IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received 12 November 2023, accepted 4 December 2023, date of publication 7 December 2023,
date of current version 14 December 2023.

Digital Object Identifier 10.1109/ACCESS.2023.3340425

RESEARCH ARTICLE

Enhancing Concrete Creep Prediction With Deep Learning: A Soft-Sorted One-Dimensional CNN Approach

CONGHUI LI[1], MING ZHANG[1], AND XIUZHI ZHANG[2]
[1]Zibo Vocational Institute, Zibo 255314, China
[2]School of Materials Science and Engineering, University of Jinan, Jinan 250022, China

Corresponding author: Conghui Li (12601@zbvc.edu.cn)

This work was supported in part by Yihui New Materials Company Ltd.

ABSTRACT Creep is a long-term property of concrete that poses a significant challenge to structural engineering. The research content of this paper concerns how to accurately predict the creep behavior of concrete. The overall purpose is to develop a robust predictive model based on the Northwestern University (NU) database that overcomes the challenges encountered when handling tabular creep data, including the lack of spatial correlations between features and the presence of noise. To achieve this goal, a novel deep learning methodology that leverages convolutional neural networks (CNNs) is introduced. First, the data preprocessing phase involves feature selection and K-nearest neighbor-based data filling to enhance the quality of the given data. A soft-sorted 1D CNN is introduced to transform tabular data into multichannel images, thereby taking advantage of spatial locality and feature sorting. Additionally, this model incorporates noise reduction mechanisms through residual network encoders, achieving enhanced data clarity and model accuracy. The mean absolute error (MAE) and root mean square deviation (RMSE) of the model increase from 7.48 and 10.76 to 5.83 and 8.88, respectively. The major findings of this research indicate a significant predictive accuracy improvement over the existing methods. The model effectively captures the complex relationships between concrete creep and a range of influencing factors, enabling precise and quantitative predictions to be obtained. These findings offer deep insights for the field of civil engineering, enabling enhanced structural design, safety, and performance prediction in concrete applications. This paper contributes to the ongoing exploration of advanced deep learning techniques for addressing complex challenges in the domains of concrete mechanics and structural engineering.

INDEX TERMS Concrete creep prediction, deep learning, soft-sorted CNN, residual network, noise reduction.

I. INTRODUCTION

The creep behavior of concrete has a critically important impact on structural design and should be thoroughly considered [1], [2]. Therefore, understanding and accurately predicting concrete creep is significant for evaluating the short- and long-term performance of concrete structures subjected to continuous loads. To address this challenge, a denoising residual neural network (DRNN) is proposed; it has exhibits excellent performance in cross-validation experiments, and its root mean square error (RMSE) and mean absolute error (MAE) are 8.88 and 5.83, respectively. Furthermore, ablation experiments are conducted to illustrate the effectiveness of the residual encoder with regard to denoising. In this study, the Northwestern University database [3] is used to train and evaluate the proposed model. The database contains useful

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

information, such as water-cement ratios, admixtures, cement type classifications, temperatures and volume-to-surface ratios.

A. SIGNIFICANCE OF THE STUDY

The creep behavior of concrete has an important effect on the long-term deformation and stability of structural members. With the passage of time, the degree of creep strain in concrete increases, which may lead to continuous settlement, deformation, and even cracking in the structure [4], [5]. In a long-term loading case, the creep behavior of concrete may also affect the overall stability of the examined structure [6].

The rate of creep is usually high in the early stages of loading and decreases with time, but it does not increase again after that [7]. While some recovery occurs when the sustained load is removed, the concrete usually does not return to its original state, and its elastic recovery and creep recovery effects are less than the deformation induced under a load [8], [9].

Additionally, creep can result in the shrinkage of concrete columns and piers, and if the shrinkage values are unequal, cracking may occur in concrete columns [10], [11], [12]. This situation can lead to the unexpected displacement of the plate under the support column, introducing additional stresses not considered during the design process [10]. Furthermore, creep in prestressed concrete can cause concrete shrinkage, reduce the tensile forces in the reinforcement mechanism, leading to a prestress loss, and ultimately decrease the ability of the structure to counterbalance external loads [13], [14], [15]. Over time, this can result in excessive deflection. Moreover, creep interacts with other factors, such as shrinkage, temperature, and humidity, thereby influencing the deflection of prestressed concrete structures [16].

B. RELATED WORK

In a dry environment, the relative humidity distribution within concrete pores at different time intervals has a significant impact on the creep behavior of a concrete structure. In this context, Parrott [17] proposed a nonlinear diffusion equation based on finite-difference and finite-element solutions that can make fundamental predictions about creep behaviors in dry concrete. However, in both short-term and long-term prediction scenarios, this equation fails to exhibit asymptotic behavior and does not align well with more extensive experimental data. Furthermore, Baant and Baweja [18] established more accurate prediction models based on the approximate proportionality between the level of moisture loss and the decrease in average relative humidity in pores, as well that as between drying shrinkage and moisture loss. Nevertheless, these models, which rely on finite parameters and empirical analysis, cannot accurately reflect creep relationships.

The accuracy of creep prediction relies on short-term concrete measurement data. Without such data, long-term creep predictions cannot be made. Therefore, based on linear regression models using historical data, a compliance function [19] J(t, t%) was introduced for the B3 model. B3 is an effective equation, and its theory is founded on the principles of solidification theory and microstress theory, with the notable advantage of encompassing all free parameters for creep deformation with elastic strain [20]. These parameters can be linearly predicted using the least-squares method to minimize prediction biases. However, this model is highly inaccurate in terms of predicting concrete lifetimes that are expected to span decades. A significant drawback is the lack of laboratory data for modern concrete, which would enable model calibration. To address this, Northwestern University compiled a new database, categorizing creep types and collecting data on various parameters, such as environmental temperature, humidity, fly ash, and admixtures. Building upon this comprehensive database, the B4 model [21] was introduced, offering a model based on concrete components and environmental conditions. In comparison with the B3 model, B4 also accounts for the influences of various types of aggregates, cement and admixtures [21]. Nevertheless, these empirical statistical models do not consider the impacts of nonlinear parameters on creep and cannot comprehensively account for the various complex scenarios encountered in real engineering projects.

With the increasing popularity of machine learning, this technology has been widely used in research on building material performance [22], [23]. Researchers have attempted to utilize statistical methods to investigate the numerical relationships between creep and other variables. Liang et al. [24] attempted to predict and analyze creep through ensemble machine learning based on a tree structure and Bayesian optimization parameters, and they statistically measured the importance levels of different variable factors. However, although a tree-based model is suitable from the perspective of model complexity, it is often easy to overfit such a model because it cannot smoothly segment a high-dimensional feature space, and it heavily relies on the quality of the input data. In addition, many researchers [25], [26], [27] have used simple single-layer artificial neural networks (ANNs) to predict creep in masonry. However, due to limitations regarding the number of parameters and the inability of simple single-layer networks to effectively express nonlinear features, the predictive performance and generalization abilities of these models are restricted. Nevertheless, as databases have improved, the ability of deep learning to study complex features has become increasingly evident, resulting in significant breakthroughs in creep prediction.

Abed et al. [28] considered the temporal dependencies of time series within their architecture and developed a time delay-focused neural network model founded on theory. Garoosiha et al. [29] used a Bayesian framework to optimize the parameters of artificial neural networks. El-Shafie and Aminah [30] introduced a nonlinear autoregressive exogenous input (NARX) creep prediction model, which

achieved improved prediction accuracy. However, these studies suffered from limited data, limited parameters, narrow model applicability, and model simplicity.

Furthermore, a hybrid model combining multigenetic genetic programming (MGGP) and an ANN [31] was used to predict concrete creep compliance. This model employed a large number of samples and parameters from the NU database for model fitting and performance evaluation purposes. Additionally, Zhu and Wang [32] pioneered the integration of convolutional neural networks (CNNs) into creep prediction. CNNs have achieved remarkable success in pattern recognition due to their ability to efficiently learn hidden, abstract features in data. The accuracy, generalization and stability of creep prediction can be improved by using an adaptive one-dimensional CNN to capture complex nonlinear data relationships. Such models accurately predict various creep characteristics. However, as tabular data lack spatial correlations, the direct introduction of one-dimensional convolutional networks may not capture valuable spatial information. Additionally, Hubler et al. [3] noted that approximately 800 of the 1400 creep curves in their NU database consider admixtures and suggested that admixtures could have a significant effect on creep. However, due to the complexity of admixtures in creep prediction, in most models, the effect of admixtures is excluded from the parameter selection process. Last, data collection errors can impact accuracy, as the utilized model may learn incorrect information. Unfortunately, none of the aforementioned methods effectively handle data noise. This research aims to propose a creep prediction method that considers the influence of admixtures and can process data noise; our results are verified using the NU dataset. In addition, reinforced concrete under a long-term continuous load is randomly selected for prediction, and each important module in the proposed model is compared to evaluate its effectiveness.

II. METHODOLOGY

The proposed method aims to enhance the accuracy of predicting and analyzing the creep behaviors of concrete elements by studying the technical patterns and environmental conditions of concrete using a CNN. Therefore, to address this challenge, the DRNN model is proposed. The DRNN model is primarily composed of four components: a soft sorting layer, a noise reduction encoding module, a CNN layer, and a prediction module. The model framework is illustrated in Fig. 1. The main notations used in this paper and their descriptions are listed in Table 1.

Prior to the input stage, given the presence of random data gaps and feature intricacies, several preprocessing techniques are employed to rectify the utilized data. Through an in-depth exploration of the influencing factors that contribute to concrete creep and a careful assessment of missing feature volumes, the relevant local data attributes are manually curated and captured. The selected features are presented in Table 1. Furthermore, the experimental results underscore

TABLE 1. Major notations used in this paper.

Notations	Descriptions
$\mathbf{x}$, $\mathbf{x}_i$	input data obtained after preprocessing.
W_i	weights of the CNN.
z	activation input.
$\mathcal{L}$	objective functions.
Θ	learning parameters of the model.
$\widehat{\mathbf{x}}_i$	predicted input data.
$\alpha(\cdot)$	activation function.
$\Omega(\cdot)$	regularization operation.
λ	a regularization parameter.
$\ell(\cdot)$	loss function.
$F(\cdot)$	residual mapping function.
$R(\cdot)$	output of the residual block.
y	real label.
$\hat{y}$	label predicted by the model.
$\tilde{y}$	average value of the predicted labels.
$\bar{y}$	average value of the real labels.
K	KNN parameter.
$\mathcal{D}$	complete dataset.
$\mathcal{D}_{\text{train}}$	training set.
$\mathcal{D}_{\text{test}}$	testing set.
$\widehat{\mathbf{D}}$	dataset with Gaussian noise.
$\mathbf{D}$	dataset without Gaussian noise.
F	features.
N	number of objects.

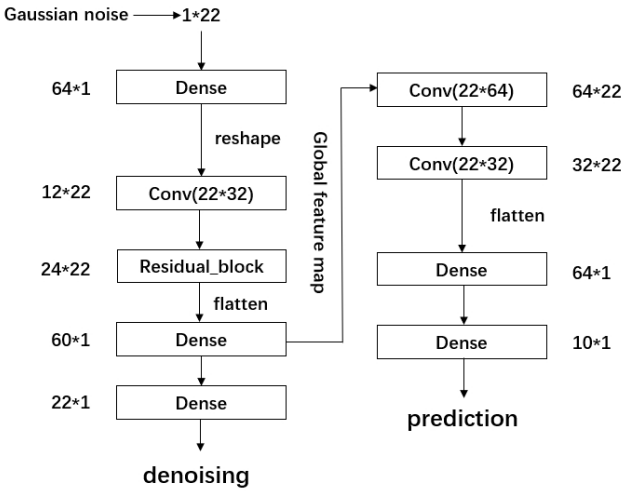

FIGURE 1. Model framework.

the considerable impact of K-nearest neighbors (KNN) on data imputation and postfeature selection. The dataset is augmented to 19,294 entries through KNN interpolation. In addition, for the model to learn to separate noise and data during the training process, we add Gaussian noise to the data to assist the noise reduction encoder in achieving this goal.

In the context of tabular data, the absence of spatial correlations among features necessitates the insignificance of the column positions. To address this issue, a soft-sorting 1-dimensional CNN is introduced prior to the model, reordering the tabular features to construct 2-D pseudoimages to give them spatial relevance.

Furthermore, a key objective of the refined methodology is to reduce the noise within the data, thereby heightening the precision and dependability of the output predictions. Similar to the fact that noise in a city can affect the quality of life of residents, noise in data usually affects the quality of data, which leads to a decrease in model accuracy [33], [34]. Based on this, a noise reduction encoder structure grounded in residual networks is incorporated into the proposed approach. This structure aims to mitigate noise and generate global features. Residual learning contributes to noise reduction within images, enhancing data clarity and, consequently, increasing the accuracy of the results. This choice is predicated on the recognition that small datasets can pose more intricate mapping challenges to the learning processes of neural networks due to the inherent incompleteness of a high-dimensional input space and the intricacies of data sampling. As such, this training strategy fosters model stability and mitigates generalization errors.

Finally, while ReLU activation is usually applied in CNNs, this approach is inappropriate for regression tasks involving tabular data. In contrast, deep learning-based regression analysis requires a smooth activation function. Therefore, the tanh function is adopted to improve the nonlinearity of the network.

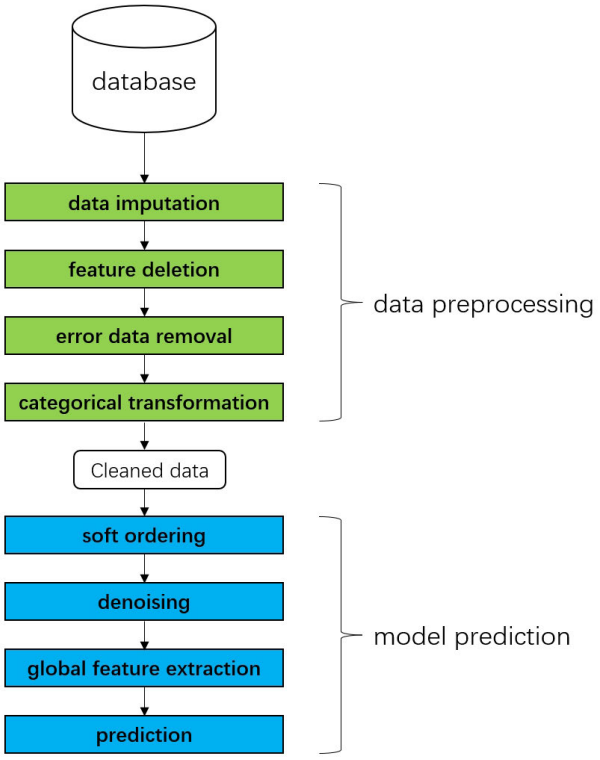

FIGURE 2. Method pipeline.

In summary, the proposed method adeptly addresses the common issues encountered in concrete creep prediction tasks, encompassing missing data, spatial feature correlations, and noise interference. The overall pipeline of the method is shown in Figure 2. It offers an effective means for anticipating the future creep behavior of concrete within its service life based on the discernible data attributes of concrete elements. This approach holds substantial promise for continued research and real-world applications within this domain.

A. SOFT SORTING

In a convolutional neural network, the convolutional kernel often serves as a potent feature extractor, capitalizing on two crucial properties that are inherent to input images: local connectivity and spatial locality [35], [36], [37]. Local connectivity signifies that each kernel establishes a connection with a confined portion of the input image during convolution. Spatial locality refers to the high correlations among the pixels/voxels that are subjected to the convolution kernel. Treating these elements in a joint manner frequently facilitates the extraction of meaningful feature representations. Nonetheless, the disposition of tabular data differs from that of images. Tabular data lack the inherent order and arrangement seen in images, rendering their features spatially uncorrelated [38].

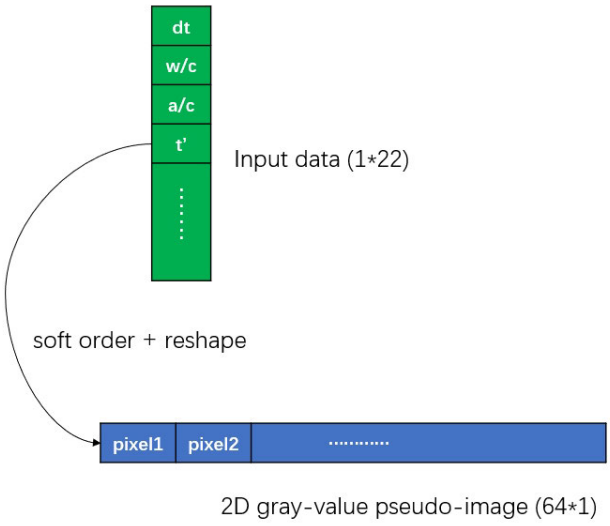

FIGURE 3. The soft sorting rule.

To enable the convolutional kernel to more effectively extract tabular features, referring to Qians work [39] on credit scoring, a direct approach that reshapes the input data into a multichannel image format is adopted. Subsequently, we employ a fully connected (FC) layer to learn the correct ordering through backpropagation, thereby addressing this challenge. The proposed approach involves a soft-sorted one-dimensional CNN, which is shown in Fig. 3. This model is primarily designed to introduce an additional feature dimension via the FC layer. This enhancement provides sufficient pixels to a 2D gray-valued pseudoimage through dimensionality extension and subsequently injects

meaningful structure information into the resulting image through feature ordering.

During this process, the data are transformed into a 64×1 feature image format, similar to an image with a limited height but a broad width. The CNN training process involves working with such images rather than conventional tabular data. As the backpropagation algorithm unfolds, the feature images are constantly adjusted to learn the accurate order and arrangement of their features.

B. RESIDUAL MODULE AND ENCODER STRUCTURE

Due to the excellent noise reduction performance of residual neural networks [40], an encoder-decoder structure is constructed based on a residual module. As shown in the example of Fig. 4,, the encoder is mainly composed of a residual module, which learns the mapping relationship of data to separate the feature noise. The decoder consists of a CNN for learning features and representing the final results.

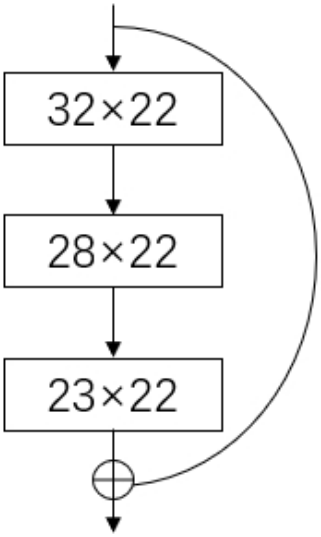

FIGURE 4. Block connection rule.

The skip connections in a ResNet allow the network to improve upon the performance of a CNN on tabular data by reducing data noise. Skip connections enable the model to effectively learn residuals and capture the differences between the input and target variables, thereby removing noise from the data and capturing the potential patterns and relationships between the input and target variables [40]. Moreover, the model is able to study the differences between the input and output instead of learning the entire input-to-output mapping [41]. This enables ResNets to achieve greater depth than traditional CNNs and learn more complex data patterns. The residual connection is defined as follows [41]:

$$R(x) = F(x, W_i) + x \tag{1}$$

where x is the input for a set of convolutional layers, $F(\cdot)$ is the residual mapping function, W_i denotes the weights of the convolutional layers, and $R(\cdot)$ is the output of the residual block.

In the encoder architecture, a methodical process is employed to manipulate the features. A residual module, which comprises one-dimensional convolutions supplemented by auxiliary noise injected into the data, is used to address interference. This constitutes the pivotal phase of the denoising process. During the convolution process, properly expanding the feature channels can enhance the saliency of the features and improve the attained denoising effect [41]. In addition, increasing the number of feature channels allows the model to better capture the nuances of local information [42]. Finally, global features are deployed to capture the common trends and patterns among the features to provide better generalization and model performance. The global features are generated by a one-dimensional convolutional layer. Cumulatively, these operations in the encoder systematically enhance the spatial correlations in the model, effectively reduce noise, and facilitate overall global feature understanding.

C. ACTIVATION

To enhance the ability of the model to represent nonlinear relationships and restrict the output range of the network layers, activation functions are essential in neural networks. Numerous activation functions have been developed, and the ReLU function is widely used in tasks such as classification because of its nonlinearity and ease of computation; however, the ReLU function is a nonlinear activation function that sets all negative values to zero. This can cause a neural network to lose information about the magnitudes of negative data, which is important for regression tasks. In contrast, linear activation functions are better suited for regression tasks because they can output any real number, including negative values. Therefore, the tanh function is favored due to its continuity, monotonicity, clear upper and lower bounds, and effectiveness in regression tasks [43]. The tanh function is adopted as the activation function for all layers except the output layer in this model. The mathematical expression of the tanh function is as follows:

$$\alpha(z) = \frac{e^{z} - e^{-z}}{e^{z} + e^{-z}} \tag{2}$$

The result output by the activation function is denoted as $\alpha(z)$, where z represents the input of the activation function and the computational result of the network layer.

D. LEARNING PROCESS

The loss function and optimizer are crucial components for implementing the backpropagation algorithm in neural networks. The loss function measures the degree of closeness

between the predicted and true values of the utilized network. In most creep prediction models, the MSE is used as the loss function to quantify the deviation between the observed and true values. In the proposed model, two tasks must be simultaneously accomplished: noise reduction and prediction. Therefore, during the learning process, both targets should be simultaneously iterated in the same loss function. This can be expressed as follows:

$$\mathcal{L} = \sum_{i=1}^{N} \left(\|\mathbf{x}_i - \widehat{\mathbf{x}}_i\|_2^2 + \alpha \ell(\widetilde{\mathbf{x}}_i, \widehat{\mathbf{x}}_i) \right) + \lambda \Omega(\Theta) \tag{3}$$

The modified objective function $\mathcal{L}$ is designed to optimize the model by considering three essential aspects: data fitting, noise removal, and regularization. In this formulation, the goal is to find an optimal set of model parameters Θ that balances these factors to enhance the model's performance and generalization capabilities. The variable N represents the total number of data samples in the dataset, and each sample is defined by the true data vector $\mathbf{x}_i$ and the corresponding predicted data vector $\widehat{\mathbf{x}}_i$. The parameter α introduces a trade-off between the noise removal term and the data fitting term, enabling control over the relative importance of each component. The loss function $\ell(\cdot)$ quantifies the dissimilarities between data points. It measures the discrepancy between the noisy data vector $\widetilde{\mathbf{x}}_i$ and the predicted data vector $\widehat{\mathbf{x}}_i$ for each sample, reflecting the model's capacity to mitigate the impact of noise in the observations. The term $\Omega(\Theta)$ contributes a regularization effect to the model. By considering the model parameters Θ, this term discourages complex and overfitted models, ultimately enhancing the model's ability to generalize to unseen data. The λ parameter controls the strength of the regularization term in the overall loss function.

E. DATA PREPROCESSING

This study utilizes the NU database to train and validate the proposed model. The database comprises 29,196 creep compliance observations acquired at various time points and includes additional dimensional characteristics, such as water-cement ratios, admixtures, cement type classifications, aggregate types, and test types [3]. These characteristics provide more valuable information and enhance the generalization performance of the model.

Due to the large proportion of missing values for some important features of the dataset, the KNN method is deployed to complete the data to some extent. KNN interpolation is a machine learning technique used to populate missing values in a dataset by predicting these values based on the values of their nearest neighbors in the dataset [44]. KNN interpolation is effective because it can use information from the surrounding data points to accurately predict missing values. However, different features may contain highly relevant information. To sum up, the accuracy and efficiency of KNN is the reason for applying it to data imputation in this study.

If all features are fully filled, redundant information may be introduced, resulting in excessive computations and reduced data completion accuracy. Therefore, we need to select some particular features to complete instead of filling in the entire dataset [45], [46]. For relatively unimportant features with few missing values, samples containing missing values should be excluded. In addition, the missing values in the NU dataset often exhibit the characteristics of missing whole segments. For example, when data are collected in a certain area over a certain period of time, some features cannot be collected for specific reasons, and all the features displayed in the dataset are missing. To simulate this situation when testing the performance of KNN, a mask of length 10 is set to randomly delete the whole data segment for an affected feature. To select the features that are worth patching, the following experimental procedure is designed 1 for KNN interpolation.

Algorithm 1 KNN-Based Missing Data Imputation

Data: Complete dataset $\mathcal{D}$, KNN parameter K
Determine the number and percentage of missing values in $\mathcal{D}$;
Select the KNN algorithm and set K;
Split $\mathcal{D}$ into a training set $\mathcal{D}_{\text{train}}$ and a test set $\mathcal{D}_{\text{test}}$;
foreach *feature* F *in* $\mathcal{D}$ **do**
  Select F for imputation;
  Fit the KNN model to $\mathcal{D}_{\text{train}}$ using other features;
  Predict the missing values in $\mathcal{D}_{\text{test}}$ using the KNN model;
  Evaluate the model performance using the MAE and RMSE;
  Update the imputation performance list F;
Select the most N effective Fs for F_{set} ;
According to concrete creep knowledge, select the important Fs from F_{set};

After feature completion and selection are performed, the 2 features that are most worthy of data completion, which are W/C and fc28, are selected. Furthermore, the KNN algorithm requires optimization. A hyperparameter optimization method, namely, a grid search, is used to discover the optimal set of hyperparameters. Specifically, a set of possible values is predefined for each hyperparameter, the model is trained and evaluated with all possible hyperparameter value combinations in the grid or matrix, and the optimal set of hyperparameters is selected based on the obtained evaluation metrics. After completing the training and hyperparameter adjustment process, the model with 50 neighbors and distance weights is chosen as the final solution.

Extreme gradient boosting (XGBoost) is an excellent tool for measuring the importance of a feature with respect to accomplishing a target task. During the process of training each decision tree, XGBoost considers the contribution of each feature and segments the features with

TABLE 2. Selected features from the NU database [3].

Symbol	Description
dt	The loading time for creep.
type	The type of creep.
elastic strain	Information regarding whether the data sheet values include the elastic strain.
w/c	The water-to-cement ratio.
a/c	The aggregate-to-cement ratio.
c	Cement dosage without additives.
cem	Cement type according to the CEB Model Code.
SiO2	Silica fume dosage based on the cement weight (%).
FlyAsh	Fly ash dosage based on the cement weight (%).
WR	Water reducer dosage based on the cement weight (%).
Re	Retarder dosage based on the cement weight (%).
AEA	Content of the air entraining agent as a % of the cement weight.
fc28	Compressive strength of the concrete after curing for 28 days.
r	Length or radius of a specimen in mm.
h	Height of a specimen in mm.
V/S	Volume-to-surface ratio in mm.
t'	Age at loading in days.
T	Temperature in C.
RH test	Environmental humidity (H) in % during the test.
sigma	Sustained stress during the test in MPa.

high contributions to reduce the value of the objective function. After training multiple decision trees, XGBoost accumulates or averages the importance of each feature in each decision tree to obtain the final importance score of each feature [47]. Therefore, empirical analysis and XGBoost parameter tuning are conducted for the features in the utilized data, and features that are weakly correlated with creep in the database are removed. In addition, because the neural network can suppress the expression of some features with relatively small contributions, 22 features are still maintained as the final features after completing the feature selection process, as shown in Table 2. In Fig. 6, dt, sigma, fc28 and so on significantly impact the creep performance of concrete. Based on the employed cleaning strategy, 19,294 samples remain as the final input data, as shown in Fig. 5. To measure the performance of the model and prevent overfitting, we divide the cleaned sample set into a training set, a test set, and a verification set at a ratio of 7:2:1.

F. EVALUATION METRICS

Four evaluation metrics are applied to measure the ability of the proposed model to predict concrete creep. They include the correlation coefficient (R), relative error (RE), MAE, and RMSE. The formulas of the four evaluation indicators are as follows:

$$RE = \frac{|\hat{y} - y|}{y} \tag{4}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{5}$$

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{6}$$

$$R = \frac{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(\hat{y}_i - \bar{\hat{y}})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}} \tag{7}$$

Here, $\hat{y}$ represents the label predicted by the model, y represents the real label, $\bar{\hat{y}}$ represents the average value of the predicted labels, $\bar{y}$ represents the average value of the real labels, and n represents the number of samples.

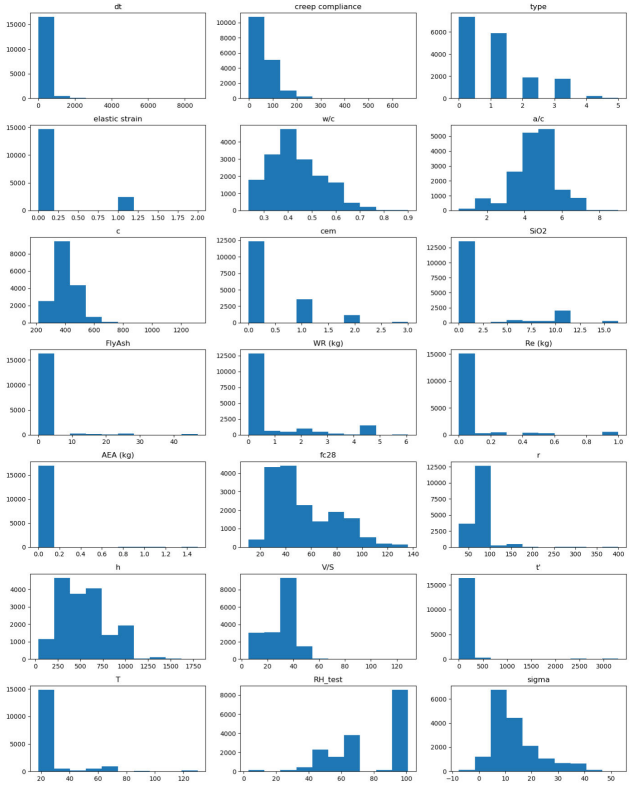

FIGURE 5. Feature distribution of the preprocessed data.

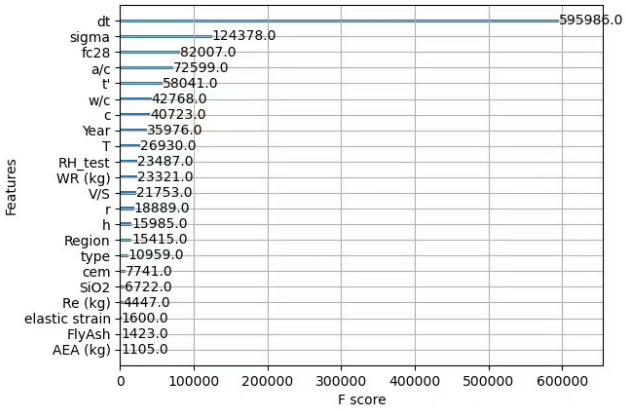

FIGURE 6. Feature importance rankings.

G. PARAMETER SETTINGS AND MODEL TRAINING PROCESS

To assess the stability of the tested models, they should be trained five times, and the means and standard deviations of their MAEs and RMSEs should be calculated. The models' parameters are randomly initialized. To facilitate the quick deployment and validation of the experiments, the PyTorch framework is used for the deep learning approach, and

the scikit-learn library is used for the tree-based approach. During training, the models and data are run on a GPU module to significantly reduce the time needed for model training. This research is performed on a server with an Intel(R) Core(TM) i9-9940 CPU, 16 GB of RAM, a GeForce RTX 3070 GPU, and a CentOS 7 system.

TABLE 3. Performance metrics yielded by the tested methods.

Methods	R	RE	MAE	RMSE
Random Forest	92.80	0.16	8.67±3.86	12.81±2.27
XGBoost	90.19	0.19	9.98±2.66	14.95±3.64
MLP	82.79	0.21	12.52±4.19	19.80±5.04
CNN	93.25	0.13	7.48±1.14	10.76±0.79
DRNN	**95.04**	**0.09**	**5.83±0.62**	**8.88±0.73**

III. RESULTS ANALYSIS

A. OVERALL PERFORMANCE

Several experiments are conducted to measure the performance achieved by the proposed method and other baseline methods on the utilized dataset and conduct a performance comparison. The baseline approaches include a random forest (RF), extreme gradient boosting (XGBoost), a multilayer perceptron (MLP) and a convolutional neural network (CNN). Each model is trained five times to calculate the means and standard deviations of its metrics. Table 3 presents the R, RE, MAE, and RMSE scores obtained by all methods, with the best results shown in bold.

Based on the prediction results described by the evaluation metrics in Table 3 and by R in Fig. 7, the following conclusions are produced. (1) The proposed model outperforms all other methods. This impressive performance is mainly due to the efficient extraction of useful information by the model. (2) On the test set, the proposed model demonstrates stable and robust performance, with the smallest standard deviation compared to those of the other models. This means that the proposed model architecture has stability properties that allow it to learn similar representations in different training iterations, making it less affected by noise and changes in the data. This is mainly due to the noise reduction process, which enables the model to suppress randomness in the data and pay more attention to the stable features of the data, resulting in more stable predictions and greatly improving the generalization capability of the proposed model. (3) Among the baseline deep learning methods, the MLP performs significantly worse than the other methods, as it requires more layers and nodes to learn complex features, making it more susceptible to overfitting. The proposed model addresses this issue by using a residual module to achieve improved generalization, while the convolutional layer helps efficiently extract high-dimensional abstract features, allowing the proposed model to capture the overall relationships between features more effectively. (4) In regression tasks, deep learning methods are superior to tree-structured methods, which may still be effective in classification tasks but have limited expression capabilities for capturing the complex nonlinear relationships between features.

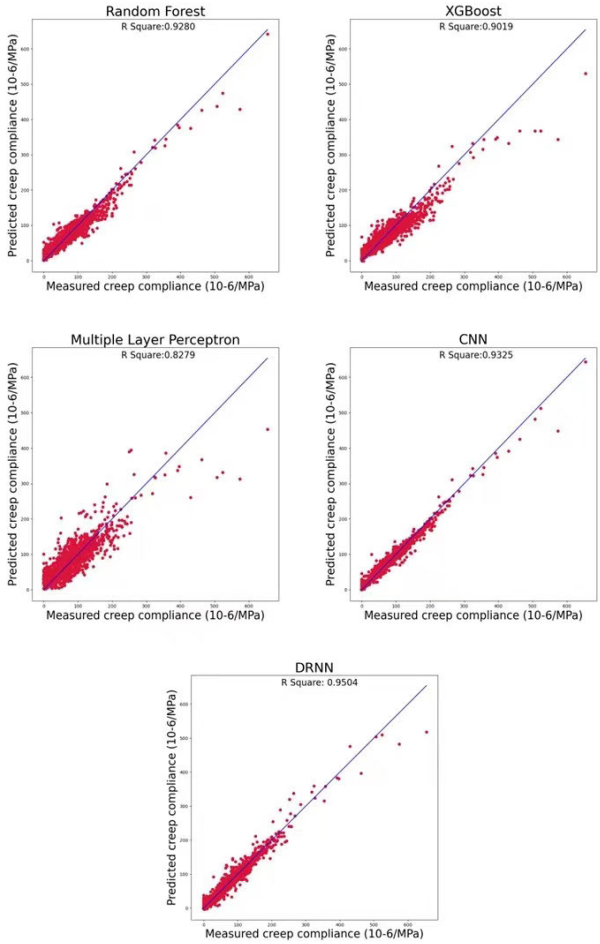

FIGURE 7. Model comparison.

TABLE 4. Performance metrics compared by the referred methods.

Methods	R	MAE	RMSE
RF+bayesian[24]	94.6	5.207	14.836
XGBoost+bayesian[24]	94.7	5.409	14.816
CNN[32]	91.19	7.014	15.563
DRNN	**95.04**	**5.83**	**8.88**

To further measure the advancements provided by the proposed model, the DTNN model is compared with some of the best machine learning and deep learning methods for creep prediction. One ensemble learning approach with Bayesian parameter optimization [24] and one effective CNN with K-means data division [32] are cited for comparison, and the results are shown in Table 4. It can be observed that the proposed model outperforms the other models except in terms of the MAE, and because of the efficient dataset preprocessing strategy, the baseline models also outperform models from other works with respect to some metrics. These results show that the DTNN is better at fitting data

and capturing changes in the target variables. However, its relatively large MAE indicates that the RF and XGBoost achieve better prediction performance than the DTNN in certain circumstances.

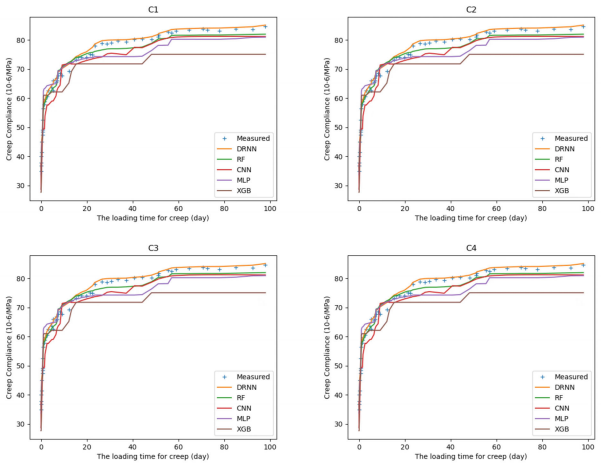

FIGURE 8. Measured and predicted creep.

TABLE 5. Partial parameters of four cases.

Case	Re	w/c	a/c	t'	RH test	T
C1	0	0.5	5.6	28	50	19.5
C2	0	0.55	6.0	24	47.5	20.5
C3	0	0.71	5.6	28	50	19.5
C4	0.5	0.5	5.4	28	47.5	20.5

B. CASE STUDY

To illustrate the practical applicability of the proposed concrete creep prediction model, four case studies are performed from the real world. The target objects are Φ200 mm x 600 mm cylinders; thus, their volume-to-surface (V/S) ratios are 42.8. Some differential treatments are performed on the sample conditions to verify the generalization ability of the proposed model: the temperatures of all samples at the time of the test are 19.5-20.5 degrees, the humidity levels are 47.5-50, the water-cement ratios are 0.4-0.6, and the ages are 24 to 28 days. The duration of the creep measurements for the four samples is 100 days. Finally, to verify the performance of the model under the influence of admixtures, a retarder with 0.5% of the weight of the cement is added to C4. For comparison purposes, the other three samples do not contain any admixtures. Some of the key attributes are shown in Table 5.

The model prediction results obtained based on the DRNN are shown in Fig. 8. For comparison, four other models are used, including the MLP, CNN, RF, and XGBoost approaches. In each scenario, the DRNN exhibits the most agreement with the experimental results. The baseline models can only accurately predict creep compliance in parts of different time periods, but the DRNN demonstrates an excellent fitting ability for all measurement periods; as such, the results of the DRNN are always better than those of the other baseline models. In these four scenarios, despite the excellent performance of some models in some scenarios, the DRNN can always obtain a very good prediction effect, which proves the stability of the DRNN with regard to obtaining prediction conclusions in different scenarios. In addition, although most models achieve good prediction performance in the early stage, their long-term prediction errors are large. Finally, because the DRNN predicts the smoothness of the resulting curve, it provides antinoise and anti-overfitting abilities.

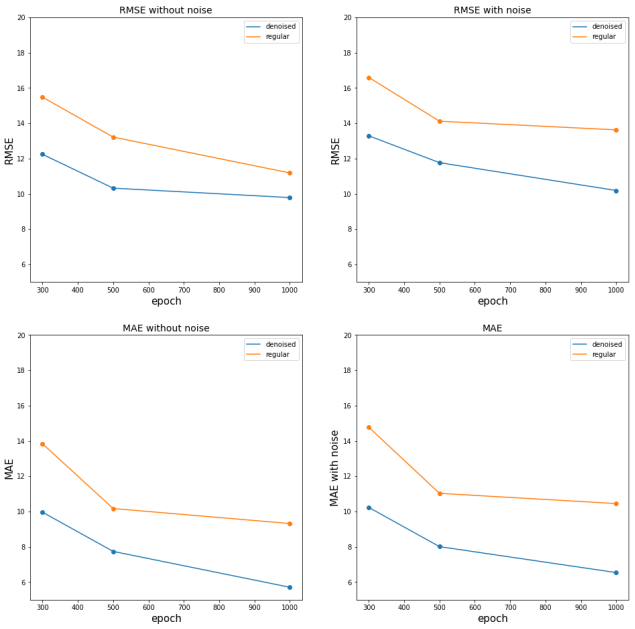

FIGURE 9. Effect of the autoencoder.

C. EFFICIENCY OF THE AUTOENCODER

To substantiate the efficacy of the residual encoder in the proposed model, a comprehensive ablation experiment is devised. This experiment serves to prove the contributions of the dual functionalities of the encoder: denoising and global feature extraction. The objective is to systematically assess the impact of the encoder on the model performance within the context of the input data.

Two dataset configurations defined as $\widehat{\mathbf{D}}$ and $\mathbf{D}$ are designed. $\widehat{\mathbf{D}}$ is infused Gaussian noise to simulate the influences of noise encountered in the real world, while "$\mathbf{D}$ remains pristine, devoid of introduced noise. Two models are also designed: one with an encoder component and the other without this component. To obtain more convincing results, the soft sorting layer is included in both models.

On both $\widehat{\mathbf{D}}$ and $\mathbf{D}$, each model is subjected to independent training and testing processes for 300, 500 and 1000 epochs, respectively. During training, suitable loss functions and

optimizers are deployed to optimize the respective model variants. Last, employing evaluation metrics encompassing the MAE and RMSE, the performance of the two model configurations is quantitatively assessed.

Fig. 9 showcases the comparative performance trends of the complete model equipped with the residual encoder and the baseline model across $\mathbf{D}$ and $\widehat{\mathbf{D}}$. On both datasets, it becomes evident that the baseline model is consistently weaker than the complete model with respect to the MAE and RMSE metrics. This superiority in performance shows that the residual encoder plays a significant role in enhancing the model's predictive accuracy, and on the data with noise, the complete model is less affected by noise than the baseline model. Consequently, the encoder's effectiveness in addressing the denoising and global feature extraction tasks emerges as a key contributing factor to the overall improved performance of the complete model.

This result substantiates that the functionalities of the residual encoder have an effect on augmenting the model's ability to capture potential concrete construction feature patterns while also effectively reducing the influence of noise. This denoising capability is particularly evident in the performance of the complete model when predicting $\widehat{\mathbf{D}}$.

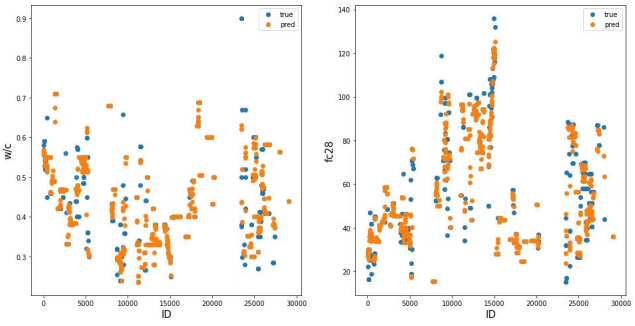

FIGURE 10. The difference between the raw dataset and the KNN-imputed dataset in terms of features W/C and fc28.

TABLE 6. The performance of KNN.

Feature name	MAE	RMSE
fc28	0.16	2.01
w/c	0.0006	0.007

D. EFFICIENCY OF KNN

The data imputation method produces particularly great results for two features, w/c and fc28, in Table 6. These results are characterized by remarkably low MAE and RMSE values. Remarkably, the MAE values obtained for the predicted data points closely align with those of their true data points, indicating a high level of precision in the imputation process. Similarly, the RMSE values are small, highlighting the excellent agreement between the predicted and original values. To further illuminate the experimental results, Fig. 10 is plotted. The figure demonstrates almost impeccable alignment between the predicted points and the actual data points. This remarkable level of correspondence emphasizes the ability of the proposed algorithm to accurately infer missing values.

IV. CONCLUSION

To address the challenge of predicting long-term deformations induced by concrete creep, this study embarks on a comprehensive exploration of data-driven methodologies, seeking reliable deep learning solutions. A network architecture is organized by considering the effect of admixtures; it includes a residual encoder for denoising and global feature extraction, one soft-sorting dense layer in the first layer of the model and a conventional CNN prediction module. This model utilizes the power of deep learning to address complex concrete-related questions.

In addition, this research constructs a series of data preprocessing methods, including a data completion algorithm based on KNN and a feature selection approach based on the feature importance and concrete material knowledge of XGBoost. A total of 19,294 samples are finally obtained while retaining sufficient valid features.

The differences among the performances of different models are investigated, and the results are interpreted. In this endeavor, evaluation metrics, including R, RE, the MAE, and the RMSE, are explored while also accounting for standard deviations to evaluate the stability of the output predictions. These assessments reinforce the robustness and reliability of the proposed model. Compared with a traditional CNN and other baseline methods, the model provides improvements of 95.04, 0.09, 5.83 and 8.88 in the four evaluation metrics, respectively.

Furthermore, through systematic experiments, including an ablation study, the effect of the residual encoder on enhancing the predictive ability of the model is substantiated. Notably, this encoder exhibits dual capabilities: denoising and global feature extraction. The ability of the resultant model to consistently outperform other approaches, as validated across datasets with and without synthetic noise, accentuates the significance of the encoder in terms of mitigating the negative effects of noise while capturing potential feature patterns.

Despite the encouraging results of this study, the models and experiments should continue to be refined.

1) First, this study introduces Gaussian noise in the experiment to simulate noise in real data, but in a real case, the intensity, distribution, and effects of noise may have more complex and disordered characteristics. Therefore, more refined noise simulations can more accurately reflect the noise situations of actual data. One potential solution is to use adversarial neural networks or codec structures to learn and model noise distributions.
2) Second, the soft sorting process is rough, and the single-layer neural network cannot learn fine-grained

spatial characteristics, resulting in pseudoimage failures. A feasible method is to design a more complex pseudoimage generator to sort spatially uncorrelated data.

3) Finally, concrete creep behavior forms data with time series characteristics. Although the proposed model has sufficient feature mining capabilities, the traditional CNN model may not fully consider the influence of time characteristics. Therefore, future research can focus on time series models with limited data. Due to the difficulty of collecting long-term concrete performance characteristics and data volume limitations, LSTM is a highly feasible scheme.

ACKNOWLEDGMENT

The author would like to thank the members of Shandong Yihui New Materials Company Ltd. for their strong support of this research and the enterprise's funding provided for this research.

REFERENCES

[1] S. A. Altoubat and D. A. Lange, "Tensile basic creep: Measurements and behavior at early age," *ACI Mater. J.*, vol. 98, no. 5, pp. 386–393, 2001.

[2] C. Zhao, B. E. Hobbs, A. Ord, P. Hornby, S. Peng, and L. Liu, "Theoretical and numerical analyses of pore-fluid flow patterns around and within inclined large cracks and faults," *Geophys. J. Int.*, vol. 166, no. 2, pp. 970–988, Aug. 2006, doi: 10.1111/j.1365-246X.2006.03049.x.

[3] M. H. Hubler, R. Wendner, and Z. P. Bažant, "Comprehensive database for concrete creep and shrinkage: Analysis and recommendations for testing and recording," *ACI Mater. J.*, vol. 112, no. 4, p. 547, Aug. 2015, doi: 10.14359/51687453.

[4] A. Domingo, C. Lázaro, F. L. Gayarre, M. A. Serrano, and C. López-Colina, "Long term deformations by creep and shrinkage in recycled aggregate concrete," *Mater. Struct.*, vol. 43, no. 8, pp. 1147–1160, Oct. 2010, doi: 10.1617/s11527-009-9573-0.

[5] M. A. Salau, "Long-term deformations of laterized concrete short columns," *Building Environ.*, vol. 38, no. 7, pp. 469–477, 2003, doi: 10.1016/S0360-1323(02)00014-8.

[6] Y. Yuan and Z. L. Wan, "Prediction of cracking within early-age concrete due to thermal, drying and creep behavior," *Cement Concrete Res.*, vol. 32, no. 7, pp. 1053–1059, 2002, doi: 10.1016/S0008-8846(02)00743-3.

[7] M. Rasoolinejad, S. Rahimi-Aghdam, and Z. P. Bažant, "Statistical filtering of useful concrete creep data from imperfect laboratory tests," *Mater. Struct.*, vol. 51, no. 6, pp. 1–14, Dec. 2018, doi: 10.1617/s11527-018-1278-9.

[8] P. Chen, W. Zheng, Y. Wang, K. Du, and W. Chang, "Strain recovery model for concrete after compressive creep," *Construct. Building Mater.*, vol. 199, pp. 746–755, Feb. 2019, doi: 10.1016/j.conbuildmat.2018.12.085.

[9] A. K. Gambali and N. K. Shanagam, "Creep of concrete," *Int. J. Eng. Develop. Res.*, vol. 2, no. 4, pp. 2321–9939, 2014.

[10] H.-J. K. Moon, K.-M. Kim, H.-S. Seok, W.-K. Lee, and B.-G. Kim, "Suggestion of the prediction model for material properties and creep of 60~80MPa grade high strength concrete," *J. Korea Inst. Building Construct.*, vol. 18, no. 6, pp. 517–525, 2018, doi: 10.5345/JKIBC.2018.18.6.517.

[11] D. Zou, T. Liu, J. Teng, C. Du, and B. Li, "Influence of creep and drying shrinkage of reinforced concrete shear walls on the axial shortening of high-rise buildings," *Construct. Building Mater.*, vol. 55, pp. 46–56, Mar. 2014, doi: 10.1016/j.conbuildmat.2014.01.034.

[12] A. Neville, "Creep of concrete and behavior of structures Part I: Problems," *Concrete Int.*, vol. 24, no. 5, pp. 59–66, 2002.

[13] J. Trevino and A. Ghali, "Relaxation of steel in prestressed concrete," *PCI J.*, vol. 30, no. 5, pp. 82–94, 1985.

[14] H. R. E. Motlagh and A. Rahai, "Prediction of long-term prestress loss in prestressed concrete beam with corrugated steel web," *Int. J. Civil Eng.*, vol. 20, no. 11, pp. 1309–1325, Nov. 2022, doi: 10.1007/s40999-022-00731-2.

[15] Z. P. Bazant, M. H. Hubler, and Q. Yu, "Damage in prestressed concrete structures due to creep and shrinkage of concrete," in *Handbook of Damage Mechanics*, 2015, pp. 515–564, doi: 10.1007/978-1-4614-5589-9.

[16] F. N. Birhane, S.-I. Kim, and S. Y. Jang, "Long-term deflection of prestressed concrete bridge considering nonuniform shrinkage and crack propagation by equivalent load approach," *Appl. Sci.*, vol. 10, no. 21, p. 7754, Nov. 2020, doi: 10.3390/app10217754.

[17] L. J. Parrott, "Factors influencing relative humidity in concrete," *Mag. Concrete Res.*, vol. 43, no. 154, pp. 45–52, Mar. 1991, doi: 10.1680/macr.1991.43.154.45.

[18] Z. P. Bažant, Y. Xi, and S. Baweja, "Improved prediction model for time-dependent deformations of concrete: Part 7—Short form of BP-KX model, statistics and extrapolation of short-time data," *Mater. Struct.*, vol. 26, no. 10, pp. 567–574, Dec. 1993, doi: 10.1007/bf02472831.

[19] Z. P. Bažant and S. Baweja, "Justification and refinements of model B3 for concrete creep and shrinkage 1. Statistics and sensitivity," *Mater. Struct.*, vol. 28, no. 7, pp. 415–430, Aug. 1995, doi: 10.1007/bf02473078.

[20] Z. P. Bazant, "Prediction of concrete creep and shrinkage: Past, present and future," *Nucl. Eng. Des.*, vol. 203, no. 1, pp. 27–38, 2001, doi: 10.1016/S0029-5493(00)00299-5.

[21] R. Wendner, M. H. Hubler, and Z. P. Bazant, "The B4 model for multi-decade creep and shrinkage prediction," in *Mechanics and Physics of Creep, Shrinkage, and Durability of Concrete*, vol. 28, Jan. 2013, pp. 429–436, doi: 10.1061/9780784413111.051.

[22] M. Wang, X. Yang, and W. Wang, "Establishing a 3D aggregates database from X-ray CT scans of bulk concrete," *Construct. Building Mater.*, vol. 315, Jan. 2022, Art. no. 125740, doi: 10.1016/j.conbuildmat.2021.125740.

[23] J. Cao, H. He, Y. Zhang, W. Zhao, Z. Yan, and H. Zhu, "Crack detection in ultrahigh-performance concrete using robust principal component analysis and characteristic evaluation in the frequency domain," *Struct. Health Monitor.*, vol. 2023, pp. 1–11, Jun. 2023, doi: 10.1177/14759217231178457.

[24] M. Liang, Z. Chang, Z. Wan, Y. Gan, E. Schlangen, and B. Šavija, "Interpretable ensemble-machine-learning models for predicting creep behavior of concrete," *Cement Concrete Compos.*, vol. 125, Jan. 2022, Art. no. 104295, doi: 10.1016/j.cemconcomp.2021.104295.

[25] M. M. R. Taha, A. Noureldin, N. El-Sheimy, and N. G. Shrive, "Artificial neural networks for predicting creep with an example application to structural masonry," *Can. J. Civil Eng.*, vol. 30, no. 3, pp. 523–532, Jun. 2003, doi: 10.1139/l03-003.

[26] M. M. R. Taha, A. Noureldin, N. El-Sheimy, and N. G. Shrive, "Neural network modelling of creep in masonry," *Proc. Inst. Civil Eng.-Struct. Buildings*, vol. 157, no. 4, pp. 279–292, Aug. 2004, doi: 10.1680/stbu.2004.157.4.279.

[27] J. P. Jiang, "Prediction of concrete strength based on BP neural network," *Adv. Mater. Res.*, vols. 341–342, pp. 58–62, Sep. 2011, doi: 10.4028/www.scientific.net/amr.341-342.58.

[28] M. M. Abed, A. El-Shafie, and S. A. B. Osman, "Creep predicting model in masonry structure utilizing dynamic neural network," *J. Comput. Sci.*, vol. 6, no. 5, pp. 597–605, May 2010, doi: 10.3844/jcssp.2010.597.605.

[29] H. Garoosiha, J. Ahmadi, and H. Bayat, "The assessment of Levenberg–Marquardt and Bayesian framework training algorithm for prediction of concrete shrinkage by the artificial neural network," *Cogent Eng.*, vol. 6, no. 1, pp. 1–15, Jan. 2019, Art. no. 1609179, doi: 10.1080/23311916.2019.1609179.

[30] A. El-Shafie and S. Aminah, "Dynamic versus static artificial neural network model for masonry creep deformation," *Proc. Inst. Civil Eng.-Struct. Buildings*, vol. 166, no. 7, pp. 355–366, Jul. 2013, doi: 10.1680/stbu.11.00024.

[31] O. A. Hodhod, T. E. Said, and A. M. Ataya, "Prediction of creep in concrete using genetic programming hybridized with ann," *Comput. Concrete*, vol. 21, no. 5, pp. 513–523, 2018, doi: 10.12989/cac.2018.21.5.513.

[32] J. Zhu and Y. Wang, "Convolutional neural networks for predicting creep and shrinkage of concrete," *Construct. Building Mater.*, vol. 306, Nov. 2021, Art. no. 124868, doi: 10.1016/j.conbuildmat.2021.124868.

[33] L.-H. Guo, S. Cheng, J. Liu, Y. Wang, Y. Cai, and X.-C. Hong, "Does social perception data express the spatio-temporal pattern of perceived urban noise? A case study based on 3,137 noise complaints in Fuzhou, China," *Appl. Acoust.*, vol. 201, Dec. 2022, Art. no. 109129, doi: 10.1016/j.apacoust.2022.109129.

[34] R. Li, H. Zhang, Z. Chen, N. Yu, W. Kong, T. Li, E. Wang, X. Wu, and Y. Liu, "Denoising method of ground-penetrating radar signal based on independent component analysis with multifractal spectrum," *Measurement*, vol. 192, Mar. 2022, Art. no. 110886, doi: 10.1016/j.measurement.2022.110886.
[35] K. O'Shea and R. Nash, "An introduction to convolutional neural networks," 2015, *arXiv:1511.08458*.
[36] W. Rawat and Z. Wang, "Deep convolutional neural networks for image classification: A comprehensive review," *Neural Comput.*, vol. 29, no. 9, pp. 2352–2449, Sep. 2017, doi: 10.1162/neco_a_00990.
[37] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, "1D convolutional neural networks and applications: A survey," *Mech. Syst. Signal Process.*, vol. 151, Apr. 2021, Art. no. 107398, doi: 10.1016/j.ymssp.2020.107398.
[38] V. Borisov, T. Leemann, K. Seßler, J. Haug, M. Pawelczyk, and G. Kasneci, "Deep neural networks and tabular data: A survey," *IEEE Trans. Neural Netw. Learn. Syst.*, 2022, doi: 10.1109/TNNLS.2022.3229161.
[39] H. Qian, P. Ma, S. Gao, and Y. Song, "Soft reordering one-dimensional convolutional neural network for credit scoring," *Knowl.-Based Syst.*, vol. 266, Apr. 2023, Art. no. 110414, doi: 10.1016/j.knosys.2023.110414.
[40] T. Wang, M. Sun, and K. Hu, "Dilated deep residual network for image denoising," 2017, *arXiv:1708.05473*.
[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 770–778, doi: 10.1109/CVPR.2016.90.
[42] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, inception-ResNet and the impact of residual connections on learning," in *Proc. 31st AAAI Conf. Artif. Intell.*, 2017, vol. 31, no. 1, pp. 4278–4284.
[43] J. Pomerat, A. Segev, and R. Datta, "On neural network activation functions and optimizers in relation to polynomial regression," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2019, pp. 6183–6185, doi: 10.1109/BigData47090.2019.9005674.
[44] A. Jadhav, D. Pramod, and K. Ramanathan, "Comparison of performance of data imputation methods for numeric dataset," *Appl. Artif. Intell.*, vol. 33, no. 10, pp. 913–933, Aug. 2019, doi: 10.1080/08839514.2019.1637138.
[45] M. Mera-Gaona, U. Neumann, R. Vargas-Canas, and D. M. López, "Evaluating the impact of multivariate imputation by MICE in feature selection," *PLoS ONE*, vol. 16, no. 7, Jul. 2021, Art. no. e0254720, doi: 10.1371/journal.pone.0254720.
[46] K. Yu, Y. Yang, and W. Ding, "Causal feature selection with missing data," *ACM Trans. Knowl. Discovery Data*, vol. 16, no. 4, pp. 1–24, Aug. 2022, doi: 10.1145/3488055.
[47] Y. Wang and X. S. Ni, "A XGBoost risk model via feature selection and Bayesian hyper-parameter optimization," *Int. J. Database Manage. Syst.*, vol. 11, no. 1, pp. 01–17, Feb. 2019, doi: 10.5121/ijdms.2019.11101.

CONGHUI LI received the bachelor's degree in computer science and technology from the Chengdu College, University of Electronic Science and Technology of China, in 2019, and the master's degree in data science from the Department of Information Technology, Monash University, in 2021. He is currently a Teaching Assistant with the Department of Artificial Intelligence and Big Data, Zibo Vocational Institute. His main research interests include deep learning and building materials.

MING ZHANG received the master's degree from the Chemistry and Chemical Engineering College, Shandong University of Technology, in 2007, and the Ph.D. degree from the Chemistry and Chemical Engineering College, University of Jinan, in 2023. He has been an Engineer with Shandong Huawei Yinkai Building Materials Science and Technology Company Ltd., since 2007. He is currently a Professor with the Department of Architectural Engineering, Zibo Vocational Institute, China. His research interest includes new building materials.

XIUZHI ZHANG received the bachelor's degree in polymer materials from the University of Jinan, in 1997, the master's degree in applied chemistry from the Nanjing University of Technology, in 2003, and the Ph.D. degree in materials science and engineering from Southeast University, in 2010. In 2005, she was the Technical Director of Shandong Jinteide Concrete Company Ltd., for one year. She is currently an Associate Professor with the School of Materials Science and Engineering, University of Jinan. Her main research interests include composite materials and inorganic nonmetallic materials.