Prediction of soil moisture using BiGRU-LSTM model with STL decomposition in Qinghai–Tibet Plateau

Biao Zhang [1] , **Tonglin Luo** [2] , **Xuchu Jiang** [Corresp. 2]

[1] School of Computer Science, Liaocheng University, LIAO CHENG, CHINA

[2] School of Statistics and Mathematics, Zhongnan University of Economics and Law, WUHAN, HUBEI, CHINA

Corresponding Author: Xuchu Jiang
Email address: xuchujiang@zuel.edu.cn

The Ali Network data based on the Tibetan Plateau can provide representative coverage of the climate and surface hydrometeorological conditions in the cold and arid region of the Qinghai-Tibet Plateau (QTP). Among them, the plateau soil moisture can effectively quantify the uncertainty of coarse resolution satellite and soil moisture models. Aiming at constructing a soil moisture prediction model for the QTP, this paper proposes a combined prediction model based on time series decomposition and a deep neural network. First, the model is preprocessed and decomposed by seasonal and trend decomposition using loess (STL), and the trend component, seasonal component and random remainder component of the original time series are gained in an additive way. Then, a bidirectional gate recurrent unit (BiGRU) model was used for the trend items, and a long short-term memory artificial neural network (LSTM) model was used to extract the fitting time sequence information for the seasonal component and the remainder component. Finally, the predicted value of the plateau soil moisture content sequence was output by the model. Based on the hourly data of soil moisture content at a depth of 5cm collected from the AL02 site of Ali Network on the QTP during 2012-2016, the model RMSE was 0.01936 and adjusted R^2 to 0.99330. It is significantly better than LSTM without STL decomposition and models with more complex structures, such as attention mechanisms or convolutional neural network (CNN) filters. At the same time, the model is better than the single STL-RNN, STL-BiGRU or STL-LSTM, which proves the effectiveness and accuracy of the combined model proposed in this paper and shows the feasibility of the deep learning method in the prediction of soil moisture in the plateau.

Prediction of soil moisture using BiGRU-LSTM model with STL decomposition in Qinghai–Tibet Plateau

Biao Zhang[1], Tonglin Luo [2], Xuchu Jiang[2*]

[1] School of Computer Science, Liaocheng University, Liaocheng, 252059, China

[2] School of Statistics and Mathematics, Zhongnan University of Economics and Law, Wuhan, 430073, China

*Correspondence: z0004994@zuel.edu.cn

Abstract: The Ali Network data based on the Tibetan Plateau can provide representative coverage of the climate and surface hydrometeorological conditions in the cold and arid region of the Qinghai-Tibet Plateau (QTP). Among them, the plateau soil moisture can effectively quantify the uncertainty of coarse resolution satellite and soil moisture models. Aiming at constructing a soil moisture prediction model for the QTP, this paper proposes a combined prediction model based on time series decomposition and a deep neural network. First, the model is preprocessed and decomposed by seasonal and trend decomposition using loess (STL), and the trend component, seasonal component and random remainder component of the original time series are gained in an additive way. Then, a bidirectional gate recurrent unit (BiGRU) model was used for the trend items, and a long short-term memory artificial neural network (LSTM) model was used to extract the fitting time sequence information for the seasonal component and the remainder component. Finally, the predicted value of the plateau soil moisture content sequence was output by the model. Based on the hourly data of soil moisture content at a depth of 5cm collected from the AL02 site of Ali Network on the QTP during 2012-2016, the model RMSE was 0.01936 and adjusted R^2 to 0.99330. It is significantly better than LSTM without STL decomposition and models with more complex structures, such as attention mechanisms or convolutional neural network (CNN) filters. At the same time, the model is better than the single STL-RNN, STL-BiGRU or STL-LSTM, which proves the effectiveness and accuracy of the combined model proposed in this paper and shows the feasibility of the deep learning method in the prediction of soil moisture in the plateau.

Key words: soil moisture; time series prediction; STL decomposition; BiGRU; LSTM

1 Introduction

1.1 Background

As the highest plateau in the world, the Qinghai-Tibet Plateau (QTP) is an important ecological security barrier for the world, playing many roles in water conservation and biodiversity protection. As an important indicator of surface hydrological information, soil moisture plays an important role in regional energy and the land water cycle [1] and is an important parameter in hydrological, meteorological and environmental studies. Its temporal variation and spatial distribution regulate the pattern, diversity and succession characteristics of vegetation [2]. The main grassland type on the QTP is alpine grassland, and the soil moisture in the root layer is mainly affected by rainfall recharge factors. Therefore, an in-depth understanding of soil water dynamics is helpful to better understand soil water maintenance and predict the potential impact of future rainfall pattern changes on key processes of alpine steppe ecosystems [3]. It is of great significance to study the spatial and temporal variation pattern of surface soil moisture on the QTP and build a soil moisture prediction model based on long-term time series data for the study of alpine grassland ecological carrying capacity, ecological construction of grassland restoration and reconstruction, and meteorological disaster monitoring in the QTP.

1.2 Literature review

Time series generated by complex systems are ubiquitous in astronomy, hydrology, meteorology, environment, finance and other fields. Traditionally, time series in this field are often modeled using numerical models or traditional statistical methods, and predictions are made. Among them, traditional statistical learning modeling methods for the development of time series, namely, modern time series analysis, first came from the autoregressive (AR) model proposed by British statistician G.U. Yule in 1927. Beginning in the 1970s, the autoregressive integrated moving average (ARIMA) became a central topic for time series analysis. In the field of natural ecology, Tan and Zheng [4] used the ARIMA model to conduct a thorough study on the change trend of the ecological footprint of water resources in China. The results of the ARIMA (2, 1, 3) model showed that from 2008 to 2012, the per capita ecological footprint of water resources in China will continue to increase, and the water crisis will become increasingly severe. Zhou W et al. [5] used the differential integrated moving average autoregression model and Holt-Winters exponential smoothing model to predict the surface subsidence in mining areas based on the induced ordered weighted average (IOWA) operator. In addition, modern numerical models have also been widely applied in this field. Su Z et al. [6], on the basic framework given by the European Centre for Medium-Range Weather Forecasts (ECMWF), used a series of interpolation methods and the current pointwise extended Kalman filter scheme to establish a numerical prediction model for soil moisture content in the QTP, which has obvious performance improvement compared with the old model.

However, with the development of sensors and Internet of Things technology, the sampled data from complex systems show multivariable and large-scale characteristics. At the same time, affected by system evolution and external interference, the data present characteristics such as nonstationarity and noise [7]. Traditional mathematical modeling methods have difficulty characterizing such complex relationships, and satisfactory results cannot be obtained in complex system modeling tasks. At this time, the application of relevant methods and technologies of machine learning and deep learning for time series analysis and prediction has become a research hotspot. Support vector machine (SVM) is a machine learning method based on statistical learning theory. Kim K et al. [8] used a support vector machine to predict the stock price index and tested the feasibility of the support vector machine for time series prediction through comparative experiments. Qing C et al. [9] proposed a new multifactor precipitation prediction model by integrating a time series model and support vector regression and accurately predicted summer precipitation in the Chifeng region. The recurrent neural network (RNN) in deep learning is a special kind of neural network that can store and extract dynamic information in time series through internal self-circulating neurons, and it is very suitable for processing time series data in ecosystems. However, the classical RNN model has the problem of gradient vanishing and gradient explosion, which makes it difficult to effectively utilize long-distance time series information. Long Short-Term Memory neural network (LSTM) solves this problem. It has the same structure as the standard RNN model, but it has a more refined internal processing unit. Kratzert F et al. [10] showed the potential of the LSTM as a regional hydrological model. The results proved that LSTM realized the long-term storage and updating of the state of the basin. Yang X et al. [11] combined and applied Particle Swarm Optimization-LSTM (PSO-LSTM) and Bidirectional LSTM (BiLSTM) models to the precipitation and air temperature data to predict the glacially derived runoff. The results presented in [11] provided a deeper understanding and a more appropriate method of predicting the glacially derived runoff in glacier-fed river basins. As a simplification and improvement of LSTM, gate recurrent unit (GRU) has achieved better results on some specific problems. Gao S et al. [12] proved that GRU and LSTM had similar performance in short-term river runoff prediction, while the GRU model had fewer model parameters and training calculations. Wang Q et al. [13] further explored the potential of the GRU model by introducing regional factors into the model and conducting multistep prediction, and the final model achieved good results.

Different from the above modeling of complex time series using a single model, many recent studies show that, especially in such complex nonstationary time series data, decomposition-based models have better performance than a single model. The common decomposition methods include Fourier transform (FT), wavelet decomposition (WD), empirical mode decomposition (EMD) and seasonal-trend decomposition using LOESS (STL). Li D et al. [14] noted that different decomposition methods are applicable to different data characteristics and fields. FT, WD and other methods often have strict mathematical assumptions, which limits the wide application of these methods. EMD and its derived CEEMDAN and other methods are completely based on the data-driven idea, but they often have problems such as modal aliasing or incomplete decomposition of random factors [15]. As a statistical method, STL decomposition has good adaptability to all kinds of time series data with different properties. The model based on STL decomposition has been applied to the prediction of time series of many complex systems. Ding J et al. [16] combined STL with a random forest model (RF) to investigate the influence of meteorological factors and precursor emission changes on ozone concentration. Based on the existing STL and LSTM models, Xu Z et al. [17] specifically processed and optimized the sequence boundary of runoff prediction, thus building a framework called SDIPBC. Qin L et al. [15] used the grasshopper optimization algorithm when STL was applied to passenger flow prediction, and the performance was improved to a certain extent.

By combining the existing research achievements, it can be found that the current time series modeling methods of scholars can be mainly divided into three aspects: the traditional mathematical modeling method, the machine learning method and the deep learning method. However, in the field of hydrology and meteorology in the QTP region, there are few studies on the analysis and prediction of long-term soil water content data using STL decomposition or deep learning methods. In this paper, based on STL decomposition, BiGRU and LSTM models were used to build a combined plateau soil moisture prediction model, and the prediction effects were compared and analyzed to verify the effectiveness and accuracy of deep learning-related methods in the analysis and prediction of long-term plateau soil moisture data.

2 Data sources and research methods

2.1 Data sources and data preprocessing

The experiment to choose the soil moisture measured data from the National Qinghai-Tibet Plateau Scientific Data Center (http://dx.doi.org/10.11888/Soil.tpdc.270028) included the observation data of soil temperature and humidity of the QTP. The observational data in this data set consist of four in situ reference networks at regional scales, namely, the Naqu, Maqu, Ali and Pari networks with different climatic and vegetation types. The Ali network, which includes Ali and Shiquanhe, is in the southwest arid region of the QTP and mainly consists of desert steppe (Figure 1). At each station of Ali Network, soil moisture content with an accuracy of 10^{-5} is recorded hourly at depths of 5, 10, 30, 50 and 80 cm. Based on previous research experience [18], it is known that microwave data can only reflect the surface soil moisture of a few centimeters, and considering that there is a large number of missing observational data of all sites of Ali Network before 2011, in this paper, soil moisture observation data recorded by the soil moisture sensor at a depth of 5cm at the AL02 site of Ali Network every one hour between 2012 and 2016 were used for research.

This paper divides the data set according to the experience ratio of the training set and the test set of 8:2. Since the original data are time series data, the data are divided into the training set and the test set by taking 2015-9-16 0:00 as the partition node. Visualization of the training set and test set data is shown in Figure 2. Finally, the sequence was normalized to map it to the interval [-1,1].

2.2 Research methods

2.2.1 STL decomposition

The STL decomposition proposed by Cleveland R B et al. [19] decomposes the time series into trend, seasonal and remainder components. STL decomposition has good generality and robustness and is applicable to time series data of various cycles or frequencies. The core of the algorithm is to extract the seasonal trend information contained in the time series more accurately by introducing local regression smoothing. STL decomposition represents the original sequence in the additive way as Equation (1):

$$x_t = T_t + S_t + R_t \quad (t = 1,2,3,\ldots,N) \#()$$

where T_t is the trend term, S_t is the seasonal term, and R_t is the remainder term.

The iterative process of the STL decomposition algorithm can be briefly described as follows:

1) Set the initial iteration value: $k = 0, T_t^k = 0$.

2) Detrending: $x_t - T_t^k$.

3) Carry out smoothing on each detrended periodic subsequence, and the sequence obtained by combining all periodic subsequences is denoted as C_t^{k+1}.

4) For C_t^{k+1}, low-pass filtering is carried out using the three times sliding average and once LOESS smoothing, L_t^{k+1} is obtained.

5) Calculate the seasonal terms: $SS_t^{k+1} = C_t^{k+1} - L_t^{k+1}$.

6) Calculate the trend term: The trend term T_t^{k+1} is obtained by LOESS smoothing $x_t^k - S_t^{k+1}$.

7) If T_t^{k+1} converges or reaches the maximum number of iterations, the iteration terminates; otherwise, go back to step 2).

The decomposition process of STL is mainly controlled by parametersn_p, n_s andn_t. The parameter n_p is the cycle length in the sequence, and the smoothing parameter of the periodic subsequence n_s is the parameter of the process in the third step. Generally, an odd number that is slightly larger than the number of cycles contained in the original sequence is taken. The trend smoothing parameter n_t is the parameter of the LOESS process in the sixth step. Cleveland R B suggests a minimum odd number greater than $\frac{1.5n_p}{1-1.5/n_s}$ in [19].

2.2.2 LSTM

The LSTM model is a kind of RNN model that was first proposed by Hochreiter and Schmidhuber in 1997 [20], which can solve the gradient disappearance and gradient explosion problems faced by RNNs in the process of long time series [21],which is specifically designed to avoid the long-term dependence problem (Figure 3). Compared with the traditional RNN model, the LSTM model can perform better in a longer time series. The hidden layer of the original RNN has only one state, so it is very sensitive to short-term input. The LSTM model adds another state based on the RNN, which is used to store the long-term state, called the cell state.

At the present moment, LSTM has three inputs: the current input value x_t, the output value of the LSTM at the previous moment h_{t-1} and the cell state of the LSTM at the previous moment C_{t-1}. There are two outputs: the LSTM output value at the current moment h_t and the cell state at the current moment C_t.

LSTM implements this mode through three gating mechanisms in the algorithm, namely, the input gate, forget gate and output gate. The input gate and output gate are used to receive, output, and correct parameters. The input gate determines how much of the network's input x_t is saved to the cell state at the current time. The output gate determines how much of the cell state C_t is output to the current output value h_t of the LSTM. The forget gate determines how much of the cell state of the previous moment C_{t-1} is retained to the cell state of the current moment C_t.

The LSTM determines the final output value h_t as Equations (2)-(5). First, it calculates the activation state value f_t of the forget gate at the current moment t:

$$f_t = \sigma\left(W_f \otimes (X_t h_{t-1}) + b_i\right) \# ()$$

where $\sigma(\cdot)$ is the sigmoid function and $\otimes$ represents dot multiplication. After the vector is multiplied by the weight matrix, it is transformed by the activation function as a gated state.

Then, calculate the value of the input gate i_t and the value of the candidate state of the input cell $\tilde{C}_t$ at moment t:

$$i_t = \sigma\left(W_i \otimes (X_t h_{t-1}) + b_i\right)$$

$$\tilde{C}_t = \sigma(W_i \otimes (X_t h_{t-1}) + b_i \# ()$$

The updated value $\tilde{C}_t$ of the cell state under the current time t can be obtained from the above calculation:

$$C_t = f_t \otimes C_{t-1} + i_t \otimes \tilde{C}_t \# ()$$

Finally, calculate the current output value of the output gate according to the update value of the cell state at the current time t:

$$O_t = \sigma(W_0 \otimes (X_t h_{t-1}) + b_0)$$

$$h_t = O_t \otimes \tanh(C_t) \# ()$$

2.2.3 BiGRU

GRU is a simplification of the LSTM model proposed by Cho et al. [22] in 2014. The LSTM model effectively alleviates the problem of gradient disappearance in the traditional RNN model. However, the shortcomings of the LSTM model, such as complex parameters and difficult training, are gradually exposed, restricting the further application of LSTM. The GRU redesigns the internal structure of the LSTM unit based on the gating idea, thus reducing the computation time and training complexity.

Similar to the LSTM model, for the input sequence $\{x_1,x_2,x_3,\ldots,x_t,\ldots x_n\}$, the GRU can successively obtain its hidden layer state h_t at time step t according to Equations (6)-(9):

$$r_t = \sigma\left(W_r x_t + b_r + W_{hr} h_{t-1} + b_{hr}\right) \#()$$

$$z_t = \sigma\left(W_z x_t + b_z + W_{hz} h_{t-1} + b_{hz}\right) \#()$$

$$n_t = \tanh\left(W_n x_t + b_n + r_t \otimes \left(W_{hn} h_{t-1} + b_{hn}\right)\right) \#()$$

$$h_t = (1 - z_t) \otimes n_t + z_t \otimes h_{t-1} \#()$$

where h_{t-1} is the hidden layer state of time step $t-1$, r_t, z_t, n_t is the gated state updated at each time step, $\sigma(\cdot)$ is a sigmoid function, and b is the bias term.

BiGRU (bidirectional GRU) builds two reverse GRU models at the same time, modeling time sequence information forward and backward, and the output of each time step is the concatenation of the output of the two GRU models. It is generally believed that the BiGRU model can better extract the front and back dependencies in time series and has a better effect for sequences with a certain front or back correlation [23].

2.2.4 Combined prediction model

Figure 2 shows that the observed data of soil moisture have a very significant seasonal variation rule with a one-year cycle. Soil moisture in summer is much higher than that in the other three quarters, and the peak value of soil moisture in summer has a trend of gradual increase with the passage of time. Based on the nature of plateau soil moisture time series data, this paper combined STL decomposition with the BiGRU model and LSTM model and proposed a new neural network combination prediction model based on STL decomposition to make use of the information extraction ability of STL decomposition and the time series fitting ability of the neural network model simultaneously. The overall framework of the model is shown in Figure 4.

Based on a series of data preprocessing, the model first extracts the trend change information and periodic change information contained in the data through STL decomposition, and the original sequence is decomposed into the trend component, seasonal component and remainder component. During decomposition, to avoid data leakage and prove the effectiveness of the model, the subsequence as a training set was first decomposed alone, and then the whole sequence was decomposed to obtain the test set. Then, the BiGRU model is used for the obtained trend component, and an LSTM model is used to fit the timing information for the seasonal component and the remainder component. Finally, the combined model extracts the hidden layer state of the last time step of each cyclic neural network model and outputs the predicted values of the three components through a fully connected layer. STL decomposed the sequence in an additive way, which made it convenient to model the three components independently. The predicted values of the three components were added to obtain the final prediction results for the plateau soil moisture content.

3 Experimental analysis

3.1 Performance metrics

In this experiment, the root mean square error (RMSE), mean absolute error (MAE) and adjusted goodness of fit (adjusted R^2) were used to compare the experimental results output by each model and judge the model's performance. Smaller values of RMSE and MAE indicate higher model accuracy. The closer R^2 is to 1, the higher the prediction accuracy of the model is, and the adjusted R^2 eliminates the influence of sequence length and the number of features in the model on the index so that the R^2 of different models can be compared with each other. The calculation formulas of RMSE, MAE, and adjusted R^2 are shown in Equation (10), Equation (11) and Equations (12)-(13), respectively.

$$RMSE = \sqrt{\frac{1}{m}\sum_{i=1}^{m}(y_{test}^{(i)} - \hat{y}_{test}^{(i)})^2} = \sqrt{MSE} \#()$$

$$MAE = \frac{1}{m}\sum_{i=1}^{m}\left|y_{test}^{(i)} - \hat{y}_{test}^{(i)}\right| \#()$$

$$R^2 = 1 - \frac{\sum_i(\hat{y}^{(i)} - y^{(i)})^2}{\sum_i(\bar{y} - y^{(i)})^2} \#()$$

$$\text{Adjusted } R^2 = 1 - \frac{(1 - R^2)(n - 1)}{n - p - 1} \#()$$

where $y^{(i)}$, $\hat{y}^{(i)}$ and $\bar{y}$ represent the true value, the model estimated value and the sample sequence mean, respectively. n is the sequence length, and p is the number of features in the model.

3.2 Experimental environment and parameter setting

The experimental environment adopted in this paper is an Intel Xeon 8358P 2.6 GHz CPU and NVIDIA RTX A5000 GPU, and the model is built based on PyTorch under Python 3.8.

The early stop mechanism is introduced in the first pretraining. When the training model loss function is without gain in 10 iterations, the iteration will be stopped. This measure can not only ensure the fitting accuracy of the model but also effectively prevent overfitting and save the training time of the model. The results of pretraining show that the model generally achieves the optimal effect when the iteration is approximately 80 times. Therefore, the training cycle is set as 100 in the subsequent experiment in this paper. The results of pretraining also show that due to the powerful fitting ability of BiGRU and LSTM models, the model with a simple structure can already achieve sufficient fitting ability under the problem studied in this paper, while the overly complex model structure will make the performance worse. To make the model obtain as much historical information as possible and exclude too much noise at the same time, the prediction window size was set as one year, that is, 24×365 hours. Based on various considerations, the main super parameters and training parameters set in the model training process are shown in Table 1.

3.3 Experimental results and analysis

3.3.1 STL decomposition results

The plateau soil moisture data used in this study have an obvious annual cycle, and the data sampling frequency is once per hour. Therefore, the cycle length parameter n_p is set as 24×365 hours, and the parameter n_s is set to 7, which is slightly larger than the number of cycles contained in the data. The parameter n_t is determined according to the empirical rule described in Section 2.2.1. The three components obtained by STL decomposition are shown in Figure 5.

According to the decomposition results, the STL algorithm can adequately extract the trend and periodic information contained in the sequence, and the seasonal term clearly shows the periodic variation in soil moisture in the plateau. The remainder sequence has a mean value of 0 and fluctuates randomly nearby, which also proves that the STL decomposition adopted is effective. It can also be seen from Figure 5 that the plateau soil moisture showed an increasing trend during 2012-2016, but there was a low trough during 2014-2015.

3.3.2 Prediction performance of the combined model

Figure 6 shows the prediction effect of the model on the three components and the resulting plateau soil moisture series on the test set. Figure 6(a) shows the fitting effect of the model on the sequence, and the residuals of the model on each component and the total sequence at each time are shown in the bar plot. Figure 6(b) is the comparison between the predicted value and the observed value. When the data points are scattered as much as possible along the diagonal line representing the completely accurate measurement, the prediction accuracy of the model is higher.

As seen from the Figure 6, the model has achieved a good fitting prediction effect on the plateau soil moisture content data. Except for a few anomalies, the remainder column in Figure 6 is very short and converges near zero. For special sections, such as abrupt points and peak values, the model also gives accurate predicted values, which proves that the model can effectively extract the information contained in the plateau soil moisture content sequence and show good robustness to various situations.

In the experiment, the model of the three components was built and fitted. The comparative experimental data in Table 2 and Figure 7 show that for the trend component, the BiGRU model used in this paper is the best, while for the seasonal component and the remainder component, the adopted LSTM model has the best performance.

For the overall plateau soil moisture content sequence, the STL-RNN, STL-GRU and STL-BiLSTM models were selected in this paper as single models under the premise of STL decomposition for performance comparison. STL-CNN-BiGRU and STL-LSTM-Attention were selected as representatives of more complex structural models for comparison, and LSTM was selected as a model without STL decomposition for comparison. The evaluation index values of each model are shown in Table 3 and Figure 8.

The combined model proposed in this paper achieved the best performance among all comparison models, and the RMSE decreased by 4.72%, 3.78% and 10.95% compared with the single models STL-RNN, STL-GRU and STL-BiLSTM, respectively. For STL-CNN-BiGRU, STL-LSTM-Attention with more complex structures and LSTM without STL decomposition, the combined model has a more obvious performance improvement, and the RMSE decreased by 63.38%, 25.88% and 28.27%, respectively.

In practice, the multistep prediction effect of the model is of more important significance. In this paper, the two comparison models with the best performance in the single-step prediction and the LSTM model without STL decomposition processing are taken as references to investigate the effects of the proposed model under different prediction horizons. The experimental results are shown in Table 4 and Figure 9.

The combined model proposed in this paper achieves the best performance under each prediction horizon. The RMSE of the combined model increased by 5.84%, 2.59%, 1.43% and 7.52% at 2 h, 8 h, 16 h and 24 h compared with the single model after STL decomposition and increased by 9.80%, 11.84%, 13.15% and 8.28% compared with the LSTM without STL treatment, respectively. With the prediction horizon expanding, the prediction effect of the combined model has a gradually increasing trend compared with the performance improvement of other models, which proves the feasibility of the model proposed in this paper in practice.

4 Generalized performance analysis

Aiming to further investigate the generalization performance of the proposed model, in this part, the proposed combined model is used to make a fitting prediction of the soil heat flux time series data, and the same comparison model is selected as in Section 3.3. The soil heat flux time series data were obtained from the National Qinghai-Tibet Plateau Scientific Data Center (http://dx.doi.org/10.11888/Meteoro.tpdc.270910). In this paper, observations from the BJ site of the Naqu Station of Plateau Climate and Environment (NPCE-BJ) at a soil depth of 10 cm during 2007-2013 were taken [23]. The data were not missing in the selected time period. The study of Ma et al. [24] shows that this series also has obvious periodic and trend changes, and its data characteristics are similar to the soil moisture content series investigated in this paper, which is suitable to be used as the data set for generalization performance analysis. The experimental results are shown in Table 5 and Figure 10.

The results of the experiment show that the model also shows optimal performance compared with the comparison model on the new data set. Under 1, 2, 8, 16 and 24 h step prediction horizons, the combined model proposed in this paper generally has a performance improvement of 3%-5% compared with the single model in the comparison model. Compared with the more complex STL-CNN-BiGRU or STL-LSTM-Attention, the performance improved by 10-70%. The results of this experiment prove that the model proposed in this paper has strong generalization ability in the study of QTP geography, climate and other fields.

5 Conclusion

Based on the hourly data of soil moisture content at a depth of 5cm collected from the AL02 site of Ali Network over the QTP from 2012 to 2016, a new combined plateau soil moisture prediction model was constructed by using STL decomposition, BiGRU and LSTM, and the prediction effect was compared and analyzed with other structural models. The results show that (1) STL decomposition can effectively extract and separate the long-term trend changes, periodic seasonal changes and random disturbances of soil moisture series on the plateau. (2) A BiGRU and two LSTM models were used to extract and fit the three subsequences obtained by STL decomposition, and the best results were obtained. (3) The RMSE of the combined model proposed in this paper reaches 0.01936, and the goodness of fit of adjustment R^2 reaches 0.99330, which is significantly higher than the LSTM without STL decomposition preprocessing and the neural network combined model with a more complex structure, such as using the attention mechanism or with the CNN layer. (4) The combined model proposed in this paper shows greater advantages in multistep prediction than in single-step prediction, which proves that the STL-based neural network combined model presented in this paper shows high accuracy, robustness and effectiveness for the plateau soil moisture sequence and has high practical application value. It also shows the feasibility of applying the deep learning method to plateau soil moisture prediction or other physical geography fields.

Author Contributions: Conceptualization, T.L.; methodology, X.J.; formal analysis, T.L.; data curation, X.J.; supervision, B.Z.; writing—original draft preparation, T.L.; writing—review and editing, B.Z. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The data used in this article is from the public data set (http://dx.doi.org/10.11888/Soil.tpdc.270028).

Conflicts of Interest: The authors declare no conflicts of interest.

References

[1] Milly P C D, Dunne K A. Sensitivity of the global water cycle to the water-holding capacity of land[J]. Journal of climate, 1994, 7(4): 506-526.

[2] Zhu X C, Shao M A, Zhu J T, et al. Temporal stability of surface soil moisture in Alpine Meadow Ecosystem on Northern Tibetan Plateau[J]. Trans Chin Soc Agric Mach, 2017, 48(8): 212-218.

[3] Xing Y, Jiang Q G, Li W Q, et al. Landscape spatial patterns changes of the wetland in Qinghai-Tibet Plateau[J]. Ecol Environ Sci, 2009, 18(3): 1010-1015.

[4] Tan X J, Zheng Q Y. Dynamic analysis and forecast of water resources ecological footprint in China[J]. Acta Ecologica Sinica, 2009, 29(7): 3559-3568.

[5] Zhou W, Zhang W, Yang Y, et al. A combined model prediction method for surface subsidence monitoring in mining areas[J]. Journal of Geodesy and Geodynamics, 2021, 41(3): 308-312.

[6] Su Z, De Rosnay P, Wen J, et al. Evaluation of ECMWF's soil moisture analyses using observations on the Tibetan Plateau[J]. Journal of Geophysical Research: Atmospheres, 2013, 118(11): 5304-5318.

[7] Han Z, Zhao J, Leung H, et al. A review of deep learning models for time series prediction[J]. IEEE Sensors Journal, 2019, 21(6): 7833-7848.

[8] Kim K. Financial time series forecasting using support vector machines[J]. Neurocomputing, 2003, 55(1-2): 307-319.

[9] Qing C, Xiaoli Z, Kun Z. Research on precipitation prediction based on time series model[C]//2012 International conference on computer distributed control and intelligent environmental monitoring. IEEE, 2012: 568-571.

[10] Kratzert F, Klotz D, Brenner C, et al. Rainfall–runoff modelling using long short-term memory (LSTM) networks[J]. Hydrology and Earth System Sciences, 2018, 22(11): 6005-6022.

[11] Yang X, Maihemuti B, Simayi Z, et al. Prediction of Glacially Derived Runoff in the Muzati River Watershed Based on the PSO-LSTM Model[J]. Water, 2022, 14(13): 2018.

[12] Gao S, Huang Y, Zhang S, et al. Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation[J]. Journal of Hydrology, 2020, 589: 125188.

[13] Wang Q, Zheng Y, Yue Q, et al. Regional characteristics' impact on the performances of the gated recurrent unit on streamflow forecasting[J]. Water Supply, 2022, 22(4): 4142-4158.

[14] Li D, Jiang F, Chen M, et al. Multi-step-ahead wind speed forecasting based on a hybrid decomposition method and temporal convolutional networks[J]. Energy, 2022, 238: 121981.

[15] Qin L, Li W, Li S. Effective passenger flow forecasting using STL and ESN based on two improvement strategies[J]. Neurocomputing, 2019, 356: 244-256.

[16] Ding J, Dai Q, Fan W, et al. Impacts of meteorology and precursor emission change on O3 variation in Tianjin, China from 2015 to 2021[J]. Journal of Environmental Sciences, 2023, 126: 506-516.

[17] Xu Z, Mo L, Zhou J, et al. Stepwise decomposition-integration-prediction framework for runoff forecasting considering boundary correction[J]. Science of The Total Environment, 2022, 851: 158342.

[18] Yan F, Wang Y. Estimation of soil moisture from Ts-EVI feature space[J]. Acta Ecologica Sinica, 2009, 9: 4884-4891.

[19] Cleveland R B, Cleveland W S, McRae J E, et al. STL: A seasonal-trend decomposition[J]. J. Off. Stat, 1990, 6(1): 3-73.

[20] Hochreiter S, Schmidhuber J. Long short-term memory[J]. Neural computation, 1997, 9(8): 1735-1780.

[21] Rakthanmanon T, Campana B, Mueen A, et al. Searching and mining trillions of time series subsequences under dynamic time warping[C]//Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining. 2012: 262-270.

[22] Cho K, Van Merriënboer B, Gulcehre C, et al. Learning phrase representations using RNN encoder-decoder for statistical machine translation[J]. arXiv preprint arXiv:1406.1078, 2014.

[23] Zhu Q, Zhang F, Liu S, et al. A hybrid VMD–BiGRU model for rubber futures time series forecasting[J]. Applied Soft Computing, 2019, 84: 105739.

[24] Ma Y, Hu Z, Xie Z, et al. A long-term (2005–2016) dataset of hourly integrated land–atmosphere interaction observations on the Tibetan Plateau[J]. Earth System Science Data, 2020, 12(4): 2937-2957.

Table 1(on next page)

Table 1 Main parameter settings of the BiGRU and LSTM models

Table 1 Main parameter settings of the BiGRU and LSTM models

Parameter	Value
Predicted time window size	24×365
Batch size	200
Training rounds	100
Number of hidden layer neurons	32
Number of model layers	1
Loss function	MSE
Activation function	ReLU
Optimizer	Adam

Table 2(on next page)

Table 2 RMSE predicted by different models for each component

Table 2 RMSE predicted by different models for each component

	Trend	**Seasonal**	**Remainder**
GRU	0.00018	0.01669	0.00965
BiGRU	**0.00011**	0.01675	0.00962
LSTM	0.00013	**0.01605**	**0.00949**
BiLSTM	0.00015	0.01667	0.00958
RNN	0.00019	0.01814	0.00983
CNN-BiGRU	0.00401	0.04619	0.02450

Table 3 (on next page)

Table 3 Comparison of evaluation indexes of each model

Table 3 Comparison of evaluation indexes of each model

Model	RMSE	MAE	Adjusted R^2
STL-BiGRU-LSTM	**0.01936**	**0.00462**	**0.99330**
STL-RNN	0.02032	0.00501	0.99160
STL-GRU	0.02012	0.00483	0.99276
STL-BiLSTM	0.02174	0.00572	0.99155
STL-CNN-BiGRU	0.05287	0.02138	0.94997
STL-LSTM-Attention	0.02612	0.00830	0.98778
LSTM	0.02699	0.00862	0.98945

Table 4(on next page)

Table 4 RMSE of models with different prediction step sizes

Table 4 RMSE of models with different prediction step sizes

Model	2 h		8 h		16 h		24 h	
	RMSE	Δ	RMSE	Δ	RMSE	Δ	RMSE	Δ
STL-BiGRU-LSTM	**0.02854**	-	**0.06105**	-	**0.08131**	-	**0.09426**	-
STL-LSTM	0.03081	+7.95%	0.06267	+2.65%	0.08248	+1.45%	0.10193	+8.14%
STL-GRU	0.03031	+6.20%	0.06687	+9.53%	0.09194	+13.09%	0.1012	+7.36%
LSTM	0.03164	+10.86%	0.06925	+13.43%	0.09361	+15.14%	0.10277	+9.03%

Table 5(on next page)

Table 5 Experimental results of generalization performance

Table 5 Experimental results of generalization performance

Model	1 h		2 h		8 h		16 h		24 h	
	RMSE	Δ	RMSE	Δ	RMSE	Δ	RMSE	Δ	RMSE	Δ
STL-BiGRU-LSTM	**0.05308**	-	**0.06641**	-	**0.09137**	-	**0.09615**	-	**0.09956**	-
STL-GRU	0.05457	+2.81%	0.06865	+3.38%	0.09402	+2.91%	0.10038	+4.40%	0.10259	+3.05%
STL-RNN	0.05583	+5.18%	0.06960	+4.80%	0.09608	+5.16%	0.09951	+3.49%	0.10394	+4.40%
STL-CNN-BiGRU	0.09413	+77.34%	0.09776	+47.22%	0.10547	+15.43%	0.10686	+11.15%	0.10670	+7.17%
LSTM	0.05613	+5.75%	0.07412	+11.62%	0.09682	+5.97%	0.10280	+6.92%	0.10358	+4.03%
STL-LSTM-Attention	0.07150	+34.70%	0.11533	+73.67%	0.13886	+51.99%	0.13447	+39.86%	0.13973	+40.35%

Figure 1

Figure 1 The location of the AL02 site on AL021

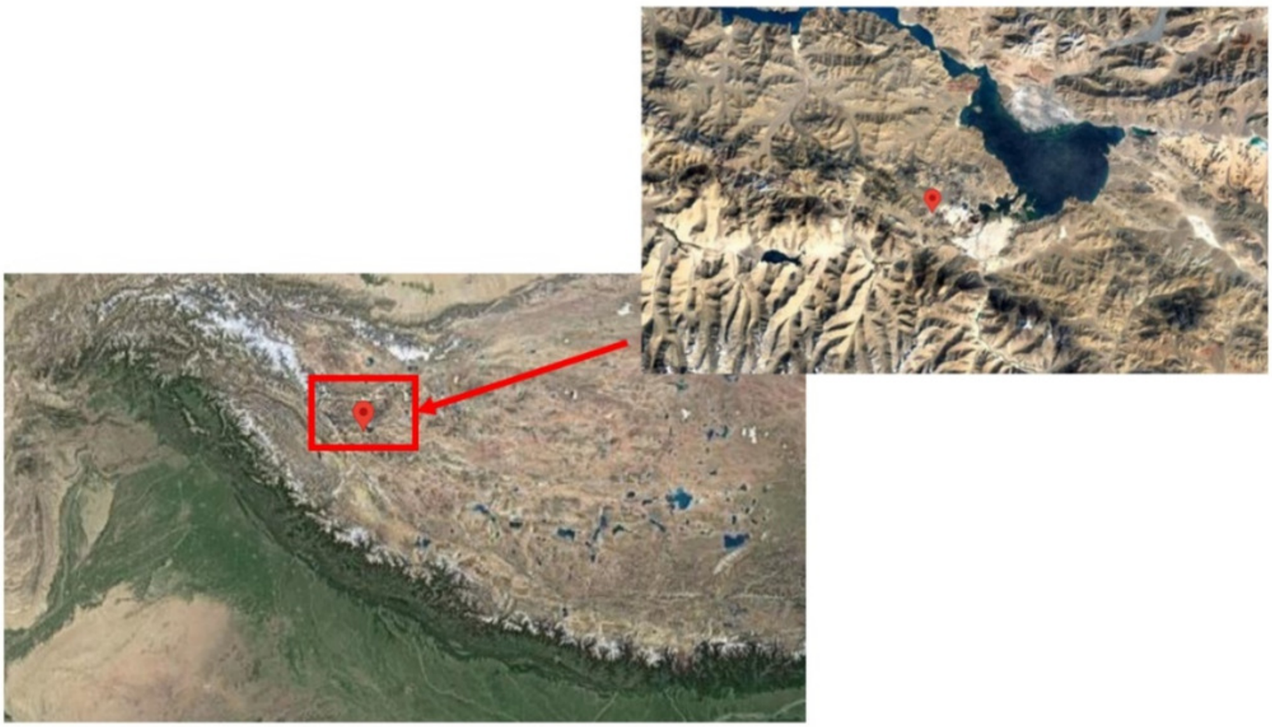

Figure 2

Figure 2 Plateau soil moisture observation sequence and division of the training and test set

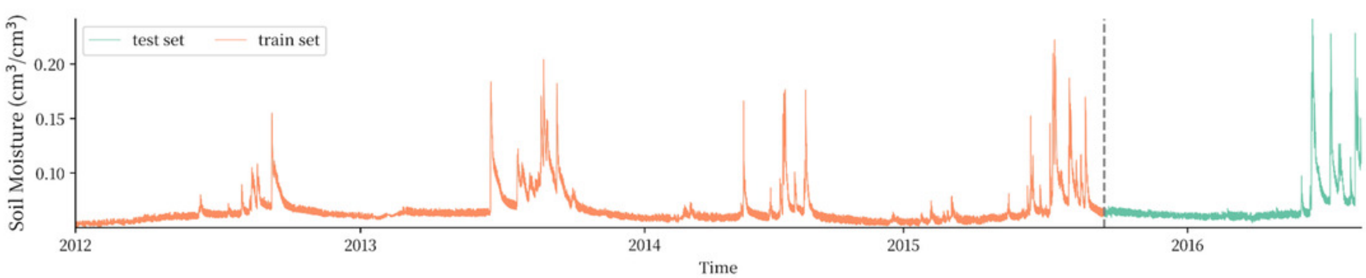

Figure 3

Figure 3 LSTM model expansion diagram

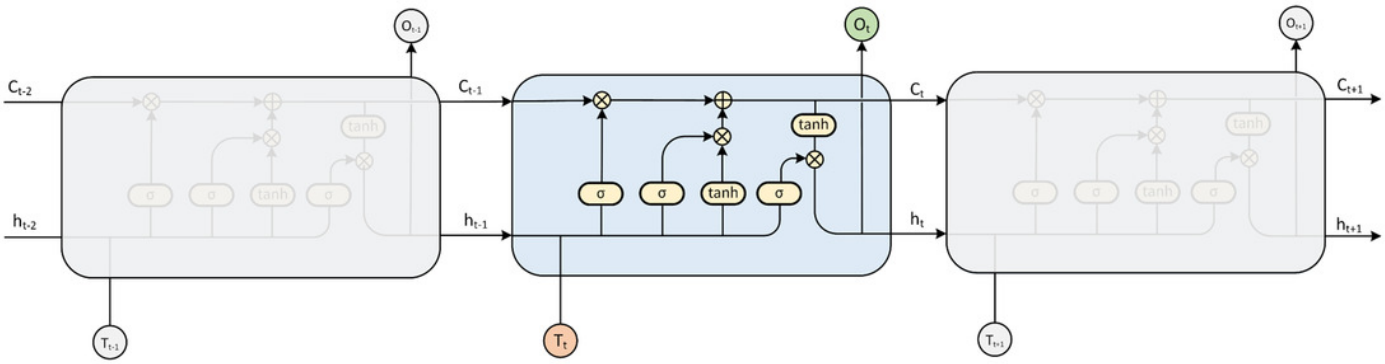

Figure 4

Figure 4 The overall framework of the model

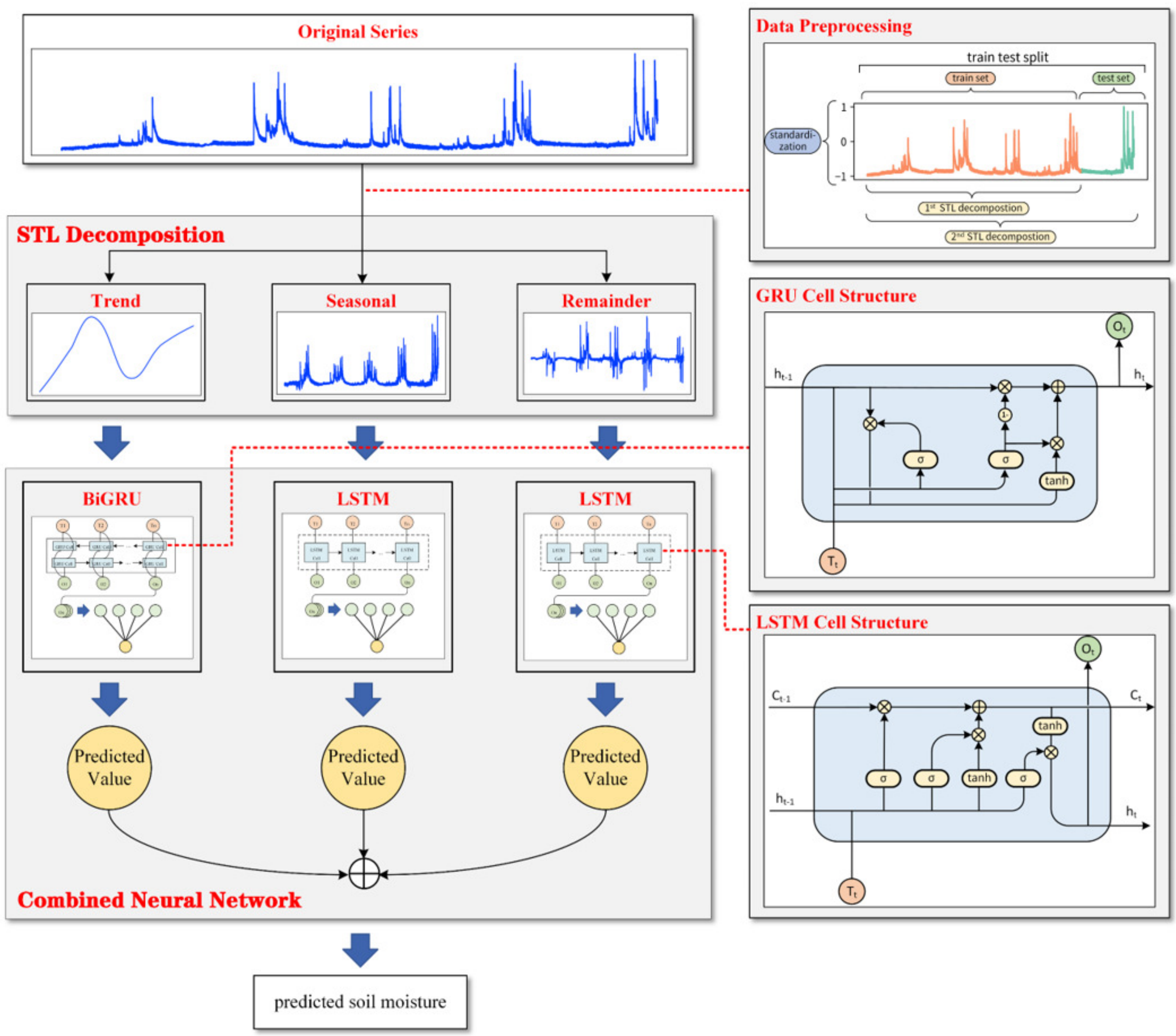

Figure 5

Figure 5 STL decomposition results

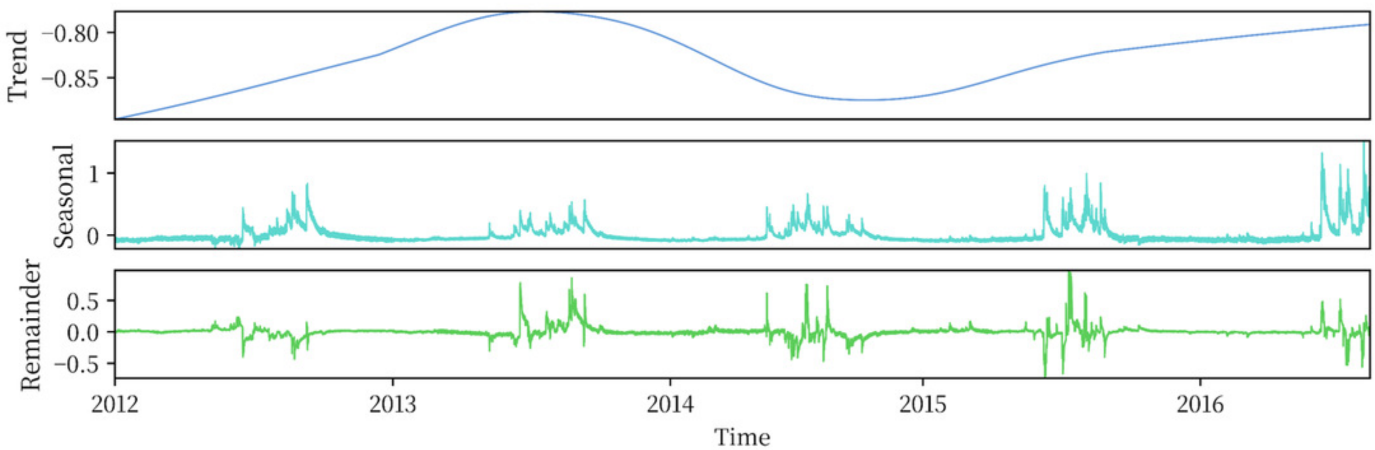

Figure 6

Figure 6 Model prediction performance on the test set

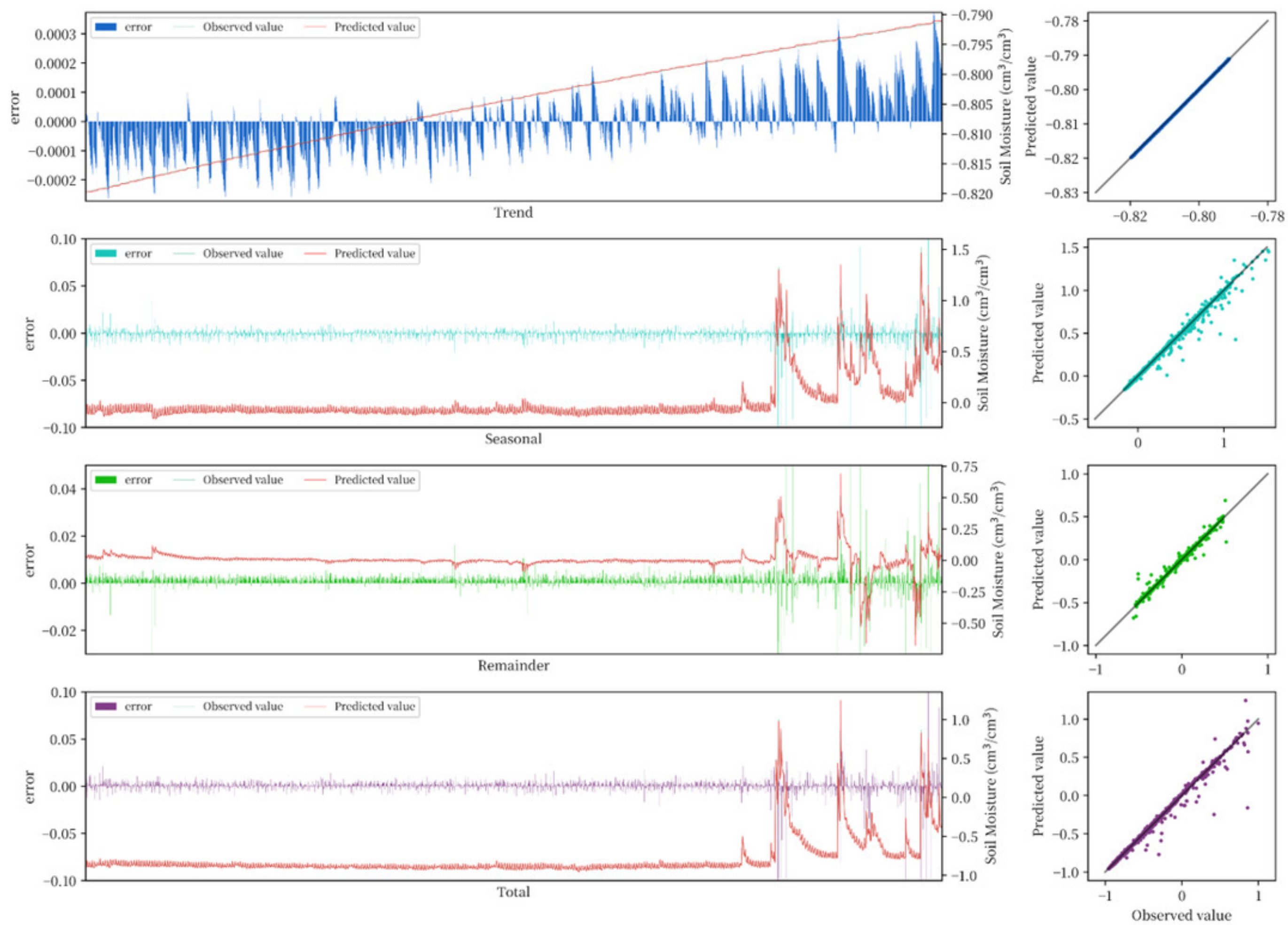

Figure 7

Figure 7 Comparison of the effects of different models on each component sequence

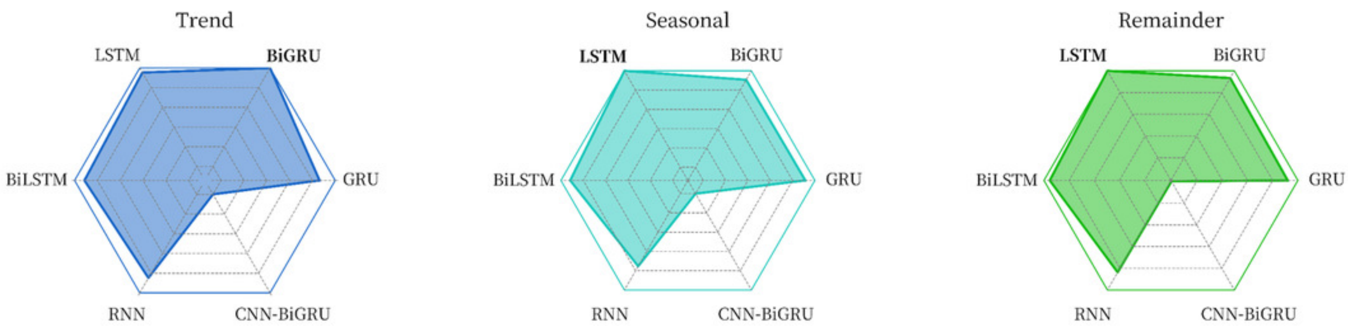

Figure 8

Figure 8 RMSE comparison of each model on the test set

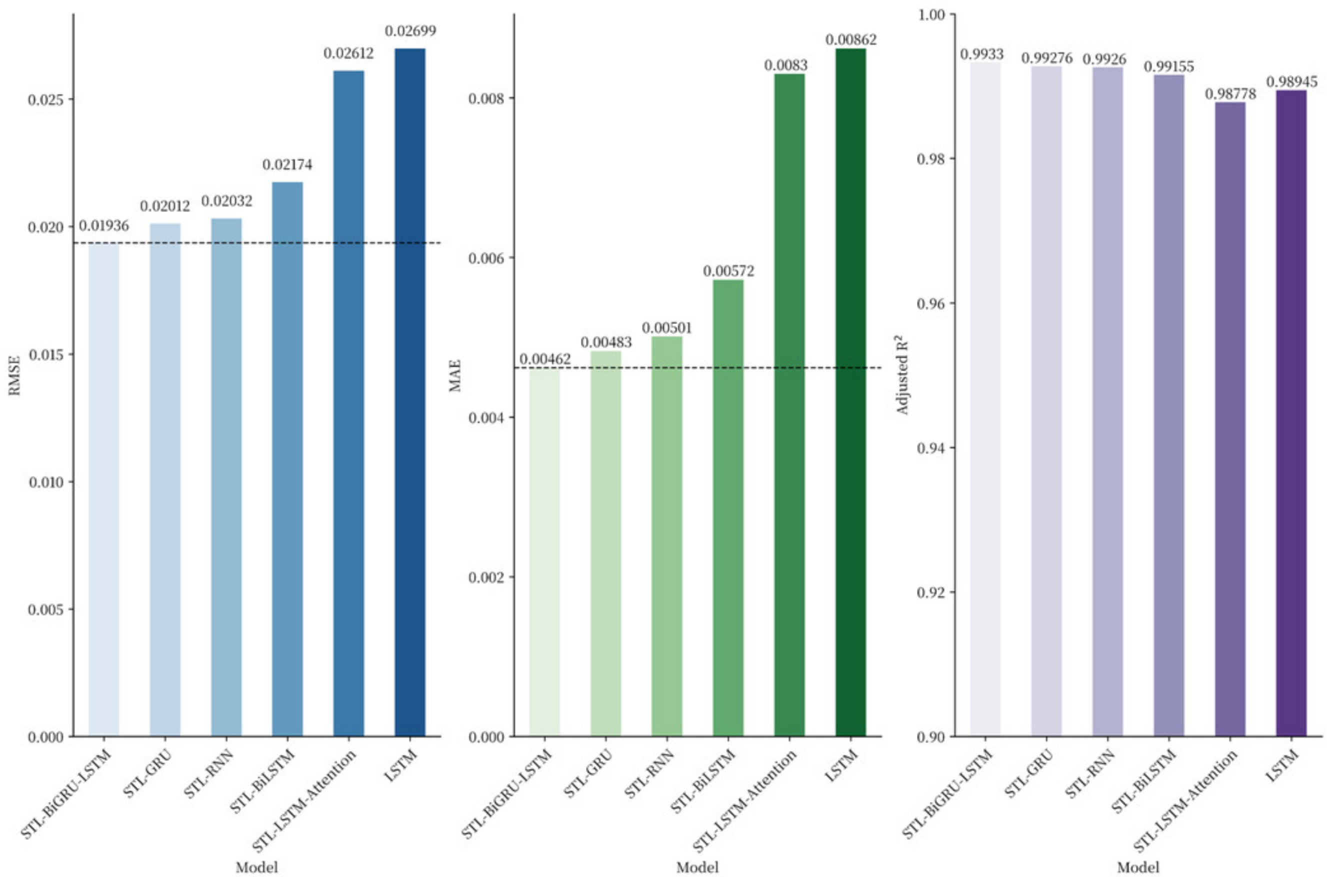

Figure 9

Figure 9 RMSE and improvement percentage of the model under different predictions

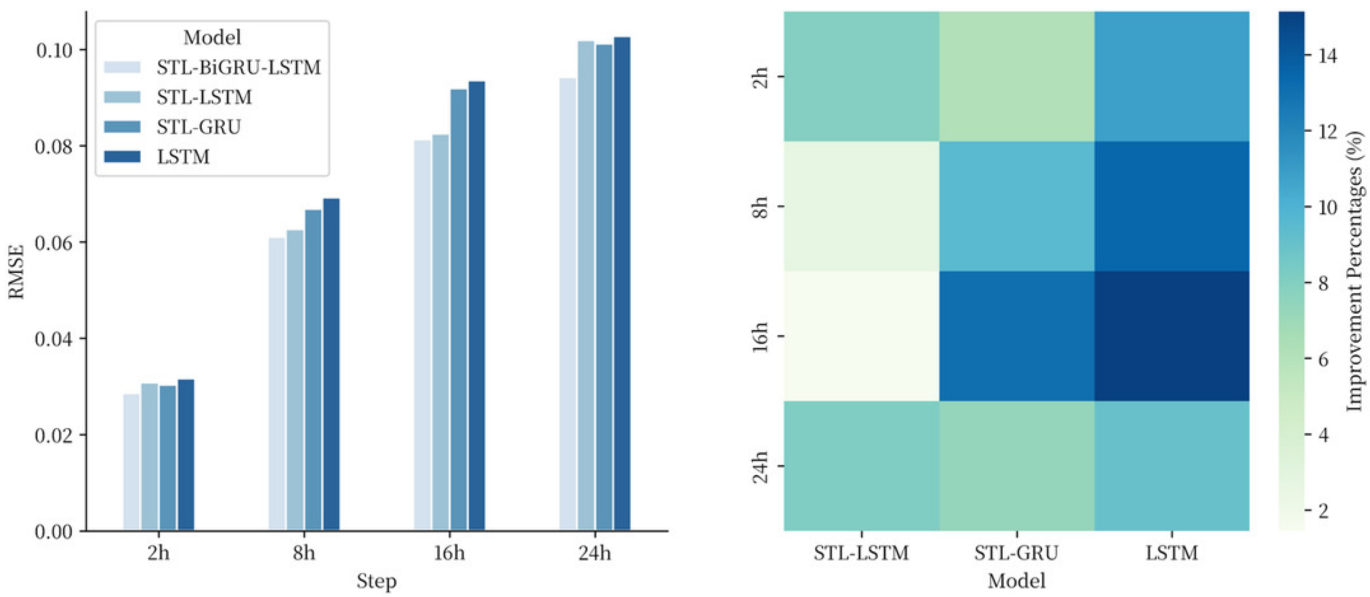

Figure 10

Figure 10 Percentage performance improvement compared to the comparison model in the generalization performance experiment