Received March 4, 2022, accepted April 2, 2022, date of publication April 5, 2022, date of current version April 13, 2022.

Digital Object Identifier 10.1109/ACCESS.2022.3165051

Landslide Risk Prediction Model Using an Attention-Based Temporal Convolutional Network Connected to a Recurrent Neural Network

DI ZHANG, JIACHENG YANG, FENGLIN LI, SHUAI HAN, LEI QIN, AND QING LI, (Member, IEEE)

National and Local Joint Engineering Laboratory of Disaster Monitoring Technology and Instruments, China Jiliang University, Hangzhou 310018, China

Corresponding author: Qing Li (lq13306532957@163.com)

This work was supported in part by the National Key Research and Development Program Project under Grant 2017YFC0804604, and in part by the Zhejiang Key Research and Development Program Project under Grant 2018C03040.

ABSTRACT Landslide risk assessment is an important component of the landslide research field. For the problem of landslide assessment indicators, we utilize the TOPSIS-Entropy method to assess the risk situation of landslide occurrences, which is easy to obtain directly from sensor data. By using the TOPSIS-Entropy method in landslide datasets, the instability margins of landslide risk are obtained, reflecting the current instability probability of the landslide body. For the landslide prediction issue, deep neural networks are used to predict the corresponding landslide instability margins (LIMs). Attention mechanism-based (Attn) temporal convolutional networks (TCN) connected with recurrent neural network (RNN) models for landslide risk prediction are proposed, including TCN-Attn-RNN and RNN-Attn-TCN, which both use an encoder–decoder architecture. The encoder in the first model uses the temporal convolutional network (TCN), and the decoder uses a neural network with an RNN architecture, including long short-term memory (LSTM) networks, gated recurrent units (GRUs), and their derivative algorithms. In the second model, the encoder uses a neural network with an RNN architecture, and the decoder uses a TCN. Combining the TOPSIS-Entropy method with TCN-Attn-RNN and RNN-Attn-TCN, reliable prediction models of landslide risk are proposed. By building a landslide simulation platform, we obtained landslide data. Compared to their counterparts, the proposed prediction models of landslide risk instability margins have better predictive effects.

INDEX TERMS Landslide risk, TOPSIS-Entropy, instability margin, attention mechanism, temporal convolutional network, recurrent neural network.

I. INTRODUCTION

Landslide risk assessment is a critical technical tool for forecasting the occurrence of deadly landslides. However, a reliable model for predicting landslide risk still faces challenging issues. First, the identification and modeling of landslide hazard triggering factors are important technical difficulties. Second, landslide early warning is hampered by ambiguity in the temporal and geographical forecasting of landslide risk. Third, the quantitative evaluation of the landslide hazard triggering factors cannot be applied to actual sites and still requires in-depth study. The goal in the landslide research field is to identify suitable assessment metrics and build accurate models to predict landslide risk.

Multisensor analysis still remains difficult. Aiming to address the issues of communication in the Internet of Things (IoT), Joshi *et al.* presented a useful approach using edge computing to process and analyze landslide data [1]. Li *et al.* presented a quantitative model of landslide probability composed of a rainfall intensity-duration threshold model, and they were successful in predicting landslide probability by utilizing geographic information system technology (GIS) [2]. Remondo *et al.* presented a procedure for landslide risk assessment that utilized spatial data to acquire a landslide

The associate editor coordinating the review of this manuscript and approving it for publication was Sajid Ali.

susceptibility model, and quantitative hazard models were obtained by past landslide frequency and magnitude [3]. Aiming at the rapid assessment of landslide risk levels, Tan *et al.* proposed a stacked autoencoder algorithm to handle the problems of current assessment methods, including time-consumption, and this algorithm saves considerable time in landslide risk assessment and shows better performance than artificial neural networks (ANNs) and radial basis function neural networks (RBFs) [4]. In the field of landslide susceptibility index, Goyes-Peñafie and Hernandez-Rojas presented an integration of discrete and continuous data in landslide analysis, and employed logistic regression (LR) and weights of evidence (WoEs), showing highly accurate landslide susceptibility [5]. For landslide sensor data reconstruction, Utomo *et al.* analyzed the situation of abnormal missing landslide data. For example, sensor data failure, external interference, or other environmental factors may be lost, and they predicted the missing data value through a long short-term memory neural network (LSTM), which shows great performance, even in the case of 90% data loss [6]. In displacement prediction, Liu *et al.* explored algorithms for the prediction of landslide displacements, and the results showed that LSTM and gated recurrent units (GRUs) perform with encouraging results [7].

Rainfall has a significant influence on the incidence of landslides. Srivastava *et al.* explored how the back propagation neural network (BPNN), support vector regression (SVR), and LSTM are used to forecast rainfall and predict the occurrence of landslides based on rainfall thresholds [8]. Khaing and Thein used a deep learning system in conjunction with the IoT to successfully forecast rainfall-induced landslides. This system forecasts a univariate time series using the baseline method, the LSTM model [9]. Landslide susceptibility mapping reliability improvement can also be achieved by a machine learning ensemble model. Di Napoli *et al.* selected and assessed 13 predisposing factors and adopted a new approach based on an ANN, a generalized boosting model, and maximum entropy algorithms. The results showed that the proposed algorithm performed well in terms of validation scores [10]. Dou *et al.* also modified the landslide assessment approach using ensemble methods and used a support vector machine (SVM) with a bagging, boosting, and stacking framework [11]. Liang *et al.* performed research to assess and compare four alternative landslide susceptibility mapping models, and the gradient boosting decision tree (GBDT) model performed best in the landslide prediction capability. Combined with cluster analysis, the sampling strategy of non-landslide points can be improved [12]. Ma *et al.* summarized the machine learning methods for landslide prevention. Three main directions: image-based landslide detection, landslide sensitivity assessment, and landslide early warning system development, were the important elements of their investigation. They also discussed the challenges and potential opportunities for machine learning in the area of landslide prevention. They combined a data-driven approach with a knowledge-driven slippery slope mechanism to explain the results of machine learning in landslide prevention research [13]. Research on predicting displacement in the event of a landslide is also progressing rapidly. Liao *et al.* proposed an improved extreme learning machine with grey wolf optimization (GWO). The cumulative landslide displacement was decomposed into trend displacement and periodic displacement. A cubic polynomial model was then used to predict the trend displacement, and after statistical analysis of the displacement data, the model was used to predict the periodic displacement [14]. Ensemble models can also be used in landslide displacement prediction. A new data-driven monitoring and forecasting approach was proposed by Li *et al.*, and they developed an autoregressive motion-averaged time series model to analyze the autocorrelation of landslide triggers. In addition, a parametric correlation model was fitted to the predicted displacements, and the link between the trigger variables and the landslide displacement values was explored [15].

Numerous ANN approaches to predict landslide displacements do not capture the potential nonstationary characteristics of landslide displacements. A discrete wavelet transform (DWT)-extreme learning machine (ELM) model based on chaos theory was proposed by Huang *et al.* to predict landslide displacement. Using the cumulative displacement time series of landslides as the dataset, the displacement of a landslide was estimated precisely [16]. Displacement occurs under the complex conditions of various impact factors, such as geological conditions and precipitation. By examining the response relationships among landslide deformation, rainfall, reservoir water level, and groundwater level, an ELM was proposed and a landslide displacement prediction model related to control factors was established by Cao *et al.* [17]. In 2014, Cheng Lian *et al.* proposed an integrated learning paradigm based on the EEMD-ELM model (extreme learning machine integrated with empirical mode decomposition) to address monitoring data for landslide displacement prediction [18]. In 2015, the aim of the BPNN was to predict slope deformation using daily and antecedent rainfall as input variables, and the model had great performance in accuracy [19].

The above scholars have shown that the application of machine learning and deep learning in landslide data analysis is effective, but they still face three major problems. First, the identification and modeling of landslide risk impact elements is an important technical challenge. Second, the uncertainty of the geographical and temporal prediction of landslide risk has an impact on landslide early warning. Third, the quantitative assessment of landslide risk impact elements still requires in-depth research. The focus in the landslide research field is to find suitable assessment indicators and build suitable models to predict landslides.

Recent breakthroughs in time-series forecasting have influenced the applied field. Data-driven models, such as LSTM, can be used to solve prediction problems in nonlinear time series and possess unique advantages [20]. Landslide processes include time-series data, such as rainfall, surface displacement, shallow moisture content, and deep moisture content. We use an encoder–decoder architecture to connect attention (Attn), temporal convolutional networks (TCNs), and recurrent neural networks (RNNs), and propose TCN-Attn-RNN and RNN-Attn-TCN landslide risk prediction models.

The contributions of this paper are shown as follows: First, we use the TOPSIS-Entropy method to access landslide risk, and the instability margin of landslide risk is obtained, which is more practical to obtain directly from sensor data than the stability and safety factors. Second, the TCN-Attn-RNN and RNN-Attn-TCN are proposed to solve the prediction issue of instability margins. The encoder–decoder architecture connects the TCN and RNN, and the attention mechanism plays the role of regulating the weight of intermediate vectors. Third, this paper integrates the assessment method of landslide instability margins (LIMs) and the landslide risk prediction methods, and solves the modeling problem from the sensor acquisition of landslide hazard information to landslide risk prediction.

Section II introduces the structure of the landslide simulation platform. In section III, the TOPSIS-entropy method and deep learning models are discussed. The TCN-Attn-RNN and RNN-Attn-TCN models for landslide risk prediction are proposed. Section IV demonstrates the effectiveness of the proposed models. Finally, the conclusions and discussion of landslide risk models are presented in section V.

II. LANDSLIDE SIMULATION PLATFORM

The landslide platform is constructed to simulate a real landslide occurrence, as shown in Figure 1.

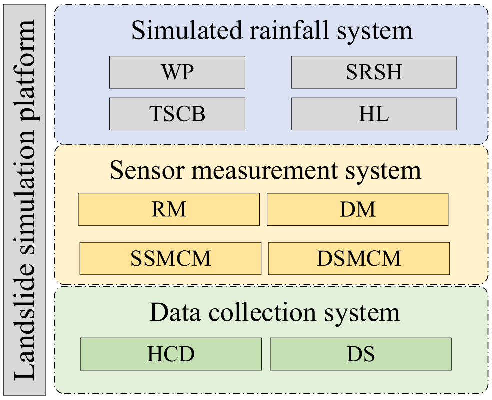
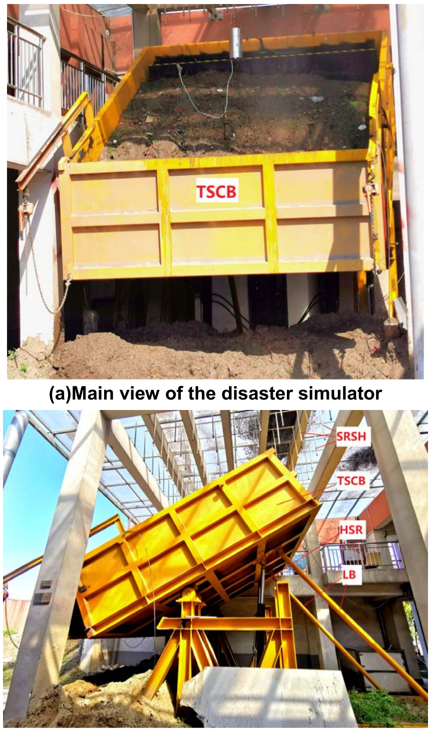

Landslide simulation platform
Simulated rainfall system
WP
SRSH
TSCB
HL
Sensor measurement system
RM
DM
SSMCM
DSMCM
Data collection system
HCD
DS

FIGURE 1. Architecture of the landslide simulation platform, including a simulated rainfall system, sensor measurement system, and a data collection system. The simulated rainfall system comprises the following: a water pump (WP), simulated rainfall sprinkler head (SRSH), test soil-carring box (TSCB), and hydraulic lifting system (HF). Rainfall measurement (RM), displacement measurement (DM), shallow soil moisture content measurement (SSMCM), and deep soil moisture content measurement (DSMCM) are all part of the sensor measurement system (DSCM). The host computer display (HCD) and database storage (DS) are two components of the data collection system.

Throughout the simulation of the landslide, we used sensors to monitor the landslide process factors, including rainfall, shallow soil moisture content, deep soil moisture content, and surface displacement. The magnitude of rainfall is an important factor affecting rainfall-induced landslides, and it also directly affects the magnitude of soil water content and indirectly affects soil stress and surface displacement. The landslide simulation platform uses heavy rainfall to simulate the landslide process, and the experiment lasts for approximately several hours, focusing on the simulation of landslides occurring in the case of 30 degree of slopes during intense rainfall. Every second, experimental data are collected, and approximately tens of thousands of datasets are used for landslide data modeling.



(a)Main view of the disaster simulator

(b)Side view of the disaster simulator

FIGURE 2. Physical diagram of the on-site disaster simulator, which includes the test soil-carrying box (TSCB), simulated rainfall sprinkler head (SRSH), hydraulic support rod (HSR), and lifting bar (LB).

Figure 2 shows a physical diagram of the test soil-carrying box at the disaster test site. The collection sensors mainly include a tipping bucket type rain sensor for measuring rainfall, a tensile displacement sensor for measuring surface displacement, and soil moisture sensors for measuring deep moisture content and shallow moisture content, as shown in Figure 3.

III. METHODOLOGY

In this section, the TOPSIS-Entropy method used to access the landslide risk instability margin is introduced first. Then, canonical sequence deep learning models, such as TCN, LSTM, and GRU models, are formulated. In addition, our proposed models for landslide risk prediction are presented, including TCN-Attn-RNN and RNN-Attn-TCN.

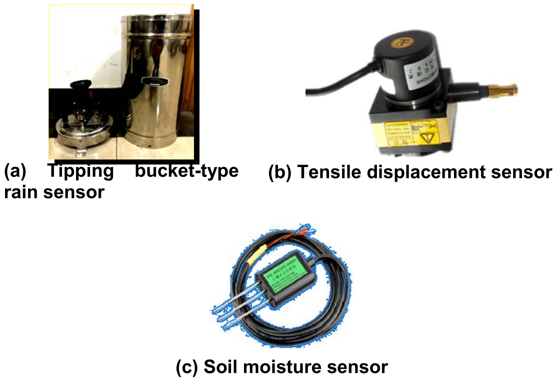

(a) Tipping bucket-type rain sensor
(b) Tensile displacement sensor
(c) Soil moisture sensor

FIGURE 3. Landslide data monitoring sensors.

A. TOPSIS-ENTROPY METHOD

The technique for order preference by similarity to the ideal solution (TOPSIS) was presented in 1981 and is also known as the superior–inferior solution distance method [21]. The TOPSIS-Entropy approach is an evaluation model constructed using the TOPSIS method in which the entropy weight method is used for weight assignment [22]. To obtain the instability margin, the data correlation among landslide influencing factors is determined using the TOPSIS-Entropy method, and the output of the TOPSIS-Entropy approach is utilized as the landslide instability margin, which effectively utilizes the data from the landslide monitoring sensors and makes it easier to calculate the landslide instability coefficient compared to the traditional method.

The landslide instability margins (LIMs) are calculated using (1)-(6), and the calculation process is shown in Figure 3. The first step is to set the four landslide influencing factors as the initialization matrix and then normalize them. Second, the weights are assigned to different landslide factors. The weights are assigned according to the magnitude of information entropy, which is calculated as follows. The very small, intermediate, and interval indicators in the matrix are normalized to very large indicators.

1. Calculate the magnitude of the weight of the jth moment of the ith observation. Here x_{ij} denotes the observation, z_{ij} denotes the normalized data, and p_{ij} denotes the weight size.

$$z_{ij} = \frac{x_{ij} - \min\{x_{1j}, x_{2j}, \ldots x_{nj}\}}{\max\{x_{1j}, x_{2j}, \ldots x_{nj}\} - \min\{x_{1j}, x_{2j}, \ldots x_{nj}\}} \tag{1}$$

$$p_{ij} = z_{ij} / \sum_{i=1}^{n} z_{ij} \tag{2}$$

2. Calculate the entropy value e_j of the jth item.

$$e_j = -(1/\ln n) \cdot \sum_{i=1}^{n} p_{ij} \ln p_{ij} \tag{3}$$

3. Then, calculate the coefficient of variation g_j; the larger the variation is, the smaller the entropy value.

$$g_j = 1 - e_j, \quad 0 < g_j < 1 \tag{4}$$

4. Calculate the weight of the jth term.

$$\omega_j = g_j / \sum_{i=1}^{n} g_i \tag{5}$$

5. Calculate the comprehensive evaluation value S_i of the ith evaluation object.

$$S_i = \sum_{j=1}^{n} \omega_j p_{ij}, \quad i = 1, 2, \cdots, n \tag{6}$$

Finally, the observed values of each time period are multiplied by the corresponding weights to obtain the landslide instability margin, which characterizes the probability of landslide occurrence. The entropy weighting method is an objective weighting method, where the smaller the degree of data variation is (i.e., the smaller the variance is), the less information the data contain and the lower the weight. The landslide instability margin reflects the magnitude of the probability of landslides on a mountain. The TOPSIS-Entropy method scores directly reflect the probability of a landslide.

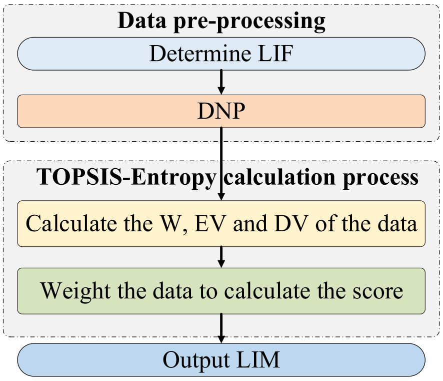

Data pre-processing
Determine LIF
DNP
TOPSIS-Entropy calculation process
Calculate the W, EV and DV of the data
Weight the data to calculate the score
Output LIM

FIGURE 4. Block diagram for solving the landslide instability margin is based on the TOPSIS-Entropy method. Source data first go through the data preprocessing, including landslide impact factors (LIF) detemination and the data normalization process (DNP), and then undergo the TOPSIS calculation process, which comprises the calculation of the weight (W) and the entropy value (EV), as well as the degree of variation (DV) of the data. Finally, the socre and output landslide instability margins (LIMs) are computed, and the data are weighted.

B. DEEP LEARNING MODELS

1) SEQUENCE MODEL

Sequence prediction by RNNs dominates the research on time-series prediction in deep learning due to its ability to capture long-term-dependent memory, which stores important information from the past and is used to forecast sequences at future moments. Framework models of RNNs, such as LSTM [23], GRUs [24] and their derivative algorithms, have shown superior performance in sequence prediction. In recent years, a dual-stage attention-based RNN model (DA-RNN) for nonlinear autoregressive exogenous scenarios has been proposed, and it captures long-term dependencies very well [25]. Canonical convolutional neural networks are generally less suitable for modeling time-series problems because their convolutional kernel size limits their ability to capture long-term-dependent information.

Bai *et al.* successfully used convolutional neural networks in sequence prediction modeling in 2018 [26], proposing an algorithm for temporal convolutional networks (TCNs) using a time-constrained model with causal convolution to capture longer dependencies, and an inflated convolution algorithm that allows for a larger sensory field, and constructing a residual connection, replacing one layer of convolution with a block of residuals. Compared with that of an RNN, the model structure of a TCN is simpler and more effective. Many scholars have extended TCNs to multivariate time-series prediction [27], [28]. Note that while TCNs solve some of the problems associated with gradient disappearance and gradient explosion in RNNs, they fall short in capturing long-term-dependent information compared to the RNN framework and transformer, which can capture temporal information of arbitrary length. Some researchers have combined CNN and RNN structures and proposed the convolutional LSTM (ConvLSTM) algorithm [29].

The attention mechanism is actually the computation of a matrix of automatically weighted coefficients corresponding to the sequence. The query and key-value pairs are mapped to the output, where query, key, and value are vectors and the output is weighted by the values [30]. Many researchers have explored attention methods for time-series forecasting, such as transformer time series [31], and Informer [32]. The attention model can be divided into two categories: hard attention and soft-attention. Hard-attention is a 0-1 problem, where a region is either attended to or not attended to. Soft attention is a continuous distribution problem of [0,1], with different scores ranging from 0 to 1 reflecting the degree of attention of each region. Soft attention is a continuous distribution problem of [0,1], that uses different scores ranging from 0 to 1, to reflect the degree of attention of each region. In essence, the attention mechanism filters out important information from a large amount of information and gathers this vital information while ignoring the unimportant information [33]. The focusing process is reflected in the attention weight coefficient, and the larger the weight coefficient is, the more important the information.

The existence of two effective attention processes demonstrates DA-RNN's superior performance. The attention mechanism is added to the input stage to focus the weights on the fundamental aspects. Before LSTM decoding provides the final output, which is supplied to extract information about the time series characteristics, the temporal attention mechanism is applied to the intermediate hidden layer output. Our research, inspired by the DA-RNN model, proposes combining the parallelism and flexibility of convolution and the timing scalability of RNNs. By combining the features of the TCN structure and RNN structure and adding the attention mechanism, we propose TCN-Attn-RNN and RNN-Attn-TCN, and we use them in landslide risk prediction to construct a novel deep learning landslide risk prediction model. Combining an RNN with a TCN and retaining the input attention mechanism and temporal attention mechanism, namely RNN-Attn-TCN and TCN-Attn-RNN, can effectively combine the advantages of RNNs and TCNs, and more effectively predict the situation of complex sequences. The application of a sequence prediction algorithm in complex landslide time-series data prediction is an innovation of our design.

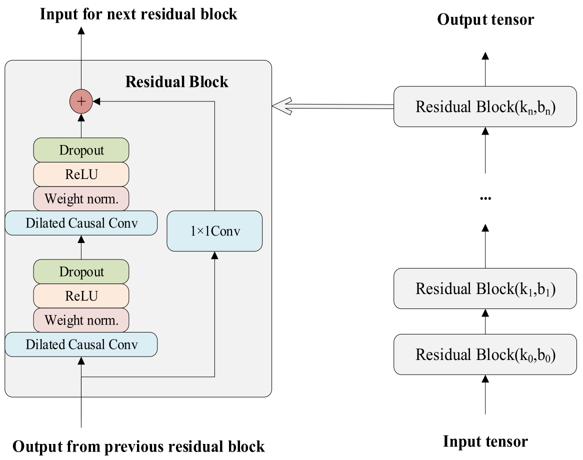

Input for next residual block
Output tensor
Residual Block
+
Dropout
ReLU
Weight norm.
Dilated Causal Conv
1×1Conv
Dropout
ReLU
Weight norm.
Dilated Causal Conv
Residual Block(k$_n$,b$_n$)
...
Residual Block(k$_1$,b$_1$)
Residual Block(k$_0$,b$_0$)
Output from previous residual block
Input tensor

FIGURE 5. Architecture of temporal convolutional neural networks (TCNs).

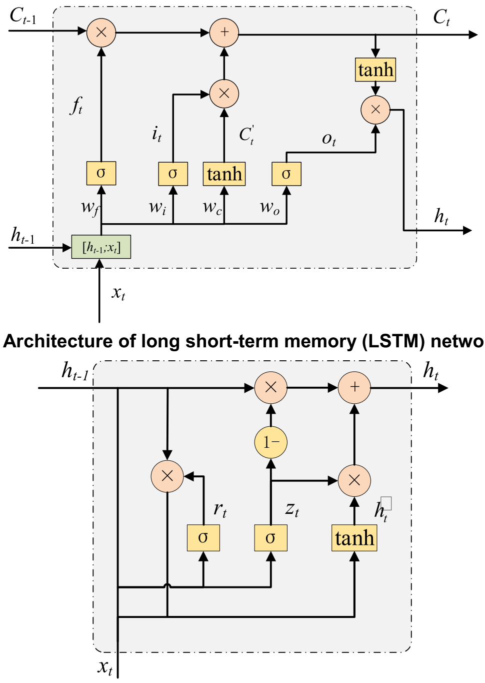
Shaojie Bai *et al.* made improvements to the basic TCN structure, such as residual connectivity and regularization [26]. The residual block is shown in Figure 5. The original one-dimensional causal convolutional layer is replaced by a residual block with 2 layers having the same expansion privacy and residual connectivity. The output of these two convolutional layers is added to the input of the residual block and fed into the next residual block. To adjust the width of the residual tensor, a 1 × 1 convolution is added. At this time, the width of the receptive field of the TCN is twice that of the original basic causal layer. Therefore, the receptive field size r can be obtained by (7).

$$r = 1 + \sum_{i=0}^{n-1} 2(k-1)b^i = 1 + 2(k-1)\frac{b^n - 1}{b - 1} \tag{7}$$

$$n = \left[\log_b\left(\frac{(l-1)(b-1)}{2(k-1)} + 1\right)\right] \tag{8}$$

where k denotes the size of the convolution kernel, b denotes the dilation base, and $k \geq b$. The number of residual blocks n is related to the length l of the input tensor and is calculated as shown in (8). The 1×1 convolution maintains the same length between the input and output for the residual block [34] and dilated causal convolution guarantees that the output will not be influenced by future information [27].

LSTM is a tpye of sequence model with an RNN architecture, and the core feature is the inclusion of gating structures such as forget gates, which enables information to be passed on, as shown in Figure 6. Even knowledge from a previous time period may be remembered at a later point in time,



(a) Architecture of long short-term memory (LSTM) networks

(b) Architecture of gated recurrent units (GRUs)

FIGURE 6. Architecture of LSTM and GRUs.

enabling long-term memory.

$$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f) \tag{9}$$
$$i_t = \sigma(W_i[h_{t-1}, x_t] + b_i) \tag{10}$$
$$C_t' = \tanh(W_c[h_{t-1}, x_t] + b_c) \tag{11}$$
$$C_t = f_t \odot C_{t-1} + i_t \odot C_t' \tag{12}$$
$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o) \tag{13}$$
$$h_t = o_t \odot \tanh(C_t) \tag{14}$$

The gate equation for LSTM is shown in (9)-(11), where the input is the cell state C_{t-1},the hidden state h_{t-1} at the previous moment, the input x_t at moment t, and the output is the cell state C_t at moment t and the hidden state h_t at moment t. The forget gate decides which of the previous moment's information must be retained, the input gate determines which of the current input information must be retained, and the output gate determines the concealed state of the following moment.

The function of the forget gate f_t is to determine what useful information to keep, which is determined by the sigmoid function, a sigmoid funciton close to 0 means more information is discarded, while a sigmoid function close to 1 means more information is kept. The input gate is used to update the state of the hidden layer by passing the state of the hidden layer at the previous moment and the current input information into the sigmoid. Then, the hidden layer information from the previous layer and the current input information are passed through the tanh function and multiplied by the sigmoid output value to output a candidate value, $C_t{}'$. The cell state is updated as shown in (12). The cell state of the previous layer is dot multiplied with the oblivion gate output, and then it is dot multiplied with $C_t{}'$, which updates the cell state C_t.

GRUs are tpyes of RNNs of the RNN architecture, that contains only two elements: the update gate and the reset gate. Their structure is shown in Figure 6. The input is the hidden layer state h_{t-1} at the previous momenta and the input x_t at moment t, and the output is the hidden layer state h_t at moment t.

$$r_t = \sigma(W_r[h_{t-1}, x_t] + b_r) \tag{15}$$
$$z_t = \sigma(W_z[h_{t-1}, x_t] + b_z) \tag{16}$$
$$\tilde{h}_t = \tanh(W[r_t \odot h_{t-1}, x_t] + b) \tag{17}$$
$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \tag{18}$$

where r_t denotes the reset gate at moment t, z_t denotes the update gate at moment t, and h_t denotes the hidden layer state at moment t. tanh and σ denote the activation functions. σ denotes the sigmoid function, which has an output between 0 and 1, and tanh is the hyperbolic tangent function, which has an output between -1 and 1. The reset gate r_t is computed using (15), which determines the magnitude of forgetting the previous moment's information, which updates the input and the previous moment's hidden layer state. The update gate z_t using (16) is similar to merging the forget gate and the input gate in LSTM and determines what information is forgotten and what information is added. Compared with LSTM, GRUs have fewer tensor operations and fewer parameters, and in some cases, they are more precise than LSTM.

Attention mechanisms were first used in machine translation tasks to achieve memory for long sequences of sentence input. The attention mechanism considers the context vector and all the sequence input information to construct the connection. The weights of the connections for each output element are learned automatically. The process of calculating attention can be viewed as querying in key and value pairs. The first step is to calculate the similarity of query Q and key K, which can be solved by finding the vector dot product of the two, by finding the vector cosine similarity of the two, or introducing additional neural networks to find the value, as (19):

$$f(\boldsymbol{Q}, \boldsymbol{K}_t) = \begin{cases} \boldsymbol{Q}^T \boldsymbol{K}, & \textit{dot} \\ \boldsymbol{Q}^T \boldsymbol{W}_a \boldsymbol{K}, & \textit{general} \\ \boldsymbol{W}_a[\boldsymbol{Q}; \boldsymbol{K}_t], & \textit{concat} \\ v_a^T \tanh(\boldsymbol{W}_a \boldsymbol{Q} + \boldsymbol{U}_a \boldsymbol{K}_t), & \textit{perceptron} \end{cases} \tag{19}$$

$$\alpha_t = \textit{softmax}(f(\boldsymbol{Q}, \boldsymbol{K}_i)) = \frac{\exp(f(\boldsymbol{Q}, \boldsymbol{K}_i))}{\sum_j \exp(f(\boldsymbol{Q}, \boldsymbol{K}_i))} \tag{20}$$

$$\boldsymbol{C} = \textit{Attention}(\boldsymbol{Q}, \boldsymbol{K}, \boldsymbol{V}) = \sum_t \alpha_t \boldsymbol{V}_t \tag{21}$$

The second step is to normalize the weights to obtain directly usable weights using (20), and then, in the third step, the weights and values are weighted and summed to obtain the attention value using (21).

2) ATTENTION-BASED TCN CONNECTED WITH AN RNN MODEL

In this study, temporal convolutional networks are connected to a recurrent neural network for a landslide risk prediction model.

As shown in Figure 7, both the TCN-Attn-RNN and the RNN-Attn-RNN models proposed in this paper use the encoder–decoder architecture. The input of the TCN-Attn-RNN model is a tensor composed of landslide time series data that is passed through the TCN through the residual block framewor. Then, a weight vector is generated through the attention mechanism and multiplied by the output of the TCN, followed by the decoder operation, where the decoder uses LSTM or a GRU of the RNN architecture.

The RNN-Attn-TCN model is similar. The difference is that the input time-series data tensors after the input attention mechanism are decoded by the RNN. Here, the RNN generalizes the RNN architecture. It can generally use LSTM or GRU. To more fully use the sequence data, the BiLSTM or BiGRU algorithm can be used. The decoder uses the TCN model. The specific steps for RNN-Attn-TCN are the same as those for TCN-Attn-RNN, so they will not be repeated.

IV. EXPERIMENT AND RESULTS

In Figure 8, the training process of the landslide risk prediction model uses the sliding window, and the sliding window is the landslide sensor data input to TCN-Attn-RNN or RNN-Attn-TCN. The inputs are rainfall, surface displacement, shallow moisture content and deep moisture content, and each moment links to a landslide instability margin. The purpose of the landslide risk prediction is to predict the landslide instability margin at a future moment. Therefore, the output of the prediction model is the landslide instability margin at the next time.

The collection process of the landslide dataset can be realized through the landslide simulation platform introduced in section II, and the ratio of the training dataset to the test set dataset is 16:3. The training process uses a supervised learning approach to train the model parameters of the TCN, attention mechanism and RNN with 200 iterations. Figure 9 shows the four measured data points of the landslide sensor and landslide instability margin. In the process of increasing rainfall, the shallow moisture content and deep moisture content also continue to increase, but the surface displacement does not change, at this time, the landslide instability margin begins to increase. When the surface displacement moves, the landslide instability margin changes greatly. After that, the soil moisture content changes very little, while the ground displacement changes drastically, and the landslide instability margin changes accordingly.

For the above data, the input dimension at each moment is 4, and the input and output time lengths are set to 100-10, 100-50, and 1000-100. The codes run on a i5-10500F CPU with a Windows 10 operating system, an NVIDIA GeForce GTX 1650 GPU, and 16 GB of memory, and the deep learning

Algorithm 1 : TCN-Attn-RNN

Initialization:

(1) Obtain the landslide source data, and then normalize the data.

(2) Obtain the LIMs using the TOPSIS-Entropy method, shown in Figure 4.

(3) Build the TCN and RNN, and then use the landslide training datasets to train the nets. Note that the input vectors are four impact factors: rainfall, surface displacement, shallow moisture content, and deep moisture content, and the output is the LIM.

Input: $\boldsymbol{X_t} = \{\boldsymbol{x_t^1}, \boldsymbol{x_t^2}, \ldots, \boldsymbol{x_t^n}\}; \boldsymbol{H}_{t-1}; \boldsymbol{Y}_{t-1}$

Step 1. Input attention mechanism: In the soft-attention mechanism, query, keys, and values are equal to the input tensor $\boldsymbol{X_t}$. The query $\boldsymbol{Q}$, keys $\boldsymbol{K}$, and values $\boldsymbol{V}$ can be obtained by (22)-(24), and the attention coefficient can be obtained by (25). Finally, output $\tilde{\boldsymbol{X}}_t$ using (26).

$$\boldsymbol{Q}_{input} = \boldsymbol{W}^{q_i}\boldsymbol{X}_t \tag{22}$$

$$\boldsymbol{K}_{input} = \boldsymbol{W}^{k_i}\boldsymbol{X}_t \tag{23}$$

$$\boldsymbol{V}_{input} = \boldsymbol{W}^{v_i}\boldsymbol{X}_t \tag{24}$$

$$\boldsymbol{I}_t = \frac{\exp(\boldsymbol{Q}_{input}^T\boldsymbol{K}_{input})}{\sum_j \exp(\boldsymbol{Q}_{input}^T\boldsymbol{K}_{input})} \tag{25}$$

$$\tilde{\boldsymbol{X}}_t = \sum_t \boldsymbol{I}_t\boldsymbol{V}_t \tag{26}$$

Step 2. Encoder processing: For the input attention output $\tilde{\boldsymbol{X}}_t$ and the hidden state of the TCN model at the $t-1$ moment $\boldsymbol{H}_{t-1}$ to be encoded, the encoding function adopts the TCN model, and the generated hidden state is $\boldsymbol{H_t} = \{h_t^1, h_t^2, \ldots, h_t^n\}$, which can be expressed as shown in (27).

$$\boldsymbol{H}_t = \boldsymbol{f}_{TCN}(\tilde{\boldsymbol{X}}_t, \boldsymbol{H}_{t-1}) \tag{27}$$

Step 3. Temporal attention mechanism: Our model uses a soft attention mechanism to generate weights $\boldsymbol{A_t} = \{a_t^1, a_t^2, \ldots .a_t^n\}$, which are calculated as shown in (28)-(31).

$$\boldsymbol{Q}_{temporal} = \boldsymbol{W}^{q_t}\boldsymbol{H}_t \tag{28}$$

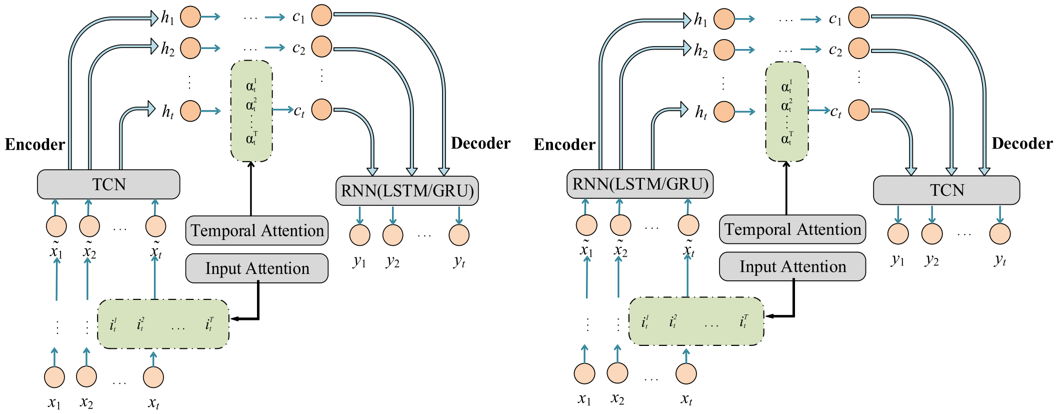
$$\boldsymbol{K}_{temporal} = \boldsymbol{W}^{k_t}\boldsymbol{H}_t \tag{29}$$

$$\boldsymbol{V}_{temporal} = \boldsymbol{W}^{v_t}\boldsymbol{H}_t \tag{30}$$

$$\boldsymbol{A}_t = \frac{\exp(\boldsymbol{Q}_{temporal}^T\boldsymbol{K}_{temporal})}{\sum_j \exp(\boldsymbol{Q}_{temporal}^T\boldsymbol{K}_{temporal})} \tag{31}$$

$$\boldsymbol{C}_t = \sum_t \boldsymbol{A}_t\boldsymbol{V}_{temporal} \tag{32}$$

where $\boldsymbol{A}_t$ is a weight vector, and the function $\boldsymbol{f}$ can take different functional forms, the details of which are available in (19). Here, the function $\boldsymbol{f}$ uses the dot product. Finally, the weighted intermediate vector $\boldsymbol{C}$ is generated by weighting and summing the $\boldsymbol{V}_{temporal}$ using (32).



(a) Architecture of TCN-Attn-RNN

(b) Architecture of RNN-Attn-TCN

FIGURE 7. Architecture of the attention-based temporal convolutional network connected to the recurrent neural network.

(Continued.) TCN-Attn-RNN

$$Y_t = f_{RNN}(C_t, Y_{t-1}) \tag{33}$$

Step 4. Decoder processing: For the intermediate vector C, the decoding function adopts the LSTM or GRU model, and the output is Y_t, which is the predicted sequence information.

Output: $Y_t; H_t$.

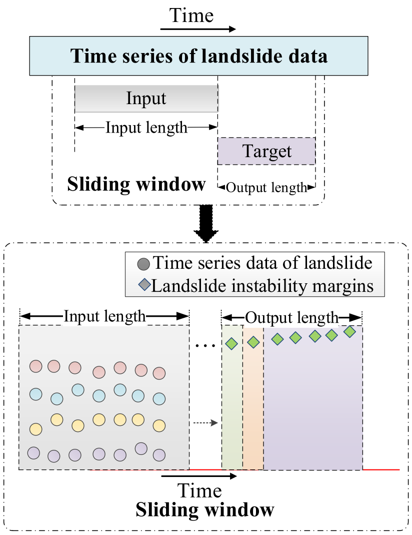


FIGURE 8. Architecture of the landslide data series prediction process.

framework includes Python 3.6, Keras 2.0.2, TensorFlow 1.9.0.

We verify the performance of the model by comparing the average absolute error (MAE), the root mean square error (RMSE) and the mean absolute percentage error (MAPE) of the model output [35]. The three metrics are calculated as follows (34)-(36), where y_i represents the ith ground truth data and $\widehat{y_i}$ represents the ith model-predicted value data.

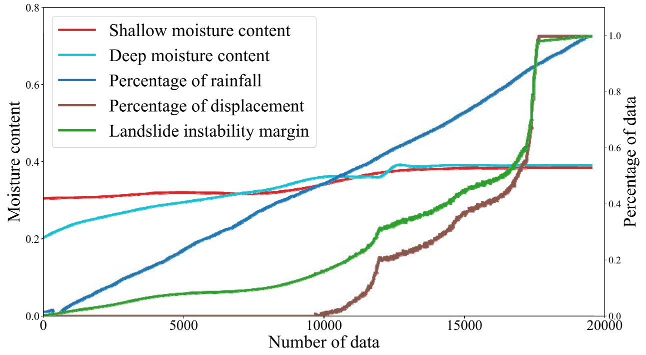


FIGURE 9. Architecture of the landslide data series prediction process.

To quantitatively show the performance improvement of our model, we use the metrics of the LSTM model as the benchmark to calculate the metric reduction percentage (MRP) of other deep learning time series prediction models. The calculation formula is shown in (37).

$$MAE(\widehat{Y}, Y) = \frac{1}{N}\sum_{i=1}^{N}|\widehat{y_i} - y_i| \tag{34}$$

$$RMSE(\widehat{Y}, Y) = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\widehat{y_i} - y_i)^2} \tag{35}$$

$$MAPE(\widehat{Y}, Y) = \frac{100\%}{N}\sum_{i=1}^{N}\left|\frac{\widehat{y_i} - y_i}{y_i}\right| \tag{36}$$

$$MRP(M_L, M_D) = (M_L - M_D) \times 100\%/M_L \tag{37}$$

Here, M_L signifies the LSTM function's metrics, and M_D denotes the metrics of other deep learning models in (37). The metric decrease percentage (MRP) is obtained using (37).

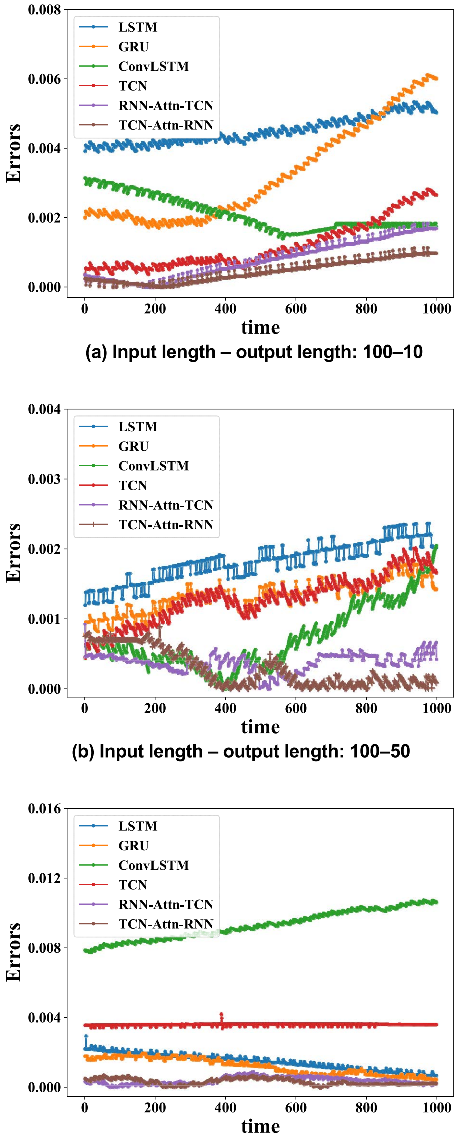
The prediction model performs better when the metric is smaller. As a result, the greater the MRP with LSTM, as the benchmark model, is, the better the model's performance, indicating that the performance of the model is improved more than that of its counterparts. A negative MRP indicates a decline in performance. The metric data in Table 1 are obtained by testing various deep learning time-series prediction models with varying inputs and prediction lengths, i.e., the statistics of RMSE, MAE, and MAPE. Table 2 is obtained after the MRP results are counted, and the different input and prediction lengths of several deep learning models are tested.

TABLE 1. Study of different prediction lengths for deep learning models.

Prediction Length		10	50	100
Input Length		100	100	1000
LSTM	RMSE	0.0857	0.0232	0.0045
	MAE	0.0637	0.0107	0.0026
	MAPE	2.7484	2.9306	8.6838
GRU	RMSE	0.0228	0.0217	0.0036
	MAE	0.0116	0.0091	0.0017
	MAPE	2.0968	2.9957	8.4160
TCN	RMSE	0.0572	0.0205	0.0043
	MAE	0.0107	0.0079	0.0033
	MAPE	2.0163	3.1493	9.5052
ConvLSTM	RMSE	0.0276	0.0203	0.0060
	MAE	0.0094	0.0079	0.0039
	MAPE	2.0000	3.0778	8.3548
RNN-Attn-TCN	RMSE	0.0207	0.0208	**0.0014**
	MAE	**0.0078**	0.0083	0.0023
	MAPE	1.9859	**1.8016**	8.3330
TCN-Attn-RNN	RMSE	**0.0203**	**0.0174**	0.0018
	MAE	0.0087	**0.0077**	**0.0016**
	MAPE	**1.9104**	1.9144	**8.2395**

TABLE 2. Study of the metric reduction percentage for different deep learning models.

Models	Metric	MRP / %		
		Input length - prediction length		
		100-10	100-50	1000-100
LSTM	RMSE	0	0	0
	MAE	0	0	0
	MAPE	0	0	0
GRU	RMSE	73.40	6.47	20.00
	MAE	81.79	14.95	34.62
	MAPE	23.71	-2.22	3.08
TCN	RMSE	33.26	11.64	4.44
	MAE	83.20	26.17	-26.92
	MAPE	26.64	-7.46	-9.46
ConvLSTM	RMSE	67.79	12.50	-33.33
	MAE	85.24	26.17	-50.00
	MAPE	27.23	-5.02	3.79
RNN-Attn-TCN	RMSE	75.85	10.34	**68.89**
	MAE	**87.76**	22.43	11.54
	MAPE	27.74	**38.52**	4.04
TCN-Attn-RNN	RMSE	**76.31**	**25.00**	60.00.
	MAE	86.34	**28.04**	**38.46**
	MAPE	**30.49**	34.68	**5.12**



(a) Input length – output length: 100–10

(b) Input length – output length: 100–50

(c) Input length – output length: 1000–100

FIGURE 10. Error comparison of different landslide risk models.

We selected typical deep learning models for landslide risk prediction. Because of their outstanding and typical prediction outcomes in deep learning time-series prediction, the LSTM, GRU, TCN, and convLSTM models are selected for comparison. In Table 1 and Table 2, when the "input length–prediction length" is set to 100–10, the RMSE and MAPE metrics of TCN-Attn-RNN are lower, decreasing by 76.31% and 30.49%, respectively, compared with those of LSTM. At this time, the MAE metric of RNN-Attn-TCN is reduced by 87.76%. The RMSE and MAE metrics of TCN-Attn-RNN are 25.00% and 28.04% lower, respectively, than those of the LSTM model when the "input length-prediction length" is set to 100–50. The MAPE of RNN-Attn-TCN, on the other hand, is 1.8016, which is 38.52% lower than the benchmark. The MAE and MAPE are lower for TCN-Attn-RNN with MRP values of 38.46% and 5.12 %, respectively, for a sliding window of 1000–100 for long-term prediction. For RNN-Attn-TCN, its RMSE metric is lowered by 68.89%. When comparing models with the same sliding window length, our model outperforms the others for three metrics. While comparing different input and output lengths, the longer the prediction is, the worse the prediction. In addition, compared to that of other models, the performance improvement is not as excellent as the short time-series prediction.

Figure 10 depicts a comparison of the prediction errors of several models for three different sliding window length instances. Our proposed models have fewer errors and are better suited for landslide time-series prediction, as evidenced by a comparison of the errors in the three sliding window situations. Meanwhile, the error volatility is greater for 100-50 because the predicted length to input length ratio is higher, which limits the model's prediction effect to some extent. For 100-10 and 1000-100, the error volatility is substantially lower. Notably, RNN-Attn-TCN will have a marginally worse prediction effect than TCN-Attn-RNN, as seen in Table 2 and Figure 10.

V. CONCLUSION AND DISCUSSION

In this work, we concentrate on the landslide risk prediction issue, namely landslide risk assessment and modeling. First, the LIMs are calculated by the TOPSIS-Entropy method, which inputs four landslide impact factors and outputs the landslide instability margin. Second, through TCN-Attn-RNN and RNN-Attn-TCN for landslide risk, the output is the future time series for the LIM. To assess the models' efficacy, a comparative analysis of a vanilla LSTM, a GRU, a TCN, ConvLSTM, and our models is conducted. Based on the evaluation of the MAE, RMSE, and MAPE, our models outperform their counterparts in terms of the metric reduction percentage (MRP). Through the high accuracy of the models, the landslide risk can be predicted by deep learning methods based on large-scale landslide data.

For the landslide risk assessment issue, we employ the TOPSIS-Entropy approach for comprehensive assessment of four landslide impact factors for landslide risk assessment. This method can immediately acquire the final LIM used for assessment from sensor data, making it very intuitive and convenient. This multisensor integrated assessment of landslides is more extensive than the single-factor assessment technique. To tackle the landslide risk prediction problem, deep learning for temporal prediction is introduced. The architecture composed of an attention-based TCN paired with an RNN optimizes the model structure of deep learning and has a better ability to handle challenging circumstances such as landslide data modeling.

Note that our models cannot operate with small amounts of data because they require a substantial amount of data for training. There are also issues with integrating the model into an actual landslide prediction site, which is something we should explore in the future. We plan to apply our methods to real-world scenarios and develop a software system to efficiently run our model as required to meet online prediction requirements. In addition, we will expand our studies to include new types of landslide sensor data, such as soil pressure and underground displacement, as more landslide impact factors equal more comprehensive landslide hazard information.

ACKNOWLEDGMENT

The authors are deeply grateful to all the graduate students of the National and Local Joint Engineering Laboratory of Disaster Monitoring Technology and Instruments for their help in the geological disaster experiment.

REFERENCES

[1] A. Joshi, J. Grover, D. P. Kanungo, and R. K. Panigrahi, "Edge assisted reliable landslide early warning system," in *Proc. IEEE 16th India Council Int. Conf. (INDICON)*, Dec. 2019, pp. 1–4.

[2] M. Li, M. Zhu, Y. Hec, Z. He, N. Wang, Z. Zheng, and G. Zhou, "Warning of rainfall-induced landslide in bazhou district," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Sep. 2020, pp. 6879–6882.

[3] J. Remondo, J. Bonachea, and A. Cendrero, "A statistical approach to landslide risk modelling at basin scale: From landslide susceptibility to quantitative risk assessment," *Landslides*, vol. 2, no. 4, pp. 321–328, Nov. 2005.

[4] F. Tan, J. Yu, Y.-Y. Jiao, D. Lin, J. Lv, and Y. Cheng, "Rapid assessment of landslide risk level based on deep learning," *Arabian J. Geosci.*, vol. 14, no. 3, pp. 1–10, Feb. 2021.

[5] P. Goyes-Peñafiel and A. Hernandez-Rojas, "Landslide susceptibility index based on the integration of logistic regression and weights of evidence: A case study in Popayan, Colombia," *Eng. Geol.*, vol. 280, Jan. 2021, Art. no. 105958.

[6] D. Utomo, L.-C. Hu, and P.-A. Hsiung, "Deep neural network-based data reconstruction for landslide detection," in *Proc. IEEE Int. Geosci. Remote Sens. Symp. (IGARSS)*, Sep. 2020, pp. 3119–3122.

[7] Z.-Q. Liu, D. Guo, S. Lacasse, J.-H. Li, B.-B. Yang, and J.-C. Choi, "Algorithms for intelligent prediction of landslide displacements," *J. Zhejiang Univ. Sci. A*, vol. 21, no. 6, pp. 412–429, Jun. 2020.

[8] S. Srivastava, N. Anand, S. Sharma, S. Dhar, and L. K. Sinha, "Monthly rainfall prediction using various machine learning algorithms for early warning of landslide occurrence," in *Proc. Int. Conf. for Emerg. Technol. (INCET)*, Jun. 2020, pp. 1–7.

[9] C. C. Khaing and T. L. L. Thein, "Prediction of rainfall based on deep learning and Internet of Things to prevent landslide," in *Proc. IEEE 9th Global Conf. Consum. Electron. (GCCE)*, Oct. 2020, pp. 190–191.

[10] M. Di Napoli, F. Carotenuto, A. Cevasco, P. Confuorto, D. Di Martire, M. Firpo, G. Pepe, E. Raso, and D. Calcaterra, "Machine learning ensemble modelling as a tool to improve landslide susceptibility mapping reliability," *Landslides*, vol. 17, no. 8, pp. 1897–1914, Apr. 2020.

[11] J. Dou, A. P. Yunus, D. T. Bui, A. Merghadi, M. Sahana, Z. Zhu, C.-W. Chen, Z. Han, and B. T. Pham, "Improved landslide assessment using support vector machine with bagging, boosting, and stacking ensemble machine learning framework in a mountainous watershed, Japan," *Landslides*, vol. 17, no. 3, pp. 641–658, Mar. 2020.

[12] Z. Liang, C. Wang, and K. U. J. Khan, "Application and comparison of different ensemble learning machines combining with a novel sampling strategy for shallow landslide susceptibility mapping," *Stochastic Environ. Res. Risk Assessment*, vol. 35, no. 6, pp. 1243–1256, Jun. 2021.

[13] Z. Ma, G. Mei, and F. Piccialli, "Machine learning for landslides prevention: A survey," *Neural Comput. Appl.*, pp. 10881–10907, Nov. 2020.

[14] K. Liao, Y. Wu, F. Miao, L. Li, and Y. Xue, "Using a kernel extreme learning machine with grey wolf optimization to predict the displacement of step-like landslide," *Bull. Eng. Geol. Environ.*, vol. 79, no. 2, pp. 673–685, Mar. 2020.

[15] H. Li, Q. Xu, Y. He, and J. Deng, "Prediction of landslide displacement with an ensemble-based extreme learning machine and copula models," *Landslides*, vol. 15, no. 10, pp. 2047–2059, May 2018.

[16] F. Huang, K. Yin, G. Zhang, L. Gui, B. Yang, and L. Liu, "Landslide displacement prediction using discrete wavelet transform and extreme learning machine based on chaos theory," *Environ. Earth Sci.*, vol. 75, no. 20, pp. 1–18, Oct. 2016.

[17] Y. Cao, K. Yin, D. E. Alexander, and C. Zhou, "Using an extreme learning machine to predict the displacement of step-like landslides in relation to controlling factors," *Landslides*, vol. 13, no. 4, pp. 725–736, Jun. 2015.

[18] C. Lian, Z. Zeng, W. Yao, and H. Tang, "Extreme learning machine for the displacement prediction of landslide under rainfall and reservoir level," *Stochastic Environ. Res. Risk Assessment*, vol. 28, no. 8, pp. 1957–1972, Dec. 2014.

[19] P. Chaturvedi, S. Srivastava, and N. Tyagi, "Prediction of landslide deformation using back-propagation neural network," in *Proc. IEEE Workshop Comput. Intell., Theories, Appl. Future Directions (WCI)*, Dec. 2015, pp. 1–5.

[20] Q. Miao, B. Pan, H. Wang, K. Hsu, and S. Sorooshian, "Improving monsoon precipitation prediction using combined convolutional and long short term memory neural network," *Water*, vol. 11, no. 5, p. 977, May 2019.

[21] *TOPSIS (Technique for Order Preference by Similarity to Ideal Solution)—A Multiple Attribute Decision Making, W: Multiple Attribute Decision Making-Methods and Applications, a State-of-the-at Survey*. Berlin, Germany: Springer-Verlag, 1981, pp. 120–140.

[22] P. Chen, "Effects of the entropy weight on TOPSIS," *Expert Syst. Appl.*, vol. 168, Apr. 2021, Art. no. 114186.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997

[24] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*.

[25] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. W. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," in *Proc. 26th Int. Joint Conf. Artif. Intell.*, Aug. 2017.

[26] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," 2018, *arXiv:1803.01271*.

[27] R. Wan, S. Mei, J. Wang, M. Liu, and F. Yang, "Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting," *Electronics*, vol. 8, no. 8, p. 876, Aug. 2019.

[28] Y. Xu, C. Hu, Q. Wu, Z. Li, S. Jian, and Y. Chen, "Application of temporal convolutional network for flood forecasting," *Hydrol. Res.*, vol. 52, no. 6, pp. 1455–1468, Dec. 2021.

[29] S. Xingjian, Z. Chen, H. Wang, D.-Y. Yeung, W.-K. Wong, and W.-C. Woo, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Proc. Adv. Neural Inf. Process. Syst.*, 2015, pp. 802–810.

[30] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.

[31] S. Li, X. Jin, Y. Xuan, X. Zhou, W. Chen, Y.-X. Wang, and X. Yan, "Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 2019, vol. 32, pp. 5243–5253.

[32] H. Zhou, S. Zhang, J. Peng, S. Zhang, J. Li, H. Xiong, and W. Zhang, "Informer: Beyond efficient transformer for long sequence time-series forecasting," in *Proc. AAAI*, 2021, pp. 1–15.

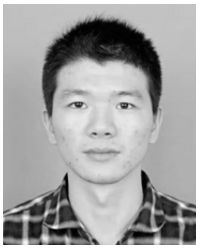
[33] N. Kitaev, L. Kaiser, and A. Levskaya, "Reformer: The efficient transformer," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–12.

[34] J. Long, E. Shelhamer, and T. Darrell, "Fully convolutional networks for semantic segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2015, pp. 3431–3440.

[35] W. Zheng and J. Hu, "Multivariate time series prediction based on temporal change information learning method," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Jan. 4, 2022, doi: 10.1109/TNNLS.2021.3137178.

DI ZHANG is currently pursuing the M.S. degree with the National and Local Joint Engineering Laboratory of Disaster Monitoring Technology and Instruments, China Jiliang University. His research interests include machine learning and intelligent methods for landslide disaster prediction, especially the application of deep neural networks in the prediction of landslide time-series data.

JIACHENG YANG received the B.S. degree from NingboTech University, in 2020. He is currently pursuing the M.S. degree with the National and Local Joint Engineering Laboratory of Disaster Monitoring Technology and Instruments, China Jiliang University. His main research interest includes landslide forecasting.



FENGLIN LI received the B.S. degree from Anhui Polytechnic University, in 2019. He is currently pursuing the M.S. degree with the National and Local Joint Engineering Laboratory of Disaster Monitoring Technology and Instruments, China Jiliang University. His main research interest includes disaster monitoring.

SHUAI HAN is currently pursuing the M.S. degree with the National and Local Joint Engineering Laboratory of Disaster Monitoring Technology and Instruments, China Jiliang University. His main research interests include seepage velocity and direction detection of tailings ponds and landslides.

LEI QIN received the B.S. degree from Weifang University, in 2019. He is currently pursuing the M.S. degree with the National and Local Joint Engineering Laboratory of Disaster Monitoring Technology and Instruments, China Jiliang University. His main research interest includes electronic circuit design for landslides.

QING LI (Member, IEEE) is currently a Professor with China Jiliang University, where he was the Dean of the School of Mechanical and Electrical Engineering, in 2017. He was selected as a Famous Teacher of the National Ten-Thousand Talents Program. He is the Executive Director of the China's Metrological Testing Association. His research interests include dynamic measurement and control and sensing technology of landslide disasters. He has completed more than 60 science and technology projects.