Time Series Prediction of Pore Water Pressure on Earth Dam Slopes Based on Recurrent Neural Network

Lin Wang, Junrui Chai*

State Key Laboratory of Eco-hydraulics in Northwest Arid Region, Xi'an University of Technology, Xi'an, 710048, China

* Corresponding author's e-mail: jrchai@xaut.edu.cn

Abstract. Landslides are common geologic hazards in engineering, often causing serious destructive consequences. The study of pore water pressure distribution on slopes has a positive effect on mitigating the hazards of landslides, but due to the limitations of the complex physical mechanisms in engineering practice, the variability of natural space, etc., which leads to the existing theoretical studies can not completely reflect the law of pore water pressure, many scholars began to use machine learning methods applied to the prediction of pore water pressure. This paper mainly uses the recurrent neural network and its three variants to predict the pore water pressure monitored in the actual project, and compares the performance of the four models. The study shows that the four models have good performance, in which the integrated training time and training effect of Gated recurrent unit model is relatively better, while the adjustment of parameters can effectively improve the training effect of the model as well as the training time.

Keywords: Slop; Recurrent neural network; Pore-water pressure; Long short-term memory; Gated recurrent unit; Bidirectional recurrent neural network

1 INTRODUCTION

Landslides are common geologic hazards in engineering and often cause severe destructive consequences1-3. It has been proposed that soil slope damage is usually caused by the pore water pressure distribution on the slope surface and the change of groundwater level. Detection of pore water pressure in engineering can provide useful information for evaluating the seepage field and slope stability4-6. Many theoretical explanations and numerical simulations have been proposed by many scholars to address this issue. However, the complex physical mechanisms, natural spatial variations, and limitations of the monitoring instrumentation are such that the measured pore water pressure does not fully reflect the complexity of the project.

With the development of artificial intelligence technology, some scholars have begun to apply machine learning methods to engineering problems. Landslides.Mustafa et al. 7-9predicted the variation of pore water pressure with rainfall infiltration using multilayer perceptron (MLP) and radial basis function (RBFNN) for several slopes in Singapore.Wei et al.10 predicted pore water pressure with rainfall infiltration by using

© The Author(s) 2024
B. Yuan et al. (eds.), *Proceedings of the 2024 8th International Conference on Civil Architecture and Structural Engineering (ICCASE 2024)*, Atlantis Highlights in Engineering 33,
https://doi.org/10.2991/978-94-6463-449-5_66

a looped neural network for a case study on a natural slope in Hong Kong using recorded rainfall as well as pore water pressure data. rainfall as well as pore water pressure data and used recurrent neural networks to predict pore water pressure.

Measured data such as air pressure and rainfall may not be available in the actual project. Therefore, in this paper, the measured values of pore water pressure are used as inputs. Recursive neural network and its variants are applied to analyze the time series of pore water pressure. The Nazixia project is used as a case study. This paper firstly introduces four kinds of recurrent neural networks. Then the recurrent neural network was applied to the time series analysis of pore water pressure engineering monitoring data. A systematic comparison of four recurrent neural networks was made. The reliability of recurrent neural networks in pore water pressure time series analysis was investigated.

2 METHODOLOGY

In this study, only monitoring data of pore water pressure was used as input. Therefore it cannot be considered as a static problem. Time series need to be considered. Therefore recurrent neural networks are used.

2.1 Standard Recurrent Neural Network

Recurrent neural networks(RNN) were first proposed by Elman (1990)11. Recurrent neural networks were originally proposed to solve time series problems. The structure of recurrent neural network is shown in Fig.1. Recurrent neural networks are built on the basis of fully connected neural networks. Unlike fully connected neural networks, neurons in the hidden layer of recurrent neural networks are interconnected. Therefore, the time-related input information can be transmitted through the connections between neurons, thus achieving the purpose of dealing with time series problems. The problem of gradient vanishing occurs due to the optimization of the loss function using Back Propagation Through Time(BPTT) during backpropaga-tion[12]. Therefore, standard RNNs can preserve short-term memory but not long-term memory.

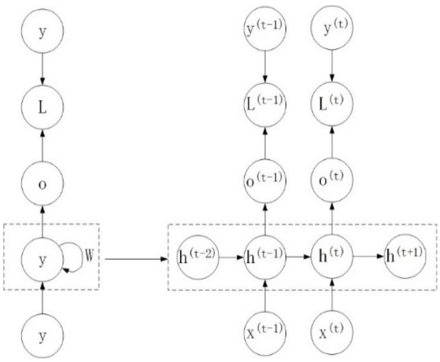

Fig. 1. Structures of standard RNN

2.2 Long Short-term Memory

Long short-term memory (LSTM) is a variant of recurrent neural network (RNN) proposed by Hochreiter and Schmidhuber (1997)13, which is used to solve the problem of gradient vanishing in RNN for long time series prediction. Compared with RNN, LSTM mainly adds a memory unit, which consists of a forgetting gate, an input gate, and an output gate for controlling the memory and transportation of information in the model.The structure of LSTM is shown in Fig. 2. The LSTM can have the ability to add and remove messages through the gating unit. Through the gate it is possible to selectively decide whether a message passes or not.

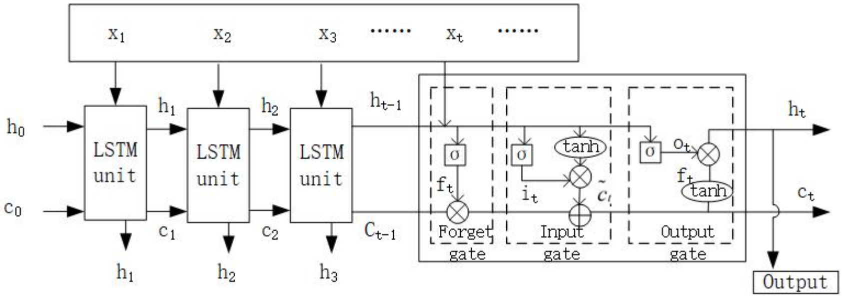

Fig. 2. Structures of LSTM

2.3 Gated Recurrent Unit

Cho et al. (2014)14 proposed Gated recurrent unit (GRU) based on Long short-term memory (LSTM) optimized for the structure of LSTM.The structure of GRU is shown in Fig. 3. Compared with the structure of LSTM, GRU combines the forgetting gate and the output gate into a single update gate, and the final model is simpler than the LSTM model. The advantage of GRU over LSTM is its simple structure, which saves a lot of time when training the model. Generally speaking, the performance of LSTM is higher than that of GRU, but in a smaller dataset, GRU will show better performance.

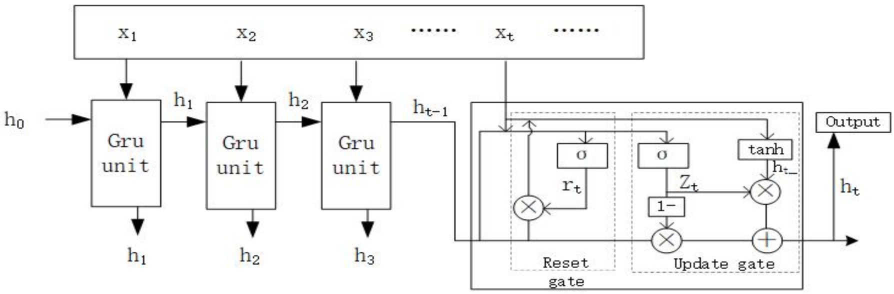

Fig. 3. Structures of GRU

2.4 Bidirectional Recurrent Neural Network

Bidirectional recurrent neural network (BiRNN) is a combination of two RNNs. One RNN processes the input sequence from front to back and one RNN processes the input sequence from back to front. This structure captures information from both the front and back directions in the sequence, thus improving the performance of the model. The structure of BiRNN is shown in Fig.4. BiRNN is able to capture both forward and backward information in a sequence and therefore can better handle long sequences of data, which can relatively improve the accuracy of the model. Forward propagation of the network requires forward and backward recursion in a bidirectional layer, and back-propagation of the network also depends on the results of forward propagation. Therefore, the gradient solution will have a very long chain and the model training is slow.

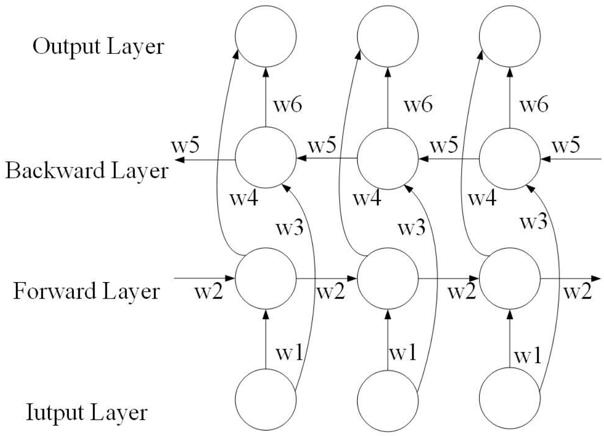

Fig. 4. Structures of Bi-RNN

3 CASE STUDY

3.1 Description of the Study Area

Datong River is a secondary tributary of the Yellow River, a tributary of Huangshui River, originated in Tianjun County, Qinghai Province, flowing from northwest to southeast through Qinghai Province, Qilian, Haiyan, Menyuan, Mutual, Ledu and other counties and Gansu Province, Tianzhu, Yongdeng two counties, and finally in Qinghai Province, Minhe County, near the town of Heungtang injected into Huangshui.

The Datong River is a mountainous river, with abundant water in the main stream and steep longitudinal slopes, harboring rich water energy resources. The length of the main channel of the Datong River is 560.7km, with a longitudinal gradient of 4.56‰, a watershed area of 15,130km2, and a catchment area of 6,593km2 above the Nako Gorge dam site.

Nazixia Hydropower Station is located in the northeastern part of Qinghai Province, Menyuan County, Yanmaituhu Township and Qilian County, Huangcheng Township of the junction, in the upper reaches of the Datong River at the end of the section (upper reaches: Heyuan ~ Ga Datan, the middle reaches: Ga Datan ~ Liancheng, the lower reaches: Liancheng ~ the mouth of the Datong River), the geographic location of the longitude of 98° 30′ ~ 103 ° 25′ East, latitude of 36 ° 30′ ~ 38 ° 25′, the highway mileage through the Qingshizui (50km) - Darbanshan - Datong County - Xining City, about 186km.

3.2 Data Processing

Data Selection.

In this study, representative p1-p6 were selected from the installed manometers. The measured data has data that cannot be detected in cases such as villagers' obstruction. Therefore, there will be problems such as missing data when taking one day as the time unit. The data of missing days will not be entered when building the model training set. Therefore, the time interval of the data in the training set and the test set in this paper is not strictly in units of one day.

Taking P1 manometer as an example, there are 602 data of P1 manometer data based on literature recommendations, the first 75% of data (the first 450 data) are selected as the training set and the last 25% of data (the last 150 data) as the test set. In the model training of this paper, the validation set and the test set share a common data set.

The model takes the previous 30 time periods as inputs and the 31st time period as the predicted value. That is, the monitoring data of the previous 30 days is used to predict the data of the 31st day. For normalization the maximum value of the training set is obtained and the minimum value of these attributes inherent in the training set. Normalization is performed on the training set and the test set is normalized with the properties of the training set. In this the training set is randomly disrupted to improve the model training accuracy.

Normalization.

Normalization is a commonly used data processing tool in machine learning. In machine learning, different features in the feature vector usually have different magnitudes, which often affects the data processing results. Normalization can make different data limited to the same magnitude and improve the efficiency and accuracy of model training15. Combined with the feature data considerations in this paper, the maximum-minimum normalization (see Eq.1) is chosen to transform the original data to the range of [0,1] in a linearization method.

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)} \tag{1}$$

Where x and x' are the original data and normalized data respectively, min(x) and max(x) are the minimum and maximum values in the original data respectively.

3.3 Development of Models

Model Structure.

This paper focuses on the study of common recurrent neural networks and their various variants, including four neural networks, simpleRNN, LSTM, GRU, and BiRNN, are discussed. The code is mainly written through Python Tensorflow. The three networks, simpleRNN, LSTM, and GRU, are similar in structure, so all three networks in this study are two-layer networks backed by a Dense layer output.

Model Optimization.

The optimization of the model structure in this study focuses on the dropout layer as well as the Adam optimization. Dropout is a commonly used optimization tool in neural network training, proposed by Hinton et al17. Its purpose is to randomly remove neurons in order to prevent overfitting in the training phase. RNN needs to keep the memory of time, so using traditional dropout method on RNN is not efficient. Therefore the method of RNNDrop is used. In this research paper, the framework used is TensorFlow, call Keras.layers.Dropout to realize the dropout of recurrent neural network, which is a high-level API that can realize the function according to the current state.

ADAM is an adaptive momentum stochastic optimization method, which is used to solve optimization problems with large data volume and high feature latitude in machine learning16. It is one of the most commonly used optimizer algorithms in machine learning nowadays. Adam optimization algorithm applied to non-convex optimization problems has more advantages: straightforward implementation, efficient computation, less memory required, invariance of gradient diagonal scaling. Suitable for solving optimization problems containing large-scale data and parameters for non-stationary objectives. Suitable for solving problems containing very noisy or sparse gradients, where the hyperparameters can be interpreted very intuitively and essentially require very little tuning parameterization.

4 RESULTS AND DISCUSSION

4.1 Comparison Between Models

This paper focuses on recurrent neural networks and their common variants, and mainly focuses on the following four neural networks: SimpleRNN, LSTM, GRU and Bi-RNN, which can be categorized into Bi-LSTM and Bi-GRU due to the characteristics of the network structure of the Bi-RNN.Taking the seepage manometer P1 as a case study, the results of four neural network models are shown in Fig 5.

Fig. 5 shows the training results of the better (coefficient of determination R2 closest to 1) of the training results of each model after debugging. The decision coefficients of the models are 0.860819 (RNN), 0.892751 (GRU), 0.881514 (LSTM), 0.892220 (Bi-GRU), 0.904257 (Bi-LSTM), respectively. It is easy to see that each model can have better training accuracy after debugging the parameters, which fully proves the robustness of machine learning. However, there is still a gap between the running efficiency and stability among the models. SimpleRNN in the TensorFlow 2.x version used in this study uses the code of TensorFlow 1.x, so the running time is not comparable to that between LSTM and GRU. Therefore, SimpleRNN will not be compared together in terms of runtime for the time being. Theoretically, the training time of SimpleRNN should be smaller than LSTM and GRU.

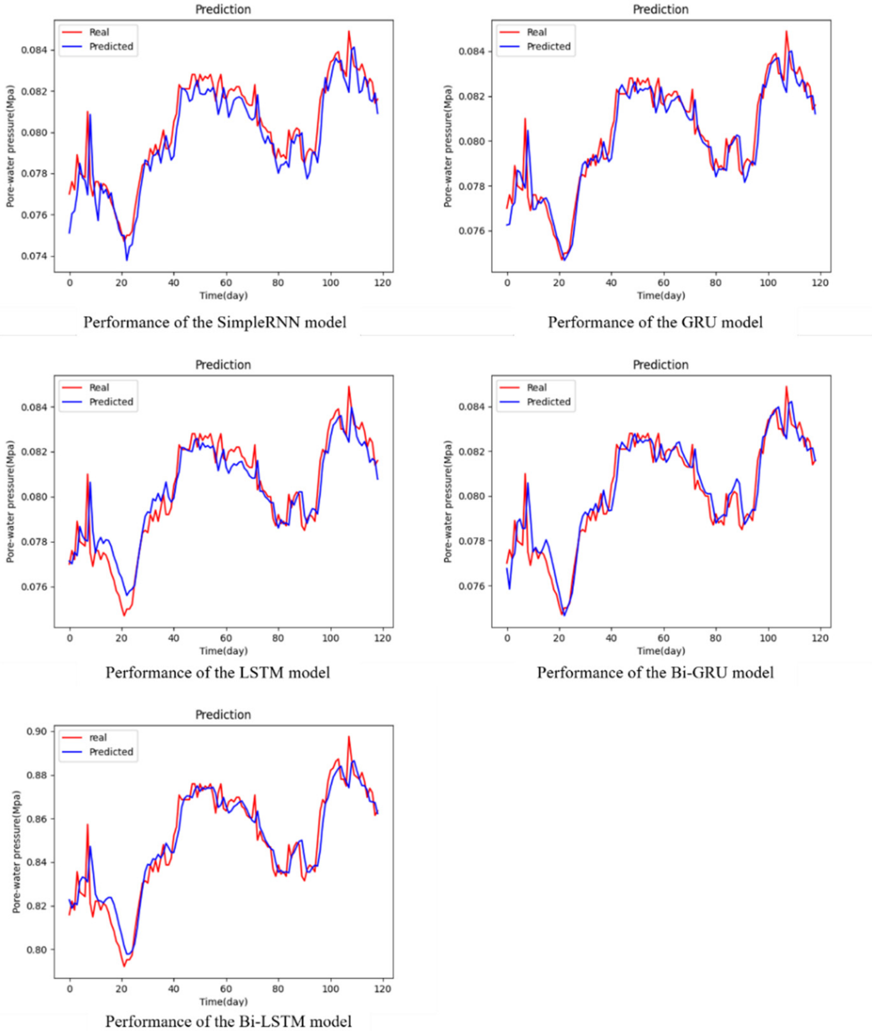

Fig. 5. Model training results

Multi-layer models can effectively improve the accuracy of model prediction as well as the ability to solve complex problems. Therefore the simpleRNN, GRU, and LSTM models are all two-layer models. The Bi-RNN itself is only a one-layer model because its network structure is quite different from the other three. Table 1 summarizes the statistics of the run results after 10 runs of the five models. Table 1 shows that SimpleRNN is worse than the other four models in terms of training accuracy. GRU as well as LSTM have higher performance in terms of accuracy as well as training time. GRU has higher efficiency due to its simpler structure compared to LSTM and takes less time to train compared to LSTM. Bi-RNN has a special network structure, and the training time is much longer than the conventional RNN model, but Bi-RNN has a higher upper limit of accuracy than the other models. Taken together, the GRU model is more suitable for time series prediction of infiltration pressure.

Table 1. Comparison of prediction performance of models at the testing period

model	R^2_max	R^2_mean	time(s)
SimpleRNN	0.88	0.76	
GRU	0.89	0.81	75.19
LSTM	0.88	0.83	80.47
Bi-GRU	0.89	0.69	495.04
Bi-LSTM	0.90	0.70	603.83

4.2 Discussion of Parameter Optimization

There are situations such as missing data in the engineering data used in this study, so the maximum value of R^2 in the roving results differs from the average value. With the network structure determined, to improve the training effect of the model it is necessary to optimize the hyperparameters of the model. In this paper, we focus on the three parameters hidden_dim,epoch,batch_size of GRU model.

As the memory architecture of CPU and GPU batch_size is mainly determined from [8,16,32,64,128]. The epoch is determined from [10,50,100,150,200,250,300,400], and hinnden_dim is traversed over 4096 parameters between (16,16) and (80,80).

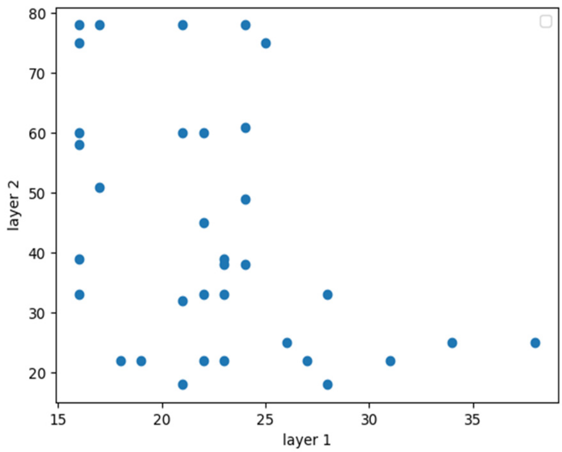

Fig. 6. Hidden neuron distribution for $R^2 > 0.89$

Fig.6 shows the distribution of hidden dim for all $R^2 > 0.89$ for 4096 parameters between (16,16) and (80,80), and it can be seen that there is no distribution when the first layer neurons are greater than 25 and the second layer neurons are greater than 40, and there is no distribution after the first layer hidden neurons are greater than 40. Thus the number of hidden neurons in the first layer is not recommended to be set too large, and the number of hidden neurons in the second layer does not seem to be limited.

Comparing the R2 corresponding to batch size and epoch, it can be found that for batch size, the impact on model accuracy is not large, for larger batch size can significantly reduce the training time, but need to increase the number of training times to ensure the accuracy of the training, for example, in the case of batch size = 128, epoch = 50, the model predicts a relatively smooth curve, while after increasing the epoch, the model predicts a curve that is more in line with the real curve. For example, when the batch size=128 and epoch=50, the curve predicted by the model is relatively smooth, while the curve predicted by the model after increasing epoch is more in line with the real curve. epoch has a relatively large impact on the model, and when epoch is too large, it will lead to overfitting of the model and seriously affect the accuracy of the model. For the GRU model, the impact of epoch too small on the model does not seem to be reflected.

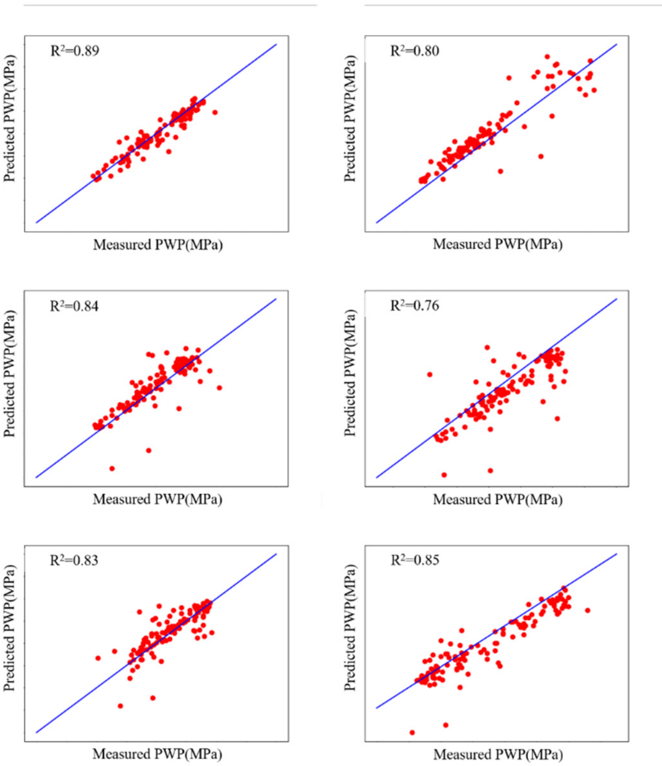

Fig. 7. Comparison of predicted and measured pore-water pressures in the testing period at P1-P6

4.3 Model Performance Evaluation

This paper focuses on the application of neural network modeling in seepage detection in earth and rock dams. Mainly focusing on the seepage pressure inside the dam body, the measured data from the seepage manometer placed inside the dam body has been used to predict the seepage pressure inside the dam in a time series by means of a recurrent neural network. A comparison of various models of recurrent networks has been made in the previous section, and the GRU model has been selected after comprehensive consideration. This section focuses on evaluating the performance of the GRU model. In this section, the p1 to p6 seepage manometer measurements are input to the GRU model to verify its applicability.

Fig.7 shows the prediction results of the model on the data of six seepage manometers, and the prediction results are relatively consistent with the real value, and the R2 can basically reach more than 0.8. Since the model has not been de-bugged in detail except for the data of the p1 seepage manometer, the accuracy of the results of the model run with the data of the rest of the seepage manometers as the inputs is relatively poor, but the applicability of the model can still be demonstrated.

5 CONCLUSIONS

Machine learning methods are widely used in time series prediction, but recurrent neural networks are seldom used in the application of seepage in earth and rock dams. In this paper, four common recurrent neural networks, namely SimpleRNN, GRU, LSTM and Bi-RNN, are used to train and compare the results, and discuss the super- parameter optimization of the models and their adaptability. It is concluded that in seepage detection, the GRU model is more suitable in terms of training accuracy and training efficiency with a relatively small amount of data. In the two-layer GRU model,batch size has little effect on the accuracy of model training, but it plays a great role in the running time of the model, and the overfitting phenomenon brought by too large epoch has a greater effect on the training accuracy of the model, and the number of neurons in the first hidden layer should not be too large. The prediction of multiple osmometer data also proves the universality of the GRU model.

The input data of this study only use the monitoring data itself as input, and it can be foreseen that some time series data in water conservancy projects can be predicted by the method of this study. However, it can also be seen that the prediction results using only the monitoring data itself as input are still insufficient in terms of accuracy. In summary, the method explored in this paper has some applicability, but it is difficult to achieve the ideal accuracy, and the prediction model of neural network can provide some reference for the actual project.

REFERENCES

1. Tang, H., Wasowski, J., Juang, C.H., 2019. Geohazards in the three Gorges Reservoir Area, China – Lessons learned from decades of research. Eng. Geol. 261, 105267 https://doi.org/10.1016/j.enggeo.2019.105267.
2. Yin, Y.P., Wang, L.Q., Zhang, W.G., Zhang, Z.H., Dai, Z.W., 2022. Research on the collapse process of a thick-layer dangerous rock on the reservoir bank. Bull. Eng.
3. Wang, L.Q., Xiao, T., Liu, S.L., Zhang, W.G., Yang, B.B., Chen, L.C., 2023. Quantification of model uncertainty and variability for landslide displacement prediction based on Monte Carlo simulation. Gondw. Res. https://doi.org/10.1016/j.gr.2023.03.006.
4. Sung, E.C., Seung, R.L., 2002. Evaluation of surficial stability for homogeneous slopes considering rainfall characteristics. J. Geotech. Geoenviron. Eng. 128 (9), 756– 763.
5. Gerscovich, D.M.S., Vargas, E.A., Campos, T.M.P.D., 2006. On the evaluation of unsaturated flow in a natural slope in Rio de Janeiro, Brazil. Eng. Geol. 88, 23– 40.
6. Zhang, L.L., Zhang, J., Zhang, L.M., Tang, W.H., 2011. Stability analysis of rainfall-induced slope failure: a review. Geotech. Eng. 164, 299– 316.
7. Mustafa, M.R., Bhuiyan, R.R., Isa, M.H., Saiedi, S., Rahardjo, H., 2012a. Effect of antecedent conditions on prediction of pore-water pressure using artificial neural networks. Mod. Appl. Sci. 6 (2), 6– 15.
8. Mustafa, M.R., Rezaur, R.B., Rahardjo, H., Isa, M.H., 2012b. Prediction of pore-water pressure using radial basis function neural network. Eng. Geol. 135– 136, 40– 47.
9. Mustafa, M.R., Rezaur, R.B., Rahardjo, H., Isa, M.H., Arif, A., 2015. Artificial neural network modeling for spatial and temporal variations of pore-water pressure responses to rainfall. Adv. Meteorol. 2015, 1– 12.
10. Wei, X.; Zhang, L.; Yang, H.Q.; Zhang, L.; Yao, Y .P . Machine learning for pore-water pressure time-series prediction: Application of recurrent neural networks. Geosci. Front. 2021, 12, 453–467.
11. Elman J L. Finding structure in time[J]. Cognitive science, 1990, 14(2): 179-211.
12. Pascanu, R., Mikolov, T., Bengio, Y., 2012. Understanding the exploding gradient problem. CoRR abs/1211 5063, 2
13. Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural Comput. 9,1735–1780
14. Cho, K., Merrienboer, B.V., 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.
15. Rojas R. Neural networks: a systematic introduction[M]. Springer Science & Business Media, 2013.
16. Kingma D P, Ba J. Adam: A method for stochastic optimization[J]. arXiv preprint arXiv:1412.6980, 2014.
17. N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting.

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.