DOI: 10.1111/mice.13212

RESEARCH ARTICLE

COMPUTER-AIDED CIVIL AND INFRASTRUCTURE ENGINEERING WILEY

Crack pattern–based machine learning prediction of residual drift capacity in damaged masonry walls

Mauricio Pereira[1] | **Antonio Maria D'Altri**[1,2] | **Stefano de Miranda**[2] | **Branko Glisic**[1]

[1]Department of Civil and Environmental Engineering, Princeton University, New Jersey, USA

[2]Department of Civil, Chemical, Environmental, and Materials Engineering, University of Bologna, Bologna, Italy

Correspondence
Branko Glisic, Department of Civil and Environmental Engineering, Princeton University, Princeton, NJ 08240, USA.
Email: bglisic@princeton.edu

Funding information
Horizon 2020 - Marie Skłodowska-Curie Actions, Grant/Award Number: 101029792

Abstract

In this paper, we present a method based on an ensemble of convolutional neural networks (CNNs) for the prediction of residual drift capacity in unreinforced damaged masonry walls using as only input the crack pattern. We use an accurate block-based numerical model to generate mechanically consistent crack patterns induced by external actions (earthquake-like loads and differential settlements). For a damaged masonry wall, we extract the crack width cumulative distribution, we derive a crack width exceedance curve (CWEC), and we evaluate the drift loss (DL) with respect to the undamaged wall. Numerous pairs of CWEC and DL are thus generated and used for training (and validating) an ensemble of CNNs generated via repeated *k*-folding cross validation with shuffling. As a result, a method for damage prognosis (Level IV of SHM) is provided. Such method appears general, inexpensive, and able to adequately predict the DL using as only input the CWEC, providing real-time support for decision making in damaged masonry structures.

1 | INTRODUCTION

Cracks in masonry structures are frequent. On the one hand, masonry has a well-known quasi-brittle behavior characterized by weak interfaces in the mortar joints (low cohesion). On the other hand, crack patterns are typically induced by seismic events (even with low intensity), foundation settlements (Napolitano & Glisic, 2020), remodel works, materials aging, and so forth. In general, the cause of damage might be known (e.g., due to an earthquake; Giaretton et al., 2016) or it could be (initially) unknown, and damage diagnostics (e.g., see D'Altri et al., 2023; Napolitano & Glisic, 2019) should be employed to understand the reasons of damage.

The presence of cracks in a masonry structure could have considerable negative effects on the structural behavior. Accordingly, damage inspection, identification, and structural health monitoring (SHM) of masonry and historic structures are nowadays main concerns for assets owners (Boscato et al., 2015; Eltouny & Liang, 2021; Ierimonti et al., 2023; Riveiro et al., 2016; Wang et al., 2018).

In the last decade, a great development has been achieved in the automatic crack detection of buildings by using images and/or point clouds (Dais et al., 2021; Dang et al., 2022; Hallee et al., 2021; Katsigiannis et al., 2023; Loverdos & Sarhosis, 2022; Malek et al., 2023; Meng et al., 2023; Ni et al., 2019; Quqa et al., 2023; Stałowska et al., 2022; Yang et al., 2018). Such development has been clearly powered by the use of machine learning (ML) algorithms toward SHM, which opened a new frontier in the damage inspection of buildings. As a result, very advanced crack detection procedures can be already found in the recent literature, also specifically developed for masonry structures.

Even though considerable advances have been lately carried out in crack pattern detection and identification, the quantification of the impact of the crack pattern on the structural and seismic response of a masonry structure is still very challenging to determine (Asjodi & Dolatshahi, 2022) and typically unknown, unless very simplistic and rough approaches are used (e.g., see FEMA, 1998). To our best knowledge, the only step forward in this direction has been very recently made by Rezaie et al. (2022) for rubble stone masonry piers. In particular, six quasi-static cyclic shear-compression tests on plastered rubble stone masonry piers (Dolatshahi & Beyer, 2022; Rezaie, Achanta et al., 2020; Rezaie, Godio et al., 2020; Rezaie et al., 2021) were used in Rezaie et al. (2022) to build a predictive ML model for stiffness degradation, residual strength, and residual displacement capacity starting from the crack pattern of the piers identified through digital image correlation (DIC) in terms of three features: maximum crack width, crack length density, and complexity dimension. Although the research in Rezaie et al. (2022) was aimed to quantify the loss of seismic performance of walls due to the presence of cracks induced by seismic actions, the generalization of such results to other masonry types, boundary conditions, and wall sizes is nontrivial, as pointed out by the authors, and further expensive experimental tests might be required to extend the applicability.

A potential way to overcome this issue could be to use accurate and detailed numerical models to generate crack patterns in masonry walls. Although there are still challenges to be addressed (e.g., computational effort, effective parameter calibration, dealing with imperfect boundary conditions, dynamic effects), numerical modeling strategies for masonry structures have shown a remarkable development in the last decades (D'Altri et al., 2020). In this framework, block-based approaches (Abdulla et al., 2017; Angiolilli et al., 2021; D'Altri et al., 2019; Ferrante et al., 2021; Lourenço & Rots, 1997; Macorini & Izzuddin, 2011; Pantò et al., 2022; Petracca et al., 2023; Serpieri et al., 2017) can be very accurate (like micromechanics-based equivalent continua (Addessi et al., 2002; De Bellis & Addessi, 2011; Trovalusci & Masiani, 2003), as they can explicitly account for the actual masonry pattern and, consequently, automatically embed the anisotropic nature of masonry and the numerous failure modes of the material.

The aim of this research is to perform damage prognosis (Level IV SHM; Rytter, 1993) by predicting the residual displacement capacity of unreinforced damaged masonry walls from the crack pattern. Such research can thus provide fundamental support to the structural analysis of full-scale cracked masonry structures, for example, in the framework of equivalent frame modeling (D'Altri et al., 2020) where the structure is idealized as a frame composed of piers and spandrels and an ultimate drift capacity is a priori assigned to each pier. However, if one or more piers are cracked, the definition of an effective residual drift capacity is not trivial. Accordingly, this research aims at the development of a simple and real-time methodology to estimate the residual drift capacity of damaged masonry walls.

In this paper, a new ML predictor employing numerically generated mechanically consistent crack patterns as training data is proposed. The desired features of this predictor are as follows:

(1) Real-time accurate predictions to support decision making in damaged masonry structures. This can be obtainable by ML approaches based on a large data set to train the predictions.
(2) Generalizable predictions. Predictions should not only be valid on a specific masonry type (or test set-up). As anticipated before, this might be obtainable by using a detailed block-based numerical model for data generation, that is, to generate mechanically consistent damaged wall conditions in an easier way with respect to experiments (Messali & Rots, 2018).
(3) Predictions based on a consistent data set. Indeed, residual displacement capacity and the representative feature of the crack pattern should be unambiguously computed given a damaged wall, independently of the masonry pattern, presence of openings, and so forth. Also, the input data (i.e., the quantification of cracks) should be unequivocally measurable on an actual wall. In addition, potential changes in stiffness and strength of the masonry wall due to the crack pattern (Dolatshahi & Beyer, 2022) are not discussed here as (i) the evaluation of the representative actual stiffness is extremely conventional (e.g., secant or tangent stiffness at a certain load ratio), and (ii) strength changes are typically significant in nearly collapse conditions, so outside the scope of this paper.
(4) Predictions feasible with different techniques. The input data should be obtainable with different technologies. For example, the representative feature of the crack pattern should be obtainable from still images, point clouds, DIC, and so forth. Also, the data generation should be possible through several models and approaches. Accordingly, the training data set might be subsequently enlarged by various research groups.

With this in mind, this paper contributes a new method for the real-time prediction of residual displacement capacity in damaged masonry walls based on an ensemble of convolutional neural networks (CNNs) using as only input the crack pattern. The damaging block-based numerical model developed and validated in D'Altri et al. (2019), where a parameter calibration on small scale experimental tests guaranteed a good agreement against large-scale cyclic tests of masonry structures, is

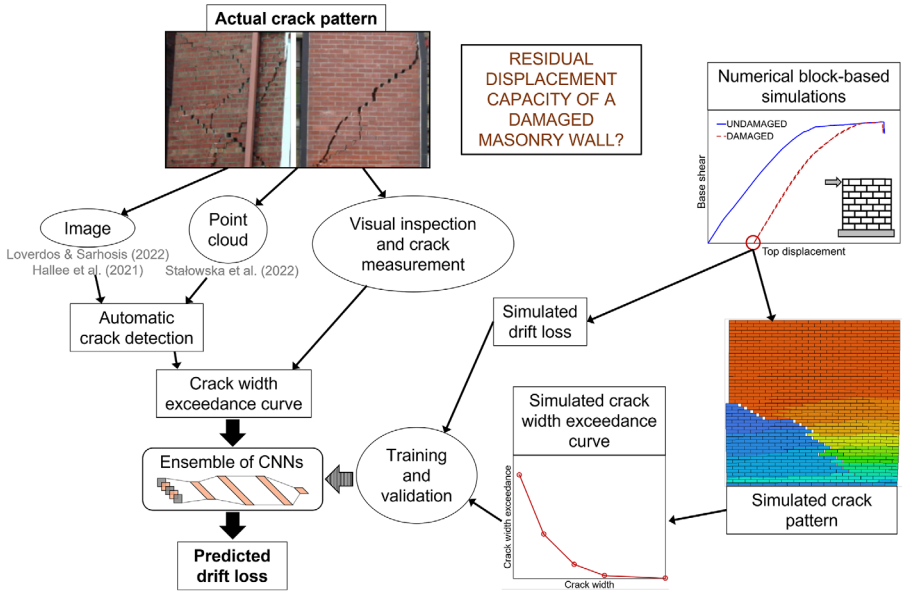


FIGURE 1 Methodology overview. Actual crack pattern pictures adapted from Giaretton et al. (2016).

used to generate simulated crack patterns induced by external actions (earthquake-like loads and differential settlements). Accordingly, numerous damaged conditions in masonry walls are simulated. For each damaged condition, the crack width cumulative distribution (CWCD) is extracted, and a crack width exceedance curve (CWEC) is subsequently derived. The CWEC is adopted as representative feature of the crack pattern, and the residual drift capacity is expressed in terms of drift loss (DL), evaluated with respect to the corresponding undamaged wall. Accordingly, DL is a direct measurement for damage prognosis, as it represents the current loss of drift capacity of a wall with respect to an undamaged condition. A specific data preprocessing is carried out to obtain convenient input data and to avoid dependencies on the block-to-wall size ratio. Many pairs of CWEC and DL are thus generated and used for training (and validating) an ensemble of CNNs, based on repeated k-folding cross validation with shuffling.

The paper is structured as follows. Section 2 presents an overview of the proposed methodology. Section 3 describes how data are generated and preprocessed. Section 4 presents and describes the ensemble of CNNs used for DL prediction. Section 5 shows and discusses the results of the crack pattern–based ML predictor. Finally, Section 6 summarizes the potentialities of the proposed methodology.

2 | METHODOLOGY

An overview of the methodology to predict the residual displacement capacity of a damaged masonry wall based only on the actual crack pattern is herein discussed (Figure 1).

First, the actual crack pattern is surveyed on a masonry wall. This could be conducted either through automatic crack detection procedures based on images (Hallee et al., 2021; Loverdos & Sarhosis, 2022) and/or point clouds (Stałowska et al., 2022), or through more traditional visual inspections and manual crack measurements. The crack pattern is then conveniently described through a discrete CWEC, derived from the CWCD. The CWEC is constituted of points, which represent the percentage of masonry joints exceeding a certain value of crack width. The CWEC, eventually normalized to avoid block-to-wall size ratio dependencies (see Section 3.3), is taken as the main representative feature of the crack pattern, as it summarizes crack patterns in a very concise and standalone way. Additionally, CWECs can be derived by means of any crack detection/measurement techniques, for any type of masonry.

The CWEC is given as input to an ensemble of CNNs, specifically trained and validated on numerous data numerically generated. The CNNs will give as output a prediction of the DL, based only on the CWEC, that is, based only on the actual crack pattern. Such real-time prediction of DL can be used to support decision making in damaged masonry structures. For example, it can be straightforwardly used by practitioners in seismic assessments when utilizing equivalent frame models (directly reducing the drift capacity of damaged piers through the DL).

The training and validation of the ensemble of CNNs is based on several pairs of CWEC and DL generated by means of numerical block-based simulations. Several damaged wall conditions with different dimensions and boundary conditions are generated and considered. On the one hand, the simulated CWEC is directly obtained

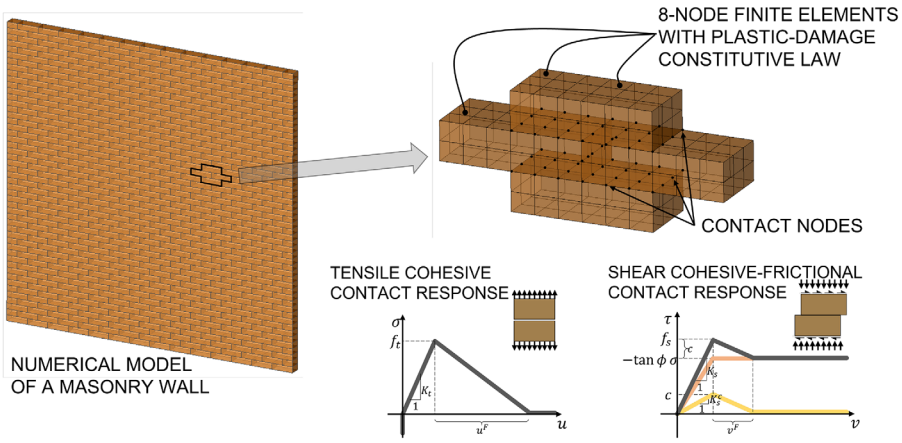


FIGURE 2 Block-based numerical modeling of masonry walls.

from the simulated crack pattern (extracted from damaged walls in equilibrium with no external horizontal forces applied). On the other hand, the simulated DL is obtained by reloading the damaged wall within a pushover analysis framework, comparing the residual drift capacity (after a normalization of the displacement baseline) with respect to the undamaged wall. Accordingly, the DL is straightforwardly related to the residual drift capacity of the damaged wall.

Any suitable masonry block–based numerical model can be utilized in this framework. The potential of using numerical simulations instead of actual experiments (Rezaie et al., 2022) to train the ML predictor, beyond the cheaper cost that can allow the generation of a much larger and diversified data set, consists of the possibility to directly assess the residual displacement capacity on each damaged wall condition, by performing a pushover analysis on the cracked wall. Hence, this allows a straightforward measurement of the DL, that does not depend on the cyclic test protocol.

To conclude, it should be highlighted that all numerical simulations are conducted off-line (a priori), as they consist in preanalyses to generate the data set. Then, the ensemble of CNNs is trained on the data set. Once the ensemble of CNNs is identified and given a CWEC, a real-time prediction of the residual drift capacity is achieved.

3 | DATA GENERATION

This section describes data generation based on numerical modeling. Data preprocessing is carried out to extract representative features of the crack pattern (CWEC), together with a representative feature of the seismic response of damaged masonry walls, that is, the residual drift capacity (definable in terms of DL). Such preprocessing leads to convenient input data for the ML predictor.

3.1 | Block-based numerical model

The damaging block-based modeling approach previously developed and validated at the full-scale in D'Altri et al. (2019) is employed to generate crack patterns. Such a model, which is briefly described in the following, considers explicitly all the blocks of the structure as nonlinear continuum bodies (discretized by eight-node solid finite elements) interacting through a contact-based node-to-surface frictional-cohesive formulation to represent the behavior of masonry joints (Figure 2). This modeling approach has been validated by D'Altri et al. (2019) against large-scale cyclic tests of masonry structures for both in- and out-of-plane responses.

The isotropic plastic-damage constitutive model developed in Lee and Fenves (1998) is assumed to govern the mechanical response of blocks. Two independent damage variables for tension ($0 \leq d_t < 1$) and compression ($0 \leq d_c < 1$) are supposed, and softening behavior could thus take place in tension and/or compression, and the uniaxial stress–strain response in tension and compression is, respectively:

$$\sigma_t = (1 - d_t)E_B\left(\varepsilon_t - \varepsilon_t^p\right), \quad \sigma_c = (1 - d_c)E_B\left(\varepsilon_c - \varepsilon_c^p\right) \tag{1}$$

where E_B is the block Young's modulus, σ_t and σ_c are the uniaxial tensile and compressive stresses, ε_t and ε_c are the uniaxial tensile and compressive strains, and ε_t^p and ε_c^p are the uniaxial tensile and compressive plastic strains. This constitutive model is then characterized by a nonassociative flow rule and a multiple-hardening Drucker–Prager–type yield surface in multiaxial stress conditions.

The block interaction is governed by a contact constrain enforced by the Lagrange multiplier method. A tensile cohesive contact response (Figure 2), governed by f_t (tensile strength), K_t (cohesive stiffness in normal direction), and u^F (excursion of normal displacement in

the postpeak branch), is supposed. Analogously, a shear cohesive-frictional contact response (Figure 2), governed by f_s (shear strength, equal to $f_s = c - \tan\phi\sigma$, c being the shear cohesion, ϕ the friction angle, and σ the contact stress), K_s (shear stiffness given by the contributions of cohesion K_s^c and friction), and v^F (excursion of slip in the softening branch), is supposed. Once reached tensile or shear strengths, the maximum value of the contact stresses in a contact point is given by:

$$\sigma = (1 - D)f_t, \qquad \tau = (1 - D)c - \tan\phi\sigma \tag{2}$$

where τ is the shear stress, and $0 \leq D \leq 1$ is the degradation scalar contact variable, which is equal to $D = 0$ at the peak of cohesion, and $D = 1$ at the full degradation of the contact point (Pereira et al., 2023).

3.2 | Numerical campaign

Solid fired clay brick masonry is assumed as material for the data generation, as it is the most representative type of masonry for existing structures (Ghiassi et al., 2019). Single-leaf running bond masonry is considered. The calibration of the model parameter, which has been conducted following the calibration strategy proposed in D'Altri et al. (2019), is shown and discussed in the Appendix, based on the experimental tests performed in Jafari and Esposito (2017) and Licciardello and Esposito (2019).

A total of 100 damaged wall conditions are generated by considering different combinations of wall sizes, boundary conditions, axial load ratios (ALRs, defined as the ratio between the actual vertical stress and the compressive strength of the material), and external actions, chosen to be representative of actual masonry walls:

(1) wall size: 2×3, 3×3, 4.2×3 ($B \times H$, in m);
(2) boundary condition: cantilever, fixed-guided (considering a rigid beam on the top surface of the wall and allowing vertical displacements in both cases);
(3) ALR: 10%, 15%, 20%;
(4) external action: monotonic and cyclic horizontal force (applied on the top surface of the wall), differential settlement (applied vertically on a portion of the bottom surface of the wall).

For each combination of these features, several damaged wall conditions are extracted.

The ultimate condition of a wall is identified when the pushover curve shows a significant base shear drop (more than 20% of the peak shear, or a sharp drop with more than 10% in less than 0.5 mm). The range of variation of the ultimate drift computed for initially undamaged walls is shown in Table 1 for different wall sizes.

TABLE 1 Range of variation of the ultimate drift computed in the simulations for initially undamaged walls.

Wall size ($B \times H$ in m)	Ultimate drift
2×3	0.49% – 0.95%
3×3	0.43% – 0.54%
4.2×3	0.43% – 0.60%

Examples of crack patterns generated on a 3×3 wall with 10% ALR are shown in Figure 3 (crack patterns are extracted while no external horizontal forces are applied). For the sake of comparison, 2×3 (Figure 4a) and 4.2×3 (Figure 4b) walls with 10% ALR (monotonic, fixed guided) are also shown.

Examples of horizontal force-top displacement curves for 3×3 walls with 10% ALR (cantilever) are shown in Figure 5. It is here recalled that each damaged wall configuration is reloaded within a pushover analysis framework to compute the residual displacement capacity (Figure 5).

A total of 100 damaged wall cases were generated, with 36 of 3×3 walls, 31 of 2×3 walls, and 33 of 5×3 walls, with ALRs ranging from 10% to 20% for all cases. There are 65 pushover, 24 cyclic, and 12 settlement load cases, with at least one case of each load for each wall size versus boundary condition. For the boundary conditions, 56 cases are cantilevered and the 44 remaining are fixed. Beyond the 100 abovementioned damaged walls, extra crack patterns are generated (see Figure 6) and used for a posteriori verification of the method, choosing cases never seen in the training/validation data set, in particular:

(1) 3×3 wall with small opening (monotonic, fixed-guided, 8% and 12% ALR);
(2) 3×3 wall with Flemish bond (monotonic and cyclic, cantilever, 12% ALR);
(3) 2×2 wall with giant granite blocks, inspired by the case study in Milani et al. (2013) and Pereira et al. (2023) (monotonic, fixed-guided, 18% ALR). In this case, the mechanical setting utilized in Pereira et al. (2023) for granite masonry has been adopted.

3.3 | Data preprocessing

CWCDs are extracted from the results of numerical simulations (Figure 7a), for damaged walls in equilibrium with no external horizontal forces applied. To simplify and consolidate the amount of data, CWCDs are conveniently turned into discrete CWECs characterized by a representative number of discrete crack width thresholds (CWTs). In the following, five CWTs are assumed (0.1 mm, 0.5 mm,

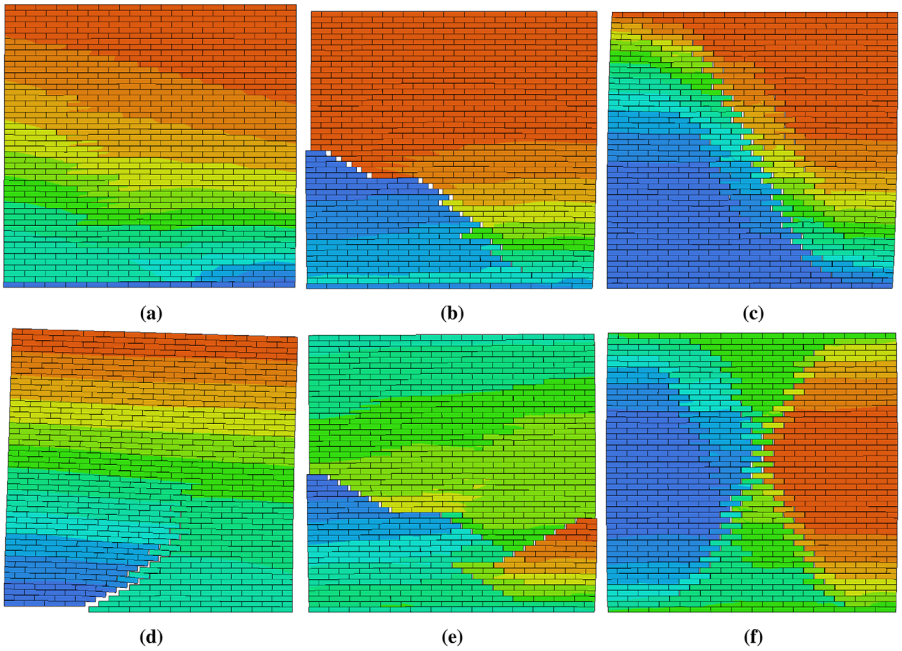


FIGURE 3 Examples of crack patterns generated on a 3×3 wall with 10% axial load ratio (ALR). (a) Monotonic, cantilever (early stage). (b) Monotonic, cantilever (later stage). (c) Monotonic, fixed-guided. (d) Settlement, cantilever. (e) Cyclic, cantilever. (f) Cyclic, fixed-guided. Magnified horizontal displacement contour plots are used to better highlight the crack pattern.

1.0 mm, 1.5 mm, 2.5 mm), see Figure 7b. It should be underlined that the minimum CWT has been adopted equal to 0.1 mm as narrower cracks are typically invisible to the naked eye (Korswagen et al., 2019).

Each damaged wall configuration is then reloaded and employed in a pushover analysis framework. The pushover curve is extracted and the baseline of the top horizontal displacement is normalized so that to begin at null displacement (Figure 8). Thereby, the residual drift capacity of the damaged wall (δ^D) is computed in the normalized baseline. Hence, for each damaged wall configuration, the DL with respect to a reference undamaged wall is computed as:

$$\mathrm{DL} = \frac{\delta^R - \delta^D}{\delta^R} \tag{3}$$

where δ^R is the drift capacity of the reference undamaged wall.

Accordingly, each damaged wall configuration is characterized by a CWEC composed of five parameters (that will be used as input values) and the DL that will be used as output.

3.3.1 | Data normalization

In order to keep consistent the input data independently from the block-to-wall size ratio, a simple normalization of the CWEC is considered here. Accordingly, the total crack length of cracks exceeding a certain CWT is normalized by a representative length of the wall. In particular, each discrete value of the CWEC is multiplied by the factor

$$\Lambda = \frac{t_j l_j}{l_w} \tag{4}$$

t_j being the total number of joints of the wall, l_j a representative length of the joint (assumed here as averaged half-length of masonry bed joints), and l_w a representative length of the wall (e.g., length of the wall diagonal). Accordingly, the i-th value of the normalized CWEC (n-CWEC) represents a normalized total length of cracks exceeding the i-th CWT.

4 | ENSEMBLE PREDICTIVE MODEL

In this section, the CNN model architecture, the training policy, and the prediction method using an ensemble of models $m(\cdot)$ generated using repeated k-folding cross validation with shuffling are presented. Usually, k-fold cross validation is employed to better assess a model performance to unseen data (James et al., 2023). This process is here employed to generate a set of data-driven predictive models that are used in conjunction at inference time. Further, it is found that allowing partial information leakage from validation data via warm start of subsequent folds

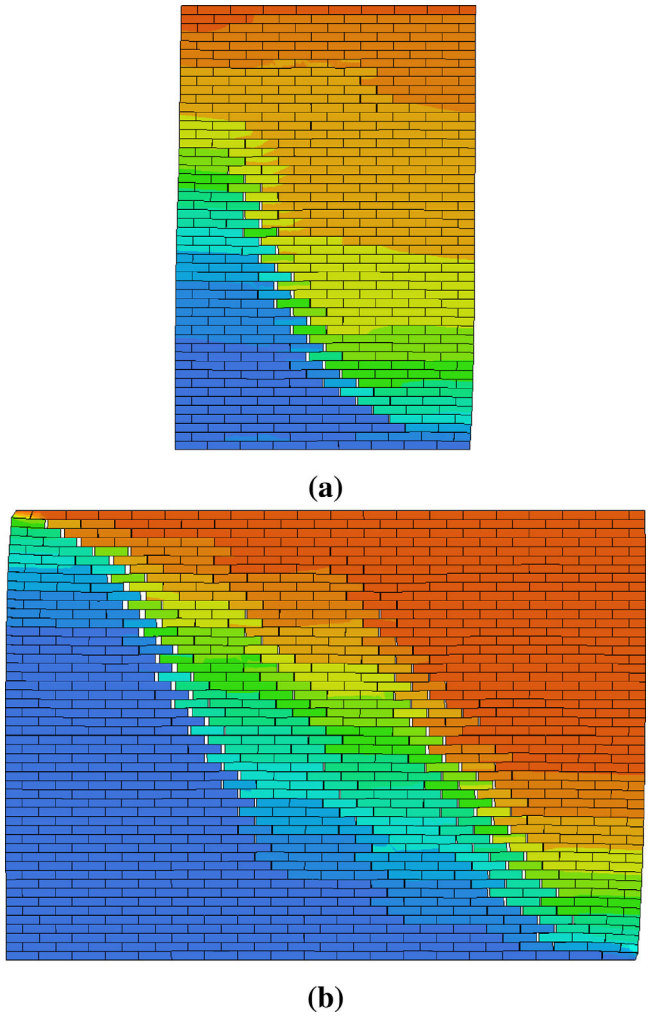


FIGURE 4 Examples of crack patterns generated on (a) 2×3 and (b) 4.2×3 walls with 10% axial load ratio (ALR; monotonic, fixed guided). Magnified horizontal displacement contour plots are used to better highlight the crack pattern.

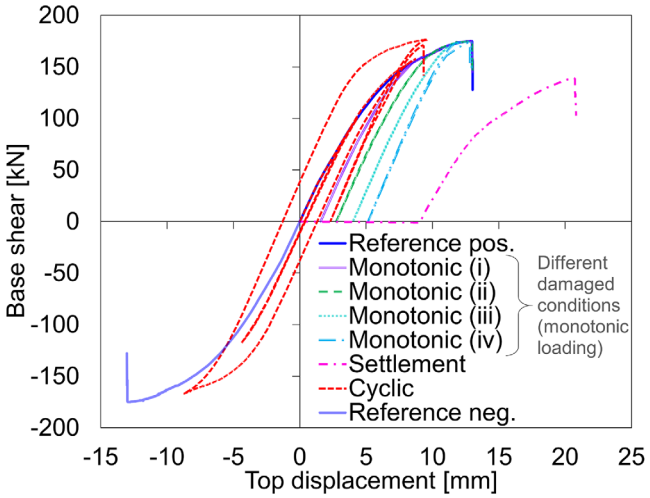


FIGURE 5 Examples of horizontal force-top displacement curves (3×3 walls with 10% axial load ratio [ALR], cantilever).

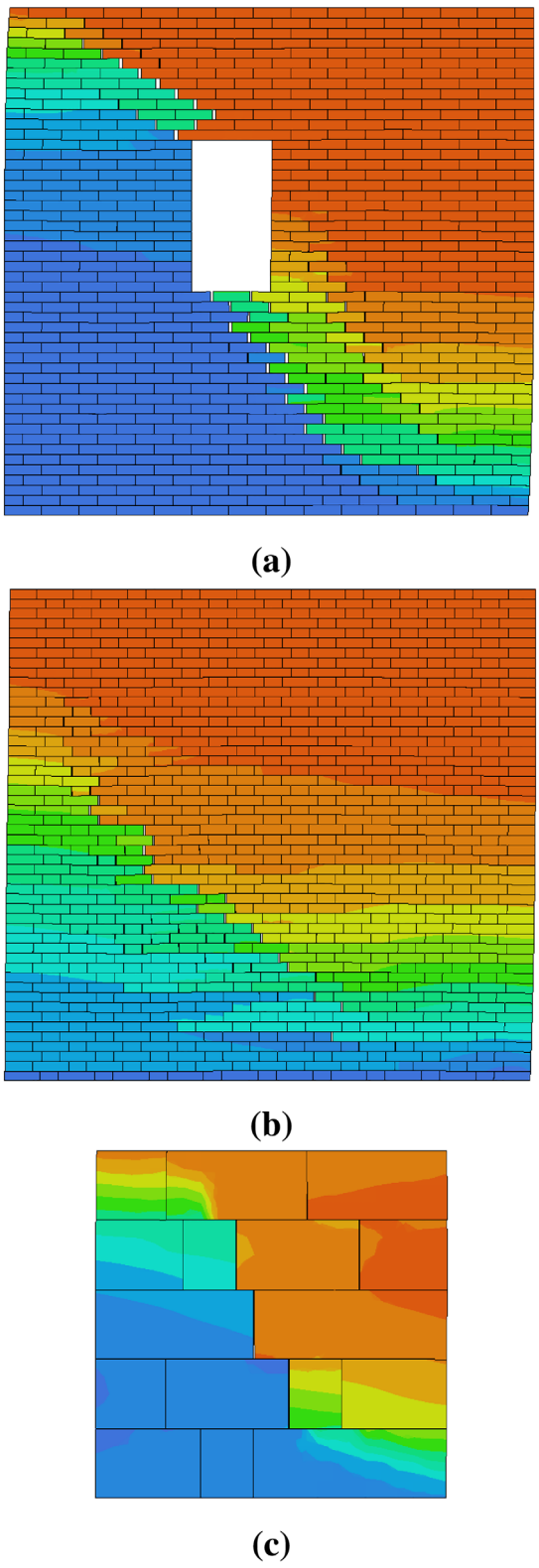
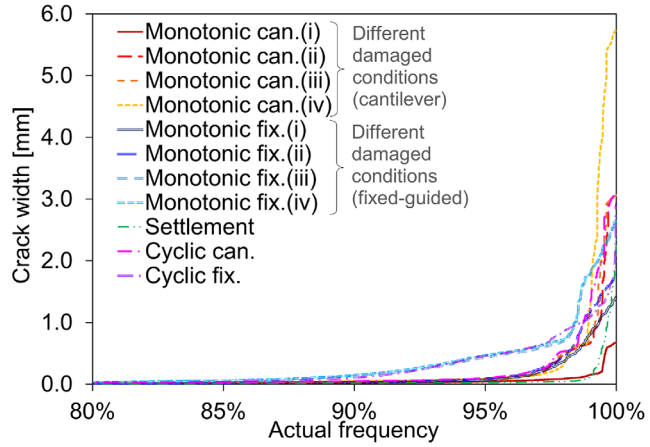
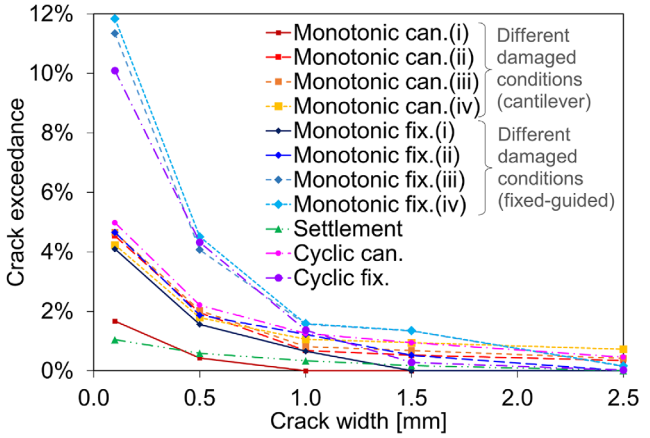


FIGURE 6 Examples of generated crack patterns used for a posteriori verification of the method: (a) 3×3 wall with small opening, (b) 3×3 wall with Flemish bond, and (c) 2×2 wall with giant granite blocks. Magnified horizontal displacement contour plots are used to better highlight the crack pattern.



(a)



(b)

FIGURE 7 Examples of (a) crack width cumulative distributions (CWCDs) and related (b) discrete crack width exceedance curves (CWECs) for crack patterns generated on a 3×3 wall with 10% axial load ratio (ALR).

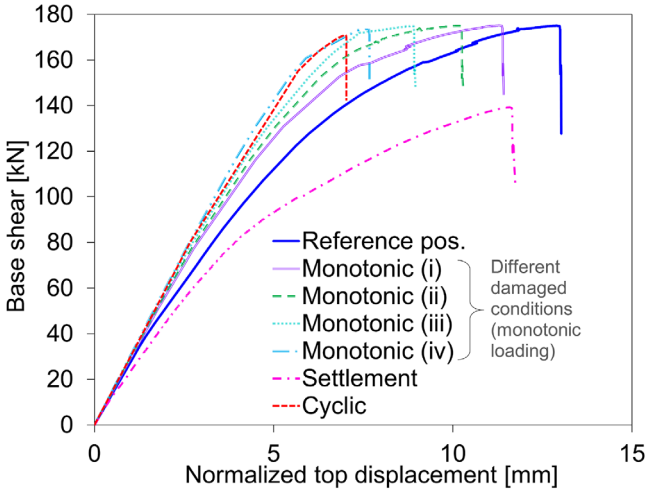


FIGURE 8 Examples of horizontal force-normalized top displacement curves for (3×3 walls with 10% axial load ratio [ALR], cantilever).

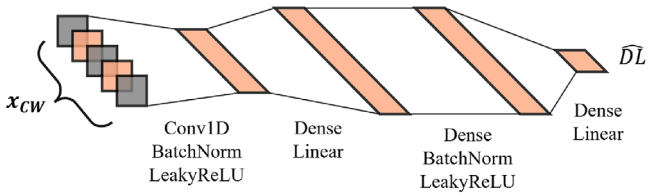


FIGURE 9 Convolutional neural network architecture.

and repetitions leads to superior performance in unseen test data.

This section is organized as follows, first the CNN architecture is introduced, followed by the training policy employed. Then, the method for the generation of the ensemble is presented, together with a results discussion to show the performance of the proposed approach.

4.1 | Neural network architecture

In the proposed method, DL prediction is performed using an ensemble of data-driven models $m(\cdot)$. Each model $m(\cdot)$ takes as input a vector $\mathbf{x}_{cw} \in \mathbb{R}^5$ representing the CWEC and outputs the expected DL. The model $m(\cdot)$ consists of a CNN, shown in Figure 9, that applies a $1D$ convolutional layer (Conv1D) to the input followed by batch normalization (BatchNorm) and with a leaky rectilinear unit (LeakyReLU) activation. The convolutional layer has a single kernel of length 3, linear activation, and no zero padding. The LeakyReLU is an activation function given by

$$\text{LeakyReLU}(x) = \begin{cases} x, & \text{if } x > 0 \\ 0.3x, & \text{otherwise} \end{cases} \tag{5}$$

This is followed by a dense connection to a layer with four neurons, followed by BatchNorm and LeakyReLU, another dense layer of equal size, and a dense connection to a single output yielding DL. The use of the convolutional architecture was motivated by the distinct gradients and curvatures observed in the CWECs (e.g., see Figure 7b). Instead of using traditional metrics of gradient and curvature, the kernels are learned directly from the data. The following two dense layers are used to increase model capacity, and the number of neurons was increased until generalization to the validation data started to drop.

4.2 | Training policy

Each model $m(\cdot)$ is trained using stochastic gradient descent (SGD) with Adam optimizer (Kingma & Ba, 2015) to minimize a mean squared error (MSE) loss with respect

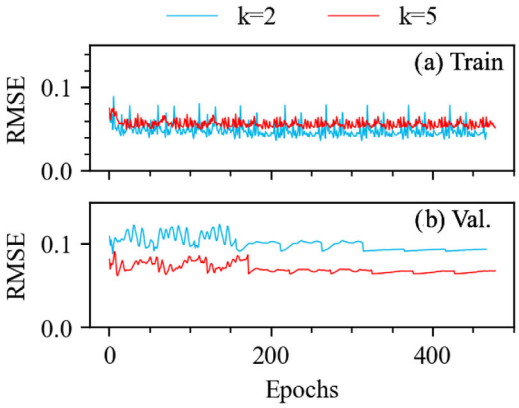


FIGURE 10 Samples of (a) training and (b) validation loss curves.

to the DL. Other loss functions can be used, such as the mean absolute error (MAE), but the MSE penalizes large errors more, and for this reason, it is a suitable loss function for the proposed application. The training policy proceeds as follows, an initial learning rate of 0.1 is set. Training proceeds until the validation loss does not improve for 50 consecutive epochs or a maximum of 500 epochs is achieved. That process is repeated for another three iterations. Because the SGD uses random data batches, repeating the process at the same learning rate raises the opportunity of finding better local optima at that learning rate scale. Then, the learning rate is reduced to 0.01 and the procedure is repeated starting at the previous lowest validation loss. Finally, the learning rate is reduced to 0.001 and the procedure is repeated.

4.3 | K-fold cross-validation ensemble

While over a 100 numerical simulations were performed, from a data-driven modeling perspective that is a relatively limited amount of data (e.g., modern computer vision data sets often exceed millions of samples; Russakovsky et al., 2015). Further, due to the presence of multiple combinations of BCs, ALRs, and loading types, the assessment of model generalization to unseen data using a single training and validation random data split is insufficient. This is because a single random train–validation data split can lead to better or worse generalization by chance, which will bias the model performance on unseen test data. For example, Figure 10 shows two training and validation loss curves resulting from two alternative data splits (in this case, there are a total of five alternative splits of the data, the plot shows only cases $k = 2$ and $k = 5$, where k indicates the data split case), which clearly exhibit different validation performances (units are omitted in the plot as the RMSE corresponds to the DL, which is a dimensionless quantity).

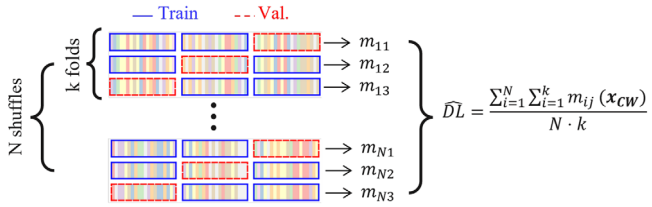


FIGURE 11 Diagram showing the ensemble of N random shuffles with k-folding for prediction of drift loss.

In this paper, the use of an ensemble of models generated by performing N repeats of k-folding cross validation with shuffling is proposed to improve model generalization to unseen data.

The k-folding cross validation is a traditional method to improve the assessment of generalization performance. In such a method, the data are split into k segments, with one segment held out for validation and the remaining $k - 1$ segments used for training. Then, k models are trained, with each model using a distinct segment for validation. A performance metric (e.g., MSE or R^2) is used for each of the k models, and statistics of the metric, such as the mean, are used to better reflect the model performance on unseen test data. Since in this work the proposed CNN model is small, it is feasible to store all k models and perform inference on new data by averaging the output of all k models. Furthermore, the k-folding process can be repeated N times while shuffling the data, such that the data points within each k segment are varied. In practice, this process creates $N \cdot k$ distinct models $m_{ij}, i, j = 1 \ldots N, 1 \ldots k$, where i corresponds to the number of data shuffles, and j corresponds to the j-th of the k data splits for a given data shuffle, as illustrated in Figure 11.

Then, given an unseen test data $\mathbf{x}_{cw}$, the predicted DL, $\widehat{DL}$, from the ensemble is

$$\widehat{DL} = \frac{\sum_{i=1}^{N}\sum_{j=1}^{k} m_{ij}(\mathbf{x}_{cw})}{N \cdot k} \tag{6}$$

The advantage of the ensemble approach is to minimize the occurrence of extreme predictive errors.

Usually, k-folding is performed by training each model from scratch, to prevent information leakage and overestimation of performance. However, it is possible to train subsequent models starting from a previously found set of weights. This can affect the estimation of model generalization to unseen data, as the preset weights contain information about the training data, which may be used as validation data in a subsequent fold. However, in the present work, it was found that allowing for a small degree of information leakage via trained weights can benefit the generalization to unseen data. This is akin to a warm start procedure, where a model is trained starting from an

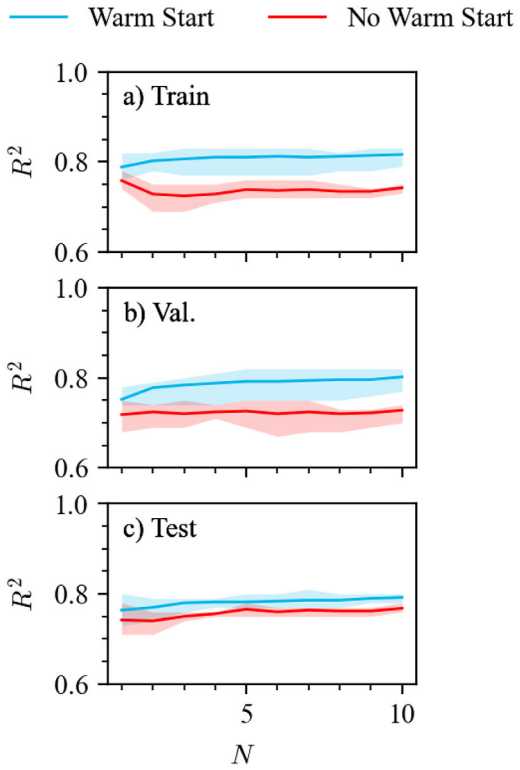


FIGURE 12 Training, validation, and testing losses for $k = 5$ segments with and without warm start.

already good set of weights (Ash & Adams, 2020). Using this approach leads to better training, validation, and testing R^2 as shown in Figure 12. Notice also that repeating the k-folding with data shuffling improves performance. For each approach (with and without warm start), the full training procedure is repeated five times. The shaded region corresponds to the min/max observed, and the solid line corresponds to the mean R^2. The full ensemble construction procedure is presented in Algorithm 1.

ALGORITHM 1 Ensemble Generation.

Require: Training and validation data
Ensure: Trained ensemble
  Initialize model m_{11}
  for $i = 1$ to N **do**
    Shuffle data
    Split shuffled data into k non-overlapping segments
    if ∃ model weights $w_{i-1,k}$ **then**
      Assign $w_{i-1,k}$ to m_{ij}
    end if
    for $j = 1$ to k **do**
      Get k^{th} training and validation data
      if ∃ model weights $w_{i,j-1}$ **then**
        Assign $w_{i,j-1}$ to m_{ij}
      end if
      Train model m_{ij} using Algorithm 1
      Get trained model weights w_{ij}
    end for
  end for

5 | RESULTS

In this section, predicted versus true DL (here, true corresponds to the DL obtained from the numerical simulations) and R^2 results are presented for training, validation, and testing using the proposed method. To better understand the method potentialities and limitations, and the impact of training and test data selection, four data scenarios (DS) are considered:

1. **Interpolation**: The data concerning 0.15 ALR are excluded from training and validation and used exclusively for testing (about 52% of the data). Supporting results presented in the previous section consider this DS. The goal is to assess model performance on unseen data whose ALR falls in-between the limits of ALRs explored during training. This scenario is better suited for the application of ML methods as opposed to extrapolation with respect to ALR.
2. **Balanced test data**: Two random samples from each case of wall size and ALR are taken for testing (about 18% of the data), and the remaining data are used for training and validation. The goal is to assess the performance impact of testing on data of similar wall sizes and ALRs but not necessarily subject to the same boundary conditions and loading.
3. **Balanced training data**: Seven random samples from each case of wall size and ALR are taken for training and validation, with the remaining (about 37% of the data) left for testing. The goal is to assess the test performance when a balanced number of ALRs and wall sizes are considered during training.
4. **Randomized**: Random data splits are used for training, validation, and testing, with 20% of data left for testing. The goal is to assess the range and average performance of the proposed method when agnostic to the block sizes, ALRs, and boundary and loading conditions. This can be a challenging scenario as data imbalances and extrapolation conditions can appear by chance.

In Figure 13, scatter plots of predicted $\widehat{DL}$ versus true DL are presented. Bounds of 10% and 20% errors are added to aid in the visual interpretation of the scatter plots. Notice that since the training process involves a random step at the formation of the k segments, some variation

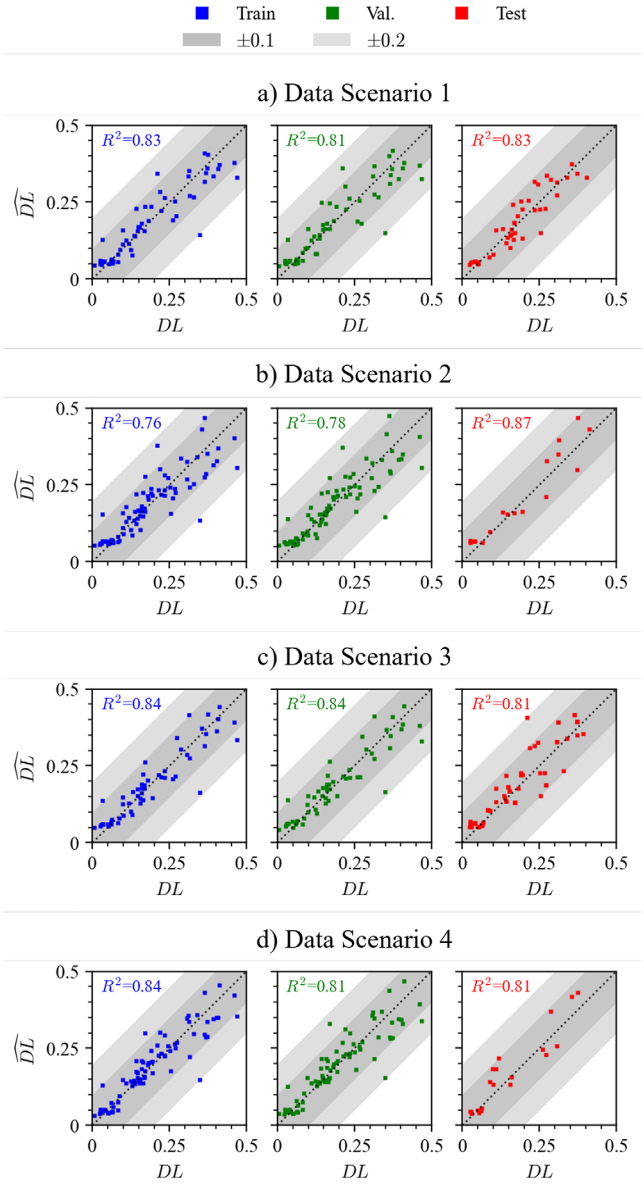


FIGURE 13 Scatter plots of predicted versus truth for training, validation, and test data with $k = 5$ segments and $N = 10$ repeats for multiple data scenarios.

can be observed on iterations of the methodology. That aspect is discussed later in this section, scatter plots of one sample iteration of the algorithm for the multiple DS aforementioned are presented.

For DS1 (Figure 13a), good DL prediction is observed on the test data, and the performance observed in the training and validation data is coherent with the test data. Although not shown here for conciseness, some performance overestimation is observed when the validation data are too small (e.g., less than 20% of the available data for training and validation). This is due to the higher uncertainty associated with performance estimation on small amount of validation data and the associated increased amount of training data, which improves coverage of loading scenarios.

For DS2 (Figure 13b), excellent DL prediction is observed on the test data. The performance observed in the training and validation data underestimates the performance on the test data, but still with good prediction performance on validation data. The reason for the excellent performance here is that the wall sizes and ALRs present in the test data have counterparts in the training data, albeit with potentially different loading and boundary conditions. This seems to facilitate prediction. The presence of more diverse data can challenge the model capacity leading to some small degradation of training and validation performance.

For DS3 (Figure 13c), good DL prediction is observed on the test data. The performance is similar to DS1, which indicates that it is possible to achieve similar performance with less data if careful case coverage is taken into account during the numerical experiment design. Further, it also suggests that potential data imbalances present in DS1 do not deteriorate the model performance.

For DS4 (Figure 13c), this iteration shows good DL prediction also with performance similar to DS1, however as will be discussed next, good performance is not guaranteed when randomly sampling the training data.

For DS 2,3,4, the proposed method is repeated five times. The goal was to assess the method repeatability. Figure 14 shows the performance in terms of R^2 (computed with respect to the true and predicted DL) for DS 2,3,4 (recall that DS1 was already considered in Figure 12) as a function of repeats of the k-folding process. For DS2 (Figure 14a), the method shows high repeatability, with consistent test performance across multiple iterations. For DS3 (Figure 14b), the method shows higher variance, but with fair to good performance across iterations, and average performance close to $R^2 \approx .75$. For DS4 (Figure 14c), poor repeatability is observed, with one iteration showing subpar test performance. In this DS, occasional critical data imbalances are observed. These imbalances can cause overfitting of the model and poor generalization. For example, notice that the corresponding training and validation R^2 for the anomalous sample (solid red in Figure 14c) are significantly higher than the remaining iterations, which enables identification of models with poor generalization. When this anomalous iteration is removed, the mean performance (purple dash-dot in Figure 14c) on DS4 is similar to DS1 and DS2.

These results indicate that the proposed method is robust under multiple DS, and that if overfitting occurs, it is possible to identify it.

To check the method generalization, DL prediction is also performed for walls not used in the training data, which considers the presence of openings, different

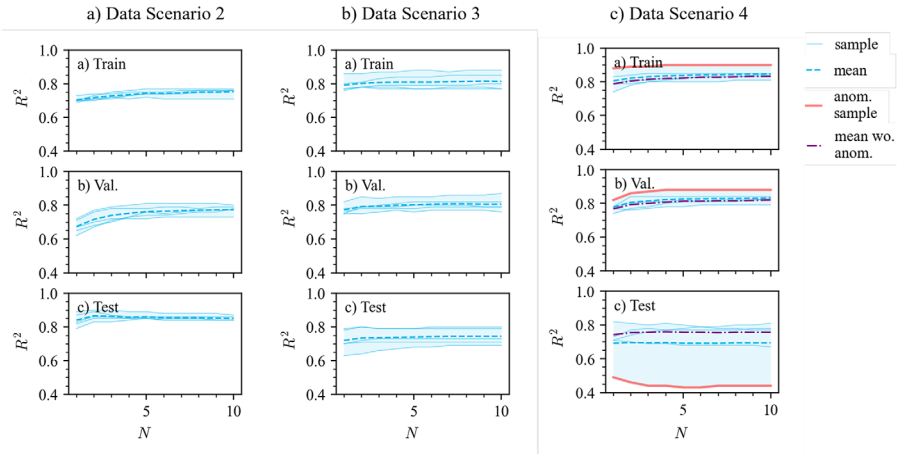


FIGURE 14 Training, validation, and testing performance under multiple data scenarios.

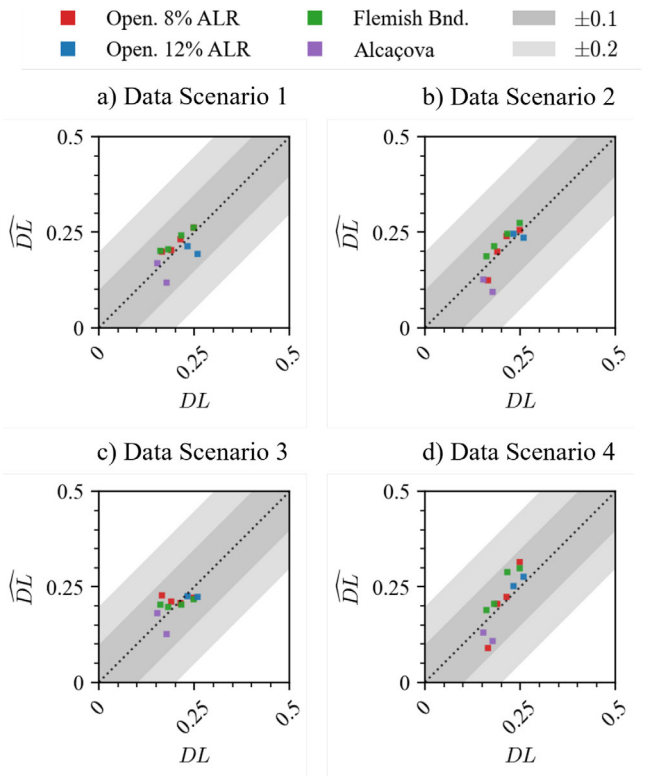


FIGURE 15 Scatter plots of predicted vs. truth for walls configurations never seen in the training and validation data set with $k = 5$ segments and $N = 10$ repeats for multiple data scenarios.

masonry textures, ALRs, and block sizes, that is, cases never seen in the training and validation data. As shown in Figure 15, predictions are accurate, all within 10% of error for multiple DS. Accordingly, given these preliminary results, the method appears generalizable to other masonry walls, with different geometries (e.g., presence of small openings), masonry patterns, ALRs, and block-to-wall ratios.

It has to be pointed out that the pure shear sliding failure mode of masonry piers should be carefully treated, as this kind of failure will plausibly never show a substantial decay of the shear force given that friction will guarantee a pseudo-constant response in shear and crushing will not occur. Although a consistent definition of the ultimate drift capacity would be not trivial for shear sliding, such failure is very rarely encountered in postearthquake inspections, as it characterizes walls with extremely small ALRs.

The ML predictor herein developed, once its parameters are trained, allows real-time predictions, that is, it manages to predict instantaneously the DL given a certain CWEC. Accordingly, it can be applied to support decision making in damaged masonry structures. For example, the DL prediction can be straightforwardly used in equivalent frame model–based seismic assessments, where the drift capacity of piers is defined a priori, by simply updating the residual drift capacity of cracked piers in numerical simulations.

6 | CONCLUSIONS

In this paper, a new approach employing an ensemble of CNNs for the prediction of residual drift capacity in damaged masonry walls has been developed using as only input the crack pattern. An accurate block-based numerical model has been employed to generate mechanically consistent crack patterns induced by earthquake-like actions and differential settlements. Although there are still challenges to be addressed in numerical modeling of masonry structures (e.g., imperfect boundary conditions to represent actual walls), the idea of numerically generating damaged walls scenarios appears promising. Any damaged wall has been described by a concise representative feature, that is, a discrete CWEC, which has been paired to the DL

with respect to the undamaged wall (evaluated within a pushover analysis framework). Numerous pairs of CWEC and DL have been generated and used for training (and validating) the ensemble of CNNs based on repeated k-folding cross validation with shuffling.

The CNN-based predictions showed good generalization to unseen data and the framework can be easily extended by increasing the available data via numerical simulation. The concise representative feature adopted for the crack pattern (i.e., the CWEC) encapsulates sufficient information for reasonably accurate predictions over distinct and differently damaged masonry walls, for example, for both seismic- and settlement-induced damage. Good predictions on masonry walls with features different from those used in the training data support the generalization potential of the proposed methodology. Accordingly, the training data set could be straightforwardly enlarged also by using other block-based numerical approaches (e.g., utilized in other research groups).

The utilization of this ML predictor in conjunction with modern automatic crack detection procedures for masonry appears straightforward, as the CWEC is easily and automatically computable on images and point clouds.

As a result, such methodology appears general, inexpensive, and able to adequately predict the DL using as only input the CWEC, providing real-time support for decision making in damaged masonry structures. The proposed method performs damage prognosis and therefore contributes to SHM Level IV.

Future developments might concern the enlargement of the training data set by adding further combinations of wall geometries, boundary conditions, and external actions (e.g., by also considering material aging). Additionally, the same data set of damaged masonry walls could be utilized to train predictors for the evaluation of stress increase in blocks due to the presence of cracks. This future development could originally support the static monitoring of masonry and historical structures.

ACKNOWLEDGMENTS

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Sklodowska-Curie grant agreement number 101029792 (HOLAHERIS project, "A holistic structural analysis method for cultural heritage structures conservation").

REFERENCES

Abdulla, K. F., Cunningham, L. S., & Gillie, M. (2017). Simulating masonry wall behaviour using a simplified micro-model approach. *Engineering Structures*, *151*, 349–365.

Addessi, D., Marfia, S., & Sacco, E. (2002). A plastic nonlocal damage model. *Computer Methods in Applied Mechanics and Engineering*, *191*(13-14), 1291–1310.

Angiolilli, M., Pathirage, M., Gregori, A., & Cusatis, G. (2021). Lattice discrete particle model for the simulation of irregular stone masonry. *Journal of Structural Engineering*, *147*(9), 04021123.

Ash, J. T., & Adams, R. P. (2020). On warm-starting neural network training. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, & H. Lin (Eds.), Proceedings of the 34th international conference on neural information processing systems, NIPS'20 (pp. 3884–3894). Curran Associates Inc.

Asjodi, A. H., & Dolatshahi, K. M. (2022). Peak drift ratio estimation for unreinforced masonry walls using visual features of damage. *Bulletin of Earthquake Engineering*, *20*(15), 8357–8379.

Boscato, G., Russo, S., Ceravolo, R., & Fragonara, L. Z. (2015). Global sensitivity-based model updating for heritage structures. *Computer-Aided Civil and Infrastructure Engineering*, *30*(8), 620–635.

Dais, D., Bal, I. E., Smyrou, E., & Sarhosis, V. (2021). Automatic crack classification and segmentation on masonry surfaces using convolutional neural networks and transfer learning. *Automation in Construction*, *125*, 103606.

D'Altri, A. M., de Miranda, S., Castellazzi, G., & Glisic, B. (2023). Numerical modelling-based damage diagnostics in cultural heritage structures. *Journal of Cultural Heritage*, *61*, 1–12.

D'Altri, A. M., Messali, F., Rots, J., Castellazzi, G., & de Miranda, S. (2019). A damaging block-based model for the analysis of the cyclic behaviour of full-scale masonry structures. *Engineering Fracture Mechanics*, *209*, 423–448.

D'Altri, A. M., Sarhosis, V., Milani, G., Rots, J., Cattari, S., Lagomarsino, S., Sacco, E., Tralli, A., Castellazzi, G., & de Miranda, S. (2020). Modeling strategies for the computational analysis of unreinforced masonry structures: Review and classification. *Archives of Computational Methods in Engineering*, *27*, 1153–1185.

Dang, L. M., Wang, H., Li, Y., Nguyen, L. Q., Nguyen, T. N., Song, H.-K., & Moon, H. (2022). Deep learning-based masonry crack segmentation and real-life crack length measurement. *Construction and Building Materials*, *359*, 129438.

De Bellis, M. L., & Addessi, D. (2011). A cosserat based multiscale model for masonry structures. International Journal for Multiscale Computational Engineering, *9*(5), 543–563.

Dolatshahi, K. M., & Beyer, K. (2022). Stiffness and strength estimation of damaged unreinforced masonry walls using crack pattern. *Journal of Earthquake Engineering*, *26*(2), 837–856.

Eltouny, K. A., & Liang, X. (2021). Bayesian-optimized unsupervised learning approach for structural damage detection. *Computer-Aided Civil and Infrastructure Engineering*, *36*(10), 1249–1269.

FEMA. (1998). FEMA 306, evaluation of earthquake damaged concrete and masonry buildings. Technical report, Federal Emergency Management Agency.

Ferrante, A., Loverdos, D., Clementi, F., Milani, G., Formisano, A., Lenci, S., & Sarhosis, V. (2021). Discontinuous approaches for nonlinear dynamic analyses of an ancient masonry tower. *Engineering Structures*, *230*, 111626.

Ghiassi, B., Vermelfoort, A., & Lourenco, P. B. (2019). Masonry mechanical properties. In *Numerical modeling of masonry and historical structures* (pp. 239–261). Elsevier.

Giaretton, M., Dizhur, D., da Porto, F., & Ingham, J. M. (2016). Construction details and observed earthquake performance of unreinforced clay brick masonry cavity-walls. *Structures*, *6*, 159–169.

Hallee, M. J., Napolitano, R. K., Reinhart, W. F., & Glisic, B. (2021). Crack detection in images of masonry using CNNs. *Sensors*, *21*(14), 4929.

Ierimonti, L., Cavalagli, N., Venanzi, I., García-Macías, E., & Ubertini, F. (2023). A Bayesian-based inspection-monitoring data fusion approach for historical buildings and its post-earthquake application to a monumental masonry palace. *Bulletin of Earthquake Engineering*, *21*(2), 1139–1172.

Jafari, S., & Esposito, R. (2017). Material tests for the characterisation of replicated solid clay brick masonry. Delft University of Technology. Report, C31B67WP1-12.

James, G., Witten, D., Hastie, T., Tibshirani, R., & Taylor, J. (2023). *An introduction to statistical learning with applications in Python*. Springer.

Katsigiannis, S., Seyedzadeh, S., Agapiou, A., & Ramzan, N. (2023). Deep learning for crack detection on masonry façades using limited data and transfer learning. Journal of Building Engineering, 107105.

Kingma, D., & Ba, J. (2015). Adam: A method for stochastic optimization. In J. Bengio, & Y. LeCun (Eds.), 3rd. international conference for learning representations.

Korswagen, P. A., Longo, M., Meulman, E., & Rots, J. G. (2019). Crack initiation and propagation in unreinforced masonry specimens subjected to repeated in-plane loading during light damage. *Bulletin of Earthquake Engineering*, *17*, 4651–4687.

Lee, J., & Fenves, G. L. (1998). A plastic-damage concrete model for earthquake analysis of dams. *Earthquake Engineering & Structural Dynamics*, *27*(9), 937–956.

Licciardello, L., & Esposito, R. (2019). Material tests for the characterisation of replicated solid clay brick masonry. Delft University of Technology. Report, C31B04-WPC.

Lourenço, P. B., & Rots, J. G. (1997). Multisurface interface model for analysis of masonry structures. *Journal of Engineering Mechanics*, *123*(7), 660–668.

Loverdos, D., & Sarhosis, V. (2022). Automatic image-based brick segmentation and crack detection of masonry walls using machine learning. *Automation in Construction*, *140*, 104389.

Macorini, L., & Izzuddin, B. A. (2011). A non-linear interface element for 3d mesoscale analysis of brick-masonry structures. *International Journal for numerical methods in Engineering*, *85*(12), 1584–1608.

Malek, K., Mohammadkhorasani, A., & Moreu, F. (2023). Methodology to integrate augmented reality and pattern recognition for crack detection. *Computer-Aided Civil and Infrastructure Engineering*, *38*(8), 1000–1019.

Meng, S., Gao, Z., Zhou, Y., He, B., & Djerrad, A. (2023). Real-time automatic crack detection method based on drone. *Computer-Aided Civil and Infrastructure Engineering*, *38*(7), 849–872.

Messali, F., & Rots, J. (2018). In-plane drift capacity at near collapse of rocking unreinforced calcium silicate and clay masonry piers. *Engineering Structures*, *164*, 183–194.

Milani, G., Esquivel, Y. W., Lourenço, P. B., Riveiro, B., & Oliveira, D. V. (2013). Characterization of the response of quasi-periodic masonry: Geometrical investigation, homogenization and application to the Guimar Aes Castle, Portugal. *Engineering Structures*, *56*, 621–641.

Napolitano, R., & Glisic, B. (2019). Methodology for diagnosing crack patterns in masonry structures using photogrammetry and distinct element modeling. *Engineering Structures*, *181*, 519–528.

Napolitano, R., & Glisic, B. (2020). Hybrid physics-based modeling and data-driven method for diagnostics of masonry structures. *Computer-Aided Civil and Infrastructure Engineering*, *35*(5), 483–494.

Ni, F., Zhang, J., & Chen, Z. (2019). Zernike-moment measurement of thin-crack width in images enabled by dual-scale deep learning. *Computer-Aided Civil and Infrastructure Engineering*, *34*(5), 367–384.

Pantò, B., Macorini, L., & Izzuddin, B. (2022). A two-level macroscale continuum description with embedded discontinuities for nonlinear analysis of brick/block masonry. *Computational Mechanics*, *69*(3), 865–890.

Pereira, M., D'Altri, A., de Miranda, S., & Glisic, B. (2023). Automatic multi-leaf nonperiodic block-by-block pattern generation and computational analysis of historical masonry structures. *Engineering Structures*, *283*, 115945.

Petracca, M., Camata, G., Spacone, E., & Pelà, L. (2023). Efficient constitutive model for continuous micro-modeling of masonry structures. *International Journal of Architectural Heritage*, *17*(1), 134–146.

Quqa, S., Landi, L., & Loh, K. J. (2023). Crack identification using electrical impedance tomography and transfer learning. *Computer-Aided Civil and Infrastructure Engineering*, *38*, 2426–2442.

Rezaie, A., Achanta, R., Godio, M., & Beyer, K. (2020). Comparison of crack segmentation using digital image correlation measurements and deep learning. *Construction and Building Materials*, *261*, 120474.

Rezaie, A., Godio, M., Achanta, R., & Beyer, K. (2022). Machine-learning for damage assessment of rubble stone masonry piers based on crack patterns. *Automation in Construction*, *140*, 104313.

Rezaie, A., Godio, M., & Beyer, K. (2020). Experimental investigation of strength, stiffness and drift capacity of rubble stone masonry walls. *Construction and Building Materials*, *251*, 118972.

Rezaie, A., Godio, M., & Beyer, K. (2021). Investigating the cracking of plastered stone masonry walls under shear–compression loading. *Construction and Building Materials*, *306*, 124831.

Riveiro, B., Lourenço, P. B., Oliveira, D. V., González-Jorge, H., & Arias, P. (2016). Automatic morphologic analysis of quasi-periodic masonry walls from lidar. *Computer-Aided Civil and Infrastructure Engineering*, *31*(4), 305–319.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., & Fei-Fei, L. (2015). Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, *115*, 211–252.

Rytter, A. (1993). Vibrational based inspection of civil engineering structures. Ph.D. Thesis, University of Aalborg.

Serpieri, R., Albarella, M., & Sacco, E. (2017). A 3d microstructured cohesive–frictional interface model and its rational calibration for the analysis of masonry panels. *International Journal of Solids and Structures*, *122*, 110–127.

Stałowska, P., Suchocki, C., & Rutkowska, M. (2022). Crack detection in building walls based on geometric and radiometric point cloud information. *Automation in Construction*, *134*, 104065.

Trovalusci, P., & Masiani, R. (2003). Non-linear micropolar and classical continua for anisotropic discontinuous materials. *International Journal of Solids and Structures*, *40*(5), 1281–1297.

Wang, N., Zhao, Q., Li, S., Zhao, X., & Zhao, P. (2018). Damage classification for masonry historic structures using convolutional neural networks based on still images. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1073–1089.

Yang, X., Li, H., Yu, Y., Luo, X., Huang, T., & Yang, X. (2018). Automatic pixel-level crack detection and measurement using fully

convolutional network. *Computer-Aided Civil and Infrastructure Engineering*, *33*(12), 1090–1109.

How to cite this article: Pereira, M., D'Altri, A. M., de Miranda, S., & Glisic, B. (2024). Crack pattern–based machine learning prediction of residual drift capacity in damaged masonry walls. *Computer-Aided Civil and Infrastructure Engineering*, 1–15. https://doi.org/10.1111/mice.13212

APPENDIX

In this appendix, the calibration of the block-based model developed in D'Altri et al. (2019) is conducted with respect to the experimental campaign on solid clay brick masonry conducted at TU Delft (Jafari and Esposito, 2017; Licciardello and Esposito, 2019), see Figure A1, using the calibration strategy proposed in D'Altri et al. (2019). Particularly, a soft calibration is conducted, given the typical considerable variability of experimental outcomes in masonry. Accordingly, the mechanical parameters are tuned so that the numerical response in small-scale tests becomes rather close to the experimental one (in terms of stiffness, strength, and postpeak behavior), that is, it is within the range of variability of the experimental outcomes. The so-calibrated mechanical properties (Table A1) are then utilized in large-scale simulations. An example of numerical–experimental comparison of the compressive response on a small-scale masonry wallet is shown in Figure A1a, while the comparison of the responses of masonry triplet tests is shown in Figure A1b.

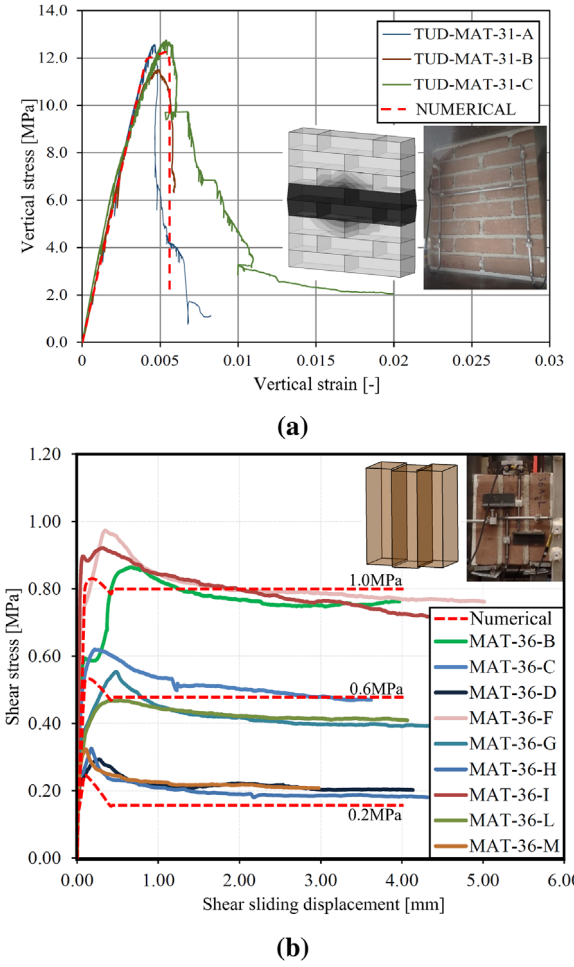


FIGURE A1 Examples of calibration of the block-based model against experimental tests on solid clay brick masonry: (a) compressive response on masonry wallets and (b) shear response on masonry triplets. Block size is equal to 0.22 m × 0.06 m × 0.10 m. Experimental findings are adapted from Licciardello and Esposito (2019).

TABLE A1 Model mechanical properties for solid clay brick masonry.

Contact mechanical properties			
Tensile behavior		**Shear behavior**	
f_t (MPa)	0.12	c (MPa)	0.14
u^F (mm)	0.40	δ^F (mm)	0.40
K_t (N/m^3)	2.0×10^{10}	K_s^c (N/m^3)	1.0×10^{10}
Frictional elastic slip 0.08 mm		ϕ (°)	38.6
Block mechanical properties			
E_B (MPa)	3000	ν (/)	0.17
Density (kg/m^3)	1600	Dilation angle (°)	10
Eccentricity (/)	0.1	f_{b0}/f_{c0} (/)	1.16
K (/)	0.67	Viscosity parameter (s)	1.0×10^{-9}

Uniaxial tensile nonlinear behavior			**Uniaxial compressive nonlinear behavior**		
Stress (MPa)	**Inelastic strain**	d_t (/)	**Stress (MPa)**	**Inelastic strain**	d_c (/)
4.0	0	0	12.0	0	0
0.4	0.004	0.9	12.0	0.004	0
			1.2	0.012	0.9