A wavelet-assisted deep learning approach for simulating groundwater levels affected by low-frequency variability

Sivarama Krishna Reddy Chidepudi, Nicolas Massei, Abderrahim Jardani, Abel Henriot, Delphine Allier, Lisa Baulon

A wavelet-assisted deep learning approach for simulating groundwater levels affected by low-frequency variability

Sivarama Krishna Reddy Chidepudi [a,b,*], Nicolas Massei[a], Abderrahim Jardani[a], Abel Henriot[b], Delphine Allier[b], Lisa Baulon[a,b]

[a] Univ Rouen Normandie, Univ Caen Normandie, CNRS, M2C, UMR 6143, F-76000 Rouen, France

[b] BRGM, 3 av. C. Guillemin, 45060 Orleans Cedex 02, France

* Email address: sivaramakrishnareddy.chidepudi@univ-rouen.fr

Abstract

Groundwater level (GWL) simulations allow the generation of reconstructions for exploring the past temporal variability of groundwater resources or provide the means for generating projections under climate change on decadal scales. In this context, analyzing GWLs affected by low-frequency variations is crucial. In this study, we assess the capabilities of three deep learning (DL) models (long short-term memory (LSTM), gated recurrent unit (GRU), and bidirectional LSTM (BiLSTM)) in simulating three types of GWLs affected by varying low-frequency behavior: inertial (dominated by low-frequency), annual (dominated by annual cyclicity) and mixed (in which both annual and low-frequency variations have high amplitude). We also tested if maximal overlap discrete wavelet transform pre-processing (MODWT) of input variables helps to better identify the frequency content most relevant for the models (MODWT-DL models). Only external variables (i.e., precipitation, air temperature as raw data, and effective precipitation (EP)) were used as input. Results indicate that for inertial-type GWLs, MODWT-DL models with raw data were notably more accurate than standalone models. However, DL models performed well for annual-type GWLs, while using EP as input, with MODWT-DL models exhibiting only minor improvements. Using raw data as input improved MODWT-DL models compared to standalone models; nevertheless, all models using EP performed better for annual-type GWLs. For mixed-type GWLs, while using EP as input, MODWT-DL models performed well, with substantial improvements over standalone models. Using raw data as input, improvement of MODWT-DL models is marginal compared to that of standalone models; nevertheless, they perform better than standalone models with EP. The Shapley Additive exPlanations (SHAP) approach used to interpret models highlighted that they preferentially learned from low-frequency in precipitation data to achieve the best simulations for inertial and mixed GWLs. This study showed that MODWT-based input pre-processing is highly suitable to better simulate low-frequency varying GWLs.

1. Introduction

Accurate and reliable groundwater level (GWL) simulations and forecasting play vital roles in water resource management (Q. Liu et al., 2022; Rahman et al., 2020). While the terms simulations and forecasting are inherently different, they are often used synonymously in an inappropriate way in groundwater studies (e.g., (Ghazi et al., 2021)), even after (Beven & Young, 2013) provided clear definitions of these terms. According to (Beven & Young, 2013), "simulation" is defined as quantitative reproduction of system behavior without reference to observed output; "forecasting" is defined as reproduction of system behavior ahead of time with observed outputs up until the onset of forecast included.

Traditionally, GWL simulations have been performed using physically based models, such as MODFLOW (Mcdonald et al., 1988) and ParFlow (Maxwell et al., 2015), in which a set of physical properties of the aquifer is implicitly used in the numerical solution of the groundwater equation. However, this type of calculation is often cumbersome and complicates the calibration of the model for approximating the hydrodynamic behavior of the aquifer expressed in observational data. Owing to the high computational and data requirements (Maxwell et al., 2015) of these models, data-driven approaches have gained traction in recent years as viable alternatives (Rahman et al., 2020; Rajaee et al., 2019; Tao et al., 2022). Furthermore, earlier studies on these approaches mainly focused on the classical multi-layer perceptron (MLP), i.e., a basic form of artificial neural network (ANN), which has outperformed MOFLOW in groundwater flow simulations (Mohanty et al., 2013). This simple network, which is considered to be the cornerstone of deep learning (DL), has demonstrated through several studies its effectiveness to approximate in black box form the highly nonlinear relationships that can link piezometric fluctuations to climate signals (Coulibaly et al., 2001; Wunsch et al., 2021). However,

the popularity of MLP networks in processing time series is decreasing in favor of a new generation of neural networks (such as LSTM and GRU). These neural networks include memory effect modules that handle temporal dependencies not accounted for in traditional neural networks, such as the MLP. This popularity is reflected in the numerous articles published in recent years that use this type of approach and highlights the effectiveness of standalone LSTMs in GWL simulations (Zhang et al., 2018), forecasting (Bowes et al., 2019), and reconstructing missing values (Vu et al., 2021).

To further improve the GWL forecasting performance of artificial intelligence (AI) algorithms, a new generation of algorithms called hybrids has been developed. These algorithms combine time-series pre-processing tools, such as principal component analysis (Cai et al., 2021), singular spectrum analysis (Yadav et al., 2020), DWT (C. Wu et al., 2021), and maximal overlap discrete wavelet pre-processing (MODWT) (Rahman et al., 2020) with AI tools. Among these, wavelet transform based pre-processing coupled with data-driven models has shown increased efficiency regarding GWL forecasting because of its ability to extract time-varying behavior (Rahman et al., 2020; Rajaee et al., 2019). However, a recent study by Quilty & Adamowski (2018) highlighted the frequent incorrect usage of wavelet transform in hydrological forecasting and recommended using boundary-corrected MODWT. This approach is now gaining traction in hydrological time series forecasting; for example, Mouatadid et al. (2019) combined LSTM with BC-MODWT for irrigation flow forecasting and found it to perform better than standalone LSTM. Rahman et al. (2020) used BC-MODWT along with random forest and XGBoost and found the approach promising for forecasting GWLs. Finally, a more recent study by Barzegar et al., (2021) coupled BC-MODWT with CNN-LSTM and found it promising for multiscale lake water level forecasting. However, it further highlighted that efforts are required to find a suitable wavelet family, filter length, and decomposition level. Even after (Quilty & Adamowski, 2018) suggested using BC-MODWT or AT, a few studies (Liang et al., 2021; C. Wu et al., 2021) continued to use DWT coupled with LSTM for groundwater forecasting. Other pre-processing techniques

have also been used for GWL forecasting; for example, (M. Wu et al., 2021) combined variational mode decomposition with an extreme learning machine (ELM), and (W. Liu et al., 2021) coupled CEEMDAN with the deep belief network (DBN) model.

All of these pre-processing techniques coupled with DL are mainly focused on forecasting, i.e., they use previous target values in the input. A recent study by (Bahmani & Ouarda, 2021) coupled EEMD and CEEMD with GEP and M5 with the aim of simulating GWL; however, they used GWLs as input, which contradicts the definition of simulation by (Beven & Young, 2013). Hence, the primary purpose of this study is to appropriately assess these approaches in simulations.

As models grow in complexity, there is increasing interest in their explanation and interpretation through approaches like SHapley Additive exPlanations (SHAP) which is mainly helpful in understanding the influence of input variables on the model simulations. According to recent literature, this is particularly true for ML and DL methods; for example, (Q. Liu et al., 2022) used the SHAP approach to interpret machine-learning models for near-term GWL simulations. This approach allowed the authors to identify that the flow volume and distance to the river and reservoir played significant roles in groundwater changes. (Anderson & Radić, 2022) also showed that DL models learn the contribution of glacial runoff from meteorological variables in streamflow modelling. Even though research in this direction is only beginning, there is significant potential for interpreting DL models, thereby increasing their credibility for wider adoption across different decision-making levels. However, SHAP interpretation is still not very common in hydrology; in particular, it has not been used to explore the impacts of pre-processing.

We identified the following lack of scientific and technical information in the existing literature: Most of the previous studies exploring wavelet usage were limited to forecasting, with previous GWLs being used as input. Consequently, these models relied heavily on high autocorrelation in previous GWLs values, resulting

in high accuracy. In contrast, our study focuses on simulation (not forecasting), in which only external factors influencing GWLs are used as input. Furthermore, we noticed a clear lack of studies looking at either forecasting or simulating GWL time series with the low-frequency variability (i.e., interannual to decadal) explaining most of the GWL variations. Indeed, being able to reproduce and establish simulations (e.g., projections) is critical for accounting for the influence of low-frequency climate variability, which can either mask or aggravate the effects of climate change on hydrology (Boé & Habets, 2014; Bonnet et al., 2022; Kingston et al., 2020).

We hypothesize that wavelet expansion, by extracting the most relevant high- to low-frequency variability information from external input variables (i.e., precipitation, air temperature, and effective precipitation), will help in achieving better simulations of GWL variations, even when these are dominated by low-frequency variability that is barely visible or even invisible in input variables. In this study, we evaluate the effectiveness of coupling recurrent-based DL algorithms (i.e., LSTM, bidirectional LSTM (BiLSTM), and GRU) with BC-MODWT signal decomposition for simulating GWL variations in the case of three piezometric time series representative of the most contrasting temporal behaviors in northern France (Baulon et al., 2022).

To test this hypothesis, we 1) evaluate the effectiveness of using raw data (i.e., precipitation and air temperature) as input against using processed data, such as effective precipitation (EP), as input, 2) compare DL methods with and without wavelet pre-processing while quantifying the performance of each method, and 3) investigate the internal functioning and plausibility testing of the DL models through the SHAP interpretative approach. The remainder of this paper is structured as follows: Section 2 presents the data and normalization. Section 3 presents the theoretical background of the methods and pre-processing techniques employed in this study. Section 4 presents the analysis and interpretation of the results. Section 5 presents our conclusions.

2. Data

We used a database consisting of a relatively long GWL time series initially taken from the ADES (Accès aux Données sur les Eaux Souterraines) database (https://ades.eaufrance.fr/; Winckel et al., 2022). These GWL data were especially selected because the anthropogenic impact on them was relatively low (Baulon et al., 2022). From this database, we selected three GWL time series that fluctuated over a 50-year period (i.e., 1970–2020) in contrasting manners: the first GWL time series has an inertial nature where the low-frequency component dominates the signal; the second is the so-called mixed GWL time series because the fluctuations are reflected by the annual and interannual components, with the latter being largely influenced by the annual cycle. Thus, the numerical simulation tools presented here are confronted with the prediction of piezometric fluctuations in aquifers with well-contrasted hydrodynamic regimes. The locations of the three stations are shown in Figure 1. The GWL time series of the three stations are shown in Figure 2.

The work presented herein is part of a much more comprehensive project study. At this stage, the current work uses precipitation, air temperature, or EP as input variables (Figure 3: Example for one station). The numerical simulation consists of training different types of DL networks with meteorological data as input. Specifically, EP is used as one input variable on the one hand, and precipitation and air temperature combined are used as input on the other hand; the output variable is piezometric responses.

Precipitation and mean air temperature data were retrieved from the SAFRAN (Système d'Analyse Fournissant des Renseignements Atmosphériques à la Neige) Reanalysis (Vidal et al., 2010), which is available at a daily time step with a spatial resolution of 8×8 km^2. Although the input time series is available at the daily time step, only the monthly averages between 1970–2020 were used to match the monthly time step of the piezometer data. EP was derived from a calculation using the water budget method of Edijatno &

Michel (1989). Conversely, the processed data came from areas in France with limited anthropogenic impacts; hence, the piezometric variations were essentially due to the climatic signal.

All data used as input and output are normalized into the network to facilitate stable convergence in the learning phase. The following equation is used for this normalization so that all data can vary in (0, 1) intervals:

$$x_{scale} = (x - x_{min})/(x_{max} - x_{min}), \quad (1)$$

Where x_{min} and x_{max} represent the minimum and maximum values of the data, respectively, whereas x and x_{scale} denote the original and scaled data, respectively.

We note that there is a common pitfall in this step, i.e., where the previous studies scaled all available data—including test data—together; this step results in data leakage by allowing the model to be influenced by the test data, which should not have been made available at this stage. Consequently, this can lead to overestimated results and undesirable expectations regarding the model's performance. In this study, this was taken into consideration, and we scaled out each input variable individually to be in the (0,1) range.

After normalization, the database is divided into two main subsets: the training and testing sets (constituting 80 and 20% of the database, respectively). Furthermore, the last 20% of the training set which is equivalent to 16% of total dataset available was used for validation as shown in Figure 4. The purpose of each split is as follows: the training data are used for fitting the model on them; the validation data are used for hyperparameter optimization and early stopping; and the testing data are used for obtaining the performance of the models on the unseen data.

Using only meteorological variables as input in this study utilizes the long time series of gridded data available at a finer spatiotemporal resolution across metropolitan France, with no missing values. For instance, SAFRAN reanalysis data are available on a daily scale for 70 years (i.e., 1951–2019) which is not the case for the GWL time series data, which are heterogeneous with missing values.

3. Theoretical background

The overall modeling methodology developed here involves the use and comparison of three types of recurrent-based deep neural networks, as well as a MODWT pre-processing approach to improve the recovery of the most useful information from the input data.

3.1 Wavelet decomposition pre-processing

In this study, we intended to test the usefulness of wavelet decomposition for detecting scale-dependent information to be used as input signals in high- to low-frequency wavelet components, thereby helping AI models to better simulate GWL variations. For instance, pre-processing approaches using wavelet transform have received particular attention in recent years; however, one significant contribution to the field (i.e., Quilty & Adamowski, 2018) highlighted the incorrect use of wavelet transform as a pre-processing tool for hydrological forecasting. In particular, the authors revealed associated future data issues when using DWT-MRA and MODWT-MRA; they suggested some best practices while testing them on wavelet-based ML methods for urban water demand forecasting.

In this methodological framework, we aim to develop the so-called direct approach (Quilty & Adamowski, 2018) and address the crucial constraints emphasized in this paper. In this approach, wavelet decomposition of the training dataset using MODWT is applied to the input signals only, thereby resulting in several high- to low-frequency wavelet components, each being used as input to the selected models.

When using MODWT-assisted DL modeling, all wavelet components of precipitation (Figure 5) and air temperature or EP time series (not shown) were used as input.

The MODWT algorithm was used to decompose the input variables into various scale levels and extract the variability of the decomposed signal at each time scale. This method is well-suited to real-world signals, as it enables the decomposition of a given signal (here, precipitation and air temperature or EP) into several components across different time scales (i.e., from high to low-frequencies), while keeping the amplitudes of the transform aligned with the amplitude in the original signal (Percival & Walden, 2000). This means that the discrete wavelet component still bears some "physical" meaning compared to the original decimated and non-redundant discrete wavelet transform. Decomposition can be achieved up to the maximum decomposition level depending on the length of the time-series; however, the decomposition depth (or filter length) must be constrained to avoid computing values being affected by boundary effects, which is even more critical when decomposition is included in a modeling framework (Quilty & Adamowski, 2018). In this study, the boundary-affected coefficients (L_J) were removed from the beginning of the input and target variables using the equation recommended by Quilty and Adamowski (2018):

$$L_J = (2^J-1)(L-1)+1.$$

Here, J is the number of decomposition levels, which has been set to 4, and L is the length of the filter used, i.e., eight wavelet/scaling values for La8, 12 for La12, and so on. Regarding the choice of wavelet filter, we chose the least asymmetric filters (i.e, La8 to La16), with periodic boundary conditions selected for the wavelet transform, meaning that the original signal is repeated after its last value is reached. The combined use of the MODWT and La filters, along with circular shifting of the computed wavelet and scaling coefficients, ensured the preservation of the phase alignment between the data and the calculated wavelet and scaling coefficients at each scale as much as possible. Scaling coefficients are not retained at all levels, but only at the final level;

in other words, all MODWT components except the last one correspond to wavelet coefficients, while the last component corresponds to scaling coefficients. Figure 5 shows the decomposed components of the precipitation at the conservative level. Further explanation of the MODWT and its main interest in hydrological applications can be found in Baulon et al. (2022) and Massei et al. (2017); for full mathematical details, we refer to Cornish et al. (2006) and Percival and Walden (2000).

3.2 Recurrent-based deep neural networks models

The long-term memory deep neural network (LSTM, Hochreiter, 1997) is a recurrent network designed to overcome leakage gradient problems and preserve long-term dependence through the inclusion of a hidden state, c_t, that retains historical information. In addition, LSTM has internal mechanisms in the form of control gates—namely forget, input, and output gates—for regulating the flow of information. The flow of processed information in simple LSTM is described in Appendix A and formulated mathematically below in three steps: In step 1, the forget gate manages the information from the prior cell state, c_{t-1}, while subsequently adding it to the present state with the help of the element-wise multiplication operator ($\otimes$) in the form of $f_t \otimes c_{t-1}$. This gate gives the binary output [0,1], with 0 and 1 indicating the deletion and retention of all previous information, respectively.

$$f_t = \sigma\left(W_f x_t + U_f h_{t-1} + b_f\right). \quad (2)$$

In the second step, the present cell state is computed in three phases. The first phase involves converting the values of x_t and h_{t-1} into the range of [-1,1] to obtain a new cell state, $\tilde{c}_t$, using an activation function (tanh).

$$\tilde{c}_t = tanh(W_c x_t + U_c h_{t-1} + b_c). \quad (3)$$

In the second phase, values resulting from the input gate (i_t) are used to reorganize the present cell state, c_t, as $i_t \otimes \tilde{c}_t$. The input gate regulates both the sequence of input data at present (x_t) and the hidden state information at $t-1$ (h_{t-1}), which is incorporated into the cell state as follows:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i). \quad (4)$$

In the final phase, the new cell state, c_t, is obtained by adding a revised cell state in step 1 ($f_t \otimes c_{t-1}$) with the updated cell state in previous phases (2. b) ($i_t \otimes \tilde{c}_t$).

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t. \quad (5)$$

In the third and last step, the information in the new cell state that must pass as an output of the present LSTM and the new hidden state to the upcoming cell is managed by the output gate (Alizadeh et al., 2021).

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o), \quad (6)$$

$$h_t = o_t \otimes tanh(c_t), \quad (7)$$

where W and U are the network weight matrices; f_t, i_t, and o_t constitute the output of the forget, input, and output gates, respectively; c_t, c_{t-1} are the cell states at t and $t-1$; σ is the sigmoid activation function; h_t and h_{t-1} are the current and previous hidden states, respectively; $\tilde{c}_t$ is the cell candidate value; and b is the bias vector.

Bidirectional LSTM trains two LSTM models, as shown in Appendix B. The first model learns the input sequence, i.e., through the forward state. In contrast, the second model learns from the opposite direction of the input sequence, i.e., through the backward state (Saeed et al., 2020), as depicted in Appendix B. Both models are merged using the concatenation mechanism by default. In other words, BiLSTMs include an additional layer of training data compared with simple LSTMs. Siami-Namini et al. (2019) showed that BiLSTMs outperformed regular LSTMs and ARIMA in time series forecasting because of the additional training layer in BiLSTM, which improves the learning of long-term dependencies. The internal processes of the cells used in BiLSTMs are explained above for the LSTM.

The gated recurrent unit (GRU) was developed by Cho et al. (2014) to address the complexity of LSTM and improve computational efficiency. While GRU is relatively similar to LSTM as shown in Appendix C, unlike LSTM, it has on gates (i.e., reset (r_t) and update (u_t)), as it does not have separate memory cells. The cell structure of GRU is shown in Appendix C. GRU and, LSTM are modified versions of RNN, where RNN is a type of ANN used for sequential data. Moreover, the GRU is computationally less expensive, with a faster learning curve owing to the lower number of learnable parameters. The GRU phases are summarized as follows:

$$u_t = \sigma(W_u x_t + U_u h_{t-1} + b_u), \quad (8)$$

$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r), \quad (9)$$

$$\tilde{c}_t = tanh(W_c x_t + U_c(r_t \otimes h_{t-1}) + b_c), \quad (10)$$

$$c_t = (1 - u_t) \otimes c_{t-1} + u_t \otimes \tilde{c}_t, \quad (11)$$

Where u_t and r_t are the update and reset gates, respectively; and b is the bias vector.

3.3 Hyperparameter tuning

Data-driven models involve using hyperparameters that must be defined during model initialization. Hyperparameters play a crucial role in controlling the overall training behavior of the model and substantially impact its performance; therefore, it is necessary to determine their optimal values. There are multiple approaches used in data-driven models for optimizing hyperparameters, the most common of which are the trial-and-error approach (Zhang et al., 2019), grid search (Afan et al., 2021), and random search. However, in recent years, an informed approach known as Bayesian optimization has gained traction in hydrological forecasting (Barzegar et al., 2021; Quilty et al., 2022; Rahman et al., 2020; Wunsch et al., 2022). The main advantage of this informed approach is that it takes less time to get comparable results as in random search, as it learns from previous iterations while allowing automatic hyperparameter selection. Here, Bayesian optimization was performed to minimize the mean squared error. The hyperparameters of all the models were tuned using Bayesian optimization, with the range of values shown in Table 1.

Based on hyperparameter space exploration, as well as on previous works revealing substantial hydrological variabilities at scales between 2–4 and 5–9 years over metropolitan France(Baulon et al., 2022; Fossa et al., 2021), northern France (Massei et al., 2010, 2017), and even Great Britain (Rust et al., 2019), the sequence length was eventually set to 48 months (i.e., four years), as shown in Table 1. Such variability is immediately evident, particularly in *Figure 2 (a, c),* and was found to originate from precipitation and large-scale climate

variability in the studies referenced earlier. For instance, such variability was also found in the North Atlantic Oscillation index (Massei et al., 2010; Massei & Fournier, 2012) or more directly related to particular sea-level pressure patterns over the Euro-Atlantic sector(Massei et al., 2017). The dropout approach was adapted to avoid overfitting the models during training. We subsequently tested values from 0.1 to 0.5 within the Bayesian optimization but noticed minimal changes, so we finally opted for the commonly used dropout value of 0.2, i.e., dropping out 20% of the neurons.

After identifying the best hyperparameters from 100 trials of Bayesian optimization using the validation set, we trained an ensemble of 30 pseudo-randomly initialized models and fitted them to the training set. This was performed to deal with epistemic uncertainty, which was mainly due to the initial model weights obtained from the random number generator seed; in other words, it addressed the uncertainties generated by the model structure. These ensemble models were tested on an unseen test set to evaluate their performances. Simulations obtained with different initializations were used to compute confidence intervals. The confidence interval limits were computed by adding and subtracting the 1.96 times standard deviation to the mean of the resulting distribution at each timestep.

3.4 Evaluation and interpretation

The current framework involved two models: 1. MODWT-assisted, i.e., including MODWT pre-processing, and 2. standalone models, i.e., without pre-processing.

Three common performance evaluation statistics were chosen to evaluate the models on both the training and test sets, namely the mean absolute error (MAE), root-mean-squared error (RMSE), and squared Pearson's correlation coefficient (R^2). Furthermore, the RMSE percentage change for MODWT-assisted models was computed with respect to standalone models to compare the performance improvement.

$$MAE = \frac{1}{N}\sum_{i=1}^{n} |GWL_{sim} - GWL_{obs}|, \qquad 0 \le MAE < +\infty, \tag{12}$$

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{n}(GWL_{sim} - GWL_{obs})^2}{n}}, \qquad 0 \le RMSE < +\infty, \tag{13}$$

$$R^2 = 1 - \frac{\sum(GWL_{sim} - GWL_{obs})^2}{\sum(GWL_{sim} - GWL_{obs_{mean}})^2}, \qquad -\infty < R^2 < 1 \tag{14}$$

$$\%\ \text{Improvement in RMSE} = \left(\frac{\mathrm{RMSE_{SA}} - \mathrm{RMSE_{WT}}}{RMSE_{SA}}\right) * 100. \quad 0 < \%\text{Improv in RMSE} < 100 \tag{15}$$

Finally, the SHAP approach (Lundberg & Lee, 2017) was implemented to interpret the results. Shapley additive explanation (also known as SHAP) is a game-centric approach gaining traction in interpreting DL models. The SHAP summary plot helps to explain the contribution of each input feature to the final simulated/predicted value through two major aspects: i) the relative importance of each variable through the magnitude of the effect, wherein a higher mean of absolute SHAP value means a higher influence and ii) the direction of the influence, whether it is a positive or negative relationship, where a higher number of points on the right side (indicated in red) shows positive relationships, and vice versa.

3.5 Software used

Deep-learning models were built using TensorFlow ((Abadi et al., 2016)) and Keras ((Chollet, 2015)). Scikit learn was used as the machine-learning framework(Pedregosa et al., 2011). All figures were prepared using Matplotlib (Hunter, 2007), pandas (McKinney, 2010), and NumPy (Van Der Walt et al., 2011). Bayesian optimization was performed using the Optuna software(Akiba et al., 2019). All this work was conducted in Python version 3.8.13, using a Dell workstation with an NVIDIA Quadro RTX 5000 GPU and 128GB RAM.

4. Performance and interpretability of the developed models

4.1 On the performances of standalone and MODWT-assisted models for GWL simulations

All MODWT-assisted and standalone models using different inputs were tested to simulate the GWLs of three types of variability. Figures 6, 7, and 8 present the comparison between observed and simulated GWL for each variability type (i.e., inertial, mixed, and annual) for different models and input types; the yellow-shaded region represents the 95% confidence interval obtained from simulations with different initializations (as explained in section 3.3). The red line shows the mean of these different simulations, whereas the black line represents the observed GWL. In addition, Tables 2 and 3 show a comparison summary of the performance evaluation metrics (i.e., MAE, RMSE, and R2) on the test set for each of the three DL methods tested (i.e., GRU, BiLSTM, and LSTM) for each variability type. Optimal hyperparameters obtained for each of the tests are shown in Appendix D.

From the overall results, the following key aspects are evident. For the inertial type, in simulations using precipitation and air temperature as input, the performance of the MODWT-assisted models ($1.52 < RMSE < 1.80$; Figures 6d-6f) is substantially improved compared to that of the standalone models ($2.46 < RMSE < 2.93$; Figures 6a-6c). In addition, the MODWT-GRU model (RMSE = 1.52; Figure 6d) outperformed the standalone GRU (RMSE = 2.93; Figure 6a). The standalone GRU performed similarly in both cases, i.e., with EP (Figure 6g) and precipitation and air temperature as input, with RMSEs of 2.89 and 2.93, respectively. In contrast, improvement in MODWT-assisted models was only apparent with precipitation and air temperature as input. So, in the case of the inertial type of GWLs, models with raw data (i.e., precipitation and air

temperature) as input (Figures 6a-6f) outperformed models with processed data (i.e., effective precipitation) (Figures 6g-6l).

For the annual type, both the standalone (1.72<RMSE<1.92; Figures 7g-7i) and MODWT-assisted models (1.69<RMSE<1.86; Figures 7j-7l) with EP as input always performed better than those with precipitation and air temperature as input (Figures 7a-7f). However, all these models resulted in better simulations, with the differences among these models with EP being very small. On the other hand, standalone models (2.30<RMSE<4.28; Figures 7a-7c) with precipitation and air temperature as input yielded less accurate results; the MODWT-assisted models (2.06<RMSE<2.44; Figures 7d-7f) improved the results but were not better than those obtained with EP as input.

For the mixed type, such as the annual, models with EP as input performed well; however, there were improvements in MODWT-assisted models (1.19<RMSE<1.51; Figures 8j-8l) with EP as input as against standalone models (1.37<RMSE<1.76; Figures 8g-8i) with EP as input. However, MODWT-assisted models (1.41<RMSE<1.59; Figures 8d-8f) with precipitation and air temperature gave comparable results with standalone models (1.50<RMSE<1.68; Figures 8a-8c) with little to no improvement.

To summarize, the improvement in the inertial type of GWL simulations using precipitation and air temperature in MODWT-assisted models was consistently high against standalone models, with an improvement in mean RMSE ranging between 26% and 48%, as shown in Table 3. While improvement varied from 10% to 47% for the annual type, it was much lowerfor the the mixed type, varying from 0.7% to 6%. On the other hand, Using EP, MODWT-assisted models improved consistently only with annual (8%-32%) and is very minor in mixed type (2%-4%). Assuming that we can process data (i.e., EP), the best combination, along with metrics, is shown in Table 4. When only precipitation and air temperature are available, the best combination, along with metrics, is shown in Table 5. Table 6 shows the best model combination for each

GWL variability. This table shows that GRU combined with MODWT decomposed inputs (precipitation and air temperature) leads to the best results for the inertial type. Conversely, BiLSTM combined with MODWT with effective precipitation was superior for the annual type. Finally, GRU combined with MODWT and effective precipitation gave better results for mixed type. But overall, GRU seemed to perform better than other models.

It is noticeable that even though effective precipitation considers both precipitation and air temperature in its computation, it did not appear capable of capturing the low-frequency variability even after MODWT decomposition. Instead, precipitation and air temperature inputs combined with MODWT-based input pre-processing seemed necessary to achieve the best simulations. Although the results indicate that MODWT-assisted models perform well, we did not find any consistent way to select the most appropriate filter length among those used in our study.

4.2. Towards a deeper investigation of what and how the models learn

Figure 9a shows the simple summary plot of standalone models with precipitation and air temperature. Figure 9b shows the SHAP summary plot for the inertial GWL type, indicating the order of importance of each decomposed component. Here, P1 to P5 represent the MODWT components of precipitation, i.e., P1 to P4 are wavelet coefficients, with P1 and P4 being the highest and lowest-frequency coefficients, respectively, while P5 represents the retained scaling coefficient. The same applies to the air temperature components T1–T5. Using the terminology defined earlier, SHAP summary plots were generated for all three GWL variability types, and three DL models were tested, as shown in Figure 10. The SHAP summary plots show that the order of importance of different features was the same for LSTM and GRU but was slightly different for the bidirectional LSTM. This might be attributed to the basic difference in the architectures of LSTM, GRU, and BiLSTM, i.e., in LSTM and GRU, information flows in only one direction.

In contrast, in bidirectional LSTM, information flows in both directions. For example, In mixed type, SHAP results showed that only BiLSTM could capture the low-frequency variability in precipitation (P5) as a major contributor to the simulations. Simulation results showed that, in this case, BiLSTM was the most efficient model (RMSE=1.41; Figure 8f).

The SHAP plots show that the inertial type always had low-frequency precipitation components as major contributors. We note that the difference in performance among the MODWT-assisted DL models is relatively small, indicating that retrieving useful information plays a more important role than the model itself. Our results (Figure 10) show that the models mainly use the air temperature input to explain/represent the annual periodic variability within GWL; the air temperature is mainly dominated by annual cyclicity. This is confirmed by the SHAP analysis, which clearly shows that the MODWT component (T3) corresponding to the annual time scale has the highest impact on simulating the annual GWL type. Barzegar et 2021 highlighted that LSTM is efficient in modeling seasonality, which explains why standalone models result in high accuracy in annual GWL time series but failed to perform accurate simulations in the case of low-frequency dominated (inertial) GWLs.

However, as emphasized in Table 6, the best results for the so-called inertial GWL type could be achieved using precipitation and air temperature time series as input and a MODWT-assisted model (best results obtained with GRU). For low-frequency dominated GWL, the DL models must access low-frequency information in the input variables to learn from it. Our experiments highlighted that, in such cases, pre-processing using MODWT helped access the information required. Figure 11a shows the normalized precipitation used as input and the lowest-frequency component it contains (red line). Although the variability explained by the low-frequency P5 component remains relatively low compared to that of the original precipitation signal (standard deviation of resp. 0.05 and 0.16 in Figure11a), it can be easily seen that the P5

component (displayed on its full range in Figure11b) matches very well the targeted GWL time series (Figure11c). One should notice here that in this case, although la10-based filters were used to ensure limited boundary effects, pre-processing still revealed capable of reaching the desired low-frequency information required to feed the model.

The GWL data used in this study are relatively free from anthropogenic influences; hence, pumping was not considered. Nevertheless, for further generalization, wherever possible, pumping data should also be considered. The GWL data can also be used for long-term historical reconstructions or direct downscaling using large-scale climate/atmospheric variables (e.g., NAO) and climate fields from reanalyses, which could also help study the hydrological evolution of GWL time series over the last century. In a benchmark study, Hagen et al. (2021) used machine learning models to identify drivers of atmospheric variables for the direct downscaling of streamflow and highlighted the need for such benchmark studies for DL. Similar studies focusing on interpretable DL for GWL downscaling are required to exploit the full potential of these models.

5. Concluding remarks

This study assessed the effectiveness of DL models for simulating different types of GWLs, including inertial, annual, and mixed. The DL models considered different input types, 1) precipitation and air temperature or 2) effective precipitation, and also considered the case where the inputs were pre-processed by the MODWT. The MODWT-assisted GRU seemed to perform well with all three types of GWLs, mainly when only precipitation and air temperature data were used as input. However, MODWT-assisted BiLSTM performed

slightly better for mixed type; interestingly, although improvement was very small (approx. 7% improvement in RMSE), SHAP results indicated the BiLSTM prioritized low-frequency better than that GRU in this case. When effective precipitation was used as input, both standalone and MODWT-assisted models consistently gave better results for annual and mixed types, but underperformed for inertial type (i.e., in such a case, using effective precipitation as input did not improve the simulation results). However effective precipitation relies on evapotranspiration, a complex variable by itself which may also be affected by high uncertainty owing to its assessment and computation. This is why capturing relevant information directly from source variables may be a better option. Nevertheless, the MODWT was still helpful in improving the results of all three models in almost all cases. Improvements were more substantial when using precipitation and air temperature as input, while they were much smaller when effective precipitation was used.

The whole framework presented in this study is flexible and reproducible, which means that each of the internal steps can be modified or replaced to check for further advancements, and can be used to adapt different types of GWL that are not currently considered here. For example, pre-processing steps can be replaced with other types of signal analysis and processing techniques, or interpretability with SHAP can be replaced with the local interpretable model-agnostic explanations or integrated gradient approach. In conclusion, using relevant information and pre-processing techniques, such as MODWT, helps DL models generate better GWL simulations.

The models presented in this study used only meteorological variables. However, in the current situation of growing concerns over frequent extreme events, such as heat waves and droughts, other input variables, such as climate indices or large-scale climate projections, should be considered when simulating the historical and future GWLs, thereby improving the decision-making processes. Hence, the models developed in the current study can project near long-term GWL simulations under different climate scenarios using GCM projections

as input. In addition, this study addressed only the epistemic uncertainty owing to the randomness in model weights. Future studies should also consider other sources of uncertainty, particularly in input data (Evapotranspiration model when using EP. Precipitation and temperature uncertainty associated with the reanalysis products used) using the current approach.

In light of this research, several questions can be raised that could serve as a basis for future studies. Does having additional variables (e.g., regional-scale hydrological/hydrometeorological variables, temporally static variables indicative of basins' physical properties, large-scale climate indices, or large-scale climate field variables from gridded reanalysis data) still require pre-processing to extract consistent information? Would other types of signal processing techniques (such as empirical/variational mode decomposed-based methods, successive LOESS or Savitzky-Golay smoothing of input data to extract low-frequency content) improve the simulations even further? Although approaches such as multi-basin training are being studied in rainfall-runoff modeling, they are yet to be explored in the context of GWL simulations. Future work should determine whether one global model is sufficient for learning various GWL variations across multiple sites simultaneously without developing a single model for each site. Concerning DL approaches, attention-based models are receiving increasing interest in recent years. However, further studies would then be required to assess whether such models would successfully account for low-frequency variations in GWLs without any help from signal pre-processing.

References

Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser,

L., Kudlur, M., … Zheng, X. (2016). *TensorFlow: Large-Scale Machine Learning on Heterogeneous Distributed Systems*.

Akiba, T., Sano, S., Yanase, T., Ohta, T., & Koyama, M. (2019). Optuna: A Next-generation Hyperparameter Optimization Framework. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2623–2631. https://doi.org/10.1145/3292500.3330701

Alizadeh, B., Ghaderi Bafti, A., Kamangir, H., Zhang, Y., Wright, D. B., & Franz, K. J. (2021). A Novel Attention-Based LSTM Cell Post-Processor Coupled with Bayesian Optimization for Streamow Prediction. *Journal of Hydrology*, *601*(February), 126526. https://doi.org/10.1016/j.jhydrol.2021.126526

Anderson, S., & Radić, V. (2022). Evaluation and interpretation of convolutional long short-term memory networks for regional hydrological modelling. *Hydrology and Earth System Sciences*, *26*(3), 795–825. https://doi.org/10.5194/hess-26-795-2022

Bahmani, R., & Ouarda, T. B. M. J. (2021). Groundwater level modeling with hybrid artificial intelligence techniques. *Journal of Hydrology*, *595*, 125659. https://doi.org/10.1016/j.jhydrol.2020.125659

Barzegar, R., Aalami, M. T., & Adamowski, J. (2021). Coupling a hybrid CNN-LSTM deep learning model with a Boundary Corrected Maximal Overlap Discrete Wavelet Transform for multiscale Lake water level forecasting. *Journal of Hydrology*, *598*, 126196. https://doi.org/10.1016/j.jhydrol.2021.126196

Baulon, L., Allier, D., Massei, N., Bessiere, H., Fournier, M., & Bault, V. (2022). Influence of low-frequency variability on groundwater level trends. *Journal of Hydrology*, *606*, 127436. https://doi.org/10.1016/j.jhydrol.2022.127436

Beven, K., & Young, P. (2013). A guide to good practice in modeling semantics for authors and referees. *Water Resources Research*, *49*(8), 5092–5098. https://doi.org/10.1002/wrcr.20393

Boé, J., & Habets, F. (2014). Multi-decadal river flow variations in France. *Hydrol. Earth Syst. Sci*, *18*, 691–708. https://doi.org/10.5194/hess-18-691-2014

Bonnet, R., Boucher, O., Vrac, M., & Jin, X. (2022). Sensitivity of bias adjustment methods to low-frequency internal climate variability over the reference period: an ideal model study. *Environmental Research: Climate*, *1*(1), 011001. https://doi.org/10.1088/2752-5295/ac6adc

Bowes, B., Goodall, J., Sadler, J., Morsy, M., & Behl, M. (2019). Toward Forecasting Groundwater Table in Flood Prone Coastal Cities Using Long Short-term Memory and Recurrent Neural Networks. *Earth and Space Science Open Archive*.

Cai, H., Shi, H., Liu, S., & Babovic, V. (2021). Impacts of regional characteristics on improving the accuracy of groundwater level prediction using machine learning: The case of central eastern continental United States. *Journal of Hydrology: Regional Studies*, *37*(September), 100930. https://doi.org/10.1016/j.ejrh.2021.100930

Chollet, F. (2015). *Keras*. GitHub. https://github.com/fchollet/keras

Cornish, C. R., Bretherton, C. S., & Percival, D. B. (2006). Maximal Overlap Wavelet Statistical Analysis With Application to Atmospheric Turbulence. *Boundary-Layer Meteorology 2006 119:2*, *119*(2), 339–374. https://doi.org/10.1007/S10546-005-9011-Y

Coulibaly, P., Anctil, F., Aravena, R., & Bobée, B. (2001). Artificial neural network modeling of water table depth fluctuations. *Water Resources Research*, *37*(4), 885–896. https://doi.org/10.1029/2000WR900368

Fossa, M., Dieppois, B., Massei, N., Fournier, M., Laignel, B., & Vidal, J.-P. (2021). Spatiotemporal and cross-scale interactions in hydroclimate variability: a case-study in France. *Hydrol. Earth Syst. Sci*, *25*, 5683–5702. https://doi.org/10.5194/hess-25-5683-2021

Ghazi, B., Jeihouni, E., Kouzehgar, K., & Haghighi, A. T. (2021). Assessment of probable groundwater changes under representative concentration pathway (RCP) scenarios through the wavelet–GEP model. *Environmental Earth Sciences*, *80*(12), 446. https://doi.org/10.1007/s12665-021-09746-9

Hagen, J. S., Leblois, E., Lawrence, D., Solomatine, D., & Sorteberg, A. (2021). Identifying major drivers of daily streamflow from large-scale atmospheric circulation with machine learning. *Journal of Hydrology*, *596*, 126086. https://doi.org/10.1016/j.jhydrol.2021.126086

Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *Computing in Science and Engineering*, *9*(3), 90–95. https://doi.org/10.1109/MCSE.2007.55

Kingston, D. G., Massei, N., Dieppois, B., Hannah, D. M., Hartmann, A., Lavers, D. A., & Vidal, J. P. (2020). Moving beyond the catchment scale: Value and opportunities in large-scale hydrology to understand our changing world. *Hydrological Processes*, *34*(10), 2292–2298. https://doi.org/10.1002/HYP.13729

Liang, Z., Liu, Y., Hu, H., Li, H., Ma, Y., & Khan, M. Y. A. (2021). Combined Wavelet Transform With Long Short-Term Memory Neural Network for Water Table Depth Prediction in Baoding City, North China Plain. *Frontiers in Environmental Science*, *9*. https://doi.org/10.3389/fenvs.2021.780434

Liu, Q., Gui, D., Zhang, L., Niu, J., Dai, H., Wei, G., & Hu, B. X. (2022). Simulation of regional groundwater levels in arid regions using interpretable machine learning models. *Science of The Total Environment*, *831*, 154902. https://doi.org/10.1016/J.SCITOTENV.2022.154902

Liu, W., Yu, H., Yang, L., Yin, Z., Zhu, M., & Wen, X. (2021). Deep learning-based predictive framework for groundwater level forecast in arid irrigated areas. *Water (Switzerland)*, *13*(18), 2558. https://doi.org/10.3390/w13182558

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *Advances in Neural Information Processing Systems*, *2017-Decem*, 4766–4775. https://github.com/slundberg/shap

Martinsen, G., Bessiere, H., Caballero, Y., Koch, J., Collados-Lara, A. J., Mansour, M., Sallasmaa, O., Pulido-Velazquez, D., Williams, N. H., Zaadnoordijk, W. J., & Stisen, S. (2022). Developing a pan-European high-resolution groundwater recharge map – Combining satellite data and national survey data using machine learning. *Science of the Total Environment*, *822*. https://doi.org/10.1016/j.scitotenv.2022.153464

Massei, N., Dieppois, B., Hannah, D. M., Lavers, D. A., Fossa, M., Laignel, B., & Debret, M. (2017). Multi-time-scale hydroclimate dynamics of a regional watershed and links to large-scale atmospheric circulation: Application to the Seine river catchment, France. *Journal of Hydrology*, *546*, 262–275. https://doi.org/10.1016/j.jhydrol.2017.01.008

Massei, N., & Fournier, M. (2012). Assessing the expression of large-scale climatic fluctuations in the hydrological variability of daily Seine river flow (France) between 1950 and 2008 using Hilbert-Huang Transform. *Journal of Hydrology*, *448–449*, 119–128. https://doi.org/10.1016/j.jhydrol.2012.04.052

Massei, N., Laignel, B., Deloffre, J., Mesquita, J., Motelay, A., Lafite, R., & Durand, A. (2010). Long-term hydrological changes of the Seine River flow (France) and their relation to the North Atlantic Oscillation over the period 1950-2008. *International Journal of Climatology*, *30*(14), 2146–2154. https://doi.org/10.1002/JOC.2022

Maxwell, R. M., Condon, L. E., & Kollet, S. J. (2015). A high-resolution simulation of groundwater and surface water over most of the continental US with the integrated hydrologic model ParFlow v3. *Geosci. Model Dev*, *8*, 923–937. https://doi.org/10.5194/gmd-8-923-2015

Mcdonald, M. G., Harbaugh, A. W., Clark, W. P., & Peck, D. L. (1988). *A modular three-dimensional finite-difference ground-water flow model. United States Department of the Interior*.

McKinney, W. (2010). Data Structures for Statistical Computing in Python. *Proceedings of the 9th Python in Science Conference*, *1*(Scipy), 56–61. https://doi.org/10.25080/majora-92bf1922-00a

Pedregosa, F., Michel, V., Grisel OLIVIERGRISEL, O., Blondel, M., Prettenhofer, P., Weiss, R., Vanderplas, J., Cournapeau, D., Pedregosa, F., Varoquaux, G., Gramfort, A., Thirion, B., Grisel, O., Dubourg, V., Passos, A., Brucher, M., Perrot andÉdouardand, M., Duchesnay, A., & Duchesnay EDOUARDDUCHESNAY, Fré. (2011). Scikit-learn: Machine Learning in Python Gaël Varoquaux Bertrand Thirion Vincent Dubourg Alexandre Passos PEDREGOSA, VAROQUAUX, GRAMFORT ET AL. Matthieu Perrot. In *Journal of Machine Learning Research* (Vol. 12).

Percival, D. B., & Walden, A. T. (2000). *Wavelet Methods for Time SeriesAnalysis*. Cambridge University Press. https://doi.org/10.1017/CBO9780511841040

Quilty, J., & Adamowski, J. (2018). Addressing the incorrect usage of wavelet-based hydrological and water resources forecasting models for real-world applications with best practices and a new forecasting framework. *Journal of Hydrology*, *563*(February), 336–353. https://doi.org/10.1016/j.jhydrol.2018.05.003

Rahman, A. T. M. S., Hosono, T., Quilty, J. M., Das, J., & Basak, A. (2020). Multiscale groundwater level forecasting: Coupling new machine learning approaches with wavelet transforms. *Advances in Water Resources*, *141*(April). https://doi.org/10.1016/j.advwatres.2020.103595

Rajaee, T., Ebrahimi, H., & Nourani, V. (2019). A review of the artificial intelligence methods in groundwater level modeling. *Journal of Hydrology*, *572*(December 2018), 336–351. https://doi.org/10.1016/j.jhydrol.2018.12.037

Rust, W., Holman, I., Bloomfield, J., Cuthbert, M., & Corstanje, R. (2019). Understanding the potential of climate teleconnections to project future groundwater drought. *Hydrol. Earth Syst. Sci*, *23*, 3233–3245. https://doi.org/10.5194/hess-23-3233-2019

Saeed, A., Li, C., Danish, M., Rubaiee, S., Tang, G., Gan, Z., & Ahmed, A. (2020). Hybrid bidirectional lstm model for short-term wind speed interval prediction. *IEEE Access*, *8*, 182283–182294. https://doi.org/10.1109/ACCESS.2020.3027977

Siami-Namini, S., Tavakoli, N., & Namin, A. S. (2019). The Performance of LSTM and BiLSTM in Forecasting Time Series. *Proceedings - 2019 IEEE International Conference on Big Data, Big Data 2019*, 3285–3292. https://doi.org/10.1109/BigData47090.2019.9005997

Tao, H., Majeed Hameed, M., Abdulameer Marhoon, H., Zounemat-Kermani, M., Salim, H., Sungwon, K., Oleiwi Sulaiman, S., Leong Tan, M., Sa'adi, Z., Danandeh Mehr, A., Falah Allawi, M., Abba, S. I., Mohamad Zain, J., Falah, M. W., Jamei, M., Dhanraj Bokde, N., Bayatvarkeshi, M., Al-Mukhtar, M., Kumar Bhagat, S., … Mundher Yaseen, Z. (2022). Groundwater Level Prediction using Machine Learning Models: A Comprehensive Review. *Neurocomputing*, *489*, 271–308. https://doi.org/10.1016/j.neucom.2022.03.014

Van Der Walt, S., Colbert, S. C., & Varoquaux, G. (2011). The NumPy array: A structure for efficient numerical computation. *Computing in Science and Engineering*, *13*(2), 22–30. https://doi.org/10.1109/MCSE.2011.37

Vidal, J. P., Martin, E., Franchistéguy, L., Baillon, M., & Soubeyroux, J. M. (2010). A 50-year high-resolution atmospheric reanalysis over France with the Safran system. *International Journal of Climatology*, *30*(11), 1627–1644. https://doi.org/10.1002/joc.2003

Vu, M. T., Jardani, A., Massei, N., & Fournier, M. (2021). Reconstruction of missing groundwater level data by using Long Short-Term Memory (LSTM) deep neural network. *Journal of Hydrology*, *597*(November 2020). https://doi.org/10.1016/j.jhydrol.2020.125776

Wu, C., Zhang, X., Wang, W., Lu, C., Zhang, Y., Qin, W., Tick, G. R., Liu, B., & Shu, L. (2021). Groundwater level modeling framework by combining the wavelet transform with a long short-term memory data-driven model. *Science of the Total Environment*, *783*, 146948. https://doi.org/10.1016/j.scitotenv.2021.146948

Wu, M., Feng, Q., Wen, X., Yin, Z., Yang, L., & Sheng, D. (2021). Deterministic analysis and uncertainty analysis of ensemble forecasting model based on variational mode decomposition for estimation of monthly groundwater level. *Water (Switzerland)*, *13*(2). https://doi.org/10.3390/w13020139

Wunsch, A., Liesch, T., & Broda, S. (2021). Groundwater level forecasting with artificial neural networks: A comparison of long short-term memory (LSTM), convolutional neural networks (CNNs), and non-linear autoregressive networks with exogenous input (NARX). *Hydrology and Earth System Sciences*, *25*(3), 1671–1687. https://doi.org/10.5194/hess-25-1671-2021

Yadav, B., Gupta, P. K., Patidar, N., & Himanshu, S. K. (2020). Ensemble modelling framework for groundwater level prediction in urban areas of India. *Science of the Total Environment*, *712*, 135539. https://doi.org/10.1016/j.scitotenv.2019.135539

Zhang, J., Zhu, Y., Zhang, X., Ye, M., & Yang, J. (2018). Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. *Journal of Hydrology*, *561*, 918–929. https://doi.org/10.1016/j.jhydrol.2018.04.065

Figures:

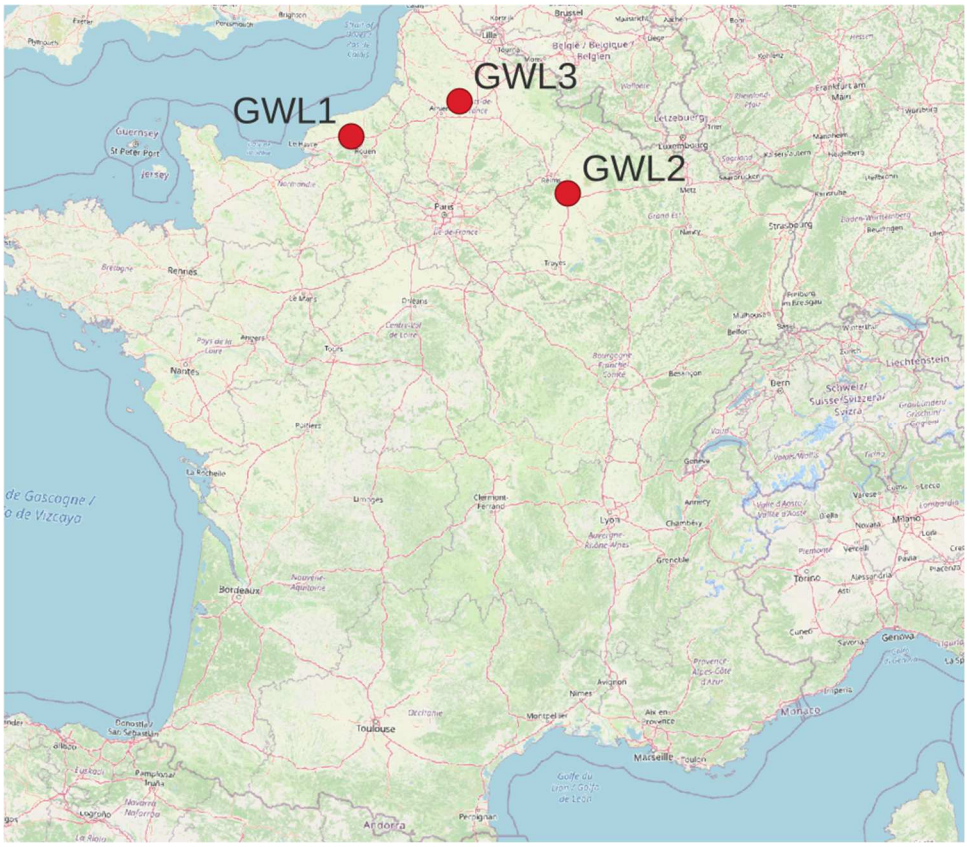


Figure 1: Study area (Red dots indicates the stations GWL1 (Inertial), GWL2 (Annual) and GWL3 (Mixed)).

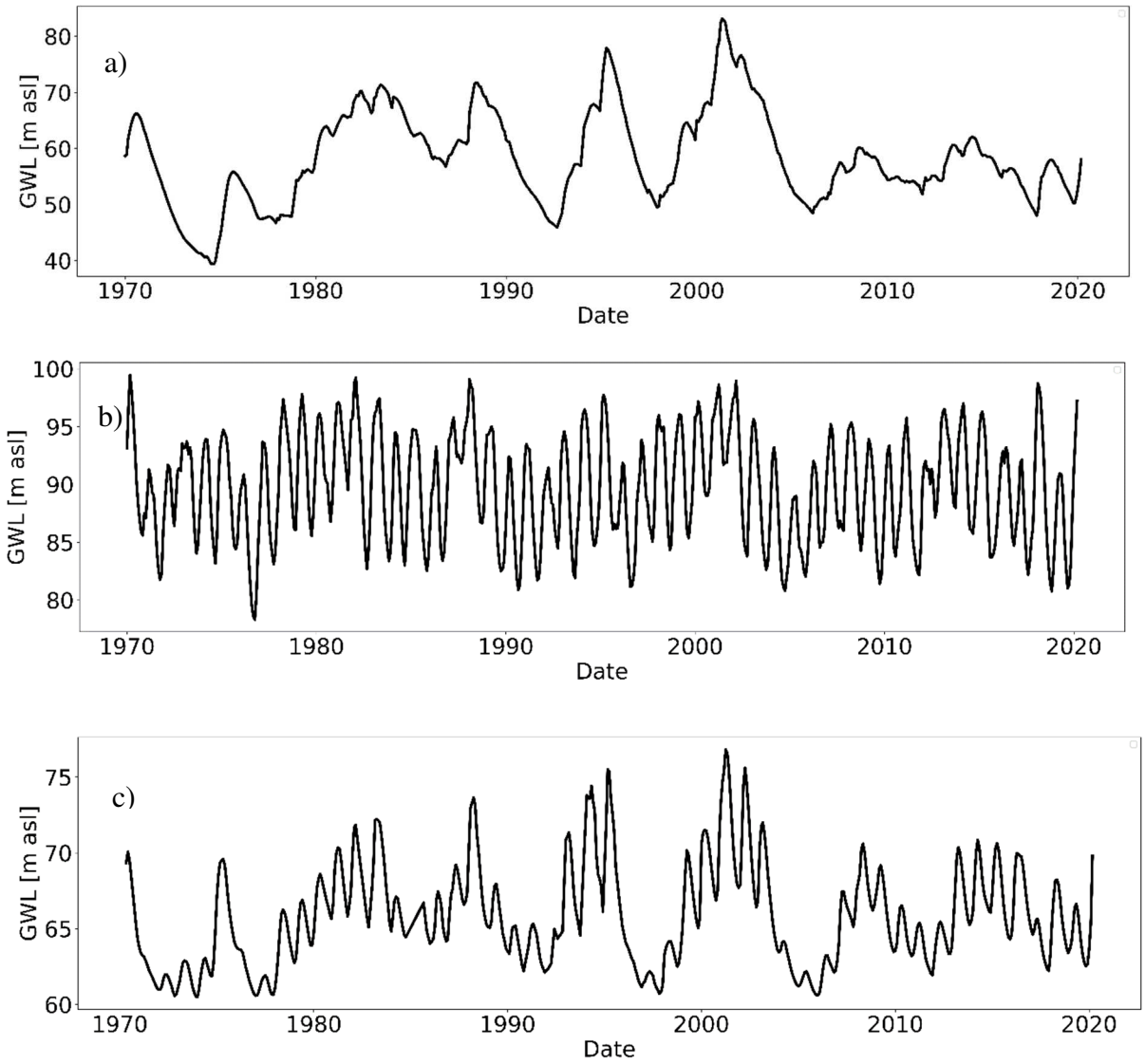


Figure 2: a) Inertial, b) annual, and c) mixed types of GWLs.

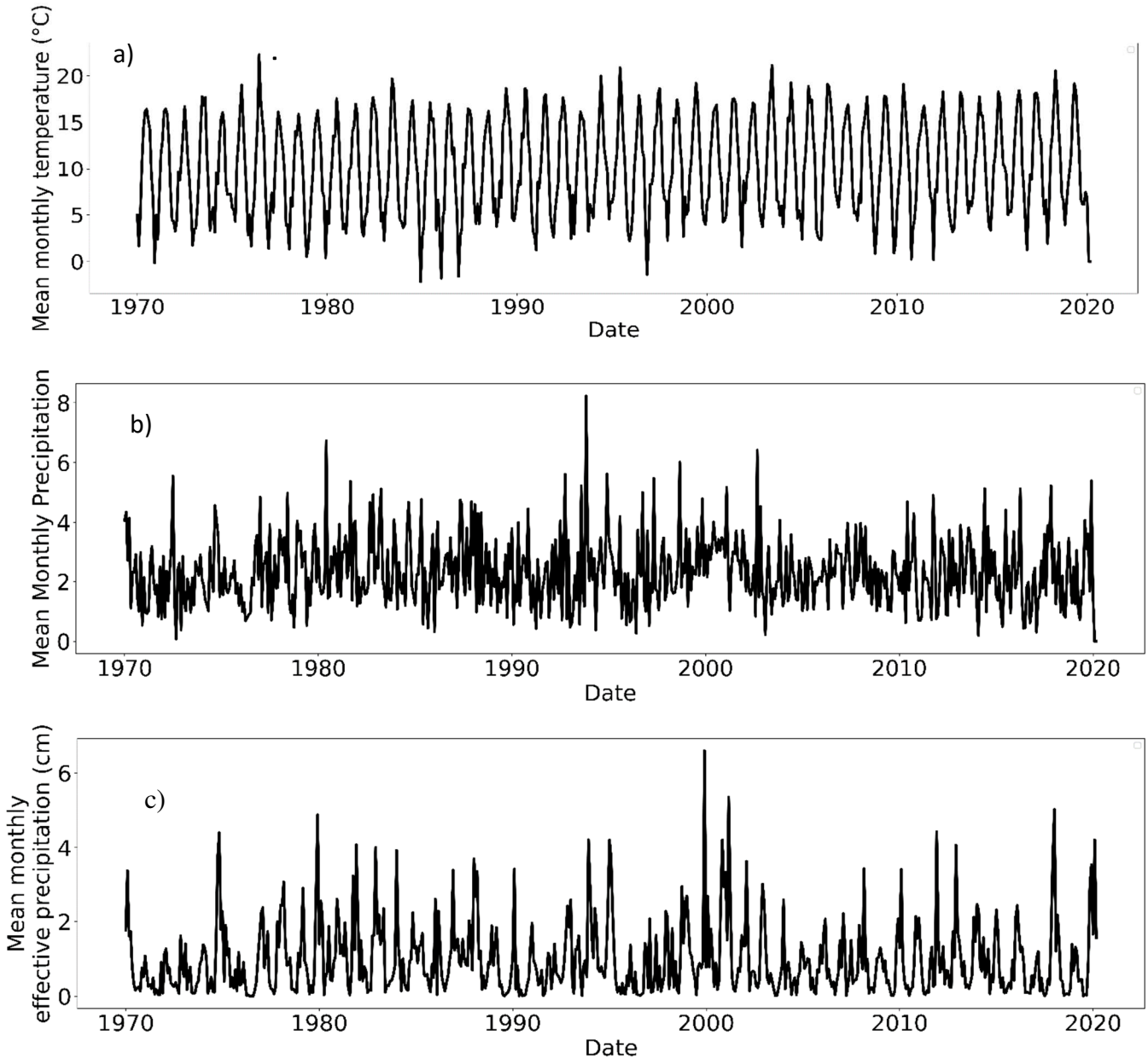


Figure 3: Time series of a) air temperature, b) precipitation, and c) effective precipitation.

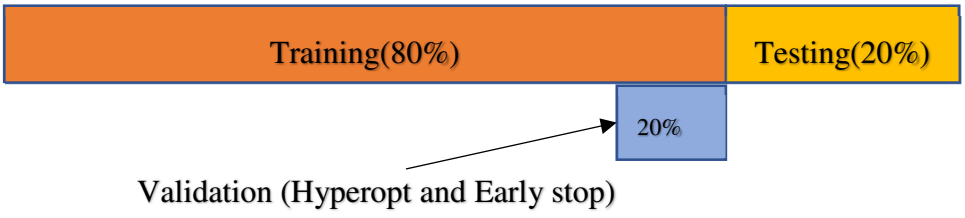


Figure 4: Data partitioning and the corresponding proportions.

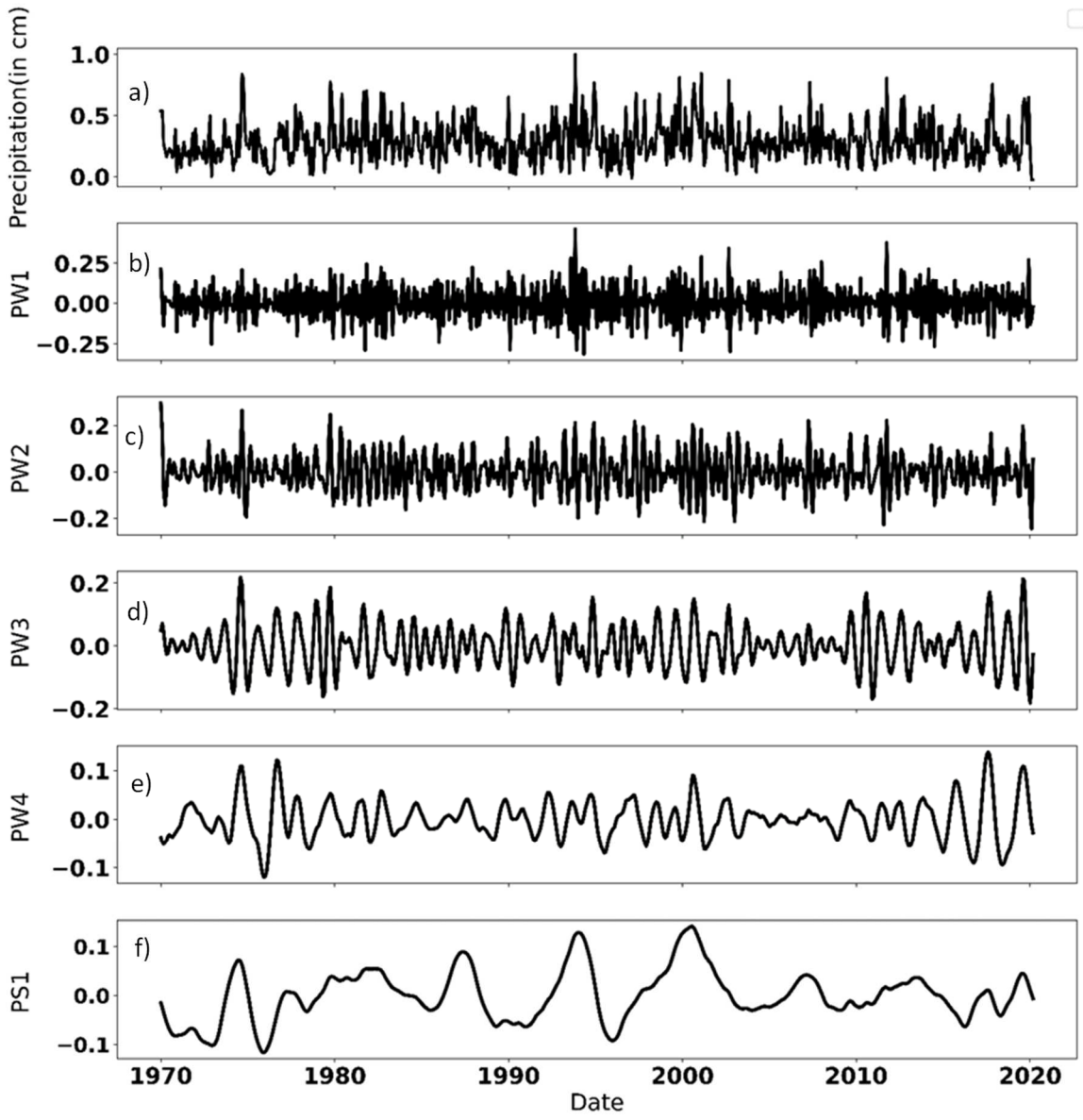


Figure 5: Decomposed components of precipitation with la10 wavelet. (a) Original signal of precipitation b-e) Wavelet coefficients (PW1 to PW4) f) Scaling coefficients (PS1)

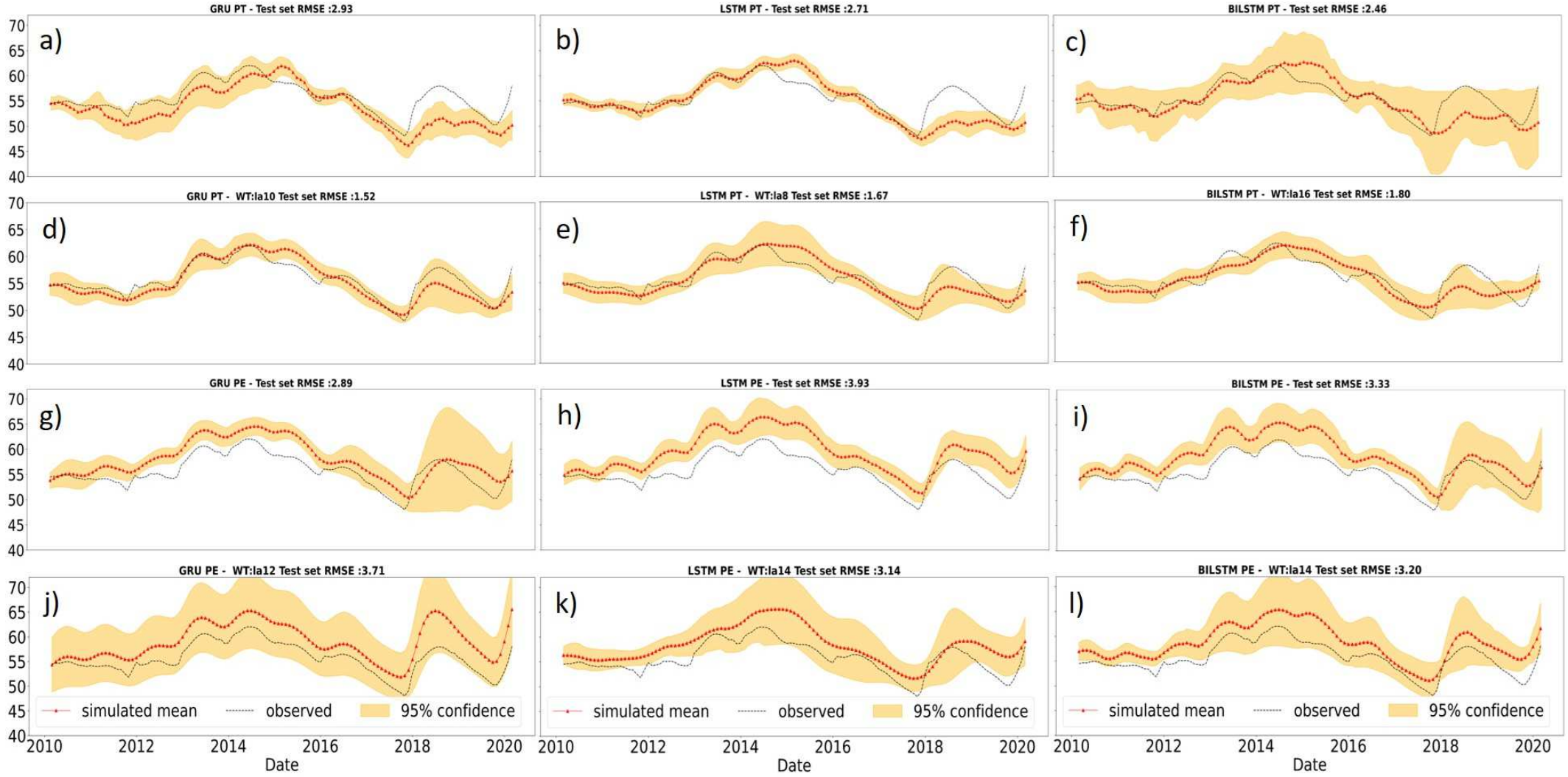


Figure 6: Results obtained for different inputs for inertial type with GRU, LSTM, and BILSTM: precipitation and air temperature (PT)(a-c), PT with wavelet transform (PTWT)(d-f), effective precipitation (PE)(g-i), and PE with wavelet transform (PEWT)(j-l). Red lines indicate the mean of the simulations and black lines indicate the observed GWL. Yellow shading represents the 95% confidence interval.

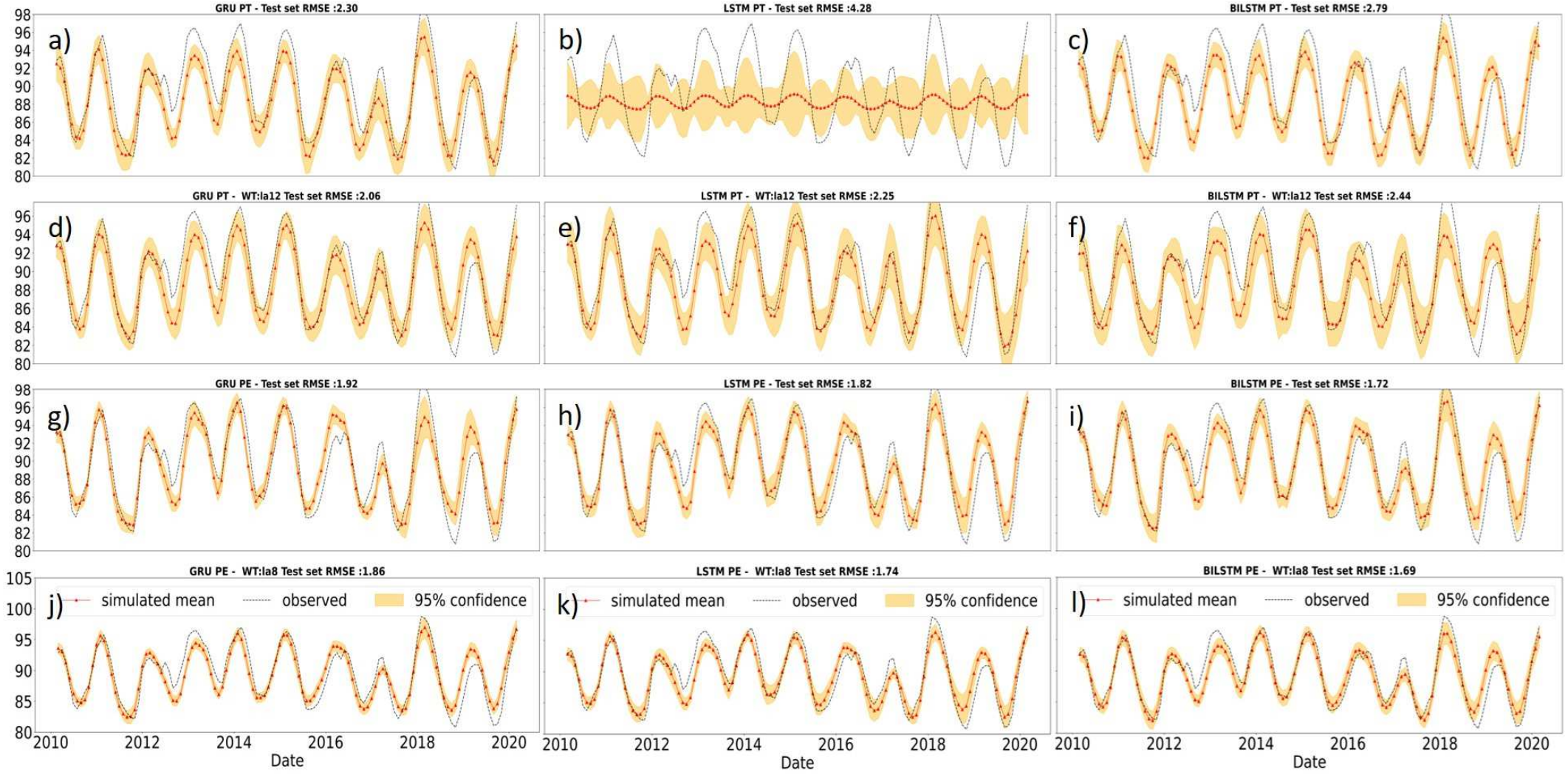


Figure 7: Results obtained for the annual type with GRU, LSTM, and BILSTM: precipitation and air temperature (PT)(a-c), PT with wavelet transform (PTWT)(d-f), effective precipitation (PE)(g-i), and PE with wavelet transform (PEWT)(j-l). Red lines indicate the mean of the simulations and black lines indicate the observed GWL. Yellow shading represents the 95% confidence interval.

30

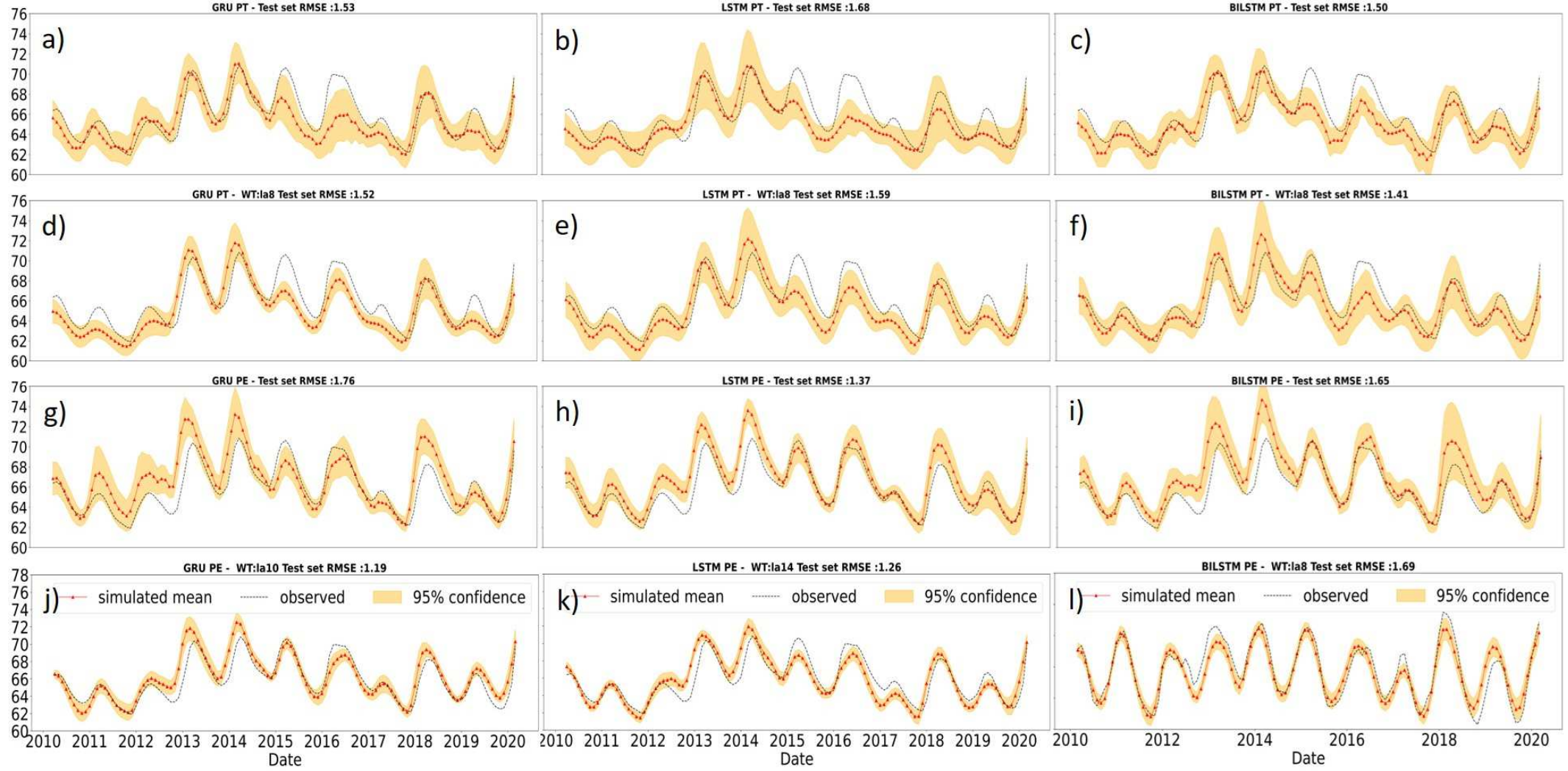


Figure 8: Results obtained for the mixed type with GRU, LSTM, and BILSTM: precipitation and air temperature (PT)(a-c), PT with wavelet transform (PTWT)(d-f), effective precipitation (PE)(g-i), and PE with wavelet transform (PEWT)(j-l). Red lines indicate the mean of the simulations and black lines indicate the observed GWL. Yellow shading represents the 95% confidence interval.

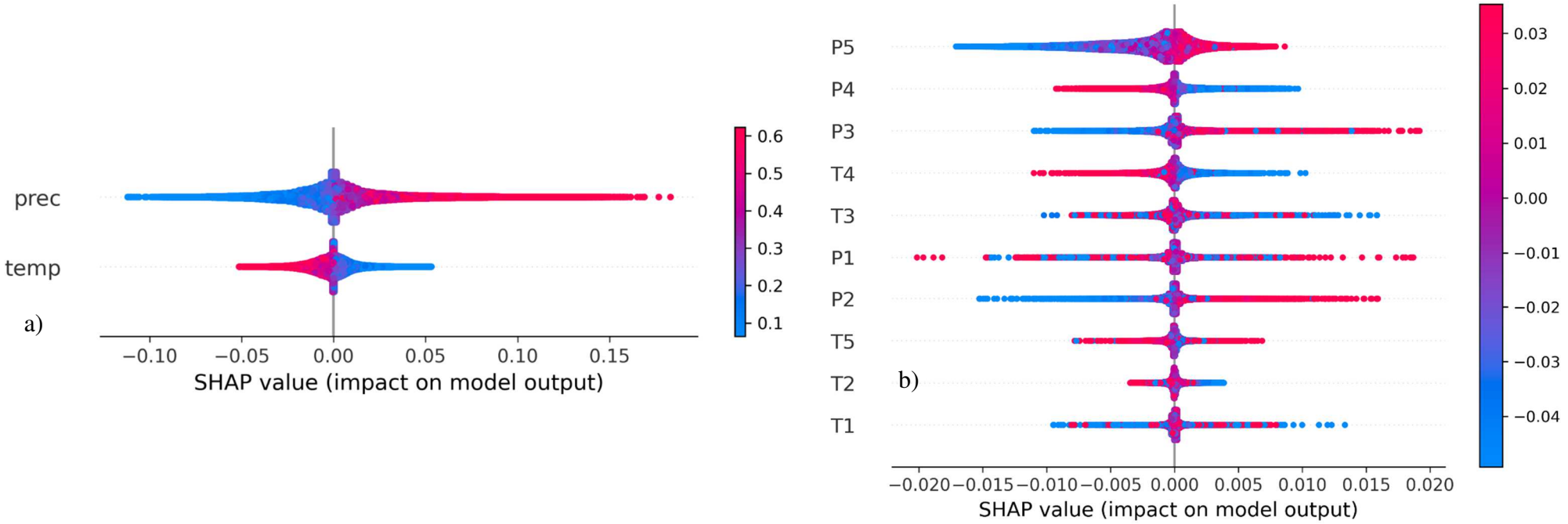


Figure 9: SHAP summary results with feature importance of each variable in the GRU: a) with precipitation and air temperature (PT) as input and b) with PT with la10 wavelet as input for the inertial type. Here P1 to P4 and T1 to T4 represents wavelet coefficients for precipitation and air temperature respectively. P5 and T5 represents scaling coefficients of precipitation and air temperature.

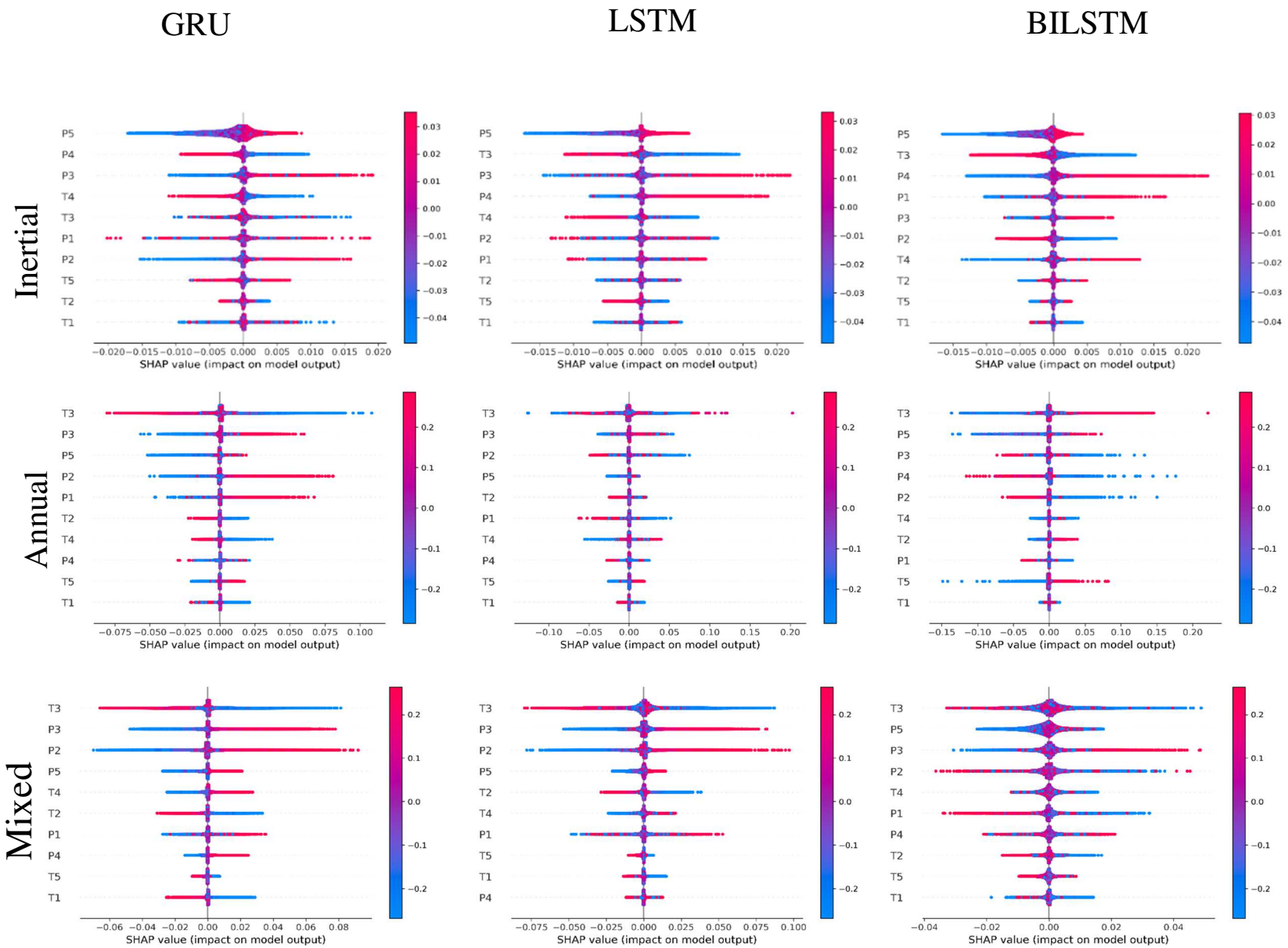
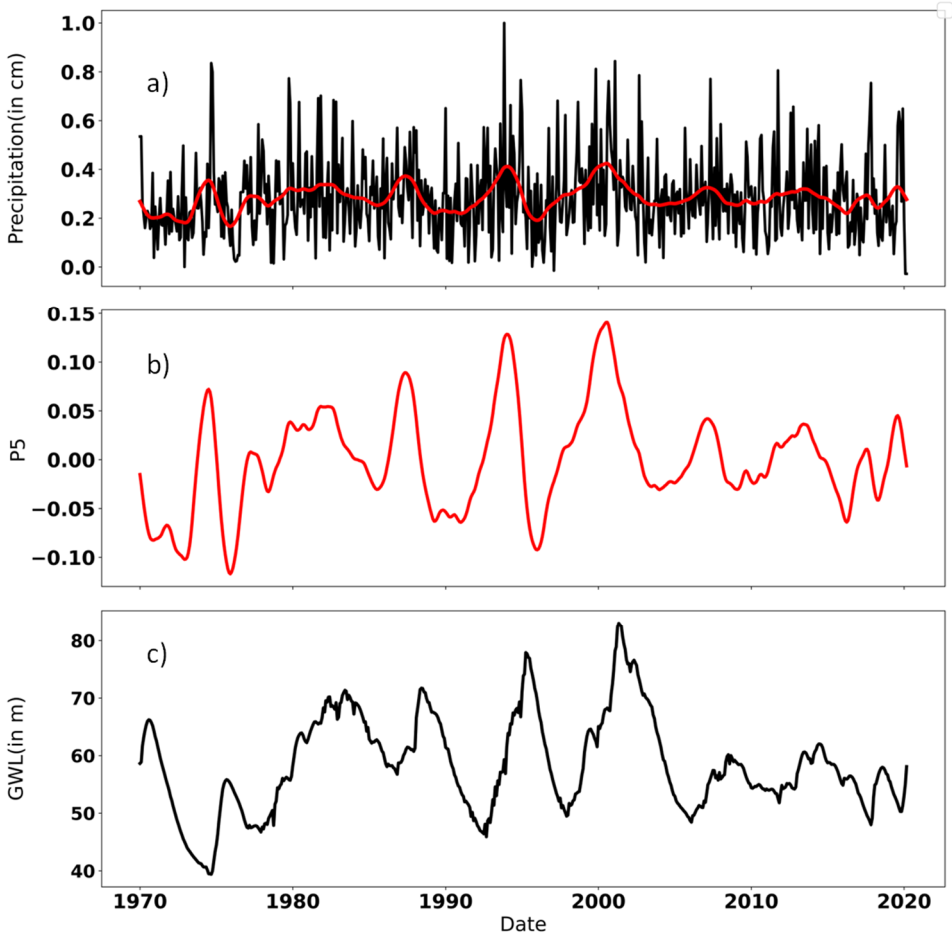


Figure 10: Comparison of SHAP summary plots for different types of GWL time series and different deep learning models. Here P1 to P4 and T1 to T4 represents wavelet coefficients for precipitation and air temperature respectively. P5 and T5 represents scaling coefficients of precipitation and air temperature.



45 *Figure 11: Comparison of a) precipitation, b) the last* MODWT component (approximation) of precipitation (P5)*, and c) the original GWL timeseries.*

Appendix A

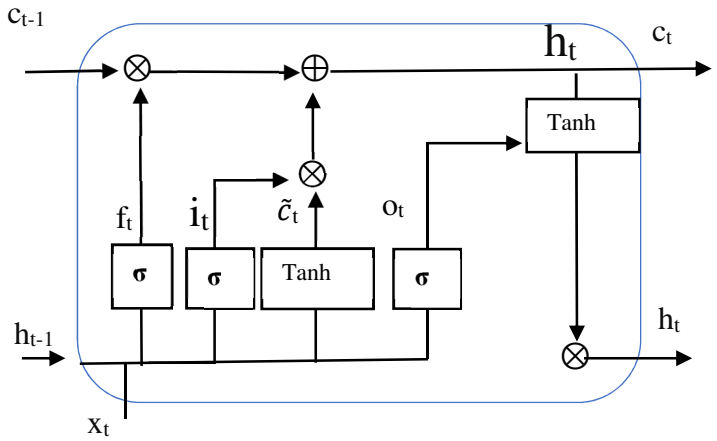


Figure 12: Schematic representation of the simple LSTM cell.

Appendix B

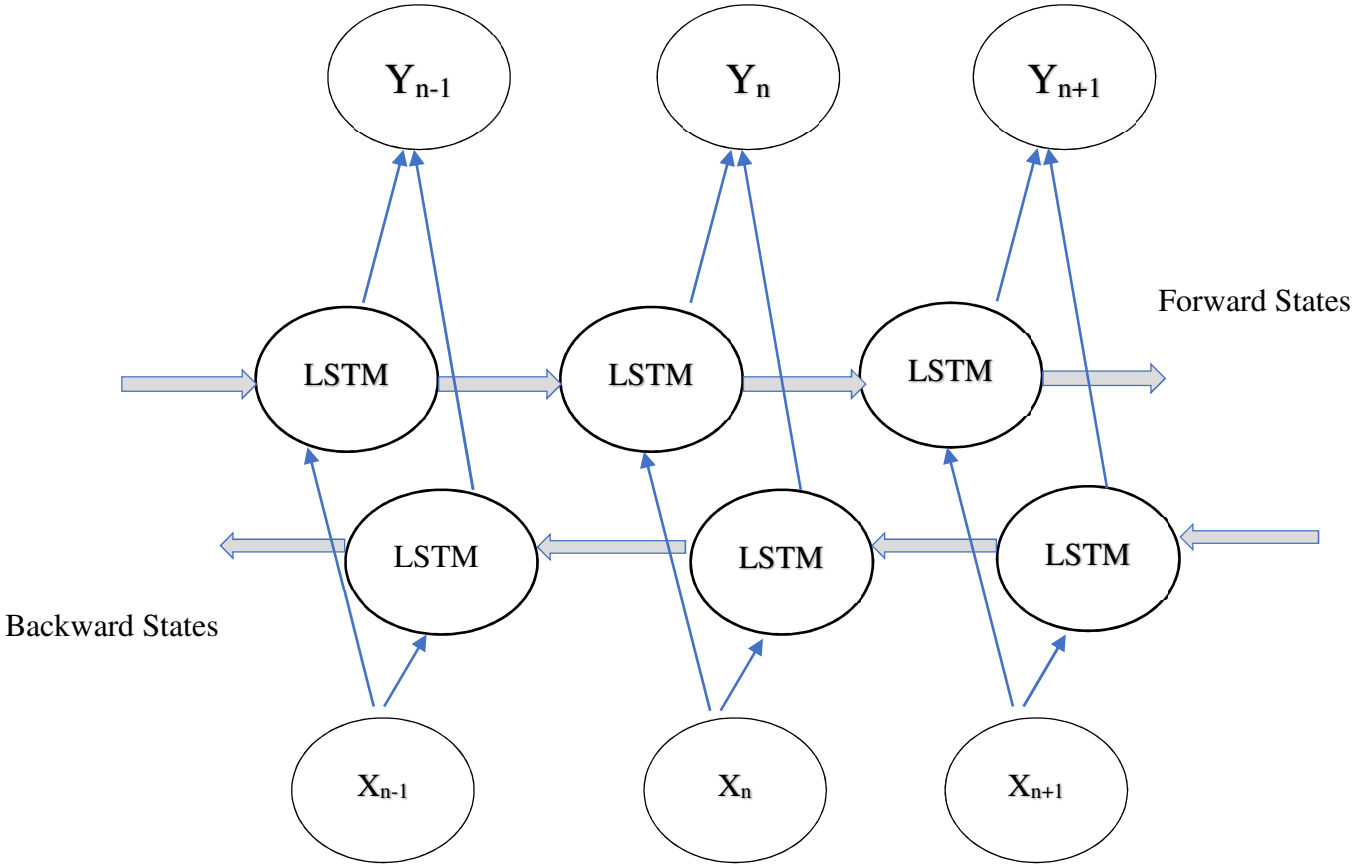


Figure 13: Schematic representation of the bidirectional LSTM (adapted and modified from Saeed et al. (2020)).

Appendix C

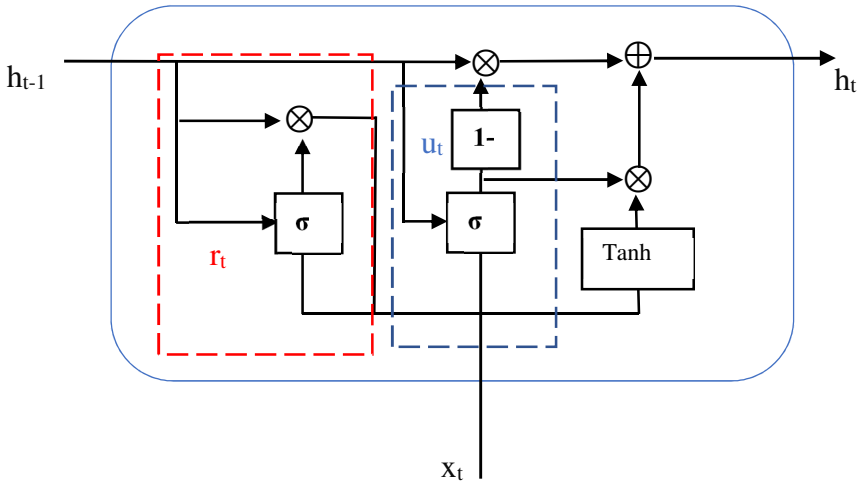


Figure 14: Schematic representation of the simple GRU cell.

TABLES:

Table 1: Hyperparameter values.

Hyperparameter	**Value considered**
Sequence length	48
Dropout	0.2
Optimizer	ADAM
Early stopping	50
Number of layers	(1,2,3,4,5,6)
Hidden neurons	(10, 20, …,100) by 10
Learning rate	(0.001,0.01) (log values)
Batch size	(16, 32, …,256) by powers of 2
Epoch	(50, 100, …,500)

Table 2: Performance metric comparison for models with effective precipitation as input.

GWL Type	Metrics	EP			EP-WT		
		GRU	LSTM	BILSTM	GRU-WT	LSTM-WT	BILSTM-WT
Inertial	R2	**0.15**	-0.57	-0.13	-0.09	-0.01	-0.04
	MAE	**2.59**	3.61	3.03	2.93	2.73	2.96
	RMSE	**2.89**	3.93	3.33	3.70	3.14	3.20
	RMSE improv. (%)**	-	-	-	-13.30	19.97	3.91
Annual	R2	0.82	0.84	0.86	0.83	0.85	**0.86**
	MAE	1.54	1.51	1.41	1.55	1.43	**1.33**
	RMSE	1.92	1.82	1.72	1.86	1.74	**1.69**
	RMSE improv. (%)	-	-	-	3.15	4.55	2.10
Mixed	R2	0.43	0.66	0.50	**0.74**	0.71	0.59
	MAE	1.36	1.01	1.19	**0.87**	0.98	1.12
	RMSE	1.76	1.37	1.65	**1.19**	1.26	1.51
	RMSE improv. (%)	-	-	-	**32.18**	8.29	8.68

** RMSE improv. (%) is improvement in RMSE in MODWT-assisted model against corresponding standalone model.

Table 3:Performance metrics comparison for models with precipitation and air temperature as input.

GWL Type	Metrics	PT			PT-WT*		
		GRU	LSTM	BILSTM	GRU-WT	LSTM-WT	BILSTM-WT
Inertial	R2	0.12	0.25	0.38	**0.76**	0.71	0.67
	MAE	2.30	1.73	1.72	**1.17**	1.28	1.44
	RMSE	2.93	2.71	2.46	**1.52**	1.67	1.80
	RMSE improv. (%)	-	-	-	48.09	38.25	26.83
Annual	R2	0.74	0.10	0.62	**0.79**	0.75	0.71
	MAE	1.89	3.62	2.28	**1.63**	1.71	1.93
	RMSE	2.30	4.28	2.79	**2.06**	2.25	2.44
	RMSE improv. (%)	-	-	-	10.58	47.39	12.54
Mixed	R2	0.57	0.48	0.59	0.58	0.54	**0.64**
	MAE	1.15	1.27	1.17	1.24	1.31	**1.06**
	RMSE	1.53	1.68	1.50	1.52	1.59	**1.41**
	RMSE improv. (%)	-	-	-	0.74	5.17	6.00

* Here, PT-WT indicates precipitation and air temperature as input along with the wavelet transform.

Table 4: Best model combinations for each type of GWL, provided that effective precipitation is available.

GWL variability	Pre-Processing	Model	Metrics
Annual	EP-WT	BILSTM	R^2=0.86, MAE=1.33, RMSE=1.69
Inertial	EP	GRU	R^2=0.15, MAE=2.59, RMSE=2.89
Mixed	EP-WT	GRU	R^2=0.74, MAE=0.87, RMSE=1.19

Table 5: Best model combinations for each type of GWL when only raw data (i.e., precipitation and air temperature) are available.

GWL variability	Pre-processing	Model	Metrics
Annual	PT-WT	GRU	R^2=0.79, MAE=1.63, RMSE=2.06
Inertial	PT-WT	GRU	R^2=0.76, MAE=1.17, RMSE=1.52
Mixed	PT-WT	BILSTM	R^2=0.64, MAE=1.06, RMSE=1.41

Table 6: Overall best models for each type of GWL variability

GWL variability	Best model	Metrics
Inertial	GRU-PT-WT	R^2=0.76, MAE=1.17, RMSE=1.52
Annual	BILSTM-EP-WT	R^2=0.86, MAE=1.33, RMSE=1.69
Mixed	GRU-EP-WT	R^2=0.74, MAE=0.87, RMSE=1.19

Appendix D

Table 7: Optimal parameters for standard and MODWT-assisted GRU Models for three types of GWLs (Mixed, inertial and annual) and two different input types (Effective precipitation (PE) or Precipitation and air temperature (PT))

GRU	PT	PTWT	PE	PEWT
Mixed	{'learning_rate': 0.00686772761630292, 'optimizer': 'adam', 'epochs': 300, 'batch_size': 64, 'n_layers': 2, 'n_units_l0': 80, 'dropout_l0': 0.2, 'n_units_l1': 40, 'dropout_l1': 0.2}	{'learning_rate': 0.003990606939187562, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 144, 'n_layers': 6, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 10, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2, 'n_units_l3': 20, 'dropout_l3': 0.2, 'n_units_l4': 90, 'dropout_l4': 0.2, 'n_units_l5': 10, 'dropout_l5': 0.2}	{'learning_rate': 0.006495625254590144, 'optimizer': 'adam', 'epochs': 350, 'batch_size': 192, 'n_layers': 2, 'n_units_l0': 100, 'dropout_l0': 0.2, 'n_units_l1': 80, 'dropout_l1': 0.2	{'learning_rate': 0.0012401104710986497, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 48, 'n_layers': 3, 'n_units_l0': 100, 'dropout_l0': 0.2, 'n_units_l1': 100, 'dropout_l1': 0.2, 'n_units_l2': 50, 'dropout_l2': 0.2}
Inertial	{'learning_rate': 0.0021546271126850858, 'optimizer': 'adam', 'epochs': 400, 'batch_size': 16, 'n_layers': 3, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 70, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2}	{'learning_rate': 0.008783758135456727, 'optimizer': 'adam', 'epochs': 250, 'batch_size': 32, 'n_layers': 3, 'n_units_l0': 30, 'dropout_l0': 0.2, 'n_units_l1': 100, 'dropout_l1': 0.2, 'n_units_l2': 10, 'dropout_l2': 0.2}	{'learning_rate': 0.0011098849592139203, 'optimizer': 'adam', 'epochs': 350, 'batch_size': 32, 'n_layers': 6, 'n_units_l0': 60, 'dropout_l0': 0.2, 'n_units_l1': 70, 'dropout_l1': 0.2, 'n_units_l2': 90, 'dropout_l2': 0.2, 'n_units_l3': 70, 'dropout_l3': 0.2, 'n_units_l4': 50,	{'learning_rate': 0.003487742134799628, 'optimizer': 'adam', 'epochs': 100, 'batch_size': 96, 'n_layers': 5, 'n_units_l0': 50, 'dropout_l0': 0.2, 'n_units_l1': 50, 'dropout_l1': 0.2, 'n_units_l2': 70, 'dropout_l2': 0.2, 'n_units_l3': 60, 'dropout_l3': 0.2, 'n_units_l4': 70, 'dropout_l4': 0.2}

			'dropout_l4': 0.2, 'n_units_l5': 100, 'dropout_l5': 0.2}	
Annual	{'learning_rate': 0.004281617773380174, 'optimizer': 'adam', 'epochs': 500, 'batch_size': 80, 'n_layers': 5, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 30, 'dropout_l1': 0.2, 'n_units_l2': 50, 'dropout_l2': 0.2, 'n_units_l3': 20, 'dropout_l3': 0.2, 'n_units_l4': 80, 'dropout_l4': 0.2}	{'learning_rate': 0.009240879941149954, 'optimizer': 'adam', 'epochs': 200, 'batch_size': 32, 'n_layers': 5, 'n_units_l0': 60, 'dropout_l0': 0.2, 'n_units_l1': 50, 'dropout_l1': 0.2, 'n_units_l2': 10, 'dropout_l2': 0.2, 'n_units_l3': 50, 'dropout_l3': 0.2, 'n_units_l4': 80, 'dropout_l4': 0.2}	{'learning_rate': 0.003990606939187562, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 144, 'n_layers': 6, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 10, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2, 'n_units_l3': 20, 'dropout_l3': 0.2, 'n_units_l4': 90, 'dropout_l4': 0.2, 'n_units_l5': 10, 'dropout_l5': 0.2}	{'learning_rate': 0.003966724265631709, 'optimizer': 'adam', 'epochs': 400, 'batch_size': 80, 'n_layers': 3, 'n_units_l0': 70, 'dropout_l0': 0.2, 'n_units_l1': 50, 'dropout_l1': 0.2, 'n_units_l2': 10, 'dropout_l2': 0.2}

Table 8:Optimal parameters for standard and MODWT-assisted LSTM Models for three types of GWLs (Mixed, inertial and annual) and two different input types (Effective precipitation (PE) or Precipitation and air temperature (PT))

LSTM	PT	PTWT	PE	PEWT
Mixed	{'learning_rate': 0.003990606939187562, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 144, 'n_layers': 6, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 10, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2,	{'learning_rate': 0.001642602063722058, 'optimizer': 'adam', 'epochs': 500, 'batch_size': 16, 'n_layers': 3, 'n_units_l0': 100, 'dropout_l0': 0.2, 'n_units_l1': 100, 'dropout_l1': 0.2, 'n_units_l2': 30, 'dropout_l2': 0.2}	{'learning_rate': 0.001724379089398 7354, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 16, 'n_layers': 4, 'n_units_l0': 30, 'dropout_l0': 0.2, 'n_units_l1': 70, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2,	{'learning_rate': 0.0011286448692694375, 'optimizer': 'adam', 'epochs': 400, 'batch_size': 16, 'n_layers': 2, 'n_units_l0': 30, 'dropout_l0': 0.2, 'n_units_l1': 70, 'dropout_l1': 0.2}

	'n_units_l3': 20, 'dropout_l3': 0.2, 'n_units_l4': 90, 'dropout_l4': 0.2, 'n_units_l5': 10, 'dropout_l5': 0.2}		'n_units_l3': 10, 'dropout_l3': 0.2}	
Inertial	{'learning_rate': 0.0019279681079 150224, 'optimizer': 'adam', 'epochs': 250, 'batch_size': 80, 'n_layers': 4, 'n_units_l0': 70, 'dropout_l0': 0.2, 'n_units_l1': 80, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2, 'n_units_l3': 10, 'dropout_l3': 0.2}	{'learning_rate': 0.0036725622777 068682, 'optimizer': 'adam', 'epochs': 500, 'batch_size': 32, 'n_layers': 5, 'n_units_l0': 60, 'dropout_l0': 0.2, 'n_units_l1': 70, 'dropout_l1': 0.2, 'n_units_l2': 10, 'dropout_l2': 0.2, 'n_units_l3': 60, 'dropout_l3': 0.2, 'n_units_l4': 10, 'dropout_l4': 0.2}	{'learning_rate': 0.0062532462360 7946, 'optimizer': 'adam', 'epochs': 350, 'batch_size': 80, 'n_layers': 3, 'n_units_l0': 30, 'dropout_l0': 0.2, 'n_units_l1': 40, 'dropout_l1': 0.2, 'n_units_l2': 20, 'dropout_l2': 0.2}	{'learning_rate': 0.003990606939187 562, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 144, 'n_layers': 6, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 10, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2, 'n_units_l3': 20, 'dropout_l3': 0.2, 'n_units_l4': 90, 'dropout_l4': 0.2, 'n_units_l5': 10, 'dropout_l5': 0.2}
Annual	{'learning_rate': 0.0097372366661 14474, 'optimizer': 'adam', 'epochs': 500, 'batch_size': 64, 'n_layers': 5, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 70, 'dropout_l1': 0.2, 'n_units_l2': 30, 'dropout_l2': 0.2, 'n_units_l3': 100, 'dropout_l3': 0.2, 'n_units_l4': 100, 'dropout_l4': 0.2}	{'learning_rate': 0.0064956252545 90144, 'optimizer': 'adam', 'epochs': 350, 'batch_size': 192, 'n_layers': 2, 'n_units_l0': 100, 'dropout_l0': 0.2, 'n_units_l1': 80, 'dropout_l1': 0.2}	{'learning_rate': 0.0043121677192 285894, 'optimizer': 'adam', 'epochs': 400, 'batch_size': 16, 'n_layers': 2, 'n_units_l0': 50, 'dropout_l0': 0.2, 'n_units_l1': 60, 'dropout_l1': 0.2}	{'learning_rate': 0.001562653587605 288, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 16, 'n_layers': 3, 'n_units_l0': 90, 'dropout_l0': 0.2, 'n_units_l1': 100, 'dropout_l1': 0.2, 'n_units_l2': 50, 'dropout_l2': 0.2}

Table 9: Optimal parameters for standard and MODWT-assisted BiLSTM Models for three types of GWLs (Mixed, inertial and annual) and two different input types (Effective precipitation (PE) or Precipitation and air temperature (PT))

BiLSTM	PT	PTWT	PE	PEWT
Mixed	{'learning_rate': 0.004212772944331677, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 80, 'n_layers': 3, 'n_units_l0': 70, 'dropout_l0': 0.2, 'n_units_l1': 80, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2}	{'learning_rate': 0.003430976690105094 7, 'optimizer': 'adam', 'epochs': 350, 'batch_size': 16, 'n_layers': 4, 'n_units_l0': 70, 'dropout_l0': 0.2, 'n_units_l1': 30, 'dropout_l1': 0.2, 'n_units_l2': 80, 'dropout_l2': 0.2, 'n_units_l3': 50, 'dropout_l3': 0.2}	{'learning_rate': 0.002966383208485377, 'optimizer': 'adam', 'epochs': 200, 'batch_size': 16, 'n_layers': 3, 'n_units_l0': 80, 'dropout_l0': 0.2, 'n_units_l1': 90, 'dropout_l1': 0.2, 'n_units_l2': 70, 'dropout_l2': 0.2}	{'learning_rate': 0.00218172101811288, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 16, 'n_layers': 2, 'n_units_l0': 100, 'dropout_l0': 0.2, 'n_units_l1': 100, 'dropout_l1': 0.2}
Inertial	{'learning_rate': 0.003990606939187562, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 144, 'n_layers': 6, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 10, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2, 'n_units_l3': 20, 'dropout_l3': 0.2, 'n_units_l4': 90, 'dropout_l4': 0.2, 'n_units_l5': 10, 'dropout_l5': 0.2}	{'learning_rate': 0.001314944556129716, 'optimizer': 'adam', 'epochs': 500, 'batch_size': 16, 'n_layers': 6, 'n_units_l0': 60, 'dropout_l0': 0.2, 'n_units_l1': 70, 'dropout_l1': 0.2, 'n_units_l2': 20, 'dropout_l2': 0.2, 'n_units_l3': 20, 'dropout_l3': 0.2, 'n_units_l4': 30, 'dropout_l4': 0.2, 'n_units_l5': 30, 'dropout_l5': 0.2}	{'learning_rate': 0.002141506081144842, 'optimizer': 'adam', 'epochs': 400, 'batch_size': 16, 'n_layers': 3, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 70, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2}	{'learning_rate': 0.003990606939187562, 'optimizer': 'adam', 'epochs': 450, 'batch_size': 144, 'n_layers': 6, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 10, 'dropout_l1': 0.2, 'n_units_l2': 40, 'dropout_l2': 0.2, 'n_units_l3': 20, 'dropout_l3': 0.2, 'n_units_l4': 90, 'dropout_l4': 0.2, 'n_units_l5': 10, 'dropout_l5': 0.2}
Annual	{'learning_rate': 0.004409980567805262, 'optimizer': 'adam',	{'learning_rate': 0.008515913194885923, 'optimizer': 'adam',	{'learning_rate': 0.004849787847878661, 'optimizer': 'adam',	{'learning_rate': 0.002051217063543 3725, 'optimizer': 'adam', 'epochs': 400,

	'epochs': 250, 'batch_size': 208, 'n_layers': 2, 'n_units_l0': 80, 'dropout_l0': 0.2, 'n_units_l1': 70, 'dropout_l1': 0.2}	'epochs': 200, 'batch_size': 16, 'n_layers': 2, 'n_units_l0': 100, 'dropout_l0': 0.2, 'n_units_l1': 40, 'dropout_l1': 0.2}	'epochs': 350, 'batch_size': 48, 'n_layers': 3, 'n_units_l0': 10, 'dropout_l0': 0.2, 'n_units_l1': 60, 'dropout_l1': 0.2, 'n_units_l2': 60, 'dropout_l2': 0.2}	'batch_size': 16, 'n_layers': 3, 'n_units_l0': 40, 'dropout_l0': 0.2, 'n_units_l1': 70, 'dropout_l1': 0.2, 'n_units_l2': 20, 'dropout_l2': 0.2}

Graphical Abstract