

Research Article

An Open-Source Framework for Modeling RC Shear Walls Using Deep Neural Networks

German Solorzano ¹ and Vagelis Plevris ²

¹Department of Civil Engineering and Energy Technology, OsloMet-Oslo Metropolitan University, Pilestredet 35, Oslo 0166, Norway

²Department of Civil and Architectural Engineering, Qatar University, P.O. Box 2713, Doha, Qatar

Correspondence should be addressed to Vagelis Plevris; vplevris@qu.edu.qa

Received 21 March 2023; Revised 27 May 2023; Accepted 22 June 2023; Published 11 July 2023

Academic Editor: Nicola Buratti

Copyright © 2023 German Solorzano and Vagelis Plevris. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Reinforced concrete (RC) shear walls macroscopic models are simplified strategies able to simulate the complex nonlinear behavior of RC shear walls to some extent, but their efficacy and robustness are limited. In contrast, microscopic models are sophisticated finite element method (FEM) models that are far more accurate and reliable. However, their elevated computational cost turns them unfeasible for most practical applications. In this study, a data-driven surrogate model for analyzing RC shear walls is developed using deep neural networks (DNNs). The surrogate model is trained with thousands of FEM simulations to predict the characteristic curve obtained when a static nonlinear pushover analysis is performed. The surrogate model is extensively tested and found to exhibit a high degree of accuracy in its predictions while being extremely faster than the detailed FEM analysis. The complete framework that made this study possible is provided as an open-source project. The project is developed in Python and includes a parametric FEM model of an RC shear wall in OpenSeesPy, the training and validation of the DNN model in TensorFlow, and an application with an interactive graphical user interface to test the methodology and visualize the results.

1. Introduction

Reinforced concrete (RC) shear walls are typically used as the main lateral-force-resisting elements in mid and high-rise buildings. Their primary purpose is to increase the building's lateral stiffness and withstand the severe in-plane shear forces generated when strong lateral loads are applied to the structure, such as those induced by earthquakes or high-speed winds. Consequently, RC shear walls are prompt to suffer significant structural damage that could compromise the integrity of the building [1]. To ensure a safe and sound structural design, it is essential to develop reliable RC shear wall modeling techniques that take into consideration the well-known nonlinear behavior that characterizes concrete structures. A significant amount of effort and research has been carried out toward this goal, including the rigorous experimental testing of RC shear walls under lateral-induced cyclic loading [2–4] and the development of various modeling techniques, which can be classified into two major categories: macroscopic and microscopic models [5].

Macroscopic models or macromodels are essentially simplified systems consisting of multiple vertical (axial) and horizontal/rotational (shear) springs that are connected together with rigid elements. Alternatively, similar techniques have been developed using truss or plane stress elements instead of springs [6]. The elements in macromodels are arranged in specific configurations that enable them to simulate the flexural and shear behavior of RC shear walls. For example, the popular multiple vertical lines element model, proposed by Vulcano et al. [7], is depicted in Figure 1(a). Macromodels that are combined with appropriated nonlinear force–displacement relationships have proven capable of reproducing the complex behavior of RC shear walls to some extent [8–12]. The main advantage of the macromodeling approach is that it is computationally efficient and relatively easy to implement. Nevertheless, one of the main setbacks is that its accuracy is questionable for certain effects. For instance, such models may considerably underestimate the compressive strain of the concrete, rendering them unable to capture the failure due to concrete crushing [13].

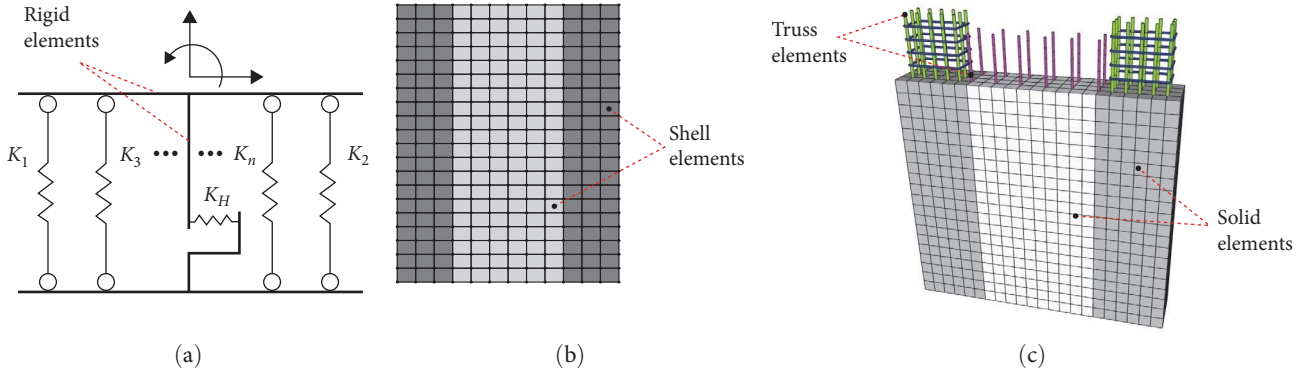


FIGURE 1: Different modeling techniques for an RC wall: (a) macromodel using the MLVM approach; (b) finite element model using shell elements; (c) finite element model using 3D solid and truss elements.

Alternatively, microscopic models consist of detailed finite element method (FEM) models using a combination of solid, shell, and beam elements that describe the geometry of the wall with significantly more detail, such as those depicted in Figures 1(b) and 1(c). Typically, micromodels are implemented with appropriated nonlinear constitutive material laws based on fracture mechanics and provide a much more reliable modeling alternative that can reproduce even severe damage scenarios with reasonable accuracy [14–18]. Nonetheless, their analysis tends to be computationally expensive due to the refined mesh and the small step required for the numerical stability of the nonlinear solution procedure. Moreover, their adequate implementation requires an experienced engineer with advanced knowledge in nonlinear computational mechanics and powerful FEM software that includes the corresponding constitutive models. Therefore, the usage of microscopic models is generally only conceived for research purposes or highly important projects that can afford the increased time and effort.

There is a tradeoff between robustness, accuracy, and computational cost between the RC shear wall macroscopic and microscopic modeling alternatives. So far, no methodology has been found to dominate over the others overwhelmingly. Consequentially, the development of new models and the improvement of existing methods are topics of particularly high importance that motivate researchers to bring innovative ideas to the table. In that context, leveraging the advantages of the recent developments in machine learning (ML), and particularly, artificial neural networks (ANNs), could significantly contribute to tackling the aforementioned challenges more efficiently [19]. For instance, finding ways to reduce the computational cost of the microscopic FEM-based models or increasing the robustness of the macroscopic modeling techniques [20], among other interesting applications in structural engineering [21, 22].

This paper develops a novel modeling alternative for RC shear walls that consists of creating a computationally efficient surrogate model using deep neural networks (DNNs). The DNN surrogate model is trained to predict the complex nonlinear response of the wall based on an extensive database of thousands of analyses that are generated with a detailed microscopic FEM model. The predicted response

corresponds to the nonlinear force–displacement curve obtained when a lateral displacement-controlled pushover static analysis is performed using a detailed FEM model. Furthermore, in this paper, a second but equally important goal is to provide, as an open-source project, the complete framework that has been developed to make this study possible. By making the source code available, we intend to contribute to reducing the gap between the research and the real-world application of ML-powered methodologies in the field of Structural Engineering. The project is entirely developed in Python and includes four major components: (i) a parametric FEM model of an RC shear wall in OpenSeesPy [23]; (ii) an automated subroutine to run thousands of analyses and generate a usable database to train ML models; (iii) the training and testing of the DNN using the TensorFlow library [24]; and (iv) a powerful application with an interactive GUI to explore and visualize the results.

The methodology presented in this paper is inspired by a previous study published by the authors where a similar surrogate model was developed to predict the lateral load capacity of an RC wall [25]. The previous study considered only a few parameters to describe the wall properties and predicted a single numerical value corresponding to the peak of the pushover curve. In this new study, the whole curve is approximated, and more variables are introduced to describe the geometry and the material properties of the RC wall, thus, significantly increasing the robustness compared to the previous methodology. Similarly, other studies using ANNs for modeling RC shear walls include the work by Barkhordari and Tehranizadeh [26] that proposed a hybrid approach using ANN and the simulated annealing algorithm to predict the response of RC shear walls. Moradi and Hariri-Ardebili [27] created a database of shear walls by assembling the results from various experimental tests available in the literature. They use the database to develop an ANN capable of approximating the response of shear walls with relatively good accuracy. Furthermore, other examples of studies that use surrogate models to alleviate the computational cost of FEM simulations can be found [28–30].

The rest of the paper is structured as follows. In Section 2, the FEM model used to create the training database is thoroughly described. In Section 3, the basic structure of an ANN

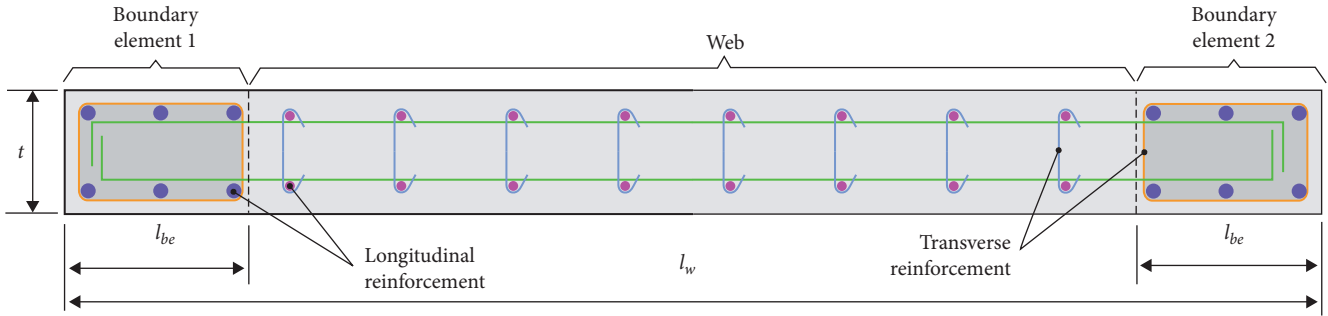


FIGURE 2: Typical RC shear wall cross-section.

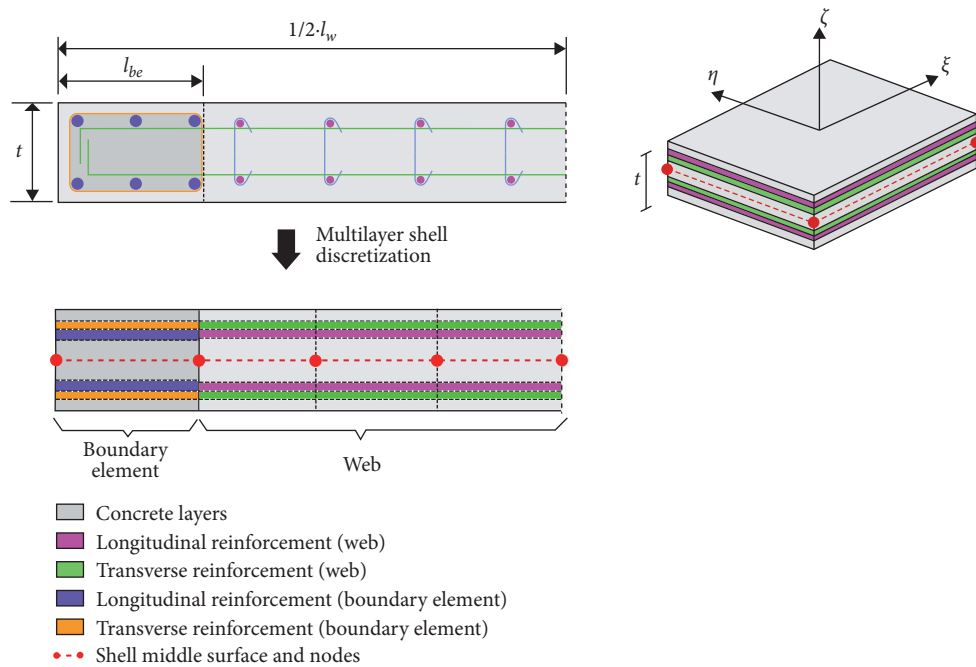


FIGURE 3: RC shear wall modeling using multilayer shell element.

is detailed, followed by a description of the creation, training, and validation of the ANN-based surrogate model in Sections 3.2 to 3.4. Various numerical examples to test the surrogate model predictions are presented in Section 3.5, and the results are discussed in Section 4. A brief description of the developed open-source framework is presented in Section 5. Finally, the conclusions of this study and the possible future directions are discussed in Section 6.

2. RC Shear Wall FE Model

2.1. Typical RC Shear Wall Reinforcement. RC shear walls are typically built with higher quantities of reinforced steel and stricter detailing criteria near the edges compared to the other sections of the wall. Such reinforcement conditions follow the high concentration of stresses near the edges resulting from the characteristic stress distribution of elements subjected to bending. When in-plane lateral loads are applied to the wall, one edge experiences compression while the other extreme is under tension. Therefore, a

standard procedure is the inclusion of so-called boundary elements placed at the edges of the wall. The steel reinforcement at the boundary elements resembles similar characteristics to the standard reinforcement that is applied to columns, providing sufficient longitudinal steel reinforcement as well as appropriate confinement to withstand the developed compressive and tensile stresses. The rest of the wall is commonly denoted as the web and typically requires a lesser quantity of steel reinforcement. A representative image of an RC shear wall cross section is given in Figure 2.

2.2. Multilayer Shell Element (MLSE) for RC Shear Walls. The MLSE is a specific shell formulation in which the total thickness of the shell is discretized into several fully bonded layers with smaller thickness values. Each layer may have different material properties, thus, making it relatively simple to model RC shear walls where the longitudinal and transverse steel reinforcement are included as smeared orthotropic layers, as depicted in Figure 3. With an appropriate layer discretization, the actual stress distribution over the

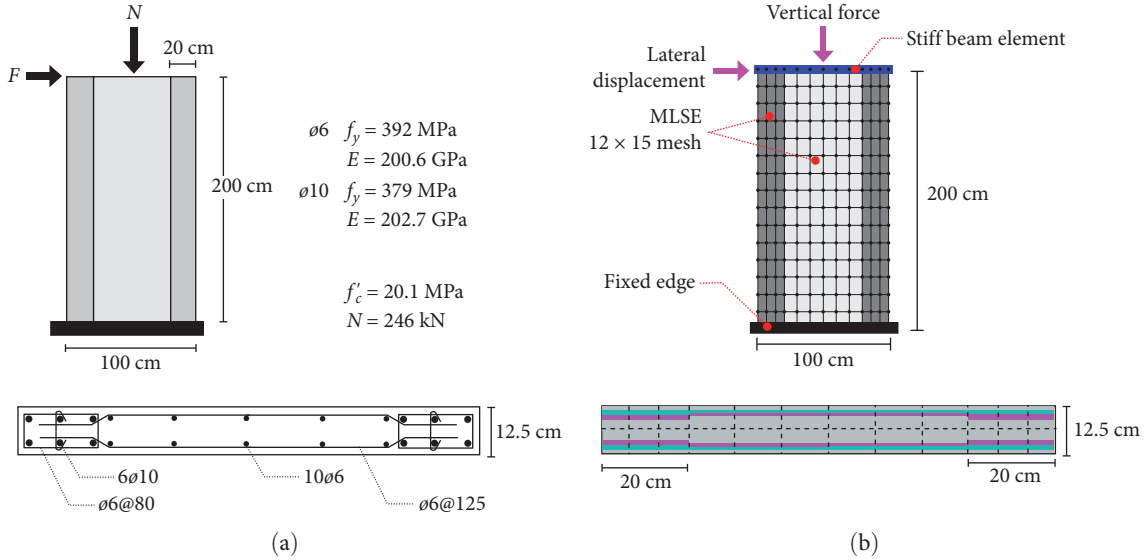


FIGURE 4: (a) Geometry and properties of the specimen SW1-1 and (b) specimen SW1-1 modeled using the MLSE formulation.

thickness of the wall can be approximated. The main principle is similar to how the popular fiber beam-column element works [31]. First, the strain and the curvature are computed at the middle layer. Next, the strains are obtained in the subsequent layers using a plane-section assumption. Finally, the stresses are obtained based on the corresponding constitutive law at each layer. The MLSE approach has been thoroughly tested in numerous studies. It has been found to be effective in capturing the coupled in-plane and out-of-plane bending, as well as the in-plane direct shear and coupled bending-shear characteristic behavior of RC shear walls [32, 33].

2.3. MLSE Model Implementation in OpenSees. This study uses the MLSE modeling approach to generate a large database of RC shear walls and the corresponding results. The model is implemented in OpenSees [34], more specifically, the Python distribution OpenSeesPy [23]. The adopted MLSE implementation is the one developed by Xie et al. [17], which has been successfully applied to the analysis of super-tall buildings [35]. Within the OpenSees environment, such MLSE is identified as the element “ShellNLDKGQ.” It includes the effect of geometric nonlinearity due to large deformations based on the updated Lagrangian formulation [36]. The layer discretization is specified by assigning to every shell element the “LayeredShell” section that stores the information regarding the thickness and the associated material model for each layer.

Each shell is discretized with eight layers. Four homogeneous concrete layers and four orthotropic steel reinforcement layers, where two are for the longitudinal rebar and the other two are for the transverse rebar, as shown in Figure 3. The meshing of the model is done so that the whole RC shear wall is discretized with 12 elements in the horizontal direction and 15 elements in the vertical direction totaling $12 \times 15 = 180$ elements. Each boundary element is discretized with $3 \times 15 = 45$ elements, and the web is discretized with

$6 \times 15 = 90$ elements. This mesh configuration yields rectangular elements with an approximate size of 15–35 cm per side, as shown in Figure 4(a). The adopted mesh is sufficient to produce accurate results as the size effect is mitigated in the material model with the adoption of the popular crack band theory [37]. A more refined meshing scheme does not improve the results considerably; thus, it is not worth the increased computational effort. This claim is pointed out [35] and corroborated in this study by testing different meshing configurations, up to 576 elements, with a computational demand $5 \times$ times larger but without any significant improvement in the results.

2.3.1. Material Models. The selected material model for the concrete layers is based on a damage mechanism and the smeared crack model [38]. It is identified within OpenSees as “PlaneStressUserMaterial” and has to be combined with the “PlateFromPlaneStress” command to properly account for the out-of-plane behavior. A total of seven material parameters are required for its definition, namely, the compressive strength at 28 days f'_c ; the tensile strength f_t ; the crushing strength f_{cu} ; the strain at the maximum compressive strength ϵ_{co} ; the strain at the crushing strength ϵ_{cu} ; the ultimate tensile strain ϵ_{tu} ; and the shear retention factor (or shear transfer coefficient) β .

The adopted material model for the steel reinforcement layers is the uniaxial Giuffre–Menegotto–Pinto formulation, which includes the characteristic post-yielding and Bauschinger effects [39, 40]. It is identified in OpenSees as “Steel02”, and it is combined with the “PlateRebar” command so that it can be incorporated into the MLSE formulation considering its corresponding angle due to its orthotropic behavior (90° for longitudinal and 0° for transverse). Its definition requires three parameters: yield stress f_y ; initial elastic tangent E_0 ; and the strain-hardening ratio b , which is the ratio between the elastic and the post-yield tangent E_p .

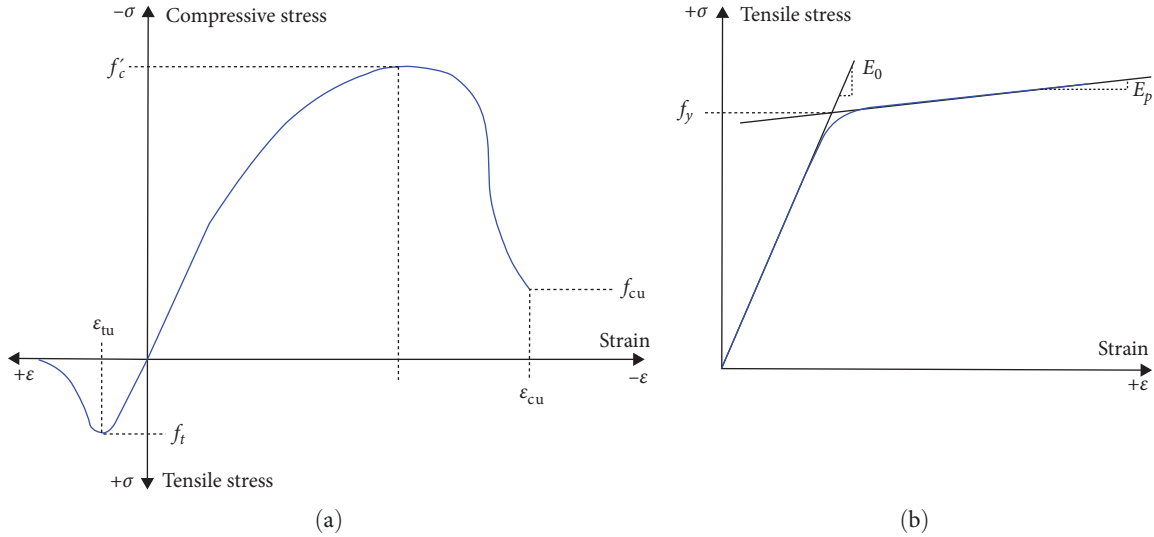


FIGURE 5: Schematic uniaxial stress–strain curve of the adopted material models: (a) concrete model; (b) steel reinforcement model.

The material models for the concrete and the steel reinforcement are illustrated with uniaxial strain–stress curves in Figure 5. Note that the curves are not in scale and that their only purpose is to provide a graphical interpretation of the material parameters.

The parameters required for the material models are commonly obtained by performing experimental testing. However, for the purposes of this study, values suggested in different sources from the literature have been mainly used. This way, the seven parameters required for the concrete material model depend entirely on the compressive strength f'_c of concrete. The fracture strength is taken as $f_t = 0.1 \cdot f'_c$ [41]. The crushing strength is $f_{cu} = 0.2 \cdot f'_c$ [42]. The strain at the maximum compressive strength is taken as $\epsilon_{co} = -0.002$ [43]. The strain at the crushing strength equals $\epsilon_{cu} = -0.01$. The ultimate tensile strain is set to $\epsilon_{tu} = 0.001$, and the shear retention factor is equal to $\beta = 0.3$ [44]. Regarding the steel material model, the yield strength f_y is the only variable. The initial elastic modulus and the strain hardening ratio are kept constant at $E_0 = 200$ GPa, and $b = 0.01$, respectively.

2.3.2. Pushover Analysis and Validation of the FEM Model. The previously described implementation of the MLSE FEM model is tested and validated with a numerical example. The example's data are taken from an open-access database of various experimental tests conducted on large-scale RC shear walls [4]. The selected example is the specimen SW1-1, whose geometry, properties, and loading conditions are depicted in Figure 4(a). A visual representation of the final form of the FEM model 2.3 is presented in Figure 4(b).

The analysis consists of two phases. In the first phase, a vertical force is applied at the top middle node that simulates the weight supported by the wall. The second phase is a lateral displacement-controlled static pushover analysis with a target displacement of 2 cm applied at the wall's top-left corner. The displacement is applied with a rate of 0.005 cm

per iteration for $2/0.005 = 400$ steps. Note that a beam element with increased stiffness is added to the top edge of the wall to distribute the vertical load and the horizontal deformation to all the nodes on the top edge. The deformed shape and the strain field in the vertical direction at various stages of the analysis are shown in Figure 6.

The pushover curve is obtained by plotting the horizontal reaction force (commonly known as base shear) and the horizontal displacement (measured at the top-left node) at every step during the analysis. To corroborate the accuracy of the FEM model, the analysis results are compared with the experimental test data found [4]. The comparison is shown in Figure 7, where it can be noticed that both curves exhibit close agreement. These results corroborate that the implemented FEM model provides a reliable approximation of the actual behavior of the RC shear wall; hence, the model is validated. In this study, only one specimen has been used for validation purposes, as the chosen MLSE implementation has been validated in more depth in other studies [17].

2.4. Model Parametrization and Sampling. The MLSE FE model is used to create a large database composed of thousands of different-sized RC shear walls and their corresponding pushover analysis results. To that end, the model is parametrized with several variables describing the geometry, steel reinforcement quantity, and material properties. The chosen sampling strategy is a random uniform distribution. Table 1 shows all the variable parameters and their corresponding lower and upper bounds used for the sampling.

The bounding values of the parameters and the sampling method are two highly important aspects of the developed methodology. In a sense, they control the quality and required quantity of the data. The quality of the data, in the context of this study, could be defined as how well the data set encompasses the real phenomenon. In other words, the generated walls must be realistic in the context of structural design. Attending to such premise, the selected bounding values

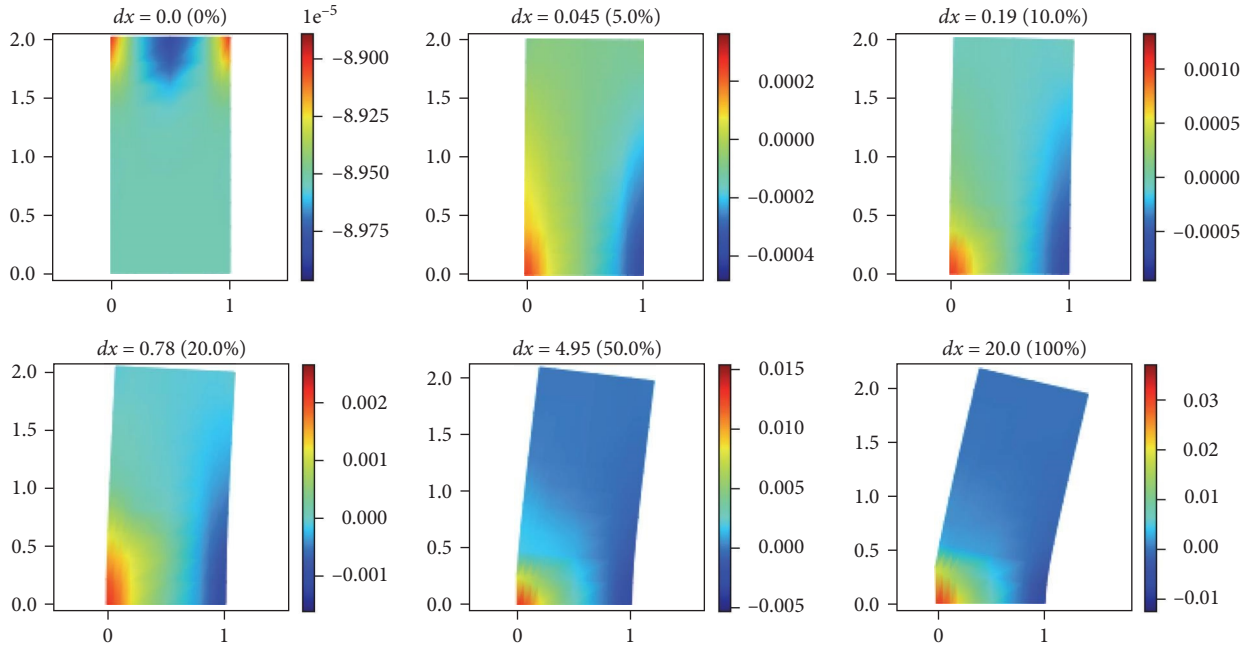


FIGURE 6: Deformation and strain field in the vertical direction at various stations throughout the pushover analysis. The units of the horizontal deformation (dx) are in mm. The scale factor for the deformations is 20.

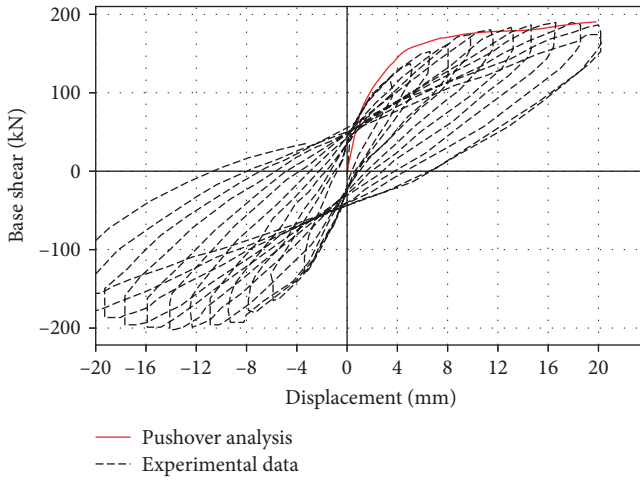


FIGURE 7: Comparison between the experimental results and the pushover curve obtained with the FEM analysis.

(see Table 1) are based on some of the design guidelines provided by the American Concrete Institute (ACI318-19) [45] as well as on structural engineering criteria. For example, from the ACI318-19, the minimum thickness allowed for a structural wall is $t = 12.5$ cm, and the smallest length-to-thickness ratio is $l_w/t = 6$, then the chosen lower bound for the wall length is set to $t \cdot 6$. Similarly, according to section 18.10.6.4 of ACI318-19, the quantity of transverse reinforcement $\rho_{t_{bc}}$ for the boundary element is approximately within the range of 0.0075–0.020 (depending on the material properties). The longitudinal reinforcement $\rho_{l_{bc}}$ is similar to those in concrete columns, so a value between 0.01 and 0.04 is reasonably selected. Note that the reinforcement is expressed

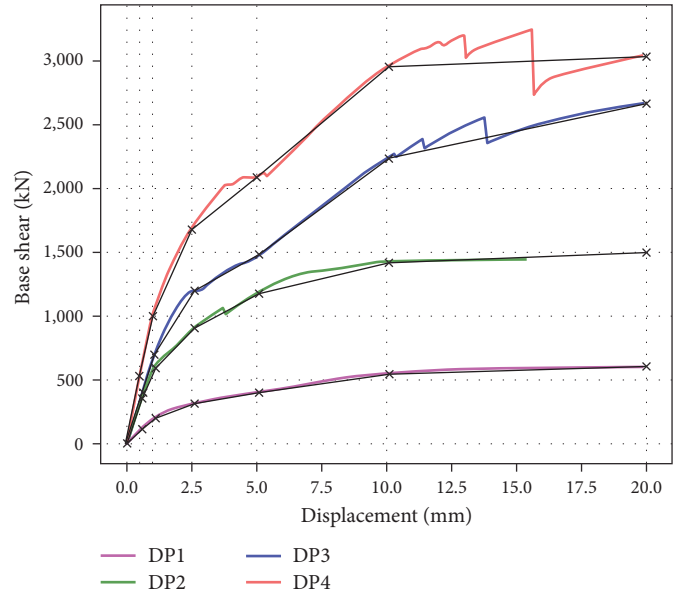
as a ratio of the corresponding concrete cross-section area. The compressive strength f'_c and the yield strength f_y vary within traditional values normally used in the construction of modern buildings. The length l_w and the height h are chosen based on the wall dimensions commonly found in medium-rise buildings. The axial load value q_a is expressed as a ratio of the maximum axial strength for concrete sections according to equation 22.4.2.2 of ACI318-19. The range from 0.010 to 0.1 is selected based on the premise that a value of 0.1 represents the loading of a wall in the bottom story of a medium-height building [2].

2.5. Database Generation. The database is created by randomly generating multiple input vectors with the 11 parameters described in Table 1. Each input vector is processed by a function that creates the corresponding FEM model and performs the static nonlinear pushover analysis. The obtained force–displacement curve is discretized by taking the base shear at seven stations of horizontal displacement. The chosen stations are 0, 0.5, 1.0, 2.5, 5.0, 10.0, and 20.0 mm. Since the base shear at 0 mm is also 0 kN, only six values are required to discretize the curve into six sections. The base shear values at these stations become the corresponding output for the given input values. Therefore, each data point in the database contains 18 values: 11 input values describing the geometry and properties of the shear wall and six output values denoting the base shear at each displacement interval. As an example, Figure 8(a) shows a table with the input and output values for four randomly chosen data points referred to as DP1, DP2, DP3, and DP4. Additionally, Figure 8(b) shows the pushover curves obtained with the FEM analysis and their discretization into six segments.

TABLE 1: Variable parameters are used for the parametric model generation.

No.	Parameter	Lower bound	Upper bound	Description
1	f'_c	25	60	Concrete compressive strength (MPa)
2	f_y	380	600	Reinforcing steel yield stress (MPa)
3	h	300	350	Wall height (cm)
4	t	12.5	40	Wall thickness (cm)
5	l_w	$t \cdot 6$	300	Wall length (cm)
6	l_{be}	$0.15 \cdot l_w$	$0.30 \cdot l_w$	BE length (cm)
7	$\rho_{l_{be}}$	0.01	0.04	BE longitudinal reinforcement ratio
8	$\rho_{t_{be}}$	0.0075	0.015	BE transversal reinforcement ratio
9	$\rho_{l_{web}}$	0.0025	$0.5 \cdot \rho_{l_{be}}$	Web longitudinal reinforcement ratio
10	$\rho_{t_{web}}$	0.0025	$0.5 \cdot \rho_{t_{be}}$	Web transversal reinforcement ratio
11	q_a	0.010	0.1	Axial load ratio

		DP1	DP2	DP3	DP4	Unit	
Input	1	f'_c	34	52	29	52	MPa
	2	f_y	563	506	502	541	MPa
	3	h	308.90	336.42	330.79	318.01	cm
	4	t	23.51	20.15	21.16	28.03	cm
	5	l_w	141.94	219.61	272.67	226.40	cm
	6	l_{be}	0.2595	0.2021	0.2471	0.2323	-
	7	$\rho_{l_{be}}$	0.0164	0.0214	0.0359	0.0374	-
	8	$\rho_{t_{be}}$	0.0114	0.0138	0.0130	0.0105	-
	9	$\rho_{l_{web}}$	0.0065	0.0102	0.0071	0.0164	-
	10	$\rho_{t_{web}}$	0.0057	0.0044	0.0046	0.0052	-
	11	q_a	0.0800	0.0679	0.0932	0.0679	-
Output	1	v1	107.87	350.47	400.74	529.41	kN
	2	v2	194.14	595.59	702.11	995.55	kN
	3	v3	311.20	902.85	1194.23	1677.85	kN
	4	v4	397.38	1180.47	1475.34	2088.64	kN
	5	v5	544.20	1416.09	2238.07	2954.78	kN
	6	v6	598.76	1493.98	2655.64	3028.49	kN



(a)

(b)

FIGURE 8: Four randomly generated data points: DP1, DP2, DP3, and DP4: (a) numerical input and output values; (b) obtained pushover curves and their six-segment discretizations.

For some input values, the pushover analysis does not converge to the target displacement of 20 mm, as seen in the data point DP2 in Figure 8(b). In such cases, the base shear at 20 mm is taken as 1.055 times the base shear at 10 mm to complete the required six output values: $BS(20) = 1.055 \cdot BS(10)$. The factor of 1.055 is taken as the average of $BS(20)/BS(10)$ measured in all the analyses where the convergence does reach the last step of 20 mm. Note that the data points with convergence issues are most likely a result of numerical instabilities typically encountered when solving nonlinear FEM analyses. A complete loss of stiffness due to material failure is not likely to happen in the scenarios tested in this study.

The selected stations at which the curve is discretized (0.5, 1.0, 2.5, 5.0, 10.0, and 20.0 mm) have been selected based on trial and error where the goal is to approximate the curve with sufficient accuracy but with the least number of segments to reduce the complexity of the problem. Alternatively, using

physically meaningful stations, such as the limit states of cracking, yielding, peak capacity, ultimate capacity, and failure, could also be useful. Nevertheless, accurately estimating such transitional states for all the data points is a challenging task that is out of the scope of this study. Instead, the selected stations have shown to be sufficient to obtain a reliable approximation of the actual pushover curve.

3. ANN Surrogate Model

3.1. ANNs Basics. ANNs are a subset of ML and are the basis of most modern artificial intelligence applications. Essentially, an ANN is a mathematical structure that attempts to mimic the complex biological structure and the information processing capacity observed in the brains of living animals. The ANN is composed of several artificial neurons arranged in sequentially connected layers; such structure can be depicted graphically in Figure 9(a).

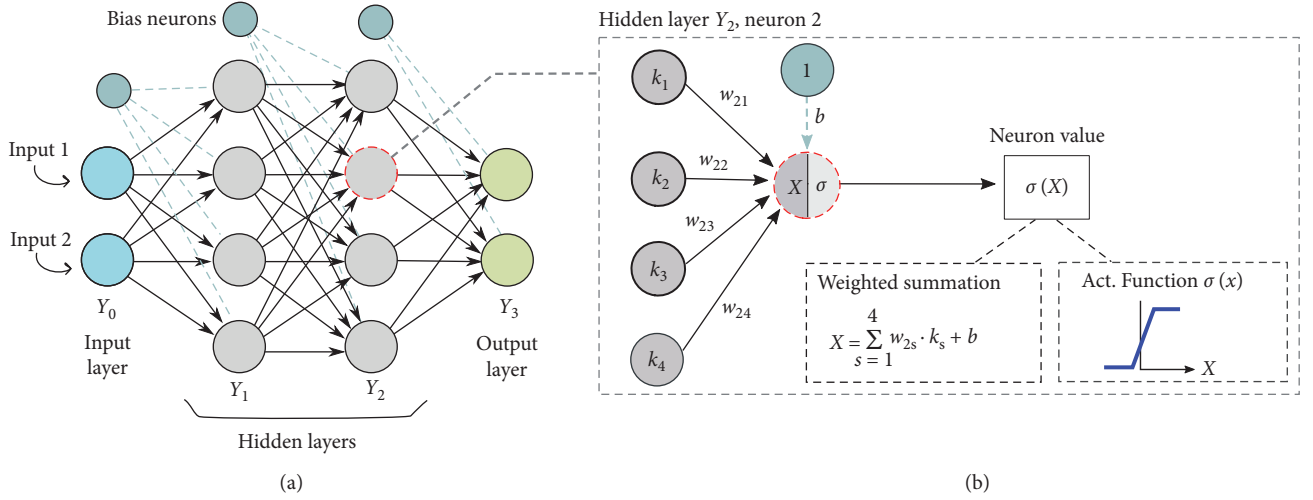


FIGURE 9: ANN graphical representation: (a) connectivity of the layers; (b) detail of a single neuron.

The structure of the ANN can be written in the following form:

$$Y_0 - Y_1 - \dots - Y_{n-1} - Y_n, \quad (1)$$

where the subscript indicates the position of the layer, starting from the input layer $z = 0$ to the output layer $z = n$, so that $z = \{0, 1, \dots, n\}$. The variable Y_z indicates the number of neurons in the z th layer. Following such notation, the ANN depicted in Figure 9(a) could be written as $2 - 4 - 4 - 2$.

The connection between neurons is represented by a numerical value known as weight w . The weight w_{ij} denotes the value for the connection between the neurons i and j , where i belongs to the layer z , and j belongs to the layer $z - 1$. In an ANN, the information flows from the input layer Y_0 to the output layer Y_n . At each layer, Y_z , the value of every neuron is computed by performing the weighted summation depicted in Figure 9(b). Such flow of information can be conveniently written as a matrix multiplication that must be computed layer by layer in a sequential order starting at the layer $z = 1$:

$$Y_z = \sigma_z(W_z Y_{z-1} + b_z), \quad (2)$$

$$W_z = \begin{bmatrix} w_{11} & \dots & w_{1q} \\ \vdots & \ddots & \vdots \\ w_{p1} & \dots & w_{pq} \end{bmatrix}, \quad (3)$$

where variables p and q are local variables that denote the number of neurons in the layers z and $z - 1$, respectively. The function σ_z indicates the activation function for the layer z . Activation functions are used to incorporate nonlinearity to the ANN and restrict the neuron's output value to a specific range. Two commonly used activation functions are the

Sigmoid and the ReLu [46]:

$$\text{ReLU: } \sigma(z) = \begin{cases} 0 & \text{for } z < 0 \\ z & \text{for } z \geq 0 \end{cases}. \quad (4)$$

$$\text{Sigmoid: } \sigma(z) = \frac{1}{1 + e^{-z}}. \quad (5)$$

The letter b in Equation (2) represents the vector that contains the bias values for the neurons in the layer n . The biases are added to increase the flexibility of the ANN by shifting the activation function to the left or right.

The keys to the proper functionality of the ANN are the weights which are also referred to as the model's features or parameters. Normally, the weights are initialized randomly; therefore, the problem becomes in finding their value so that the output Y_n becomes meaningful. The procedure that is followed to find the appropriate value for the weights is a numerical optimization problem referred to as training. It is solved by an iterative process of trial and error using large volumes of data that must be known a priori.

The basic functionality of an ANN could be described with the following mathematical analogy. Suppose that for a given problem $f(x) = y$, the function f is unknown. Then, a function denoted as g , is created to find an approximation $g(w, x) = y_p$ such that $y_p \approx y$. The problem then becomes in finding a value for w that minimizes the error $e = y - y_p$. Such minimization problem is solved by an iterative process of trial an error that utilizes known information. In this analogy, the known information refers to a database T that consists of a number m of pairs of input and output values such as $T = \{x_1, y_1, x_2, y_2, \dots, x_m, y_m\}$. The ultimate goal is that with an appropriate training procedure, the function $g(w, x)$ should reach a state that is capable of extrapolating the information contained in the training data to predict any value that lays in-between.

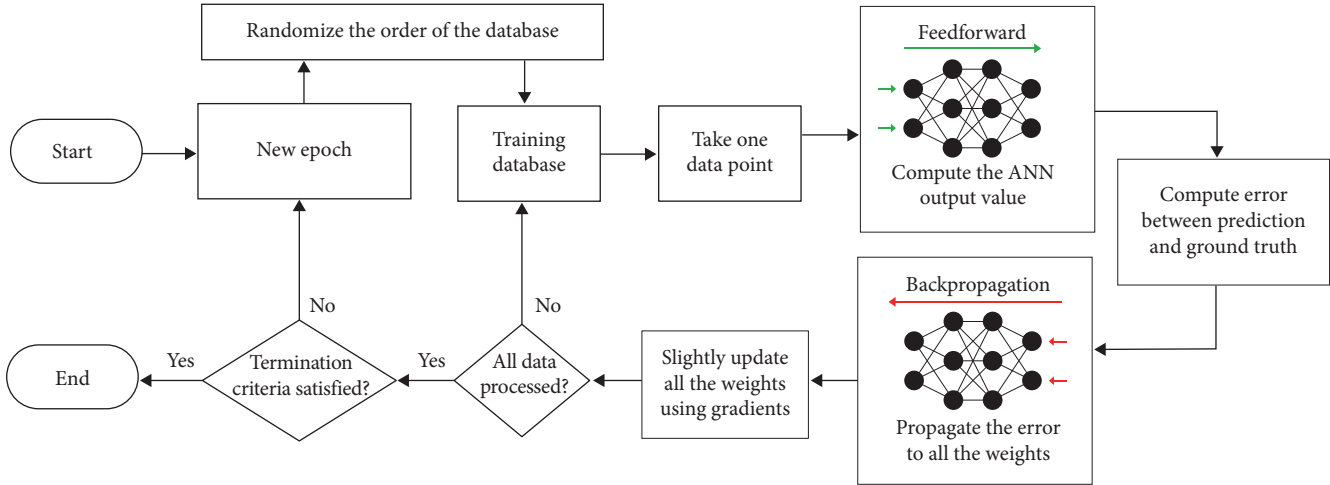


FIGURE 10: Flowchart of the training procedure of an ANN using the stochastic gradient descent and backpropagation algorithm.

One of the most popular training strategies for ANNs is the backpropagation (BP) algorithm applied with the stochastic gradient descent (SGD) updating scheme. The fundamental principle is to compute the total error and propagate it to all the weights in the ANN. Then, a gradient for the error with respect to the model weights is defined, and the weights are adjusted accordingly. A graphical simplification of the procedure is depicted in Figure 10. A complete description of the BP algorithm and its detailed formulation can be found [47].

Every time that the whole training database has been processed by the updating scheme is called an epoch. Depending on the size of the training database and the ANN, several epochs may be required to train the model. The most common stopping criterion is monitoring the error of a small subset of the training database known as the validation data. The validation data is not used in the training process; however, the error in the validation data is measured after every epoch. If the error increases for several consecutive epochs, it may be a signal of overfitting, which may compromise the generalization capabilities of the model. Hence, the training is stopped to prevent the overfitting.

The scheme described in this section corresponds to a type of ANN known as a feedforward neural network or BP neural network (BPNN). The final size and properties of the ANN are commonly known as the hyperparameters. For example, the number of layers, number of neurons per layer, type of activation functions, and type of algorithm for training. Typically, several combinations of hyperparameters are tested, and the configuration that shows the best performance is selected.

3.2. ANN as Surrogate Model for FE Simulations. A surrogate model is a simplified model that approximates the behavior of a more complex model, and it is often used when the complex model is computationally expensive to evaluate [48]. In this work, an ANN is used as a surrogate model of the expensive RC shear wall FEM model described in Section 2. The surrogate model is trained to approximate the characteristic pushover curve with a much lower computational cost.

The data required for the training are generated using the same expensive FEM model that the surrogate model is intended to replace. While this may seem counter-intuitive, it has to be noted that the data creation and the training processes are typically one-time operations that can be done efficiently using techniques such as parallelization and GPU processing. Once the surrogate model is fully trained, it can be used indefinitely as a less computationally intensive alternative.

3.3. Construction of the ANN Surrogate Model. The surrogate model is a fully connected BPNN implemented in Python using the TensorFlow [24] library. Its final configuration is a five-layered network with three hidden layers containing 200 neurons each: 11 – 200 – 200 – 200 – 6. The total number of trainable weights (or features) can be computed as follows: $(11 + 1) \cdot 200 + (200 + 1) \cdot 200 + (200 + 1) \cdot 200 + (200 + 1) \cdot 6 = 84,006$ (where the +1 value corresponds to the bias neuron added to every layer). Due to its size, the proposed ANN falls into the category of deep learning; therefore, it could also be referred to as a DNN [49].

For the activation functions, two types are used. The ReLu function (Equation (4)) is used in the first four layers, and a linear function $\sigma(x) = x$ in the output layer. The weights are initialized by a uniform random distribution from 0 to 1. The selected error function for the training is the mean square error (MSE): $\sum_{i=1}^s (y_i - \hat{y}_i)^2$, where y_i is the ground truth, \hat{y}_i is the predicted output, and s is the number of data points. The adopted optimization scheme is the Adam algorithm [50], which is an improvement to the SGD that considers the exponentially weighted average of the gradients. It is particularly well-suited for large data sets and high-dimensional parameter spaces. The training database contains 2,500 data points generated with the process described in Section 2.5. The input data are normalized to add stability to the ANN. For validation purposes, 10% of the training data are separated and monitored during the training phase. The network is set to run for 200 epochs; however, an early stopping algorithm is introduced to avoid

TABLE 2: Results of various different ANN configurations.

ANN size	Features	MSE	R	R^2
11-25-25-25-6	1,756	29,344	0.9641	0.9298
11-50-50-50-6	6,006	28,088	0.9662	0.9340
11-70-70-75-6	12,756	27,376	0.9659	0.9332
11-100-100-100-6	22,006	27,791	0.9660	0.9331
11-150-150-150-6	48,006	27,905	0.9654	0.9322
11-200-200-200-6	84,006	5,046	0.9919	0.9838

The minimum MSE obtained is marked in bold.

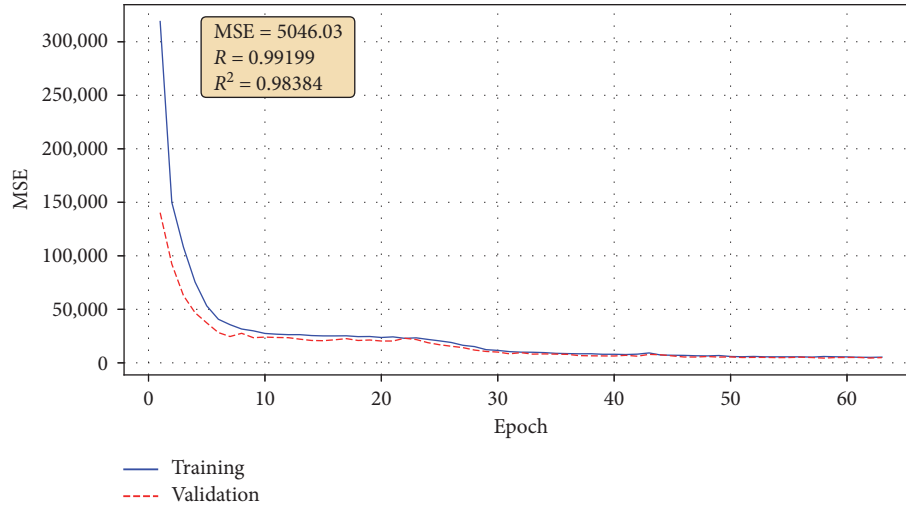


FIGURE 11: ANN surrogate model training history.

overfitting. The stopping criterion is set to terminate the training when the error in the validation database is increased for five consecutive epochs.

The previously described hyperparameters are the final selection after testing several configurations of the ANN. The obtained results of some of those tested variants are presented in Table 2. It can be appreciated that the 11-200-200-200-6 ANN yields the lowest error and better correlation values by far. Despite the relatively large size of the surrogate model with 84,006 features, it is fast enough to produce real-time results, taking less than 1 ms to predict the output for a set of input values on a typical computer. Additionally, its storage on a digital file consumes approximately 1 MB, which is substantially small for a modern computer. Thus, for this particular application, there is no practical advantage in optimizing the hyperparameters any further, as the proposed 11-200-200-200-6 ANN configuration works very well, providing good and quick solutions without becoming extremely large in size.

3.4. Surrogate Model Training and Performance. To create the surrogate model, 2,500 data points are generated as training data. The training process runs for 65 epochs and takes around 30 s to finalize. The evolution of the error throughout the training, for both the validation and training data, is presented in Figure 11.

Two metrics are used to determine the correlation between the predicted data and the ground truth: the

correlation coefficient (R) and the coefficient of determination (R^2). The R score measures the direction and strength of the linear relationship of both variables, while R^2 measures the proportion of variance of the predictions. In the context of ML, these metrics provide a simple way of judging the performance of a regression model. If both R and R^2 have values close to 1, it indicates that the error is low and the model is effective. A comprehensive description of the formulation and the limitations of these metrics is provided [51].

A second database containing 150 data points is generated as a testing set to assess the surrogate model's performance properly. The generated database is entirely new in the sense that it has not been used in the training process; thus, it serves as an effective test to check the generalization capabilities of the model. Such testing set is fed to the trained surrogate model, and the correlation coefficients R and R^2 are computed for each of the six output variables. The results are presented in Figure 12.

It can be noted that there is a good correlation between the predicted data and the ground truth. The resulting average values of the metrics R , and R^2 , are 0.9919, and 0.9838, respectively. Such result hints that the predictions of the model are accurate with high correlation and low error with respect to the ground truth, even for data that were not used in the training. Therefore, the training database has been sufficient for the surrogate model to generalize and learn the corresponding mapping.

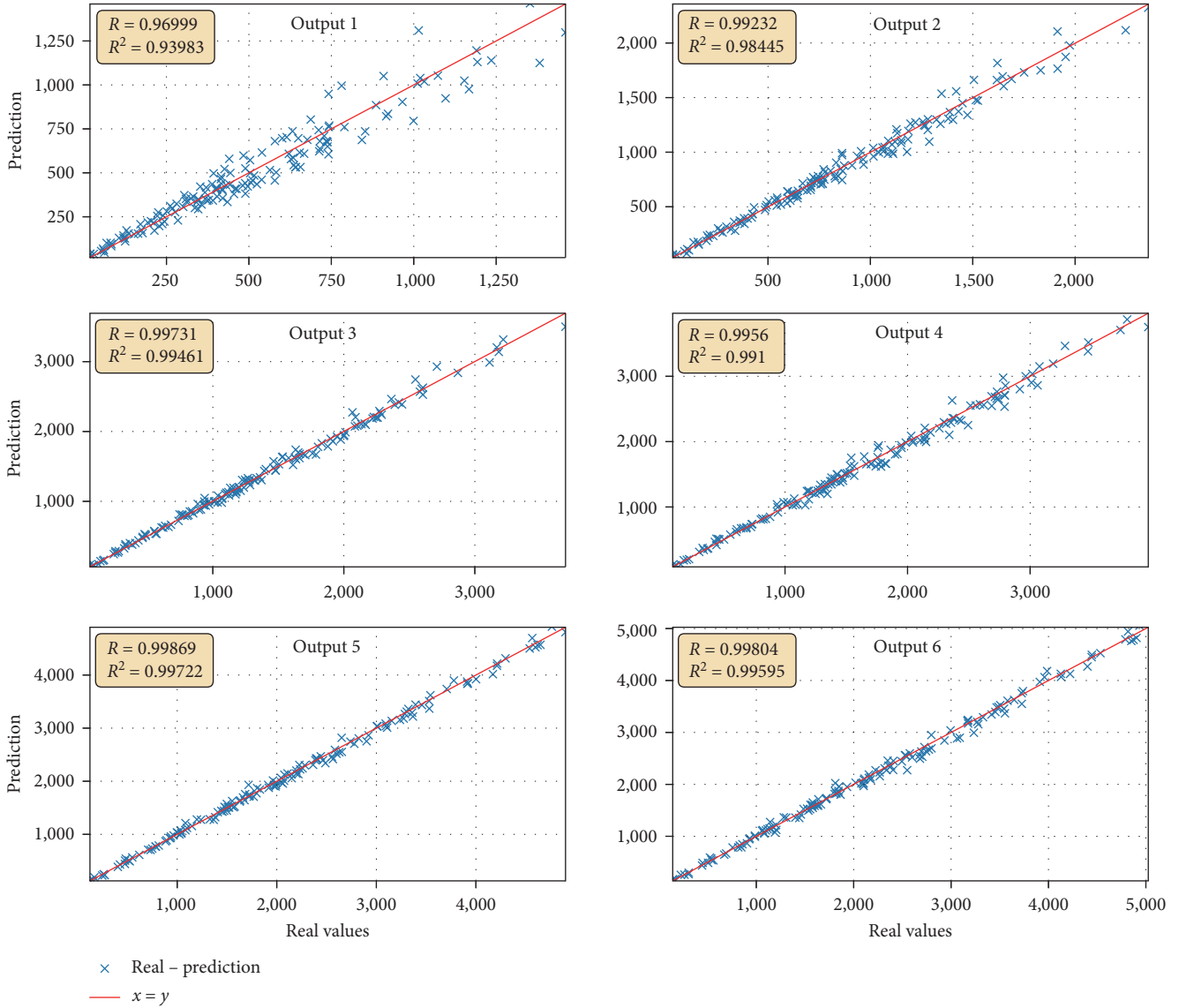


FIGURE 12: Correlation value between the ground truth and the prediction for each of the six output variables of the surrogate model.

3.5. *Numerical Examples.* A numerical example is presented to further corroborate the effectiveness of the surrogate model by picturing the input and output data in an engineering context. The input is generated randomly, and it is rounded to the following values: $f'_c = 55$, $f_y = 430$ MPa, $h = 330$, $t = 35$, $l_w = 250$ cm, $l_{be} = 0.2385$ (or 60 cm), $\rho_{l_{be}} = 0.0229$, $\rho_{t_{be}} = 0.0118$, $\rho_{l_{web}} = 0.0122$, $\rho_{t_{web}} = 0.0063$, $q_a = 0.064$ (or 2,632 kN). The resulting geometry is presented in Figures 13(a) and 13(b). The steel reinforcement depicted in the picture is an approximation using rebar of size $\phi 20$ that corresponds to a diameter of 19.05 mm. The inputs are normalized and then fed to the surrogate model, which outputs the values: $v_1 = 933.6$, $v_2 = 1553.64$, $v_3 = 2359.92$, $v_4 = 2862.27$, $v_5 = 3294.38$, $v_6 = 3358.61$ (units in kN). The six output values v_1 to v_6 correspond to the base shear at 0.5, 1.0, 2.5, 5.0, 10, and 20 mm, respectively. With such information, the obtained pushover curve is plotted in Figure 13(c). The curve obtained using the FEM

model for the same input values is also presented for comparison. There is a close agreement between both curves, indicating that the prediction is substantially accurate.

An additional set of eight input vectors is randomly generated as a final test of the surrogate model. For each input vector, the nonlinear static pushover analysis is performed using the FEM model, and then it is compared with the prediction from the surrogate model. The eight results can be observed in Figure 14. It can be seen by a simple visual inspection that the surrogate model is indeed able to accurately approximate the pushover curve of each analyzed input vector, despite the different shape of the curve in each case.

4. Discussion

4.1. *Advantages.* The main advantages of the developed strategy are summarized as follows:

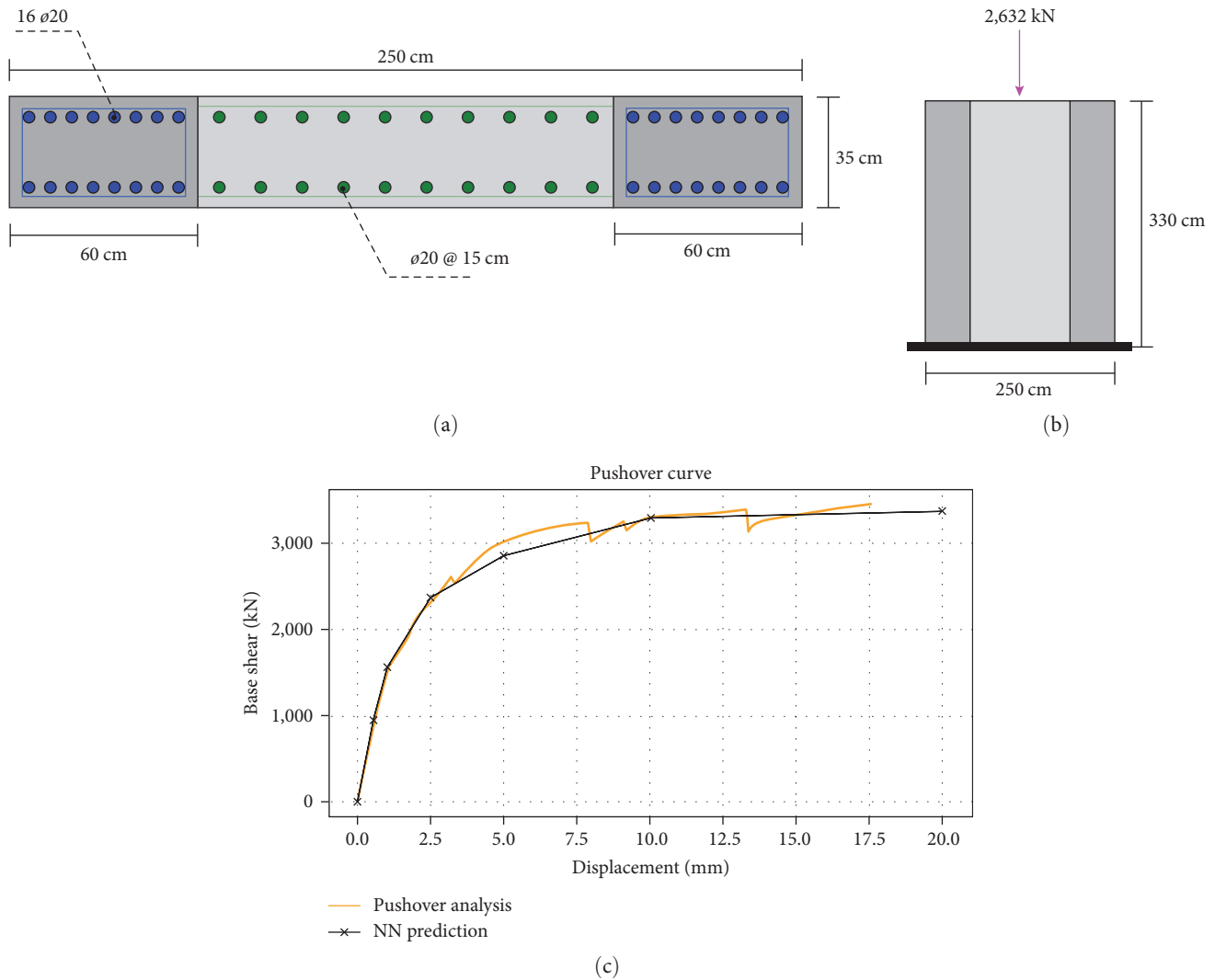


FIGURE 13: Numerical example of the surrogate model: (a) cross-section of the RC shear wall; (b) elevation view of the RC shear wall; (c) comparison between surrogate model and FEM results.

- (i) *Speed*: The main advantage of the surrogate model is that it is extremely fast in comparison to the original FEM model. The achieved time reduction is from an average of 40 s per analysis with the FEM model to 0.025 ms with the surrogate model. The surrogate model is significantly faster than the conventional FEM model.
 - (ii) *Adaptability*: Once the methodology is developed, the surrogate model can be easily extended or modified to tackle different challenges if necessary. For example, changing the input variable bounding values and adding more data to increase the generalization capabilities of the model or modifying the ANN structure to increase its accuracy.
 - (iii) *Portability*: An ANN created with the TensorFlow framework can be easily serialized and utilized across different platforms and programming languages. In contrast, the FE model is usually restricted to a particular software or solver.
 - (iv) *Improved convergence*: Since the ANN is, in principle, a regression model, the entire pushover curve is always predicted, and there are no converge issues that result from numerical instabilities as those encountered when using FEM models. An example can be seen in Figure 14 for the curves DP3 and DP4, where the FEM model failed to converge, but the surrogate model was able to approximate the complete response.
- 4.2. *Limitations*. There are also a few limitations when using an ANN-based surrogate model as a substitute for the FEM model, such as the well-known tradeoff of accuracy with respect to the exact FEM simulation. The accuracy of the ANN's predictions will depend on its complexity and the quality of the training. If the training data do not represent the full range of the input parameters or if the ANN is not complex enough to capture the relationships between the inputs and outputs, the accuracy of the predictions may be

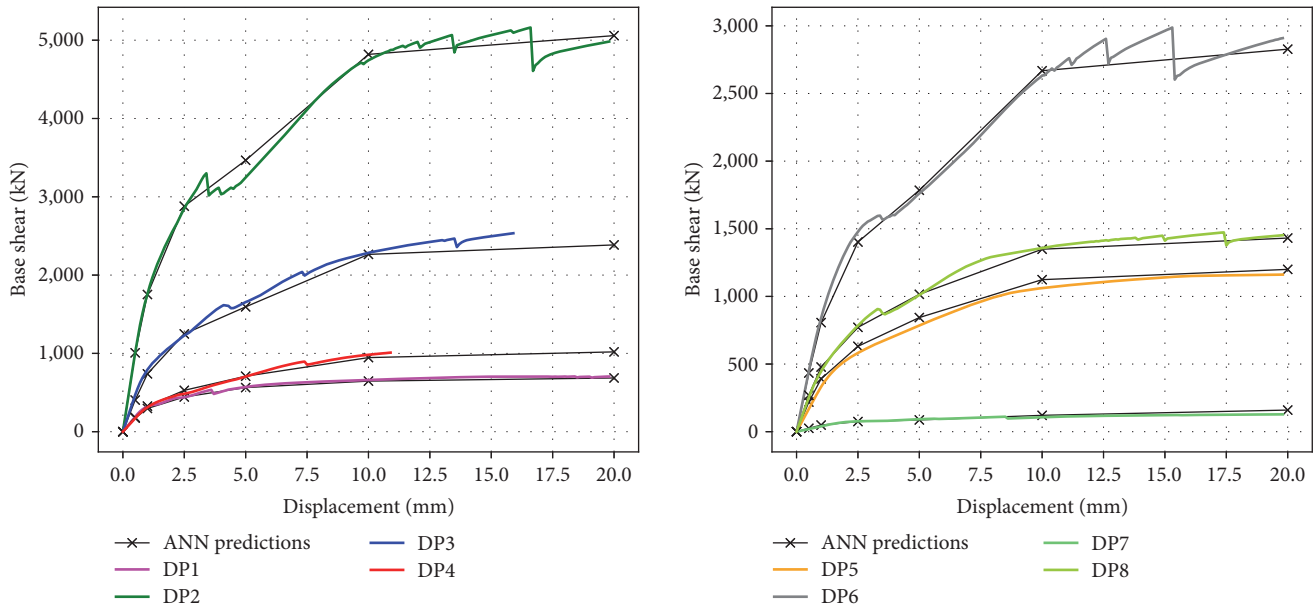


FIGURE 14: Comparison of the surrogate model and the FEM results for eight randomly-generated input vectors.

poor. Additionally, ANNs are commonly defined as “black box” models, meaning that it is difficult to interpret how they arrived at their predictions, making it challenging to understand the system’s underlying physics. However, in this case, this is not an issue as the physics of the problem is known from the detailed FEM model; the ANN is only used to increase the speed of the final solution.

4.3. Computational Efficiency. All the operations, including the solutions of the FEM models, the training and testing of the ANN, and the generation of the database are conducted using a regular PC with an Intel Core i7-6700 HQ CPU @2.60 GHz and 16 GB RAM.

The static nonlinear pushover analysis using the FEM model takes, on average, 41.3 s to finalize. In contrast, loading the trained ANN surrogate model from a file takes 0.05 s. Once the ANN is loaded, predicting the output for a set of input values takes 0.000025 s. This amounts to an improvement of 1,652,000 times in terms of computational time needed for a nonlinear analysis. Nonetheless, it is not fair to compare the models without taking into account the considerable computational time and effort that is required to generate the training data. Considering the 41.3 s average of each FEM analysis, the generation of the 2,500 data points required for the training take approximately 28.68 hr to complete. However, the generation of the training database is a process that has to be done only once. Furthermore, the data may be generated using parallel processing or high-performance computing systems which can further significantly reduce the time required. After the model is trained, it is ready to be used indefinitely, providing results at lightning speed.

4.4. Database and Sampling. As stated in Section 2.5, the bounding values of the parameters and the sampling method are two fundamental aspects of the developed methodology.

In that regard, even though the bounding values are carefully selected based on structural engineering criteria, the selected sampling scheme may still produce some unrealistic wall geometries. For example, a data point where the minimum vertical reinforcement ratio is selected together with the maximum transverse reinforcement. Such a scenario is uncommon in RC shear walls, and it is actually unrealistic. Unfortunately, there is no trivial solution to this problem. One simple alternative is to hard-code some rules to impose certain criteria into the data generation process. However, this is usually not a clever idea because despite the fact that some of the generated geometries may not be practical from an engineering or economic perspective, they may still be compliant with the building design code. Thus, this can introduce some bias into the predictive model leading to a compromise in its performance. Additionally, adding certain rules in the data generation could turn some parameters into discrete values rather than continuous variables, which may have a negative impact in the accuracy of the model.

Other sampling strategies may improve the data generation strategy, such as using the Latin hypercube sampling (LHS) method [52]. Nonetheless, it would most-likely not improve much the accuracy or performance of the data-driven model. At best, it would only reduce the amount of data required. With that in mind, for the present study, the authors have chosen to keep a clean and simple strategy, that is, generate the data randomly in a continuous way within specific bounds. It is fundamental to keep in mind that data-driven methodologies are normally nonstatic, meaning that the available data are usually evolving, and the models are retrained and improved constantly. In any case, the sampling method is not restricted to the strategy followed by the authors. In fact, changing the sampling strategy from a uniform random distribution to a more advanced scheme, such as the LHS, could be achieved by changing a few lines of code

in the provided open-source framework that is introduced in Section 5.

4.5. Applications and Future Work. To visualize some of the potential applications, it is important to keep in mind that the developed surrogate model is not meant to fully replace the traditional FE modeling of RC shear walls, which follows a strong theoretical framework built on decades of research. Instead, it is envisioned as a powerful strategy that could replace a computationally expensive model with a significantly faster approximation for certain types of applications. For instance, for the optimal design of RC shear walls [53], failure probability analysis, uncertainty quantification, and risk assessment [54]. These applications usually involve running a large number of analyses, sometimes hundreds of thousands, with different parameters and conditions in order to estimate risk levels, failure probabilities, or optimal configurations. Therefore, in such scenarios, having a simpler but much faster model is preferred over a highly detailed computationally expensive one.

Another potential application is using the surrogate model to improve the macroscopic modeling technique for the analysis and design of buildings. That may be achieved by using the predicted pushover curve as the force–displacement relation in the horizontal shear spring of a macro model. This can increase the effectiveness and accuracy of the macro modeling approach [55].

There are various directions in which the methodology described in this study could be continued or enhanced with further research. A potential improvement could be achieved by increasing the input variables bounding ranges or adding more variables to describe other properties of the RC shear wall to significantly increase the surrogate model's generalization capabilities. Alternatively, increasing the amount of output information so that the surrogate model is made capable of predicting additional results, such as the stresses, strains, or any other relevant output quantity that can be extracted from the FEM model. Similarly, increasing the number of stations in which the curve is discretized to obtain a more refined discretization could also be easily implemented.

5. The Open-Source Framework

The full framework that has been developed to create the surrogate model is published as an open-source project in the GitHub platform under the following link: <https://github.com/germansr/RC.SheerWall.DNN.SurrogateModel>.

The project includes (1) the parametric code for the multilayered-shell FEM model of the RC shear wall in OpenSeesPy (it contains various 2D visualization routines, such as those used to create Figures 5 and 6). (2) A script that takes the parametric FEM model to create random RC shear wall geometries runs the corresponding nonlinear static pushover analyses and stores the results on a text file. (3) A routine that cleans the results and creates a functional database with the required 11 inputs and six outputs for the training of the surrogate model. (4) A script to create, train, and test the ANN-based surrogate model using TensorFlow and other

scientific libraries. (5) A powerful application with an interactive GUI to analyze RC shear walls using OpenSees and to test the surrogate model in real-time with neat visualization graphics. (6) Other useful scripts for testing and visualization.

The project is entirely developed in Python and consists of thousands of lines of code that are well-structured, modularized in several files, and extensively commented. We strongly believe that the project can serve as a motivation for other researchers and professionals to explore the advantages of using ML and data-driven solutions in structural engineering applications.

5.1. Visualizing the Results of the Surrogate Model. When dealing with ML-oriented methodologies, the physical concepts behind the problem being solved may not be entirely clear and could easily get lost in the data. Therefore, providing visual feedback could significantly enhance the experience and efficacy when using ML-based methodologies. For these reasons, a simple yet powerful interactive application has been developed to test and visualize the results of the surrogate model; see Figure 15.

The surrogate model has to be trained and serialized to a text file before using the application. When the application is initialized, the surrogate model is loaded from the text file and used to make real-time predictions based on the current values of the input variables, which are controlled by 11 interactive components known as sliders, see Figure 15(a). The sliders provide a simple way of changing the input values within the upper and lower bounds of the training data. Every time that the user modifies the value of any slider, the displayed geometry (Figures 15(a) and 15(b)) is refreshed, and the surrogate model is fired in the background to update the corresponding pushover curve almost instantly (Figure 15(d)).

The application also includes an easy way of testing the predictions by incorporating a button that runs the pushover analysis in the background. Note that the FEM analysis is not required to obtain the predictions, and the optional analysis only serves as a means to demonstrate the accuracy of the results for any given set of inputs. The analysis is performed using the same FEM model that is used to generate the training data, which has been described in this paper in Section 2.3. A small loading screen is added to indicate that the analysis is running. Upon finalizing, the model is temporarily locked (the input values cannot be updated), and the analytical pushover curve is plotted and compared to the surrogate model's current prediction. Additionally, when the model is analyzed, a small module can be called to visualize the FEM analysis results in more detail; see Figure 15(e). The model can be unlocked at any time to analyze different cases. The GUI is constructed using the Tkinter library, typically included in the regular Python distribution.

6. Conclusion and Outlook

6.1. Conclusion. In this study, a deep-learning surrogate model using DNNs has been developed for the analysis of RC shear walls. The surrogate model is capable of approximating the characteristic nonlinear curve that is obtained when a lateral pushover analysis is performed using a

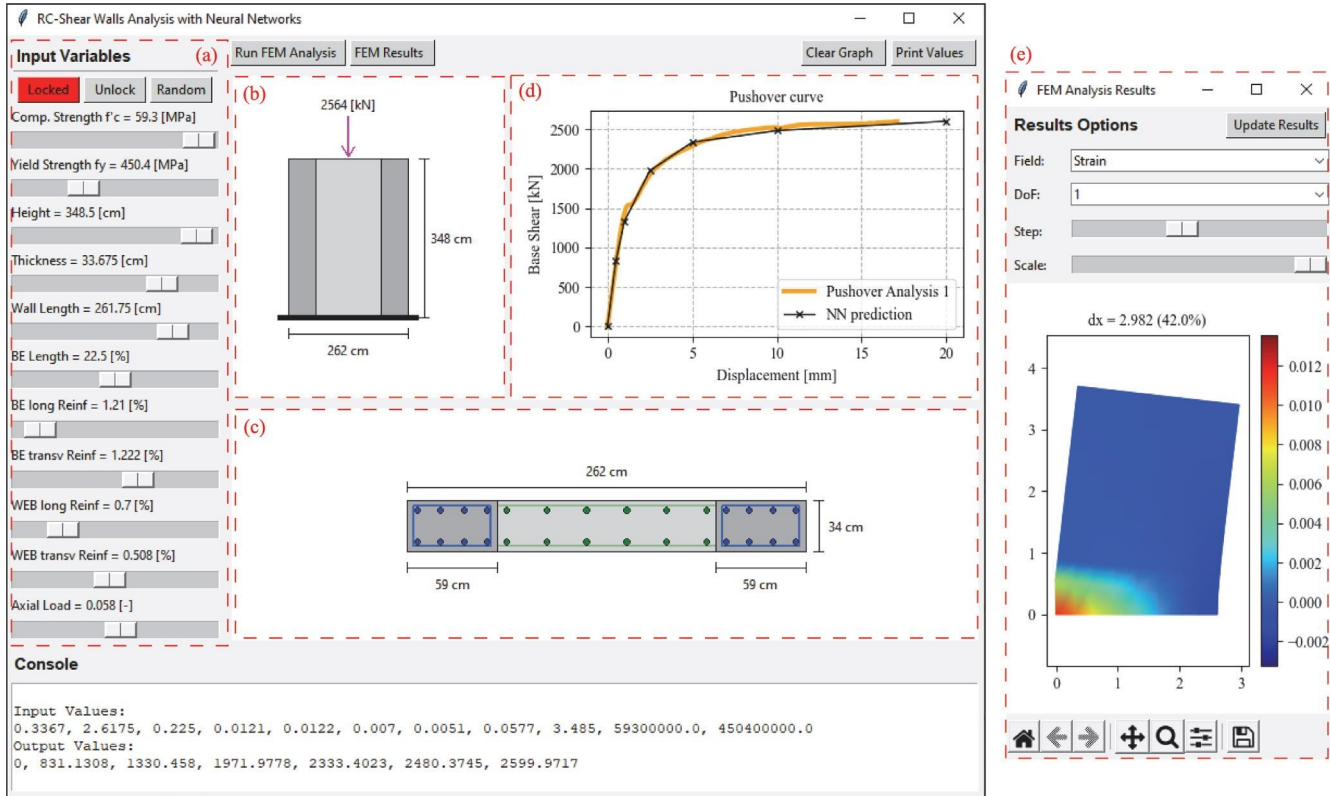


FIGURE 15: Main screen of the interactive application was developed to test and visualize the surrogate model results.

conventional FEM analysis. The DNN-based surrogate model is constructed in such a way that it takes as input 11 values describing the RC shear wall's geometry, properties, and loading conditions. The output is a vector of six values corresponding to the nonlinear pushover curve discretized into six segments. The final configuration of the DNN is a BPNN architecture composed of 11 input nodes, three hidden layers with 150 nodes each, and six output nodes. The data used for the training have been generated with a sophisticated FEM model using multi-layered shell elements and state-of-the-art constitutive laws for the reinforcement steel and concrete materials. A total of 2,500 data points were generated for the training database, which took about 13 hr using parallel processing. After a successful training procedure, the surrogate model has been thoroughly tested and compared to the FEM model in several examples. The obtained results have shown good accuracy and a strong correlation between the predicted data and the FEM simulations, thus, indicating that the ANN-based surrogate model is indeed a viable and computationally efficient alternative. The most significant advantage is the speed of the surrogate model, which takes roughly 0.025 ms to run in contrast to the 40 s that the FEM analysis takes to complete. Thus, the improvement in speed is an impressive factor of 1.6 million times faster than the nonlinear FEM analysis. Additionally, another advantage is that there are no convergence problems as those encountered when using nonlinear FEM procedures. With the surrogate model, the entire pushover curve is always predicted.

Another significant contribution of this paper is that the complete source code and framework that made this study possible has been made available as an open-source project, along with an interactive application to test and visualize the results. We hope that this approach facilitates further research and development in the area and contributes to narrowing the gap between research-oriented and practical applications of ML-powered methodologies in the field of structural engineering.

Data Availability

The full framework used to support the findings of this study has been deposited in the GitHub repository as an open-source project under the following link: <https://github.com/germansr/RC.ShearWall.DNN.SurrogateModel>.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

References

- [1] M. Fintel, "Performance of buildings with shear walls in earthquakes of the last thirty years," *PCI Journal*, vol. 40, no. 3, pp. 62–80, 1995.
- [2] I. D. Lefas, M. D. Kotsovos, and N. N. Ambraseys, "Behavior of reinforced concrete structural walls: strength, deformation characteristics, and failure mechanism," *ACI Structural Journal*, vol. 87, no. 1, pp. 23–31, 1990.

- [3] R. G. Oesterle, J. D. Aristizabal-Ochoa, A. E. Fiorato, H. G. Russell, and W. G. Corley, *Earthquake Resistant Structural Walls—Test of Isolated Walls Phase II*, Construction Technology Laboratories, California, USA, 1979.
- [4] X. L. Lu, Y. Zhou, J. H. Yang, J. Qian, C. Song, and Y. Wang, “SLDRCE database on static tests of structural members and joint assemblies—shear walls R11,” Tech. Rep., Institute of Structural Engineering and Disaster Reduction, Tongji University, Shanghai, China, Shanghai, China, 2008.
- [5] A. Jalali and F. Dashti, “Nonlinear behavior of reinforced concrete shear walls using macroscopic and microscopic models,” *Engineering Structures*, vol. 32, no. 9, pp. 2959–2968, 2010.
- [6] K. Kolozvari, K. Orakcal, and J. W. Wallace, “New openses models for simulating nonlinear flexural and coupled shear-flexural behavior of RC walls and columns,” *Computers & Structures*, vol. 196, pp. 246–262, 2018.
- [7] A. Vulcano, V. V. Bertero, and V. Colotti, “Analytical modeling of R/C structural walls,” in *Proceedings of the 9th World Conference on Earthquake Engineering*, vol. 6, pp. 41–46, International Association of Earthquake Engineering, Tokyo/Kyoto, Japan, 1988.
- [8] T. Takayanagi and W. C. Schnobrich, “Non-linear analysis of coupled wall systems,” *Earthquake Engineering & Structural Dynamics*, vol. 7, no. 1, pp. 1–22, 1979.
- [9] X. Lu and Y. Chen, “Modeling of coupled shear walls and its experimental verification,” *Journal of Structural Engineering*, vol. 131, no. 1, pp. 75–84, 2005.
- [10] V. Colotti, “Shear behavior of RC structural walls,” *Journal of Structural Engineering*, vol. 119, no. 3, pp. 728–746, 1993.
- [11] K. Kolozvari, C. Arteta, M. Fischinger et al., “Comparative study of state-of-the-art macroscopic models for planar reinforced concrete walls,” *ACI Structural Journal*, vol. 115, no. 6, pp. 1637–1657, 2018.
- [12] S. M. R. Mortazavi and B. Zaeimdar, “Shear wall modeling with asymmetric openings by macro elements,” *Structures*, vol. 29, pp. 899–910, 2021.
- [13] K. Orakcal and J. W. Wallace, “Flexural modeling of reinforced concrete walls—experimental verification,” *ACI Structural Journal*, vol. 103, no. 2, pp. 196–206, 2006.
- [14] İ. Kazaz, A. Yakut, and P. Güllkan, “Numerical simulation of dynamic shear wall tests: a benchmark study,” *Computers & Structures*, vol. 84, no. 8-9, pp. 549–562, 2006.
- [15] O. Saghalian, F. Nateghi, and O. Rezaifar, “Comparison of using different modeling techniques on prediction of the nonlinear behavior of R/C shear walls,” *International Journal of Engineering*, vol. 27, no. 2, pp. 269–282, 2014.
- [16] F.-Y. Liao, L.-H. Han, and Z. Tao, “Performance of reinforced concrete shear walls with steel reinforced concrete boundary columns,” *Engineering Structures*, vol. 44, pp. 186–209, 2012.
- [17] L. Xie, X. Lu, X. Lu, Y. Huang, and L. Ye, “Multi-layer shell element for shear walls in OpenSees,” in *Computing in Civil and Building Engineering*, pp. 1190–1197, American Society of Civil Engineers, 2014.
- [18] J. R. de Barros Silva and B. Horowitz, “Nonlinear finite element analysis of reinforced concrete shear walls,” *Revista IBRACON de Estruturas e Materiais*, vol. 13, no. 6, Article ID e13603, 2020.
- [19] J. Ghaboussi, “Advances in neural networks in computational mechanics and engineering,” in *Advances of Soft Computing in Engineering*, Z. Waszczyszyn, Ed., vol. 512 of *CISM International Centre for Mechanical Sciences*, pp. 191–236, Springer, Vienna, 2010.
- [20] G. Solorzano and V. Plevris, “Computational intelligence methods in simulation and modeling of structures: a state-of-the-art review using bibliometric maps,” *Frontiers in Built Environment*, vol. 8, 2022.
- [21] A. Ahmad, V. Plevris, and Q.-u.-Z. Khan, “Prediction of properties of FRP-confined concrete cylinders based on artificial neural networks,” *Crystals*, vol. 10, no. 9, Article ID 811, 2020.
- [22] M. E. A. Ben Seghier, J. A. F. O. Corriea, J. Jafari-Asl, A. Malekjafarian, V. Plevris, and N.-T. Trung, “On the modeling of the annual corrosion rate in main cables of suspension bridges using combined soft computing model and a novel nature-inspired algorithm,” *Neural Computing and Applications*, vol. 33, pp. 15969–15985, 2021.
- [23] M. Zhu, F. McKenna, and M. H. Scott, “OpenSeesPy: Python library for the OpenSees finite element framework,” *SoftwareX*, vol. 7, pp. 6–11, 2018.
- [24] M. Abadi, A. Agarwal, P. Barham et al., *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*, Cornell University, 2016.
- [25] G. Solorzano and V. Plevris, “ANN-based surrogate model for predicting the lateral load capacity of RC shear walls,” in *The 8th European Congress on Computational Methods in Applied Sciences and Engineering*, ECCOMAS Congress, Oslo, Norway, 2022.
- [26] M. S. Barkhordari and M. Tehranizadeh, “Response estimation of reinforced concrete shear walls using artificial neural network and simulated annealing algorithm,” *Structures*, vol. 34, pp. 1155–1168, 2021.
- [27] M. J. Moradi and M. A. Hariri-Ardebili, “Developing a library of shear walls database and the neural network based predictive meta-model,” *Applied Sciences*, vol. 9, no. 12, Article ID 2562, 2019.
- [28] R. Quevedo-Reina, G. M. Álamo, L. A. Padrón, and Juan J. Aznárez, “Surrogate model based on ANN for the evaluation of the fundamental frequency of offshore wind turbines supported on jackets,” *Computers & Structures*, vol. 274, Article ID 106917, 2023.
- [29] H. T. Mai, J. Kang, and J. Lee, “A machine learning-based surrogate model for optimization of truss structures with geometrically nonlinear behavior,” *Finite Elements in Analysis and Design*, vol. 196, Article ID 103572, 2021.
- [30] D. W. Abueidda, S. Koric, and N. A. Sobh, “Topology optimization of 2D structures with nonlinearities using deep learning,” *Computers & Structures*, vol. 237, Article ID 106283, 2020.
- [31] F. F. Taucer, E. Spacone, and F. C. Filippou, *A fiber beam-column element for seismic response analysis of reinforced concrete structures*, Ph.D. thesis, University of California, Berkeley, 1991.
- [32] P. Hallinan and H. Guan, “Layered finite element analysis of one-way and two-way concrete walls with openings,” *Advances in Structural Engineering*, vol. 10, no. 1, pp. 55–72, 2007.
- [33] H. Guan and Y.-C. Loo, “Flexural and shear failure analysis of reinforced concrete slabs and flat plates,” *Advances in Structural Engineering*, vol. 1, no. 1, pp. 71–85, 1997.
- [34] F. McKenna, M. H. Scott, and G. L. Fenves, “Nonlinear finite-element analysis software architecture using object composition,” *Journal of Computing in Civil Engineering*, vol. 24, no. 1, pp. 95–107, 2010.
- [35] X. Lu, L. Xie, H. Guan, Y. Huang, and X. Lu, “A shear wall element for nonlinear seismic analysis of super-tall buildings using OpenSees,” *Finite Elements in Analysis and Design*, vol. 98, pp. 14–25, 2015.

- [36] L. Jiang, M. W. Chernuka, and N. G. Pegg, "A co-rotational, updated Lagrangian formulation for geometrically nonlinear finite element analysis of shell structures," *Finite Elements in Analysis and Design*, vol. 18, no. 1–3, pp. 129–140, 1994.
- [37] Z. P. Bažant and J. Planas, "Crack band models and smeared cracking," in *Fracture and Size Effect in Concrete and Other Quasibrittle Materials*, pp. 1–47, Routledge, 1st edition, 1998.
- [38] A. Hillerborg, M. Modéer, and P.-E. Petersson, "Analysis of crack formation and crack growth in concrete by means of fracture mechanics and finite elements," *Cement and Concrete Research*, vol. 6, no. 6, pp. 773–781, 1976.
- [39] M. Menegotto, "Method of analysis of cyclically loaded RC plane frames including changes in geometry and non-elastic behavior of elements under normal force and bending," pp. 15–22, 1973, Zurich, Switzerland.
- [40] F. C. Filippou, E. P. Popov, and V. V. Bertero, "Effects of Bond Deterioration on Hysteretic Behavior of Reinforced Concrete Joints," Tech. Rep. OCLC: 65472460, Earthquake Engineering Research Center, University of California, Berkeley, California, 1983.
- [41] S. P. Shah, S. E. Swartz, and C. Ouyang, *Fracture Mechanics of Concrete: Applications of Fracture Mechanics to Concrete, Rock and Other Quasi-Brittle Materials*, John Wiley & Sons, 1995.
- [42] D. C. Kent and R. Park, "Flexural members with confined concrete," *Journal of the Structural Division*, vol. 97, no. 7, pp. 1969–1990, 1971.
- [43] J. B. Mander, M. J. N. Priestley, and R. Park, "Theoretical stress-strain model for confined concrete," *Journal of Structural Engineering*, vol. 114, no. 8, pp. 1804–1826, 1988.
- [44] M. Y. H. Bangash, *Concrete and Concrete Structures: Numerical Modelling and Applications*, Elsevier Applied Science, London, United Kingdom, 1989.
- [45] ACI Committee 2019, "ACI 318-19 building code requirements for structural concrete," Farmington Hills, MI, 2019.
- [46] G. Solorzano and V. Plevris, "Design of reinforced concrete isolated footings under axial loading with artificial neural networks," in *14th International Conference on Evolutionary and Deterministic Methods for Design, Optimization and Control*, pp. 118–131, ECCOMAS Proceedia, Streamed from Athens, Greece, 2021.
- [47] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, 1986.
- [48] J. Kudela and R. Matousek, "Recent advances and applications of surrogate models for finite element method computations: a review," *Soft Computing*, vol. 26, pp. 13709–13733, 2022.
- [49] L. Alzubaidi, J. Zhang, A. J. Humaidi et al., "Review of deep learning: concepts, CNN architectures, challenges, applications, future directions," *Journal of Big Data*, vol. 8, Article ID 53, 2021.
- [50] D. P. Kingma and J. Ba, "Adam: a method for stochastic optimization," arXiv: 1412.6980 [cs], 2017.
- [51] V. Plevris, G. Solorzano, N. Bakas, and M. E. A. Ben Seghier, "Investigation of performance metrics in regression analysis and machine learning-based prediction models," in *The 8th European Congress on Computational Methods in Applied Sciences and Engineering*, pp. 1–26, ECCOMAS Congress, Oslo, Norway, 2022.
- [52] K. Nasrollahzadeh, M. A. Hariri-Ardebili, H. Kiani, and G. Mahdavi, "An integrated sensitivity and uncertainty quantification of fragility functions in RC frames," *Sustainability*, vol. 14, no. 20, Article ID 13082, 2022.
- [53] G. Solorzano and V. Plevris, "Optimum design of RC footings with genetic algorithms according to ACI 318-19," *Buildings*, vol. 10, no. 6, Article ID 110, 2020.
- [54] C. L. Segura Jr., S. Sattar, and M. A. Hariri-Ardebili, "Quantifying material uncertainty in seismic evaluations of reinforced concrete bridge column structures," *ACI Structural Journal*, vol. 119, no. 3, pp. 141–152, 2022.
- [55] G. Solorzano and V. Plevris, "DNN-MLVEM: a data-driven macromodel for RC shear walls based on deep neural networks," *Mathematics*, vol. 11, no. 10, Article ID 2347, 2023.