranium groundwater plumes: contrasting mechanisms at two DOE sites in the groundwater–river interaction zone, Journal of contaminant hydrology, 147, 45–72, 2013.

Zachara, J. M., Chen, X., Murray, C., and Hammond, G.: River stage influences on uranium transport in a hydrologically dynamic groundwater-surface water transition zone, Water Resources Research, 52, 1568–1590, 2016.

Zachara, J. M., Chen, X., Song, X., Shuai, P., Murray, C., and Resch, C. T.: Kilometer-scale hydrologic exchange flows in a gravel-bed river corridor and their implications to solute migration, Water Resources Research, n/a, e2019WR025 258, https://doi.org/10.1029/2019WR025258, https://agupubs.onlinelibrary.wiley.com/doi/abs/10.1029/2019WR025258, e2019WR025258 2019WR025258, 2020.

Zhang, D., Lindholm, G., and Ratnaweera, H.: Use long short-term memory to enhance Internet of Things for combined sewer overflow monitoring, Journal of Hydrology, 556, 409 – 418, https://doi.org/https://doi.org/10.1016/j.jhydrol.2017.11.018, http://www.sciencedirect.com/science/article/pii/S0022169417307722, 2018.

Zhang, G. P.: Time series forecasting using a hybrid ARIMA and neural network model, Neurocomputing, 50, 159–175, 2003.

Zhu, W., Lan, C., Xing, J., Zeng, W., Li, Y., Shen, L., and Xie, X.: Co-Occurrence Feature Learning for Skeleton Based Action Recognition Using Regularized Deep LSTM Networks, https://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/11989/12149, 2016.

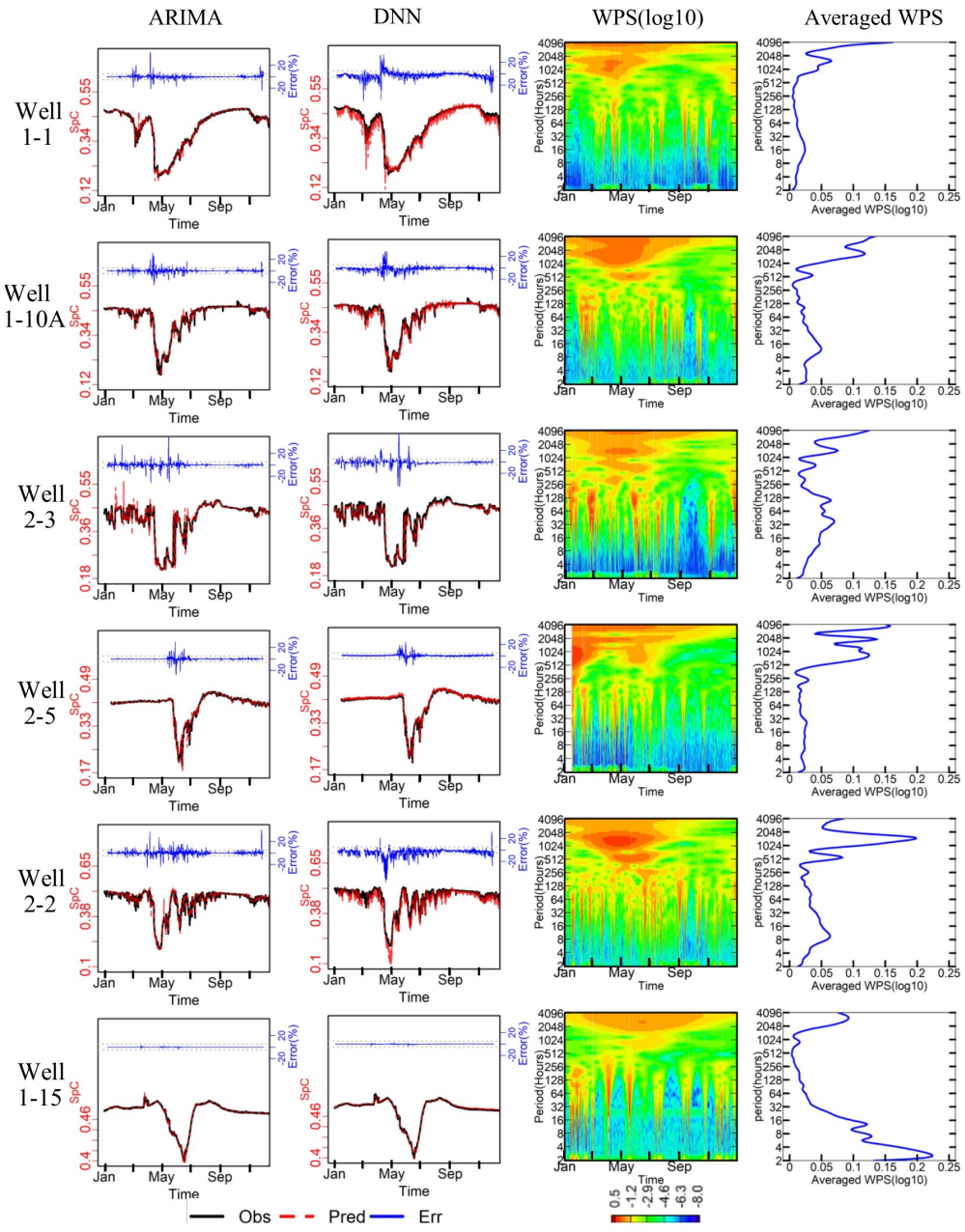

Figure 9. Columns 1 and 2 show time series of model predictions (in red) from ARIMA and DNN methods, respectively, assuming 24-hour synthetic gap in the SpC data, compared with observations (in black) and the relative errors (in blue). The best model configurations were used for all models. The testing data come from year 2016 for wells 1-1, 1-10A, 2-2, and 2-3, from year 2017 for well 1-15 and from 2008 for well 2-5. Column 3 is the spectrogram of each well and column 4 is the WPS averaged over for the corresponding year.

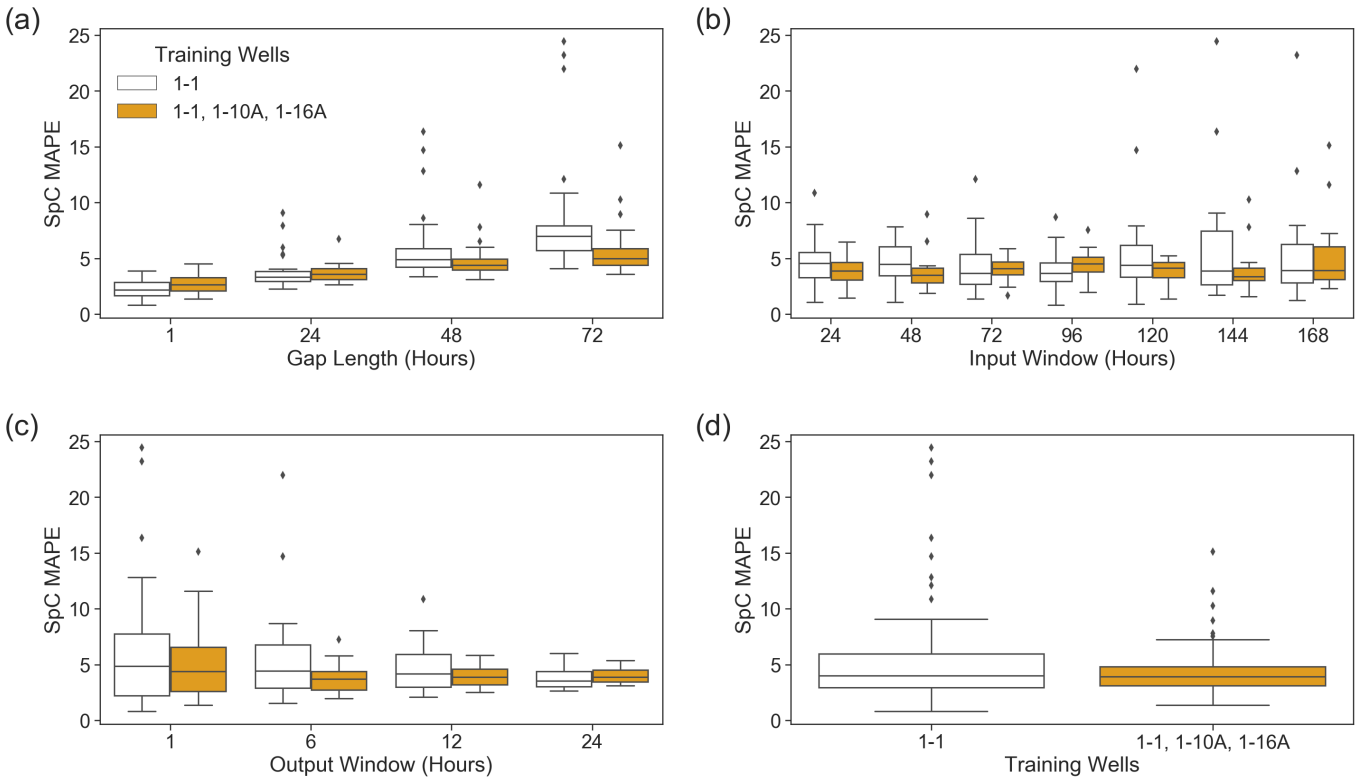

Figure 10. Comparing performance between single-well (~~red~~well 1-1) and multi-well DNN models (wells 1-1, ~~respectively, in~~ 1-10A and 1-16A) for filling ~~24-hour gap lengths, compared with observations~~ SpC data gaps for well 1-1 during the testing period (~~black~~year 2011)~~and relative error~~. (~~blue~~a) distribution of SpC MAPE vs. ~~The ARIMA~~ tested gap lengths; (b) distribution of SpC MAPE vs. model ~~takes 72-hour inputs, whereas the~~ input ~~and output~~ window ~~sizes for the LSTM model are 72 and 6 hours, respectively~~size M; (c) distribution of SpC MAPE vs. ~~The LSTM~~ model ~~is trained on 4 years~~ output window size N; (d) distribution of ~~data from well 2-3. Column 3 is~~ SpC MAPE aggregated by wells used to train the ~~spectrogram of each well for the year 2011, column 4 is the averaged WPS for year 2011.~~models

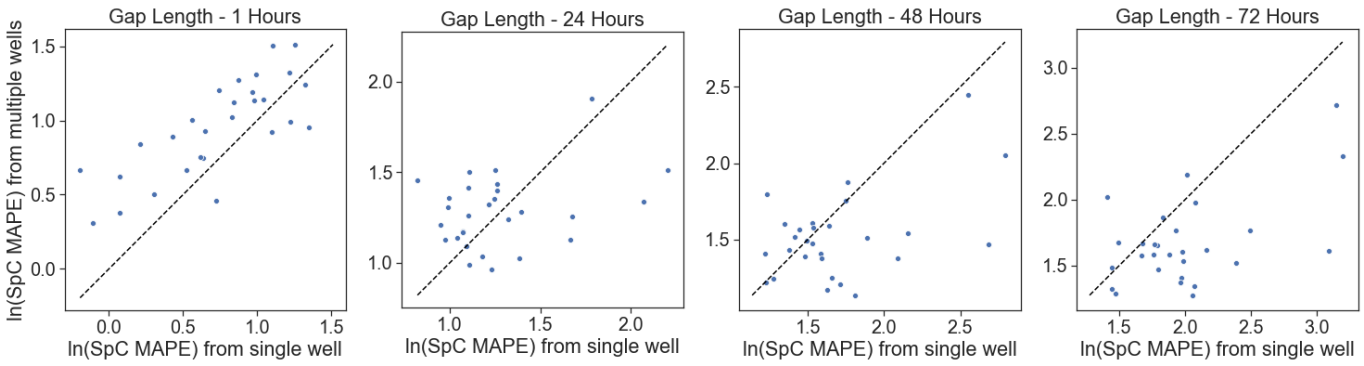

Figure 11. Comparing performance between single-well (well 1-1) and multi-well DNN models (wells 1-1, 1-10A and 1-16A) for filling SpC data gaps of various lengths for well 1-1 during the testing period (year 2011). The subplots (a)-(d) correspond to a gap length of 1, 24, 48 and 72 hours, respectively. Each data point represents a unique model configuration (size of input window, size of output window). The x-axis and y-axis are the natural log of SpC MAPE for single-well and multi-well DNN models, respectively. The dashed black line in each plot is the 1:1 line.