Barkhordari and Massone
Int J Concr Struct Mater (2022) 16:33
https://doi.org/10.1186/s40069-022-00522-y

International Journal of Concrete Structures and Materials

RESEARCH Open Access

Failure Mode Detection of Reinforced Concrete Shear Walls Using Ensemble Deep Neural Networks

Mohammad Sadegh Barkhordari[1] and Leonardo M. Massone[2*]

Abstract

Reinforced concrete structural walls (RCSWs) are one of the most efficient lateral force-resisting systems used in buildings, providing sufficient strength, stiffness, and deformation capacities to withstand the forces generated during earthquake ground motions. Identifying the failure mode of the RCSWs is a critical task that can assist engineers and designers in choosing appropriate retrofitting solutions. This study evaluates the efficiency of three ensemble deep neural network models, including the model averaging ensemble, weighted average ensemble, and integrated stacking ensemble for predicting the failure mode of the RCSWs. The ensemble deep neural network models are compared against previous studies that used traditional well-known ensemble models (AdaBoost, XGBoost, LightGBM, CatBoost) and traditional machine learning methods (Naïve Bayes, K-Nearest Neighbors, Decision Tree, and Random Forest). The weighted average ensemble model is proposed as the best-suited prediction model for identifying the failure mode since it has the highest accuracy, precision, and recall among the alternative models. In addition, since complex and advanced machine learning-based models are commonly referred to as black-box, the SHapley Additive exPlanation method is also used to interpret the model workflow and illustrate the importance and contribution of the components that impact determining the failure mode of the RCSWs.

Keywords: deep neural network, failure mode, shear wall, classification

1 Introduction

In building design, shear walls are commonly employed to protect the structure from lateral forces such as earthquake ground motions. Shear walls are a cost-effective way to reinforce a building's structural system against lateral loads. A reinforced concrete structural wall (RCSW) improves the building's stiffness in the wall plane, lowering the building's lateral sway and boosting its stability in that plane. For this reason, this system is widely used in buildings. Existing RCSW buildings are assessed and retrofitted following local jurisdictional regulations and it is critical to predict the severe damage and failure modes of RCSWs accurately. According to the aspect ratio (the ratio of height to the wall length), RCSWs are classified as slender or squat (Barkhordari et al., 2021; Massone et al., 2021). Slender walls (aspect ratio > 3) are more prone to ductile failure characterized by bar buckling and concrete crushing, bar fracture, or global or local lateral instabilities. Squat walls (aspect ratio < 1.5) are prone to have shear-controlled failure mechanisms, which can be characterized by diagonal tension, diagonal compression (web crushing), or shear sliding at the base. Walls with an aspect ratio of 1.5 to 3.0 (moderate-aspect-ratio walls) display behavior that is characterized by yielding in flexure and failing in shear.

In recent years, researches on the application of machine learning models for damage estimation and monitoring in civil engineering have recently been

*Correspondence: lmassone@ing.uchile.cl
[2] Department of Civil Engineering, University of Chile, Blanco Encalada 2002, Santiago, Chile
Full list of author information is available at the end of the article
Journal information: ISSN 1976-0485 / eISSN 2234-1315

published. They can be categorized into two main types, including: (1) regression-based methods and (2) classification-based methods. Some of the previous studies used various machine learning algorithms to estimate the performance of shear walls. Moradi et. al. (2020) studied the application of the radial basis function network to assess the impacts of the rectangular opening on the behavior of the steel plate shear walls. They suggested the proposed network that can be used to design new walls or retrofit existing ones while consuming less time and requiring no specific software knowledge. Using the extreme gradient boosting (XGBoost) technique, Feng et. al. (2021) developed a forecasting model for predicting the shear strength of reinforced concrete squat walls. According to studies, the XGBoost model provides a decent prediction for shear strength, with an average computed-to-measured ratio of 1.0. Chen et. al. (2018) and Nguyen et. al. (2021) utilized neural networks for shear strength prediction of reinforced concrete squat walls. Gondia et. al. (2020) used genetic programming, a kind of artificial intelligence, to develop an expression for the shear strength of reinforced concrete squat walls. Keshtegar et. al. (2021a, 2021b) used neural network merged with adaptive harmony search algorithm and support vector regression with response surface model to estimate the ultimate shear capacity of RCSWs. Pizarro et. al. (2021) and Pizarro and Massone (2021) developed a convolutional network-based solution to produce the ultimate engineering floor plan of reinforced shear wall concrete buildings using a dataset of 165 Chilean residential layouts. Barkhordari and Tehranizadeh (2021) developed a hybrid technique, the neural network with simulated annealing, for predicting the response of the RCSWs, such as forces and bending moment at the base of the wall, curvature of the wall, and normal strain in the vertical/horizontal direction. Parsa and Naderpour (2021) used the support vector regression with meta-heuristic optimization algorithms for the estimation of the shear strength of the RCSWs. Mangalathu et. al. (2020) investigated the effectiveness of eight machine learning techniques in detecting RCSW failure mechanisms. To sum up, only Mangalathu et. al. (2020) examined the efficiency of the ensemble learning algorithms, except ensemble deep neural network algorithms, in predicting the failure mode of the RCSWs.

Even though the aspect ratio marks tendencies of general behavior, several factors affect the failure mode of the RCSWs. In addition, appropriate methodologies for assessing their likely mechanism of failure during earthquake occurrences are required to acquire a better knowledge of the seismic behavior of existing structures and to create appropriate retrofit solutions. In addition, although there have been attempts to predict the failure mode with different machine learning approaches (Mangalathu et al., 2020), no previous research has used ensemble deep neural network models to predict the failure mechanism of the RCSWs. In this study, model averaging, weighted average, and integrated stacking are three ensemble learning approaches that are employed for failure mode detection of the RCSWs. Moreover, the relevance of input variables such as aspect ratio (or more general, moment-to-shear length ratio), steel ratio, concrete strength, axial load, among others, has not been studied. The aim here is to evaluate the efficiency of the ensemble deep neural network models for assessing the failure mechanism of the RCSWs and investigate the relevance of the main variables and how they relate to failure modes.

2 Method and Material

Here, Keras (Chollet 2015) is used to develop deep learning models, which is an open-source library for artificial neural networks that supports a Python interface. A brief description of the data used in this study, various parts of the base neural networks, and ensemble algorithms are provided below.

2.1 Database

Test data generated by experimental studies on RCSWs, where specimens are tested using cyclic loading protocols, is used in this study, 393 experimental data of RCSWs tested that were collected from the literature by Mangalathu, Jang (Grammatikou et al., 2015; Mangalathu et al., 2020; Usta et al., 2017). It is worth noting that the RC walls in this repository are traditional RCSWs; they do not contain testing of repaired or precast RCSWs, nor do they include walls that have been subjected to dynamic loading. RCSWs have a distribution of 3 types of cross-section configurations: 238 rectangular walls, 95 barbell walls, and 60 flanged walls (Fig. 1). The distribution of the failure mode (output) of RCSWs is shown in Table 1. Design parameters are listed in Table 2. In Table 2, P is the axial load, A_g is the gross area of the section, f_c is the compressive strength of concrete, A_b is the boundary element area, $\rho_{i,x/y}$ is the reinforcing ratio in the horizontal/vertical direction, t_w is the wall thickness, H is the height of the wall, l_w is the length of the wall, and f_y is the yield strength of the reinforcement. Due to insufficient research funds and scientific equipment capacities, many experimental investigations have been undertaken with small-scale specimens. The use of dimensionless values for the input variables is desirable to estimate the failure of the shear walls. Therefore, parameters are normalized to use dimensionless variables. The input variables are M/Vl_w, A_b/A_g, l_w/t_w, P/f_cA_g, $\rho_{vb}f_{y,vb}/f_c$, $\rho_{vw}f_{y,vw}/f_c$, $\rho_{hb}f_{y,hb}/f_c$, and $\rho_{hw}f_{y,hw}/f_c$. Table 3 shows the statistical

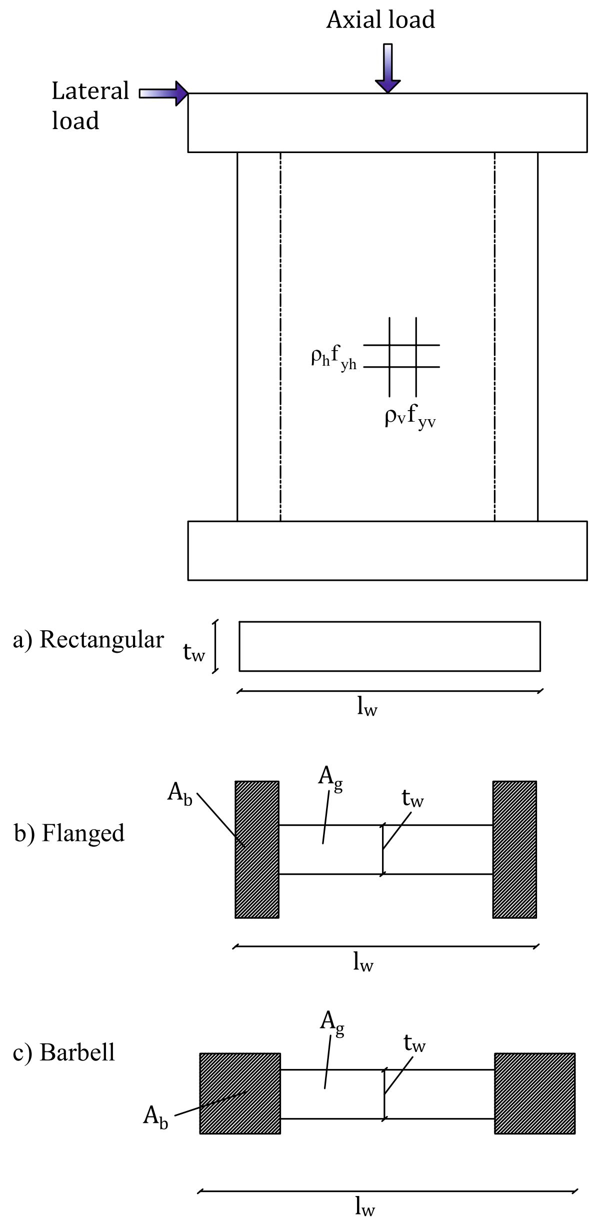


Fig. 1 Cross-sections of the reinforced concrete walls.

properties of the input variables, where Min., Max., and STD are the minimum, maximum, and standard deviation of variables, respectively. All 393 datasets are split randomly into training and testing sets. 80% of the total of 393 datasets is utilized randomly for model development and 20% of the datasets are chosen to determine the model accuracy. Input data are normalized so that all values are within the range of − 1 and 1. The limitation of the data is that the target class has an uneven distribution of observations.

Fig. 2 provides a stacked bar for types of walls/section vs. failure mode. In Fig. 2, F, FS, S, and SL stand for 'flexural failure', 'flexure–shear failure', 'shear failure', and 'sliding shear failure'. Fig. 2a shows the distribution of types of walls vs. failure mode. It is clear that the number of failure modes is different for different types of walls. The graph shows that all slender walls have only flexural failure. Furthermore, there is some variation in the graph. For instance, there is a large growth in the number of walls with shear failure when comparing the moderate sector with the squat sector. Fig. 2b shows the distribution of section vs. failure mode. It is apparent from this graph that rectangular walls mostly have flexural failure. In addition, flanged walls mostly have shear failure. This is most likely because when walls have the boundary element, their flexural strength increases. As a result, shear failure may occur before flexural failure under lateral loading if the lateral force relating to flexural strength is greater than the lateral force corresponding to shear strength.

2.2 Basic Models

For almost all ensemble methods, a series of models must first be created as basic models (or sub-model) to form an ensemble model. This means that several models are taught using training data. The baseline models are constructed of five different deep neural networks. Models in Keras are defined as a sequence of layers. Each layer has some nodes (neurons). When generating deep neural networks, one of the most common questions is what should be the number of neurons per layer? Here, the learning rate, activation functions, optimizer, and the number of neurons per layer are determined using the Keras-Tuner library that helps to pick the optimal set of hyperparameters for deep neural networks. Fig. 3 is a simplified form of the workflow. Here, six activation functions (Sigmoid, Relu, Softplus, Tanh, Selu, and Elu) are considered.

Table 1 Distribution of failure mode.

The number of specimens			
Flexure failure modes (i.e., bar buckling and concrete crushing, bar fracture, or global or local lateral instability)	Shear failure modes [i.e., diagonal tension, diagonal compression (web crushing)]	Flexure–shear failure (i.e., yielding in flexure and failing in shear)	Sliding-shear failure
152	122	96	23

Table 2 Statistical features of database.

Aspect ratio (M/Vl_w)	0.25–4.1	Axial load ratio (P/f_cA_g)	0.0–0.5
Boundary element area to area of cross-section ratio (A_b/A_g)	0.0–0.44	Gross cross-sectional area (A_g—mm^2)	7800–825,400
The ratio of length to thickness of the wall (l_w/t_w)	4.35–57	Compressive strength of concrete (f_c—MPa)	13.7–130.8
Web reinforcing ratio in horizontal direction (ρ_{hw})	0.0–0.037	Web reinforcing ratio in vertical direction (ρ_{vw})	0.0–0.037
B. E.* reinforcing ratio in horizontal direction (ρ_{hb})	0.0–0.13	B. E. reinforcing ratio in vertical direction (ρ_{vb})	0.0–0.1
Yield strength of horizontal reinforcement (MPa)	Web ($f_{y,hw}$)	0.0–1262	
	B. E. ($f_{y,hb}$)	0.0–1253	
Yield strength of vertical reinforcement (MPa)	Web ($f_{y,vw}$)	0.0–1001	
	B. E. ($f_{y,vb}$)	0.0–776	

*B. E. = boundary element.

Table 3 Statistical characteristics of the input variables.

	M/Vl_w	l_w/t_w	$\rho_{vw}f_{y,vw}/f_c$	$\rho_{hw}f_{y,hw}/f_c$	$\rho_{vb}f_{y,vb}/f_c$	$\rho_{hb}f_{y,hb}/f_c$	P/f_cA_g	A_b/A_g
Mean	1.39	14.7	0.0829	0.0796	0.384	0.128	0.067	0.098
STD	0.71	8.7	0.0731	0.0600	0.374	0.210	0.105	0.127
Min	0.25	4.3	0	0	0	0	0	0
Max	4.1	37	0.488	0.327	2.65	1.26	0.50	0.443

The optimizer is a relevant component of the training phase. The optimizer function assists the network in determining how to update the weights to minimize the loss. Eight optimization algorithms are utilized namely adaptive gradient (Adagrad) (Duchi et al., 2011), Adaptive delta (Adadelta) (Zeiler & Adadelta, 2012), Stochastic Gradient Descent (SGD) (Krogh & Hertz, 1992), Root Mean Square Prop (RMSProp) (Zeiler & Adadelta, 2012), Adaptive Moment Estimation (Adam) (Kingma & Ba, 2014), Adamax (a variant of Adam based on infinity norm) (Kingma & Ba, 2014), Nadam (Adam with Nesterov momentum) (Dozat, 2016), and Follow-the-regularized-leader (Ftrl) (McMahan et al., 2013). In this study, the basic input parameters contain M/Vl_w, A_b/A_g, l_w/t_w,$\rho_{vb}F_{y,vb}/f_c$, $\rho_{hb}F_{y,hb}/f_c$, P/f_cA_g, $\rho_{hw}F_{y,hw}/f_c$, and $\rho_{vw}F_{y,vw}/f_c$. The output of the model is predicted failure mode class, including 'flexural failure', 'flexure–shear failure', 'shear failure', and 'sliding shear failure'.

Among all, five basic models are selected in this study based on their performance on test data. Table 4 summarizes the information of the final basic models (submodel) that had the highest accuracy. Overfitting occurs when a model learns the knowledge and noise in the training set to the degree where it degrades the model's performance on hold-out data. Sometimes in ensemble techniques, the sub-models may have overfitting problems. In this study, the learning curve of all models was monitored to ensure that overfitting did not occur. In this study, the learning curve (e.g., Fig. 4) of all sub-models is monitored to ensure that overfitting did not occur.

2.3 Model Averaging Ensemble (MAE)

MAE is an ensemble method that involves training many models on the same data set (Brownlee, 2018). The outputs from each of the trained models are then added together, and the average is used as the final predicted value. The number of models needed for the ensemble can vary depending on the solution space complexity. One technique is to construct new models on a constant schedule (increasing the number of layers), add them to the group, and then assess their contribution to performance by predicting on a test set. Fig. 5 shows how the MAE method works using submodels (Table 4).

2.4 Weighted Average Ensemble (WAE)

The averaging performed in the MAE method means that the output values of the sub-models have an equal effect on the predicted final value (Brownlee, 2018). The WAE approach makes it possible for superior models to have a larger share of the predicted final value, while less talented models have a smaller share. In this method, a weight is assigned to the output of each subset model. The value of these weights is usually determined by an optimization algorithm.

Metaheuristic search algorithms are divided into three categories (Ahmadianfar et al., 2021), named by swarm-based algorithms, evolutionary-based algorithms, and trajectory-based algorithms. Evolutionary algorithms are developed mostly from Darwin's theory of urgency and natural selection. Differential evolution

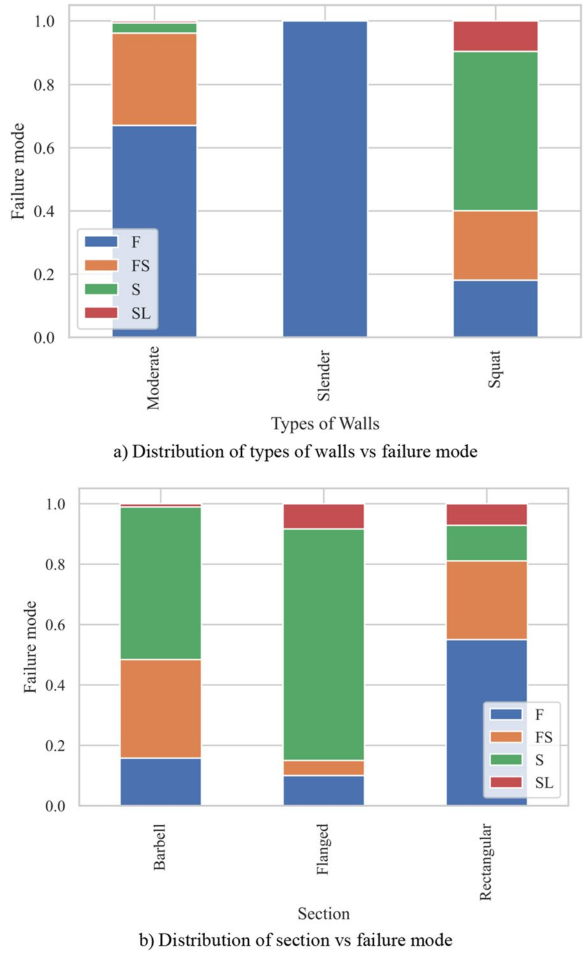


Fig. 2 Distribution of section/types of walls vs. failure mode. **a** Distribution of types of walls vs. failure mode. **b** Distribution of section vs. failure mode.

(DE) is a kind of evolutionary algorithm (Bennis & Bhattacharjya, 2020) that initializes the members of the algorithm with several potential solutions at the start. The differential evolution iterative technique is then continued by applying the difference vector based on the DE operators, including mutation, crossover, and the selection mechanism. Following that, each solution is evaluated using a specified objective function in an iterative optimization process. The mutation process aims to vary the population member vector for the next iteration based on any available information from the previous step in the search (Eq. 1):

$$\vec{y}_i(t+1) = m_\mathrm{r} \cdot \left(x_{2,j}(t) - x_{3,j}(t)\right) + x_{1,j}(t), \tag{1}$$

where $\vec{y}_i$ is the mutant vector (MV) of the ith member, $x_{1,j}$, $x_{2,j}$, and $x_{3,j}$ are chosen at random from the population, and m_r is the scaling factor that fine-tunes the size of perturbation in the process. The crossover operator is utilized to broaden the population genetic variation among the MVs. As a result, the MV replaces its elements with those of the current population. The selection mechanism is used to determine which of the offspring individuals (and their parent) will survive in the next cycles, as well as to keep the pre-determined population size constant. The population is built using the individuals, which are selected between the trial and their predecessor vectors, which have better performance in terms of the objective function. Here, differential evolution is used to calculate the weight of each sub-model because of its advantages, such as discovering the global minimum of a search space independent of the initial values, fast convergence, and the usage of a few control factors (Karaboga & Cetinkaya, 2004). Fig. 6 shows the flowchart of the WAE method with the differential evolution algorithm.

2.5 Integrated Stacking Ensemble (ISE)

Although the average of the model can be improved by weighting the influences of each sub-model, it may be further improved by teaching a completely new model (a neural network) to discover how to better combine each of the sub-models, using the so-called Integrated Stacking Model (ISE) (Brownlee, 2018; Naimi & Balzer, 2018). The new model is usually called meta-learner, where the sub-models are integrated with a neural network. In other words, the ISE can be interpreted as a single huge model which then learns how to merge the results from every single sub-model in the most efficient way possible. Here, the architecture of the meta-learner consists of only one hidden layer with 5 neurons. The process of determining the hidden layer's neurons is just a case of trial and error (after examining the range of 2 to 20 neurons, the best performance of the ISE model with at least 5 neurons is obtained). Fig. 7 represents a diagram to understand the ISE model process.

3 Results and Discussion

For the MAE method, the number of members can change the result. Therefore, the influence of the number of sub-models on the model's accuracy is explored, and the best model with the minimum members is chosen. Fig. 8 shows the effect of the number of members versus the accuracy. Increasing the size of the ensemble model (adding sub-model) has been done by first creating a new model with the first two sub-models, that is, sub-model 1 and sub-model 2 from Table 4, and for each subsequent ensemble model another sub-model is added to the previous group, examining the accuracy of the ensemble model on the test set. It can be seen that from

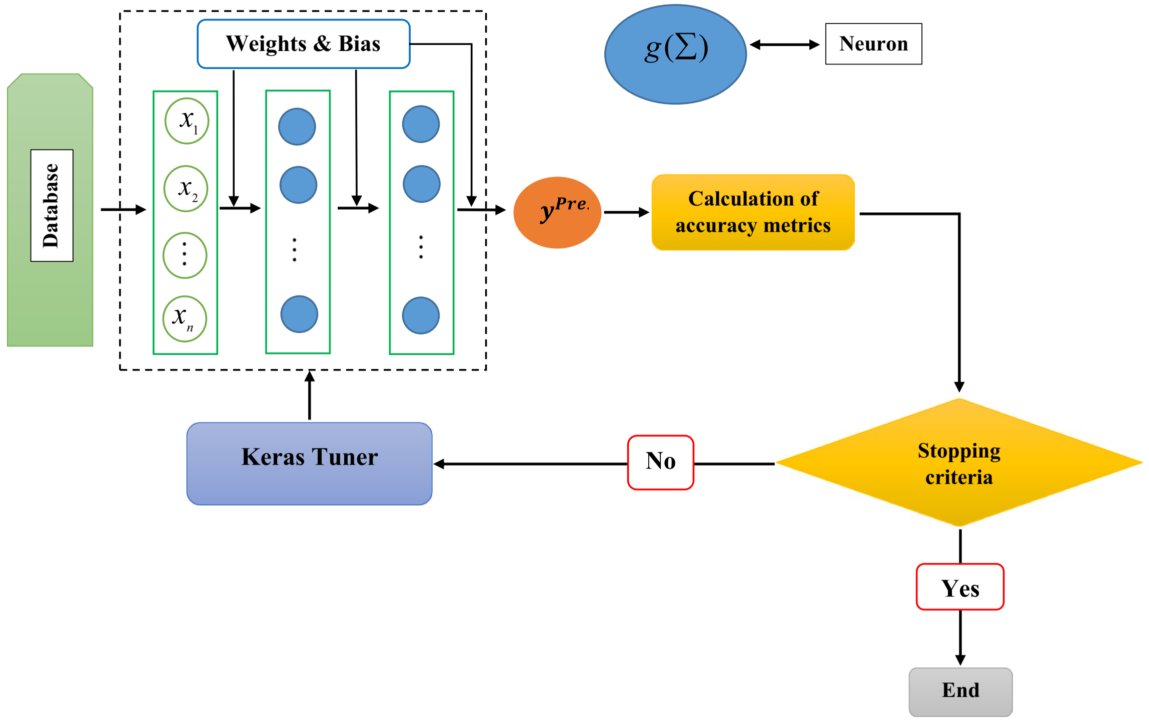


Fig. 3 Model workflow.

Table 4 Characteristics of the basic models.

Sub-model	1	2	3	4	5
Number of neurons					
Layer 1	40	25	40	25	20
Drop rate	0.02	0.01	0.01	–	–
Layer 2	80	80	60	20	50
Drop rate	–	–	0.01	–	–
Layer 3	–	20	60	80	30
Layer 4	–	–	15	80	60
Layer 5	–	–	–	40	80
Layer 6	–	–	–	–	25
Activation	Tanh	Tanh	Tanh	Tanh	Tanh
Optimizer	Adam	Adam	Adam	Adam	Adam
Learning rate	0.01	0.01	0.01	0.01	0.01
R^2 score (test set)	0.83	0.81	0.82	0.84	0.81

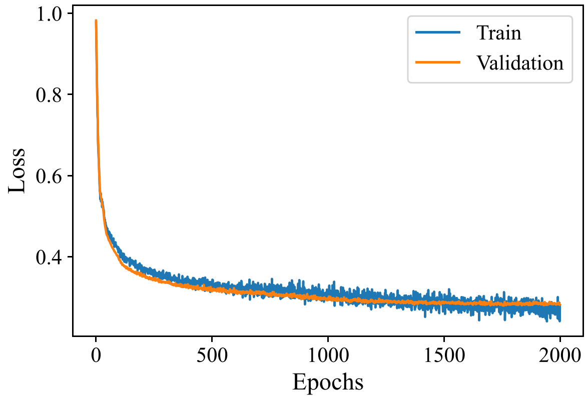


Fig. 4 Learning curve of sub-model 1.

1 to 2 sub-models, there is a marked rise in the ensemble model's accuracy. From 2 to 3 sub-models, there is a modest rise in the ensemble model's accuracy. This is followed by a constant accuracy for models with more than three members. As a result, a model with three members (sub-models 1–3) is selected for this method.

As mentioned, the WAE model permits higher-performing models to have a bigger proportion, while lower-performing models have a lesser share by assigning a weight to the sub-models' output. Table 5 shows optimized weights, which are determined using the differential evolution algorithm. The WAE models' accuracy with the optimized weights is 0.987.

The last ensemble model is the ISE model. The accuracy of the ISE model also is 0.962. Considering the stochastic nature of neural networks' learning algorithm, it's possible that each time a neural network model is trained, it will

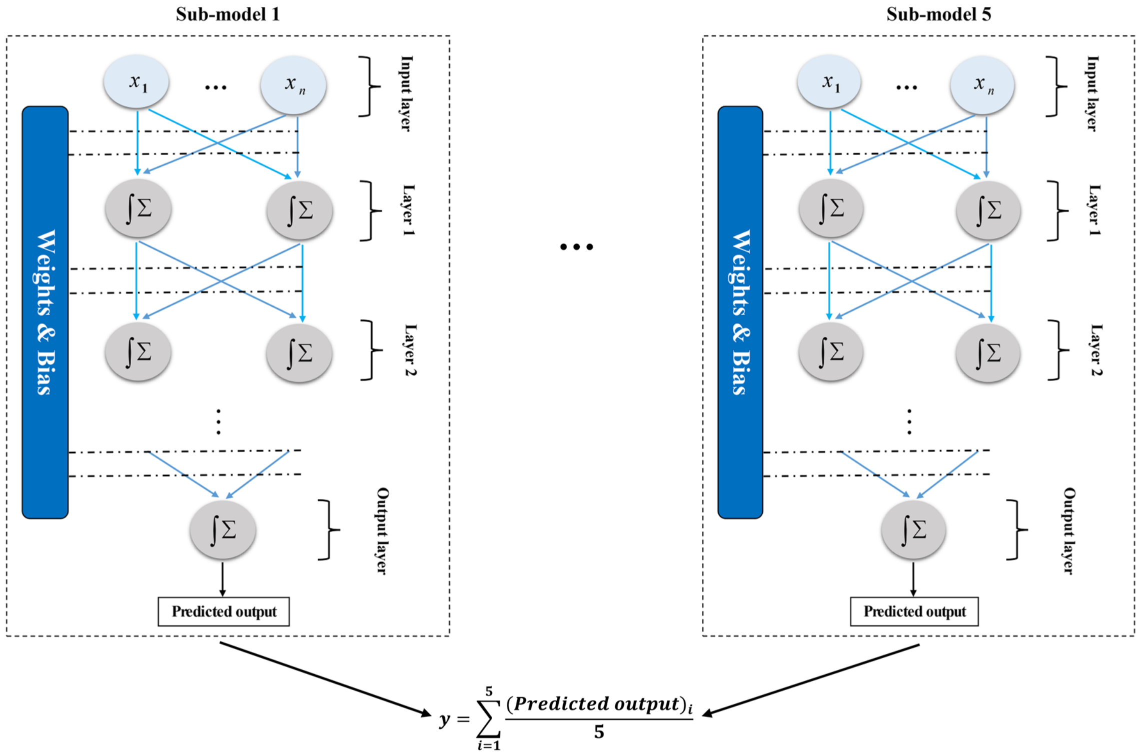


Fig. 5 Model averaging ensemble framework.

discover a mild/significantly various version of the mapping function between inputs and outputs, that is, neural networks have a high variance, resulting in differences in performance on the training and test sets. WAE models work virtually well in most situations because different neural networks do not always produce the same errors on the test set (Goodfellow et al., 2017). In addition, submodels have different numbers of layers and neurons in each layer. Change in the number of layers helps to consider the various levels of nonlinearity. The number of neurons in the layer is also important since they consider the interaction between the parameters. This means that by increasing the number of neurons and creating more relationships, if these relationships are not appropriate, it may reduce the efficiency of the neural network and the accuracy of network prediction and vice versa.

ISE model performs worse than WAE. This could be due to the local minima. The feed-forward neural network, which is trained using backpropagation has a variety of drawbacks, such as falling into local minima and learning at a slow rate (Lee et al., 1991).

Fig. 9 shows the confusion matrix of various models. In Fig. 9, F, FS, S, and SL stand for 'flexural failure', 'flexure–shear failure', 'shear failure', and 'sliding shear failure'. The failure modes successfully identified by the classification algorithm are represented by the diagonal cells in the confusion matrix, whereas the failure modes incorrectly predicted are represented by the off-diagonal cells. The lowest cell on the right of the figure shows the overall accuracy (Eq. 2). The column on the far right of the confusion matrix indicates the precision metric (Eq. 3). The row at the bottom of the confusion matrix indicates the recall metric (Eq. 4):

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{FP} + \text{FN} + \text{TN}}, \tag{2}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}, \tag{3}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}, \tag{4}$$

where TN indicates that the model predicted 'False' and the real outcome was 'False', FP denotes that the model predicted 'True' but the real outcome was 'False', FN means that the real outcome was 'False', and the model predicted 'True', and TP denotes that the model predicted 'True' and the real outcome was 'True'. Among the ensemble algorithms, the WAE model fares much better. The WAE model has the highest accuracy

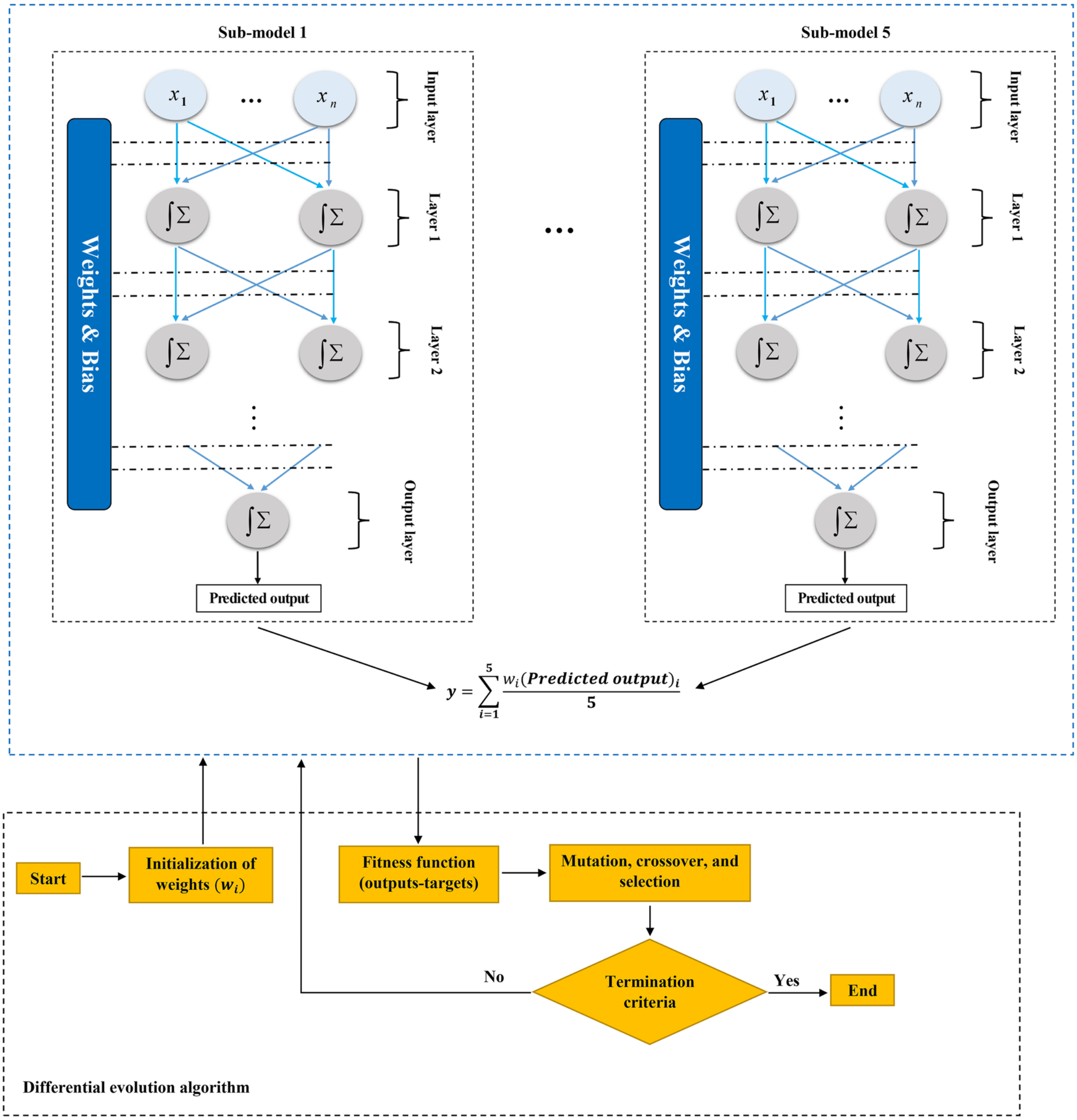

Sub-model 1
Sub-model 5
Weights & Bias
x_1
x_n
Input layer
Layer 1
Layer 2
Output layer
$\int\Sigma$
Predicted output
$y = \sum_{i=1}^{5} \frac{w_i(Predicted\ output)_i}{5}$
Start
Initialization of weights (w_i)
Fitness function (outputs-targets)
Mutation, crossover, and selection
Termination criteria
No
Yes
End
Differential evolution algorithm

Fig. 6 Flowchart of the WAE method with differential evolution algorithm.

with 0.987 for the test set, whereas the MAE and ISE models have the same accuracy (0.962). It appears that recognizing the FS failure mode is often challenging, and the MAE model, WAE model, and ISE model all have 0.88, 0.94, and 0.84 precision in identifying the FS failure mode in the test set, respectively.

3.1 Comparisons with Previous Studies

Mangalathua et. al. (2020) used eight machine learning models to establish a model to distinguish the failure mode of the RCSWs. The following is a list of machine learning models that Mangalathu et. al. (2020) utilized to determine the failure modes of concrete shear walls.

1. Naive Bayes classifier (Domingos & Pazzani, 1997; Osisanwo et al., 2017): A probabilistic machine learning technique called a Naive Bayes classifier is utilized to perform classification tasks. The Bayes theorem lies at the core of the classifier. It is one of the most basic Bayesian network models, but when combined with kernel density estimation, it can reach higher precision.

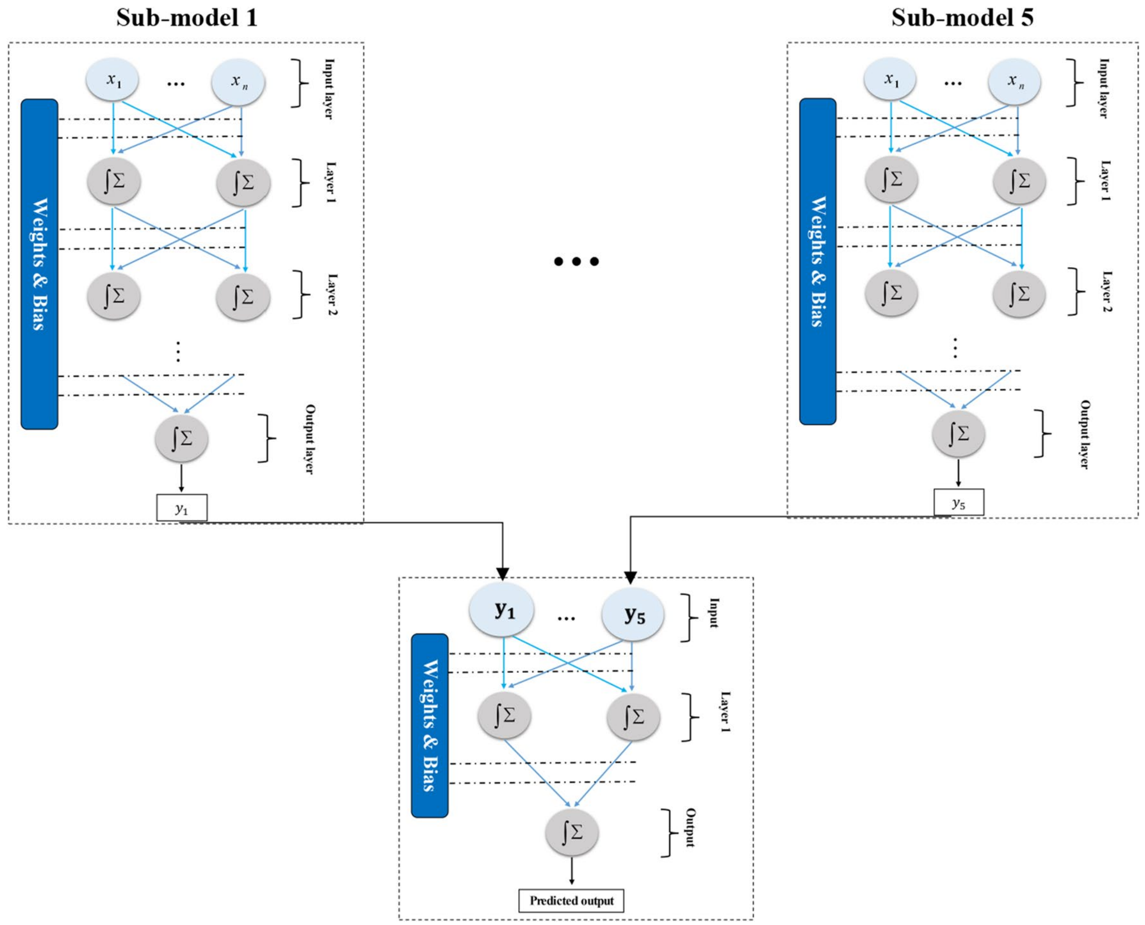


Fig. 7 Flowchart of the ISE method.

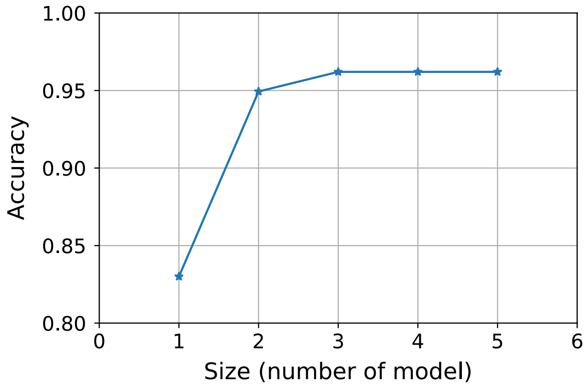


Fig. 8 Effect of the number of ensemble members.

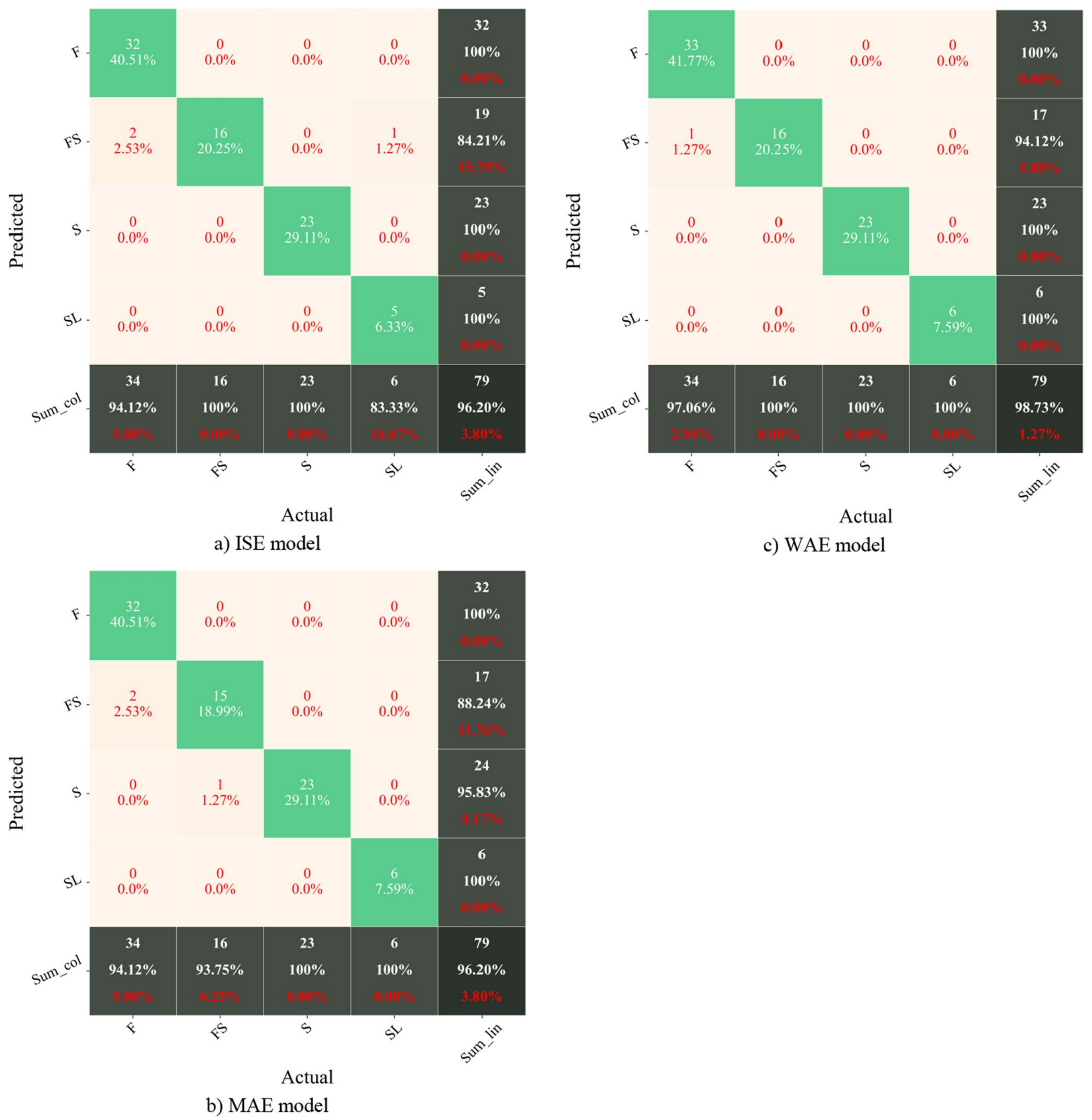
2. k-Nearest Neighbors regression (Altman, 1992): The k-Nearest Neighbors regression is the most basic nonparametric regression method. The method tries to find the closest one of the k groups that contain given input *x* and return the mean of the data values in that group. In other words, the KNN algorithm believes that similar objects are close together. To put it another way, related items are close together.
3. Decision tree (Jaworski et al., 2017; Quinlan, 1983): A decision tree is generated by repeatedly dividing the dataset into a sequence of subsets. The training set is made up of pairs (*x*, *y*), where *y* is the label that corresponds to the pair (*x*, *y*). The learning approach divides the training data set into classes based on *x*, seeking to make each group's assignments as similar as possible. The teaching process must choose a characteristic and a corresponding threshold for that characteristic, using which the data will be divided.
4. Random Forests (Breiman, 2001): Using ensemble learning, it is feasible to merge a group of decision trees into a bigger composite tree that outperforms its individual elements. The composite tree helps to reduce decision trees' main flaw: large variance. Random forest classifiers aid by averaging out the estima-

Table 5 Optimized weights.

Sub-model	1	2	3	4	5
Weight	0.0089	0.2868	0.1712	0.2930	0.2398

tions of numerous basic trees to reduce variation by training the integrated trees using random subsets of the training dataset.

5. Ensembles are groups of models that work together to achieve a classifier (Friedman et al., 2001). Bagging and boosting are the two main methods for creating ensembles. Individual high-variance classifiers benefit from bagging since the majority of the classifiers try to smooth out the individual classifiers, resulting in a more reliable joint solution. Mangalathua et. al. (2020) used CatBoost, XGBoost, AdaBoost, and LightGBM which are boosting methods. Boosting, on the other hand, is especially useful for high-bias



a) ISE model

	F	FS	S	SL	Sum_lin
F	32 40.51%	0 0.0%	0 0.0%	0 0.0%	32 100% 0.00%
FS	2 2.53%	16 20.25%	0 0.0%	1 1.27%	19 84.21% 15.79%
S	0 0.0%	0 0.0%	23 29.11%	0 0.0%	23 100% 0.00%
SL	0 0.0%	0 0.0%	0 0.0%	5 6.33%	5 100% 0.00%
Sum_col	34 94.12% 5.88%	16 100% 0.00%	23 100% 0.00%	6 83.33% 16.67%	79 96.20% 3.80%

c) WAE model

	F	FS	S	SL	Sum_lin
F	33 41.77%	0 0.0%	0 0.0%	0 0.0%	33 100% 0.00%
FS	1 1.27%	16 20.25%	0 0.0%	0 0.0%	17 94.12% 5.88%
S	0 0.0%	0 0.0%	23 29.11%	0 0.0%	23 100% 0.00%
SL	0 0.0%	0 0.0%	0 0.0%	6 7.59%	6 100% 0.00%
Sum_col	34 97.06% 2.94%	16 100% 0.00%	23 100% 0.00%	6 100% 0.00%	79 98.73% 1.27%

b) MAE model

	F	FS	S	SL	Sum_lin
F	32 40.51%	0 0.0%	0 0.0%	0 0.0%	32 100% 0.00%
FS	2 2.53%	15 18.99%	0 0.0%	0 0.0%	17 88.24% 11.76%
S	0 0.0%	1 1.27%	23 29.11%	0 0.0%	24 95.83% 4.17%
SL	0 0.0%	0 0.0%	0 0.0%	6 7.59%	6 100% 0.00%
Sum_col	34 94.12% 5.88%	16 93.75% 6.25%	23 100% 0.00%	6 100% 0.00%	79 96.20% 3.80%

Fig. 9 Confusion matrix of various models. **a** ISE model. **b** MAE model. **c** WAE model.

classifiers that take a long time to adjust to new content.

Tables 6, 7 and 8 show the three performance measures of the models used by Mangalathu et. al. (2020) and the ensemble models examined in this study. It should be noted that the Mangalathu et. al. (2020) database was used in this study. Weighted-average precision or recall means that precision or recall are calculated for each class and weight by the number of instances of each class. Overall, we see the WAE model outperforms the other approaches. The best model of Mangalathu et. al. (2020) has an accuracy of 0.86 while the WAE model has an accuracy of 0.99. In terms of other performance measures, the WAE model also fares well.

Because various splits of the data can generate significantly diverse results, repeated tenfold cross-validation is performed to measure the best deep neural network's performance. This indicates that the data is apportioned into training and test sets with a 90–10 split every time. Fig. 10 shows this by presenting model performance using tenfold cross-validation for 10 repetitions. The green triangle reflects the arithmetic mean, whereas the orange line denotes the distribution's median. The average appears to be around 0.85, 0.84, and 0.8 for the Random Forest (Fig. 10b), CatBoost (Fig. 10c), and Decision Tree (Fig. 10d), respectively. For the WAE model, the accuracy score fluctuates slightly around 0.98. As a result, these scores can be considered the most reliable estimate of models' performance. Also, the analysis of the test accuracy of the Random Forest, CatBoost, and Decision Tree clearly demonstrates a variance in the performance of the models trained on the dataset using tenfold cross-validation. It is now understood that although common machine learning techniques provide more flexibility and can scale in response to the amount of accessible training data, they learn using a stochastic learning algorithm (Brownlee, 2018; Maclin & Opitz, 2011), which means they are susceptible to the specifics of the training data and may discover a various set of hyperparameters each time they are trained, which in turn considers different levels of nonlinearity and level of interaction between parameters, resulting in different predictions (Brownlee, 2018; Maclin & Opitz, 2011). These algorithms include a lot of instability, which can be problematic when trying to come up with a final model to utilize for generating predictions (Brownlee, 2018; Maclin & Opitz, 2011). Training deep neural networks instead of a single model and combining the results from these models is a powerful way to lower the variation. This is also known as ensemble deep neural network models, which is used in this study, because it can not only minimize prediction variance, but also produce results that are better than any single model.

Table 6 Accuracy of various methods.

Model	Accuracy
Naïve Bayes (Mangalathu et al., 2020)	0.76
K-Nearest Neighbors (Mangalathu et al., 2020)	0.85
Decision Tree (Mangalathu et al., 2020)	0.8
MAE model	0.96
Random Forest (Mangalathu et al., 2020)	0.86
AdaBoost (Mangalathu et al., 2020)	0.67
XGBoost (Mangalathu et al., 2020)	0.83
MWAE model	0.84
LightGBM (Mangalathu et al., 2020)	0.8
CatBoost (Mangalathu et al., 2020)	0.84
WAE model	0.98
ISE model	0.96

Table 7 Weighted-average precision of various methods.

Models	Precision
Naïve Bayes (Mangalathu et al., 2020)	0.76
K-Nearest Neighbors (Mangalathu et al., 2020)	0.85
Decision Tree (Mangalathu et al., 2020)	0.80
MAE model	0.96
Random Forest (Mangalathu, 2020)	0.87
AdaBoost (Mangalathu et al., 2020)	0.68
XGBoost (Mangalathu et al., 2020)	0.83
MWAE model	0.83
LightGBM (Mangalathu et al., 2020)	0.80
CatBoost (Mangalathu et al., 2020)	0.84
WAE model	0.99
ISE model	0.97

Table 8 Weighted-average recall of various methods.

Techniques	Recall
Naïve Bayes (Mangalathu et al., 2020)	0.76
K-Nearest Neighbors (Mangalathu et al., 2020)	0.85
Decision Tree (Mangalathu et al., 2020)	0.79
MAE model	0.96
Random Forest (Mangalathu et al., 2020)	0.86
AdaBoost (Mangalathu et al., 2020)	0.67
XGBoost (Mangalathu et al., 2020)	0.83
MWAE model	0.84
LightGBM (Mangalathu et al., 2020)	0.80
CatBoost (Mangalathu et al., 2020)	0.84
WAE model	0.99
ISE model	0.96

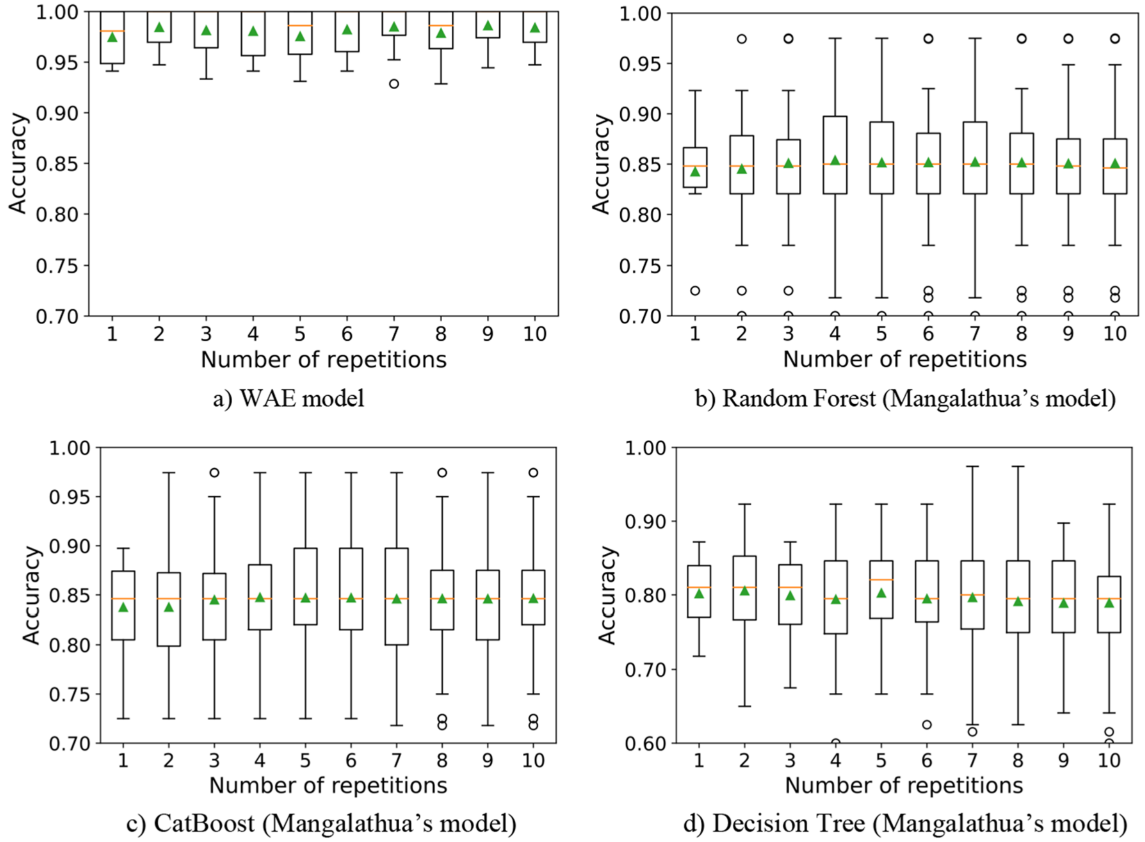


Fig. 10 Box and Whisker plots of accuracy. **a** WAE model. **b** Random Forest (Mangalathua's model). **c** CatBoost (Mangalathua's model). **d** Decision Tree (Mangalathua's model).

3.2 Model Features Analysis

In this work, SHAP (Lundberg & Lee, 2017) is utilized to analyze the WAE model's predictions. SHAP is a game theory-based technique that can be used to indicate how the parameters affect the response. The output model in SHAP is created by adding input variables in a linear form (Eq. 5):

$$f(x) = k(x') = \varphi_0 + \sum \varphi_i x'_i, x = h(x'). \tag{5}$$

In Eq. 5, $f(x)$ is the original model, x is the original input, k is the explanation model for $f(x)$. A connection is made between x and x' employing a function called $h_x(x')$. The decision score for each class is averaged across the samples in the training set to approximate φ_0, which is stored as the expected value attribute of the explainer. The unknowns of Eq. 5 are calculated using Eq. 6:

$$\varphi_i(f,x) = \sum_{z' \subseteq x'} \frac{|z'|!(M - |z'| - 1)!}{M!}\left[f_x(z') - f_x(z' \backslash i)\right], z' \subseteq x'$$
$$f_x(z') = f(h_x(z')) = E[f(z)|z_s], z' \subseteq x'. \tag{6}$$

In Eq. 6, M is the number of simplified input, $|z'|$ the count of entries that are non-zero in z', S is the set of non-zero indices in z', $(z' \backslash i)$ denote setting $z_i^0 = 0$, and $E[f(z)|z_s]$ is SHAP explanation.

The SHAP summary chart, shown in Fig. 11, ranks features according to their importance in identifying failure modes. As we can see, the model's most critical component is the aspect ratio (M/Vl_w). This is most likely due to the relative involvement of shear and flexural deformations. Flexural deformations cause the majority of lateral deformations in slender walls. The contribution of shear deformations is notably higher for moderate-aspect-ratio walls and especially short walls due to the presence of load transmission systems (e.g., strut action). The effect of each feature on each output (type of failure mode) is also shown in different colors. As an example, the ratio of length to thickness of the wall (l_w/t_w) has a greater effect on the wall with flexural-shear failure mode. In other words, for the ratio of length to thickness of the wall, the mean (|SHAP|) value is about (0.16–0.09) = 0.07 in shear failure mode class, and (0.28–0.17) = 0.11 in flexural-shear

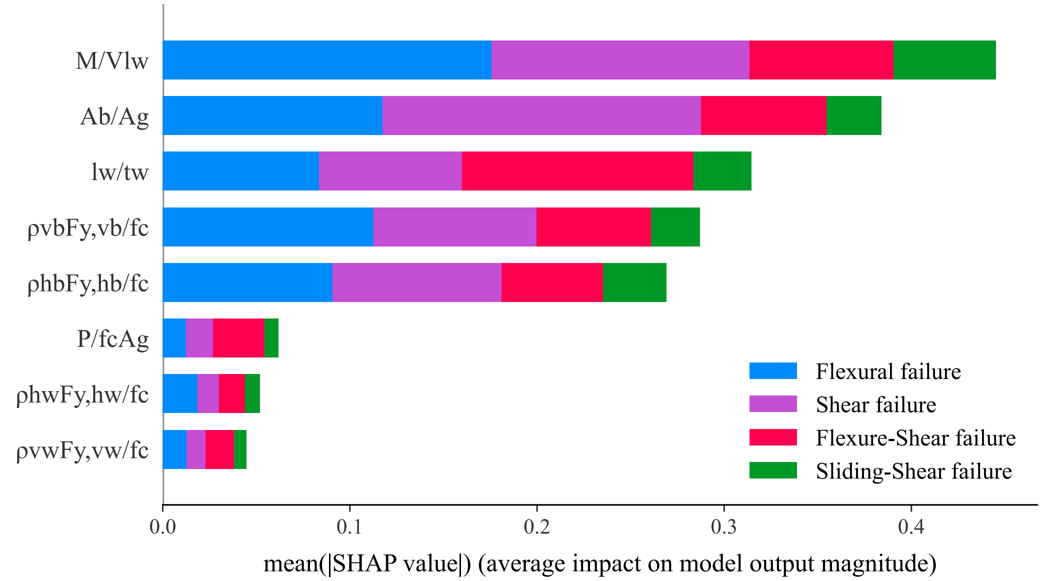


Fig. 11 Failure mode—SHAP summary plot.

failure mode class, which means that the feature l_w/t_w can influence predicting the flexural-shear failure mode more than the shear failure mode. The other features that mostly affect the detection of different types of failure modes are the boundary element area to area of cross-section ratio (A_b/A_g), and the vertical and horizontal boundary element reinforcing contribution ($\rho_{vb} f_{y,vb}/f_c$; $\rho_{hb}f_{y,hb}/f_c$).

To visualize the impact of the characteristics on the decision scores associated with each class, a different type of summary plot is employed (Fig. 12). In Fig. 12, the attributes that have the greatest impact on the decision score for each class are located at the top and blue-colored or red-colored points represent low values or high values of the parameter, respectively. Except for the flexure–shear failure mode and shear failure mode class (Fig. 12b, c), the aspect ratio has the most impact on the model output. As the value of the aspect ratio increases, their impact also increases and the model is more likely to predict flexure failure class (Fig. 12a) which corresponds to a larger probability of walls yielding in flexure before reaching the nominal shear strength of the wall; consequently, flexural behavior dominates the inelastic response.

On the other hand, the boundary element area to area of cross-section ratio is the next most important feature (Fig. 12a), and lower values of this feature correspond to a higher chance to predict flexure failure class. This observation is likely related to the wall flexural strength increasing when the boundary element area is augmented. As a result, if the lateral force related to flexural capacity is greater than the lateral force corresponding to shear capacity, shear failure may happen before flexural failure under lateral loading. In the case of the shear failure class (Fig. 12c), the boundary element area to area of cross-section ratio has the most impact on the model output. The impact of the aspect ratio is almost inverse to its effect in the flexure failure class (Fig. 12a). Low values of the feature increase the likelihood of shear failure. In the case of the flexure–shear failure mode class (Fig. 12b), although mostly influenced by the ratio of length to thickness of the wall, there is no clear correlation probably associated with the difficulty of assigning and identifying such failure mode. In the case of shear sliding failure mode, the aspect ratio feature also has the most impact. The model is more likely to anticipate shear sliding failure mode as the aspect ratio lowers (Fig. 12d) because the shear sliding strength tends to remain constant with wall height, but lateral load might increase with height reduction by preventing flexural failure. Regarding the least important factors, the SHAP values of axial load ratio and web reinforcing ratio in vertical/horizontal direction are almost close to zero for all failure types signifying that the axial load ratio and the web reinforcing ratio are the least important factors compared to other parameters. The effect of these factors on the failure mode identification cannot be interpreted in Fig. 12 since the dots are mixed and do not show the change in the SHAP value with the variation of the input features appropriately. In addition, the maximum SHAP value of the aspect ratio for flexural failure cases is higher than

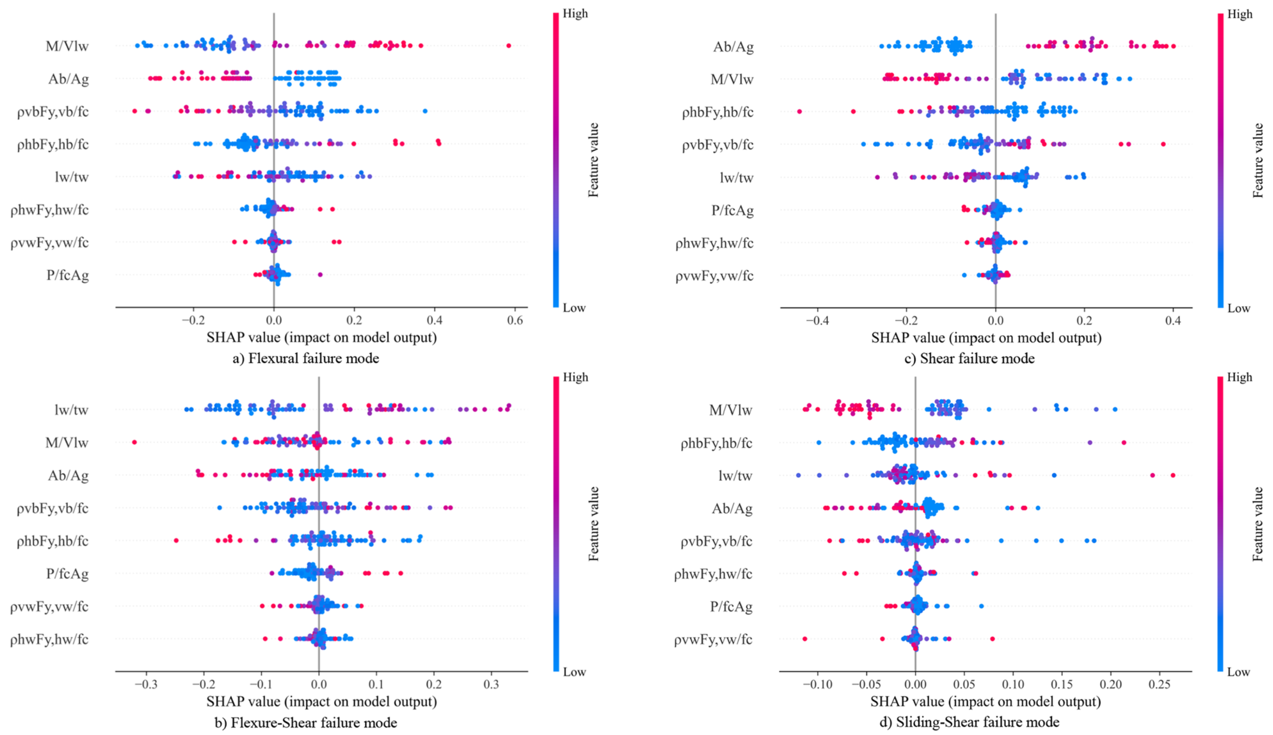


Fig. 12 Summary plot for each class. **a** Flexural failure mode. **b** Flexure–shear failure mode. **c** Shear failure mode. **d** Sliding-shear failure mode.

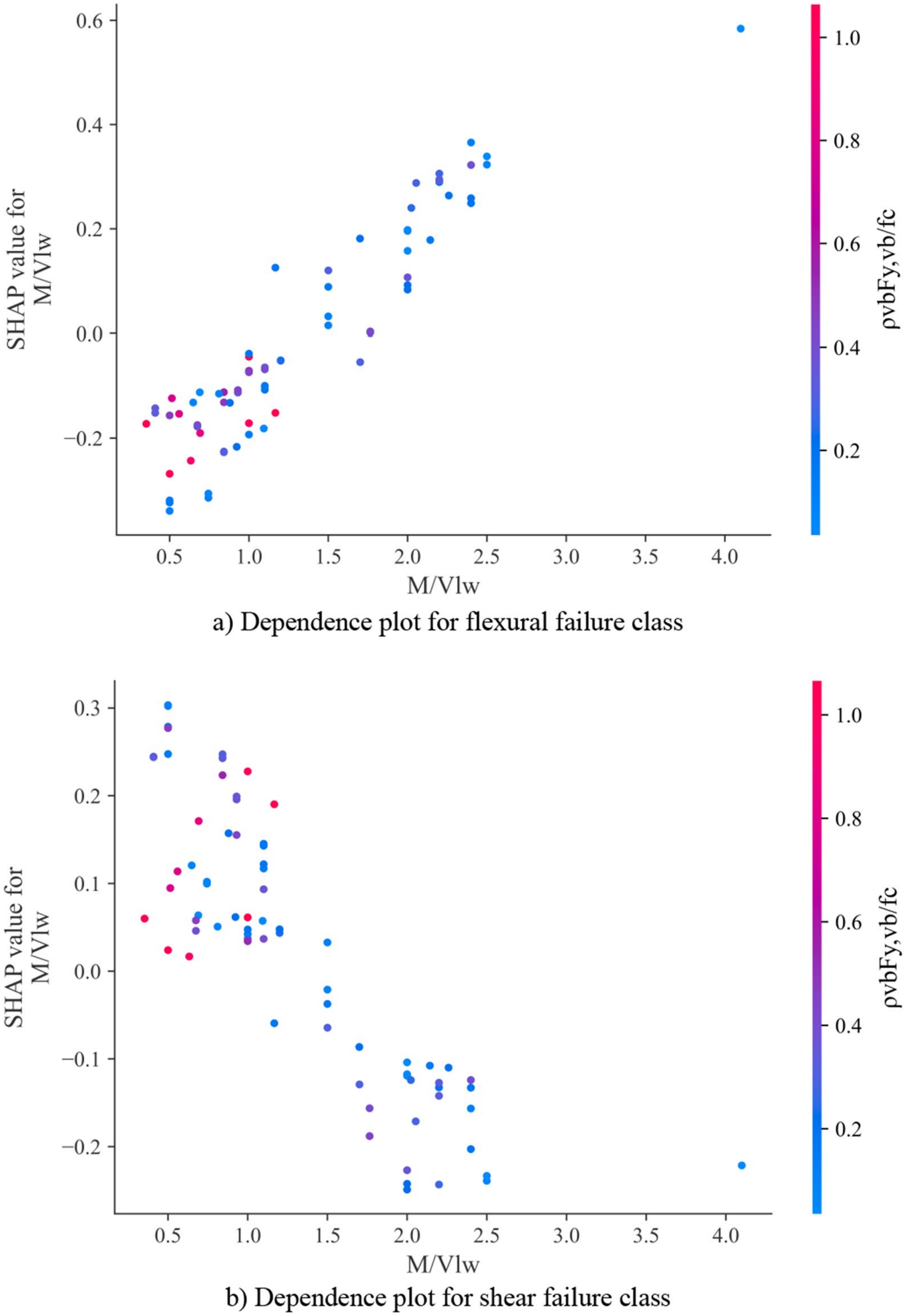
the other cases, which means that a small increase in the aspect ratio value increases the probability of flexural mode more than in other cases. For flexure–shear failure mode (Fig. 12b), the cross-sectional aspect ratio, defined as the ratio of length to thickness of the wall, is the most dominant parameter. A similar trend was reported by Lowes et. al. (ACI Committee, 2019) for the cross-sectional aspect ratio, where their study showed that walls with moderate to high shear stress demand and higher cross-sectional aspect ratio are susceptible to flexure–shear failures.

Fig. 13a shows the value of the aspect ratio on the x-axis and the SHAP value of it with respect to flexural failure mode on the y-axis by changing the reinforcing ratio in the vertical direction ($\rho_{\mathrm{vb}}f_{\mathrm{y,vb}}/f_{\mathrm{c}}$). The blue points represent lower values of $\rho_{\mathrm{vb}}f_{\mathrm{y,vb}}/f_{\mathrm{c}}$. Blue dots are almost on the right-hand side of Fig. 13a, where values of aspect ratio are high. Hence, increasing the aspect ratio while the boundary element reinforcing ratio in the vertical direction is low is resulting in a higher chance of flexural failure mode. For Fig. 13b, despite some noise, SHAP values (with respect to shear failure) for low aspect ratio are above zero, which suggests that increasing the boundary element reinforcing ratio in the vertical direction ($\rho_{\mathrm{vb}}f_{\mathrm{y,vb}}/f_{\mathrm{c}}$), while the wall aspect ratio is low, increases the probability of shear failure mode which can be caused by the reinforcement. The use of boundary element reinforcement can help to increase the wall flexural strength, delay the beginning of bar buckling and enhance the inner concrete's normal strain capacity.

3.3 Comparisons with Design Code

Three types of failure for structural walls, namely flexural, shear and shear sliding failures can be categorized using ACI 318–19 design code (ACI Committee, 2019) and the concept of strength calculation. It is evident that if the shear strength of the RCSWs (Eq. 7) is lower than the shear force associated with flexural capacity, the failure occurs in shear mode. The ACI suggested a shear friction limit (Eq. 8), which is commonly used for shear sliding in walls. This equation is used as a guideline for when sliding shear takes over. In this paper, ACI 318–19 is utilized for calculating the shear and flexural capacity of RCSWs. The ACI suggested that a shear stress limit of $0.66\sqrt{f_c'}$ MPa be used as a guideline to prevent diagonal compression failure:

$$\begin{aligned} &V_n = A_{\mathrm{cv}}(\alpha_{\mathrm{c}}\sqrt{f_{\mathrm{c}}'} + \rho_t f_y) \\ &\text{if wall aspect ratio} \leq 1.5 \rightarrow \alpha_{\mathrm{c}} = 0.25 \\ &\text{if wall aspect ratio} \geq 2.0 \rightarrow \alpha_{\mathrm{c}} = 0.17, \end{aligned} \tag{7}$$



a) Dependence plot for flexural failure class

b) Dependence plot for shear failure class

Fig. 13 SHAP dependency analysis. **a** Dependence plot for flexural failure class. **b** Dependence plot for shear failure class.

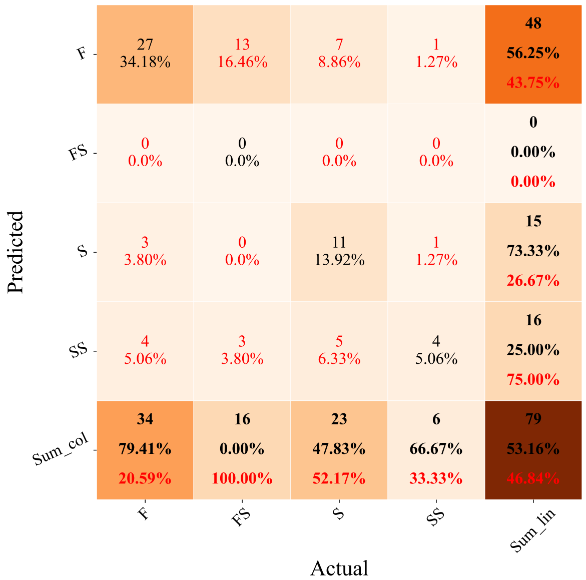


Fig. 14 Confusion matrix based on code concept.

$$
\begin{aligned}
&V_n = \mu A_{\mathrm{vf}} f_y < V^n_{\max}, \mu = 1.4\\
&V^n_{\max} = \min(0.2f'_{\mathrm{c}}A_{\mathrm{c}}, (3.3 + 0.08f'_{\mathrm{c}})A_{\mathrm{c}}, 11A_{\mathrm{c}}),
\end{aligned}
\tag{8}
$$

where V_{n} is nominal shear strength, A_{cv} is the gross area of the section, f'_{c} is concrete strength, ρ_{t} is the transverse reinforcement ratio, f_{y} is the yield strength of the transverse reinforcement, μ is coefficient of friction, and A_{vf} is the area of reinforcement crossing the assumed shear plane to resist shear, A_{c} is the area of concrete section resisting shear transfer. Fig. 14 shows the confusion matrix based on the code concept for the test set. It can be seen that by using ACI 318–19 the accuracy in failure mode prediction is almost 53.2% which is much lower than the WAE model's accuracy (with 98.7%). In addition, code concepts cannot be used to identify flexure–shear failure mode. For this reason, walls with flexure–shear failure mode are either in the flexural failure mode group or in the shear sliding failure mode group.

4 Conclusion

Reinforced concrete structural walls (RCSWs) are often utilized as the major lateral force-resisting mechanism for residential and commercial low-to-high rise buildings in locations prone to severe magnitude earthquakes. Many analytical models and experimental studies have been carried out to investigate the nonlinear behavior of reinforced concrete structural walls, identifying failure modes of the RCSWs. In addition, there have been several studies that have investigated the failure mode of the RCSWs using machine learning methods. This study aimed to examine and determine the efficiency of the ensemble neural networks to predict the failure mechanism of the RCSWs. The strongest model for predicting the failure mode of the RCSWs is determined by evaluating ensemble deep neural network models: model averaging, weighted average, and integrated stacking. Ensemble models are based on 5 neural network sub-models, whose performance in terms of accuracy (R^2 score) ranges between 0.81 and 0.84. The following is a summary of the primary conclusions:

- The weighted average ensemble model outperforms the other ensemble neural network models, yielding an accuracy of 0.987, since it is capable of carrying forward the better sub-models with higher weights.
- The performance of the weighted average ensemble model of this study is compared with well-known ensemble models (AdaBoost, XGBoost, LightGBM, and CatBoost) and traditional machine learning methods (Naïve Bayes, K-Nearest Neighbors, Decision Tree, and Random Forest). The weighted average ensemble model outperforms traditional well-known ensemble models in detecting the failure mode since it has the highest accuracy, precision, and recall among the other models.
- Merging the estimations from multiple deep neural networks counters the variance of a single trained model. The outcomes are predictions that are less susceptible to the specific details of the training data, and selection of the training scheme and the process of finding the right combination of hyperparameter values.
- A game theory-based technique is employed to explain the weighted average ensemble model's predictions. The results of this technique showed that the effective parameters in shear wall failure depend on the failure mechanism. But in all four types of failure modes, the aspect ratio of the wall was ranked either first or second.
- Other features that mostly affect the detection of different types of failure modes are the boundary element area to area of cross-section ratio, the ratio of length to thickness of the wall, and the vertical and horizontal boundary element reinforcing contribution.
- Comparison between the results of the weighted average ensemble model against internationally recognized building standard code (ACI 318–19 design code and the concept of strength calculation) shows that the machine learning model is more accurate in identifying the failure mechanism of the RCSWs. In

addition, code concepts cannot be used to identify flexure–shear failure mode.

The results evidence the capability of ensemble models to improve the predictability capacity of failure modes in shear walls based on neural networks sub-models, also explaining consistently the impact of feature values in the failure mode occurrence.

Acknowledgements
Not applicable.

Author contributions
MSB: formal analysis, investigation, resources, data curation, and writing—original draft. LMM: conceptualization, methodology, and writing—review. Both authors read and approved the final manuscript.

Authors' information
(1) Mohammad Sadegh Barkhordari, Ph.D., Department of Civil and Environmental Engineering, Amirkabir University of Technology, Tehran, Iran.
(2) Leonardo M. Massone, Professor, Department of Civil Engineering, University of Chile, Blanco Encalada 2002, Santiago, Chile.

Funding
Not applicable.

Data availability
The datasets used and/or analyzed during the current study are available from the corresponding author on reasonable request.

Declarations

Consent for publication
Authors give consent for the publication of identifiable details, which can include photograph(s) and/or videos and/or text and/or details within the text ("Material") to be published in the *International Journal of Concrete Structures and Materials.*

Competing interests
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Author details
[1]Department of Civil and Environmental Engineering, Amirkabir University of Technology, Tehran, Iran. [2]Department of Civil Engineering, University of Chile, Blanco Encalada 2002, Santiago, Chile.

Received: 15 December 2021 Accepted: 27 March 2022
Published online: 30 May 2022

References

Ahmadianfar, I., et al. (2021). RUN beyond the metaphor: An efficient optimization algorithm based on Runge Kutta method. *Expert Systems with Applications, 181*, 115079.

ACI Committee. (2019). Building code requirements for structural concrete (ACI 318–19). *Building code requirements for structural concrete.* American Concrete Institute.

Altman, N. S. (1992). An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician, 46*(3), 175–185.

Barkhordari, M. S., & Tehranizadeh, M. (2021). Response estimation of reinforced concrete shear walls using artificial neural network and simulated annealing algorithm. *Structures.* Elsevier.

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

Bennis, F., & Bhattacharjya, R. K. (2020). *Nature-inspired methods for metaheuristics optimization: Algorithms and applications in science and engineering* (Vol. 16). Springer.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Brownlee, J. (2018). *Better deep learning: Train faster, reduce overfitting, and make better predictions*. Machine Learning Mastery.

Chen, X., et al. (2018). Prediction of shear strength for squat RC walls using a hybrid ANN–PSO model. *Engineering with Computers, 34*(2), 367–383.

Chollet, F. (2015). *Keras*. https://keras.io

Domingos, P., & Pazzani, M. (1997). On the optimality of the simple Bayesian classifier under zero-one loss. *Machine Learning, 29*(2), 103–130.

Dozat, T. (2016). *Incorporating nesterov momentum into adam.*

Duchi, J., Hazan, E., & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research, 12*(7), 2121–2159.

Feng, D.-C., et al. (2021). Interpretable XGBoost-SHAP machine-learning model for shear strength prediction of squat RC walls. *Journal of Structural Engineering, 147*(11), 04021173.

Friedman, J., Hastie, T., & Tibshirani, R. (2001). *The elements of statistical learning* Springer series in statistics (Vol. 1). Springer.

Gondia, A., Ezzeldin, M., & El-Dakhakhni, W. (2020). Mechanics-guided genetic programming expression for shear-strength prediction of squat reinforced concrete walls with boundary elements. *Journal of Structural Engineering, 146*(11), 04020223.

Goodfellow, I., Bengio, Y., & Courville, A. (2017). *Deep learning* Adaptive computation and machine learning series (pp. 321–359). The MIT Press.

Grammatikou, S., Biskinis, D., & Fardis, M. N. (2015). Strength, deformation capacity and failure modes of RC walls under cyclic loading. *Bulletin of Earthquake Engineering, 13*(11), 3277–3300.

Jaworski, M., Duda, P., & Rutkowski, L. (2017). New splitting criteria for decision trees in stationary data streams. *IEEE Transactions on Neural Networks and Learning Systems, 29*(6), 2516–2529.

Karaboga, N., & Cetinkaya, B. (2004). Performance comparison of genetic and differential evolution algorithms for digital FIR filter design. In *International conference on advances in information systems*. Springer.

Keshtegar, B., et al. (2021a). Novel hybrid machine leaning model for predicting shear strength of reinforced concrete shear walls. *Engineering with Computers*. https://doi.org/10.1007/s00366-021-01302-0

Keshtegar, B., et al. (2021b). Predicting load capacity of shear walls using SVR–RSM model. *Applied Soft Computing, 112*, 107739.

Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. arXiv preprint retrieved from arXiv:1412.6980

Krogh, A., & Hertz, J. A. (1992). A simple weight decay can improve generalization. *Advances in neural information processing systems.* MIT Press.

Lee, Y., Oh S.- H., & Kim, M. W. (1991). The effect of initial weights on premature saturation in back-propagation learning. In *IJCNN-91-Seattle international joint conference on neural networks*. IEEE.

Lundberg, S. M., & Lee, S. I. (2017). A unified approach to interpreting model predictions. In *Proceedings of the 31st international conference on neural information processing systems*.

Maclin, R., & Opitz, D. (2011). Popular ensemble methods: An empirical study. ArXiv e-prints retrieved from arXiv:1106.0257

Mangalathu, S., et al. (2020). Data-driven machine-learning-based seismic failure mode identification of reinforced concrete shear walls. *Engineering Structures, 208*, 110331.

Massone, L. M., López, C. N., & Kolozvari, K. (2021). Formulation of an efficient shear-flexure interaction model for planar reinforced concrete walls. *Engineering Structures, 243*, 112680.

McMahan, H. B., et al. (2013). Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD international conference on knowledge discovery and data mining*.

Moradi, M. J., et al. (2020). Prediction of the load-bearing behavior of spsw with rectangular opening by RBF network. *Applied Sciences, 10*(3), 1185.

Naimi, A. I., & Balzer, L. B. (2018). Stacked generalization: An introduction to super learning. *European Journal of Epidemiology, 33*(5), 459–464.

Nguyen, D.-D., et al. (2021). A machine learning-based formulation for predicting shear capacity of squat flanged RC walls. *Structures.* Elsevier.

Osisanwo, F., et al. (2017). Supervised machine learning algorithms: Classification and comparison. *International Journal of Computer Trends and Technology (IJCTT), 48*(3), 128–138.

Parsa, P., & Naderpour, H. (2021). Shear strength estimation of reinforced concrete walls using support vector regression improved by teaching–learning-based optimization, Particle Swarm optimization, and Harris Hawks optimization algorithms. *Journal of Building Engineering, 44*, 102593.

Pizarro, P. N., et al. (2021). Use of convolutional networks in the conceptual structural design of shear wall buildings layout. *Engineering Structures, 239*, 112311.

Pizarro, P. N., & Massone, L. M. (2021). Structural design of reinforced concrete buildings based on deep neural networks. *Engineering Structures, 241*, 112377.

Quinlan, J. R. (1983). Learning efficient classification procedures and their application to chess end games. *Machine learning* (pp. 463–482). Springer.

Usta, M., et al. (2017). *ACI 445B shear wall database*.

Zeiler, M. D. (2012). Adadelta: An adaptive learning rate method. arXiv preprint retrieved from arXiv:1212.5701

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.