Evaluation of Daily Suspended Sediment Load Using Deep Learning Models

Bibhuti Bhusan Sahoo

MVR College of Engineering and Technology

Chitaranjan Dalai (✉ cdchittaranjan1@gmail.com)

Indian Institute of Technology Kharagpur <https://orcid.org/0000-0002-1268-9015>

B Srikanth

MVR College of Engineering and Technology

Mani Bhushan

Rashtrakavi Ramdhari Singh Dinkar College of Engineering begusarai

Research Article

Keywords: Deep Learning, GRU, LSTM, Missouri River, Suspended sediment load

Posted Date: March 7th, 2022

DOI: <https://doi.org/10.21203/rs.3.rs-1156624/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Evaluation of daily Suspended sediment load using Deep learning models

Bibhuti Bhusan Sahoo¹ Chitaranjan Dalai^{2*} B Srikanth³ Mani Bhushan⁴

¹Associate Professor, Department of Civil Engineering MVR College of Engineering & Technology, Vijayawada

²PhD Research Scholar, Department of Civil Engineering, Indian Institute of Technology Kharagpur, west Bengal India

³PhD Research Scholar, Department of Civil Engineering Vel Tech Rangarajan Dr. Sagunthala R&D Institute of Science and Technology, Chennai

⁴Assistant Professor Department of Civil Engineering RRS DCE Begusarai

***Corresponding Author**

Email: cdchittaranjan1@gmail.com

Chitaranjan Dalai

PhD Research Scholar, Department of Civil Engineering Indian Institute of Technology

Kharagpur, West Bengal-721302

Orcid Id: **0000-0002-1268-9015**

Abstract: Precise and reliable suspended sediment load (SSL) prediction models can help ensure the integrated water resources management in a river basin. This study considered two deep learning (DL) models i.e. gated recurrent unit (GRU) and long short term memory (LSTM) for daily SSL modeling in the Missouri River at Omaha, NE gauging station in the United States. The established models are verified with various statistical measures. The assessment of prediction accurateness of the DL models showed that the GRU was the best model in SSL prediction. The coefficient of determination was 0.871 for the GRU and 0.865 for LSTM. Besides, the GRU model has a lesser root mean square error (RMSE) and mean absolute error (MAE) compared to the LSTM. In summary, both the developed DL model can be used competently in SSL modeling. However, GRU utilizes less training constraints and thus uses a lesser amount of memory, performs quicker, and trains faster than LSTM, while LSTM needs additional data to be more accurate.

Keywords: Deep Learning; GRU; LSTM; Missouri River; Suspended sediment load

Introduction

An accurate estimate of suspended sediment load (SSL) in the river system plays a crucial role in sustainable water resources management. Transportation of the sediment in the river is a result of a complex process (Rajaei et al. 2010). In recent times, the significance of an accurate estimate of sediment load in rivers has gained considerable attention from researchers across the globe. Studies have been directed to lessen the difficulties of the problem by developing competent methods that do not require much algorithm and theory. In this manner, established techniques such as Artificial

Neural Network (ANN), Support Vector Machines (SVM), and Multiple Layer Perception (MLP) are widely used for SSL modeling. In past years, due to its capability to model complex nonlinear relationships, the use of artificial intelligence (AI) approaches is growing, especially in circumstances where the exact type of the relationship between the variables involved is unclear.

Over the past two decades, the applications of data-driven AI models gained more popularity among the researchers for SSL modeling. Some of the AI-based data-driven models used for SSL modeling includes SVM (Cimen 2008; Jie and Yu 2011; Lafdani et al. 2013), ANN (Alp and Cigizoglu 2007; Lafdani et al. 2013; Melesse et al. 2011; Zhu et al. 2007), MLP (Cigizoglu 2004), Adaptive Neuro-Fuzzy Inference System (ANFIS) (Kisi and Zounemat-Kermani 2016; Vafakhah 2013), Genetic Programming (GP) (Aytek and Kişi 2008; Kisi et al. 2012), Gene Expression Programming (GEP) (Azamathulla et al. 2013), Model Tree (MT) (Khosravi et al. 2018; Reddy and Ghimire 2009) and many more. The complexity and non-linearity of SSL variables, however, make it difficult for these models to classify the SSL-affecting processes. To improve the accuracy of SSL prediction, more advanced prediction models are therefore still needed.

Deep learning (DL), a cutting-edge technique in the field of artificial intelligence, has been proven its capabilities in various areas. Some of the successful applications of the DL model includes image classification (Li et al. 2019; Zhang et al. 2019), computer vision (O'Mahony et al. 2019; Weiss et al. 2020; Wu et al. 2020), speech recognition (Fayek et al. 2017; Kanda et al. 2019; Nassif et al. 2019; Purwins et al. 2019), time series analysis (Cai et al. 2019; Fawaz et al. 2019; Hua et al. 2019).

Long short-term memory (LSTM) is a popular DL model used for univariate time series modeling. The LSTM is a type of recurrent neural network (RNN) that accumulates extended historical data for collection, representation, and storage in the hidden memory. Also, it updates over time to ensure constancy in the time information in memory for long periods (Barzegar et al. 2020). Some of the promising application of LSTM in the field of water resources includes streamflow modeling (Hu et al. 2020; Zuo et al. 2020), evapotranspiration modeling (Afzaal et al. 2020), water quality modeling (Barzegar et al. 2020), low flow modeling (Sahoo et al. 2019), rainfall-runoff Modeling (Li et al. 2020). Another DL model named gated recurrent unit (GRU) was created as a modification of LSTM with a more straightforward structure. Some of the applications of GRU are discussed by (Ding et al. 2019; Wang et al. 2018; Yu et al. 2019).

To the best of our knowledge, there is no specific evidence for using the GRU method in modeling daily SSL. In this paper, we compare and investigate the prediction performance of GRU and LSTM for SSL modeling. The rest of the article is systematized as following; Gauging station and data collection information are described in Section 2. In Section 3, the proposed DL models, i.e., LSTM, GRU, is described. Model development and hyperparameters optimization are discussed in section 4. Model Performance evaluation is done in Sections 5. Section 6 deals with results and discussion where the application for the daily SSL and outcomes are briefed followed by conclusions.

Gauging station and data collection

Daily suspended-sediment load concentration data (tones per day) of the Missouri River at gauging station Omaha, Nebraska (Station No: 6610000, drainage area = 836048.162 km², Latitude:

41°15'32"N and Longitude: 95°55'20"W) (Fig. 1) functioned by the US Geological Survey (USGS), was used to train and test all the models developed in this study. The daily SSL data from October 2008 to September 2018 (Fig. 2) acquired from the sediment data portal of USGS (<https://cida.usgs.gov/sediment/#>).

Fig. 1 Missouri River at Omaha, NE gauging station in the United States

Fig. 2 River SSL time series (October 2008 to September 2018)

Methodology

Long short-term memory (LSTM)

LSTM was presented by Sepp Hochreiter and Jürgen Schmidhuber (Hochreiter and Schmidhuber 1997). The LSTM is a particular type of recurrent neural network (RNN) explicitly designed to evade the long-term dependency problem and to overcome some of the primary shortcomings of RNN, i.e., vanishing gradient and exploding gradient (Chung and Shin 2018). The structure of LSTM block contains a memory cell and three multiplicative gating units, an input gate which supplements information to the cell state, a forget gate, which eliminates the information that is no longer essential by the model, and an output gate, which picks the information to be shown as output. Fig. 3 shows a simple LSTM architecture.

Fig. 3 A simple LSTM architecture

$$f_t = s \left(W_f \times [h_{t-1}, x_t] + b_f \right) \quad (1)$$

$$i_t = s \left(W_i \times [h_{t-1}, x_t] + b_i \right) \quad (2)$$

$$C_t^o = \tanh \left(W_C \times [h_{t-1}, x_t] + b_C \right) \quad (3)$$

$$C_t = f_t C_{t-1} + i_t \tilde{C}_t \quad (4)$$

$$o_t = s \left(W_o \times [h_{t-1}, x_t] + b_o \right) \quad (5)$$

$$h_t = o_t \tanh(C_t) \quad (6)$$

In these formulas, all W variables are weights, and b variables are biases, and both are learned during the training phase.

More details about LSTM can be found in the papers by Hochreiter and Schmidhuber (1997) and Gers et al. (1999).

The Gated Recurrent Unit (GRU)

The GRU is a specific type of LSTM endorsed by Cho et al. (2014). It has shown many advantages over LSTM in many tasks. It trains the model with a lower training timing and fewer constraints than a typical LSTM. Fig. 4 shows the structure of a gated recurrent unit network.

Fig. 4 A simple GRU architecture

The memory state and hidden state of the LSTM are united in GRU. The constraints and control gates in the GRU network are abridged, allowing the GRU to lessen time required to train the model. The GRU framework consists primarily of the reset gate $r_{(t)}$, update gate $z_{(t)}$, fresh memory $\hat{h}_{(t)}$, and GRU present state $h_{(t)}$. Besides, $h_{(t-1)}$ it is the GRU state of the earlier moment and $x_{(t)}$ is the contribution of the present condition. Both the reset gate $r_{(t)}$ and the update gate $z_{(t)}$ collect the input containing the similar GRU unit's output obtained at an earlier phase $h_{(t-1)}$ and input data related to the current time step $x_{(t)}$. Note that W_n and U_n are the weight matrices of input and

output data from previous cells, while b_n represents bias vectors with $n \in \{r, h, z\}$. The above weights and bias together identify the aforementioned mentioned gates and fresh memory. Furthermore, $\odot$ and $\oplus$ are pointwise multiplication and pointwise addition, respectively. The most notable improvement in GRU relative to LSTM is considered to be the reduction in memory units and the drop in the number of control gates. Below are the modified gate structures discussed. First, the reset gate $r_{(t)}$ is accountable for evaluating how much of the information of the earlier phase $h_{(t-1)}$ can be used in the current state $\hat{h}_{(t)}$, which is functioned by a "sigmoid" activation function σ within 0 to 1.

$$r_{(t)} = \sigma \left(W_r \times x_{(t)} + U_r h_{(t-1)} + b_r \right) \quad (7)$$

Following, the fresh memory $\hat{h}_{(t)}$ is updated by brief the fresh input $x_{(t)}$, and the information of reset gate $r_{(t)}$ "Hyperbolic tangent" (tanh) is used to produce a vector of fresh memory. It is worth mentioning that the GRU uses hidden cells directly instead of individual memory cells as in LSTM to store memory information

$$\hat{h}_{(t)} = \tanh \left(U_h \left(r_{(t)} \odot h_{(t-1)} \right) + W_h x_{(t)} + b_h \right) \quad (8)$$

The updated gate $z_{(t)}$, which is the result of GRU simplifying the integration of input and forgotten gates in LSTM into one gate, determines the influence of the previous state $h_{(t-1)}$ on the current state via the sigmoid function. If the value is one, all the previous state information will be

transmitted to the current state. In contrast, the current state is determined only by the fresh memory $\hat{h}_{(t)}$.

$$z_{(t)} = s \left(W_z x_{(t)} + U_z h_{(t-1)} + b_z \right) \quad (9)$$

At last, the current hidden state or output $h_{(t)}$ determined by the previous state $h_{(t-1)}$ and the fresh memory $\hat{h}_{(t)}$,

$$h_{(t)} = (1 - z_{(t)}) \odot h_{(t-1)} \oplus z_{(t)} \odot \hat{h}_{(t)} \quad (10)$$

Model development and hyperparameters optimization

We trained these models using Keras deep learning library (Chollet 2015) intel core i3-3110m CPU @ 2.40ghz, 4GB RAM. While developing a deep learning model, it is crucial to select a suitable loss function and an optimizer. A loss function (or objective function) is one of the essential parameters required to compile a model and a method of evaluating how well the deep learning algorithm models the dataset. There are several different common loss functions to choose from the cross-entropy loss, the mean-squared error (MSE), the Huber loss, and the hinge loss. However, most of the studies use MSE as a loss function related to deep learning models used for regression or prediction. Adam is an adaptive learning rate optimization algorithm that has been designed specifically for training deep learning models is used as an optimizer in this study. We trained the model with other optimizers like stochastic gradient descent (SGD), Root Mean Square Propagation (RMSprop), Adaptive Gradient algorithm (AdaGrad), but Adam performed better

than others. Optimizing the hyperparameters of any deep learning model is a vital part of any model for better results.

The type of activation mechanism chosen for nodes is another important concern. After a trial and error procedure, the Sigmoid and linear activation functions are engaged for hidden and output layers, respectively, for the LSTM model. We tuned the hyperparameters of each model based on their Keras packages. The amount of epochs of each model is 500. For LSTM, GRU, we set their units, batch size as 5, and 50, respectively. We used 75 % of data for training 25 % of the data for testing. From the training set, 33 % of the data is randomly used for validation. Further the original SSL data is transformed using the equation (11) before training the model, and finally, in the post-processing stage, the predicted value was back-transformed into their actual scale.

$$d' = \ln(d) \quad (11)$$

Where d is the original SSL value, and d' is the transformed value.

The analysis was carried out with various time lags as input to develop the models i.e. four models (M1, M2, M3, M4) for GRU and four models (M1, M2, M3, M4) for LSTM with lag 1, lag 3, lag 6 and lag 12 as shown in table 1

Table 1 Four models used in the course of the analysis with the corresponding predictor variable as input for GRU and LSTM

Where SSL_t denotes the suspended sediment load at time t

Model Performance Evaluation

In the study, the performance of the developed models was evaluated using various statistical matrices and compare their values in the testing set.

Coefficient of determination

$$R^2 = 1 - \frac{\sum_{i=1}^n (SSL_{i(Observe)} - SSL_{i(Predict)})^2}{\sum_{i=1}^n (SSL_{i(Observe)} - \overline{SSL_{Observe}})^2} \quad (12)$$

Root-mean-square error (RMSE)

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (SSL_{i(Observe)} - SSL_{i(Predict)})^2} \quad (13)$$

Mean absolute error (MAE)

$$MAE = \frac{1}{n} \sum_{i=1}^n |SSL_{i(Observe)} - SSL_{i(Predict)}| \quad (14)$$

Nash-Sutcliffe Efficiency (NSE)

$$NSE = 1 - \frac{\sum_{i=1}^n (SSL_{i(Observe)} - SSL_{i(Predict)})^2}{\sum_{i=1}^n (SSL_{i(Observe)} - \overline{SSL_{Observe}})^2} \quad (15)$$

Where $SSL_{(Observe)}$ is the observed suspended sediment load concentration, $SSL_{(Predict)}$ is the predicted suspended sediment load concentration, and $SSL_{(Mean)}$ is the mean of the observed suspended sediment concentrations and n is the number of data points. In regression, the R^2 is a statistical measure of how well the regression predictions approximate the real data points. An R^2

value of 1 indicates that the regression predictions perfectly fit the measured data, while the value of 0 indicates no linear relationship. NSE indicates how well the plot of observed versus simulated data fits the 1:1 line that ranges from minus infinity to 1, with greater values demonstrating superior agreement (Nash and Sutcliffe 1970). The RMSE parameter provides a quantitative indication of the model's absolute error in terms of the units of the variable, with the characteristic that larger errors obtain greater attention than smaller ones. This characteristic can help eliminate techniques with meaningful errors.

Results and discussion

The focus of this research was to explore the efficiency of two DL models, ie the GRU and LSTM models to predict SSL. The models were assessed and compared using statistical indicators, including R^2 , RMSE, NSE, and MAE, in the training and testing periods. The training and testing results are showed in Table 2 with various lags (lag 1, lag 3, Lg 6, lag 12). The GRU model with input lag 3 performed better among other all models developed with various inputs in the case of GRU. The various statistical scores which support the superiority of the lag 3 GPU model comprise with lowest RMSE (13747), MAE (54882), and higher NSE (0.830), R^2 (0.871). In the case of the LSTM model, the lag 1 model outperformed with RMSE (13778), MAE (5626), and higher NSE (0.829), R^2 (0.865). While comparing GPU and LSTM model GPU Performed better with lag 3 with the lowest RMSE error. The scatter plots of the observed and predicted SSL in the testing period (Fig. 5) for GRU with lag 3 and (Fig. 6) for LSTM with lag 1 show that the GRU model performed better than the other models. The training and validation loss of the developed model i.e GPU (lag 3) and LSTM (lag 1) is also shown in Fig. 7 and Fig. 8 respectively.

Table 2 Evaluation of models in SSL prediction on the training and testing period for lag 1, lag 3, lag 6, lag 12.

Fig. 5 Scatter plot showing observed and predicted SSL during testing period GRU

Fig. 6 Scatter plot showing observed and predicted SSL during testing period LSTM

Fig.7 Model loss during training and validation period GRU (lag3)

Fig. 8 Model loss during training and validation period LSTM (lag1)

Conclusions

In the present study, we compare and analyze the prediction performance of two deep learning models namely GRU and LSTM for SSL modeling for the Missouri River at Omaha, Nebraska in the United States. To examine the predictive performance of the model, the results were studied using various statistical matrices. Evaluation results of the actual performance of the models showed that the GRU model outperformed the LSTM model. Finally, we advocate that the developed deep learning models can be used for modeling the daily suspended sediment load. GRU utilizes less training constraints and thus utilizes fewer memory, performs quicker, and trains faster than LSTM, while LSTM needs additional data to be more accurate. In short, if the available data is more or the precision is very critical, please go to LSTM, while GRU goes for less memory consumption and faster activity.

Ethics declarations**Author contributions:**

BBS contributed to conceptualization, methodology, software, and writing original draft preparation. BS performed interpretation of results, review, and editing contributed to visualization. CD performed review and editing. MB carried out formal analysis. All authors read and approved the final manuscript.

Data availability statement:

Data availability depends upon the request of the researchers.

Conflict of interest:

The authors declare that they have no conflicts of interest.

Ethical approval:

This article does not contain any studies with human participants or animals performed by any of the authors.

Informed consent:

Informed consent was obtained from all individual participants included in the study.

References:

- Afzaal H, Farooque AA, Abbas F, Acharya B, Esau T (2020) Computation of Evapotranspiration with Artificial Intelligence for Precision Water Resource Management *Applied Sciences* 10:1621
- Alp M, Cigizoglu HK (2007) Suspended sediment load simulation by two artificial neural network methods using hydrometeorological data *Environmental Modelling & Software* 22:2-13
- Aytek A, Kişi Ö (2008) A genetic programming approach to suspended sediment modelling *Journal of hydrology* 351:288-298
- Azamathulla HM, Cuan YC, Ghani AA, Chang CK (2013) Suspended sediment load prediction of river systems: GEP approach *Arabian Journal of Geosciences* 6:3469-3480
- Barzegar R, Aalami MT, Adamowski J (2020) Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model *Stochastic Environmental Research and Risk Assessment*:1-19
- Cai M, Pipattanasomporn M, Rahman S (2019) Day-ahead building-level load forecasts using deep learning vs. traditional time-series techniques *Applied energy* 236:1078-1088
- Cho K, Van Merriënboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y (2014) Learning phrase representations using RNN encoder-decoder for statistical machine translation *arXiv preprint arXiv:1406.1078*
- Chollet F (2015) Keras: Deep learning library for theano and tensorflow URL: <https://keras.io/k7T1>
- Chung H, Shin K-s (2018) Genetic algorithm-optimized long short-term memory network for stock market prediction *Sustainability* 10:3765
- Cigizoglu HK (2004) Estimation and forecasting of daily suspended sediment data by multi-layer perceptrons *Advances in Water Resources* 27:185-195
- Cimen M (2008) Estimation of daily suspended sediments using support vector machines *Hydrological Sciences Journal* 53:656-666
- Ding M, Zhou H, Xie H, Wu M, Nakanishi Y, Yokoyama R (2019) A gated recurrent unit neural networks based wind speed error correction model for short-term wind power forecasting *Neurocomputing* 365:54-61
- Fawaz HI, Forestier G, Weber J, Idoumghar L, Muller P-A (2019) Deep learning for time series classification: a review *Data Mining and Knowledge Discovery* 33:917-963
- Fayek HM, Lech M, Cavedon L (2017) Evaluating deep learning architectures for Speech Emotion Recognition *Neural Networks* 92:60-68
- Gers FA, Schmidhuber J, Cummins F (1999) Learning to forget: Continual prediction with LSTM
- Hochreiter S, Schmidhuber J (1997) Long short-term memory *Neural computation* 9:1735-1780
- Hu Y, Yan L, Hang T, Feng J (2020) Stream-Flow Forecasting of Small Rivers Based on LSTM *arXiv preprint arXiv:2001.05681*
- Hua Y, Zhao Z, Li R, Chen X, Liu Z, Zhang H (2019) Deep learning with long short-term memory for time series prediction *IEEE Communications Magazine* 57:114-119

- Jie L, Yu S Suspended sediment load estimate using support vector machines in Kaoping River basin. In: International Conference on Consumer Electronics, Communications and Networks (IEEE), XianNing, China, 2011. pp 16-18
- Kanda N, Fujita Y, Horiguchi S, Ikeshita R, Nagamatsu K, Watanabe S Acoustic modeling for distant multi-talker speech recognition with single-and multi-channel branches. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2019. IEEE, pp 6630-6634
- Khosravi K, Mao L, Kisi O, Yaseen ZM, Shahid S (2018) Quantifying hourly suspended sediment load using data mining models: case study of a glacierized Andean catchment in Chile Journal of Hydrology 567:165-179
- Kisi O, Dailr AH, Cimen M, Shiri J (2012) Suspended sediment modeling using genetic programming and soft computing techniques Journal of Hydrology 450:48-58
- Kisi O, Zounemat-Kermani M (2016) Suspended sediment modeling using neuro-fuzzy embedded fuzzy c-means clustering technique Water Resources Management 30:3979-3994
- Lafdani EK, Nia AM, Ahmadi A (2013) Daily suspended sediment load prediction using artificial neural networks and support vector machines Journal of Hydrology 478:50-62
- Li S, Song W, Fang L, Chen Y, Ghamisi P, Benediktsson JA (2019) Deep learning for hyperspectral image classification: An overview IEEE Transactions on Geoscience and Remote Sensing 57:6690-6709
- Li W, Kiaghadi A, Dawson CN (2020) High Temporal Resolution Rainfall Runoff Modelling Using Long-Short-Term-Memory (LSTM) Networks arXiv preprint arXiv:200202568
- Melesse A, Ahmad S, McClain M, Wang X, Lim Y (2011) Suspended sediment load prediction of river systems: An artificial neural network approach Agricultural Water Management 98:855-866
- Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models part I—A discussion of principles Journal of hydrology 10:282-290
- Nassif AB, Shahin I, Attili I, Azzeh M, Shaalan K (2019) Speech recognition using deep neural networks: A systematic review IEEE Access 7:19143-19165
- O'Mahony N et al. Deep learning vs. traditional computer vision. In: Science and Information Conference, 2019. Springer, pp 128-144
- Purwins H, Li B, Virtanen T, Schlüter J, Chang S-Y, Sainath T (2019) Deep learning for audio signal processing IEEE Journal of Selected Topics in Signal Processing 13:206-219
- Rajae T, Nourani V, Zounemat-Kermani M, Kisi O (2010) River suspended sediment load prediction: application of ANN and wavelet conjunction model Journal of Hydrologic Engineering 16:613-627
- Reddy MJ, Ghimire BN (2009) Use of model tree and gene expression programming to predict the suspended sediment load in rivers Journal of Intelligent Systems 18:211-228
- Sahoo BB, Jha R, Singh A, Kumar D (2019) Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting Acta Geophysica 67:1471-1481
- Vafakhah M (2013) Comparison of cokriging and adaptive neuro-fuzzy inference system models for suspended sediment load forecasting Arabian Journal of Geosciences 6:3003-3018
- Wang Y, Liao W, Chang Y (2018) Gated recurrent unit network-based short-term photovoltaic forecasting Energies 11:2163
- Weiss Y, Ferrari V, Sminchisescu C, Hebert M (2020) Advances in Architectures and Theories for Computer Vision International Journal of Computer Vision:1-2
- Wu X, Sahoo D, Hoi SC (2020) Recent advances in deep learning for object detection Neurocomputing
- Yu Z, Niu Z, Tang W, Wu Q (2019) Deep learning for daily peak load forecasting—a novel gated recurrent neural network combining dynamic time warping IEEE Access 7:17184-17194
- Zhang J, Xie Y, Wu Q, Xia Y (2019) Medical image classification using synergic deep learning Medical image analysis 54:10-19

- Zhu Y-M, Lu X, Zhou Y (2007) Suspended sediment flux modeling with artificial neural network: an example of the Longchuanjiang River in the Upper Yangtze Catchment, China *Geomorphology* 84:111-125
- Zuo G, Luo J, Wang N, Lian Y, He X (2020) Decomposition ensemble model based on variational mode decomposition and long short-term memory for streamflow forecasting *Journal of Hydrology*:124776

Figures

Figure 1

Missouri River at Omaha, NE gauging station in the United States

Figure 2

River SSL time series (October 2008 to September 2018)

Figure 3

A simple LSTM architecture

Figure 4

A simple GRU architecture

(a)

(b)

Figure 5

Scatter plot showing observed and predicted SSL during testing period GRU

(a)

(b)

Figure 6

Scatter plot showing observed and predicted SSL during testing period LSTM

Figure 7

Model loss during training and validation period GRU (lag3)

Figure 8

Model loss during training and validation period LSTM (lag1)