Predicting Autogenous Shrinkage of Concrete Including Superabsorbent Polymers and Other Cementitious Ingredients Using Convolution-based Algorithms

Mohammad Sadegh Barkhordari[1], Sobhan Ghavaminejad[2], Mohsen Tehranizadeh[1*]

[1] Department of Civil Engineering, Amirkabir University of Technology (Tehran Polytechnic), Hafez Ave, Valiasr Square, 1591634311 Tehran, Iran
[2] Faculty of Engineering, Pardis Science and Technology Branch, Islamic Azad University, Damavand Ave, 1658174583 Pardis, Iran
* Corresponding author, e-mail: tehranizadeh@aut.ac.ir

Abstract

In this paper, the effectiveness of ensemble convolution-based deep learning models is evaluated for predicting autogenous shrinkage/swelling of cementitious materials. Various ensemble learning techniques are employed, including Simple Average Ensemble, Snapshot Ensemble, and Stacked Generalization, to develop predictive models. The models are trained and evaluated using performance metrics such as Root Mean Squared Error, Coefficient of Determination, Overall Index of model performance, Mean Absolute Error, and 95% Uncertainty. The results show that the integrated stacking model (ISM) outperforms other models in terms of predictive accuracy. Furthermore, the SHapley Additive exPlanation (SHAP) technique was used to interpret the ISM model. The analysis reveals that the most influential factors affecting shrinkage predictions include time, aggregate to cement ratio (A/C), superabsorbent polymer (SAP) content, water to binder ratio, cement content, water to cement ratio, and silica fume content. Also, the ISM model was compared with models developed previously by other researchers, namely, K-Nearest Neighbors (KNN), Random Forest (RF), Gradient Boosting (GB), and Extreme Gradient Boosting (XGB). With the lowest RMSE and MAE values, the ISM model has exceptional accuracy, demonstrating its capacity to create predictions that closely resemble observed values. Additionally, it has the highest coefficient of determination value, demonstrating its effectiveness in explaining a sizable percentage of the data variance. The Overall Index (OI) statistic shows that the ISM model performs exceptionally well, indicating that it captures more of the underlying information in the data. Additionally, it displays lower 95% confidence intervals, demonstrating greater assurance in its forecasts.

Keywords

concrete shrinkage, machine learning, autogenous shrinkage prediction, ensemble network

1 Introduction

Due to its affordability, robustness, and readily accessible local components, concrete is the most used substance created by human. Many different types of cementitious materials have been created, including self-compacting, lightweight, and high- or ultra-high-performance concrete [1]. These materials, which were carefully chosen based on the necessary mechanical qualities and endurance needed based on their potential exposure to the environment, can be vulnerable to many sorts of degradation, with cracking being one of the most harmful [2]. Many factors can cause cracking to develop during a structure's lifetime. Restrained shrinkage, which happens when a structural part has a tendency to shrink but is unable to do so because of surrounding components, is one of the main reasons for cracking within the first few weeks following casting [3, 4]. Autogenous shrinkage is the term employed to describe the decrease in the macroscopic volume of concrete brought on by chemical shrinkage and self-desiccation as a consequence of the ongoing hydration of cement paste in an atmosphere with stable temperature and humidity [5, 6]. Concrete autogenous shrinkage can be minimized by adding specific additives [7]. Many different additives can be employed, including both natural and synthetic ingredients [8].

For instance, certain light-weight aggregates, such as pumice, can aid in reducing shrinkage because of their porosity, which gradually releases water in the first few days following casting [9]. Concrete shrinkage and crack-

ing have been successfully reduced by the use of super-absorbent polymers (SAP), which are polymer particles that can absorb and hold additional water during mixing and then release it within the first few days [10]. It has been demonstrated that SAP can significantly minimize autogenous shrinkage, drying shrinkage, and stress development when added to the mixture at 0.2–0.6% of cement mass [11, 12]. It should be noted, though, that as the SAP particles empty, certain further deformations might take place [11, 12]. In addition, the high content of SAP is not desirable since it can impair mechanical properties [13, 14]. Although the chemical structure of SAP varies, it has been discovered that SAP with various absorption qualities can effectively reduce autogenous shrinkage [10]. SAP can help with drying, plastic shrinkage, self-healing, and freeze-thaw resistance in addition to reducing autogenous shrinkage [10].

It is crucial to comprehend how SAP affects autogenous shrinkage in high-performance and ultra-high-performance concrete mixes that contain extra cementitious elements to minimize damaging cracking in contemporary concrete. Such complex behavior can be found and predicted using machine learning (ML) approaches [15]. In reality, artificial intelligence has been effectively applied in several civil engineering applications, including forecasting the strength, creep, fracture detection, and microstructural characteristics of concrete. The precision and robustness of ensemble ML algorithms against overfitting make them an attractive option. Nevertheless, the majority of current ML algorithms for drying shrinkage modeling or autogenous shrinkage modeling do not take SAP into account.

In this study, a diverse range of advanced ensemble convolution-based deep learning algorithms are employed to predict concrete shrinkage effectively. Our approach incorporates Snapshot ensembles, Integrated Stacking Model (ISM), and Separate Stacking Model (SSM), each offering distinct advantages in handling complex prediction tasks. Snapshot ensembles leverage the concept of training multiple models simultaneously with different initializations, enabling us to capture a broader spectrum of features and enhance model robustness [16, 17]. ISM, on the other hand, combines predictions from multiple base learners in a unified framework, exploiting the complementary strengths of individual models to achieve superior predictive performance [16, 17]. Meanwhile, SSM employs a distinct approach by training and combining separate models on different subsets of the dataset, promoting diversity and mitigating overfitting [16, 17].

This paper's main goal is to evaluate the effectiveness of ensemble convolution-based deep learning models for the autogenous shrinkage prediction in concrete composites and to offer a model that is more precise than the existing models. The manuscript includes an overview of the theory and method used to create the models, a discussion of the findings, and an analysis of the top model candidate using the SHapley Additive exPlanation (SHAP) theory to identify the characteristics that have the greatest impact on shrinkage and swelling predictions. In addition, the results of the best model developed in this research are compared with the results of recently developed models. The sections of this article are as follows: The data sets are explained in Section 2. The concept of ensemble convolution-based learning models is presented in Section 3. Discussing the findings is explained in Section 4. Finally, Section 5 presents the conclusions.

2 Database description

The database utilized in this study was created by Hilloulin and Tranand [18]. It is currently being employed by others as a standard for model development and comparison. The database including 437 autogenous shrinkage curves was utilized to create a prediction model for autogenous shrinkage of concrete containing superabsorbent polymers and other cementitious ingredients. These autogenous shrinkage curves yielded more than 1800 shrinkage data points. Table 1 shows 14 inputs and one out variables. The compressive strength and Young's modulus are purposefully omitted (as inputs) to build a practical model that only needs the simple inputs and no extra experiments are required. The shrinking/swelling value at a certain age is the model's targeted value. Additionally, the shrinkage measurement technique, like inductive sensors, laser sensors, hydrostatic scales, and buoyancy method, among others, is not taken into account because some databases neglect to provide information about the shrinkage measurement approach.

The correlation matrix of the input from a database focusing on autogenous shrinkage was computed. Fig. 1 displays correlation coefficients between input variables. Based on the correlation matrix, a moderately strong correlation was observed between the water-to-cement ratio and water-to-binder ratio (a correlation coefficient of 0.61). In addition, a correlation coefficient of −0.78 indicates a strong inverse relationship between the aggregate-to-cement ratio and the amount of cement used in the concrete mix. In other words, as the aggregate-to-cement ratio

Table 1 Modeling parameters for autogenous shrinkage of concrete

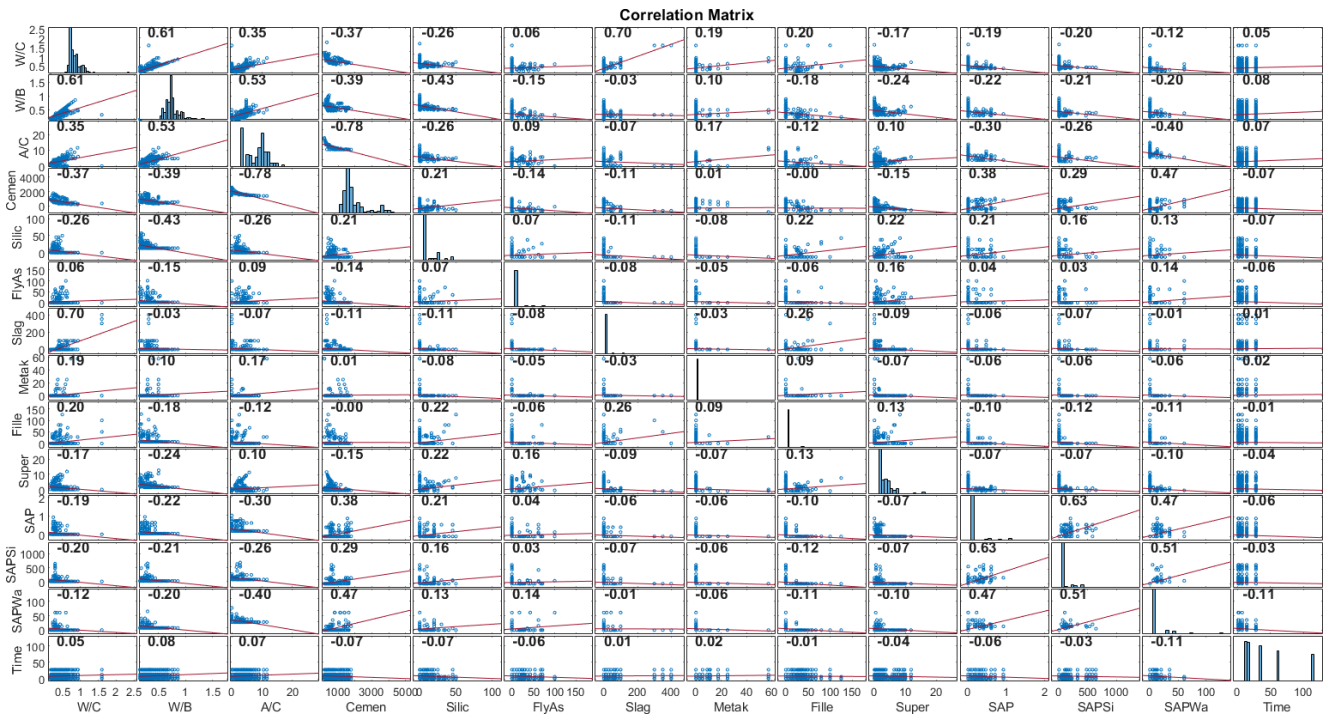|	Input/output	Symbol	Mean	Standard deviation	Min.	Max.
1	Water-to-cement ratios	W/C	0.40	0.17	0.17	1.60
2	Water-to-binder ratios	W/B	0.33	0.11	0.16	0.86
3	Aggregate-to-cement ratio	A/C	2.83	2.02	0.00	11.56
4	Cement (kg/m^3)	Cemen	637.41	364.99	167.40	1762.00
5	Silica fume (%)	Silic	4.80	8.44	0.00	50.00
6	Fly ash (%)	FlyAs	5.13	14.32	0.00	100.00
7	Slag (%)	Slag	8.05	36.71	0.00	400.00
8	Metakaolin (%)	Metak	0.70	4.85	0.00	57.40
9	Filler (%)	Fille	4.51	13.83	0.00	125.00
10	Superplasticizer (%)	Super	1.39	1.98	0.00	11.82
11	SAP (%)	SAP	0.06	0.16	0.00	0.92
12	SAP size (μm)	SAPSi	43.01	107.15	0.00	645.00
13	SAP water uptake (%)	SAPWa	4.51	10.89	0.00	61.00
14	Time (days)	Time	9.06	9.40	1.00	28.00
15	Shrinkage/Swelling (μm)	–	−280.93	491.77	−3818.90	1166.70

Fig. 1 Correlation coefficients between input variables

increases (meaning more aggregate relative to cement), the amount of cement used decreases. Additionally, correlations were identified between the percentages of SAP, the size of SAP, and the water uptake of SAP. Reasons for these correlations:

1. Water-cement ratio, water-binder ratio, aggregate-cement ratio, and cement content correlations: These parameters are crucial factors in the concrete mix design. The water-cement ratio influences the workability and strength of the concrete. The water-binder ratio also affects the workability and strength. The aggregate/cement ratio, as well as the absolute cement content, have a significant impact on the fundamental composition and density of concrete. This relationship is crucial because it controls the complex interactions between the various components that make up the concrete matrix. Changes in these characteristics significantly impact the way

important components, such cement's binding qualities and the distribution of fine and coarse aggregates, work together to build the final concrete's performance, durability, and structural integrity. Correlations among these parameters suggest that changes in one may necessitate adjustments in others to maintain the desired properties and performance of the concrete mixture.

2. Percentage of SAP, SAP Size, and SAP water uptake correlations: SAP is materials added to concrete to improve its properties like shrinkage control. The correlation between the percentage of SAP and SAP size could be explained by the fact that a larger SAP size might have a higher water-absorbing capacity, affecting the overall water content in the concrete mixture. The correlation between the percentage of SAP and SAP water uptake indicates that higher SAP content could lead to more water absorption, potentially affecting the autogenous shrinkage behavior of the concrete. The interplay of these factors suggests that the incorporation of SAP introduces a complex interaction between material properties and performance characteristics, leading to observed correlations.

3 Ensemble model

With an emphasis on developing strong and flexible models, deep learning has grown in popularity. These models are very beneficial for tackling challenging issues like image recognition and predictive analytics [19]. Deep learning networks have a significant disadvantage despite the fact that they increase flexibility and scalability in proportion to the amount of training data available. They are trained using a stochastic process, thus each time they are trained, they may find a different set of weights since they are sensitive to the details of the training data. Because of this, the model's predictions can differ, which is known as high variance in convolutional neural networks. While trying to develop a final model, this might be frustrating. Researchers have created ensemble learning techniques, which entail training many models and aggregating their predictions, to overcome this problem. When compared to using a single model alone, ensemble learning can even give outcomes that are more accurate in terms of prediction variance. A more reliable prediction model is produced by combining different models, which enhances performance and increases stability [19].

When working with complex data sets, where it might be challenging to pinpoint the most crucial features or relationships, ensemble learning is particularly useful. Ensemble learning enables researchers to capture a wider range of patterns and relationships by utilizing multiple models, resulting in a more thorough understanding of the data [20]. Convolutional neural networks are just one example of ML learning, which is a general strategy that may be used with any ML model. It has been widely applied in deep learning and has shown success in a variety of tasks, including speech recognition, image classification, and natural language processing [21]. The method is also helpful when the underlying data is prone to errors, noise, or other types of uncertainty. When this occurs, ensemble learning can aid in minimizing the effect that these variables have on the final predictions. In this study, simple average ensemble, stacked generalization, and convolutional neural networks (CNN) as well as deep forest are used to develop a model for the autogenous shrinkage/swelling prediction of cementitious materials with SAP and additional cementitious materials. Since CNNs are base learners, in the next section a brief explanation about them is provided.

3.1 Convolutional neural network (CNN)

Convolutional neural networks (CNNs) are a kind of neural networks that are frequently employed in computer vision applications like segmentation, object detection, and image and video recognition [22]. By applying a collection of trainable filters, called kernels or weights, to the input data, CNNs are intended to automatically and adaptively learn spatial hierarchies of features from raw input data. A high-level representation of the input data that may be used for classification or other tasks is created by combining the learned features. The ability of CNNs to automatically learn features from unprocessed input data without the need for feature engineering is one of their main advantages. As a result, they are especially beneficial where the input data may be complex and high-dimensional. Using the spatial correlations between the characteristics in the input data is another benefit of CNNs. CNNs can learn to recognize patterns and features that are spatially connected by applying filters to small areas of the input data.

Convolutional layers, pooling layers, and fully connected layer (flatten layer) are the fundamental components of a CNN (Fig. 2). In CNNs, convolutional layers serve as the primary feature extractors. Each feature map represents a different feature or pattern that the network has learned, and they are created by applying a series of filters to the input data. By backpropagating the error from the input layer to the output layer and modifying the filter
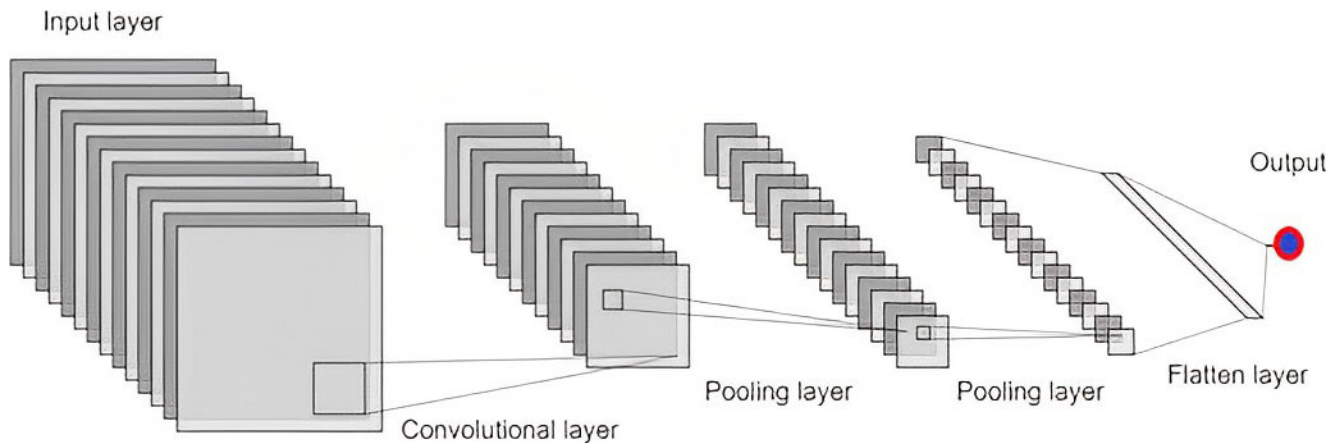
Fig. 2 Fundamental components of the CNN

weights accordingly, the filters are learned. The feature maps created by the convolutional layers are down-sampled using pooling layers. They maintain the most crucial features while lowering the spatial resolution of the feature maps. This aids in lowering the network's computational load and avoiding overfitting. The output of the convolutional and pooling layers is connected to the output layer using fully connected layers. To create the final result, they do a weighted sum of the input data and then apply a nonlinear activation function. The network is given nonlinearity by using the rectified linear unit (ReLU) activation function. They use a rectified linear activation function, which has been demonstrated to enhance network performance, on the output of the preceding layer. The weights of the filters are then modified based on the error, which enhances the network's functionality. The back-propagation algorithm is used to effectively train networks through a method called chain rule.

3.2 Simple average ensemble

It is feasible to produce a prediction that is more reliable and consistent and is less likely to be impacted by random variations in the data by integrating the predictions of various models. The Simple Average Ensemble (SAE) (Fig. 3) is a well-liked ML method for combining the results of various models. The process is simple; to produce the final prediction, the outputs from each model are averaged. As it considers the advantages and disadvantages of each particular model, this technique can enhance the performance of the model. This indicates that it can be applied to a variety of situations without the need for specialist knowledge or equipment. Moreover, it has been demonstrated to be successful in enhancing the functionality of individual models, particularly when working with noisy or ambiguous data. The ability to reduce overfitting, which can be a prevalent issue when working with complex models, is another advantage of the SAE. When working with
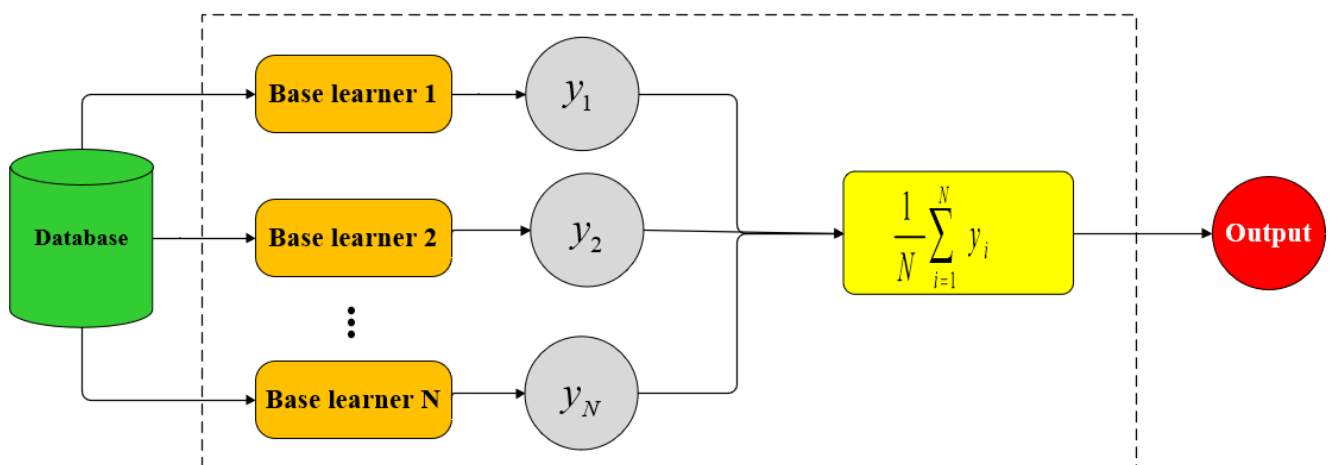

Fig. 3 Simple average ensemble (SAE) flowchart

high-dimensional data or when there is a lot of noise or fluctuation in the data, this can be quite helpful.

3.3 Snapshot ensembles

Snapshot ensembles are a potent and ground-breaking deep learning technique that combines the advantages of various models during training to improve the performance and stability of neural networks [17]. Snapshot ensembles present a novel method where a single model goes through multiple "snapshots" of its learning trajectory during training, in contrast to typical assembling approaches that rely on averaging the predictions of separately trained models. The model's performance is captured in these snapshots at various points, and the ensemble is created by merging the data from these snapshots. Utilizing the cyclical learning rate schedule is the core concept underlying Snapshot ensembles. Deep learning models are typically taught utilizing a constant learning rate for the training procedure. Contrarily, the cyclical learning rate policy alternates between raising and reducing the learning rate regularly, which has been demonstrated to accelerate model generalization. Models can explore the solution space and break out of local minima with the aid of this policy.

Snapshot ensembles' training can be divided into the following steps:

1. Model Training: The procedure starts with typical methods for training a neural network. The parameters of the network are changed using backpropagation and an optimization technique like Stochastic Gradient Descent (SGD) or one of its variants after the network has been exposed to the training data.
2. Schedule for Cyclical Learning Rate: The learning rate fluctuates throughout the training procedure within a predetermined range. Hyperparameters like cycle length and learning rate range can be selected through experimentation or using domain expertise. The learning rate can be changed to explore different areas of the solution space by changing the model's optimization trajectory.
3. Creation of Snapshots: The model's parameters are saved at certain points during the training. Here, the parameters and performance of the model are recorded every 400 epochs.
4. Formation of the Ensemble: Following training, the ensemble is created by mixing the saved models. This can be done by combining the predictions of the models at inference time or by averaging the weights of the models from the chosen snapshots.

A method called a learning rate schedule is employed to preserve a diverse set of proficient ensembles during a single training session. This leads to significant modifications in the model's weights and consequently alters the model's traits in each saved snapshot. The technique involves adjusting the learning rate as training epochs progress using the cosine annealing learning rate equation (Eq. (1)). Initially, the learning rate begins at a high value, then decreases rapidly to a minimal value before rising once more. This cyclic process is repeated to yield favorable weights after each cycle, capturing a snapshot of the model's state:

$$lr(t) = \frac{lr_0}{2}\left(\cos\left(\frac{\pi \% \left(t-1, \lfloor T/M \rfloor\right)}{\lfloor T/M \rfloor}\right) + 1\right), \tag{1}$$

where $lr(t)$ is the learning rate at epoch t, lr_0 is the maximum learning rate, T is the total epochs, M is the number of cycles, % is the modulo operation, and $\lfloor \ \rfloor$ presents a floor operation. Fig. 4 shows a plot of the learning rate schedule $T = 4000$, $M = 20$, $lr_0 = 0.05$.

3.4 Stacked generalization

A meta-learning technique called stacked generalization (SG), commonly referred to as stacking, integrates numerous basic models to enhance the predictive performance of an ML model (Fig. 5) [17, 23]. For stacking to function, a variety of base models must first be trained on training data, and their predictions must then be fed into a meta-learner, a higher-level model. Here, CNNs are used as basic models. The meta-learner then discovers how to merge the base model predictions into a final prediction. Stacking can outperform any single base model in terms of accuracy and error rates by combining the advantages of numerous base models. Another benefit of stacking is
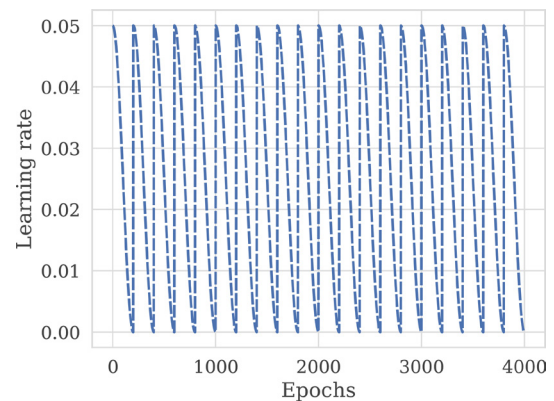

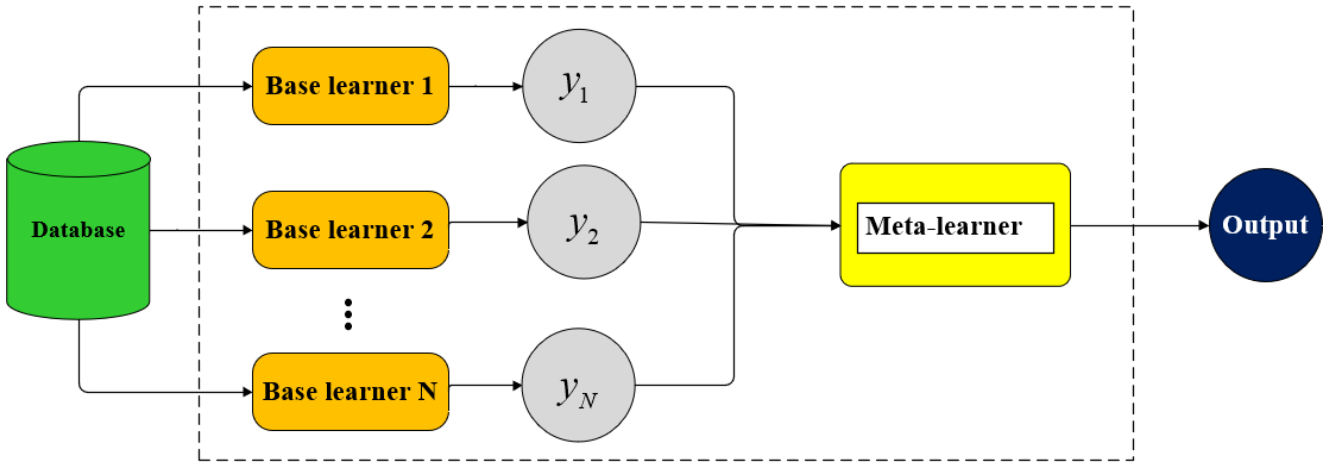**Fig. 4** Snapshot learning rate schedule

Fig. 5 Stacked generalization flowchart

that it can handle a variety of ML tasks, including classification, regression, time-series forecasting, and anomaly detection. As it can be applied to any number of base models and meta-learners and is simple to embed into existing ML pipelines, the SG is also a versatile and scalable approach. The SG is divided into two categories. If a neural network is used as a meta-learner, it is called an integrated stacking model (ISM), and otherwise, it is called a separate stacking model (SSM).

3.4.1 ISM

The meta-learner, constituting a shallow neural network architecture, is characterized by possessing a solitary hidden layer. Notably, the parameters comprising the weights and biases of the sub-models remain static throughout the training process of the stacked generalization model. Only the parameters governing the weights and biases of the meta-learner are subjected to modification during the training procedure. An iterative experimentation protocol known as GridSearchCV is employed to ascertain optimal configurations encompassing neuron quantities in the layer, selection of optimizers, and tuning of hyper-parameters, among other parameters, for the meta-learner. The training dataset is utilized for this purpose. The second table, denoted as Table 2, presents an overview of the attributes associated with the meta-learner.

Table 2 Properties of meta-learner

Meta-learner	
Number of neurons in layer 1	5
Activation	ReLU
Optimizer	Adam
Learning rate	0.02
Metric	MSE

3.4.2 SSM

Unlike ISM, in the SSM, the meta-learner is not a neural network. In this study, DecisionTree regressor (DTR), GradientBoosting regressor (GBR), RandomForest regressor (RFR), AdaBoost regressor (ABR), and Bagging regressor (BR) algorithms are utilized as the meta-learner in the SSM [24]:

- DTR [25]: for regression problems, the DTR is a supervised ML algorithm. It builds a binary tree structure with each leaf node holding a predicted numerical value and each inside node representing a choice based on a chosen attribute. Recursively dividing the data into subgroups that minimize the variance of the target variable within each subset is how the model is constructed. By moving up the tree from the root to a leaf node and then following the choices made at each node, one can determine the forecast for a new data point. The output is the expected value at the leaf node. The DTR can be modeled mathematically as:

$$y = f(x) = \sum_{i=1}^{N_{leaf}} y_{leaf_i} \times \left\| (x \in leaf_i), \right. \tag{2}$$

where y is the predicted target value, $f(x)$ is the model's prediction function, N_{leaf} is the number of leaf nodes in the tree, y_{leaf_i} is the predicted value at the ith leaf node, $\|(.)$ is the indicator function that evaluates to 1 if the condition is true and 0 otherwise, and x represents the input features.

- GBR [26]: a stronger predictive model is produced using the ensemble technique known as gradient boosting by combining a number of weak learners, frequently decision trees. The GBR constructs trees in a sequential manner, focusing each one on the

flaws of the one before it. The output of the model is the total of all predictions made by each tree, with each forecast scaled by a learning rate. By modifying the forecasts of the next trees to lessen the residuals of the previous ones, the technique minimizes the loss function. The GBR's forecast can be expressed mathematically as:

$$y = \sum_{i=1}^{N_{trees}} \eta \times f_i(\boldsymbol{x}),$$ (3)

where y is the predicted target value, N_{trees} is the number of trees in the ensemble, η is the learning rate, controlling the contribution of each tree, and $f_i(\boldsymbol{x})$ is the prediction of the ith tree.

- RFR [27]: another ensemble method, RFR builds numerous decision trees and aggregates their forecasts to improve accuracy and reduce overfitting. By training each tree on a distinct sample of the data and only taking into account a subset of features at each node split, it introduces randomization. The average of all individual trees' forecasts yields the final prediction. The RFR's mathematical forecast can be expressed as follows:

$$y = \frac{1}{N_{trees}} \sum_{i=1}^{N_{trees}} f_i(\boldsymbol{x}),$$ (4)

where: y is the predicted target value, N_{trees} is the number of trees in the random forest, and $f(\boldsymbol{x})$ is the prediction of the ith tree.

- ABR [28]: by concentrating on the incorrectly categorized instances, the ensemble strategy known as the ABR iteratively enhances the model's performance. It trains a string of weak learners and gives samples that were incorrectly labeled with higher weights. The weighted average of all learners' predictions forms the final forecast. The ABR's prediction is denoted mathematically as:

$$y = \sum_{i=1}^{N_{learners}} \alpha_i \times f_i(\boldsymbol{x}),$$ (5)

where y is the predicted target value, $N_{learners}$ is the number of weak learners in the ensemble, α_i is the weight assigned to the ith learner, and $f(\boldsymbol{x})$ is the prediction of the ith learner.

- BR [29]: bagging, which stands for Bootstrap Aggregating, is a method for reducing variance in ensembles by training numerous instances of the same model on various bootstrap samples of the

training data. By averaging the results of each individual model's projections, the final prediction is produced. It functions effectively with erratic models that are delicate to variations in the training set. The BR's prediction is represented as follows:

$$y = \frac{1}{N_{models}} \sum_{i=1}^{N_{models}} f_i(\boldsymbol{x}),$$ (6)

where y is the predicted target value, N_{models} is the number of models in the ensemble, and $f(\boldsymbol{x})$ is the prediction of the ith model.

3.5 Performance evaluation

To assess the reliability and accuracy of the applied models, several performance indicators are utilized to quantitatively evaluate the agreement between the observed and predicted values. These indicators include:

- Root Mean Squared Error (RMSE, Eq. (7)): the RMSE measures the average difference between the predicted and observed values. It is calculated by taking the square root of the average squared differences between the predicted and observed values.
- Coefficient of Determination (R^2, Eq. (8)): The R^2 value evaluates the goodness of fit of the developed models and indicates the adequacy of the input variables in predicting the target variable. It is calculated as 1 minus the ratio of the sum of squared differences between the predicted and observed values to the sum of squared differences between the observed values and their mean.
- Mean Absolute Error (MAE, Eq. (9)): Mean Absolute Error (MAE) serves as a measure for gauging the average extent of discrepancies between anticipated and factual values within a dataset. This metric provides a direct means of assessing the precision of a model's forecasts. In terms of mathematical computation, MAE involves determining the absolute disparity between each projected value and its associated actual value. Subsequently, these absolute disparities are averaged across the entire set of data points.
- 95% Uncertainty (U_{95}, Eq. (10)): the U_{95} represents a 95% uncertainty band to verify the validity of the model. It is calculated as 1.96 times the square root of the sum of the squared standard deviation and RMSE.
- Overall Index of model performance (OI, Eq. (11)): OI is a mathematical concept that includes all real numbers greater than 0 and less than infinity. In the context of model development, OI with an optimistic

value of zero is used to evaluate the sufficiency and goodness of fit of the developed models. This means that the models are evaluated based on how well they fit the data within this range, with a lower optimistic value indicating a better fit.

$$RMSE = \sum_{i=1}^{N} \frac{1}{N} \left(P_{i,P} - P_{i,O} \right)^2 \qquad (7)$$

$$R^2 = 1 - \frac{\sum_{i=1}^{N} \left(P_{i,P} - P_{i,O} \right)^2}{\sum_{i=1}^{N} \left(P_{i,O} - \bar{P}_{i,O} \right)^2} \qquad (8)$$

$$MAE = \frac{1}{N} \sum_{i=1}^{N} \left| P_{i,O} - P_{i,P} \right| \times \qquad (9)$$

$$U_{95} = 1.96 \sqrt{STDEV^2 + RMSE^2} \qquad (10)$$

$$OI = \frac{1}{2} \left\{ 2 - \frac{RMSE}{P_{i,O}^{max} - P_{i,O}^{min}} - \frac{\sum_{i=1}^{N} \left(P_{i,O} - P_{i,P} \right)^2}{\sum_{i=1}^{N} \left(P_{i,O} - \bar{P}_{i,O} \right)^2} \right\} \qquad (11)$$

In Eqs. (7) to (11):
- $P_{i,O}$ and $P_{i,P}$: these variables denote the observed and predicted values, respectively;
- $\bar{P}_{i,O}$: this variable denotes the average of the observed shear strength;
- N: this variable represents the number of samples in the dataset.

4 Result and discussion

The procedure for developing applied models is briefly discussed before presenting numerical results. The total dataset is divided into three subsets: 70% for training, 15% for validation, and 15% for testing. The same subsets are used for all models. Base learner models use the feed forward-back propagation procedure for learning. 5 models with different structures and architecture, which are displayed in Appendix A, are trained on the same dataset for 4000 epochs. After each forward pass through a network, backpropagation performs a backward pass while adjusting the model's parameters (weights and biases). GridSearchCV from Scikit-Learn is used to try out several values for hyperparameters and select the best values [30].

Overfitting constitutes a prevalent predicament encountered in ML, whereby a model assimilates training data with an undue degree of precision, encapsulating not solely the fundamental patterns but also stochastic fluctuations intrinsic to the data. This outcome engenders a model that manifests exceptional performance metrics when evaluated against the training data, yet falters in the generalization phase when confronted with novel and unobserved data instances. This circumstance emanates from the model's proclivity to evolve into an intricately convoluted and excessively specialized construct, inextricably bound to the intricacies of the training dataset, thereby relinquishing its capacity to furnish precise prognostications upon unfamiliar data points.

This analogy likens to rote memorization of solutions to a particular suite of queries, bereft of a comprehensive comprehension of the underlying concepts, rendering the acquired knowledge ineffectual when confronted with novel inquiries. Learning curves serve as invaluable diagnostic instruments for discerning instances of overfitting within ML models. The learning curves of all the base learners are shown in Fig. 6. Examining the curves shows that the problem of overfitting has not occurred for these models.

The predictions are averaged to create a model averaging ensemble. To determine the impact of the number of members on train phase accuracy in the SAE method, a sensitivity analysis of the member number is performed. Fig. 7 shows the changes in the number of members of a set (ensemble size) versus performance. The performance of each individual model is shown as a blue circle. The results indicate that a model averaging ensemble with 5 models has the maximum value of performance. In other words, initially, as the number of members increases, the value of the R^2 has increased, but gradually, it converges to a value of approximately 0.95.

The second ensemble learning technique discussed in the preceding section is known as the Snapshot ensemble. As explained earlier, this method involves training the base model using a dynamic learning rate schedule. During the training process, different models are saved at various points, and these models are subsequently evaluated and selected to create the final ensemble prediction. Here, base learner 4 is used to train and save the models at various epochs. Fig. 8 shows R^2 values of individual and ensemble snapshot models. Twenty-one models have been saved during the training process. By adding individual models into the ensemble snapshot model, the R^2 value fluctuates around 0.85. Consequently, the ensemble model consisting of the twenty recorded models has been considered for further investigation in the research.

4.1 Comparison of the ML models

In this study, a comprehensive evaluation of eight predictive models, namely ISM, SAE, SNAP, ABR, BR, DTR,

Fig. 6 Learning curves and Exp./Pre. Values for base learners; (a) Learning curve of the base learner 1; (b) Exp./Pre. values for the base learner 1;
(c) Learning curve of the base learner 2; (d) Exp./Pre. values for the base learner 2; (e) Learning curve of the base learner 3; (f) Exp./Pre. values
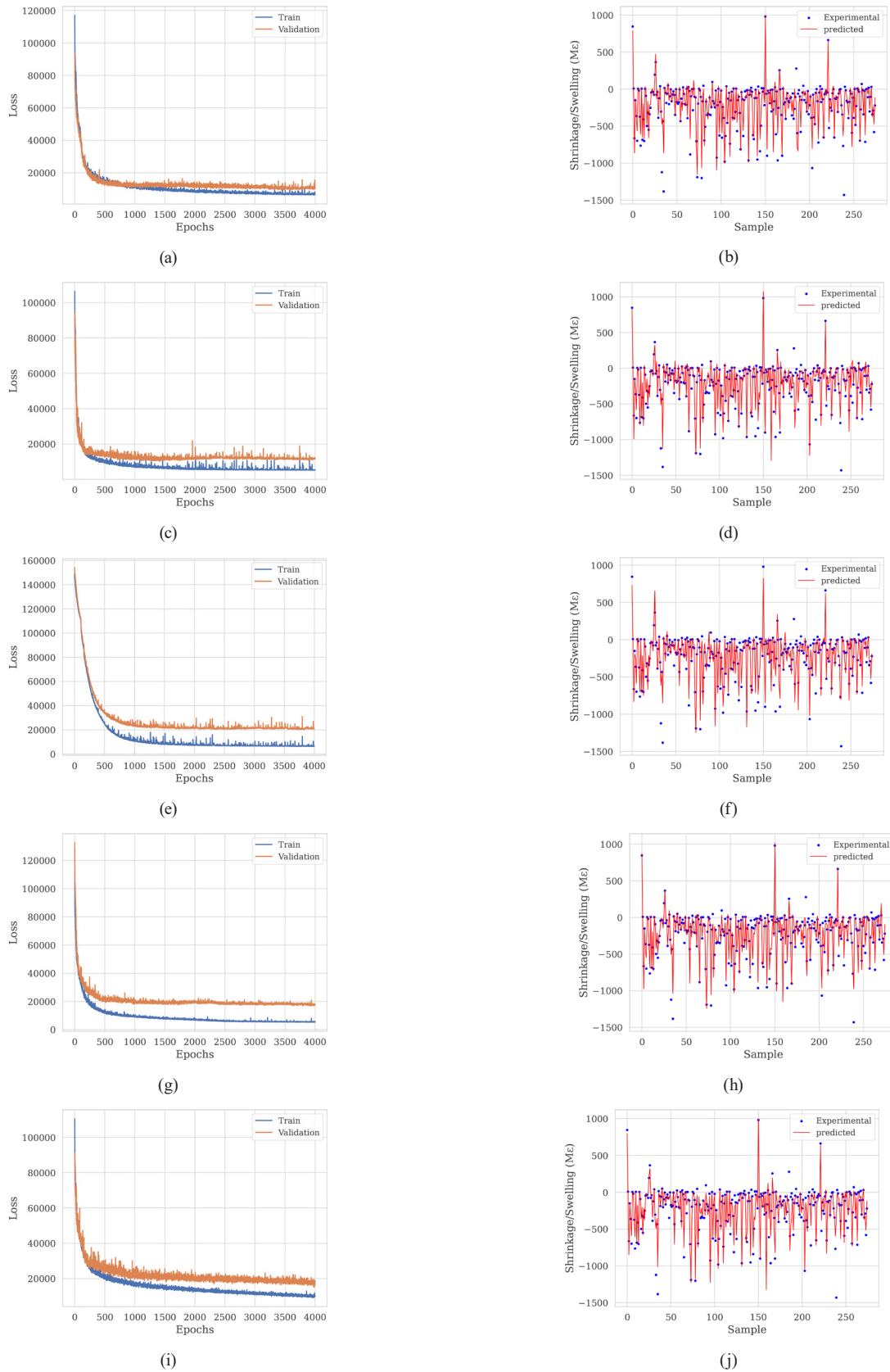for the base learner 3; (g) Learning curve of the base learner 4; (h) Exp./Pre. values for the base learner 4; (i) Learning curve of the base learner 1;
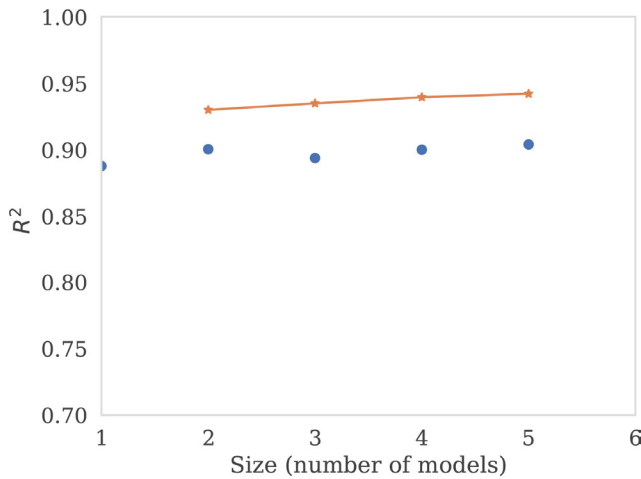(j) Exp./Pre. values for the base learner 1
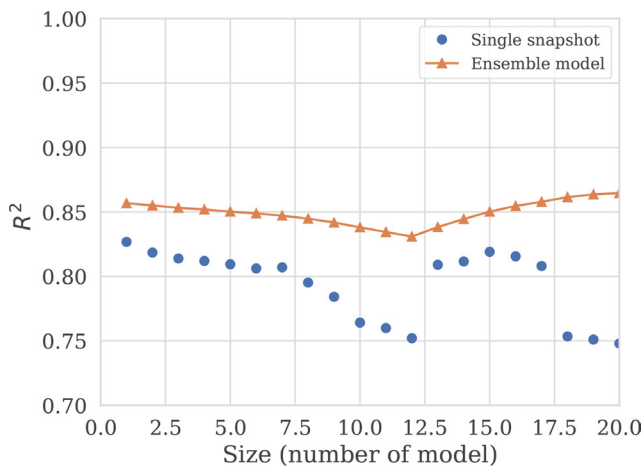
Fig. 7 Changes in the number of members

Fig. 8 R^2 values of individual and ensemble snapshot models

GBR, and RFR, was conducted using various performance metrics. The models were developed for a specific task, and their performance was assessed using five key metrics: Root Mean Squared Error (RMSE), Coefficient of Determination (R^2) Overall Index of model performance (OI), Mean Absolute Error (MAE), and 95% Uncertainty (U_{95}). Tables 3 and 4, along with Fig. 9, display the values corresponding to each metric.

During the training phase (Table 3), results show that the models varied in their predictive capabilities. In terms of RMSE, which measures the average magnitude of errors

between predicted and actual values, SSM-DTR demonstrated the lowest value (63.61), indicating its superior performance in minimizing prediction errors. SSM-RFR and SSM-BR followed closely with RMSE values of 65.55 and 66.57, respectively. The Coefficient of Determination (R^2) values offer insights into the proportion of variance in the target variable that is explained by the models. Among the models, SSM-DTR, SSM-RFR, and SSM-BR exhibited relatively high R^2 values, with SSM-DTR achieving an R^2 value of 0.9604. This signifies that these models are proficient in capturing the variability of the target variable and providing a good fit to the data.

The Overall Index of model performance (OI) takes into account multiple performance aspects of the models. The OI values for the models ranged between 0.9085 and 0.9673. Notably, SSM-DTR and SSM-RFR attained high OI values, indicating their balanced performance across various metrics. The Mean Absolute Error (MAE) measures the average magnitude of absolute errors between predicted and actual values. SSM-DTR stood out with the lowest MAE value of 11.43, implying that, on average, its predictions deviated less from the actual values compared to other models. Additionally, the 95% Uncertainty (U_{95}) values provide an estimate of the dispersion of predictions around the true values. In this case, SNAP and SSM-ABR demonstrated the highest U_{95} values, suggesting higher uncertainty in their predictions compared to other models. SSM-DTR attained lowest U_{95} value.

In the testing phase (as shown in Table 4), the outcomes indicate that the models exhibited varying degrees of predictive capabilities. When considering the Root Mean Square Error (RMSE), which gauges the average extent of discrepancies between projected and actual values, it becomes evident that ISM displayed the most favorable outcome with the smallest value (65.08). This outcome underscores the superior performance of ISM in minimizing the errors associated with predictions. SAE and SSM-BR closely followed suit with RMSE scores of 66.76 and 69.51, respectively. In terms of the R^2 values, which

Table 3 Model results – Training phase

Metrics	Models							
	ISM	SAE	SNAP	SSM-ABR	SSM-BR	SSM-DTR	SSM-GBR	SSM-RFR
RMSE	74.73	76.99	117.62	91.31	66.57	63.61	68.51	65.55
MAE	33.79	36.38	69.01	59.28	20.32	11.43	27.30	19.38
OI	0.9575	0.9554	0.9085	0.9407	0.9648	0.9673	0.9631	0.9656
U_{95}	207.18	213.45	326.04	253.13	184.56	176.35	189.94	181.75
R^2	0.9454	0.9425	0.8703	0.9191	0.9566	0.9604	0.9541	0.9580

Table 4 Model results – Testing phase

Metrics	Models							
	ISM	SAE	SNAP	SSM-ABR	SSM-BR	SSM-DTR	SSM-GBR	SSM-RFR
RMSE	65.08	66.76	132.16	86.67	69.51	82.24	71.29	72.26
MAE	31.528	32.31	81.80	58.66	32.51	38.39	34.45	33.10
OI	0.9646	0.9632	0.8826	0.9433	0.9607	0.9481	0.9590	0.9581
U_{95}	180.19	185.07	366.67	240.16	192.74	228.04	197.66	200.32
R^2	0.9569	0.9550	0.8202	0.9242	0.9504	0.9306	0.9478	0.9464

Fig. 9 Models' evaluation curves; (a) RMSE metric; (b) MAE metric; (c) OI metric; (d) U_{95} metric; (e) R^2

furnish insights into the proportion of variability in the target variable accounted for by the models, ISM, SAE, and SSM-BR demonstrated notable R^2 values. Notably, ISM achieved an impressive R^2 value of 0.9569. This signifies the proficiency of these models in capturing the intricate fluctuations within the target variable, thereby establishing a robust fit to the provided data.

The comprehensive evaluation of model performance, encapsulated within the Overall Index (OI), takes into consideration a multitude of performance metrics. Within the testing phase, the OI values spanning across the models ranged from 0.8826 to 0.9646. It is particularly noteworthy that both ISM and SAE garnered high OI values, a reflection of their well-rounded performance

across various evaluative criteria. Furthermore, the Mean Absolute Error (MAE), which measures the average magnitude of absolute discrepancies between projected and actual values, highlighted ISM's outstanding performance. With a remarkably low MAE value of 31.528, ISM emerges as the frontrunner in terms of maintaining predictions closely aligned with actual values, surpassing the performance of other models in this regard. Additionally, the 95% Uncertainty (U_{95}) values provide a glimpse into the spread of predictions around the true values. In this context, both SNAP and SSM-ABR exhibited the highest U_{95} values, indicating elevated uncertainty in their predictions compared to their model counterparts. Notably, SSM-ISM registered the lowest U_{95} value, suggesting a comparatively reduced level of uncertainty in its predictions.

These discernible findings contribute substantial insights for the purpose of model selection. Drawing from the results of the testing phase, it becomes apparent that ISM emerges as the most suitable candidate. This conclusion applies specifically to the task of predicting autogenous shrinkage/swelling of cementitious materials, where additional cementitious materials and SAP are considered.

4.2 Model interpretation

Even while advanced ML prediction models are extremely accurate and efficient, their use in structural engineering is limited by the fact that they are opaque and difficult to understand. In this study, the proposed ISM model is interpreted using SHapley Additive exPlanation (SHAP) [31], a recent game theory-based agnostic technique. The model is broken down by SHAP into the sum of the impact of each input parameter, and for each feature, a value that indicates the influence of each input on the model output is computed. By comparing the model's output with and without the parameter, the importance of each variable in SHAP is assessed. The input variable with the highest absolute SHAP value is considered to be the most crucial component [32].

In SHAP, the explanation model ($K(\boldsymbol{x'})$) is used to approximate the original model output ($M(\boldsymbol{x})$) as follows:

$$M(\boldsymbol{x}) = K(\boldsymbol{x'}) = E_0 + \sum_{i=1}^{NI} E_i \boldsymbol{x'}_i, \qquad (12)$$

where $\boldsymbol{x'}$ is the simplified input that transfers to $\boldsymbol{x}$ through a function ($\boldsymbol{x} = h\boldsymbol{x}(\boldsymbol{x'})$), NI is the number of inputs, and $E_i \in \mathbb{R}$ denotes the effect of each feature.

The portion of each parameter (E_i) in SHAP is determined as:

$$E_i(M, \boldsymbol{x}) = \sum_{\boldsymbol{T'} \subseteq \boldsymbol{x'}} \frac{|\boldsymbol{T'}|!(NI - |\boldsymbol{T'}| - 1)!}{NI!} \left[\begin{matrix} M_x(\boldsymbol{T'}) \\ -M_x(\boldsymbol{T'} \setminus i) \end{matrix} \right], \qquad (13)$$

where $\boldsymbol{T'} \in \{0, M\}$, $|\boldsymbol{T'}|$ is the number of non-zero entries in $\boldsymbol{T'}$, $\boldsymbol{T'} \subseteq \boldsymbol{x'}$ is all $\boldsymbol{T'}$ vectors where non-zero entries are a subset of the non-zero entries in $\boldsymbol{x'}$, $\boldsymbol{T'} \setminus i$ denotes setting $T_i^0 = 0$, and $M_x(\boldsymbol{T'}) = E\left[(M(T) | \boldsymbol{T}_s)\right]$ denotes SHAP values, where S is the set of non-zero entries in $\boldsymbol{T'}$.

Fig. 10 illustrates the outcomes obtained from the model interpretation using SHAP techniques. The elements presented in the upper section of the visual representation (depicted in Fig. 10(a)) can be linked to the more substantial
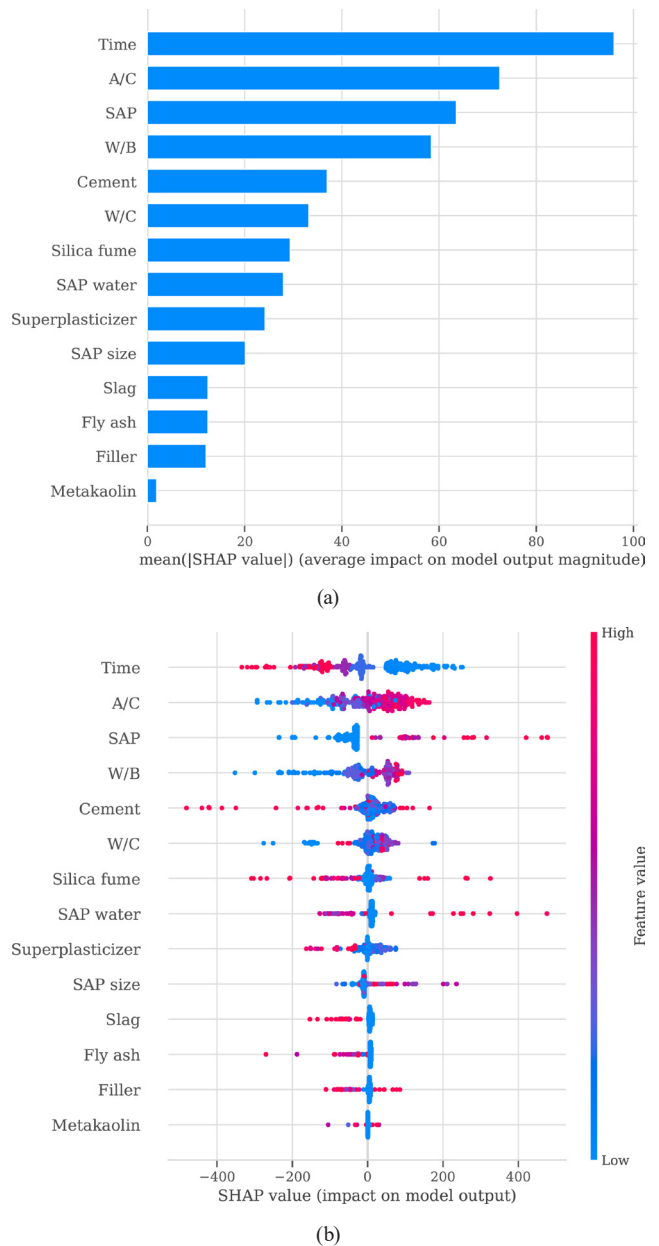

(a)

(b)

Fig. 10 SHAP-based model interpretation; (a) Relative relevance of each input parameter; (b) Summary plot for inputs' effect

contributions affecting the model's outcomes. Predominant factors impacting predictions related to shrinkage encompass time (measured in days since the initiation of shrinkage assessments), the ratio of aggregate to cement (A/C), the content of SAP (Superabsorbent Polymer), the ratio of water to binder, the content of cement, the ratio of water to cement, and the amount of silica fume used. Each factor bears significance in influencing prediction outcomes and holds practical implications [33, 36]:

- Time: time is pivotal in shrinkage prediction due to the evolving properties of concrete over time. Concrete undergoes continuous hydration processes, leading to alterations in microstructure and moisture content. Early stages exhibit higher shrinkage rates due to abundant water content, while later stages may experience diminished shrinkage as hydration reactions decelerate. Accurate modeling of temporal effects aids in long-term performance evaluations of concrete structures.

- Ratio of Aggregate to Cement (A/C): the A/C ratio significantly influences shrinkage as aggregates provide internal resistance against volume changes induced by cement hydration. Higher A/C ratios typically reduce shrinkage due to the dilution effect of aggregates, while lower ratios may exacerbate shrinkage by providing inadequate restraint. Understanding A/C ratio effects facilitates optimizing concrete mix designs for desired shrinkage performance and structural integrity.

- Content of Superabsorbent Polymer (SAP): SAPs mitigate shrinkage by absorbing excess water during hydration, diminishing capillary stresses and shrinkage cracking. Higher SAP contents correlate with lower shrinkage as more water is retained within the concrete matrix. Proper SAP dosage and distribution are critical for shrinkage reduction without compromising other concrete properties. The predictive model's ability to capture the nonlinear SAP-shrinkage relationship aids in informed decision-making regarding SAP incorporation for shrinkage control.

- Ratio of Water to Binder: the water-to-binder ratio significantly influences concrete shrinkage by determining hydration kinetics and water availability. Lower ratios generally elevate shrinkage due to limited water for hydration, while higher ratios may increase shrinkage initially due to greater volume changes. Balancing this ratio is crucial for achieving desired concrete properties, including workability, strength, and shrinkage performance.

- Content of Cement: cement content directly impacts shrinkage through hydration-induced volume changes. Higher cement contents typically amplify shrinkage due to increased cementitious materials, whereas lower contents may reduce shrinkage but compromise strength and durability. Understanding these trade-offs is essential for designing sustainable concrete mixes with optimal performance.

- Ratio of Water to Cement: The water-to-cement ratio governs hydration efficiency and subsequent volume changes, significantly affecting shrinkage. Higher ratios generally correlate with higher shrinkage due to increased water availability, while lower ratios may reduce shrinkage but affect workability and strength development. Optimizing this ratio involves balancing shrinkage considerations with other performance requirements.

- Amount of Silica Fume: Silica fume alters concrete microstructure and pore characteristics, influencing shrinkage behavior. Increased silica fume content densifies the concrete matrix, reducing pore connectivity and capillary stresses, leading to lower shrinkage. However, excessive silica fume may hinder workability due to increased viscosity. Effective utilization requires optimizing dosage for shrinkage reduction while maintaining workability and mechanical properties.

A distinct boundary is discernible for all these variables, differentiating their influence on the model output in terms of high and low feature values (Fig. 10(b)): an elevated A/C ratio leads to increased SHAP values, signifying reduced shrinkage; greater SAP content contributes to decreased shrinkage and is often identified as the most pivotal factor; increased time values correlate with higher levels of shrinkage; augmented water-to-binder and water-to-cement ratios tend to diminish shrinkage; on the contrary, a higher replacement ratio of silica fume or increased cement content predominantly triggers heightened shrinkage. Additionally, larger SAP sizes counteract the beneficial effects of SAP and result in greater shrinkage when contrasted with smaller SAP sizes. These effects correspond closely with findings from experimental observations [18]. Conversely, factors exerting less influence have been identified, including superplasticizer, fly ash, slag, filler, and calcined clay content. These deductions drawn

from the SHAP values are in harmony with empirical observations, as these parameters are acknowledged to wield a comparatively minor impact on shrinkage in comparison to the previously mentioned ones.

4.3 Comparison with previous research

Hilloulin and Tran [18] have developed four kinds of state-of-the-art ML models for predicting autogenous shrinkage of concrete using the database that was also used in this article. The four ML algorithms were: K-Nearest Neighbors (KNN), Random Forest (RF), Gradient Boosting (GB), and Extreme Gradient Boosting (XGB). In Section 4, the results of their models are compared with the results of the best model (ISM) developed in this article. The training set and test set used to create the ISM model in this article were also used for the training and evaluation of the KNN, RF, GB, and XGB models.

During the training phase (Table 5), the XGB outperformed the others with a score of 66.179 in terms of the RMSE, indicating that it had the lowest average prediction error. The RF model followed closely with an RMSE of 76.404, suggesting good predictive accuracy. On the other hand, the KNN had the highest RMSE of 185, indicating that it had the least accurate predictions among the models. Moving on to the MAE, once again, the XGB performed the best with an MAE of 21.9, signifying its ability to make predictions close to the actual values. The RF had the second-lowest MAE at 34.01, showcasing its strong predictive capabilities as well. The KNN had the highest MAE of 111.34, indicating a relatively larger margin of error in its predictions.

The coefficient of determination (R^2) further emphasizes the XGB's excellence, scoring the highest at 0.95719. This indicates that the XGB explained a substantial portion of the variance in the data, suggesting it is a highly effective model for this specific problem. The RF closely followed with an R^2 of 0.94585, showcasing its strong explanatory power. The KNN, on the other hand, had an R^2 of 0.66967, suggesting that it could only explain a moderate amount of variance in the data. However, when it comes to the metric

of the overall information (OI), it's evident that the RF and XGB do not perform as well. The RF and XGB had OI values of 0.95597 and 0.96517, respectively, indicating that they captured more of the data's underlying information compared to the KNN, GB, and ISM. Lastly, considering the 95% uncertainty (U_{95}) values, it's clear that the KNN had the highest U_{95} of 512.25, implying higher uncertainty in its predictions compared to the other models. The GB and ISM also exhibited relatively high uncertainty with U_{95} values of 350.4 and 207.18, respectively. In contrast, the RF and XGB had lower U_{95} values of 211.81 and 183.47, indicating greater confidence in their predictions.

In the testing phase (Table 6), the ISM model stands out as the top performer in terms of the RMSE, with the lowest score of 65.08, indicating its exceptional accuracy in predicting the target variable. The XGB model closely follows with an RMSE of 123.13, showcasing its strong predictive capabilities. On the other hand, the KNN exhibits the highest RMSE at 217.9, suggesting that it has the least accurate predictions among the models, while the GB and RF fall in between with RMSE values of 150.7 and 136.24, respectively. Examining mean absolute error (MAE), once again, the ISM model outperforms its counterparts with the smallest MAE of 31.52, signifying its ability to make predictions that are closest to the actual values. The XGB follows as the second-best performer with an MAE of 59.56, showcasing its competence in making accurate predictions. Conversely, the KNN has the highest MAE at 135.72, indicating a larger margin of error in its predictions, and the GB and RF fall in the middle with MAE values of 92.912 and 71.716, respectively. The coefficient of determination highlights the ISM's excellence, leading the pack with an R^2 of 0.956, which signifies its capacity to explain a substantial portion of the variance in the data. The XGB follows closely with an R^2 of 0.845, demonstrating its robust explanatory power. Conversely, the KNN lags with an R^2 of 0.516, suggesting that it can only account for a moderate amount of variance in the data, while the GB falls in between with an R^2 of 0.766.

Table 5 Models' comparison – Training phase

| Metrics | Models | | | | |
	KNN [18]	RF [18]	GB [18]	XGB [18]	ISM
RMSE	185	76.404	126.39	66.179	74.73
MAE	111.34	34.01	83.034	21.9	33.79
OI	0.79513	0.95597	0.89627	0.96517	0.9575
U_{95}	512.25	211.81	350.4	183.47	207.18
R^2	0.66967	0.94585	0.85253	0.95719	0.945

Table 6 Models' comparison – Testing phase

| Metrics | Models | | | | |
	KNN	RF	GB	XGB	ISM
RMSE	217.9	136.24	150.7	123.13	65.08
MAE	135.72	71.716	92.912	59.56	31.52
OI	0.71	0.876	0.851	0.896	0.964
U_{95}	603.14	377.96	418.11	341.62	180.19
R^2	0.516	0.809	0.766	0.845	0.956

When considering the overall information (OI) metric, the ISM model appears to be more effective, with OI values of 0.964, indicating that they capture more of the underlying information in the data compared to the XGB, GB, and RF. The KNN has the lowest OI value. Lastly, analyzing the 95% uncertainty (U_{95}) values, KNN has the highest U_{95} of 603.14, suggesting a higher level of uncertainty in its predictions compared to the other models. The GB and XGB also exhibit relatively high uncertainty with U95 values of 418.11 and 341.62, respectively. In contrast, RF and the ISM model have lower U_{95} values of 377.96 and 180.19, respectively, indicating greater confidence in their predictions. In summary, the ISM model outperforms other models.

Boxplots of the results in the testing phase are displayed in Fig. 11. Fig. 11 shows that the predicted shear strength values for RF, XGM, and ISM models thoroughly follow the experimental results. However, comparing the difference between experimental and predicted values from Fig. 11 indicates that the KNN had the worst performance in testing phase.

The superior performance of the (ISM) over other convolution-based algorithms be attributed to several key factors [37, 40]:

1. Model Fusion Strategy: ISM combines the strengths of multiple base models through a sophisticated stacking mechanism. Unlike SAE, which relies on simple averaging of predictions, ISM integrates predictions from diverse base models in a strategic manner, leveraging the complementary nature of individual models. This fusion strategy enables ISM to capture nuanced patterns and relationships within the data more effectively, enhancing predictive accuracy [15].
2. Feature Engineering and Selection: ISM incorporates advanced feature engineering techniques and robust feature selection mechanisms since its meta-learner is a neural network. By identifying and incorporating relevant features while filtering out noise and irrelevant information, ISM optimally leverages the available data, resulting in improved model performance compared to other techniques like Snapshot Ensemble and SSM.
3. Model Diversity: ISM harnesses the power of diverse base models, each trained using distinct algorithms or architectures. This diversity mitigates the risk of overfitting and increases the model's ability to generalize to unseen data. In contrast, Snapshot Ensemble and SSM may exhibit limited diversity in their ensemble members, potentially constraining their predictive capabilities.
4. Model Complexity and Flexibility: ISM's ability to capture nonlinear relationships and interactions between predictor variables and shrinkage outcomes allows it to outperform simpler models like KNN. The flexibility of ensemble methods enables ISM to adapt to the inherent complexity of the concrete shrinkage prediction problem more effectively.
5. Robustness to Noise and Outliers: ISM's ensemble nature and robust aggregation strategy make it less susceptible to noise and outliers in the data compared to individual models like RF, GB, and XGB. By combining predictions from multiple models, ISM can effectively filter out erroneous predictions and focus on the underlying patterns in the data.

5 Conclusions

In this paper, the effectiveness of ensemble convolution-based deep learning models for predicting autogenous shrinkage and swelling in cementitious materials was evaluated. The goal was to offer a model that surpasses existing models in terms of precision. To achieve this, a thorough exploration of various ensemble learning techniques was conducted, including Simple Average Ensemble, Snapshot Ensemble, and Stacked Generalization, and compared their performance.

Findings reveal that ensemble learning techniques when applied to the prediction of autogenous shrinkage and swelling, offer substantial advantages. Among the models evaluated, the Integrated Stacking Model (ISM) emerged as the most suitable candidate for this specific task. ISM demonstrated superior performance in terms of RMSE, Coefficient of Determination (R^2), Mean Absolute Error (MAE), and Overall Index of model performance (OI)
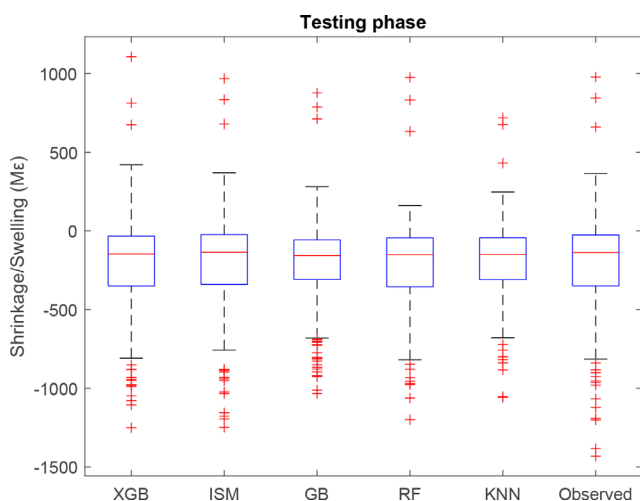
Fig. 11 Boxplots for the autogenous shrinkage of the concrete

during both the training and testing phases. This suggests that ISM not only captures the intricate patterns in the data but also generalizes well to unseen instances.

Moreover, we employed SHapley Additive exPlanation (SHAP) to interpret the ISM model, shedding light on the key factors influencing autogenous shrinkage and swelling predictions. The analysis highlighted the significant impact of variables such as time, the ratio of aggregate to cement (A/C), the content of Superabsorbent Polymer (SAP), water-to-binder and water-to-cement ratios, cement content, and the amount of silica fume used. These insights provide valuable information for concrete mix design and construction practices.

The comparison with previous research adds valuable insights to the study's conclusions. Overall, the ISM model consistently demonstrates superior performance compared to the previously developed models, namely K-Nearest Neighbors (KNN), Random Forest (RF), Gradient Boosting (GB), and Extreme Gradient Boosting (XGB). The ISM model demonstrates remarkable accuracy with the lowest RMSE and MAE values, indicating its ability to make predictions that closely align with actual values. Furthermore, it achieves the highest coefficient of determination value, signifying its proficiency in explaining a substantial portion of the variance in the data. The ISM model excels in the Overall Index (OI) metric, suggesting that it captures more of the underlying information in the data. Additionally, it exhibits lower 95% uncertainty values, highlighting increased confidence in its predictions.

The limitations of the research can be summarized as follows:

- Limited Generalizability: While the study focuses on predicting autogenous shrinkage of concrete containing superabsorbent polymers (SAP) and additional cementitious materials, its findings may not be readily applicable to other concrete compositions and applications due to various factors. Concrete mix designs can vary significantly based on climate, intended use, and available materials, potentially impacting the predictive accuracy of the models across different contexts.

- Data Quality and Representation: The effectiveness of machine learning models heavily relies on the quality and representativeness of the training data. Although the database used in this study is considered standard for model development and comparison, there could still be limitations regarding data completeness, accuracy, and representativeness. Therefore, incorporating additional data from new experiments in the future is necessary to enhance the robustness of the models.

- Limited Comparison with Alternative Approaches: While the study compares the performance of ensemble convolution-based deep learning models with other machine learning algorithms previously proposed in the literature, it only examines a specific set of models (K-Nearest Neighbors, Random Forest, Gradient Boosting, and Extreme Gradient Boosting). Alternative modeling approaches, such as metaheuristic algorithms, which could potentially offer comparable or superior results, are not included in the analysis. Exploring a broader range of modeling techniques could offer valuable insights into the most effective strategies for predicting concrete shrinkage.

In conclusion, this research demonstrates the efficacy of ensemble convolution-based deep learning models, particularly ISM, in predicting autogenous shrinkage and swelling in cementitious materials. The use of ensemble learning techniques enhances the accuracy and reliability of predictions, making them a valuable tool for engineers and researchers in the field of structural engineering. Furthermore, the model interpretation through SHAP contributes to a deeper understanding of the underlying factors influencing autogenous shrinkage and swelling, paving the way for more informed decision-making in concrete construction and design.

References

[1] Barkhordari, M. S., Tehranizadeh, M., Scott, M. H. "Numerical modelling strategy for predicting the response of reinforced concrete walls using Timoshenko theory", Magazine of Concrete Research, 73(19), pp. 988–1010, 2021.
https://doi.org/10.1680/jmacr.19.00542

[2] Barkhordari, M. S., Jawdhari, A. "Finite Element Analysis of UHPC Beams with Transverse Openings", International Interactive Symposium on Ultra-High Performance Concrete, 3(1), 79, 2023.
https://doi.org/10.21838/uhpc.16692

[3] Kheir, J., Klausen, A., Hammer, T. A., De Meyst, L., Hilloulin, B., Van Tittelboom, K., Loukili, A., De Belie, N. "Early age autogenous shrinkage cracking risk of an ultra-high performance concrete (UPHC) wall: Modelling and experimental results", Engineering Fracture Mechanics, 257, 108024, 2021.
https://doi.org/10.1016/j.engfracmech.2021.108024

[4] Song, Y., Damiani, R. M., Lange, D. A. "Continuous monitoring of the moisture, shrinkage, and carbonation effects on foam concrete performance", Construction and Building Materials, 411, 134185, 2024.
https://doi.org/10.1016/j.conbuildmat.2023.134185

[5] Ballekere Kumarappa, D., Peethamparan, S., Ngami, M. "Autogenous shrinkage of alkali activated slag mortars: Basic mechanisms and mitigation methods", Cement and Concrete Research, 109, pp. 1–9, 2018.
https://doi.org/10.1016/j.cemconres.2018.04.004

[6] Shen, D., Wen, C., Zhu, P., Wu, Y., Yuan, J. "Influence of Barchip fiber on early-age autogenous shrinkage of high strength concrete", Construction and Building Materials, 256, 119223, 2020.
https://doi.org/10.1016/j.conbuildmat.2020.119223

[7] Yang, L., Shi, C., Wu, Z. "Mitigation techniques for autogenous shrinkage of ultra-high-performance concrete – A review", Composites Part B: Engineering, 178, 107456, 2019.
https://doi.org/10.1016/j.compositesb.2019.107456

[8] Aghaee, K., Khayat, K. H. "Effect of shrinkage-mitigating materials on performance of fiber-reinforced concrete – An overview", Construction and Building Materials, 305, 124586, 2021.
https://doi.org/10.1016/j.conbuildmat.2021.124586

[9] Akcay, B., Tasdemir, M. A. "Effects of distribution of lightweight aggregates on internal curing of concrete", Cement and Concrete Composites, 32(8), pp. 611–616, 2010.
https://doi.org/10.1016/j.cemconcomp.2010.07.003

[10] Mechtcherine, V., Wyrzykowski, M., Schröfl, C., Snoeck, D., Lura, P., De Belie, N., Mignon, A., Van Vlierberghe, S., Klemm, A. J., Almeida, F. C. R., Filho, J. R. T., Boshoff, W. P., Reinhardt, H.-W., Igarashi, S.-I. "Application of super absorbent polymers (sap) in concrete construction—update of rilem state-of-the-art report", Materials and Structures, 54(2), 80, 2021.
https://doi.org/10.1617/s11527-021-01668-z

[11] Kang, S.-H., Hong, S.-G., Moon, J. "Shrinkage characteristics of heat-treated ultra-high performance concrete and its mitigation using superabsorbent polymer based internal curing method", Cement and Concrete Composites, 89, pp. 130–138, 2018.
https://doi.org/10.1016/j.cemconcomp.2018.03.003

[12] Shen, D., Shi, H., Tang, X., Ji, Y., Jiang, G. "Effect of internal curing with super absorbent polymers on residual stress development and stress relaxation in restrained concrete ring specimens", Construction and Building Materials, 120, pp. 309–320, 2016.
https://doi.org/10.1016/j.conbuildmat.2016.05.048

[13] Aghaee, K., Khayat, K. H. "Benefits and drawbacks of using multiple shrinkage mitigating strategies on performance of fiber-reinforced mortar", Cement and Concrete Composites, 133, 104714, 2022.
https://doi.org/10.1016/j.cemconcomp.2022.104714

[14] Liu, L., Fang, Z., Huang, Z., Wu, Y. "Solving shrinkage problem of ultra-high-performance concrete by a combined use of expansive agent, super absorbent polymer, and shrinkage-reducing agent", Composites Part B: Engineering, 230, 109503, 2022.
https://doi.org/10.1016/j.compositesb.2021.109503

[15] Chen, L., Fakharian, P., Eidgahee, D. R., Haji, M., Arab, A. M. A., Nouri, Y. "Axial compressive strength predictive models for recycled aggregate concrete filled circular steel tube columns using ANN, GEP, and MLR", Journal of Building Engineering, 77, 107439, 2023.
https://doi.org/10.1016/j.jobe.2023.107439

[16] Armaghani, D. J., Rasekh, H., Asteris, P. G. "An advanced machine learning technique to predict compressive strength of green concrete incorporating waste foundry sand", Computers and Concrete, 33(1), pp. 77–90, 2024.
https://doi.org/10.12989/cac.2024.33.1.077

[17] Barkhordari, M. S., Massone, L. M. "Ensemble techniques and hybrid intelligence algorithms for shear strength prediction of squat reinforced concrete walls", Advances in Computational Design, 8(1), pp. 37–59, 2023.
https://doi.org/10.12989/acd.2023.8.1.039

[18] Hilloulin, B., Tran, V. Q. "Using machine learning techniques for predicting autogenous shrinkage of concrete incorporating superabsorbent polymers and supplementary cementitious materials", Journal of Building Engineering, 49, 104086, 2022.
https://doi.org/10.1016/j.jobe.2022.104086

[19] Tran, V.-L., Thai, D.-K., Kim, S.-E. "Efficient framework for ultimate strength prediction and production-based CO_2 emission optimization of CFST columns using categorical boosting algorithm and moth flame optimization", Composite Structures, 333, 117943, 2024.
https://doi.org/10.1016/j.compstruct.2024.117943

[20] Dong, X., Yu, Z., Cao, W., Shi, Y., Ma, Q. "A survey on ensemble learning", Frontiers of Computer Science, 14(2), pp. 241–258, 2020.
https://doi.org/10.1007/s11704-019-8208-z

[21] Gu, J., Wang, Z., Kuen, J., Ma, L., Shahroudy, A., Shuai, B., Liu, T., Wang, X., Wang, G., Cai, J. "Recent advances in convolutional neural networks", Pattern Recognition, 77, pp. 354–377, 2018.
https://doi.org/10.1016/j.patcog.2017.10.013

[22] Cavaleri, L., Barkhordari, M. S., Repapis, C. C., Armaghani, D. J., Ulrikh, D. V., Asteris, P. G. "Convolution-based ensemble learning algorithms to estimate the bond strength of the corroded reinforced concrete", Construction and Building Materials, 359, 129504, 2022.
https://doi.org/10.1016/j.conbuildmat.2022.129504

[23] Barkhordari, M. S., Armaghani, D. J., Asteris, P. G. "Structural Damage Identification Using Ensemble Deep Convolutional Neural Network Models", Computer Modelling in Engineering & Sciences, 134(2), pp. 835–855, 2023.
https://doi.org/10.32604/cmes.2022.020840

[24] Liu, B., Lu, W., Olofsson, T., Zhuang, X., Rabczuk, T. "Stochastic interpretable machine learning based multiscale modeling in thermal conductivity of polymeric graphene-enhanced composites", Composite Structures, 327, 117601, 2024.
https://doi.org/10.1016/j.compstruct.2023.117601

[25] Lu, S., Koopialipoor, M., Asteris, P. G., Bahri, M., Armaghani, D. J. "A Novel Feature Selection Approach Based on Tree Models for Evaluating the Punching Shear Capacity of Steel Fiber-Reinforced Concrete Flat Slabs", Materials, 13(17), 3902, 2020.
https://doi.org/10.3390/ma13173902

[26] Le, H.-A., Le, D.-A., Le, T.-T., Le, H.-P., Le, T.-H., Hoang, H.-G. T., Nguyen, T.-A. "An extreme gradient boosting approach to estimate the shear strength of FRP reinforced concrete beams", Structures, 45, pp. 1307–1321, 2022.
https://doi.org/10.1016/j.istruc.2022.09.112

[27] Thai, D.-K., Le, D.-N., Doan, Q. H., Pham, T.-H., Nguyen, D.-N. "Classification models for impact damage of fiber reinforced concrete panels using Tree-based learning algorithms", Structures, 53, pp. 119–131, 2023.
https://doi.org/10.1016/j.istruc.2023.04.062

[28] Ning, C.-L., Wang, M., Yu, X. "Failure-mode–independent prediction model for the peak strength of reinforced concrete columns using Bayesian neural network: A probabilistic approach", Advances in Structural Engineering, 25(9), pp. 1923–1942, 2022.
https://doi.org/10.1177/13694332221081187

[29] Amin, M. N., Iftikhar, B., Khan, K., Javed, M. F., AbuArab, A. M., Rehman, M. F. "Prediction model for rice husk ash concrete using AI approach: Boosting and bagging algorithms", Structures, 50, pp. 745–757, 2023.
https://doi.org/10.1016/j.istruc.2023.02.080

[30] Bisong, E., Bisong, E. "Introduction to scikit-learn", In: Building Machine Learning and Deep Learning Models on Google Cloud Platform: A Comprehensive Guide for Beginners, Apress, 2019, pp. 215–229. ISBN 978-1-4842-4470-8
https://doi.org/10.1007/978-1-4842-4470-8_18

[31] Lundberg, S. M., Lee, S.-I. "A unified approach to interpreting model predictions", [preprint] arXiv, arXiv:1705.07874, 25 November 2017.
https://doi.org/10.48550/arXiv.1705.07874

[32] Ahmed, A., Uddin, M. N., Akbar, M., Salih, R., Khan, M. A., Bisheh, H., Rabczuk, T. "Prediction of shear behavior of glass FRP bars-reinforced ultra-highperformance concrete I-shaped beams using machine learning", International Journal of Mechanics and Materials in Design, 20(2), pp. 269–290, 2024.
https://doi.org/10.1007/s10999-023-09675-4

[33] Mao, Y., Liu, J., Shi, C. "Autogenous shrinkage and drying shrinkage of recycled aggregate concrete: A review", Journal of Cleaner Production, 295, 126435, 2021.
https://doi.org/10.1016/j.jclepro.2021.126435

[34] Dey, A., Vastrad, A. V., Bado, M. F., Sokolov, A. Kaklauskas, G. "Long-term concrete shrinkage influence on the performance of reinforced concrete structures", Materials, 14(2), 254, 2021.
https://doi.org/10.3390/ma14020254

[35] Tang, S., Huang, D., He, Z. "A review of autogenous shrinkage models of concrete", Journal of Building Engineering, 44, 103412, 2021.
https://doi.org/10.1016/j.jobe.2021.103412

[36] Zhan, P.-m., He, Z.-h. "Application of shrinkage reducing admixture in concrete: A review", Construction and Building Materials, 201, pp. 676–690, 2019.
https://doi.org/10.1016/j.conbuildmat.2018.12.209

[37] Dogan, G. "Machine learning-based shear strength prediction of exterior RC beam-column joints", Multiscale and Multidisciplinary Modeling, Experiments and Design, 2024.
https://doi.org/10.1007/s41939-023-00338-7

[38] Ghanizadeh, A. R., Delaram, A., Fakharian, P., Armaghani, D. J. "Developing Predictive Models of Collapse Settlement and Coefficient of Stress Release of Sandy-Gravel Soil via Evolutionary Polynomial Regression", Applied Sciences, 12(19), 9986, 2022.
https://doi.org/10.3390/app12199986

[39] Isleem, H. F., Chukka, N. D. K. R., Bahrami, A., Kumar, R., Sor, N. H. "Nonlinear finite element and machine learning modeling of tubed reinforced concrete columns under eccentric axial compression loading", Alexandria Engineering Journal, 92, pp. 380–416, 2024.
https://doi.org/10.1016/j.aej.2024.02.026

[40] Thai, D.-K., Le, D.-N., Doan, Q. H., Pham, T.-H., Nguyen, D.-N. "A hybrid model for classifying the impact damage modes of fiber reinforced concrete panels based on xgboost and horse herd optimization algorithm", Structures, 60, 105872, 2024.
https://doi.org/10.1016/j.istruc.2024.105872

Appendix A

input_3	input:	[(None, 14, 1)]
InputLayer	output:	[(None, 14, 1)]

conv1d_24	input:	(None, 14, 1)
Conv1D	output:	(None, 14, 16)

conv1d_20	input:	(None, 14, 1)
Conv1D	output:	(None, 14, 32)

conv1d_25	input:	(None, 14, 16)
Conv1D	output:	(None, 14, 16)

conv1d_21	input:	(None, 14, 32)
Conv1D	output:	(None, 14, 32)

conv1d_26	input:	(None, 14, 16)
Conv1D	output:	(None, 14, 16)

conv1d_22	input:	(None, 14, 32)
Conv1D	output:	(None, 14, 32)

conv1d_27	input:	(None, 14, 16)
Conv1D	output:	(None, 14, 16)

dropout_4	input:	(None, 14, 32)
Dropout	output:	(None, 14, 32)

concatenate_4	input:	[(None, 14, 16), (None, 14, 16)]
Concatenate	output:	(None, 14, 32)

dropout_5	input:	(None, 14, 32)
Dropout	output:	(None, 14, 32)

conv1d_23	input:	(None, 14, 32)
Conv1D	output:	(None, 14, 16)

concatenate_5	input:	[(None, 14, 32), (None, 14, 16)]
Concatenate	output:	(None, 14, 48)

conv1d_28	input:	(None, 14, 48)
Conv1D	output:	(None, 14, 25)

flatten_3	input:	(None, 14, 25)
Flatten	output:	(None, 350)

dense_9	input:	(None, 350)
Dense	output:	(None, 60)

dense_11	input:	(None, 60)
Dense	output:	(None, 15)

dense_12	input:	(None, 15)
Dense	output:	(None, 1)

Fig. A1 Structures and architecture of base learner 1

conv1d_29_input	input:	[(None, 14, 1)]
InputLayer	output:	[(None, 14, 1)]

conv1d_29	input:	(None, 14, 1)
Conv1D	output:	(None, 14, 32)

conv1d_30	input:	(None, 14, 32)
Conv1D	output:	(None, 14, 32)

conv1d_31	input:	(None, 14, 32)
Conv1D	output:	(None, 14, 32)

conv1d_32	input:	(None, 14, 32)
Conv1D	output:	(None, 13, 32)

flatten_4	input:	(None, 13, 32)
Flatten	output:	(None, 416)

dense_13	input:	(None, 416)
Dense	output:	(None, 100)

dense_14	input:	(None, 100)
Dense	output:	(None, 20)

dense_15	input:	(None, 20)
Dense	output:	(None, 1)

Fig. A2 Structures and architecture of base learner 2

Fig. A3 Structures and architecture of base learner 3

input_5	input:	[(None, 14, 1)]
InputLayer	output:	[(None, 14, 1)]

conv1d_38	input:	(None, 14, 1)
Conv1D	output:	(None, 14, 16)

conv1d_34	input:	(None, 14, 1)
Conv1D	output:	(None, 14, 16)

conv1d_39	input:	(None, 14, 16)
Conv1D	output:	(None, 14, 16)

conv1d_35	input:	(None, 14, 16)
Conv1D	output:	(None, 14, 16)

conv1d_40	input:	(None, 14, 16)
Conv1D	output:	(None, 14, 16)

conv1d_36	input:	(None, 14, 16)
Conv1D	output:	(None, 14, 16)

conv1d_41	input:	(None, 14, 16)
Conv1D	output:	(None, 14, 16)

dropout_2	input:	(None, 14, 16)
Dropout	output:	(None, 14, 16)

concatenate_12	input:	[(None, 14, 16), (None, 14, 16)]
Concatenate	output:	(None, 14, 32)

conv1d_37	input:	(None, 14, 16)
Conv1D	output:	(None, 14, 16)

dropout_3	input:	(None, 14, 32)
Dropout	output:	(None, 14, 32)

concatenate_11	input:	[(None, 14, 16), (None, 14, 16)]
Concatenate	output:	(None, 14, 32)

concatenate_13	input:	[(None, 14, 32), (None, 14, 32)]
Concatenate	output:	(None, 14, 64)

conv1d_42	input:	(None, 14, 64)
Conv1D	output:	(None, 14, 40)

flatten_4	input:	(None, 14, 40)
Flatten	output:	(None, 560)

dense_11	input:	(None, 560)
Dense	output:	(None, 60)

dense_12	input:	(None, 60)
Dense	output:	(None, 15)

dense_13	input:	(None, 15)
Dense	output:	(None, 1)

Fig. A4 Structures and architecture of base learner 4

input_6	input:	[(None, 14, 1)]
InputLayer	output:	[(None, 14, 1)]

conv1d_51	input:	(None, 14, 1)
Conv1D	output:	(None, 14, 32)

conv1d_49	input:	(None, 14, 1)
Conv1D	output:	(None, 14, 32)

conv1d_52	input:	(None, 14, 32)
Conv1D	output:	(None, 14, 32)

conv1d_50	input:	(None, 14, 32)
Conv1D	output:	(None, 14, 16)

dropout_11	input:	(None, 14, 32)
Dropout	output:	(None, 14, 32)

conv1d_53	input:	(None, 14, 32)
Conv1D	output:	(None, 14, 32)

dropout_10	input:	(None, 14, 16)
Dropout	output:	(None, 14, 16)

concatenate_11	input:	[(None, 14, 16), (None, 14, 32)]
Concatenate	output:	(None, 14, 48)

flatten_7	input:	(None, 14, 48)
Flatten	output:	(None, 672)

dense_21	input:	(None, 672)
Dense	output:	(None, 50)

dropout_12	input:	(None, 50)
Dropout	output:	(None, 50)

dense_22	input:	(None, 50)
Dense	output:	(None, 20)

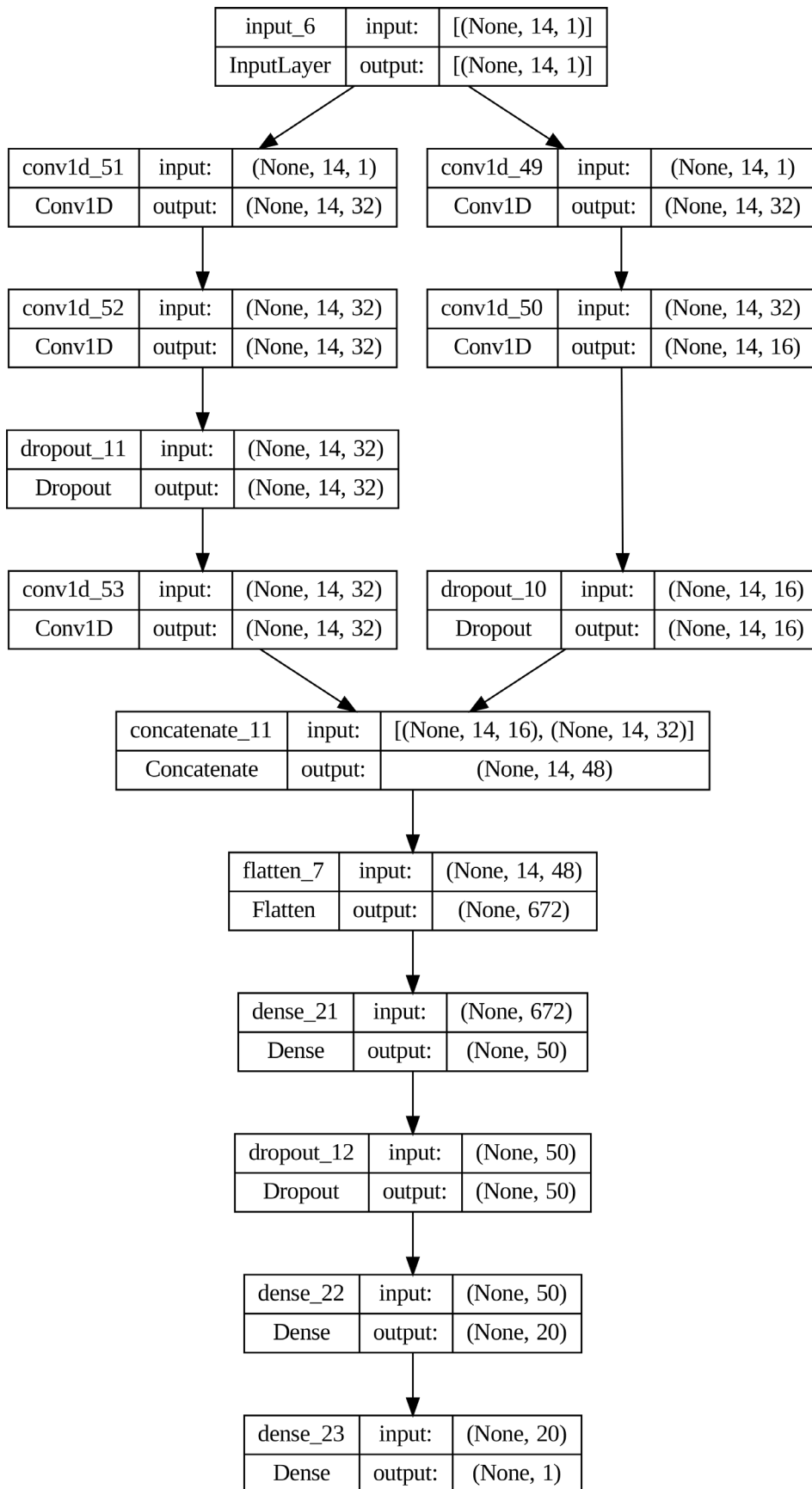| dense_23	input:	(None, 20)
Dense	output:	(None, 1)

Fig. A5 Structures and architecture of base learner 5