https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

Combining Recurrent Neural Networks with Variational Mode Decomposition and Multifractals to Predict Rainfall Time Series

Hai Zhou, Daniel Schertzer, and Ioulia Tchiguirinskaia
Hydrology Meteorology & Complexity (HM&Co), Ecole des Ponts ParisTech, Champs-sur-Marne, France
Correspondence: Hai Zhou (hai.zhou@enpc.fr)

Abstract. Rainfall time series prediction is essential for monitoring urban hydrological systems, but it is challenging and complex due to the extreme variability of rainfall. A hybrid deep learning model (VMD-RNN) is used in order to improve prediction performance. In this study, variational mode decomposition (VMD) is first applied to decompose the original rainfall time series into several sub-sequences according to the frequency domain, where the number of decomposed sub-sequences is determined by power spectral density (PSD) analysis. To prevent the disclosure of forthcoming data, non-training time series are sequentially appended for generating the decomposed testing samples. Following that, different recurrent neural network (RNN) variant models are used to predict individual sub-sequences and the final prediction is reconstructed by summing the prediction results of sub-sequences. These RNN-variants are long short-term memory (LSTM), gated recurrent unit (GRU), bidirectional LSTM (BiLSTM) and bidirectional GRU (BiGRU), which are optimal for sequence prediction. In addition to three common evaluation criteria, mean absolute error (MAE), root mean square error (RMSE) and mean absolute percentage error (MAPE), the framework of universal multifractals (UM) is also introduced to assess the performance of predictions, which enables the extreme variability of predicted rainfall time series to be characterized. The study employs two rainfall time series with daily and hourly resolutions, respectively. The results indicate that the hybrid VMD-RNN model provides a reliable one-step-ahead prediction, with better performance in predicting high and low values than the pure LSTM model without decomposition.

1 Introduction

Prediction of rainfall time series plays an important role in monitoring urban hydrological systems and their geophysical environment. Accurate and trustworthy predictions can serve as an early warning of floods and other extreme events, as well as a guide for water resource allocation. Although predicting rainfall time series is not a novel concept, it has remained fundamentally difficult due to the extreme variability, in fact intermittency, of rainfall over a wide range of space-time scales, i.e. increasingly heavy precipitation is concentrated over smaller and smaller fractions of the space-time.

Classical forecast models are either process-driven physical models or data-driven statistical models. The former represents the most important physical processes and numerically solves the governing equations based on initial and boundary conditions (Lynch, 2008). Due to the fact that rainfall depends on a variety of land, ocean, atmospheric processes, and their complex interactions, physical models are developed based on simplifications of those processes, in particular by truncating

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

the scales and introducing rather ad-hoc parametrisations. This greatly increases their unpredictability (Bauer et al., 2015). On the contrary, data-driven models strive to establish a link between input and output data to predict time series without regard to underlying physical processes (Reichstein et al., 2019). In general, they provide a unique output therefore with no information on the uncertainty generated by the nonlinearity of the involved processes. A sort of hybrid approach has been developed using stochastic models physically based on the cascade paradigm (e.g., Schertzer and Lovejoy, 1987; Marsan et al., 1996; Schertzer and Lovejoy, 2004, 2011). This ensures that intermittency is directly taken into account, including in the generation of uncertainty.

The explosion of supercomputing and data availability offers immense potential for data-driven models to significantly contribute to prediction (Schultz et al., 2021). There are several methods available for predicting rainfall time series, including linear and nonlinear models. The traditional linear data-driven model is the autoregressive integrated moving average (ARIMA) (Chattopadhyay and Chattopadhyay, 2010), which ignores the nonlinearity of the relationship between input and output time series, leading to poor prediction ability. Because of increased data availability and computing power, various deep learning (DL) models have been proposed and applied in predicting nonlinear time series (Lara-Benítez et al., 2021).

Recurrent neural network (RNN) models are a subset of deep learning models, which have been specifically designed to solve sequential prediction problems (Elman, 1990). However, standard RNN struggles with long-term dependence and exhibits the gradient vanishing or exploding problems (Hochreiter and Schmidhuber, 1997). RNN variants, such as long short-term memory (LSTM), gated recurrent unit (GRU), bidirectional LSTM (BiLSTM) and bidirectional GRU (BiGRU), are intended to alleviate the limitations of standard RNN. These variant models have been employed in various fields (e.g., Graves et al., 2013; Cho et al., 2014; Su et al., 2020; Lin et al., 2022), including time series prediction (e.g., Ma et al., 2015; Ding et al., 2019; Gauch et al., 2021). In particular, great efforts have been devoted to predicting rainfall time series (e.g., Ni et al., 2020; Barrera-Animas et al., 2022; He et al., 2022), as shown in Table 1.

However, these pure variant models are not always capable of efficiently handling extremely nonlinear time series with several noisy components without the need for appropriate preprocessing. Decomposition is a typical preprocessing method in time series analysis, which can extract hidden information to aid in the comprehension of the complex original time series. For decomposition approaches, wavelet decomposition (Pati et al., 1993), empirical mode decomposition (EMD) (Huang et al., 1998) and variational mode decomposition (VMD) (Dragomiretskiy and Zosso, 2013) are commonly used to decompose original data. Relevant studies on time series prediction by combining decomposition technique with deep learning models are also presented in Table 1. Because wavelet decomposition is highly dependent on the choice of the mother wavelet function, its adaptability in decomposing time series is limited (Hadi and Tombul, 2018). Meanwhile, EMD suffers from boundary effects, mode mixing, and a lack of exact mathematical foundations (Devi et al., 2020). In comparison, VMD, which is theoretically sound, presents the advantage of solving the mode overlap problem. Considering the dominant characteristics of VMD in decomposing nonlinear time series and the beneficial performance of variant RNN models in predicting complex sequential problems, this study employs a hybrid model based on VMD and RNN for rainfall time series prediction.

In the VMD process, the number of generated sub-sequences has an influence on the decomposition performance (Dragomiretskiy and Zosso, 2013; Zuo et al., 2020). The occurrence of modal mixing phenomena can be attributed to an excessive number

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

of sub-sequences, while an insufficient number of sub-sequences can lead to inadequate extraction of information from the original sequences. In this study, we define the number of decomposed sub-sequences by analyzing the power spectral density (PSD) of the last sub-sequence. In order to mitigate the risk of revealing future data during the decomposition of non-training time series, a precautionary approach has been implemented. This approach differs from the direct method of decomposing the testing time series using VMD in the hydrological field (e.g., He et al., 2019; Xie et al., 2019). The non-training data is added to the training set in a sequential manner to create a new time series, and the amount of newly generated time series is equal to the number of non-training data points. The VMD technique is thereafter used to decompose the aforementioned new time series into several sub-sequences. Subsequently, the final data point of each newly generated sub-sequence is retrieved and designated as non-training data, which is then used to build validation and testing samples.

When evaluating performance in predicting time series, MAE, RMSE and MAPE criteria are widely used (e.g., Lara-Benítez et al., 2021; Wu et al., 2021). They can define the difference between predicted and observed values, but they cannot describe the variability of rainfall time series over a wide range of scales. For reasons discussed above, a commonly used technique for describing such variability is stochastic multifractals, which are based on the concept of multiplicative cascades (Schertzer and Lovejoy, 1987; Gupta and Waymire, 1993; Deidda, 2000; Schertzer et al., 2010). To be more specific, universal multifractals (UM) with only three scale-independent parameters have been extensively employed to represent the scaling variability of rainfall (e.g., Tessier et al., 1993; Schertzer et al., 1997; Harris et al., 1997; Gires et al., 2013; Yu et al., 2014). The analysis of variability using UM technology contributes to the reliability of performance evaluations.

The rest of this article is organized as follows. In section 2, the corresponding methodologies are presented in detail, including VMD, RNN variants and UM. Two rainfall time series with daily and hourly resolutions are performed by VMD-RNN in section 3. The results are discussed and analyzed in section 4. Finally, conclusions and future work are given in section 5.

[Table 1 about here.]

2 Methodology

2.1 Variational mode decomposition

The primary process of variational mode decomposition (VMD) is constructing and solving the variational problem (Dragomiretskiy and Zosso, 2013). For rainfall time series $f(t)$, the variation problem is described as identifying K sub-sequences $u_k(t)$with center frequency ω_k to minimize the sum value of the estimated bandwidth of each $u_k(t)$. The constrained condition is that the aggregation of the sub-sequences $u_k(t)$ should be equal to the original sequence $f(t)$. The constrained variational problem can be expressed as follows:

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_{k=1}^{K} \left\| \partial_t \left[\left(\delta(t) + \frac{j}{\pi t} \right) * u_k(t) \right] e^{-j\omega_k t} \right\|_2^2 \right\} \qquad \text{s.t.} \sum_{k=1}^{K} u_k = f(t) \tag{1}$$

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

where $\{u_k(t)\} = \{u_1(t), u_2(t), ..., u_K(t)\}$ and $\{\omega_k\} = \{\omega_1, \omega_2, ..., \omega_K\}$ are shorthand notations for decomposed sub-sequences and their center frequencies, respectively; $\delta(t)$ is the Dirac distribution, the symbol $*$ denotes convolution and $e^{-j\omega_k t}$ is a phasor describing the rotation of the complex signal in time, with $j^2 = -1$.

The variational problem is addressed efficiently using the alternate direction method of multipliers (ADMM). The modes $u_k(t)$ are updated by Wiener filtering in the Fourier domain with a filter tuned to the current center frequency, see Eq. (2), then the center frequencies ω_k are updated as the center of gravity of the corresponding mode's power spectrum, expressed as Eq. (3), and finally the Lagrangian multiplier λ enforcing exact constraints is updated as the dual ascent by Eq. (4). The updating procedure is repeated until the convergence condition is satisfied, as in Eq. (5).

$$\hat{u}_k^{n+1}(\omega) \leftarrow \frac{\hat{f}(\omega) - \sum_{i<k} \hat{u}_i^{n+1}(\omega) - \sum_{i>k} \hat{u}_i^{n}(\omega) + \frac{\hat{\lambda}^n(\omega)}{2}}{1 + 2\theta(\omega - \omega_k^n)^2} \tag{2}$$

$$\omega_k^{n+1} \leftarrow \frac{\int_0^\infty \omega \left|\hat{u}_k^{n+1}(\omega)\right|^2 d\omega}{\int_0^\infty \left|\hat{u}_k^{n+1}(\omega)\right|^2 d\omega} \tag{3}$$

$$\hat{\lambda}^{n+1}(\omega) \leftarrow \hat{\lambda}^n(\omega) + \tau\left(\hat{f}(\omega) - \sum_k \hat{u}_k^{n+1}(\omega)\right) \tag{4}$$

$$\sum_k \frac{\left\|\hat{u}_k^{n+1} - \hat{u}_k^n\right\|_2^2}{\left\|\hat{u}_k^n\right\|_2^2} < \epsilon \tag{5}$$

where $\hat{u}_k^{n+1}(\omega)$,$\hat{f}(\omega)$ and $\hat{\lambda}^{n+1}(\omega)$ represent the Fourier Transforms of $u_k^{n+1}(t)$,$f(t)$ and $\lambda^{n+1}(t)$, respectively; n is the iterations, θ is a quadratic penalty term, τ is the iterative factor that indicates VMD's noise tolerance and ϵ denotes the convergence tolerance.

2.2 Recurrent neural network

Recurrent neural network models perform deep learning by a unique recurrent structure (Elman, 1990), as illustrated in Figure 1. In terms of time series predicting, the recurrent units remember earlier information, processing not only new data but also previous outputs to generate an up-to-date prediction. However, RNN models have difficulty dealing with long-term information. Additionally, standard RNN suffers from the gradient vanishing or exploding problem. To overcome the constraints of standard RNN, long short-term memory (LSTM), gated recurrent unit (GRU), bidirectional LSTM (BiLSTM) and bidirectional GRU (BiGRU), these variants of RNN are designed. Their working principles are explained in detail as follows.

[Figure 1 about here.]

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

2.2.1 Long short-term memory

LSTM models are explicitly constructed with special recurrent structures to remember information for long periods, and they have three gates to control the cell state that stores and conveys information (Hochreiter and Schmidhuber, 1997), which is depicted as Figure 2. The forget gate f_t determines how much information should be forgotten from the cell state, which constructs the long-term memory, as represented in Eq. (6). The input gate i_t is responsible for deciding what new information should be stored in the cell, and the corresponding equations are Eq. (7) and Eq. (8). The output gate o_t is to generate outputs, Eq. (9), and update the cell states C_t and the hidden states h_t, expressed as Eq. (10) and Eq. (11) respectively.

[Figure 2 about here.]

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + b_f\right) \tag{6}$$

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + b_i\right) \tag{7}$$

$$\tilde{C}_t = \tanh\left(W_{xC}x_t + W_{hC}h_{t-1} + b_C\right) \tag{8}$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + b_o\right) \tag{9}$$

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \tag{10}$$

$$h_t = o_t \otimes \tanh(C_t) \tag{11}$$

where σ and tanh are activation functions, denoting sigmoid function and hyperbolic tangent function, respectively; x_t is the input and $\tilde{C}_t$ is candidate memory; W_{xf}, W_{xi}, W_{xC}, W_{xo} and W_{hf}, W_{hi}, W_{hC}, W_{ho} represent the corresponding weights to x_t and h_{t-1}; b_f, b_i, b_C and b_o are the related bias vectors; $\otimes$ indicates element-wise multiplication.

2.2.2 Gated recurrent unit

GRU also overcomes the drawbacks of standard RNN. Unlike LSTM, however, it only has two gates: a reset gate and an update gate (Cho et al., 2014). The rest gate r_t is accountable for the short-term dependencies by determining which historical

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

data should be forgotten, represented as Eq. (12). The update gate z_t manages the long-term dependencies by controlling what information is delivered to the future, Eq. (13). The hidden state h_t is then updated according to Eq. (4) and Eq. (15). The update gate performs functions similar to the forget and input gates of LSTM, so the recurrent structure of GRU (Figure 3) is less complex, which makes it more efficient computationally from a theoretical standpoint (Chung et al., 2014).

$$r_t = \sigma\left(W_{xr}x_t + W_{hr}h_{t-1} + b_r\right) \tag{12}$$

$$z_t = \sigma\left(W_{xz}x_t + W_{hz}h_{t-1} + b_z\right) \tag{13}$$

$$\hat{h}_t = tanh(W_{xh}x_t + W_{hh}(r_t h_{t-1}) + b_h) \tag{14}$$

$$h_t = z_t \otimes h_{t-1} + (1 - z_t) \otimes \hat{h}_t \tag{15}$$

[Figure 3 about here.]

2.2.3 Bidirectional recurrent neural network

Bidirectional RNN (BiRNN) is an RNN variant model that takes into account both past and future information to predict the target (Schuster and Paliwal, 1997; Graves and Schmidhuber, 2005). The architecture of a bidirectional RNN is seen in Figure 4. It adds an additional hidden layer to the RNN construction so that information can be conveyed backward. The hidden state h_t is obtained by concatenating the forward and backward hidden states, $\overrightarrow{h}_t$ and $\overleftarrow{h}_t$, implying that the output is generated by combining information from two hidden layers. To avoid the limitations of standard RNN, BiLSTM and BiGRU are used instead of BiRNN, which have excellent performance in time series prediction.

[Figure 4 about here.]

2.3 Universal multifractals

Universal multifractals (UM) have been widely used to describe nonlinear phenomena that have a multiplicative structure, such as rainfall. The core principle of the framework of UM is briefly explained here, and interested readers could refer to references (e.g., Schertzer and Lovejoy, 1987, 2011; Lovejoy and Schertzer, 2007) for more details. Let's denote ε_λ is a conservative field at resolution λ (=L/l, the ration between the outer scale of the phenomenon L and the observation scale l), the statistical moment of order q can be defined as:

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

$$\langle \varepsilon_\lambda^q \rangle \approx \lambda^{K(q)} \tag{16}$$

where $K(q)$ is the moment scaling function, characterizing the variability of the field at all scales.

In the UM framework, the moment scaling function $K(q)$ can be determined by two scale-invariant parameters C_1 and α in the conservative field, expressed as Eq. (17) (Schertzer and Lovejoy, 2011). C_1 is the mean intermittency co-dimension, which measures the average sparseness of the field. α is the multifractality index ($0 \leq \alpha \leq 2$), which indicates how fast the intermittency evolves when considering singularities slightly different from the average field singularity.

$$K(q) = \begin{cases} \frac{C_1}{\alpha - 1}(q^\alpha - q) & \alpha \neq 1 \\ C_1 q \ln q & \alpha = 1 \end{cases} \tag{17}$$

The trace moment (TM) technique can be used to estimate UM parameters (Schertzer and Lovejoy, 2011; Gires et al., 2013). The steps in the technique are as follows: first, calculate the empirical statistical moment $\langle \varepsilon_\lambda^q \rangle$ (corresponding to the trace moment of fluxes) of order q for each resolution λ, then plot the logarithm of the average field $\langle \varepsilon_\lambda^q \rangle$ versus the logarithm of λ, later perform linear regression to obtain the slope $K(q)$, and finally, according to the theoretical expression of $K(q)$ (Eq. (17)), C_1 is given by $K'(1) = C_1$ and α by $K''(1) = \alpha C_1$ because $\langle \varepsilon_\lambda^q \rangle = 1$, i.e. $K(1) = 0$ for the conservative field.

An alternative method for directly estimating the UM parameters C_1 and α is the double trace moment (DTM) (Lavallée et al., 1993; Gires et al., 2012). Based on the assumption that the conservative field $\varepsilon_\lambda^{(\eta)}$ is renormalized by upscaling the η-power of the field at maximum resolution,

$$\varepsilon_\lambda^{(\eta)} = \frac{\varepsilon_\lambda^\eta}{\langle \varepsilon_\lambda^\eta \rangle} \tag{18}$$

Then, the statistical moment $K(q, \eta)$ of order q is defined as:

$$\left\langle \varepsilon_\lambda^{(\eta)q} \right\rangle \approx \lambda^{K(q,\eta)} \tag{19}$$

with

$$K(q, \eta) = K(q\eta) - \eta K(q) \tag{20}$$

In the specific framework of UM, the statistical moment $K(q, \eta)$ can be expressed as:

$$K(q, \eta) = \eta^\alpha K(q) \tag{21}$$

Therefore, UM parameters C_1 and α are obtained according to the slope and intercept of the linear portion of the log-log plot $K(q, \eta)$ vs η.

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

When a multifractal field ϕ_λ is non-conservative ($\langle\phi_\lambda\rangle \neq 1$), it is usually assumed that it can be written as:

$$\phi_\lambda = \varepsilon_\lambda \lambda^{-H} \tag{22}$$

where ε_λ is a conservative field ($\langle\varepsilon_\lambda\rangle = 1$) of the moment scaling function $K_c(q)$ depending only on C_1 and α; H is the non-conservation parameter ($H = 0$ for the conservative field).

The moment scaling function $K(q)$ of ϕ_λ is given by:

$$K(q) = K_c(q) - Hq \tag{23}$$

H can be estimated using the following formula Tessier et al. (1993) :

$$\beta = 1 + 2H - K_c(2) \tag{24}$$

where β is the spectral slope that characterizes the power spectrum of a scaling field, which follows a power law over a wide range of wave numbers:

$$E(k) \propto k^{-\beta} \tag{25}$$

Theoretically, a fractional integration of order H (equivalent to a multiplication by k^H in the Fourier space) is performed to retrieve ε_λ from ϕ_λ. A common approximation is to take ε_Λ as the absolute value of the fluctuation of ϕ_Λ at the maximum resolution of Λ and renormalizing it, shown as Eq. (26) in the one-dimension (Lavallée et al., 1993). Then, ε_λ is obtained by upscaling ε_Λ.

$$\varepsilon_\Lambda = \frac{|\phi_\Lambda(i+1) - \phi_\Lambda(i)|}{\langle|\phi_\Lambda(i+1) - \phi_\Lambda(i)|\rangle} \tag{26}$$

3 Case study

3.1 Study area and datasets

Two rainfall time series with daily and hourly resolutions in Champs-sur-Marne were collected from NASA research The POWER Project (https://power.larc.nasa.gov) that provides meteorological datasets. The daily time series covered January 1, 2001 to December 31, 2020 (a total of 7305 data), of which from January 1, 2001 to January 7, 2015(5120 data, accounting for 70% of the total dataset) were selected as the training set while the remaining were used as the non-training set. The non-training set was further divided into a validation set to tune hyperparameters according to loss changes and a testing set

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

(1024 data, from March 14, 2018 to December 31, 2020) to evaluate the predicting performance, as presented in Figure 5(a). In addition, the rainfall time series with hourly resolution for the period between January 1, 2001 and November 1, 2001 (a total of 7305 data) was also studied and divided into three sets: a training set (5120 data), a validation set (1161 data), and a testing set (1024 data), as shown in Figure 5(b).

[Figure 5 about here.]

3.2 Model process

3.2.1 The implementation of VMD-RNN

In order to avoid using information from the future, the original rainfall time series was first divided into the training and non-training sets, and then the training set was decomposed into several sub-sequences and applied to train the models (Zhang et al., 2015; Zuo et al., 2020). To predict in the testing set, time series from the non-training set were sequentially appended to the training set, and the decomposition process was repeated with the rainfall time series of the next step appended. Following that, four variant RNN models were used to predict individual sub-sequences. The root mean square error (RMSE) was used to select the ideal RNN model with the optimal parameters for each sub-sequence. In addition to RMSE, UM was also employed to evaluate prediction performances, characterizing the extreme variability of time series. The implementation of the hybrid deep learning model (VMD-RNN) is summarized as follows and presented in Figure 6.

Step 1: Divide the original rainfall time series $f(t)$ (t=1,2,...,N, where N is the length of total data) into a training set $f_T(t)$ (t=1,2,...,N_t, where N_t is the training set length) and a non-training set $f_N(t)$ (t=1,2,...,N_n, where N_n is the non-training set length).

Step 2: Use VMD to decompose the training set $f_T(t)$ into sub-sequences $u_{Ti}(t)$ (i=1,2,...,K).

Step 3: Sequentially append the non-training data $f_N(t)$ to the training set to generate N_n new appended sequences $f^j_{NT}(t)$ (j=1,2,...,N_n and t=1,2,...,$N_t + j$), and repeat decomposing each append sequence $f^j_{NT}(t)$ into K sets of appended sub-sequences $u^j_{NTi}(t)$ (i=1,2,...,K).

Step 4: Exact the last sample $u^j_{NTi}(N_t + j)$ of each set of appended sub-sequences $u^j_{NTi}(t)$ as a non-training sample and divide the generated non-training samples $N_{vte} = N_n$ into two subsets: validation samples N_v and testing samples N_{te}.

Step 5: For each sub-sequence, combine data from the training set and validation samples as history data, which is then used to train four variant RNN models and tune hyperparameters to find an ideal predicting model with optimal parameters.

Step 6: For each sub-sequence, input testing samples into the corresponding predicting models and obtain an individual predicted result $y_i(t)$ (i=1,2,...,K).

Step 7: Aggregate the predicted results of each sub-sequence to generate the final predicted result $y(t) = \sum_{i=k}^{K} y_i(t)$.

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

Step 8: Use the framework of UM to analyze the predicted and actual time series in the testing set.

[Figure 6 about here.]

3.2.2 Parameters of VMD

The decomposition performance of VMD is affected by the decomposition level K, the quadratic penalty term θ, the convergence tolerance ε, and the noise tolerance τ. In this study, the number of K was identified by observing the power spectral density (PSD) of the last sub-sequence. The value of K was determined on the training set with 5120 data. First, an initial K value was given, such as $K = 5$, and there were five sub-sequences (IMFs) with the same length of training set. Then, each sub-sequence was divided into 40 samples with 128 data, to perform the spectral analysis and plot the corresponding PSD of sub-sequences. After that, K was increased by one and the plotting PSD was repeated until the PSD of the last sub-sequence exhibited an evident change, compared with the previous last sub-sequence. For daily time series, the optimal number of K was 8, which is depicted in Figure 7, whereas $K = 6$ for hourly time series. Based on the trial and error, other parameters of VMD were suggested as: θ=100, ε=1e-9 and τ=0.

[Figure 7 about here.]

3.2.3 Parameters of RNN

In the process of training, hyperparameters such as the number of inputs, epoch, hidden layers, and hidden units all influence the performance of models. Without loss of generality, the first sub-sequence (IMF1) is taken as an example to describe the determination of the ideal RNN structure with the optimal hyperparameters. The specific process is as follows: First, initial a single hidden layer model with 5, 10, 15 input neurons and 1 output neuron, run different variants of RNN model (LSTM, GRU, BiLSTM, BiGRU) for various hidden neurons 32, 64 and 128. All experiments were intended to run for 10,000 epochs (one epoch is defined as when an entire dataset is passed forward and backward through the neural network only once), but early stopping with a large patience value (=200) was applied to prevent unnecessary overfitting, which means the model will stop the training if the performance on the validation dataset does not improve after 200 epochs. After adjusting hyperparameters, the ideal model with optimal parameters was found for the first sub-sequence (IMF1) where RMSE is the least.

Then, different second hidden layers with hidden neurons 32, 64 and 128 were added to the first hidden layer with optimal parameters in order to discover the optimal parameters for the second hidden layer. By analogy, a third hidden layer was added. Through the above method, the variant RNN model structures of IMF1-IMF8 components were obtained, as shown in Table 2.

[Table 2 about here.]

3.3 Open sources

This study made extensive use of open-source software. Python 3.8 was the programming language. The packages, Numpy (Van Der Walt et al., 2011), Pandas (McKinney et al., 2011), and Scikit-Learn (Pedregosa et al., 2011), were used to preprocess

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

data. Tensorflow (Abadi et al., 2016) and Keras (Chollet et al., 2018) were the deep learning frameworks used to analyze time series, and Matplotlib (Hunter, 2007) was used to create all the resulting figures. The decomposition of time series by VMD was implemented based on the package of vmdpy (Carvalho et al., 2020), which is derived from the original VMD Matlab toolbox (Dragomiretskiy and Zosso, 2013). TM and DTM analysis were performed to calculate UM parameters according to the Multifractal toolbox that was provided by the website (https://hmco.enpc.fr/portfolio-archive/multifractals-toolbox) (Gires et al., 2011, 2012, 2013).

4 Result analysis

To verify the effectiveness of the hybrid VMD-RNN model, the benchmark method (pure LSTM model without decomposition) was introduced. The benchmark also used the previous 5-day rainfall values to predict the next day's rainfall. The parameters for the pure LSTM were adjusted by trial and error. The qualitative and quantitative analysis of one-step-ahead predicted rainfall time series from two different models were conducted.

4.1 Daily rainfall series

Figure 8 shows the predicted daily time series in the testing set. It compares the predicted results of the VMD-RNN hybrid model and the pure LSTM model with the actual data. It can be clearly observed that the hybrid model has a better fit for most of the points. The comparison of prediction performance with and without VMD for daily time series in the testing set can be seen in Figure 9. The scatter plot demonstrates that the VMD-RNN model has superior performance in predicting high and low values for daily time series, whereas the benchmark LSTM model is incapable of doing so. The predicted values obtained by the baseline model exhibit a considerable deviation from the best linear fitting line (blue dotted line), which indicates inferior performance compared to the VMD-RNN model.

[Figure 8 about here.]

[Figure 9 about here.]

It was also necessary to know which model performed better from the quantitative aspect. Table 3 compares the results of three widely used criteria: RMSE, MAE, and MAPE. It can be seen that the three criteria of VMD-RNN are plainly lower, so the hybrid model outperforms the pure model. It further confirms the strong capability of the hybrid model in rainfall prediction.

[Table 3 about here.]

In addition to calculating the prediction error, the UM technique was also introduced to evaluate prediction performance since it enables the extreme variability of rainfall time series to be characterized. According to Tessier et al. (1996), the rainfall series in France exhibits a rough scaling break phenomenon between 16 days and 30 days. Therefore, the analysis of UM starts with a range of scales from 1 day, increasing in powers of two to an outer scale of 16 days. Figure 10 presents $log\langle\varepsilon_\lambda^q\rangle$ versus

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

$log\lambda$ over the range of q between 0.3 and 2.5 with a coefficient of determination greater than 0.99, the log-log plot of $\left\langle \varepsilon_{\lambda}^{(\eta)q} \right\rangle$ vs λ for $q = 1.5$, and the corresponding log-log plot of $K(q,\eta)$ vs η.

[Figure 10 about here.]

All the parameter values estimated using the TM and DTM methods are listed in Table 4. It is worthwhile to note that the values of α and C_1 obtained using the DTM technique have some differences from those estimated by using TM, but the overall difference is not significant. From the standpoint of the results of UM analysis, predicted time series by VMD-RNN are closer to actual time series than predicted by LSTM without decomposition for daily time series.

[Table 4 about here.]

4.2 Hourly rainfall series

Figure 11 displays the hourly time series in the testing set with 1024 data. The qualitative analysis reveals that the predicting performance of the hybrid VMD-RNN model and the simple LSTM model without decomposition is nearly similar for hourly rainfall time series. This is further confirmed by Figure 12, which depicts the comparison of predicted results and actual rainfall. The scatter plot reveals that the predicted values from VMD-RNN basically agree with the corresponding actual values, but the values predicted from the baseline LSTM model do not yield the same level of alignment.

[Figure 11 about here.]

[Figure 12 about here.]

On the other hand, Figure 13 shows the results of UM analysis for hourly time series, and Table 5 presents the estimated UM parameters α and C_1 from TM and DTM analyses. These results indicate quantitatively that the predictive performance of the VMD-RNN model is comparable to that of the pure LSTM model, without obviously demonstrating the benefits of decomposition.

[Table 5 about here.]

[Figure 13 about here.]

5 Conclusions and future work

In this study, the hybrid VMD-RNN model was used as a methodology for forecasting rainfall with a one-step lead time. VMD was first used to extract hidden information to understand the complex original time series. Then variants of RNN were applied to handle problems involving sequential prediction. By combining the dominant characteristics of VMD in decomposing nonlinear time series and the favourable performance of variant RNN models in predicting complex sequential problems, the

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

hybrid model based on VMD and RNN was employed to predict rainfall time series with daily and hourly resolution. The framework of UM was subsequently introduced to evaluate the performance of predicting rainfall time series.

According to the above study, the following conclusions could be drawn: (1) The number of K in the process of VMD can be determined by analysing the PSD of the respective last sub-sequence. (2) The hybrid VMD-RNN model provides a reliable one-step-ahead prediction, with better performance in predicting high and low values than the pure LSTM model. (3) From the perspective of UM analysis, the values of α and C_1 obtained from predicted daily time series by VMD-RNN are closer to actual time series than those by LSTM without decomposition, thus validating the usefulness and applicability of the hybrid model. (4) For hourly rainfall time series, the prediction performance of VMD-RNN is comparable to that of the pure LSTM model, without significantly displaying the benefits of decomposition.

However, there are still some limits to this study, and corresponding improvements will be implemented in future work. For real applications, a one-step-ahead prediction is insufficient, multi-step-ahead rainfall prediction is therefore currently under investigation. On the other hand, the hybrid model does not account for the advantage of UM in analyzing the randomness and complexity of rainfall, so it is necessary to develop a new model that combines UM and DL in order to enhance the performance of predicting complex time series.

Code and data availability. The source python code of VMD is available at https://github.com/vrcarva/vmdpy (Carvalho et al., 2020). The Multifractal toolbox is provided by the website (https://hmco.enpc.fr/portfolio-archive/multifractals-toolbox) (Gires et al., 2013, 2012, 2011). Two rainfall time series with daily and hourly resolutions in Champs-sur-Marne are collected from The POWER Project (https://power.larc.nasa.gov).

Author contributions. HZ, DS and IT developed the concept for the manuscript; HZ was responsible for conducting the data analysis, coding the hybrid model, and writing the first draft. DS and IT supervised the entire research and assisted with answering questions. All authors reviewed and edited the paper.

Competing interests. The authors declare that they have no conflicts of interest to report regarding the present study.

Acknowledgements. The first author is funded by the China Scholarship Council (202006120045).

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

References

Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., Devin, M., Ghemawat, S., Irving, G., Isard, M., et al.: {TensorFlow}: A System for {Large-Scale} Machine Learning, in: 12th USENIX symposium on operating systems design and implementation (OSDI 16), pp. 265–283, 2016.

Barrera-Animas, A. Y., Oyedele, L. O., Bilal, M., Akinosho, T. D., Delgado, J. M. D., and Akanbi, L. A.: Rainfall prediction: A comparative analysis of modern machine learning algorithms for time-series forecasting, Machine Learning with Applications, 7, 100 204, 2022.

Bauer, P., Thorpe, A., and Brunet, G.: The quiet revolution of numerical weather prediction, Nature, 525, 47–55, 2015.

Carvalho, V. R., Moraes, M. F., Braga, A. P., and Mendes, E. M.: Evaluating five different adaptive decomposition methods for EEG signal seizure detection and classification, Biomedical Signal Processing and Control, 62, 102 073, 2020.

Chattopadhyay, S. and Chattopadhyay, G.: Univariate modelling of summer-monsoon rainfall time series: comparison between ARIMA and ARNN, Comptes Rendus Geoscience, 342, 100–107, 2010.

Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y.: Learning phrase representations using RNN encoder-decoder for statistical machine translation, arXiv preprint arXiv:1406.1078, 2014.

Chollet, F. et al.: Keras: The python deep learning library, Astrophysics source code library, pp. ascl–1806, 2018.

Chung, J., Gulcehre, C., Cho, K., and Bengio, Y.: Empirical evaluation of gated recurrent neural networks on sequence modeling, arXiv preprint arXiv:1412.3555, 2014.

Deidda, R.: Rainfall downscaling in a space-time multifractal framework, Water Resources Research, 36, 1779–1794, 2000.

Devi, A. S., Maragatham, G., Boopathi, K., and Rangaraj, A.: Hourly day-ahead wind power forecasting with the EEMD-CSO-LSTM-EFG deep learning technique, Soft Computing, 24, 12 391–12 411, 2020.

Ding, M., Zhou, H., Xie, H., Wu, M., Nakanishi, Y., and Yokoyama, R.: A gated recurrent unit neural networks based wind speed error correction model for short-term wind power forecasting, Neurocomputing, 365, 54–61, 2019.

Dragomiretskiy, K. and Zosso, D.: Variational mode decomposition, IEEE transactions on signal processing, 62, 531–544, 2013.

Elman, J. L.: Finding structure in time, Cognitive science, 14, 179–211, 1990.

Gauch, M., Kratzert, F., Klotz, D., Nearing, G., Lin, J., and Hochreiter, S.: Rainfall–runoff prediction at multiple timescales with a single Long Short-Term Memory network, Hydrology and Earth System Sciences, 25, 2045–2062, 2021.

Gires, A., Tchiguirinskaia, I., Schertzer, D., and Lovejoy, S.: Analyses multifractales et spatio-temporelles des précipitations du modèle Méso-NH et des données radar, Hydrological Sciences Journal–Journal des Sciences Hydrologiques, 56, 380–396, 2011.

Gires, A., Tchiguirinskaia, I., Schertzer, D., and Lovejoy, S.: Influence of the zero-rainfall on the assessment of the multifractal parameters, Advances in water resources, 45, 13–25, 2012.

Gires, A., Tchiguirinskaia, I., Schertzer, D., and Lovejoy, S.: Development and analysis of a simple model to represent the zero rainfall in a universal multifractal framework, Nonlinear Processes in Geophysics, 20, 343–356, 2013.

Graves, A. and Schmidhuber, J.: Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neural networks, 18, 602–610, 2005.

Graves, A., Jaitly, N., and Mohamed, A.-r.: Hybrid speech recognition with deep bidirectional LSTM, in: 2013 IEEE workshop on automatic speech recognition and understanding, pp. 273–278, IEEE, 2013.

Gupta, V. K. and Waymire, E. C.: A statistical analysis of mesoscale rainfall as a random cascade, Journal of Applied Meteorology and Climatology, 32, 251–267, 1993.

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

Hadi, S. J. and Tombul, M.: Streamflow forecasting using four wavelet transformation combinations approaches with data-driven models: a comparative study, Water Resources Management, 32, 4661–4679, 2018.

Harris, D., Seed, A., Menabde, M., and Austin, G.: Factors affecting multiscaling analysis of rainfall time series, Nonlinear Processes in Geophysics, 4, 137–156, 1997.

He, R., Zhang, L., and Chew, A. W. Z.: Modeling and predicting rainfall time series using seasonal-trend decomposition and machine learning, Knowledge-Based Systems, 251, 109 125, 2022.

He, X., Luo, J., Zuo, G., and Xie, J.: Daily runoff forecasting using a hybrid model based on variational mode decomposition and deep neural networks, Water resources management, 33, 1571–1590, 2019.

Hochreiter, S. and Schmidhuber, J.: Long short-term memory, Neural computation, 9, 1735–1780, 1997.

Huang, N. E., Shen, Z., Long, S. R., Wu, M. C., Shih, H. H., Zheng, Q., Yen, N.-C., Tung, C. C., and Liu, H. H.: The empirical mode decomposition and the Hilbert spectrum for nonlinear and non-stationary time series analysis, Proceedings of the Royal Society of London. Series A: mathematical, physical and engineering sciences, 454, 903–995, 1998.

Hunter, J. D.: Matplotlib: A 2D graphics environment, Computing in science & engineering, 9, 90–95, 2007.

Lara-Benítez, P., Carranza-García, M., and Riquelme, J. C.: An experimental review on deep learning architectures for time series forecasting, International journal of neural systems, 31, 2130 001, 2021.

Lavallée, D., Lovejoy, S., Schertzer, D., and Ladoy, P.: Nonlinear variability and landscape topography: analysis and simulation, Fractals in geography, pp. 158–192, 1993.

Lin, J., Zhong, S.-h., and Fares, A.: Deep hierarchical LSTM networks with attention for video summarization, Computers & Electrical Engineering, 97, 107 618, 2022.

Lovejoy, S. and Schertzer, D.: Scale, scaling and multifractals in geophysics: twenty years on, in: Nonlinear dynamics in geosciences, pp. 311–337, Springer, 2007.

Lynch, P.: The origins of computer weather prediction and climate modeling, Journal of computational physics, 227, 3431–3444, 2008.

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.

Marsan, D., Schertzer, D., and Lovejoy, S.: Causal space-time multifractal processes: Predictability and forecasting of rain fields, Journal of Geophysical Research: Atmospheres, 101, 26 333–26 346, 1996.

McKinney, W. et al.: pandas: a foundational Python library for data analysis and statistics, Python for high performance and scientific computing, 14, 1–9, 2011.

Ni, L., Wang, D., Singh, V. P., Wu, J., Wang, Y., Tao, Y., and Zhang, J.: Streamflow and rainfall forecasting by two long short-term memory-based models, Journal of Hydrology, 583, 124 296, 2020.

Pati, Y. C., Rezaiifar, R., and Krishnaprasad, P. S.: Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, in: Proceedings of 27th Asilomar conference on signals, systems and computers, pp. 40–44, IEEE, 1993.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., et al.: Scikit-learn: Machine learning in Python, the Journal of machine Learning research, 12, 2825–2830, 2011.

Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., Carvalhais, N., et al.: Deep learning and process understanding for data-driven Earth system science, Nature, 566, 195–204, 2019.

Schertzer, D. and Lovejoy, S.: Physical modeling and analysis of rain and clouds by anisotropic scaling multiplicative processes, Journal of Geophysical Research: Atmospheres, 92, 9693–9714, 1987.

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

Schertzer, D. and Lovejoy, S.: Space–time complexity and multifractal predictability, Physica A: Statistical Mechanics and its Applications, 338, 173–186, 2004.

Schertzer, D. and Lovejoy, S.: Multifractals, generalized scale invariance and complexity in geophysics, International Journal of Bifurcation and Chaos, 21, 3417–3456, 2011.

Schertzer, D., Lovejoy, S., Schmitt, F., Chigirinskaya, Y., and Marsan, D.: Multifractal cascade dynamics and turbulent intermittency, Fractals, 5, 427–471, 1997.

Schertzer, D., Tchiguirinskaia, I., Lovejoy, S., and Hubert, P.: No monsters, no miracles: in nonlinear sciences hydrology is not an outlier!, Hydrological Sciences Journal–Journal des Sciences Hydrologiques, 55, 965–979, 2010.

Schultz, M., Betancourt, C., Gong, B., Kleinert, F., Langguth, M., Leufen, L., Mozaffari, A., and Stadtler, S.: Can deep learning beat numerical weather prediction?, Philosophical Transactions of the Royal Society A, 379, 20200 097, 2021.

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

Su, C., Huang, H., Shi, S., Jian, P., and Shi, X.: Neural machine translation with Gumbel Tree-LSTM based encoder, Journal of Visual Communication and Image Representation, 71, 102 811, 2020.

Tessier, Y., Lovejoy, S., and Schertzer, D.: Universal multifractals: Theory and observations for rain and clouds, Journal of Applied Meteorology and Climatology, 32, 223–250, 1993.

Tessier, Y., Lovejoy, S., Hubert, P., Schertzer, D., and Pecknold, S.: Multifractal analysis and modeling of rainfall and river flows and scaling, causal transfer functions, Journal of Geophysical Research: Atmospheres, 101, 26 427–26 440, 1996.

Van Der Walt, S., Colbert, S. C., and Varoquaux, G.: The NumPy array: a structure for efficient numerical computation, Computing in science & engineering, 13, 22–30, 2011.

Wu, H., Xu, J., Wang, J., and Long, M.: Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting, Advances in Neural Information Processing Systems, 34, 2021.

Xie, T., Zhang, G., Hou, J., Xie, J., Lv, M., and Liu, F.: Hybrid forecasting model for non-stationary daily runoff series: a case study in the Han River Basin, China, Journal of Hydrology, 577, 123 915, 2019.

Yu, Z.-G., Leung, Y., Chen, Y. D., Zhang, Q., Anh, V., and Zhou, Y.: Multifractal analyses of daily rainfall time series in Pearl River basin of China, Physica A: Statistical Mechanics and its Applications, 405, 193–202, 2014.

Zhang, X., Peng, Y., Zhang, C., and Wang, B.: Are hybrid models integrated with data preprocessing techniques suitable for monthly streamflow forecasting? Some experiment evidences, Journal of Hydrology, 530, 137–152, 2015.

Zuo, G., Luo, J., Wang, N., Lian, Y., and He, X.: Decomposition ensemble model based on variational mode decomposition and long short-term memory for streamflow forecasting, Journal of Hydrology, 585, 124 776, 2020.

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

List of Figures

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

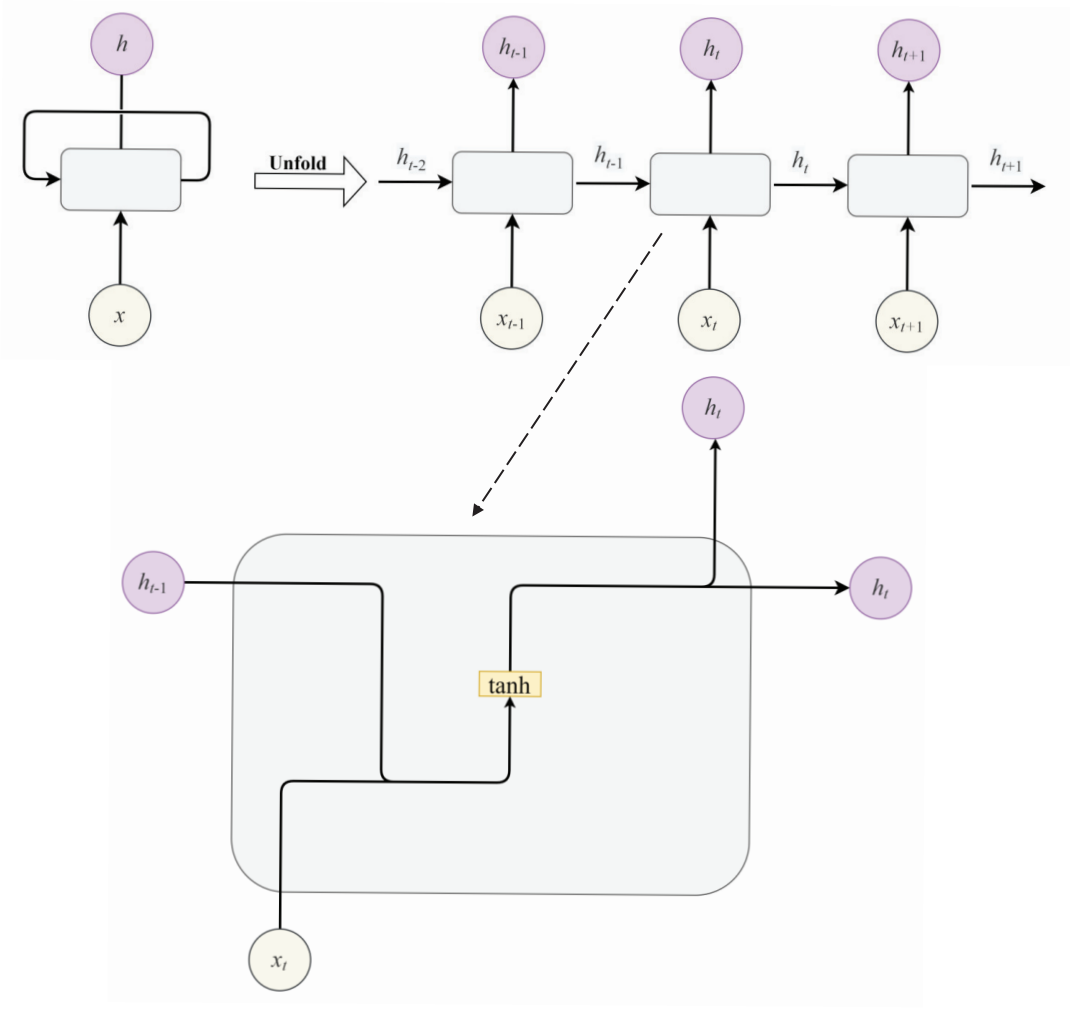


Figure 1. The structure of standard RNN

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

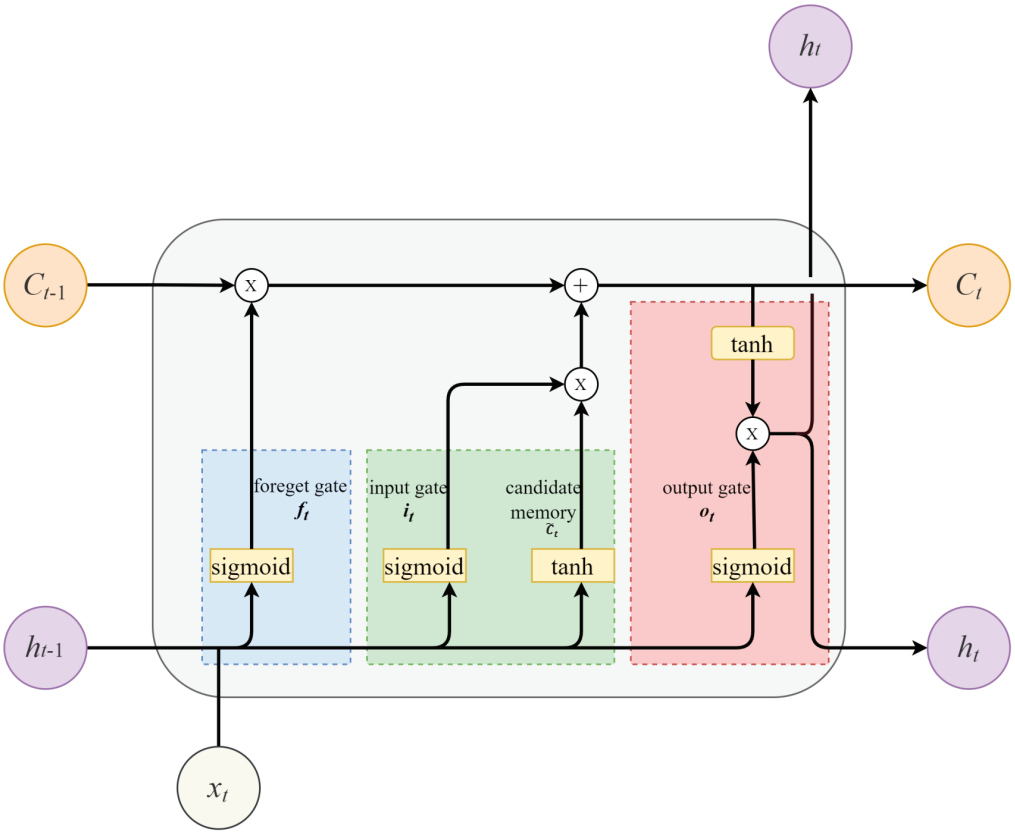


Figure 2. The recurrent structure of LSTM

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

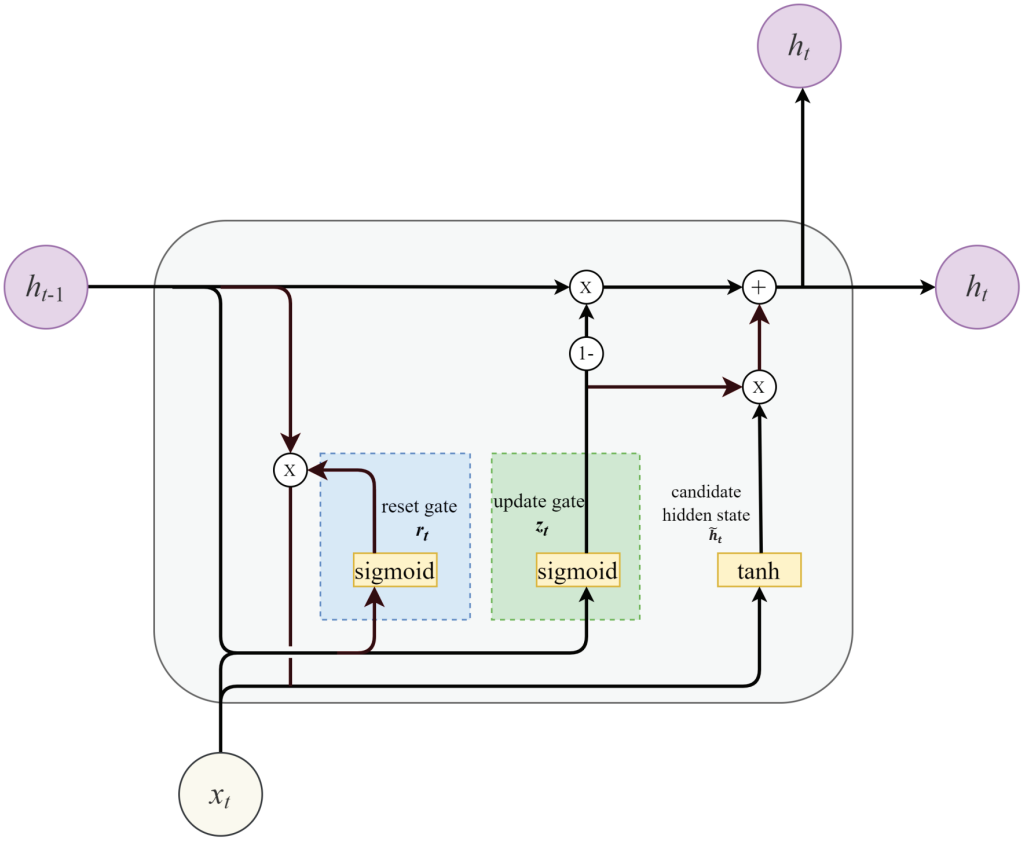


Figure 3. The recurrent structure of GRU

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

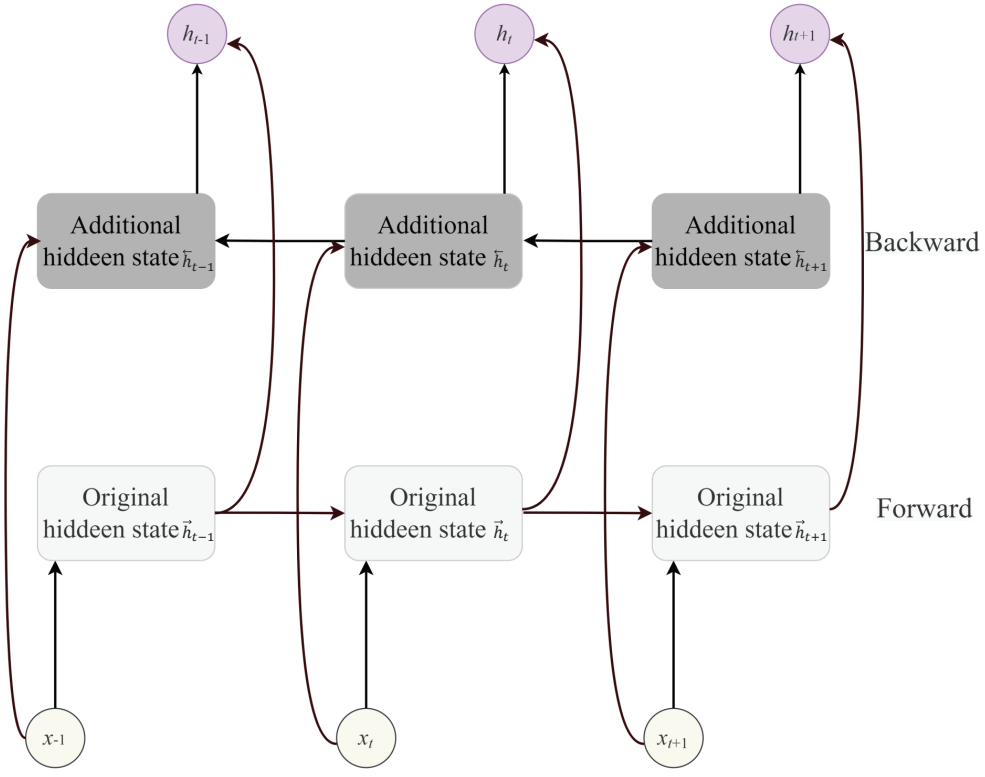
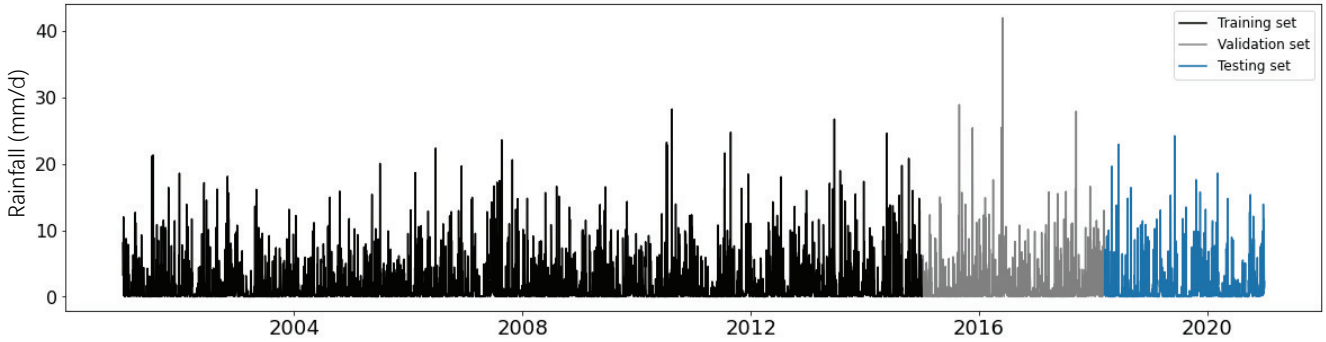
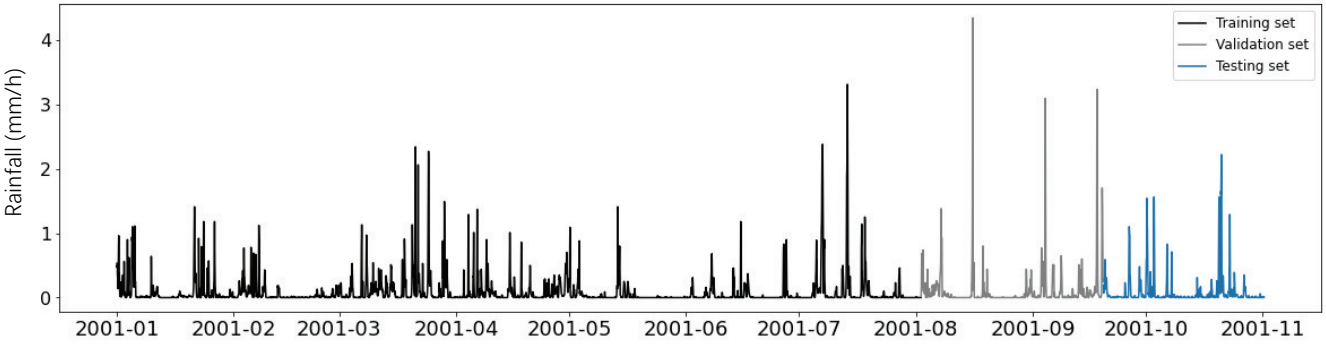


Figure 4. The structure of BiRNN

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024



(a) daily time series



(b) hourly time series

Figure 5. Original rainfall time series

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

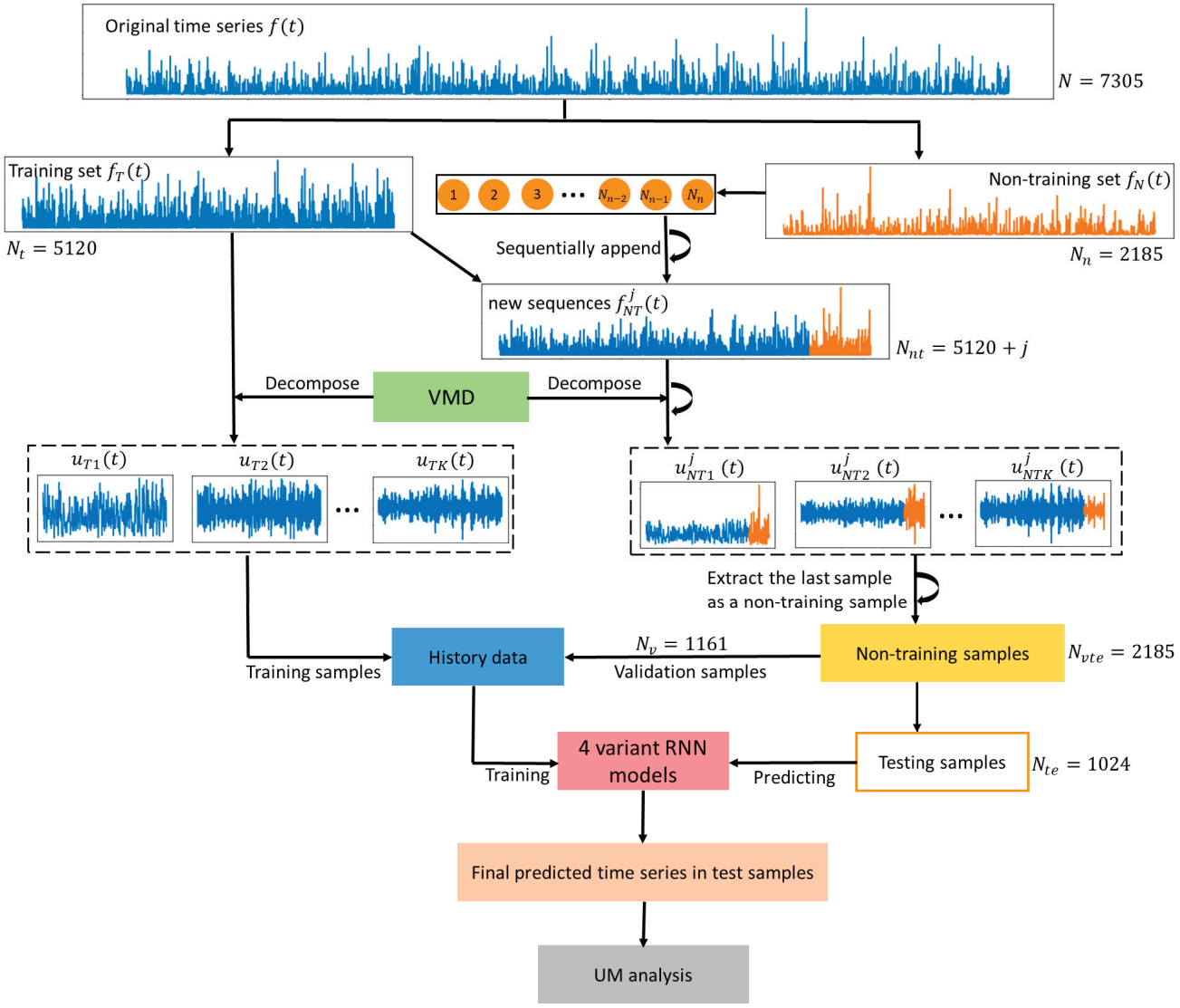


Figure 6. The process of the VMD-RNN model

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

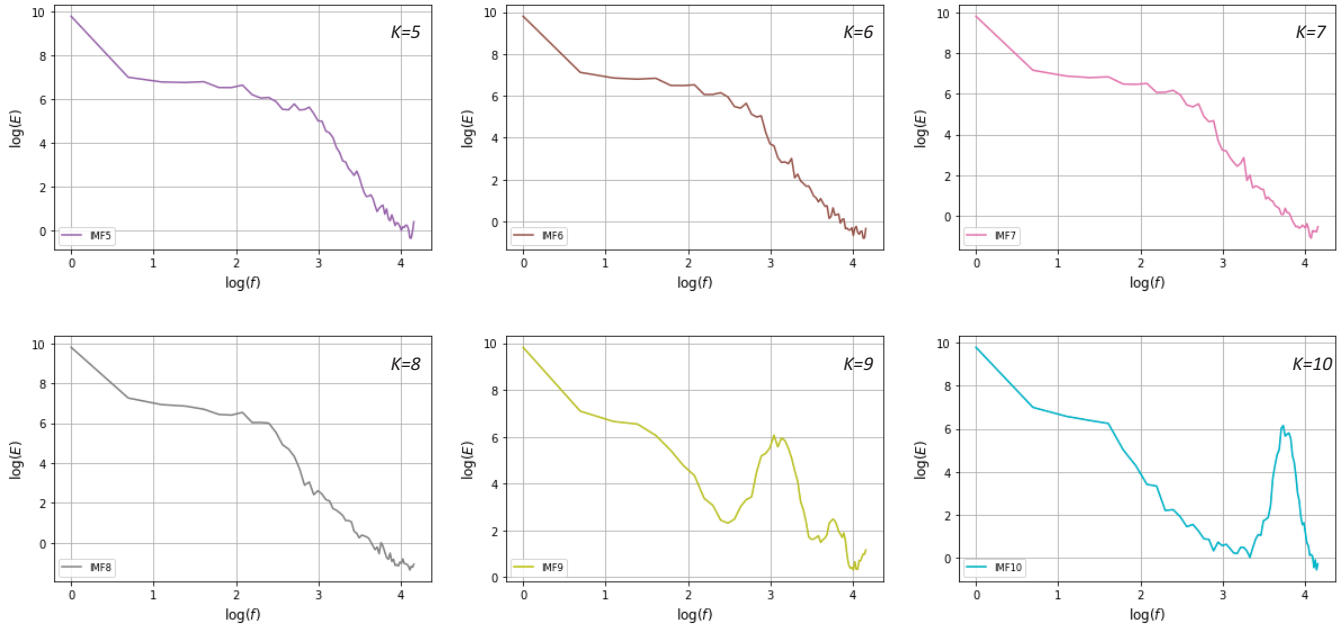


Figure 7. PSD of the corresponding last sub-sequence when K from 5 to 10

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

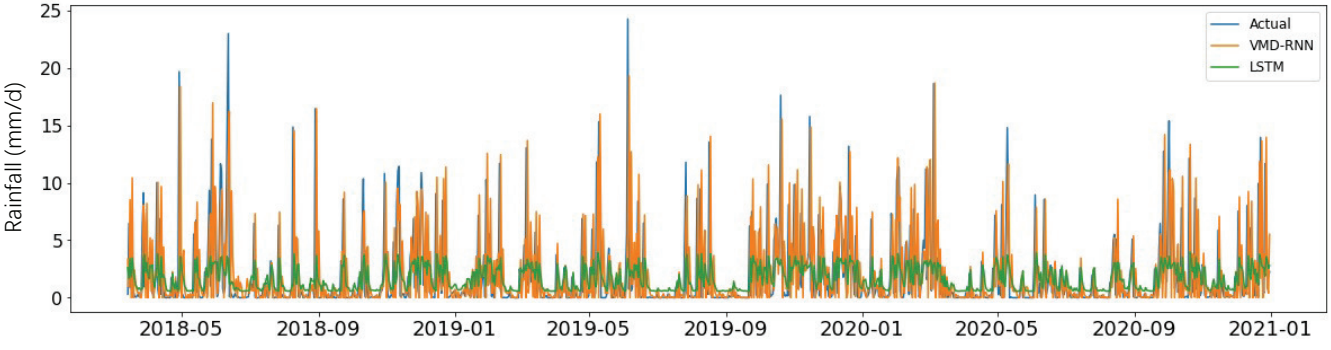


Figure 8. Predicted and actual daily time series in the testing set

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

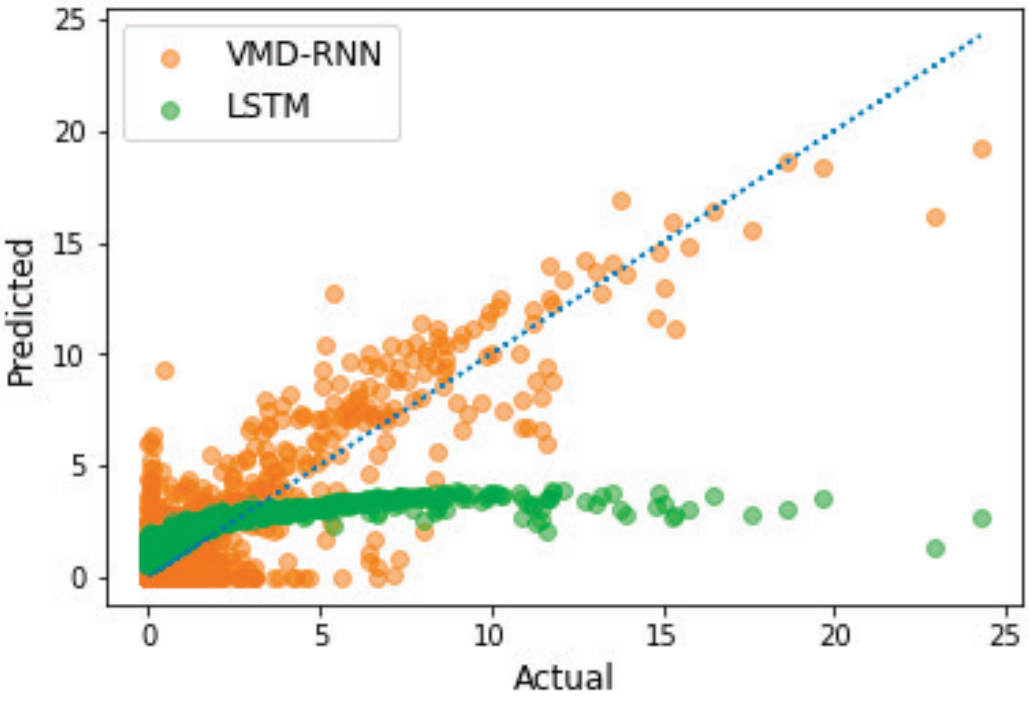
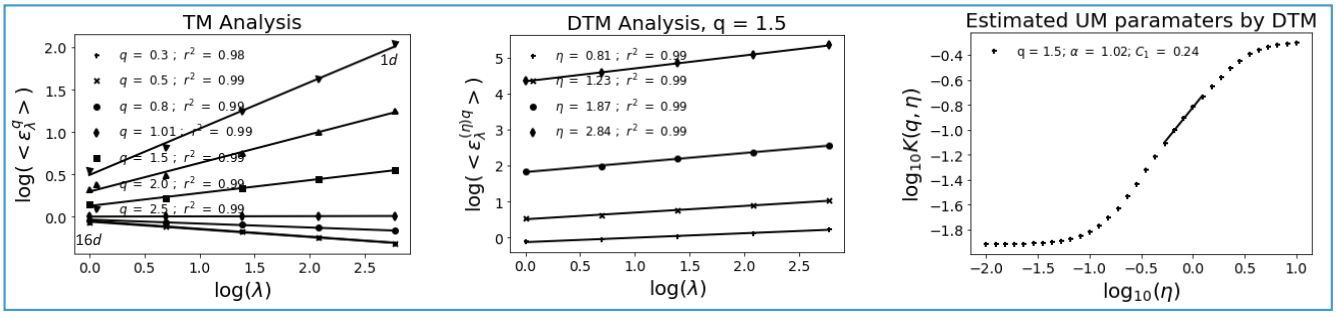
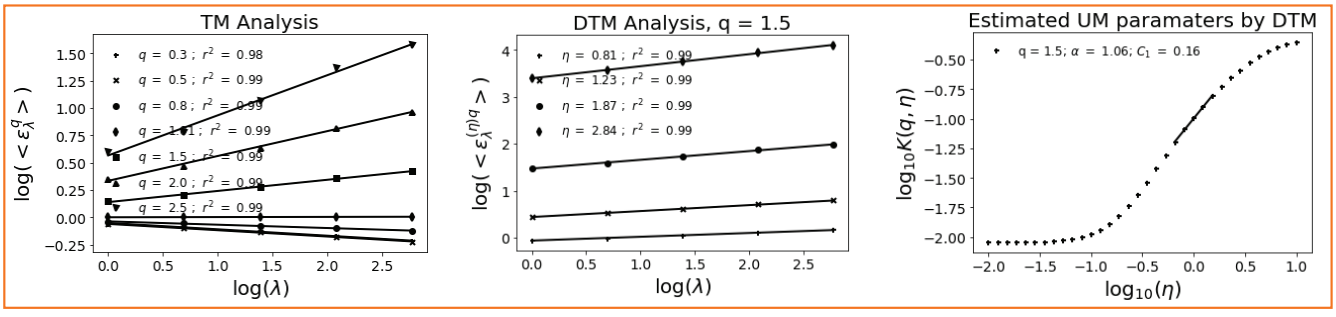


Figure 9. The comparison between predicted and actual daily rainfall values

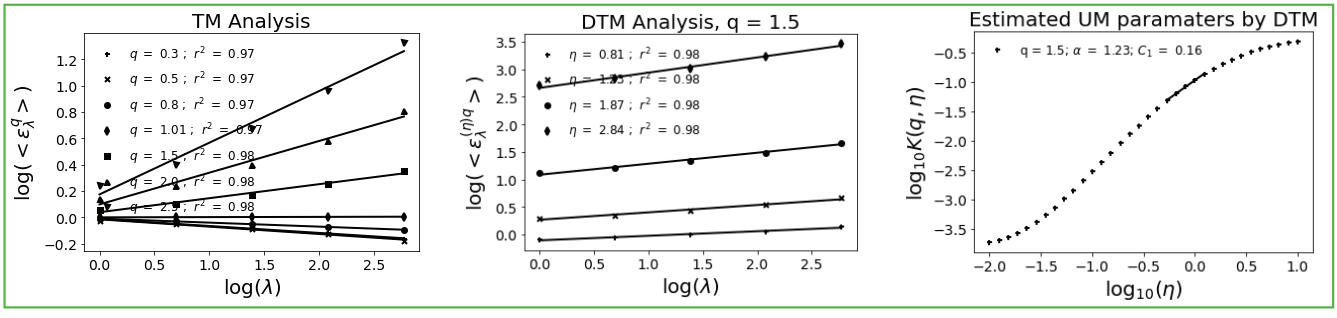
https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024



(a) actual time series



(b) predicted time series by VMD-RNN



(c) predicted time series by LSTM without decomposition

Figure 10. UM results for daily time series in the testing set

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

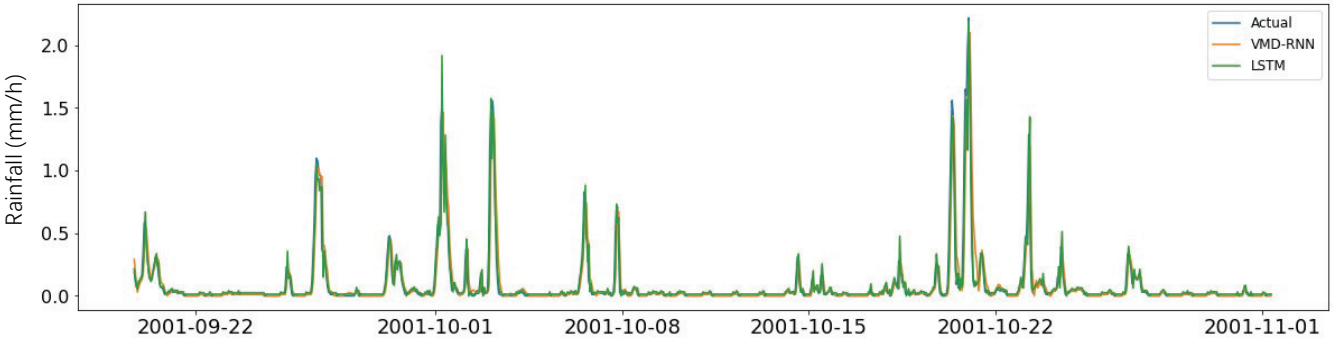


Figure 11. Predicted and actual hourly time series in the testing set

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

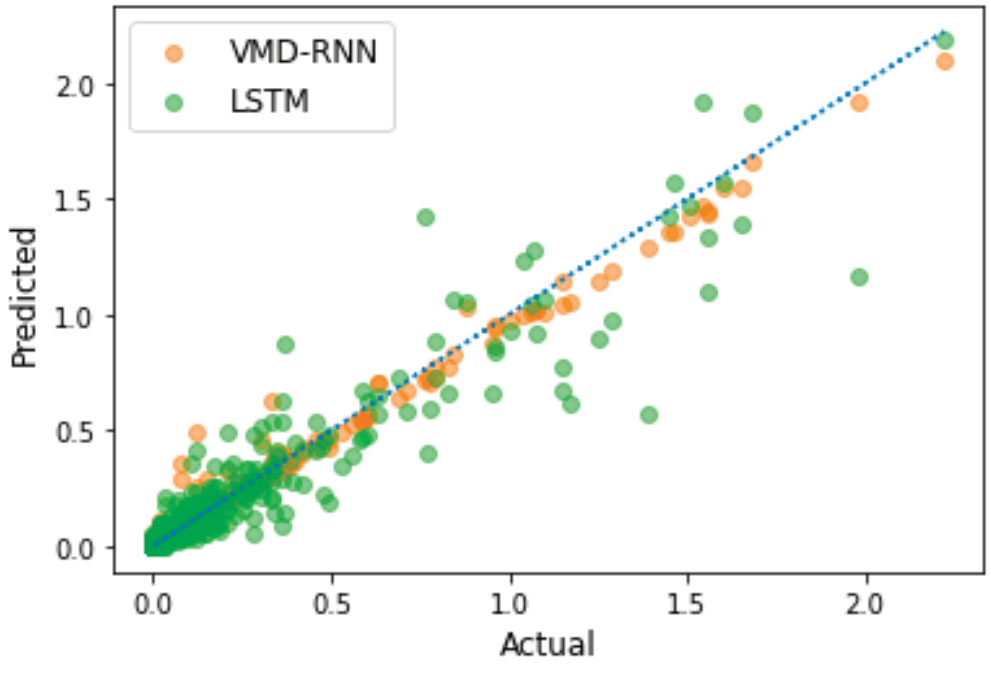
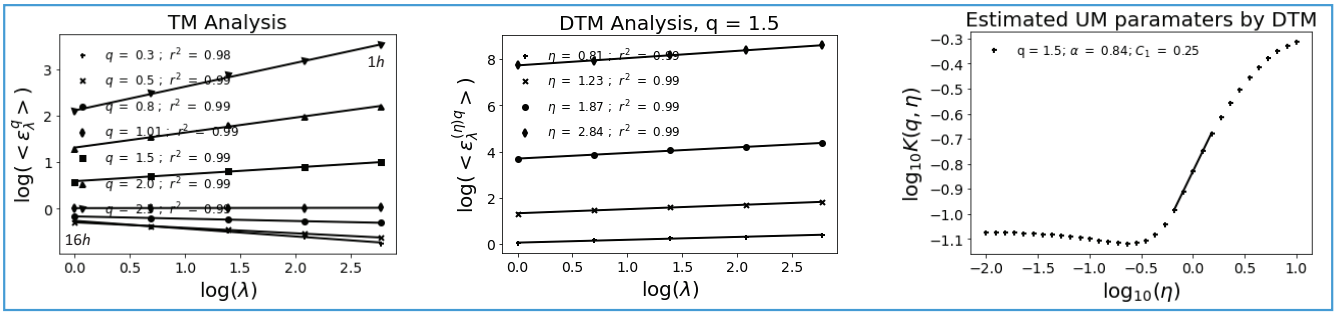
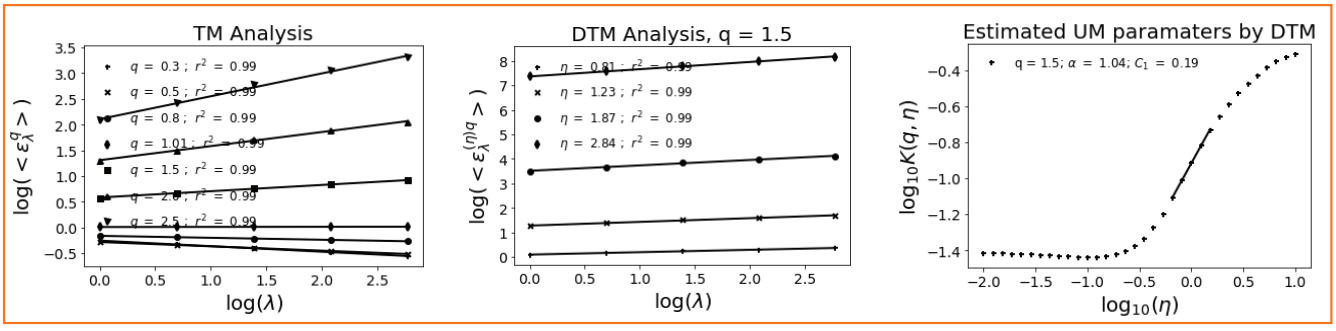


Figure 12. The comparison between predicted and actual hourly rainfall values

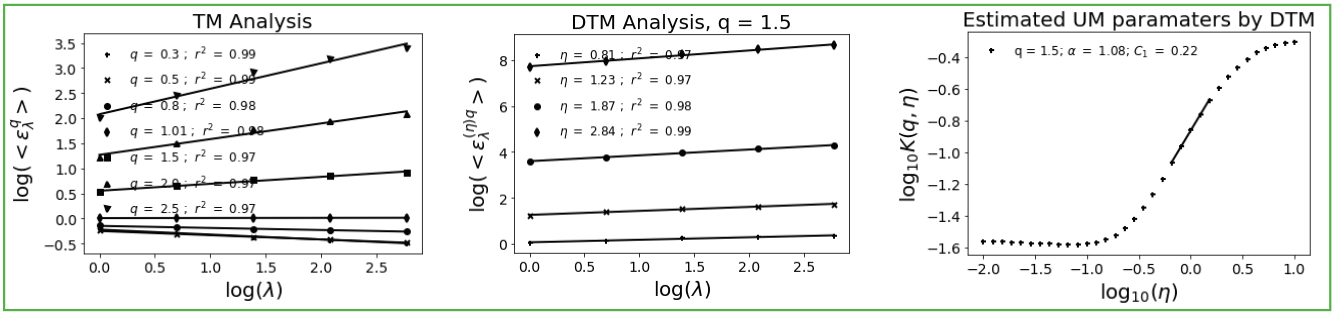
https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024



(a) actual time series



(b) predicted time series by VMD-RNN



(c) predicted time series by LSTM without decomposition

Figure 13. UM results for hourly time series in the testing set

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

List of Tables

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

Table 1. Relevant studies on time series prediction using deep learning models

Reference Evaluation methods	Models	Applications
(Ma et al., 2015) MAPE, MSE	LSTM	traffic speed
(Ding et al., 2019) RMSE, MAE	GRU	wind power
(Gauch et al., 2021) NSE	multi-timescale LSTM	daily and hourly rainfall-runoff
(Ni et al., 2020) RMSE, NSE, MARE	wavelet-LSTM, convolutional LSTM	monthly streamflow and rainfall
(Barrera-Animas et al., 2022) RMSE, MAE, RMSLE	Stacked-LSTM, Bidirectional-LSTM	hourly rainfall time series
(He et al., 2022) RMSE, NSE, MAE, Accuracy	STL-ML	daily rainfall time series
(Hadi and Tombul, 2018) RMSE, NSE	ANN with wavelet transformation	daily streamflow
(Devi et al., 2020) MAE, RMSE, MAPE, MASE	EEMD-CSO-LSTM-EFG	hourly wind power
(He et al., 2019) MAE, RMSE, NSE	VMD-DNN	daily runoff
(Xie et al., 2019) MAE, RMSE, NSE	VMD-DBN-IPSO	daily runoff series
(Zuo et al., 2020) NSE, NRMSE, PPTS	VMD-LSTM	daily streamflow
this study* RMSE, MAE, MAPE, UM	VMD-RNN	daily and hourly rainfall

* This study incorporates four RNN models, namely LSTM, GRU, Bidirectional LSTM, and Bidirectional GRU. The RNN model with superior architecture was selected for each subsequence.

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

Table 2. Variant RNN models of IMF1-IMF8

VMD component	Model type	Numbers of input	Model structure
IMF1	GRU	5	128-128
IMF2	BiLSTM	15	64
IMF3	BiGRU	15	64-64-64
IMF4	LSTM	10	64
IMF5	LSTM	10	64-64-64
IMF6	BiLSTM	15	64
IMF7	BiLSTM	10	128-128
IMF8	BiGRU	15	32-32

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

Table 3. Prediction errors for daily time series in the testing set

	MAE	RMSE	MAPE
VMD-RNN	0.726	0.852	9.853
LSTM	6.825	2.612	10.475

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

Table 4. Estimated UM parameters for daily time series in the testing set

	TM		DTM	
	α	C_1	α	C_1
Actual	0.89	0.25	1.02	0.24
VMD-RNN	0.98	0.16	1.06	0.16
LSTM	1.11	0.17	1.23	0.16

https://doi.org/10.5194/egusphere-2023-2710
Preprint. Discussion started: 10 January 2024

Table 5. Estimated UM parameters for hourly time series in the testing set

	TM		DTM	
	α	C_1	α	C_1
Actual	0.55	0.26	0.84	0.25
VMD-RNN	0.79	0.21	1.04	0.19
LSTM	0.97	0.22	1.08	0.22