Developing a Long Short-Term Memory (LSTM) based Model for Predicting Water Table Depth in Agricultural Areas

Akshay Kharbanda (20170137)

akshay.kharbanda@students.iiit.ac.in

Abstract

Predicting water table depth over the long-term in agricultural areas presents great challenges because these areas have complex and heterogeneous hydrogeological characteristics, boundary conditions, and human activities; also, nonlinear interactions occur among these factors. Therefore, a new time series model based on Long Short-Term Memory (LSTM), was developed in this study as an alternative to computationally expensive physical models. The proposed model is composed of an LSTM layer with another fully connected layer on top of it, with a dropout method applied in the first LSTM layer. In this study, the proposed model was applied and evaluated in five sub-areas of Hetao Irrigation District in arid northwestern China using data of 14 years (2000-2013). The proposed model uses monthly water diversion, evaporation, precipitation, temperature, and time as input data to predict water table depth. A simple but effective standardization method was employed to pre-process data to ensure data on the same scale. 14 years of data are separated into two sets: training set (2000-2011) and validation set (2012-2013) in the experiment. As expected, the proposed model achieves higher R2 scores (0.789-0.952) in water table depth prediction, when compared with the results of traditional feed-forward neural network (FFNN), which only reaches relatively low R2 scores (0.004-0.495), proving that the proposed model can preserve and learn previous information well. Furthermore, the validity of the dropout method and the proposed model's architecture are discussed. Through

experimentation, the results show that the dropout method can prevent overfitting significantly. In addition, comparisons between the R2 scores of the proposed model and Double-LSTM model (R2 scores range from 0.170 to 0.864), further prove that the proposed model's architecture is reasonable and can contribute to a strong learning ability on time series data. Thus, one can conclude that the proposed model can serve as an alternative approach predicting water table depth, especially in areas where hydrogeological data are difficult to obtain.

1 Introduction

Groundwater provides an important source of water for domestic, agricultural, and industrial use. However, groundwater resources are vulnerable to overexploitation, climate change, and biochemical pollution. As a result, many areas over the world face groundwater shortages. An example of such areas is the Hetao Irrigation District, one of the largest irrigation districts in China that is located in the arid area of the Yellow River watershed. The Yellow River serves as the main source of irrigation water in this district. However, the availability of irrigation water from the Yellow River has decreased dramatically, with intensified water resource use in the Yellow River watershed. Therefore, groundwater has become an important source of supplementary irrigation water in Hetao Irrigation District. The effective management of groundwater resources, especially in the context of increased groundwater demands for agriculture use, is necessary to provide sustainable use of water resources in Hetao Irrigation District. Sustainable groundwater planning and management requires accurate forecasting of water table depth. An accurate and reliable assessment of water table depth can help engineers and decision-makers to:

(1) develop optimal water resource allocation strategies

(2) adjust crop patterns in different sub-irrigation areas

(3) develop optimal irrigation schedules while controlling the effects of salinity related to intensive irrigation.

The objective of this study is to develop an effective and accurate method for predicting water table depth that can be used to help engineers and decision-makers manage groundwater resources and make management decisions. Using physically based models to predict water table depth in Hetao Irrigation District is particularly challenging because the district covers a large area, lacks abundant hydrogeological data, and has strong spatial and seasonal variability in freeze-thaw periods.

ANN methods are used in this study for predicting water table depth, because ANN has a strong self-learning ability, which is suitable for complicated problems. However, traditional FFNN models do not have the ability to learn time series data because they cannot preserve previous information, resulting in limited prediction capability for long-term time series data, such as data related to water table depth. This problem can be resolved by using more advanced FFNN models. Recently, researchers have integrated FFNN with other methods such as genetic algorithms, wavelet transform, and singular spectrum analysis. Genetic algorithms can be applied to optimize neural networks. Wavelet transform and singular spectrum analysis can pre-process time series data, then add processed time series data into neural networks and thus allow FFNN models to learn time series data very well. However, these advanced FFNN models require complicated procedures related to data pre-processing.

The present study focuses on a time series prediction task, so recurrent neural network (RNN) is a suitable choice. A RNN model has internal self-looped cells, allowing the RNN to "remember" time series information and making it adept at performing time series tasks. In addition, in this study, LSTM, a special kind of RNN, that works well in processing long term time series data, was chosen due to its sophisticated network structure. Compared with the aforementioned advanced FFNN model, the proposed LSTM-based model only applied a very simple data pre-processing method. LSTM is a famous deep learning model. It is recurrent, where connections between units form a directed cycle allowing data to flow both forwards and backwards within the network; then the previous information can be preserved for future use.

The goal of this study was to develop a two-layer LSTM-based model for predicting water table depth in Hetao Irrigation District. The model contained one layer of LSTM and a fully connected layer atop of the LSTM layer. The model employed monthly water diversion, evaporation, precipitation, temperature and time as input variables for predicting water table depth in the district.

2 Materials and methods

2.1 Study area

Hetao Irrigation District lies within Bayannaoer City, Inner Mongolia, China. The Yellow River and Wolf Mountain form the northern and southern boundaries, respectively. Characterized by narrow strips of flat terrain and fertile land, the entire Hetao Irrigation District has been divided into five sub-areas from west to east as follows: Ulanbuh, Jiefangzha, Yongji, Yichang and Urad (Fig. 1).

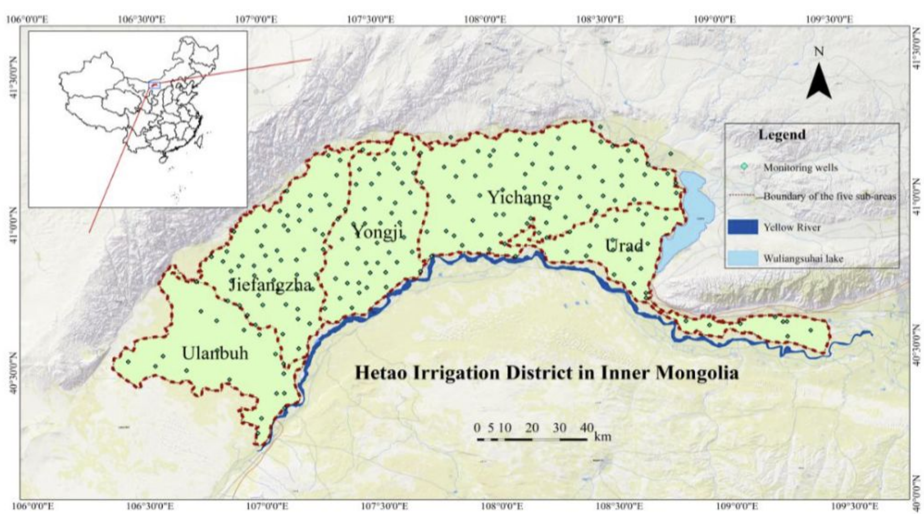

Fig. 1. Location of Hetao irrigation district, the boundary of five sub-areas, and the observation wells.

Mean annual precipitation is 169.4 mm, about 70% of which falls during June-September; the maximum precipitation occurs in August. The mean, minimum and maximum temperatures were 3.9°C, -14.6°C, and 28.4°C, respectively. Hetao Irrigation District covers 11,073 km2; the central part of the district is about 180 km long and 60 km wide. About 52% (5740 km2) of the entire terrestrial area is irrigated. A total of 227 monitoring wells are used in the district (Fig. 1) with the water table depth measured every 5 days. Because monthly data are used in the model, the measured water table depth is averaged by month.

2.2 Data and statistical analysis

Our experiments used 14 years (2000–2013) of time series data from the five sub-areas in Hetao Irrigation District. Since people in this district use surface water, pumping volume is very small, so it was omitted in this study. Therefore, at the beginning, the data included water diversion, precipitation, evaporation, temperature, water table depth, area of the region, water consumption for industry and domestic use. Then Lasso regression, a statistical method that can be used to select variables, was applied to choose important variables from the original data, based on regression weights. Finally, only five variables were chosen for this study: water diversion, precipitation, evaporation volume, temperature, and water table depth. Time series of these variables are presented for the five sub-areas of the district (Fig. 2).

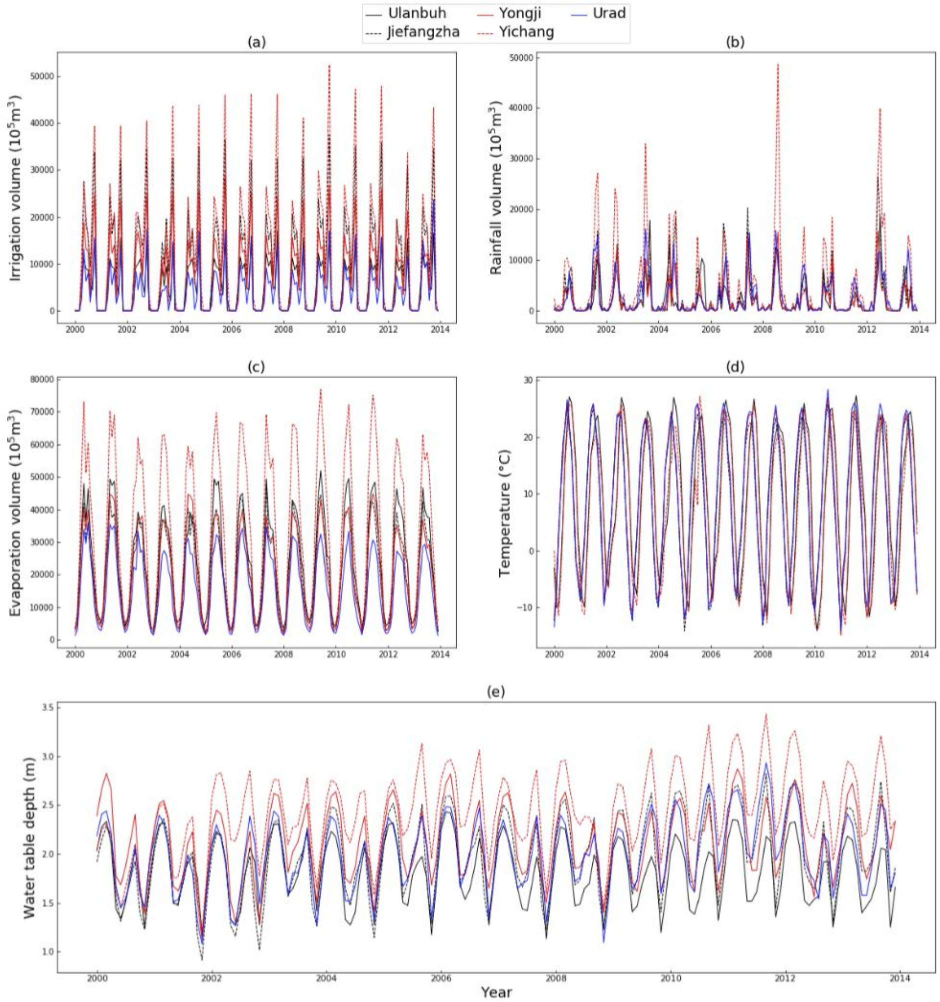

Fig. 2. Variables for different sub-areas, including (a) water diversion, (b) precipitation (c) evaporation volume, (d) temperature and (e) water table depth.

From Fig. 2, these data were found to have periodic characteristics. Therefore, time (at the monthly scale) has also been used as an input variable to improve the model's ability to generalize, creating a sixth variable. In the present study, the first 12 years of time series data were used as a training set, and the next two years of data was used as a validation set. A statistic description of the water table depth of different sub-areas is shown in Table 1. Table 1 shows that Ulanbuh had the smallest standard deviation (0.331 m) and range of water table depth (1.297 m), meaning the water table depth of Ulanbuh is more stable than that of other areas.

Table 1 Statistic description of water table depth of five sub-areas.

Areas	Average (m)	Maximum(m)	Minimum (m)	Standard Deviation (m)	Skewness
Ulanbuh	1.817	2.430	1.133	0.331	-0.167
Jiefangzha	1.970	2.821	0.914	0.422	-0.140
Yongji	2.111	2.870	1.129	0.379	-0.108
Yichang	2.421	3.430	1.202	0.429	-0.222
Urad	2.002	2.932	1.080	0.375	-0.103

2.3 Data pre-processing

As shown in Fig. 2, the input data varied widely. These differences will have a negative effect on the model's ability to learn. Therefore, all six of the variables were standardized to ensure they remain on the same scale. This pre-processing can guarantee a stable convergence of parameters in the model developed in the present study. The standardization formula is as follows:

$$x_{\mathrm{ij}}^{(\mathrm{new})} = \frac{x_{\mathrm{ij}} - \overline{x}_{\mathrm{i}}}{\sigma_{\mathrm{i}}}$$

where x_{ij} represents data in i^{th} year, j^{th} month; $\overline{xi}$ and σ_i are the average and standard deviation of data in the i^{th} year, respectively.

3 Methodology

3.1 Long Short-Term Memory Network (LSTM)

LSTM is a special kind of RNN. RNNs have connections between neurons and form a directed cycle, creating an internal self-looped cell, which allows it to display dynamic temporal behavior. RNNs have chain-like structures of repeating modules (Fig. 3(a)). These structures can help RNNs to "remember"previous information, which allows the RNNs to process arbitrary (long time) sequences.

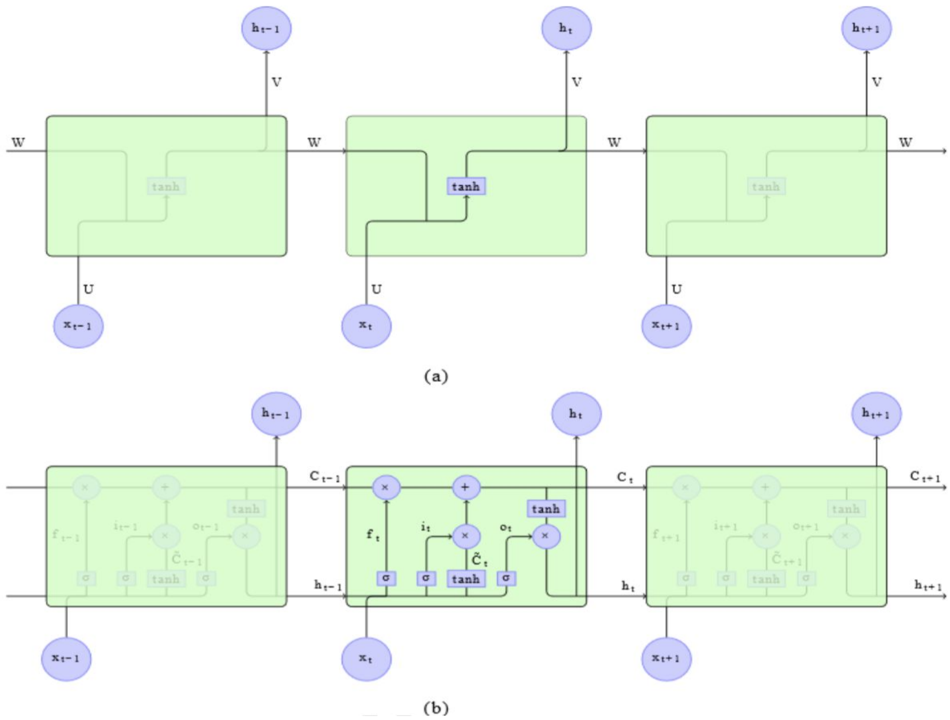

Fig. 3. (a) Chain like structure of the recurrent neural network. The self-connected hidden units allow information to be passed from one step to the next. (b) A graphical representation of LSTM's memory block.

The gradients of the RNNs can be computed via Back-Propagation Through Time. However, Back-Propagation Through Time is not sufficiently efficient to learn a pattern from long term dependency because of a gradient vanishing problem. This problem can be solved by the structure of LSTMs. Like RNNs, LSTMs also have chain-like modules, but the repeating modules have more complicated structures. Each repeating module of LSTMs contains a memory block. The memory block contains four parts: a CEC (the Constant Error Carousel) cell in addition to three special multiplicative units called gates. The CEC cell runs straight down the entire chain without any activation function and thus the gradient does not vanish when Back-Propagation Through Time is applied to train a LSTM. Therefore, LSTMs have been shown to learn long-term dependencies more easily than RNNs because information can easily flow along the cells unchanged. Also, the input, forget and output gates in each memory block can control the flow of information inside the memory block.

3.2 Dropout for Neural Networks

Deep neural networks are well suited to process big data. However, DNNs with large numbers of parameters can easily be overfitting, especially when data are limited. Dropout provides an effective regularization method that can be used to solve this problem. The most important idea of the dropout method is that at each training iteration, when the neural network is updating a certain layer where the dropout is applied, it randomly does not update, or "dropout" some neurons in this layer. This means that a part of the neural network was sampled and it was trained at one iteration. In each iteration of training, a different part of the network was sampled and trained. With dropout, the weights of the neurons learned through backpropagation become somewhat more insensitive to the weights of the other neurons. Thus, dropout can help to prevent the networks from relying on certain neurons in the layers too much. A neural network, whose first and second hidden layers have been applied dropout, is shown in Fig. 4.

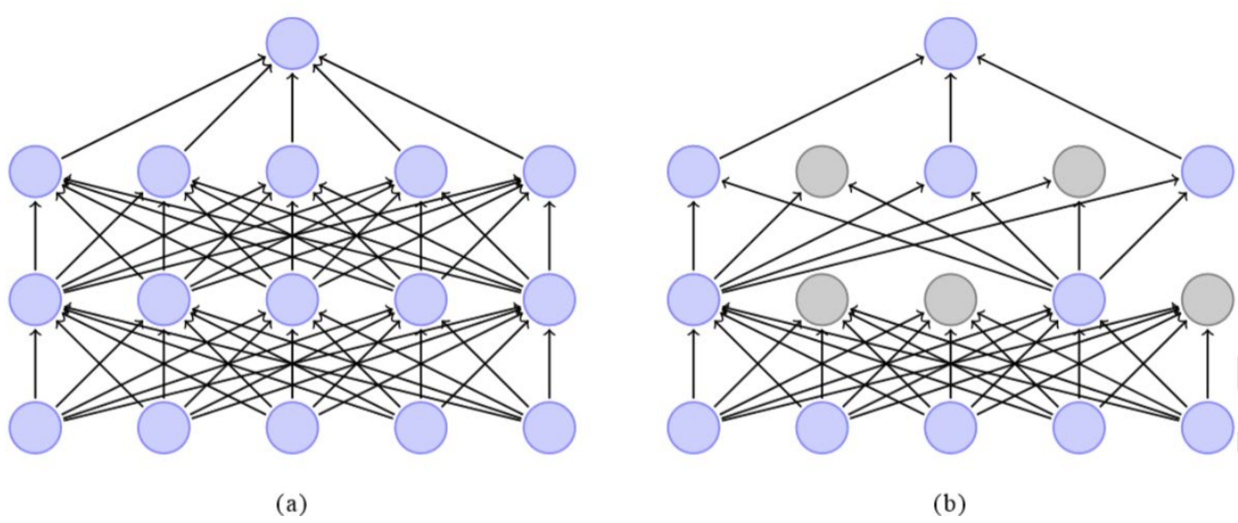

Fig. 4. (a) A standard feed-forward neural network with two hidden layers; (b) Applying dropout to two hidden layers on network (a). Grey units on (b) have been dropped.

3.3 Our proposed model framework

In this work, we were interested in predicting water table depth. This is a time series problem because the current water table depth has changed in a way that is dependent on previous. This time series prediction was cast as a regression problem.

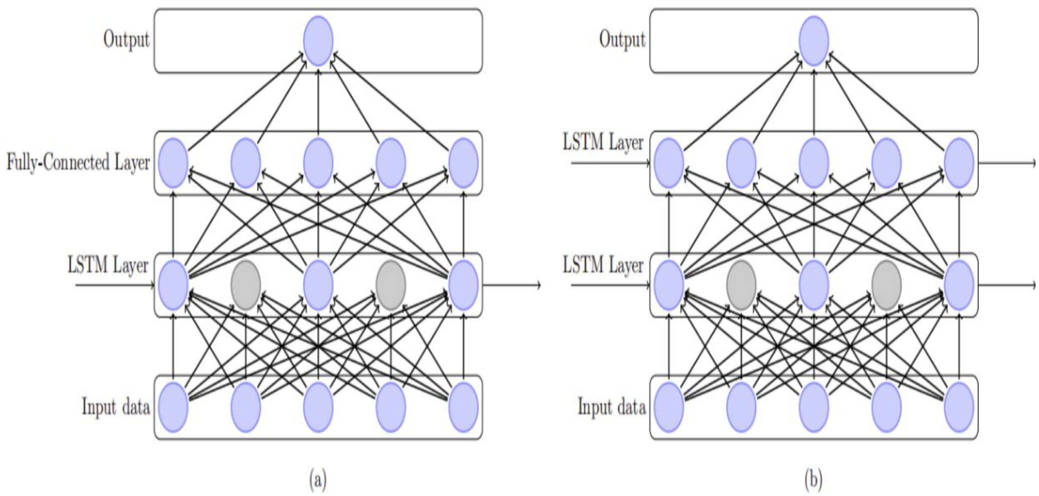

Fig. 6. (a) Structure of the proposed model. Dropout has been applied at the Long Short-Term Memory (LSTM) layer. (b) Structure of the Double-LSTM model. Dropout has been applied at the first LSTM layer. Grey units have been dropped.

The proposed model is illustrated in Fig. 6(a). The input data were first put into the LSTM layer. The input gate of the LSTM layer will recompose input data and decide which input data is important; this process is similar to principal component analysis (PCA). However, the LSTM layer can preserve previous information, which can help to improve the ability of the model to learn time series data. A fully connected layer is set atop the LSTM layer in order to improve the model's learning ability. Moreover, dropout is set on the LSTM layer to prevent overfitting. The loss function is defined below:

$$LOSS=\sum_{i=1}^{N}(y_i-\hat{y}_i)^2$$

where y_i is measured value at time i; and $\hat{y}_i$ is predicted value at time i.

However, this model framework has some limitations. First, the initial parameters of the proposed model will affect the final results. In addition, even though an LSTM layer has a strong ability to learn time series data, its fitting ability may be insufficient. Therefore, a fully connected layer was added atop of a single LSTM layer. Nevertheless, exactly how many LSTM layers should be used as hidden layers in order to reach the optimized results remained unknown.

3.4 Model evaluation criteria

In this study, the root mean square error (RMSE), and coefficient of determination (R2) were used to evaluate the model's accuracy between the measured and predicted values. R2 measures the degree of how well the outcomes are replicated by the model, ranging between [-∞, 1] where for optimal model prediction an R2 score close to 1 is preferred. The R2 equation is as follows,

$$R^2 = 1 - \frac{SS_{RES}}{SS_{TOT}} = \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y})^2}$$

where y_i is measured value at time i, $\bar{y}_i$ is mean of y_i(i = 1,..., N) and $\hat{y}_i$ is predicted value at time i.

Diverse types of information related to the predictive capacities of the model were measured through RMSE. RMSE measures the prediction precision which creates a positive value by squaring the errors. RMSE scores range between[0, ∞], and the model prediction is ideal if RMSE is 0. RMSE is defined as,

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \hat{y}_i)^2}{N}}.$$

4 Results

4.1 Water table depth prediction results

As mentioned above, in the training process, the first 12 years of data were used to train the proposed model in different areas. After the models had been trained, each model was validated through use of the validation set and tuned the hyper-parameters of the models. Two performance metrics of different models in different sub-areas and “Hetao” were computed in order to obtain the optimal model hyper-parameters (the number of hidden neurons in the LSTM layer, and the learning rate). The optimal hyper-parameters of the proposed model used for water table depth prediction are shown in Table 2 (first row).

Table 2 Results of Yongji’s water table depth prediction, with different hyper-parameters set to the proposed model. The hyper-parameters in the first row are used in this study.

Neurons	Learning rate	Dropout	Iterations	Loss	R^2	Time (min)
40	10^{-4}	**0.5**	**20000**	9.86	**0.82**	4.38
40	10^{-4}	0.5	40000	7.11	0.82	8.41
40	10^{-4}	0.5	5000	30.10	0.35	1.21
40	10^{-4}	0.5	20000	0.72	0.59	5.31
40	10^{-4}	0.5	20000	30.72	0.64	4.02
40	10^{-3}	0.5	20000	2.43	0.62	4.46
40	10^{-5}	0.5	20000	54.43	0.69	4.29
70	10^{-4}	0.5	20000	23.54	0.49	9.76
10	10^{-4}	0.5	20000	23.37	0.41	3.30

Note that models for different areas use the same hyper-parameters. In order to illustrate how hyper-parameters affect the results, different hyper-parameters were set for the proposed model and the

Yongji sub-area was used as an example. Results, including training loss, R2 and running time, are also displayed in Table 2. From Table 2 it can be learned that a higher learning rate (10^{-3}) may cause the optimization task to miss the optimal point (R2 was only 0.62); however, a lower learning rate (10^{-5}) may help to avoid overshooting but may cause the model to use a longer time to converge. Too many training iterations may not ensure more optimal results (the results from 20,000 and 40,000 training iterations were the same). In addition, it is common to set a dropout probability to 0.5 in the deep learning field. Moreover, too many neurons (70 in this study) are computationally expensive (20,000 iterations in 9.76 min) and often cause overfitting, meanwhile, having an insufficient number of neurons (10 in this study) may decrease the network's learning abiliy.

4.2 Dropout effect

In order to explore the dropout method that was used to prohibit overfitting, the results from the proposed model were compared with a model with same architecture but without applying dropout in the LSTM layer. The same hyper-parameters (40 hidden neurons, 10^{-4} learning rate and 20,000 training iterations) were set in the model without dropout and its performance was compared with the proposed model in water table depth prediction. Training loss and two evaluation metrics were used for evaluating these two models. Fig. 9 shows the prediction results of the proposed model and the model without dropout in three areas.

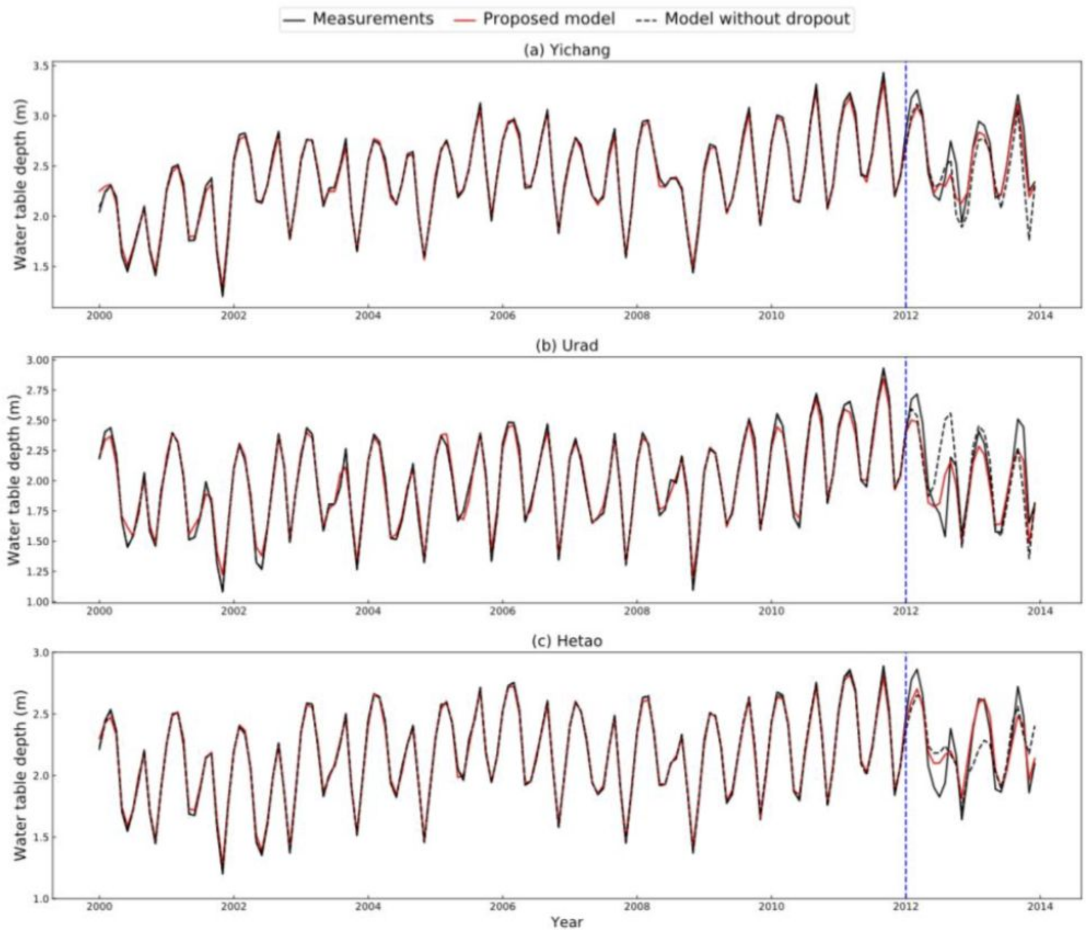

Fig. 9. Comparison of measured and simulated water table depth using the proposed model and model without dropout in different areas. The blue dash line separates the data into two sets: the training and validating sets. For brevity, we only present three typical areas here.

The evaluation metrics and training loss are shown in Table 3. From Fig. 9, one can see that the proposed model and model without dropout provided very similar results in the training process in different areas. However, the results in the validating process showed stronger deviations from the predicted value. Table 3 shows that the training loss of the proposed model ranged from 7.34 to 11.42, whereas that of the model without dropout ranged from 0.725 to 1.204. However, the R2 of the proposed model was higher than that of the model without dropout, which ranged from 0.368 to 0.874; also, the RMSE of the proposed model was smaller than that of model without dropout, which ranged from 0.114 m to 0.289 m. These findings indicate that even though the model without dropout can fit training data perfectly, it cannot predict the last two years of water table depth as well as the proposed model. This phenomenon is a typical overfitting. However, from the reported result of the proposed model, the conclusion could be drawn that the dropout method can help the proposed model by preventing overfitting.

5 Conclusions

The major conclusions are as follows:

(1) The proposed model provides a promising new method for predicting water table depth, as evidenced by satisfactory performance on water table depth prediction in five sub-areas and "Hetao".

(2) The architecture of the proposed model is reasonable. The LSTM layer helps to maintain previous information and contributes to learning the time series data. The dropout method helps to prevent overfitting during the training process. A fully connected layer atop the LSTM layer helps to improve the learning and fitting ability of the model.

(3) The newly proposed model provides a valuable tool for predicting water table depth. It can serve as an alternative model to predict water table depth in places with complex hydrogeological characteristics and hydrogeological data are difficult to obtain.