Reservoir inflow prediction using a hybrid model based on deep learning

Bin Luo¹, Yang Fang², Handong Wang¹ and Di Zang²,*

¹Changjiang survey, planning, design and research co., LTD. Wuhan China
²Department of Computer Science and Technology Tongji University. Shanghai China

*E-mail: zangdi@tongji.edu.cn

Abstract. Reservoir inflow prediction plays a significant role in the field of hydrological prediction. Accurate and reliable prediction of reservoir inflow is the key to flood control decision. In this paper, we combine the deep belief network (DBN) with the Long Short-Term Memory (LSTM) to present a hybrid model based on deep learning (HDL) for reservoir inflow prediction. We take a full consideration of the basin flow and rainfall factors, which significantly affect the inflow flow. According to the rainfall data, we divide the corresponding flow data into two cases: rain and no rain. The proposed approach consists of three parts: we apply the DBN to learn the characteristics of the flow data and get predicted values of the reservoir inflow in case of rain and in case of no rain respectively. Then, we use the basin rainfall data and adopt the LSTM to fit the differential between the predicted inflow value generated by DBN in case of rain and the real inflow value, and get the predicted differential. Finally, the outputs of these three parts are added to obtain the final predicted result. Experiments are evaluated by the historical flow and rainfall data of a reservoir in China, and the results have proved that our method is effective and has higher prediction accuracy.

1. Introduction
Reservoir inflow prediction is one of the most important tasks and challenges in hydrology. Hydrological phenomenon is of high fuzziness, complexity and randomness since it is influenced by many uncertainty factors such as rainfall, the characteristics of topography, river channel and soil distribution of the catchment area. There are quite a few methods have been proposed for hydrological forecasting, which can be classified into 3 major categories: statistical methods based on hydrology, machine learning methods and deep learning methods.

Over the past decades, there have been plenty of traditional methods for the hydrologic inflow prediction such as process-based hydrology model [1], auto-regressive moving average (ARMA), and auto-regressive integrated moving average (ARIMA) [2]. ARIMA model is also applied to forecast the stream flow of the Hirakud reservoir in [3]. Moreover, a probabilistic method named SCHADEX was presented for flood volumes estimation in [4]. Generally, these conventional methods usually rely on a number of assumptions for natural environment. However, it is hard to reflect the stochastic and nonlinear nature of hydrological data due to various factors such as climate, natural geography and human activities that significantly affect them.

Afterwards, considering the nonlinear characteristics of the real data of the hydrology, a great deal of machine learning approaches are used for forecasting in the field of hydrology. For example, some classic models are Bayesian-based methods [5], [6], Support Vector Machines (SVM) [7], [8], fuzzy...
inference systems [9], and Artificial Neural Network [10]. To achieve better prediction accuracy and efficiency, Cheng et al. [11] developed an artificial neural network model based on quantum-behaved particle swarm optimization for daily reservoir runoff prediction. Hu et al. [12] applied principal component analysis (PCA) and improved the traditional BP algorithm to predict river runoff. Although these methods can improve the accuracy of hydrological prediction to a certain extent, they belong to “shallow” learning category and have certain deficiencies such as large number of parameters, time-consuming, and easily falling into local optimum, which cannot be ignored in practical application.

Recently, several deep learning methods, which can achieve better prediction performance owing to its “deeper” representations, have been proposed as well. Bai et al. [13] employed deep belief networks (DBNs) to represent multiscale features and proposed a hybrid model to handle the daily reservoir inflow prediction. Liu et al. [14] presented a deep learning model based on stacked auto encoders (SAE) and back propagation neural networks (BPNN) and adopt it to predict the stream flow. Wu et al. proposed a context-aware attention LSTM (CA-LSTM) network to extended the original LSTM network for flood prediction [15].

Deep learning has a powerful ability to automatically extract complex features of data, thus it is a very great way to apply it to analyse a large number of historical hydrological data. However, it has not been widely applied in reservoir inflow prediction so far. The existing deep learning network models are generally weak in the analysis for specific problems, and have such shortcomings as low prediction accuracy, simple network structure and poor scalability.

To improve the accuracy of reservoir inflow prediction, in this paper, we propose a hybrid model based on deep learning (HDL) and apply it to predict the reservoir inflow. We first take the historical flow data as the input of a DBN model to obtain the predicted inflow value F1 in the case of no rain. Then, the DBN model is used again to generate the predicted inflow value F2 in case of rain according to the mapping relationship between the flow data and the reservoir inflow learned in case of no rain. Next, we use the rainfall data as the input of a LSTM to fit the differential between the real inflow value and F2, and obtain a predicted value of the differential ΔF in case of rain. Finally, we add up these results F1, F2 and ΔF generated by three modules to get the predicted reservoir inflow value. The experimental results have demonstrated that our proposed approach achieves higher accuracy than the comparative methods in reservoir inflow prediction.

2. Dataset description and processing

In this paper, the data sets used for experiments is the historical hydrological data of a reservoir in China from 7/7/2010 to 31/10/2017, which consists of three parts: the first part is the flow data collected by 3 control stations of 3 tributaries within the catchment of the reservoir; the second part is the rainfall data collected by 123 rainfall control stations located between three tributaries and the reservoir; and the third part is the inflow data collected by a control station of the reservoir. Data in flood periods from May to October every year except 2010 are available for experiments. Here the sampling interval of data is 6 hours so that there are four recorded time points in a day: 2 am, 8 am, 14 pm and 20 pm. Therefore, the flow indicates the instantaneous flow measured every 6 hours and rainfall indicates the depth of water after a 6-hour rainfall record. To reflect the changes of flow and rainfall, the previous 8 days of flow or rainfall data are used as input of the model to predict the reservoir inflow for next 24 hours. Hence, the length of the input time series is 32. Namely, to predict the reservoir inflow at 2 am on May 2, 2017, the historical data of 2 am on May 1, 2017 and its 31 previous time points are needed. Data from 2010 to 2016 are selected as training samples and the data of 2017 are used as a test samples in our work.

For the data processing, firstly, the original data need to be pre-processed to remove abnormal elements which are hard to fit and seriously affect the prediction accuracy. Then, we classify the data of tributary flow and the reservoir inflow (flow data) at each time point into two categories (rain and no rain) according to the sum of the rainfall values recorded by rainfall control stations at corresponding time points. To be specific, the flow data are classified into the rain category if the sum
of rainfall values is greater than 10mm, otherwise the flow data are classified into the no rain category if the sum of rainfall values is less than or equal to 10mm. In addition, we adopt the method of coarse-grained clustering to divide the catchment area of the rainfall control stations into several regions with obvious boundaries according to the position information of all rainfall control stations. In this paper, we get four regions by dividing, and then sum up the rainfall values recorded by the rainfall control stations in each region at each time point to get four sum values, which are used as the input data of the LSTM module.

3. Presented approach
Figure 1 shows the system architecture of our approach, which mainly includes four major steps: first of all, we build a DBN model to extract and learn the sophisticated characteristics of historical tributary flow and the reservoir inflow data (flow data), and generate the predicted reservoir inflow value F1 in the case of no rain. Secondly, the historical flow data are mapped to generate the predicted reservoir inflow value F2 in case of rain by the same DBN model using the parameters obtained in the first step. After that, we calculate the difference between the value of real reservoir inflow and the value F2 to get the differential Δ. Thirdly, we develop a LSTM model to learn and capture the corresponding relation between the rainfall data after clustering and the Δ, so as to generate the predicted differential value ΔF. At last, the predicted result of reservoir inflow is the sum of the above three parts.

3.1. DBN module for inflow prediction in case of no rain
The DBN module is utilized to predict the reservoir inflow in case of no rain in the first step. We selected the reservoir inflow data belonging to no rain category as the label for the DBN module, and the corresponding input data is a one-dimensional vector transformed from the historical flow data. As mentioned in section 2, the length of the input time series is 32, thus the length of the input data is 128 since the historical flow data come from 3 tributaries and a reservoir. If there exist data belonging to the rain category in the input data, we replace them with the average of their adjacent values.

The structure of the DBN module is shown in figure 2, DBN is a probability generation model, which is composed of restricted boltzmann machines (RBM). Our DBN model consists of a RBM and an artificial neural network (ANN), wherein RBM is composed of a visible layer and a hidden layer, and neurons are bidirectional and fully connected between the visible and hidden layers. Besides, ANN is composed of two hidden layers with hidden size of 1000 and a fully connected layer. The size of the input visible layer in RBM is the length of the input data, and the size of the RBM hidden layer is set to 1000. The output of each fully connected layer is defined as the following formula:

\[ y^l = f(w^l x^{l-1} + b^l) \]

where, \( y^l \), \( w^l \) and \( b^l \) are the output, the weight and the bias of layer \( l \), respectively. \( x^{l-1} \) is the output of layer \( l-1 \). \( f \) is the activation function, Here, the activation functions of the hidden layers in RBM and ANN are all sigmoid activation functions, which can be defined as:
$$f(x) = \frac{1}{1+e^{-x}}$$

(2)

The last output fully connected layer has no activation function, which is a linear output.

Figure 2. The structure of the DBN module.

The training process of DBN network consists of pre-training and fine-tuning. In the process of pre-training, the RBM is trained unsupervised to make feature vectors map to different feature spaces and ensure that feature information is retained as much as possible. After the RBM has been trained, the output feature vectors of RBM are input to ANN to initialize the weight parameters of ANN network, and then fine tune the entire DBN network through back propagation.

We adopt the mean square error (MSE) as the loss function to calculate the difference between the ground truth and the predicted result. Assuming there are N training samples, $y'$ and $y$ respectively represent the predicted value and the ground truth, then the loss function $L$ is defined as:

$$L = \frac{1}{N} \sum_{i=1}^{N} (y' - y)^2$$

(3)

In this paper, the loss function of both DBN and LSTM are optimized by Adam method. In this part, we employ the DBN to find the corresponding relationships between the training data and the training label by exploring and extracting their sophisticated natures, and finally get the reservoir inflow value $F1$ in case of no rain.

3.2. DBN module for inflow prediction in case of rain

Based on the same DBN model, to predict the reservoir inflow in case of rain, namely, the reservoir inflow to be predicted belongs to rain category. Similarly, we replace those input data belonging to the no rain category with the average of their adjacent values. Then, the corresponding relationships between the data and the label learned in section 3.1 are directly applied here for analysis in case of rain. In other word, the input data is mapped to generate the predicted reservoir inflow $F2$ in case of rain according to the training parameters obtained in section 3.1. Finally, we calculate the difference between the ground truth in case of rain and the corresponding predicted value $F2$ to get the differential $\Delta$.

3.3. LSTM module for reservoir inflow differential prediction

Considering that the reservoir inflow should be equal to the sum of the tributary flows and the total rainfall of the reservoir basin, however, we find that it is difficult to establish a reasonable mapping relationship between the original rainfall data and the reservoir inflow. Since the differential $\Delta$ mentioned in 3.2 can be regarded as the result of rainfall, inspired by the excellent performance of
LSTM in modeling the dependencies of sequential data, we develop the LSTM module to fit the differential of reservoir inflow based historical rainfall data.

LSTM is a special kind of recurrent neural network, whose key construction unit is the cell. A LSTM cell is composed of three gate structures (input gate, forget gate and output gate) and a state vector transmission line, in which the state vector transmission line is responsible for long-term memory and the three gates are responsible for short-term memory selection. Therefore, the input vector can be deleted or added by gate setting, so that information can be selectively passed. What’s more, the problem of gradient disappearance can be solved since the forgetting gate and the output gate can forget part of the information.

The structure of the LSTM network model is shown in figure 3, our LSTM module consists of one cell which is unrolled by time step. As described in section 2, we take the sum of rainfall of the four regions as the input data of the LSTM module. Owing to we use the data of the previous 8 days to make the prediction and there are 4 time points every day, the number of time step is 32 (32 cells unrolled by time), that is, for each prediction, the sum of the four rainfall data of 32 historical time points needs to be input, so the input is 32 one-dimensional vectors. The input data of LSTM is entered sequentially, one input in the one time step. At the same time, the differential Δ obtained in the second 3.2 section is used as the label for training. It is worth noting that here the data and labels used for network training are set to 0 if there is no rain. Considering that it is unreasonable to use rainfall values to fit negative values, thus set the differentials less than 0 to zero as well.

![Figure 3. The structure of the LSTM module.](image)

In the LSTM model, first of all, the LSTM has to decide what information should be forgotten from the cell state, which is determined by a sigmoid layer called the “forget gate”. Assuming \(h_{t-1}\) represents the output of the previous cell, \(x_t\) represents the input of current cell, the output of the forget gate can be written as:

\[
f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)
\]

where \(f_t\), \(W_f\), \(b_f\) respectively represent the output, the weight and the bias of forget gate of current cell, \(\sigma\) represents the sigmoid activation function. The output is a value in the range 0 to 1, where 1 means "completely retained" and 0 means ‘completely forgotten’. Next, the LSTM decides what information to be saved in the cell state, which consists of two parts: first, a sigmoid layer called the “input gate” determines what information needs to be updated, second, a tanh layer generates a new candidate value vector \(C'_t\). Then, the two parts are combined to make an update to the cell's state. The input gate of current cell \(i_t\) and the new candidate value vector \(C'_t\) can be represented as formula (5) and formula (6) respectively:

\[
i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)
\]

\[
C'_t = \tanh(W_c \cdot [h_{t-1}, x_t] + b_c)
\]

where \(W_i\), \(b_i\) respectively represent the weight and the bias of input gate of current cell. \(W_c\), \(b_c\) respectively represent the weight and bias of \(C'_t\) of current cell.

The old state \(C_{t-1}\) is multiplied by \(f_t\) to forget the information we decided to forget before, and then added by \(i_t \cdot C'_t\) to get the new state \(C_t\) which can be defined as follows:

\[
C_t = f_t * C_{t-1} + i_t * C'_t
\]

and
Finally, the LSTM decides what to output. Firstly, the sigmoid layer called “output gate” is used to determine which part of the cell state will be output. The cell state is then passed through the tanh layer (to get a value in the -1 to 1 range) and multiplied by the output of the sigmoid layer. Therefore, the output gate $o_t$ and the final output $h_t$ of the current cell can be represented as formula (8) and formula (9) respectively:

$$o_t = \sigma(W_o \cdot [h_{t-1},x_t] + b_o)$$  \hspace{1cm} (8)

$$h_t = o_t \cdot \tanh(C_t)$$  \hspace{1cm} (9)

The expression of loss function to be optimized in LSTM model is the same as formula (3). Through the training and learning of LSTM network, predicted differential value ΔF is obtained in case of rain. Finally, the predicted value $F$ of reservoir inflow obtained by our method is: $F1+F2+ΔF$.

4. Results and evaluation

Our experiments are conducted on the server with i7-5820K CPU, 48GB memory and NVIDIA GeForce GTX1080 GPU, and we employ the TensorFlow framework of deep learning to implement our approach. The learning rates of the DBN model and LSTM model are set to 0.0001 and 0.001 respectively, and the total number of network iterations of both DBN and LSTM is 20000.

In order to evaluate the performance of the proposed approach, first, we calculate three error values: error $\geq 10\%$, error $\geq 15\%$ and error $\geq 20\%$, which respectively represent the percentage of data with an error of 10% or more in the total data, the percentage of data with an error of 15% or more in the total data, and the percentage of data with an error of 20% or more in the total data. Moreover, we calculate three performance evaluation criteria: Mean Relative Error (MRE), Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) which are defined as formula (10), formula (11) and formula (12) respectively:

$$MRE = \frac{1}{N} \sum_{i=1}^{N} \frac{|y' - y|}{y}$$  \hspace{1cm} (10)

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |y' - y|$$  \hspace{1cm} (11)

$$RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (y' - y)^2}$$  \hspace{1cm} (12)

Where $y'$ represents the predicted value, $y$ represents the real value, $N$ represents the number of samples in test set.

We compare our approach with the methods of conventional ANN, the single DBN model and single LSTM model. In the comparison model, ANN has two hidden layers with 128 neuron nodes. All comparative models don’t take into account the impact of rainfall data on reservoir inflow, only use historical flow data (flow data of 3 tributaries and inflow data of the reservoir) of the previous eight days to predict reservoir inflow. Besides, these models also don’t analyze the flow data separately in terms of rain or no rain. Tables 1-2 show the comparison results of error and comparison results of MRE, MAE and RMSE.

| Models   | error $\geq10\%$ | error $\geq15\%$ | error $\geq20\%$ |
|----------|------------------|------------------|------------------|
| Our method | 0.177143         | 0.075714         | 0.037143         |
| ANN      | 0.311429         | 0.181429         | 0.105714         |
| DBN      | 0.225714         | 0.104286         | 0.052857         |
| LSTM     | 0.220000         | 0.098571         | 0.051724         |
Table 2. Comparison results of MRE, MAE, RMSE.

| Models   | MRE(%) | MAE  | RMSE  |
|----------|--------|------|-------|
| Our method | 6.29   | 1254.39 | 1963.68 |
| ANN      | 8.95   | 1889.29 | 3006.75 |
| DBN      | 6.99   | 1470.64 | 2376.26 |
| LSTM     | 6.83   | 1466.65 | 2339.94 |

As shown in tables, our proposed method gets the lowest error compared with the other three methods. In terms of three error values, the results of error >=10%, error >=15% and error >=20% obtained by our model are respectively 0.177143, 0.075714 and 0.037143. It can be seen from the results that our model can achieve a high accuracy rate of 97% with an allowable error of more than 20%. For the results of these three criteria, the values of MRE, MAE and RMSE yielded by our method are also the lowest of the listed methods, which are 6.29 %, 1254.39 and 1963.68, respectively. In the comparison model, the traditional machine learning model ANN has the highest prediction error, while DBN and LSTM perform much better which can be attributed to their powerful deep feature representation capability.

In our model, we simultaneously take advantage of the excellent performance of DBN and LSTM. On the one hand, we apply the deep structure of DBN to digging the latent features of flow data, on the other hand, we take full advantage of LSTM's ability to learn about long-term dependence and apply it to learn and explore sophisticated features of the rainfall time series data. In addition, the coarse-grained clustering of rainfall data and the classification of flow data according to rain or no rain are significantly contribute to improvement of prediction accuracy. Figure 4 shows the fitting curves of the predicted reservoir inflow generated by our model and its corresponding real reservoir inflow. In the figure, the x axis represents time series, the y axis represents the value of reservoir inflow, and the predicted reservoir inflow is represented by the red curve and the real reservoir inflow is represented by the blue curve. As can be seen from the figure, these two curves almost completely overlap, which not only indicates that our model has a high degree of fitting, but also reflects the high prediction accuracy of our method.

![Figure 4](image)

**Figure 4.** The fitting curve of the predicted value and the ground truth.

5. Conclusion

In this paper, a hybrid model based on DBN and LSTM has been proposed for reservoir inflow prediction. To improve prediction accuracy, in terms of data processing, we classify the flow data into two categories according to the situation of rain or no rain. Additionally, we adjust the granularity of rainfall data by clustering and dividing the rainfall control stations according to their positions. In terms of model design, we first employ DBN to discover effective characteristics from flow data, and obtain the predicted values of the reservoir inflow in the case of rain and no rain respectively.
Furthermore, in view of the impact of rainfall data on the reservoir inflow, we develop LSTM, which has excellent performance in fitting time series data, to deal with the problem of fitting between rainfall data and the differential of reservoir inflow in case of rain, so as to obtain the predicted differential value of reservoir inflow in case of rain. Finally, the final reservoir inflow value is obtained by organically combining the results of the above three parts. The experimental results have demonstrate that the proposed approach not only achieves higher prediction accuracy, but also has higher robustness and generalization.

Acknowledgement
This work is supported by the National Natural Science Foundation of China under grants 61876218, 61872271 and 61573259; and the National Key R&D Program of China, grant number [2018YFC1508006].

References
[1] R.J Zhao. The Xinanjiang Model Applied in China. Journal of Hydrology, 135(1–4), 371-81(1992).
[2] Samir. Parameters Estimate of Autoregressive Moving Average and Autoregressive Integrated Moving Average Models and Compare Their Ability for Inflow Forecasting. Journal of Mathematics and Statistics, 8(3), 330-38 (2012).
[3] A. Rath, K.S.Bhoi, S.Samantaray, P.C. Swain. Flow forecasting of hirakud reservoir with ARIMA model. International Conference on Energy, Communication, Data Analytics and Soft Computing (ICECDS). IEEE, Chennai, India, 9 pages (2017).
[4] Emmanue Paquet, Federico Garavaglia, Rémy Garçon and Joël Gaillhard. The schadex method: A semi-continuous rainfall–runoff simulation for extreme flood estimation. Journal of Hydrology, 495(15), 23-37 (2013).
[5] S. Han, P. Coulibaly. Bayesian flood forecasting methods: A review. Journal of Hydrology, 551, 340-51(2017).
[6] Yirui Wu, Weigang Xu, Jun Feng, S. Palaiahnakote and Tong Lu. Local and Global Bayesian Network based Model for Flood Prediction. 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, Beijing, China, 6 pages (2018).
[7] B. Bell, B. Wallace and D. Zhang. Forecasting river runoff through Support Vector Machines. 2012 IEEE 11th International Conference on Cognitive Informatics and Cognitive Computing (ICCI*CC). IEEE, Kyoto, Japan, 7 pages (2012).
[8] M. C. Wu, G. F. Lin and H. Y. Lin. Improving the forecasts of extreme streamflow by support vector regression with the data extracted by self-organizing map. Hydrological Processes, 28(2), 386-97 (2014).
[9] Y. H. Lin, C. C. Chiu, P. C. Lee and Y. J. Lin. Applying fuzzy grey modification model on inflow forecasting. Engineering Applications of Artificial Intelligence, 25(4), 734-43 (2012).
[10] M. T. Sattari, K. Yurekli and M. Pal. Performance evaluation of artificial neural network approaches in forecasting reservoir inflow. Applied Mathematical Modelling, 36(6), 2649-57 (2012).
[11] C. T. Cheng, W. J. Niu, Z. K. Feng, J. J. Shen and K. W. Chau. Daily reservoir runoff forecasting method using artificial neural network based on quantum-behaved particle swarm optimization. Water, 7(8), 4232-46 (2015).
[12] H. Daidi, S. Yuan and G. Mingtao. PCA-IBP model application in medium and long-term runoff forecasting. International Conference on Computer Science and Network Technology (ICCSNT). IEEE, Harbin, China, 4 pages (2016).
[13] Y. Bai, Z. Chen, J. Xie, and C. Li. Daily reservoir inflow forecasting using multiscale deep feature learning with hybrid models. Journal of Hydrology, 532, 193-206 (2016).
[14] F. Liu, F. Xu, and S. Yang. A Flood Forecasting Model Based on Deep Learning Algorithm via Integrating Stacked Autoencoders with BP Neural Network. 2017 IEEE Third International
Conference on Multimedia Big Data (BigMM). IEEE Computer Society, California, USA, 4 pages (2017).

[15] Y. Wu, Z. Liu, W. Xu, J. Feng, S. Palaiahnakote and T. Lu. Context-Aware Attention LSTM Network for Flood Prediction. 2018 24th International Conference on Pattern Recognition (ICPR). IEEE, Beijing, China, 6 pages (2018).