



**University of
Nottingham**

UK | CHINA | MALAYSIA

Real-Time Assessment of Tunnelling-Induced Damage to Structures within the Building Information Modelling Framework

Thesis submitted to the University of Nottingham for the degree of
Doctor of Philosophy, 29th December 2024.

Ali Gamra

20140438

Supervised by

**Jelena Ninić (UoB)
Bahman Ghiassi (UoB)
Carlos Osorio Sandoval (UoN)
Georgia Thermou (UoN)**

Abstract

During the initial design phases of complex multi-disciplinary systems such as urban tunnelling, the appraisal of different design alternatives can ensure optimal designs in terms of costs, construction time, and safety. To enable the evaluation of a large number of design scenarios and to find an optimal solution that minimises the impact of tunnelling on existing structures, the design and assessment process must be efficient, yet provide a holistic view of model interaction, including Soil-Structure Interaction (SSI) effects. In this thesis, an integrated tunnel design tool is proposed for the initial design phases to predict building damage due to ground settlements induced by tunnelling, leveraging empirical and analytical solutions as well as simulation-based meta-models. Furthermore, the visualisation of ground settlements and building damage categories is enabled by integrating these solutions within a Building Information Modelling (BIM) framework for tunnelling. This approach allows for near real-time assessment of structural damage induced by settlements, considering SSI and the non-linear material behaviour of buildings. Because this approach is implemented on a BIM platform for tunnelling, it offers numerous benefits. Firstly, the design can be optimised directly in the design environment, thus eliminating errors in data exchange between designers and computational analysts. Secondly, the effect of tunnelling on existing structures can be effectively visualised within the BIM by producing risk maps and visualising the scaled deformation field, which allows for a more intuitive understanding of design actions and collaborative design. Having a fully parametric design model and real-time predictions, therefore, enables the assessment and visualisation of tunnelling-induced damage for large tunnel

sections and multiple structures in an effective and computationally efficient way.

Acknowledgements

I would like to express my gratitude to my supervisors, Jelena Ninić and Bahman Ghiassi, for their expert guidance throughout my PhD. Their understanding, support, and belief in my work have been truly invaluable. Their contributions were immensely rewarding to me, and without their encouragement and expertise, I would not have reached this milestone. It was truly and heartfully an honour to collaborate with them on several original research contributions.

I am also deeply thankful to the other academics whose support was pivotal to my journey, including Walid Tizani for helping with the important paperwork and ensuring that I was always on track, Carlos Sandoval for his academic friendship, guidance, and technical support and advice, Alec Marshall for the thoughtful advice and attentiveness to my work, and Georgia Thermou for her guidance and support, among many others. Furthermore, I am grateful to the Centre of Structural Engineering and Informatics (CSEI) research group for their invaluable activities, meetings, and discussions, which helped me broaden my knowledge and research interests, as well as to the Department of Civil Engineering and the University of Nottingham (UoN) for providing the resources, funding, facilities, and many other things that made this research possible.

I extend my heartfelt appreciation to my peers, researchers, and colleagues, as well as members of the PGES, student services, and IT team. Their support, collaboration, and camaraderie made this experience both rewarding and memorable.

A special note of gratitude goes to my mum for her sacrifices and invaluable support, as well as to my family and friends for their understanding and encouragement throughout this journey.

Finally, thank you all for your support, understanding, and belief in me.

List of Publications

Journal Articles

1. Ali Gamra, Jelena Ninić, Bahman Ghiassi, “Optimising synthetic datasets for machine learning-based prediction of building damage due to tunnelling,” *Tunnelling and Underground Space Technology*, Volume 152, 2024, 105961, ISSN 0886-7798, <https://doi.org/10.1016/j.tust.2024.105961>. <https://www.sciencedirect.com/science/article/pii/S0886779824003791>

Presented in Chapters: 1, 2, 3, 4, 5, 6, 7, and 9.

2. Jelena Ninić, Ali Gamra, Bahman Ghiassi, “Real-time assessment of tunnelling-induced damage to structures within the building information modelling framework,” *Underground Space*, Volume 14, 2024, Pages 99-117, ISSN 2467-9674, <https://doi.org/10.1016/j.undsp.2023.05.010>. <https://www.sciencedirect.com/science/article/pii/S2467967423000995>

Presented in Chapters: Abstract, 1, 2, 3, 4, 6, 7, 8, and 9.

Contributions: contributed to the numerical part of the literature, numerical validation and modelling, generating the synthetic dataset used for the training of Machine Learning (ML) as well as fine-tuning of the ML algorithm. In addition to that, the author was the primary contributor to sections 2.4, 3.2.2, 4.1, 4.2, and 4.3, with additional input on other sections, including the abstract, introduction, and conclusions.

Conference Papers and Abstracts

1. Ivan Izonin, Ali Gamra, Oleslav Boychuk, Jelena Ninić, Roman Tkachenko, Stergios Mitoulis, “PCA-NuSVR Framework for Predicting Local and Global Indicators of Tunnelling-induced Building Damage,” *1st International Conference on Smart Automation & Robotics for Future Industry (SMARTINDUSTRY 2024)*, CEUR-WS.org, 2024, Pages 32-46.

Presented in Chapters: 4, 5, 6, 7, and 9.

Contributions: Provided the dataset for the training of the proposed method. In addition to that, the author was the primary contributor to section 2.1, with additional input on other sections, including the abstract and introduction.

2. Ali Gamra, Jelena Ninić, Bahman Ghiassi, “A Comprehensive Parametric Study on the Impact of Underground Tunnelling on Neighbouring Structures Using Machine Learning,” *4th International Symposium of Machine Learning and Big Data in Geoscience*, Cork, Republic of Ireland, August 2023.

Presented in Chapters: 4, 6, 7, and 9.

3. Ali Gamra, Jelena Ninić, “Real-time Tunnelling-induced Damage Prediction of Brittle Structures Using ANNs Integrated with BIM,” *Computational Methods and Information Models in Tunnelling, EURO:TUN 2021*, Bochum, Germany, June 2022.

Presented in Chapters: Abstract, 1, 2, 3, 4, 6, 7, 8, and 9.

4. Ali Gamra, Jelena Ninić, Bahman Ghiassi, “Investigation of tunnelling-induced building damage on nearby surface structures supported by numerical simulations,” *UK Association for Computational Mechanics Conference*, Nottingham, UK, April 2022.

Presented in Chapters: 4, 6, and 9.

Contents

Abstract	i
Acknowledgements	iii
listpublications	iv
List of Tables	xii
List of Figures	xiii
Abbreviations	xx
Chapter 1 Introduction	7
1.1 Related Work	8
1.2 The Gap in the Literature	11
1.3 Research Question and Contributions	12
1.4 Thesis Aim	13
1.5 Thesis Objectives	13
1.6 Thesis Structure	14
Chapter 2 Literature Review	16
2.1 Tunnelling-Induced Settlements	16
2.1.1 The Volume Loss Method: An Overview	18
2.2 Building Damage	22
2.2.1 Definitions of Ground and Foundation Movements	22
2.2.2 Evaluation of Risk of Damage	25
2.2.3 Categories of Damage	26
2.2.4 Concept of Limiting Tensile Strain Method	28
2.2.4.1 Evaluating Strain in a Simple Beam	29
2.2.4.2 Influence of Horizontal Strain	31

2.3	Soil-Structure Interaction	32
2.3.1	Modified Vlasov Method	34
2.3.2	The Relative Stiffness Approach	37
2.4	Numerical Modelling of Masonry Building	38
2.4.1	Smeared Cracking Concepts	40
2.4.2	The Concrete Damaged Plasticity Model	41
2.4.3	Plasticity Parameters of the Concrete Damaged Plasticity Model	43
2.4.4	Stress-Strain Relationship and the Characteristic Length, h	44
2.4.5	Material Modelling and Sensitivity Analysis	46
2.5	Sensitivity Analysis and Sampling	47
2.5.1	Sampling Methods	48
2.5.2	Application to Structural Modelling	48
2.6	Machine Learning Algorithms	49
2.6.1	Artificial Neural Networks	51
2.6.2	Data Preprocessing and Training Optimisation	52
2.6.3	Validation of Model Generalisation	54
2.6.4	Performance Metrics for Model Evaluation	54
2.7	Summary	57
Chapter 3	Methodology Overview	59
3.1	Input Identification and Numerical Modelling	61
3.2	Generating the Synthetic Dataset	62
3.2.1	Integration to Model Prototype	66
3.3	Post-Analysis Evaluation Using Damage Metrics, Machine Learning, and Building Information Modelling	67
3.4	Summary	69
Chapter 4	Numerical Modelling	71
4.1	Mesh Independence Study	72
4.2	Shear Wall and Confined Loading Conditions	76
4.3	Wall Exposed to Settlements	83

4.4	Soil-Structure Interaction using the Euler Bernoulli Beam on a Two Parameter Elastic Foundation Code	89
4.4.1	Advantages, Limitations, and Implications of the EBBEF2p Model	94
4.5	Summary	95
Chapter 5	Generation of Synthetic Dataset	98
5.1	Identification of Input Parameters	100
5.2	Creating the Sample Space	107
5.3	Generating Concrete Damaged Plasticity Material Models	109
5.4	Generating Soil-Structure Interaction Using the Euler Bernoulli Beam on a Two Parameter Elastic Foundation	113
5.5	Model Prototype (Model 0)	115
5.6	Automating Model Generation	117
5.7	High Performance Computing for ‘.inp’ File Execution	120
5.8	An Iterative Procedure for Data Rebalancing	123
5.9	Summary	126
Chapter 6	Evaluation of Local and Global Damage to Buildings - Case Study of Numerical Simulations	127
6.1	Evaluating Building Damage using Various Criteria	130
6.1.1	Case Studies - Local Damage Evaluation	132
6.1.2	Case Studies - Global Damage Evaluation	137
6.1.3	Case Studies - Limiting Tensile Tensile Method Assessment	140
6.1.4	Case Studies - Overall Damage and ε Over the façade	143
6.2	Summary of Methods For All Case Studies	146
6.3	Distribution of Damage Variables	148
6.4	An Iterative Procedure of Data Rebalancing	153
6.4.1	Implementation of the Iterative Procedure	153
6.5	Correlation Between Damage Metrics Across the Entire Set of Simulations	174
6.6	Summary	177

Chapter 7	Machine Learning Training and Evaluation	179
7.1	Training Performance on Different Datasets	181
7.2	Impact of Feature Engineering and Split Ratio on Performance	189
7.3	Artificial Neural Network with Particle Swarm Optimisation	194
7.4	Principal Component Analysis NU-Support Vector Regression Approach on Dataset	197
7.5	Summary	202
Chapter 8	Integration into Building Information Modelling	204
8.1	Algorithmic Implementation	206
8.2	Prediction of Damage Risk in Building Information Modelling Using Meta- Models	209
8.3	Summary	210
Chapter 9	Conclusion, Study Limitations, and Future Work	212
9.1	Study Limitations	217
9.2	Generalizability and Computational Implications	219
9.3	Future Work	220
Bibliography		222
Appendices		240
Appendix A	Code Listings and Implementation Details	241
Appendix B	Dynamo Script for the Evaluation of Building Damage due to Tunnelling, Snapshots	246
Appendix C	Source Code for Model Sampling Using the Modified Latin Hypercube Method	250
Appendix D	Source Code for Model Assembly in ABAQUS Using the ABAQUS Python Development Environment and Model Prototype	258

List of Tables

2.1	Classification of visible damage to walls with particular reference to ease of repair of plaster and brickwork or masonry (adjusted from (Burland et al. (1977))).	27
2.2	Typical values of maximum building slope and settlement for damage risk assessment (adjusted from (Rankin (1988))).	28
2.3	Indicative values for t of low-rise housing (adjusted from (Skinner and Charles (2004))).	29
2.4	Relationship between category of damage and limiting tensile strain (ε_{lim}) (after (Boscardin and Cording (1989))).	30
2.5	Recommendations for estimating the value of h	44
4.1	Model material properties and applied loads.	78
4.2	Comparison between FEM and experimental results for the global and local assessments of building damage.	88
4.3	Input parameters for beam on modified Vlasov foundation, using the example (Teodoru (2009b)). H_s denotes the depth of the soil medium (m).	90
5.1	Input parameters relating to the VL method.	101
5.2	Input parameters related to soil while using the EBBEF2p method . . .	102
5.3	Input parameters related to the building material using the CDP model.	103
5.4	Input parameters relating to building façade geometry (2D).	104
5.5	Input parameters related to the contact area between building and soil.	104
5.6	Summary of input parameters in FEM implementation.	105
5.7	Correlations between dependent and independent parameters.	106

5.8	Constraints based on physical and numerical integrity.	107
5.9	Material properties for CDP development for a random selection of input samples.	112
5.10	Relevant parameters for the evaluation of SSI using EBBEF2p model for the same sample points of the CDP.	113
5.11	Other model component specifications.	119
6.1	Local building damage evaluation metrics of Figure 6.4.	134
6.2	Global building damage evaluation metrics of the case studies presented in Figure 6.7.	139
6.3	Models input parameters for LSTM metrics.	142
6.4	Results for LSTM models presented in Figure 6.9.	143
6.5	% of damaged areas, and ε_t values of Figure 6.11 case studies.	144
6.6	Categories of damage for the models presented earlier as case studies for each damage criteria in comparison to their global categorisation of damage (Part 1).	147
6.7	Categories of damage for the models presented earlier as case studies for each damage criteria in comparison to their global categorisation of damage (Part 2).	147
6.8	Average values (normalised) for each of $Y_{i \text{ Region } 1, 2 \text{ or } 3}$ of input parameter using μ_{X_i} and σ_{X_i} with % of total values falling within each region. . . .	161
6.9	Minimum values of input parameters used throughout iterations.	163
6.10	Maximum values of input parameters used throughout iterations.	163
6.11	Categories of damage of metrics according to literature.	174
7.1	Hyperparameter limits for each model.	181
7.2	Performance metrics for different models across various damage metrics. .	183
7.3	Cross-validation using the Max. Crack Width as output variable and a total of 5 folds with a 20-80 split.	187
7.4	Cross-validation using $\varepsilon_{h,\max}$ as output variable.	188

7.5	ANN performance with different transformation techniques on local and global outputs.	192
7.6	Performance metrics for different thresholds across global and local criteria.	193
7.7	Performance metrics for different data splits across global and local outputs.	193
7.8	Results of ANN-PSO performance on local and global outputs.	196
7.9	Values of the performance metrics for predicted local-global damage indicators via the proposed PCA-NuSVR framework for the assessment of building damage due to tunnelling.	198
7.10	R^2 -values of the local and global indicators for the assessment of building damage using different ML-based regressors.	201
8.1	Parameters for the description of geometric-semantic BIM for tunnelling.	205
8.2	Classification of building damage with the colour code for visualisation using the LTSM damage classification system of (Burland et al. (1977)). .	207

List of Figures

2.1	Description of the settlement trough, building geometry, sagging, and hogging zones.	19
2.2	3D settlement profile induced by tunnelling.	21
2.3	Definition of foundation movement (inspired by the works of Burland et al. (1977)).	23
2.4	Damage curves derived from case studies categorising damage based on β and $\varepsilon_{h,\max}$ according to (Boscardin and Cording (1989)).	32
2.5	Family tree of AI types including ML, deep learning, computer vision and robotics (inspired by the work of (Huang et al. (2021))).	50
2.6	Structure of a 4-layer ANN for predicting building strain due to tunnelling (O_1 to O_m corresponding to $Y_i(X_n, w)$).	52
3.1	Concept for real-time damage assessment within BIM.	60
3.2	Principal stages of methodology towards research goals.	61
3.3	Preparation stage (pre-analysis) schema of the methodology.	63
3.4	Assessment stage (post-execution) of FEM models (back-end of the methodology).	68
4.1	Assembly of bar in tension with an applied maximum displacement of 0.05 mm.	73
4.2	F-D diagrams of the simple bar in tension using different constitutive approaches including (a) Unregularised stress-strain (b) Regularised stress-strain (c) and Internally Regularised stress-displacement.	75
4.3	Macro-modelling of the shear panel.	78

4.4	Failure patterns of experimental wall (After Lourenco (1996)).	79
4.5	Tension damage distributions contours maps from FEM simulations (values > 0.99 indicate fully developed damage “red”).	80
4.6	Comparison of diagrams between experiment curves and FEM output. . .	80
4.7	Parametric studies on CDP plasticity parameters (a) ψ_{CDP} (b) e_{CDP} (c) σ_{0b}/σ_{0c} (d) k_{CDP} (e) μ_{CDP}	84
4.8	Numerical modelling of Giardina (2013) experiment of masonry façade exposed to settlements.	86
4.9	FEM simulation output of masonry façade under hogging mode of deformation, illustrating tension damage distributions (d_t).	87
4.10	Applied settlements and global displacements of the masonry façade FEM vs Experimental.	89
4.11	Beam on elastic foundation: half-plane geometry and medium property (Teodoru (2009b)).	90
4.12	Comparing results using the EBBEF2p implementation on benchmark data from (Teodoru (2009b)).	92
4.13	Comparison of settlement profiles with varying parameters values using the EBBEF2p code (a) E varied (b) E_s varied.	92
4.14	Actual and predicted settlements of the west wall due to tunnelling. . . .	93
5.1	Generating synthetic dataset methodology overview.	99
5.2	Schematic of tunnel–SSI showing key input parameters including tunnel depth (Z_0), diameter (D), offset (e), building height (H), length (L), and settlement trough width (i_x). The settlement profile follows a Gaussian curve (Equation 2.1). Hatched area indicates (VL). No openings are considered in the façade ($O = 0\%$). The soil–structure interface is represented by the normal and shear stiffness parameters (K_{norm}) and (K_{shear}), respectively.	100
5.3	Sample distributions of material parameters.	110

5.4	Sample distributions of soil and tunnel parameters.	110
5.5	Sample distributions of additional soil and building parameters.	111
5.6	CDP material behaviour in tension, compression and corresponding damage degradation, respectively.	112
5.7	SSI of buildings with various levels of resistance to deformation.	114
5.8	Model prototype with a basic definition and properties of the various components, including the application of load and area of the contact interface.	117
5.9	Model assembly in FEM, comprising geometry, material (CDP), applied settlements SSI (boundaries), and numerical settings.	121
5.10	Software to transfer, access, and execute ABAQUS ‘.inp’ files.	122
5.11	Flowchart of the iterative procedure including feature selection, statistical exploration, and range adjustment.	124
6.1	Flowchart illustrating the process of damage evaluation and processing pipeline.	128
6.2	Damage distributions (DAMAGET) of building segment exposed to sagging mode of deformation.	131
6.3	Input parameter distributions (in normalised terms) for the models used for the local damage evaluation of the selected case studies in Figure 6.4.	134
6.4	Several categories of building damage are observed for the various configurations (DAMAGET contours). In addition to presenting the location of Maximum Crack Width.	135
6.5	Foundation nodes displacements with increased values of $S_v(x)$ until reaching $S_{v,max}$	136
6.6	Initiation and propagation of Maximum Crack Width values with increased values of normalised $S_v(x)$	137
6.7	Example structures with significant damage according to global metrics (illustration of DAMAGET contours).	139

6.8	Input parameter distributions (in normalised terms) for the models used for the global damage evaluation of the selected case studies in Figure 6.7.	140
6.9	Building location WRT settlement trough for the assessment of building damage according to the LTSM method.	142
6.10	Input parameter distributions (in normalised terms) for the models used for the global damage evaluation of the selected case studies in Figure 6.11.	144
6.11	Evaluating damage according to damaged elements, illustrating DAM-AGET contours on building façade.	145
6.12	Original distributions of local damage metrics with simulations.	149
6.13	Initial distributions of global metrics with simulations.	151
6.14	Initial distributions of LTSM damage criteria and $S_{v,max}$	152
6.15	Initial distributions of additional metrics (Damaged Area and ε_t) with simulations.	152
6.16	Importance score of input parameters using the RFR, totalling a sum of 1.	155
6.17	Damage distributions in normalised values of the Max. Crack Width against the most influential parameters and for the original dataset. . . .	157
6.18	Damage distributions in normalised values of $\varepsilon_{h,max}$ against the most influential parameters and for the original dataset.	159
6.19	Distribution of input parameters between iterations using radar charts. .	165
6.20	Distributions of local metrics with increased iterations (y-axis normalised between 0 and 1).	166
6.21	Distributions of local metrics with increased iterations (y-axis normalised).	167
6.22	Distributions of additional global and local metrics with increased iterations (y-axis normalised).	169
6.23	Damage distributions in normalised values of the Max. Crack Width against the most influential parameters and for the entire dataset.	171
6.24	Damage distributions in normalised values of $\varepsilon_{h,max}$ against the most influential parameters and for the entire dataset.	172
6.25	Importance score of input parameters using the RFR for the entire dataset.	173

6.26	Closest 4 correlations found between pairs of metrics using the entire dataset.	175
6.27	Furthest 4 correlations found between pairs of metrics using the entire dataset.	176
7.1	Actual vs predicted values using the Max. Crack Width metric. Illustrating standardised values too (sometimes helping in visualising data better).	184
7.2	Actual vs predicted values using the $\varepsilon_{h,\max} \times 10^{-3}$ metric.	185
7.3	Loss function (MSE) convergence for ANN with increased iterations. . . .	186
7.4	Transformed local output (Max. Crack Width) variable using different transformation techniques.	190
7.5	Transformed global output ($\varepsilon_{h,\max}$) variable using different transformation techniques.	191
7.6	Schema for training of ANN optimised by PSO.	195
7.7	Actual vs predicted values plots in addition to the convergence performance of the PSO.	196
7.8	Flowchart of the proposed PCA-NuSVR framework (Izonin et al. (2024)).	199
7.9	Actual vs predicted values using the PCA-NuSVR model on testing sets for the (a) local output (b) global output.	200
8.1	Assembly of the whole tunnel-soil-building interaction model in BIM using dynamo in Revit.	206
8.2	User interface for input of design parameters (left) and tunnel model with visualised settlements and damage category (right).	207
8.3	Algorithm for model generation and visualisation of settlements and damage category.	208
8.4	Prediction of tunnelling-induced damage category based on meta-model for different values of e (these are 0 D , 1.5 D , 2.25 D , 3 D , respectively). “left to right in figures” and same for a), b), and c). D is the μ value obtained from the sampling procedure after the iterative procedure and given as 6.6 m.	209

8.5	Real-time prediction of tunnelling-induced damage risk for multiple buildings in the design environment.	211
B.1	Nodes describing the input parameters (identified from the pink background in Dynamo) for the tunnel, soil and building including the additional geometry-related parameters.	247
B.2	Nodes describing the actions (identified from the greenish cyan background in Dynamo) for the geometry of the former input elements. Including a view command and a transparency percentage allowing the effective visualisation of tunnel geometry, soil and building (including position). . . .	248
B.3	Nodes describing the outputs (identified from the orange background in Dynamo) to calculate settlements, and building damage evaluation using the Machine Learning (ML) and damage level visualisation using colour code and colour range which is visualised directly on the building geometry.	249

Abbreviations

AEC Architecture, Engineering, and Construction.

AI Artificial Intelligence.

AL Active Learning.

ANN Artificial Neural Networks.

APDE ABAQUS Python Development Environment.

API Application Programming Interface.

BIM Building Information Modelling.

BO Bayesian Optimisation.

CDP Concrete Damaged Plasticity.

COM Chakeri & Onver method.

CSEI Centre of Structural Engineering and Informatics.

DR Deflection Ratio.

EBBEF2p Euler Bernoulli Beam on Two Parameter Elastic Foundation.

EE Elementary Effect.

EI Expected Improvement.

F-D Force-Displacement.

FDM Finite Difference Method.

FEM Finite Element Method.

GB Gradient Boosting.

GF Greenfield.

HPC High Performance Computing.

HPO Hyperparameter optimisation.

HS2 High Speed Two.

LGBM Light Gradient Boosting Machine.

LH Latin Hypercube.

LR Learning Rate.

LTSM Limiting Tensile Strain Method.

MAE Mean Absolute Error.

MCDA Multi-Criteria Decision Analysis.

ML Machine Learning.

MLH Modified Latin Hypercube.

MSE Mean Squared Error.

NuSVR Nu Support Vector Regression.

OAAT One-At-A-Time.

PCA Principal Component Analysis.

POD Proper Orthogonal Decomposition.

PSO Particle Swarm Optimisation.

R Pearson Correlation Coefficient.

Nomenclature

$[C(u, \dot{u})]$ Damping matrix (N·s/m)

$[K(u)]$ Displacement-dependent stiffness matrix (N/m)

$[k_e]$ Stiffness matrix of the flexural beam element (N/m)

$[k_f]$ Stiffness matrix of the elastic foundation (first subgrade) (N/m)

$[k_r]$ Stiffness matrix of the rigid base (second subgrade) (N/m)

$[M(u, \dot{u})]$ Mass matrix (kg)

α_f Angular strain

β Angular distortion (relative rotation)

ΔL Change in building length (m)

ΔS_v Relative or differential settlement between two points (m)

Δ Relative deflection (m)

γ Dimensionless shape parameter

γ_s Soil unit weight (kN/m³)

λ Learning rate

μ_{CDP} Viscosity parameter in CDP model

μ_{X_i}	Mean of input parameter X_i
ν	Poisson's ratio
ν_s	Poisson's ratio of the soil
$\phi(z)$	Function describing vertical deflection variation with depth
ϕ	Threshold for feature importance
Ψ	Bias term in neural network
ψ_{CDP}	Dilation angle in CDP model
σ_c	Compressive stress (MPa)
σ_m	Compressive strength at failure (MPa)
σ_r	Required face support pressure (kPa)
σ_s	Surface surcharge (kPa)
σ_t	Tensile stress (MPa)
σ_w	Building stress
σ_{R^2}	Standard deviation of R^2
σ_{rRMSE}	Standard deviation of $rRMSE$
σ_{X_i}	Standard deviation of input parameter X_i
θ_f	Rotation angle at foundation or ground level
ε_c^p	Plastic strain in compression
ε_c^{in}	Inelastic strain in compression
ε_h	Horizontal strain
ε_m	Equivalent inelastic strain at failure

ε_t	Total tensile strain
ε_t^p	Plastic strain in tension
ε_t^{ck}	Cracking strain
ε_{crit}	Critical tensile strain
$\varepsilon_{h,max}$	Maximum horizontal tensile strain
ε_{lim}	Limiting tensile strain
$\varepsilon_{t,max}$	Maximum total tensile strain
φ	Angle of internal friction of the soil (°)
$\{\ddot{u}\}$	Acceleration vector (m/s ²)
$\{\dot{u}\}$	Velocity vector (m/s)
$\{d_e\}$	Nodal degrees of freedom vector (m, rad)
$\{F(t, u, \dot{u})\}$	External force vector (N)
$\{R_e\}$	Distributed load vector (N and N·m)
$\{S_e\}$	Nodal load vector (N and N·m)
$\{u\}$	Nodal displacement vector (m)
A	Empirical parameter in settlement equation
B	Building width
c	Cohesion (Mohr–Coulomb parameter) (kPa)
D	Tunnel diameter (m)
d_c	Damage variable in compression
d_t	Damage variable in tension

E_s	Young's modulus of the soil (kPa)
e_t	Tunnel offset from centreline
e_{CDP}	Flow potential eccentricity in CDP model
E_{init}	Initial elastic modulus
Err	Error between predicted and target output
$f()$	Activation function
G_f	Fracture energy (N/m)
H	Building height (m)
h	Characteristic length (m)
H_s	Influence zone depth in the soil medium (m)
$I(Y_i)$	Importance score of output Y_i
i_y	Width parameter along longitudinal axis (m)
i_z	Distance from the tunnel centreline to the inflection point (m)
K	Trough width parameter
k_{fold}	Number of folds in cross-validation
k_{CDP}	Ratio of stress invariants in CDP model
k_{norm}	Normal subgrade stiffness (N/m ³)
k_{shear}	Shear subgrade stiffness (N/m ³)
$K_{shear\ coef.}$	Shear interaction coefficient at soil-structure interface
L	Building length (m)
L_t	Tunnel length (m)

L_t	Tunnel length
L_w	Building length
L_{12}	Distance between two nodes (m)
LH	Latin Hypercube sampling
m	Parameter in compression constitutive equation
$m_{pattern}$	Number of training patterns
n	Set of input parameters
$n_{combined}$	Combined set of significant parameters
n_{global}	Globally significant input parameters
n_{local}	Locally significant input parameters
q	Uniformly distributed load per unit length (N/m)
S	Estimated settlement (mm)
s	Slope
$S_v(x)$	Ground settlement function relative to the transverse distance x (m)
$S_{v,max}$	Maximum settlement at the tunnel centreline (m)
u_t	Cracking displacement
U_x	Horizontal displacement
w_{ij}	Weight from input to hidden layer
w_{jk}	Weight from hidden to output layer
x	Perpendicular distance to the tunnel centreline (m)
X_i	Input neuron

X_i	Input parameter i
y	Distance from tunnel face along longitudinal axis (m)
y_b	Building displacement
Y_i	Output neuron
Y_i	Output response for input parameter X_i
y_{face}	Distance from tunnel entrance to the tunnel face (m)
$Y_{i,\text{Region 1}}$	Average output in Region 1 for Y_i
$Y_{i,\text{Region 2}}$	Average output in Region 2 for Y_i
$Y_{i,\text{Region 3}}$	Average output in Region 3 for Y_i
z	Depth in the vertical direction (m)
Z_0	Tunnel depth (m)

Chapter 1

Introduction

Growing urbanisation, expansion of cities, and the demand for national and transnational high-speed mobility have raised the need for efficient and environmentally friendly transport infrastructure. In addition, due to the limited available surface space in cities, development projects often rely on underground space. To accommodate this, major urban tunnelling projects have been constructed in the last few decades (e.g., Crossrail and High Speed Two (HS2) in the UK). Despite the advanced technologies used for underground construction in these projects, existing overground structures have been significantly damaged due to tunnelling-induced settlements (DeJong et al. (2019); Milillo et al. (2018)).

The response of existing structures to tunnelling-induced ground movement in urban tunnelling is a fully coupled problem of tunnel-SSI. Therefore, many design considerations, such as tunnel location, existing buildings and infrastructure above and below ground, the construction method, related construction details (e.g., driving parameters), geometrical properties (e.g., depth, diameter, lining thickness), ground behaviour, and possible critical geological conditions, play an important role in the selection of the optimal design solution (Loganathan and Poulos (1998)). This optimal design solution must satisfy several design criteria, such as the structural integrity and durability of the tunnel struc-

ture for demanding use over 100+ years, face stability during tunnel construction, and control of tunnelling-induced settlements to minimise the impact on the existing environment. This thesis specifically focuses on the response of buildings founded on shallow strip foundations, which are common in urban environments and particularly susceptible to tunnelling-induced ground movements (Burland et al. (1977); Boscardin and Cording (1989)). To achieve this, analytical, empirical, and numerical methods are employed to evaluate these design objectives.

1.1 Related Work

Tunnelling activities inevitably induce ground displacements, which can adversely affect nearby structures. These displacements depend on several geotechnical and operational factors, and their prediction has been the focus of a wide body of empirical, analytical, and numerical research (Mair et al. (1993); Broms and Bennermark (1967); Powderham (1994); Loganathan and Poulos (1998)).

In the presence of surface structures, the interaction between soil and structure alters the settlement induced by ground tunnelling. To relate the strain on the building to the ground settlement, certain assumptions are required. One of the approaches that simulate the interaction is the Limiting Tensile Strain Method (LTSM), a simple 2D analytical method that represents the structure as a linear elastic beam model with its geometrical properties and stiffness, enabling a fully decoupled soil-structure analysis. Due to its simplicity, LTSM has disadvantages that may lead to conservative results, in some cases underestimating the predicted outcome (Burland and Wroth (1975)). Over the last three decades, many authors have proposed analytical solutions for the prediction of tunnelling-induced damage in structures (Franzius (2003); Boscardin and Cording (1989); Potts and Addenbrooke (1997); Goh and Mair (2011)). A comparison of different analytical solutions for SSI due to tunnelling can be found in (Giardina et al. (2018)).

To tackle the disadvantages of analytical and empirical methods, numerical models are used to evaluate a broad range of possible outcomes, including non-linear responses (Burland et al. (2002); Potts and Zdravković (2001)). Timoshenko beams founded on an elastic continuum half-space with rigid links have been employed to infer the maximum building strains based on a “direct strain-based approach” (Franza et al. (2020)). An equivalent beam model has also been proposed for the assessment of tunnelling-induced damage in masonry structures with pre-existing cracks (Acikgoz et al. (2021)). In parallel, extensive multi-disciplinary research has been carried out to promote the development of numerical models and design concepts capable of addressing the manifold complex interactions and processes in urban tunnelling (Meschke (2018)). As a result, sophisticated, process-oriented computational models have been developed to capture various aspects of mechanised tunnelling, including SSI (Ninić et al. (2017a, 2014)). Advanced computational models incorporating the non-linear behaviour of soil, buildings, and SSI, following either a coupled or an uncoupled approach, have also been used for the prediction of tunnelling- or settlement-induced damage to structures (Son and Cording (2005); Giardina (2013); Yiu et al. (2017); Fargnoli et al. (2015); Boldini et al. (2018); Miliziano and de Lillis (2019)). However, these advanced 3D computational models are typically characterised by a high degree of detail at the cost of long computation times. To address this issue, parallelisation strategies for High Performance Computing (HPC) have been applied (Bui and Meschke (2020); Ninić et al. (2019)).

Another alternative solution to this problem is the substitution of computationally expensive simulations with surrogate models trained offline (Freitag et al. (2018); Ninić et al. (2017b)). This technique has recently been applied to assess tunnelling-induced damage to structures (Obel et al. (2020); Cao et al. (2020a, 2022)). Firstly, Obel et al. (2020) developed a meta-model-based approach to predict the non-linear structural response to tunnelling-induced settlements, investigating two surrogate modelling techniques (response surfaces and neural networks) to replace numerical models of a façade exposed to analytically calculated surface settlements. Cao et al. (2020a) proposed an approach

in which two decoupled Finite Element Method (FEM) models were developed to calculate the non-linear response of the façade and the surface settlements resulting from tunnel-SSI, and were then coupled using two types of Artificial Neural Networks (ANN) in combination with Proper Orthogonal Decomposition (POD) to optimise TBM parameters and minimise the impact of tunnelling on the building. The authors extended their approach to 3D building models with consideration of polymorphic uncertainty (Cao et al. (2022)).

In the last decade, for large infrastructural projects that constitute complex multidisciplinary systems, the BIM framework has been increasingly employed due to several features related to information management, processing, visualisation, and analysis throughout the project life-cycle (Smith (2014); Daller et al. (2016)). A BIM can store geometrical-semantic data about the project, such as ground, tunnels, above-ground infrastructure, and all parameters associated with these components, within its repository (Koch et al. (2017)). Furthermore, BIM facilitates the processing of data to generate meaningful information that can be effectively presented and visualised, and ultimately used to support decision-making. To enable a seamless workflow, the data in a BIM is organised in an object-oriented way, allowing links to be established between objects (Eastman et al. (2008)). The final step in the BIM system is analysis.

One of the many benefits of the data obtained from a BIM is that it can be analysed using a variety of methods developed over the years, which can be significant in the decision-making process. For example, in urban tunnelling, a complex system requires sophisticated analysis to identify a viable option for construction. BIM permits the integration of complex models and effective analysis tools, enabling multiple scenarios to be easily evaluated. This results in a powerful means of supporting decision-making, particularly during the early design phases.

1.2 The Gap in the Literature

BIM for tunnelling has been successfully linked to FEM simulations for applications in both urban mechanised tunnelling and conventional tunnel excavation Ninić et al. (2017b, 2020); Alsahly et al. (2020); Ninić et al. (2021); Fabozzi et al. (2021), while Providakis et al. (2019) used a BIM platform to process and visualise tunnelling-induced settlements and the risks posed to existing infrastructure based on empirical solutions. Both approaches, the assessment of design alternatives in BIM based on FEM simulations or empirical models, have shown promising results. However, each has limitations: the empirical model applied in Providakis et al. (2019) neglects SSI effects and non-linear structural behaviour. The assessment of tunnel design in a BIM environment based on FEM simulations is computationally expensive and may therefore limit the number of design alternatives that can be studied. Other recently developed approaches for the efficient investigation of building response to tunnelling and different design alternatives have been implemented as stand-alone tools Cao et al. (2020a, 2022), and thus lack integration with design platforms (e.g., BIM) for more holistic assessment.

In addition to the above gap, a further two gaps were identified in the literature. These are presented below as follows:

- There is currently no comprehensive strategy for sampling model input parameters when addressing complex tunnel–SSI problems computationally, including studies that aim to predict the influence of individual model parameters on the overall response of the system, such as in (Cao et al. (2020a, 2022); Obel et al. (2020); Giardina et al. (2015); Son and Cording (2005)). Developing a foundational sampling procedure is fundamental to establishing a robust modelling approach that considers all possible configurations of the model.
- Building damage due to tunnelling has been investigated from various perspectives in numerous studies, such as in (Burland and Wroth (1974); Burland et al. (1977);

Rankin (1988); Charles and Skinner (2004); Boscardin and Cording (1989); Burland (1997); Son and Cording (2005); Giardina et al. (2015); Cao et al. (2022)). However, with the exception of Giardina (2013), none have addressed damage comprehensively, particularly across multiple buildings, by providing an overview of the different damage assessment models and categorisation threshold distributions. This includes the use of both localised criteria (e.g., crack development) and global criteria (e.g., tilt (t), slope (s), horizontal tensile strain ($\varepsilon_{h,\max}$), and Total Damaged Area (A_{damaged})). By conducting such a comprehensive comparative study, one may draw insightful conclusions regarding the perceived levels of different damage types for a single model, and for the entire dataset as a whole, thereby achieving a holistic local–global assessment of building damage.

1.3 Research Question and Contributions

Research Question. How can a BIM-integrated surrogate-modelling framework be developed to provide rapid, yet accurate, predictions of tunnelling-induced building damage while explicitly accounting for non-linear building response to SSI effects?

Scientific Contributions

1. **Comprehensive sampling strategy:** a sampling-with-constraints algorithm that explores several hundreds feasible tunnel–soil–structure configurations (addresses Gap 1).
2. **Validated FEM dataset:** high-fidelity simulations covering the sampled space, publicly released for reproducibility.
3. **Local–global damage framework:** integration of crack-based, strain-based and total-damaged-area criteria across multiple buildings (addresses Gap 2).

4. **Surrogate models:** Principal Component Analysis (PCA)-Nu Support Vector Regression (SVR) and ANN achieving between (1.5% – 9%) Relative Root Mean Squared Error (rRMSE) relative to FEM results across all output variables, with a speed-up of near-real-time.
5. **BIM plug-in (Dynamo):** a user-friendly tool that brings the surrogate into BIM, enabling designers to assess damage scenarios in real time.

1.4 Thesis Aim

The overarching aim of this thesis is to develop, validate, and deploy a BIM integrated surrogate-modelling framework that delivers rapid yet accurate predictions of tunnelling-induced building damage, while explicitly capturing the non-linear response arising from SSI. The framework will be released as an interactive Dynamo plug-in, enabling engineers to explore alternative tunnel–soil–building configurations in real time and to make informed design decisions during the early stages of a project.

1.5 Thesis Objectives

- **Model verification and validation (O1)** Rigorously verify and validate high-fidelity FEM models of tunnel–SSI, including (i) a shear wall under combined loading and (ii) a masonry façade subject to settlement profiles.
- **Constraint-aware sampling strategy (O2)** Design and implement a sampling scheme with explicit physical and numerical constraints, capable of generating several tens of thousand feasible tunnel–soil–building scenarios per study.
- **Generation of a validated FEM dataset (O3)** Produce and release an openly accessible dataset of ≈ 1000 3D FEM simulations of 2D façade that span the sam-

pled parameter space and serve as ground truth for surrogate training.

- **Local–global damage-assessment framework (O4)** Formulate and code a unified set of local (crack- and strain-based) and global (e.g., total-damaged-area) damage metrics for single and multiple buildings, calibrated against published experimental and field data.
- **Global Sensitivity Analysis (SA) (O5)** Quantify the influence of each tunnel, soil, and structural parameter on the damage metrics, thereby guiding surrogate-model architecture and input selection.
- **Surrogate-model development and benchmarking (O6)** Train, optimise, and benchmark Machine Learning (ML) that reproduce FEM responses with as little as $\leq 1.5\%$ errors and achieve speed-ups considered near-real-time.
- **BIM integration and real-time decision support (O7)** Embed the best-performing surrogate in a Dynamo-for-Revit plug-in that (i) auto-extracts geometric and material data from BIM, (ii) returns damage predictions in under one second, and (iii) visualises risk levels to support iterative tunnel-alignment and building-mitigation studies.

1.6 Thesis Structure

- **Literature Review (Chapter 2):** Reviewing current research and methods across various multidisciplinary aspects of the problem, including tunnelling-induced settlements, building damage, SSI, numerical modelling, and ML.
- **Methodology Overview (Chapter 3):** Describing the sequence linking the research objectives and highlighting the selected methods. This comprehensive overview covers all steps from the identification of input parameters through to the BIM implementation and application.

- **Numerical Modelling (Chapter 4):** Detailed exploration of the numerical modelling of building materials under combined loading conditions, including settlements. In addition, the analytical validation of the SSI model is presented.
- **Generation of Synthetic Dataset (Chapter 5):** Building on the findings from the literature while demonstrating the use of Python and the ABAQUS Python Development Environment (APDE) to automate the generation of model scenarios and the preparation steps for simulation execution.
- **Evaluation of Local and Global Damage to Buildings – Case Study of Numerical Simulations (Chapter 6):** Presentation of local and global damage assessments based on numerical case studies. This is followed by a discussion of the differences between model outputs and the correlations between metrics. Furthermore, the overall distribution of damage across all simulations is illustrated, with rebalancing techniques applied (e.g., performing SA and other statistical methods) to enhance sparsely populated damage level regions. The entire damage assessment process is automated using Python scripting to minimise human error.
- **Machine Learning Training and Evaluation (Chapter 7):** Using the dataset from Chapter 6, the performance of various ML algorithms is evaluated.
- **Integration into Building Information Modelling (Chapter 8):** Implementation of the trained ML algorithm to evaluate building damage within BIM through scripting. This includes the development of parametric input selection and prediction of settlements and damage for visualisation purposes.
- **Conclusion, Study Limitations, and Future Work (Chapter 9):** Summarising the main research contributions, identifying the current limitations of the study, and proposing directions for future research.

Chapter 2

Literature Review

2.1 Tunnelling-Induced Settlements

Ground movements due to tunnelling are inevitable, as it is nearly impossible to create a void by excavation and provide an infinitely stiff lining to fill it perfectly. Therefore, a certain degree of ground deformation occurs at the depth of the tunnel. This, in turn, triggers a chain of movements, resulting in settlements at the ground surface. These settlements become more significant as the tunnel depth (Z_0) decreases.

Tunnelling-induced settlements can be classified based on their primary causes: (i) settlements due to ground loss during tunnel excavation, (ii) settlements resulting from tunnel lining deformation (particularly relevant for large-diameter, shallow tunnels), and (iii) settlements caused by post-construction soil consolidation (Guglielmetti et al. (2008)). The first two are often associated with short-term, undrained conditions, while the third typically occurs over longer periods.

Early investigations of tunnelling-induced settlements based on field observations are notably found in the works of (Peck (1969); Attewell and Woodman (1982); Attewell et al. (1986)). It was empirically determined that the shape of the settlement trough (in both

longitudinal and transverse directions) followed a form similar to a normally distributed (Gaussian) curve, commonly referred to as the “Greenfield (GF)” in geotechnical terms. Despite their widespread use, these methods were limited to a narrow set of parameters, including tunnel depth, diameter, and the Volume Loss (VL) parameter.

Later investigations led to the development of numerous analytical models that incorporated additional parameters, improving the accuracy of settlement predictions. These included the construction method, tunnel-driving details, initial stress state, radial displacement due to ground loss, tunnel ovalisation, and the stress–strain behaviour of the surrounding soil (Loganathan and Poulos (1998); Verruijt and Booker (1996)). Furthermore, for different soil types, Franza and Marshall (2019) developed empirical and semi-analytical expressions for sandy soils through experimental investigations, capturing key effects from parameters such as the cover-to-diameter ratio and relative density. Additionally, semi-analytical methods were developed for predicting ground movements in multi-layered clayey soils, based on a modified version of Loganathan and Poulos (1998) and combined with elastic equivalent theory (Cao et al. (2020b)).

With the development of computational methods, the use of FEM to predict tunnelling-induced settlements has increased. FEM plays a key role in addressing some of the deficiencies found in previous empirical and analytical methods.

For example, Ninić and Meschke (2015); Giardina et al. (2015); Yiu et al. (2017); Cao et al. (2022) employed FEM to generate 2D and 3D tunnel–soil models, investigating Tunnel Boring Machine (TBM) operational and design parameters such as slurry pressure, grouting pressure, non-linear building behaviour, and time-dependent construction processes. In addition to spatial and time analysis, FEM enables the modelling of complex non-linear behaviours of the tunnel, soil, and their interaction. It permits accurate representation of ground layers, groundwater tables, various material properties, and more.

Later developments introduced the integration of ML for near real-time prediction of

tunnelling-induced settlements, thereby overcoming the computational expense and time-intensive nature of FEM methods (Ninić et al. (2018, 2017a)).

2.1.1 The Volume Loss Method: An Overview

As previously mentioned, the settlement trough is generally expressed in terms of a Gaussian distribution at a point source, for example, the centre of the excavation. Equation 2.1 shows the Gaussian function that can be used to describe the surface settlement trough, as proposed by Peck based on studied case histories of field-observed data (Peck (1969); O'Reilly and New (1982)).

$$S_v(x) = S_{v,\max} \exp\left(-\frac{x^2}{2i_x^2}\right) \quad (2.1)$$

Here, $S_v(x)$ is the ground settlement function relative to the distance from the centreline (m), $S_{v,\max}$ is the maximum settlement on the tunnel centreline (m), x is the perpendicular distance to the tunnel centreline, and i_x is the perpendicular distance from the tunnel centreline to the point of inflection on the surface settlement trough in the transverse direction (m). A typical settlement trough in the transverse direction is illustrated in Figure 2.1. The change of slope occurs at the inflection point, which separates the regions of Hogging and Sagging modes.

In clayey soils, it is assumed that the volume of surface settlements equals the ratio between the volume of excavated soil and the theoretical volume of tunnel per unit length, which is denoted as VL (Potts and Addenbrooke (1997)). This assumption applies primarily to the short-term undrained response of the ground. Based on this assumption, and the assumption that the settlement profile can be described using a Gaussian curve, the maximum settlement $S_{v,\max}$ can be calculated accordingly.

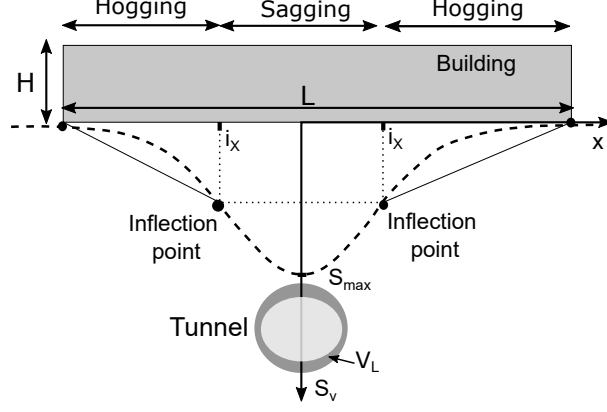


Figure 2.1: Description of the settlement trough, building geometry, sagging, and hogging zones.

$$S_{v,\max} = \sqrt{\frac{\pi}{2}} \frac{V L D^2}{4 i_x} \quad (2.2)$$

where D is the tunnel diameter (m). It is common to find values of VL in the range of 0.2–5% according to various sources including (Mair et al. (1996); Marshall et al. (2012); Burland et al. (2001a)). Similarly, the inflection point, where $i_x = K \times Z_0$, defines the trough width (m). Here, K is a parameter that typically ranges between 0.4 and 0.7 (O'Reilly and New (1982)).

As an example of a more advanced model, Chakeri and Ünver (2013) proposed an equation for estimating the maximum ground settlement ($S_{v,\max}$) based on 3D FDM modelling of a specific case study. Their formulation, referred to here as COM, was derived from numerical and observed results and incorporates several Mohr-Coulomb strength parameters. The equation is expressed as follows:

$$S_{v,\max} = AS \quad \text{where,} \quad (2.3)$$

$$A = 1.8825 \frac{D}{Z_0} \quad \text{and,}$$

$$S = 1699.2 \left(\left(\frac{\gamma_s Z_0 + \sigma_s - (c + 0.3\sigma_\tau)}{E_s} \right) (1 - \nu_s)(1 - \sin \varphi) \right)^{0.8361}$$

Where γ_s is the soil unit weight (kN/m³), Z_0 is the tunnel depth (m), σ_s is the surface surcharge (kPa), c is the cohesion (as a Mohr-Coulomb parameter) (kPa), σ_τ is the required face support pressure (kPa), E_s is Young's modulus of the soil (kPa), ν_s is Poisson's ratio of the soil, and φ is the angle of internal friction of the soil (°). The unit of S is (mm).

It is noted that this model is included to illustrate how more detailed empirical or numerical formulations can be integrated into the proposed framework. Although Equation 2.3 was developed for a specific site using Mohr-Coulomb parameters, the overall approach adopted in this study is not restricted to this or any other particular model. Instead, users may adopt alternative settlement estimation methods (e.g., based on VL, field measurements, or analytical expressions) depending on the project requirements, available soil data, and design stage.

While parts of the study consider short-term undrained conditions (e.g., in early design stages), the methodology presented is not confined to undrained behaviour alone. The flexibility of the framework enables application across a range of soil types and loading conditions, depending on the level of model complexity and data availability.

To capture the longitudinal variation in settlement, Hajjar et al. (2015) proposed the following expression, which complements the transverse and vertical components presented above:

$$\frac{S_v(y)}{S_{v,\max}} = \begin{cases} \exp\left(-\frac{(y - y_{\text{face}} + i_y)^2}{(2i_y^2)}\right) & \text{for } y > y_{\text{face}} - i_y \\ 1 & \text{for } y \leq y_{\text{face}} - i_y \end{cases} \quad (2.4)$$

Here, y is the distance from the tunnel face along the longitudinal axis (m), with $y = 0$ at the face and positive values extending behind it in the direction of tunnel advance. y_{face} represents the distance from the tunnel entrance to the face (m), and i_y is a width param-

eter characterising the distance between the inflection point and the point of maximum settlement (m).

By combining the transverse and longitudinal settlement profiles, the ground settlement induced by tunnelling is obtained using Equations 2.1, 2.3 and 2.4. This yields the following expression for the 3D settlement profile.

$$S_v(x, y, z) = \frac{S_v(y)}{S_{v,\max} \exp\left(-\frac{x^2}{2i_x^2}\right)} \begin{cases} \exp\left(-\frac{(y - y_{\text{face}} + i_y)^2}{(2i_y^2)}\right) & \text{for } y > y_{\text{face}} - i_y \\ 1 & \text{for } y \leq y_{\text{face}} - i_y \end{cases} \quad (2.5)$$

Therefore, $S_{v,\max}$ can be calculated either based on the VL parameter using Equation 2.2 (e.g., settlement method 1), or using the more comprehensive approach in Equation 2.3 (e.g., settlement method 2). Regardless of the chosen method, the resulting 3D settlement profile is shown in Figure 2.2.

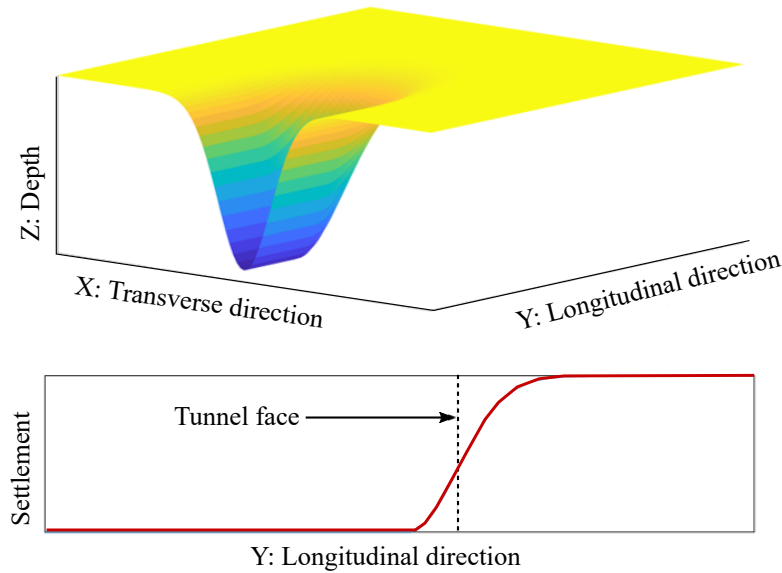


Figure 2.2: 3D settlement profile induced by tunnelling.

To enable broader application of the developed tool within BIM, and in addition to methods 1 and 2, other solutions such as those described in Franza and Marshall (2019); Camós

et al. (2016); Pinto and Whittle (2014) can also be implemented and made available to users depending on the design requirements.

2.2 Building Damage

As described above, tunnelling excavations in soft ground result in ground movements that can significantly affect buildings and infrastructure in urban environments. Hence, accurately predicting both settlements and potential damage to structures is vital. These considerations are essential during the planning, design, and construction phases of urban tunnelling projects (Mair et al. (1996)).

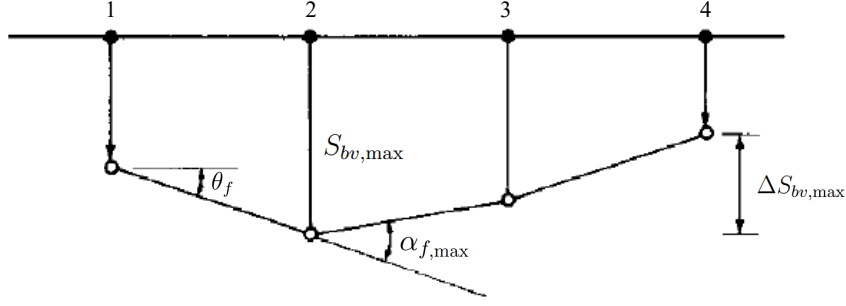
This section presents a summary of design approaches commonly used for the prediction of building damage due to tunnelling, covering definitions, risk evaluation, and various damage categorisations, including the concept of critical strain ($\varepsilon_{\text{crit}}$).

2.2.1 Definitions of Ground and Foundation Movements

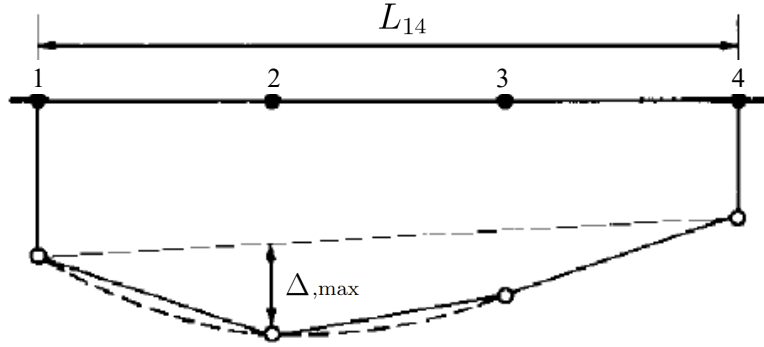
A review of the literature by Burland and Wroth (1974) uncovered a variety of inconsistent symbols and terminology used to quantify building deformation. To address this, they proposed a consistent set of definitions based on the displacements, either measured or calculated, of discrete points on a building's foundation. The variables used to quantify building deformation, illustrated in Figure 2.3, are described below:

1. **Horizontal strain** (ε_h): Due to a change in building length ΔL (m), a positive change in distance over the building length L (m) results in a positive average ε_h , indicating tension, while a negative change indicates compression. Where:

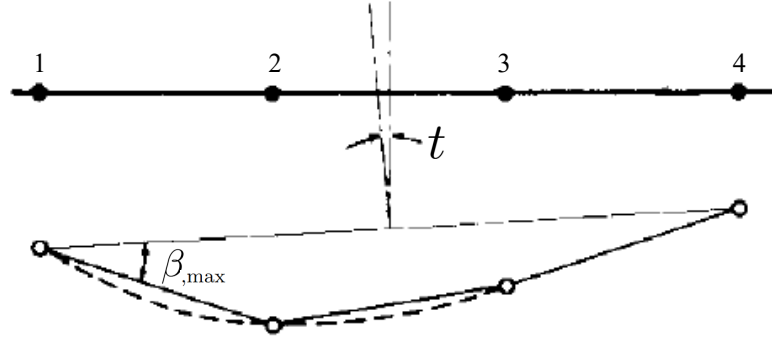
$$\varepsilon_h = \frac{\Delta L}{L} \quad (2.6)$$



(a) Definition of settlement, ($S_{bv,max}$), relative settlement, (ΔS_{bv}), rotation (θ_f) and angular strain (α_f).



(b) Definition of relative deflection (Δ) and deflection ratio Deflection Ratio (DR)



(c) Definition of tilt (t) and relative rotation (angular distortion) (β).

Figure 2.3: Definition of foundation movement (inspired by the works of Burland et al. (1977)).

2. **Settlement** ($S_{bv,max}$) (m or mm): This term implies a vertical displacement of the building foundation.
3. **Relative or differential settlement** (ΔS_{bv}): The relative vertical displacement between two points along the building foundation. The maximum relative settlement is denoted as $\Delta S_{bv,max}$. Where:

$$\Delta S_{bv} = S_{bv} @ \text{ node 1} - S_{bv} @ \text{ node 2} \quad (2.7)$$

Dividing this by the distance between the nodes, L_{12} (m), yields the value of the slope s .

4. **Rotation** (θ_f): Describes the change in angle of a line segment between two adjacent points at the foundation or ground level.
5. **Angular strain** (α_f): The angular strain is measured between two consecutive building segments. Where:

$$\alpha_f @ 2 = \frac{\Delta S_{bv, 12}}{L_{12}} + \frac{\Delta S_{bv, 23}}{L_{23}} \quad (2.8)$$

where 1, 2, and 3 are three consecutive points along the building foundation.

6. **Relative deflection** (Δ): The displacement of a point relative to the line connecting two reference points on the foundation, with the sign convention similar to that of α_f .
7. **DR**: This is the amount of relative deflection Δ , divided by the segment's length L , given by:

$$DR = \frac{\Delta}{L} \quad (2.9)$$

The sign convention follows that of the Δ and α_f definitions.

8. **Tilt** (t): Describes the rigid body rotation of a structure's end, or a well-defined part of it, where:

$$t = \frac{\Delta S_{bh}}{H} \quad (2.10)$$

where ΔS_{bh} is the relative horizontal displacement between two horizontally aligned points along the height of the building, and H is the building height (m).

9. **Angular distortion (Relative rotation) (β):** The rotation between two reference points relative to tilt t , given by:

$$\beta = s - t \quad (2.11)$$

It is important to note that the definitions provided above apply solely to in-plane deformations and do not account for behaviour in 3D space. While the approach proposed by Burland and Wroth (1974) is widely adopted and recommended, Rankin (1988) suggests that a larger number of observation points and detailed information on the foundations and superstructure are required. However, in practice, such information is rarely available, and interpretation can be challenging.

2.2.2 Evaluation of Risk of Damage

The assessment of the risk of damage to buildings typically follows a three-stage procedure, as employed in the construction of major projects such as the Jubilee Line Extension, the Channel Tunnel Rail Link, and the Crossrail Project (Mair et al. (1996)).

Stage one: Preliminary assessment: At this preliminary phase, it is practical to assume that the foundations of the buildings conform to the shape of the settlement trough. The evaluation of the critical deformation parameters is based on the slope of the surface trough and the magnitude of $S_{v,\max}$.

Stage two: second-stage assessment: In the second stage, a simple beam representing the building is placed on a GF site, and the portion of settlement occurring beneath the building is used to measure the maximum tensile strain value ($\varepsilon_{h,\max}$), thereby obtaining the corresponding potential damage category (Mair et al. (1996)). In practice, however,

due to SSI effects, the actual damage is often less severe than the category initially estimated.

Stage three: Detailed evaluation: For buildings classified in the second-stage assessment as being at high risk of damage, a detailed evaluation must be carried out. This includes the numerical modelling of buildings using FEM, incorporating geometrical details and non-linear material properties. More accurate predictions of tunnelling-induced settlements that account for SSI effects Mair et al. (1996) mark the point where the investigation mainly focuses. This involves examining buildings subjected to excessive movements due to tunnelling, which may potentially lead to collapse. At this stage, predicting structural behaviour using advanced numerical modelling and ML becomes inevitable.

2.2.3 Categories of Damage

Numerous studies have investigated building damage using various observation points. Burland et al. (1977) described the level of building damage based on approximate crack width, as outlined in Table 2.1.

Burland (1997) highlights the importance of the dividing line between categories 2 and 3. Previous studies of several case histories show that damage of category 2 and below can result from a variety of causes which are not always related to ground movements (e.g., shrinkage or thermal effects). The identification of such causes is usually very difficult and frequently it results from a combination of things. The causes are typically easier to identify if damage exceeds Category 2 and it is often associated with ground movements, implying the importance of division between these categories.

In addition, Rankin (1988) proposed criteria for the assessment of building damage using slope s and settlement $S_{bv,max}$, as shown in Table 2.2. They recommend that structures falling under categories 3 and 4, and in some cases, even category 2, should undergo

Table 2.1: Classification of visible damage to walls with particular reference to ease of repair of plaster and brickwork or masonry (adjusted from (Burland et al. (1977))).

Degree of damage	Description of typical damage ⁺ (Ease of repair is underlined)	Approximate crack width (mm)
1. Very slight	Hairline cracks of less than about 0.1 mm are classed as negligible. <u>Fine cracks which can easily be treated during normal decoration.</u> Perhaps isolated slight fracture in the building. Cracks in external brickwork are visible on close inspection.	$< 1^*$
2. Slight	Cracks easily filled. <u>Re-decoration probably required.</u> Several slight fractures showing inside of the building. Cracks are visible externally and <u>some re-pointing may be required</u> externally to ensure weather tightness. Doors and windows may stick slightly.	$< 5^*$
3. Moderate	The cracks require <u>some opening up and can be patched by a mason.</u> Current cracks can be <u>masked by suitable linings.</u> <u>Repainting of external brickwork and possibly a small amount of brickwork to be replaced.</u> Doors and windows sticking. Service pipes may fracture. Weather tightness is often impaired.	5 to 15* or a number of cracks ≥ 3
4. Severe	<u>Extensive repair work involving breaking-out and replacing sections of walls, especially over doors and windows.</u> Windows and door frames are distorted, the floor sloping noticeably, walls leaning or bulging noticeably, and some loss of bearing in beams. Service pipes disrupted.	15 to 25* but also depends on number of cracks
5. Very severe	<u>This requires a major repair job involving partial or complete re-building.</u> Beams lose bearing, walls lean badly and require shoring. Windows were broken with distortion. The danger of instability.	usually $> 25^*$ but depends on number of cracks

⁺ In assessing the degree of damage account must be taken of its location in the building or structure.

^{*} Crack width is only one aspect of damage and should not be used on its own as a direct measure of it.

a detailed evaluation. This emphasises the importance of considering multiple factors, including soil and groundwater conditions, the structural elements of the building, and any pre-existing deformations, in order to determine the likelihood of further movements and the structure's tolerance to them.

Table 2.2: Typical values of maximum building slope and settlement for damage risk assessment (adjusted from (Rankin (1988))).

Risk Category	Maximum Slope of Building	Maximum Settlement of Building (mm)	Description of Risk
1.	Less than 1/500	Less than 10	Negligible: superficial damage unlikely.
2.	1/500 to 1/200	10 to 50	Slight: Possible superficial damage which is unlikely to have structural significance.
3.	1/200 to 1/50	50 to 75	Moderate: Expected superficial damage and possible structural damage to buildings, possible damage to relatively rigid pipelines
4.	Greater than 1/50	Greater than 75	High: Expected structural damage to buildings. Expected damage to rigid pipelines, possible damage to other pipelines.

Furthermore, Skinner and Charles (2004), using tilt (t), suggested criteria for acceptable values of t in low-rise dwellings, developed through case histories. These criteria apply to the overall tilt of a building and are based on the performance of two-storey structures. This, in turn, may be more conservative for single-storey buildings and less so for three-storey buildings. The proposed indicative values for acceptable tilt are presented in Table 2.3.

2.2.4 Concept of Limiting Tensile Strain Method

Through a series of extensive tests on masonry panels, Burland and Wroth (1974) established that visible cracks are associated with a well-defined average value of tensile strain. This has been designated as the critical tensile strain ($\varepsilon_{\text{crit}}$), measured over a gauge length of at least one metre. Extending this framework, Boscardin and Cording (1989) developed the concept of differing levels of tensile strain by analysing seventeen case records of structural damage caused by excavation-induced subsidence across a variety of building types. They demonstrated that the categories of damage presented in

Table 2.3: Indicative values for t of low-rise housing (adjusted from (Skinner and Charles (2004))).

Classification	Tilt (t)	Comment
Design limit value	1/400	The maximum acceptable differential settlement across the building is related to the design limit value for tilt. If the building is likely to tilt more than this limit value, ground treatment or deep foundations may be required.
Noticeability	1/250	The point at which the tilt of a building becomes noticeable will depend on the type and purpose of the building, and the powers of observation and perception of the occupiers. The tilt of low-rise housing typically is noticed when it is in the region of 1/250 to 1/200.
Monitoring	1/250	When tilt is noticed it is advisable to make some measurements to confirm that the building has tilted. If the measured tilt is greater than 1/250, monitoring should be carried out to determine whether the tilt is increasing.
Remedial action	1/100	Where tilts of this magnitude are measured, or the measured rate of increase of tilt indicates that this degree of tilt will be exceeded, some remedial action should be taken. This is likely to include re-leveling the building, perhaps by grouting or underpinning and jacking.
Ultimate limit	1/50	If tilt reaches this level, the building may be regarded as in a dangerous condition, and remedial action either to re-level or demolish the building will be required urgently.

Table 2.1 could be broadly related to ranges of the limiting tensile strain (ε_{lim}). These categorisations were further detailed in Table 2.4, underlining the table’s significance in linking expected building deformations to the severity of resulting damage.

2.2.4.1 Evaluating Strain in a Simple Beam

Timoshenko (1957) describes an expression for the total mid-span deflection Δ of a centrally loaded beam, accounting for both bending and shear stiffness, as follows:

Table 2.4: Relationship between category of damage and limiting tensile strain (ε_{lim}) (after (Boscardin and Cording (1989))).

Category of damage	Normal degree of severity	Limiting tensile strain ε_{lim} (%)
0	Negligible	0 - 0.05
1	Very slight	0.05 - 0.075
2	Slight	0.075 - 0.15
3	Moderate*	0.15 - 0.3
4 to 5	Severe to very severe	> 0.3

* Note: Boscardin and Cording (1989) describe the damage corresponding to ε_{lim} in the range 0.15 - 0.3% as 'moderate to severe'. However, none of the cases quoted by them exhibit severe damage for this range of strains. There is therefore no evidence to suggest that tensile strains up to 0.3% will result in severe damage.

$$\Delta = \frac{P}{48} \cdot \frac{L^3}{EI} \left[1 + \frac{18}{L^2} \cdot \frac{I}{H} \cdot \frac{E}{G} \right] \quad (2.12)$$

where P is the centrally applied point load at mid-span (N), E is Young's modulus (MPa), G is the shear modulus (MPa), I is the second moment of area of the building material and geometry (m^4), and H is the beam height or depth (m).

Equation 2.12 can be rewritten in terms of the DR and the maximum bending strain $\varepsilon_{b,\text{max}}$, as:

$$\frac{\Delta}{L} = \varepsilon_{b,\text{max}} \cdot \frac{L}{12h'} \left[1 + \frac{18}{L^2} \cdot \frac{I}{H} \cdot \frac{E}{G} \right] \quad (2.13)$$

where $\varepsilon_{b,\text{max}}$ is the maximum strain due to bending and h' is the distance from the neutral axis to the edge of the beam in tension (m). In this context, $h' = H$ for hogging and $h' = 0.5 \times H$ for sagging.

Similarly, for shear strain $\varepsilon_{d,\text{max}}$, Equation 2.12 becomes:

$$\frac{\Delta}{L} = \varepsilon_{d,\text{max}} \left[1 + \frac{L^2}{18} \cdot \frac{H}{I} \cdot \frac{G}{E} \right] \quad (2.14)$$

where $\varepsilon_{d,\max}$ is the maximum strain due to shear.

By substituting $\varepsilon_{b/d,\max}$ with ε_{\lim} in Equations 2.13 and 2.14, the limiting values of the deflection ratio (DR) for cracking in simple beams under bending and shear are defined. It becomes apparent that for any given value of ε_{\lim} , the limiting DR is a function of the beam's aspect ratio (L/H), the ratio of bending to shear stiffness (E/G), and the position of the neutral axis, which influences the second moment of area I .

2.2.4.2 Influence of Horizontal Strain

Boscardin and Cording (1989) incorporated the horizontal tensile strain into the analysis through simple superposition, assuming that the deflected beam is subject to uniform extension over its depth. Therefore, the resultant extreme fibre flexural strain ($\varepsilon_{b,r}$) is given by:

$$\varepsilon_{b,r} = \varepsilon_{b,\max} + \varepsilon_{h,\max} \quad (2.15)$$

The resultant shear strain can be evaluated using Mohr's circle of strain, given by:

$$\varepsilon_{d,r} = \varepsilon_{h,\max} \left(\frac{1 - \nu}{2} \right) + \sqrt{\varepsilon_{h,\max}^2 \left(\frac{1 + \nu}{2} \right)^2 + \varepsilon_{d,\max}^2} \quad (2.16)$$

where ν is the Poisson's ratio of the building material. Thus, the maximum tensile strain value is the larger of the two values obtained from Equations 2.15 and 2.16. By knowing the beam's dimensions (L and H), and for a given value of DR, and $\varepsilon_{h,\max}$, it is possible to evaluate the value of maximum tensile strain ($\varepsilon_{b,r}$ or $\varepsilon_{d,r}$) in terms of h' , E/G , and ν . The resulting value of ($\varepsilon_{b,r}$ or $\varepsilon_{d,r}$) can then be used with Table 2.4 to assess the potential damage to buildings due to tunnelling.

In addition to the previously discussed tables, Boscardin and Cording (1989) applied

deep beam theory to develop a series of curves correlating $\varepsilon_{h,\max}$, β , and cracking damage levels for masonry structures. Similarly, Burland (1997) presented comparable curves relating DR and $\varepsilon_{h,\max}$, drawn from Table 2.4, to predicted damage levels. Their research included case studies from tunnel construction, shallow mines, and braced excavations. They concluded that most observed damage levels fell within the predicted ranges set by these curves.

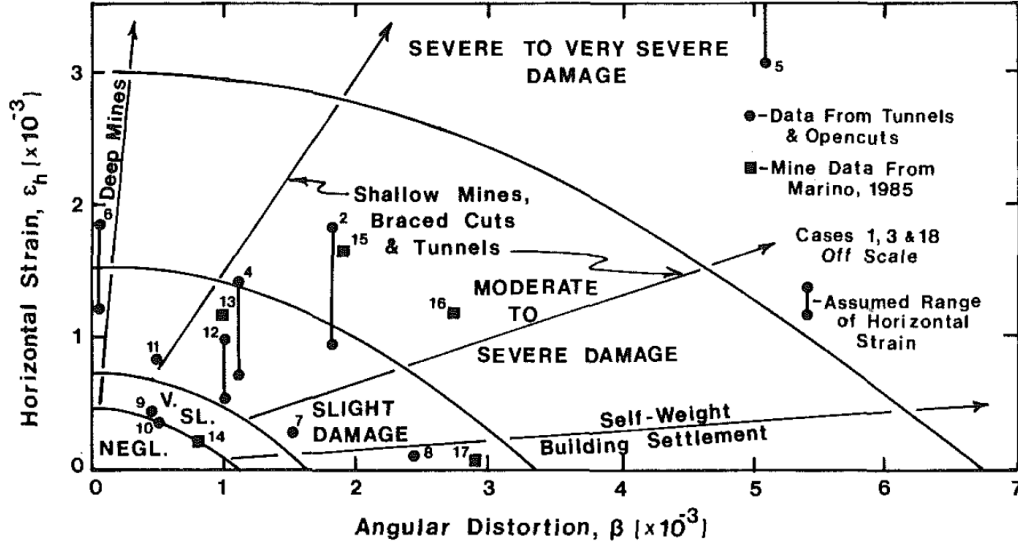


Figure 2.4: Damage curves derived from case studies categorising damage based on β and $\varepsilon_{h,\max}$ according to (Boscardin and Cording (1989)).

All presented damage categorisation frameworks are fundamental to the evaluation of building damage. They offer diverse criteria to contextualise the level of induced damage and enable comparisons between various approaches, incorporating both local and global aspects of structural response.

2.3 Soil-Structure Interaction

As previously indicated, assessing building damage often involves the simplified assumption that buildings are infinitely flexible i.e., they offer no resistance to soil deformation. This assumption is commonly applied in preliminary and second-stage damage assess-

ments, where the building is presumed to conform entirely to the ground surface deformation (GF site). As a result, deformation parameters such as DR and ($\varepsilon_{b,r}$ or $\varepsilon_{d,r}$) tend to be overestimated, often categorising the predicted damage as more severe than in reality.

The role of building stiffness in mitigating induced settlement is briefly addressed. For instance, in the case of the Manson House in London, Frischmann et al. (1994) demonstrated that building stiffness can reduce differential settlement (ΔS_{bv}) beneath the structure due to the resistance provided by the superstructure. This highlights the importance of considering SSI in damage assessments.

Numerous studies have investigated the SSI phenomenon using a range of analytical, numerical, and experimental approaches. One of the earliest is the Winkler (1867) model, which treats the soil as a set of discrete, independent springs. While simple, it fails to account for soil continuity. This limitation was addressed by the Vlasov and Leont'ev (1966) model, which introduced a more representative two-parameter elastic foundation that captures both normal and shear subgrade stiffness. This model assumes the building behaves as an elastic beam resting on a semi-continuous medium.

Building on this work, several other models were developed to conduct parametric studies or incorporate additional variables, such as different loading types or material property variation with depth (Teodoru and Muşat (2008); Vallabhan and Das (1991)). Acikgoz et al. (2021) proposed an equivalent cracked beam model to overcome limitations of elastic simplification. However, this method requires prior knowledge of crack location and depth, which makes it unsuitable for use in scenarios where pre-tunnelling conditions are unknown.

Subsequent studies expanded on this foundation modelling concept to include more realistic scenarios. For instance, Teodoru and Muşat (2008) and Vallabhan and Das (1991) incorporated depth-dependent properties and different loading types. However, these analytical approaches often require simplifications that limit their applicability. To over-

come this, Acikgoz et al. (2021) proposed a cracked-beam model that better simulates the in-service condition of buildings. Still, it depends on prior knowledge of crack location and depth-information rarely available in pre-tunnelling scenarios.

With the advent of modern computational tools, finite element and finite difference methods have become increasingly common for SSI studies (Yiu et al. (2017); Jenck and Dias (2004); Cheng et al. (2007); Teodoru (2009b)). A landmark contribution was made by Potts and Addenbrooke (1997), who developed design curves based on extensive 2D FEM parametric analyses of beams with varying stiffness relative to soil. These curves enable practitioners to account for structural stiffness when predicting deformation troughs, significantly improving the reliability of damage assessments.

This methodology was further refined by Franzius et al. (2004), who included additional parameters such as building self-weight, interface behaviour, and foundation depth. Their results remained consistent with the original design curves, reinforcing their practical relevance.

In parallel, experimental studies using centrifuge modelling Ritter et al. (2017); Loganathan et al. (2000); Franza et al. (2020) have been essential in validating these numerical methods. By including complex foundation geometries and pile-soil interactions, these experiments help bridge the gap between theoretical predictions and real-world behaviour.

2.3.1 Modified Vlasov Method

Building on the analytical work of Vallabhan and Das (1991), Teodoru (2009b) developed a custom FEM tool named Euler Bernoulli Beam on Two Parameter Elastic Foundation (EBBEF2p), based on a two-parameter elastic foundation model. In this method, the beam is discretised into multiple line elements of length (l), with two nodes per element. The elements are connected only at their nodes, and all forces are applied at these discrete

points.

Prior to structural analysis, the stiffness of the soil must be established. Using variational calculus, Vlasov's parameters are defined as:

$$k_{\text{norm}} = \int_0^{H_s} \frac{E_s(1 - \nu_s)}{(1 + \nu_s)(1 - 2\nu_s)} \left(\frac{d\phi}{dz} \right)^2 dz, \quad (2.17)$$

$$k_{\text{shear}} = \int_0^{H_s} \frac{E_s}{2(1 + \nu_s)} \phi^2 dz \quad (2.18)$$

where k_{norm} and k_{shear} (N/m³) are the normal and shear subgrade stiffnesses, H_s is the influence zone depth in the soil medium (m), and $\phi(z)$ describes vertical deflection variation with depth:

$$\phi(z) = \frac{\sinh \left[\gamma \left(1 - \frac{z}{H_s} \right) \right]}{\sinh \gamma} \quad (2.19)$$

γ is a dimensionless shape parameter, determined iteratively using:

$$\left(\frac{\gamma}{H_s} \right)^2 = \frac{1 - 2\nu_s}{2(1 - \nu_s)} \cdot \frac{\int_{-\infty}^{+\infty} \left(\frac{dS_{bv}}{dx} \right)^2 dx}{\int_{-\infty}^{+\infty} S_{bv}^2 dx} \quad (2.20)$$

The computed values of k_{norm} and k_{shear} are then used in stiffness matrices:

$$[k_e] = \frac{EI}{l^3} \begin{bmatrix} 12 & 6l & -12 & 6l \\ 6l & 4l^2 & -6l & 2l^2 \\ -12 & -6l & 12 & -6l \\ 6l & 2l^2 & -6l & 4l^2 \end{bmatrix} \quad (2.21)$$

$$[k_f] = \frac{l k_{\text{norm}}}{420} \begin{bmatrix} 156 & 22l & 54 & -13l \\ 22l & 4l^2 & 13l & -3l^2 \\ 54 & 13l & 156 & -22l \\ -13l & -3l^2 & -22l & 4l^2 \end{bmatrix} \quad (2.22)$$

$$[k_t] = \frac{k_{\text{shear}}}{30l} \begin{bmatrix} 36 & 3l & -36 & 3l \\ 3l & 4l^2 & -3l & -l^2 \\ -36 & -3l & 36 & -3l \\ 3l & -l^2 & -3l & 4l^2 \end{bmatrix} \quad (2.23)$$

The global system is then assembled:

$$([k_e] + [k_f] + [k_t])\{d_e\} = \{S_e\} - \{R_e\} \quad (2.24)$$

where $[k_e]$ is the stiffness matrix of the flexural beam element (N/m), $[k_f]$ is the stiffness matrix of the elastic foundation (first subgrade) (N/m), and $[k_t]$ is the stiffness matrix of the rigid base (second subgrade) (N/m), $\{d_e\}^T = \{d_{v_1}, d_{\theta_1}, d_{v_2}, d_{\theta_2}\}$ (m, rad) are the nodal DOFs. $\{S_e\}^T$ and $\{R_e\}^T$ (N and N·m) are the nodal loads and distributed load vectors, respectively:

$$\{R_e\}^T = \left\{ \frac{ql}{2}, \frac{ql^2}{12}, \frac{ql}{2}, -\frac{ql^2}{12} \right\} \quad (2.25)$$

The formulation for EBBEF2p presented above can be directly implemented in BIM as a plug-in (Dynamo node) using Python. It is used to compute the settlement profile resulting from SSI. In the current implementation, the SSI effect is considered for the response of individual buildings to tunnelling-induced settlements, without accounting for mutual effects between nearby buildings, which are assumed negligible in the case of detached structures.

This model has been implemented as a Python-based Dynamo plug-in within BIM, allowing automated estimation of SSI settlement profiles. However, it currently assumes independent building responses and does not account for group interaction effects, which are assumed negligible for detached structures.

The limitations of this implementation, including the inherent assumptions of this method and computational scalability, are further discussed in Section 4.

2.3.2 The Relative Stiffness Approach

A widely referenced method for estimating the influence of structural stiffness on tunnelling-induced settlement is the set of design curves developed by (Potts and Addenbrooke (1997)). These were derived from a series of two-dimensional FEM parametric studies investigating the response of elastic beams on non-linear elastic-plastic soil, calibrated to resemble London Clay. The studies varied key parameters such as tunnel depth, beam width, eccentricity (beam position relative to the tunnel centreline), and stiffness ratios between the structure and the ground.

The resulting design curves quantify the reduction in settlement due to building stiffness by comparing it to the settlement profile of a ground-free (GF) site. This enables a simplified adjustment of the predicted deformation and associated damage levels, mitigating the overestimation commonly associated with the flexible-structure assumption. As such, they are particularly useful in early-stage assessments and have been widely cited in practice-oriented research.

However, Potts and Addenbrooke (1997) also cautioned that the curves should be applied judiciously, as they are based on a specific range of conditions. Their use outside the bounds of the original parametric studies, such as for different soil types, complex geometries, or foundation systems, may lead to unreliable predictions. In such cases, especially where buildings are structurally sensitive or higher damage categories are an-

anticipated, more rigorous, structure-specific analyses using numerical modelling techniques are recommended.

The relative stiffness approach is reviewed here for background context. It is not adopted in the present study, which instead applies a modified Vlasov-based finite element method for SSI analysis (see Section 2.3.1).

2.4 Numerical Modelling of Masonry Building

Masonry structures are inherently well-suited to resist gravitational loads due to their compressive strength. However, they remain vulnerable under extreme loading conditions such as seismic excitation or differential settlements, particularly when compared to more ductile structural systems like steel or Reinforced Concrete (RC) frames. The modelling of masonry structures presents challenges due to the material's composite and brittle nature. To address this, several strategies have been developed, each operating at different levels of scale and fidelity. D'Altri et al. (2020) categorised these strategies into four groups, in line with the earlier classification proposed by Lourenço et al. (1995) for the first two. These include:

- **Block-based Models (Micro-modelling):** This approach explicitly models bricks, mortar, and joints, capturing local interactions and heterogeneity. While it allows detailed simulation of crack initiation and propagation, its high computational cost limits its practical use to small-scale panels or substructures. Studies such as Roca et al. (2010) employed this approach to replicate experimentally observed crack paths under monotonic loading, showing good agreement but highlighting scale limitations.
- **Continuum Models (Macro-modelling):** Here, masonry is idealised as a homogeneous material with averaged properties derived from tests or homogenisation.

The simplification allows for coarser meshing and efficient computation, making it suitable for global behaviour assessment. However, as shown by Lourenco (1996), these models often struggle to accurately capture localised cracking patterns without additional refinements or embedded discontinuity models.

- **Geometry-based and Macro-Element Models:** These models treat walls or panels as structural macroelements with simplified mechanics, relying heavily on geometry and empirical calibration. In macro-element modelling, the building is reduced to a set of rigid blocks or macro-panels, each governed by phenomenological behaviour. This trade-off in accuracy allows for rapid assessment of seismic vulnerability, as seen in historical building assessments (Lagomarsino et al. (2013)).

Advanced modelling of masonry structures often requires the consideration of material and geometric non-linearities. As outlined by Bathe et al. (1975), these analyses are performed using incremental-iterative procedures, which are classified as either non-linear static or non-linear dynamic:

- **Non-linear Static Analysis:** In this load-controlled, time-independent analysis, displacements or forces are applied incrementally. Common solution algorithms include Newton-Raphson, Quasi-Newton, Arc-Length, and Riks methods. The governing equation is:

$$[K(u)] \cdot \{u\} = \{F_{\text{ext}}\} - \{F_{\text{int}}(u)\} \quad (2.26)$$

where $[K(u)]$ is the displacement-dependent stiffness matrix (N/m), $\{u\}$ is the nodal displacement vector (m), $\{F_{\text{ext}}\}$ is the external load vector (N), and $\{F_{\text{int}}(u)\}$ is the internal force vector (N). This method has been widely used in pushover analyses of masonry walls (Giardina (2013); Lourenco (1996)).

- **Non-linear Dynamic (Time History) Analysis:** This approach captures inertia and damping effects during transient loading events. It is particularly useful when convergence in static procedures fails due to high non-linearity. The general

form of the dynamic equilibrium equation is:

$$[M(u, \dot{u})] \cdot \{\ddot{u}\} + [C(u, \dot{u})] \cdot \{\dot{u}\} + [K(u)] \cdot \{u\} = \{F(t, u, \dot{u})\} \quad (2.27)$$

where $[M(u, \dot{u})]$ is the mass matrix (kg), $[C(u, \dot{u})]$ is the damping matrix (N·s/m), $\{\ddot{u}\}$ is the acceleration vector (m/s²), $\{\dot{u}\}$ the velocity vector (m/s), and $\{F(t, u, \dot{u})\}$ the external force vector (N).

Dassault Systèmes Simulia Corp (2021) and others have demonstrated the effectiveness of this method in overcoming convergence issues in simulations of out-of-plane wall collapse.

2.4.1 Smeared Cracking Concepts

In continuum-based models, cracking in masonry is often represented using the smeared cracking approach introduced by (Rashid (1968)). This formulation distributes cracking across finite elements through modified stress-strain relationships, allowing for crack development without remeshing. Unlike discrete cracking models, the smeared approach retains mesh topology and avoids bias in crack orientation.

Smeared crack models are broadly categorised into fixed and rotating crack models. In fixed crack models, the crack orientation is determined at initiation and remains constant throughout the analysis. However, this can lead to stress-locking or unrealistic stress rotations. In contrast, rotating crack models allow the crack orientation to evolve with the principal strain direction, offering improved numerical stability and more accurate simulation of multi-directional cracking. This evolution was pivotal in advancing the use of continuum models in masonry simulation, as highlighted by Rots and Blaauwendraad (1989), who demonstrated the superior performance of rotating crack models in capturing diagonal cracking in shear-dominated masonry walls.

Despite their popularity, smeared cracking models require careful calibration of post-peak

softening behaviour to avoid mesh dependency and convergence issues. These limitations continue to be a topic of ongoing research, especially in modelling cyclic or dynamic responses.

2.4.2 The Concrete Damaged Plasticity Model

The non-linear material behaviour of the masonry façade in this thesis is modelled using the Concrete Damaged Plasticity (CDP) model available in ABAQUS. This choice aligns with the adopted Macro-modelling approach, in which the composite behaviour of masonry is represented as a homogeneous continuum and damage is smeared throughout the material rather than explicitly modelled at joints. Alternative constitutive models for masonry include the Micro-modelling of bricks and mortar joints as distinct materials Lourenco (1996), the discrete element method Pulatsu et al. (2016), and damage-plasticity models tailored specifically for masonry-like materials (Addessi et al. (2002)). While these approaches offer increased accuracy in capturing localised failure modes, they require extensive calibration and computational cost, which can be prohibitive in large-scale façade simulations. The CDP model, originally formulated for concrete but now widely adopted for quasi-brittle materials including masonry Genikomsou and Polak (2017); Saloustros et al. (2014), provides a robust and computationally efficient framework that meets the requirements of this study. Its availability as a built-in model in ABAQUS further facilitates its implementation and integration within the broader FEM workflow developed in this thesis. Unlike smeared cracking concepts, the CDP model in ABAQUS is a continuum, plasticity-based damage model tailored for concrete-like materials. It assumes that tensile cracking and compressive crushing are the primary failure mechanisms.

Under uniaxial tension, the stress–strain response is linear up to the peak tensile stress, σ_t , beyond which Micro-cracking initiates. This is followed by strain softening and localised deformation. Similarly, under uniaxial compression, the response is initially linear

until the yield stress σ_c , after which strain hardening continues up to the ultimate compressive stress σ_{cu} . This is followed by a softening response that reflects the progressive degradation of the material. While idealised, this behaviour aligns with experimental findings (Kaushik et al. (2007); Pluijm (1992)).

When unloading occurs from the softening branch of the stress-strain curve, the material exhibits a reduction in stiffness compared to its initial elastic modulus E_{init} . This degradation is quantified using two scalar damage variables, d_t for tension and d_c for compression. These are functions of plastic strain ($\varepsilon_t^{pl}, \varepsilon_c^{pl}$), temperature Θ , and other field variables f_i :

$$d_t = d_t(\varepsilon_t^{pl}, \Theta, f_i); \quad 0 \leq d_t \leq 1, \quad (2.28) \quad d_c = d_c(\varepsilon_c^{pl}, \Theta, f_i); \quad 0 \leq d_c \leq 1. \quad (2.29)$$

The damage variables range from 0 (undamaged material) to 1 (complete failure). The stress-strain relationships for the damaged material under uniaxial loading conditions are defined as:

$$\sigma_t = (1 - d_t)E_{\text{init}} (\varepsilon_t - \varepsilon_t^{pl}) \quad (2.30) \quad \sigma_c = (1 - d_c)E_{\text{init}} (\varepsilon_c - \varepsilon_c^{pl}) \quad (2.31)$$

Post-failure behaviour is defined using cracking and inelastic strain. The cracking strain is:

$$\varepsilon_t^{ck} = \varepsilon_t - \varepsilon_{0t}^{el}, \quad \text{where} \quad \varepsilon_{0t}^{el} = \frac{\sigma_t}{E_{\text{init}}} \quad (2.32)$$

The inelastic (crushing) strain in compression is:

$$\varepsilon_c^{in} = \varepsilon_c - \varepsilon_{0c}^{el}, \quad \text{where} \quad \varepsilon_{0c}^{el} = \frac{\sigma_c}{E_{\text{init}}} \quad (2.33)$$

These definitions allow the CDP model to represent degradation and post-peak material softening under complex loading conditions.

2.4.3 Plasticity Parameters of the Concrete Damaged Plasticity Model

In addition to the uniaxial stress–strain curves, the CDP model in ABAQUS requires the specification of at least five additional plasticity parameters. These include:

- The dilation angle, ψ_{CDP} , in the $(p-q)_{CDP}$ plane, which is model-dependent and can vary significantly (Genikomsou and Polak (2015)).
- The flow potential eccentricity, e_{CDP} , a small positive number that controls the asymptotic approach of the hyperbolic flow potential.
- The ratio of initial equibiaxial compressive yield stress to initial uniaxial compressive yield stress, σ_{0b}/σ_{0c} .
- The parameter k_{CDP} , defined as the ratio of the second stress invariant on the tensile meridian $q_{(TM)}$ to that on the compressive meridian $q_{(CM)}$ at initial yield, for a given pressure invariant p_{CDP} . This is defined under the condition that the maximum principal stress is negative ($\sigma_c < 0$), consistent with the convention that tension is positive and compression is negative in ABAQUS. The recommended range for k_{CDP} is $0.5 < k_{CDP} < 1$.
- The viscosity parameter, μ_{CDP} , used for viscoplastic regularisation of the concrete constitutive equations in ABAQUS Standard.

Additional parameters such as temperature and predefined field variables are optional and may be included where necessary.

ABAQUS recommends default values of $e_{CDP} = 0.1$, $\sigma_{0b}/\sigma_{0c} = 1.16$, $k_{CDP} = 2/3$, and $\mu_{CDP} = 0$. However, parametric studies are advised to determine whether alternative values provide a more accurate fit to experimental results for specific applications (Rainone et al. (2023); Behnam et al. (2018); Silva et al. (2021)).

2.4.4 Stress-Strain Relationship and the Characteristic Length,

h

In models with extensive unreinforced regions, such as wall-bearing structures, the tensile behaviour described earlier can lead to unreasonable mesh sensitivity. However, the widely accepted approach proposed by Hillerborg et al. (1976) introduces fracture energy $G_{f(c \text{ or } t)}$ as a material parameter, which significantly mitigates this issue. This concept defines the energy required to open a unit area of crack and is rooted in principles of brittle fracture mechanics.

Notably, Lourenco (1996); Feenstra (1993); Rots (1988) introduced a characteristic length parameter, h , to normalise the stress–strain relationship. This parameter, when used with $G_{f(c \text{ or } t)}$, enables the formulation of a mesh-objective post-failure response, leading to convergent results regardless of mesh refinement. The parameter h generally depends on the element type, shape, size, and integration scheme. According to Feenstra (1993), it can be computed using:

$$h = a_h \sqrt{A_e} = a_h \left(\sum_{\xi=1}^{n_\xi} \sum_{\eta=1}^{n_\eta} \det(\mathbf{J}) w_\xi w_\eta \right)^{\frac{1}{2}} \quad (2.34)$$

where w_ξ and w_η are the Gaussian integration weights, and ξ, η are isoparametric coordinates of integration points. The factor a_h equals 1 for quadratic elements and $\sqrt{2}$ for linear elements, as reported by (Rots (1988)). Estimates for h , obtained through trial-and-error fitting to discrete crack models, can be made with sufficient accuracy depending on the mesh type and integration order. These estimates are presented in Table 2.5.

Table 2.5: Recommendations for estimating the value of h .

Element Order	Element Type	Equivalent Length h
Linear	Tetrahedral	l_e
	Hexahedral	$l_e \sqrt{2}$
Quadratic	Tetrahedral	$0.5 l_e \sqrt{2}$
	Hexahedral	l_e

Where l_e represents the length along an element edge assuming that the elements have an aspect ratio of 1:1 (m or mm).

According to Lourenco (1996), incorporating h into the constitutive equation for material in compression results in the following formulation:

$$\sigma(\varepsilon_c^{in}) = \begin{cases} \sigma_i + (\sigma_c - \sigma_i) \cdot \sqrt{\frac{2\varepsilon_c^{in}}{\varepsilon_p} - \frac{(\varepsilon_c^{in})^2}{\varepsilon_p^2}}, & \text{for } \varepsilon_c^{in} \leq \varepsilon_p \\ \sigma_c + (\sigma_m - \sigma_c) \cdot \left(\frac{\varepsilon_c^{in} - \varepsilon_p}{\varepsilon_m - \varepsilon_p} \right)^2, & \text{for } \varepsilon_p < \varepsilon_c^{in} \leq \varepsilon_m \\ \sigma_r + (\sigma_m - \sigma_r) \cdot \exp\left(m \cdot \frac{\varepsilon_c^{in} - \varepsilon_m}{\sigma_m - \sigma_r}\right), & \text{for } \varepsilon_c^{in} > \varepsilon_m \end{cases} \quad (2.35)$$

with

$$m = 2 \left(\frac{\sigma_m - \sigma_c}{\varepsilon_m - \varepsilon_p} \right) \quad (2.36) \quad \varepsilon_m = \left(\frac{75 G_{fc}}{67 h \sigma_c} \right) + \varepsilon_p \quad (2.37)$$

Here, it is assumed that all stress values in the inelastic law are determined from the peak as: $\sigma_i = \frac{1}{3}\sigma_c$, $\sigma_m = \frac{1}{2}\sigma_c$, and $\sigma_r = \frac{1}{10}\sigma_c$. The equivalent inelastic strain, ε_p , corresponds to the peak compressive strength. If experimental data is unavailable, this value can be assumed as 2×10^{-3} (Lourenco (1996)).

To prevent snap-back behaviour at the constitutive level, the following condition must be satisfied; otherwise, the strength limit must be reduced:

$$\varepsilon_m \geq \frac{\sigma_m}{E_{init}} + \varepsilon_p \quad (2.38) \quad \sigma_m = \left(\frac{75 G_{fc} E_{init}}{67 h} \right)^{\frac{1}{2}} \quad (2.39)$$

The uniaxial tensile stress–inelastic strain relationship proposed by Feenstra (1993) is employed. Two types of softening laws are commonly used: linear and exponential. These are described as follows:

$$\sigma(\varepsilon_t^{ck}) = \begin{cases} \sigma_t \left(1 - \frac{\varepsilon_t^{ck}}{\varepsilon_{tu}} \right), & \text{for linear softening} \\ \sigma_t \exp \left(-\frac{\varepsilon_t^{ck}}{\varepsilon_{tu}} \right), & \text{for parabolic softening} \\ \text{where } \varepsilon_{tu} = k_{\text{soft}} \frac{G_{ft}}{h\sigma_t} & \end{cases} \quad (2.40)$$

Here, $k_{\text{soft}} = 2$ for linear softening and $k_{\text{soft}} = 1$ for parabolic softening, representing the internal damage parameter.

If the condition in Equation 2.41 is not satisfied, the strength must be reduced according to Equation 2.42:

$$h \leq k_{\text{soft}} \frac{G_{ft} E_{\text{init}}}{\sigma_t^2} \quad (2.41) \quad \sigma_t = \left(\frac{k_{\text{soft}} G_{ft} E_{\text{init}}}{h} \right)^{\frac{1}{2}} \quad (2.42)$$

ABAQUS also offers two alternative models to simulate the brittle behaviour of materials based on a stress–displacement rather than a stress–strain response. In this approach, the fracture energy G_{ft} can be defined directly as a material property. The cracking displacement u_t , which marks the complete loss of tensile strength, is given by:

$$u_t = \frac{2G_{ft}}{\sigma_t} \quad (2.43)$$

Alternatively, σ_t can be defined directly in tabular form as a function of the associated G_{ft} . The post-cracking response is assumed to follow a linear loss of strength.

2.4.5 Material Modelling and Sensitivity Analysis

To ensure accurate simulation of building damage, particularly in masonry façades, it is crucial to identify which modelling parameters most significantly influence the predicted structural response. While the preceding sections introduced material constitutive models, such as the CDP model and fracture energy-based formulations, these rely on

numerous input parameters, whose effects may vary with structural configuration and loading conditions. Therefore, a systematic SA is essential to quantify the influence of key parameters, including material properties, geometric characteristics, and boundary conditions. Such analysis enhances model robustness and supports the prioritisation of parameters for calibration and data collection. To achieve this, appropriate sampling strategies are employed to enable comprehensive and efficient exploration of the parameter space, with consideration for both computational cost and memory allocation.

2.5 Sensitivity Analysis and Sampling

SA is a fundamental tool for quantifying the influence of input parameters on model outputs, particularly in complex non-linear simulations such as those encountered in structural and damage modelling. In the context of masonry façades, which often involve material and geometric uncertainties, SA provides insight into which parameters most significantly affect structural performance predictions.

Various global SA methods have been established in the literature. The Morris method Morris (1991) is widely used for screening purposes due to its computational efficiency, offering a way to estimate both the mean and standard deviation of Elementary Effect (EE). More comprehensive, Variance-based (VB) methods, such as Sobol indices Saltelli et al. (2008), enable the decomposition of output variance into first-order, higher-order, and interaction effects, but typically require a large number of simulations. As noted by Saltelli et al. (2004), the trade-off between accuracy and computational cost must be considered carefully when selecting an appropriate SA method for large-scale simulations.

Recent studies have also explored the integration of ML techniques into the SA framework. RF regression, for instance, has been used as a surrogate model to approximate the input–output relationship and extract feature importance (Antoniadis et al. (2021); Baptiste et al. (2017); Menze et al. (2009)). This approach has the advantage of capturing

complex non-linear interactions while maintaining robustness against overfitting, making it especially suitable for high-dimensional input spaces. Feature importance can be evaluated via permutation techniques or through variance reduction metrics at decision tree splits.

2.5.1 Sampling Methods

The performance of any SA technique depends not only on the sensitivity index but also on the quality of the sampling. A well-distributed sample ensures that the input space is comprehensively explored, thus enabling more robust conclusions about parameter importance.

Several sampling techniques are reported in the literature. The One-At-A-Time (OAAT) method is the simplest, varying each input parameter individually while keeping others fixed. Although systematic, OAAT becomes inefficient for high-dimensional models and may miss interaction effects.

Latin Hypercube (LH) sampling, by contrast, is a stratified technique that maximises the coverage of the input space with relatively few samples. Each input parameter's range is divided into s_{strat} equally probable intervals, and samples are drawn such that each interval is represented once. This ensures that all portions of the parameter domain are sampled without clustering, making LH particularly effective in complex simulations where sample efficiency is important (Saltelli et al. (2008); Obel et al. (2020); Gamra et al. (2024)).

2.5.2 Application to Structural Modelling

Despite the increasing availability of SA tools, their application to structural and damage modelling, particularly in the context of masonry façades, remains limited. Most

studies either focus on analytical sensitivity indices or explore surrogate modelling independently (Zhao and DeJong (2023)). The combined application of LH sampling with RF-based SA has not yet been widely explored in this domain.

To address this gap, the present study adopts a hybrid strategy that combines LH sampling with RF regression to evaluate parameter influence. This approach balances computational efficiency with the ability to capture non-linear and interaction effects, making it suitable for assessing complex damage responses in Macro-scale structural simulations.

2.6 Machine Learning Algorithms

As previously introduced in Section 2.5.1, RF was used as a tool for SA through feature importance ranking. However, in this section, the focus shifts to its application as a predictive machine learning model.

Artificial Intelligence (AI) is increasingly regarded as a powerful alternative to classical modelling techniques, including advanced numerical methods such as FEM and FDM. AI is the branch of computer science where machines and software systems simulate human-like intelligence. A general overview of the AI family tree is presented in Figure 2.5. Among the various categories, ML is found to be the most widely adopted over the past decade and a half in engineering applications, especially in geotechnics, as shown in (Cao et al. (2022); Obel et al. (2020); Ninić et al. (2017b, 2024)). This includes ANN, Support Vector Regression (SVR) (a kernel-based regression model), and Random Forest Regression (RFR) (an ensemble-based method using decision trees), all of which have been applied successfully in predicting tunnelling-induced ground movements and assessing building damage (Mahmoodzadeh et al. (2022); Hou et al. (2022)).

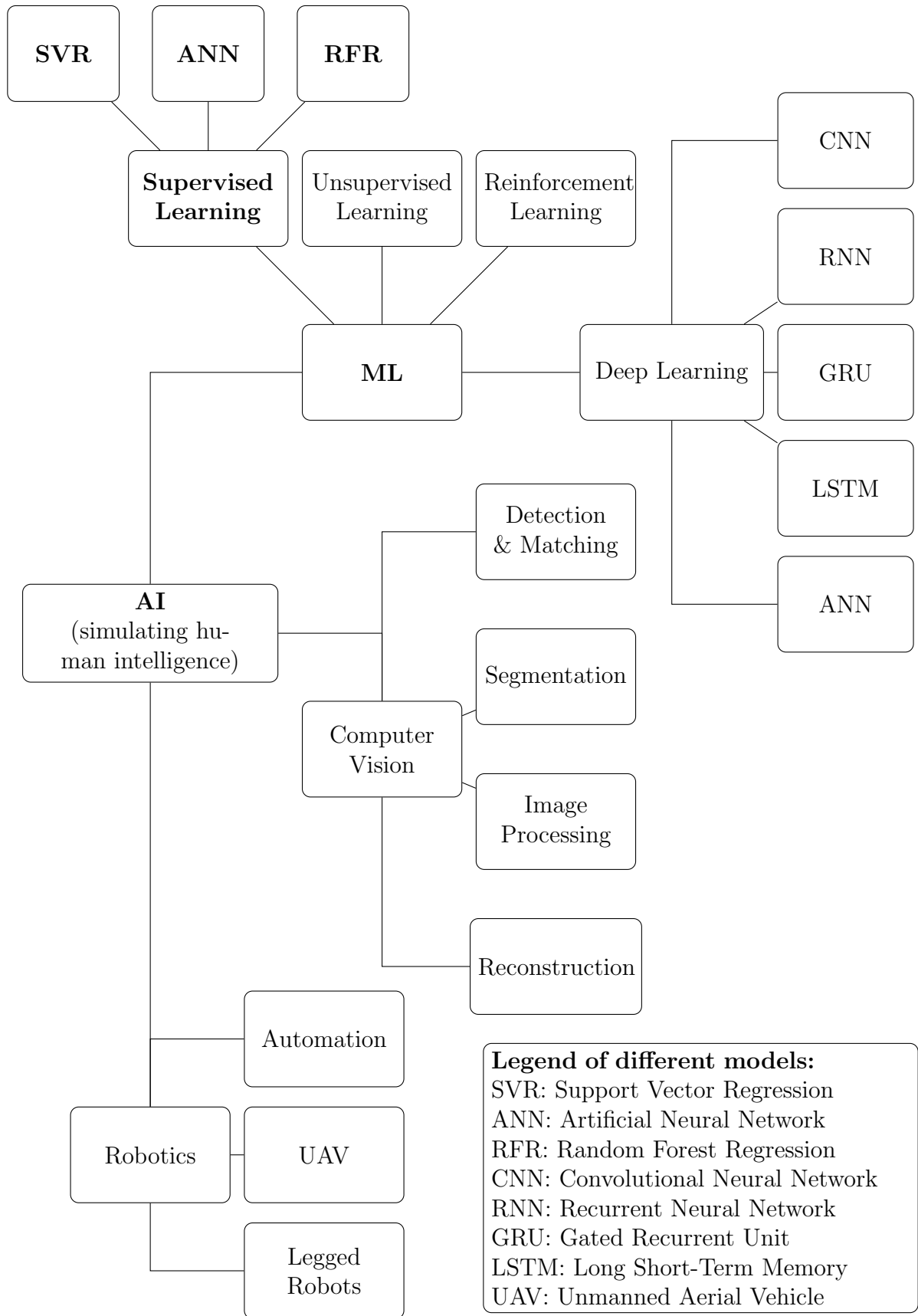


Figure 2.5: Family tree of AI types including ML, deep learning, computer vision and robotics (inspired by the work of (Huang et al. (2021))).

2.6.1 Artificial Neural Networks

Among the commonly used models in geotechnical engineering is the ANN algorithm Ninić et al. (2018) due to its strong predictive performance, which often surpasses other models in similar applications (Cao et al. (2022); Obel et al. (2020)). The ANN algorithm attempts to replicate the human brain's learning mechanisms by recognising patterns across large, highly non-linear datasets. It establishes a relationship between the input parameters X_i (input neurons) and output Y_i (output neurons) via a network of hidden neurons, each connected by weights w , as shown in Equation 2.44.

$$Y_i(X_n, w) = f \left(\sum_{j=1}^{m_{\text{neurons}}} (w_{jk} + \Psi_j) f \left(\sum_{i=1}^n (w_{ij} X_i + \Psi_i) \right) \right) \quad (2.44)$$

The activation function $f()$ transforms the weighted inputs and biases before passing them to the next layer. The weights w_{ij} and w_{jk} connect input-to-hidden and hidden-to-output layers, respectively, while Ψ represents the bias term.

The training process adjusts the weights w to minimise the error Err between the predicted output Y_i and target value T , across m_{pattern} training patterns. The weights are updated via gradient descent as follows:

$$w_{k_{\text{iter}}} = w_{k_{\text{iter}}-1} + \Delta w = w_{k_{\text{iter}}-1} - \lambda \frac{\partial Err}{\partial w} \quad (2.45)$$

where

$$Err = \sum_{j=1}^{m_{\text{pattern}}} (Y_i - T_j)^2 \quad (2.46)$$

The model quality depends on the architecture (e.g., number of hidden layers and neurons), and the learning process parameters such as learning rate λ and iteration count.

A typical structure is illustrated in Figure 2.6.

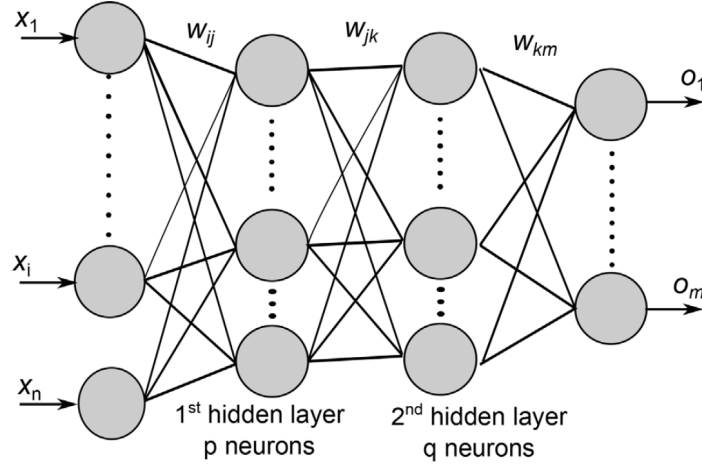


Figure 2.6: Structure of a 4-layer ANN for predicting building strain due to tunnelling (O_1 to O_m corresponding to $Y_i(X_n, w)$).

2.6.2 Data Preprocessing and Training Optimisation

Data Normalisation or Standardisation: To eliminate the influence of input magnitudes and dimensional differences, it is necessary to perform either normalisation Ninić et al. (2018) or standardisation (z-score) (Hou et al. (2022)).

For normalisation, inputs and outputs are mapped to $[0.1, 0.9]$. For a parameter X_i , the normalised value is given by:

$$X_{i, \text{ norm}} = \frac{X_i - X_{i, \min}}{X_{i, \max} - X_{i, \min}} (X_{i, \max}^* - X_{i, \min}^*) + X_{i, \min}^* \quad (2.47)$$

Where $X_{i, \min}$ and $X_{i, \max}$ are the original variable limits, and $X_{i, \min}^* = 0.1$, $X_{i, \max}^* = 0.9$.

For standardisation, values are scaled to have a mean of 0 and standard deviation of 1:

$$X_{i, z} = \frac{X_i - \mu_{X_i}}{\sigma_{X_i}} \quad (2.48)$$

Where μ_{X_i} and σ_{X_i} are the mean and standard deviation of X_i .

Data Splitting: Data splitting divides the dataset into training, testing, and validation sets. The training set fits the model, the test set tunes hyperparameters and prevents overfitting, and the validation set assesses performance on unseen data.

Different ratios have been used in the literature:

- Ninić et al. (2018): 80% training, 15% testing, 5% validation.
- Obel et al. (2020); Cao et al. (2020a): 80% training, 10% testing, 10% validation.
- Cao et al. (2020a): 70% training, 15% testing, 15% validation.
- Hou et al. (2022): 90% training, 10% testing (including half for validation).

Ultimately, the optimal ratio is case-dependent and may require testing different configurations to achieve better accuracy. Thus, multiple ratios should be considered during ML model training.

Model Optimisation Hyperparameter optimisation (HPO):

In essence, every strategy seeking the best solution (model optimisation) can be used for tuning the ML hyperparameters. Various methods exist for model optimisation, including model-free strategies like grid search and random search, as well as those that use statistical models to predict optimal solutions, such as Bayesian optimisation methods (Feurer and Hutter (2019)). Additionally, stochastic methods like the Particle Swarm Optimisation (PSO) have been applied in this context (Ninić et al. (2024)).

Grid search is the most basic HPO method, using a full factorial design. It evaluates the Cartesian product of a finite set of hyperparameter values, testing all possible combinations. The drawback is its computational cost, as it requires a large number of evaluations to explore the full search space. Random search, by contrast, explores the space by randomly sampling configurations within a limited budget. It can sometimes outperform

grid search by allocating more evaluations to potentially influential hyperparameters, although it may still miss the optimal solution.

Bayesian Optimisation (BO) is a more advanced technique for global optimisation of hyperparameters. It operates iteratively, using a probabilistic surrogate model and an acquisition function to select the next point for evaluation. With each iteration, the surrogate model is refined based on all previous observations, improving its predictions. A common acquisition function in BO is the Expected Improvement (EI) (Feurer and Hutter (2019)). For PSO, the optimisation process begins with a population (swarm) of particles that each represent a potential solution. Each particle has a velocity v_{ij} and a position pos_{ij} . The particle's position is updated in each iteration according to its best-known position $pos_{ij}^{p_{best}}$ and the global best position $pos_{ij}^{g_{best}}$ found by the swarm (Ninić et al. (2024)).

2.6.3 Validation of Model Generalisation

k_{fold} cross-validation is a widely used technique in ML to assess a model's ability to generalise to unseen data. This method ensures that performance accuracy is not the result of overfitting. The dataset is divided into k_{fold} equal parts; the model is trained on $k_{fold} - 1$ folds and validated on the remaining one. This process is repeated so that each fold serves as the validation set once. Kohavi (1995) concluded that moderate values of k_{fold} between 10 and 20 strike a good balance by reducing variance at the cost of slightly increased bias. Conversely, smaller values of k_{fold} (e.g., 2–5), particularly with small datasets, tend to increase variance due to instability in the training subsets.

2.6.4 Performance Metrics for Model Evaluation

Performance metrics play a fundamental role in evaluating the predictive accuracy and robustness of supervised ML models, particularly in regression tasks relevant to tunnelling-

induced soil displacements and building damage prediction. Rather than relying on individual predictions, model performance is typically assessed across the entire dataset using error-based metrics.

In the literature, a variety of performance metrics have been employed to assess model suitability for geotechnical applications. For instance, Ninić et al. (2018) and Ninić et al. (2024) adopted the Relative Root Mean Squared Error (rRMSE) (%) to facilitate comparison across datasets with different scales, emphasising its advantage in normalising the root mean square error by the mean of observed values. Similarly, Coefficient of Determination (R^2) was used by Obel et al. (2020) and Shreyas and Dey (2019) to measure the proportion of variance explained by the model, a common choice in structural damage prediction tasks due to its intuitive interpretability. Tang and Na (2021) further incorporated the correlation coefficient Pearson Correlation Coefficient (R) to examine linear dependency between predicted and observed values.

Despite the widespread use of these metrics, each has distinct strengths and limitations. For example, the Mean Absolute Error (MAE), defined as:

$$\text{MAE} = \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i| \quad (2.49)$$

provides a direct measure of average error (in original units) but lacks sensitivity to large deviations. Conversely, the Mean Squared Error (MSE) and its root form, Root Mean Squared Error (RMSE), defined respectively as:

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2 \quad (2.50)$$

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2} \quad (2.51)$$

penalise larger errors more heavily due to squaring, making them suitable when outlier

sensitivity is desirable, a choice reflected in Shreyas and Dey (2019) when prioritising safety in structural deformation estimates.

The coefficient of determination, R^2 , is commonly defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^N (Y_i - \hat{Y}_i)^2}{\sum_{i=1}^N (Y_i - \mu_{Y_i})^2} \quad (2.52)$$

and remains a staple due to its intuitive percentage-based interpretation, though its sensitivity to data variance can be misleading for skewed datasets.

In contrast, the relative root mean square error, rRMSE, offers a scale-independent measure:

$$\text{rRMSE (\%)} = \left(\frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (Y_i - \hat{Y}_i)^2}}{\mu_{Y_i}} \right) \times 100 \quad (2.53)$$

and has been shown to provide more consistent comparability across heterogeneous conditions in tunnelling studies (Ninić et al. (2024)).

Lastly, the Pearson correlation coefficient R , calculated by:

$$R = \frac{\sum (X_i - \mu_{X_i})(Y_i - \mu_{Y_i})}{\sqrt{\sum (X_i - \mu_{X_i})^2 \sum (Y_i - \mu_{Y_i})^2}} \quad (2.54)$$

was used by Tang and Na (2021) to assess the strength of linear relationships, particularly when evaluating how well input features predict observed displacements.

In summary, while multiple performance metrics are available, their selection should be based on the modelling objective. For tunnelling-related building damage, R^2 is favoured for model fit assessment, whereas rRMSE (%) ensures fair comparison across varying magnitudes. The literature suggests that combining multiple metrics provides a more comprehensive evaluation than relying on a single criterion (Obel et al. (2020); Ninić et al. (2018); Gamra et al. (2024)).

2.7 Summary

In this chapter, various components that constitute the methodological foundation of this research are explored. The elements discussed above, when used in conjunction, enable the achievement of the aims and objectives of this thesis. Below is a summary of the methods and concepts addressed:

- Tunnelling-induced settlements: A review of historical and contemporary approaches for estimating ground movements, including the Volume Loss (VL) and Chakeri & Onver method (COM) methods. Definitions of key empirical parameters such as settlement trough width i_x and the settlement coefficient K were also discussed.
- Building damage: Definitions of ground and foundation movements, categorisation of damage levels, and stages of damage risk assessment. The Limiting Tensile Strain Method (LTSM) and its reliance on horizontal strain ε_h were introduced for evaluating structural deformation.
- SSI: Analytical and numerical modelling approaches, including the Modified Vlasov model implemented via the Euler Bernoulli Beam on Two Parameter Elastic Foundation (EBBEF2p) computer code, and a Relative Stiffness-based approach using FEM. The iterative estimation of the interaction factor γ and associated limitations were reviewed.
- Numerical modelling of masonry: Modelling strategies including Micro- and Macro-modelling, discrete and continuum methods, and non-linear material behaviour. Special emphasis was placed on the Concrete Damaged Plasticity (CDP) model and the role of mesh-objective constitutive modelling using the characteristic length h .
- Sensitivity Analysis (SA) and Sampling: A critical overview of global SA approaches, highlighting the importance of efficient sampling techniques such as Latin Hypercube (LH) sampling. Emphasis was placed on the use of machine learning-

based models, particularly Random Forest (RF) regression, for identifying influential parameters in high-dimensional structural simulations.

- Machine Learning (ML) algorithms: Focus on supervised learning models such as Artificial Neural Networks (ANN), including preprocessing steps like data normalisation, standardisation, and splitting. Model training strategies were also reviewed, including hyperparameter tuning, generalisation via cross-validation, and evaluation through standard performance metrics.

These topics collectively form the basis of the methodology, which is presented in detail in the following Chapter 3.

Chapter 3

Methodology Overview

As briefly mentioned in the introduction chapter, the aim is to enable real or near-real-time assessment of tunnelling-induced building damage within a BIM environment, as illustrated in Figure 3.1. This approach is implemented as a user-friendly plug-in for industry-standard BIM design software, allowing versatile integration of various analysis and visualisation tools.

Initially, a parametric tunnel information model is developed using state-of-the-art BIM design tools, offering a user-friendly interface to control tunnel design parameters. The prediction of ground settlements based on these parameters is then implemented using empirical solutions such as (Potts and Addenbrooke (1997); Chakeri and Ünver (2013); Hajjar et al. (2015); Tan (2019)). The effects of SSI between the overlying building and the induced ground movements are then evaluated using the Vlasov model (Teodoru (2009a)).

Finally, the damage induced in buildings due to SSI is estimated through simulation-based meta-models. These are developed using the best-performing ML algorithm, trained on a synthetic dataset generated via a high-fidelity FEM model of a masonry structure accounting for non-linear material behaviour. The dataset is further optimised through

advanced optimisation algorithms such as BO or PSO.

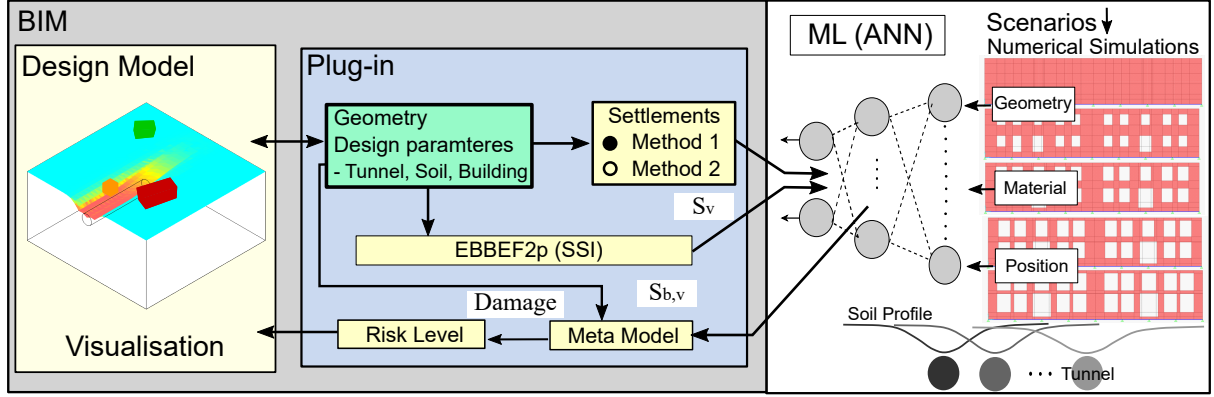


Figure 3.1: Concept for real-time damage assessment within BIM.

The FEM models form the basis of the methodology, integrating parameters such as building length (L), percentage of openings (O), eccentricity relative to tunnel axis (e), and settlement magnitude ($S_{v,max}$). Structural responses are evaluated in terms of induced damage and classified using common damage scales such as the five-level classification from Boscardin and Cording (1989), ranging from “negligible” to “severe.”

The trained meta-models are embedded in the BIM environment and use input parameters from tunnel design, surface settlements ($S_{v,max}$) accounting for SSI, and building geometry and material characteristics. Both settlement and risk levels are then visualised within the BIM interface (Figure 3.1, left: “Design Model”).

Compared to existing approaches that use complex coupled SSI models for damage prediction Cao et al. (2020a, 2022), this method achieves similar predictive capabilities by integrating simpler empirical and analytical methods widely accepted in practice. This enables a more flexible and practical implementation within BIM environments.

The methodology is inspired by model design and analysis strategies proposed in the literature Giardina et al. (2015); Cao et al. (2022); Obel et al. (2020), and is divided into two main stages (illustrated in Figure 3.2):

1. **Preprocessing phase:** Identification of input parameters, numerical modelling

and validation (i.e., Verification and Validation (VV)), and generation of synthetic datasets.

2. **Postprocessing phase:** Evaluation of local and global damage risks, sensitivity analysis (SA), ML model training and deployment, and implementation within the BIM environment.

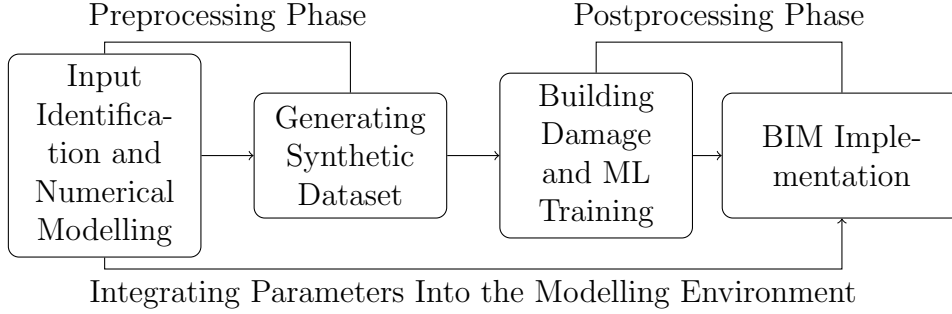


Figure 3.2: Principal stages of methodology towards research goals.

3.1 Input Identification and Numerical Modelling

The methodology commences with the identification of input parameters, followed by the development and validation of both numerical and analytical models. These include FEM and analytical SSI approaches. The FEM simulations are validated against experimental data, while the SSI models are compared with observations from real-world field data, providing two complementary validation pathways. The following steps are undertaken to generate rich and diverse scenarios of different buildings exposed to tunnelling:

- Analysing real cases, experiments, and models for validation of the analytical and numerical approaches. This includes studies such as Lourenco (1996); Teodoru (2009b); Giardina (2013); Frischmann et al. (1994), which provide comprehensive data on real projects and model scales relevant to tunnelling-induced damage.

- Developing accurate modelling strategies to capture building behaviour under tunnelling effects, particularly accounting for non-linear structural response. Notable strategies are highlighted by (Giardina (2013); Yiu et al. (2017)).
- Identifying relevant input parameters for the adopted modelling strategy, including parameter domains and increments. This selection is supported by systematic literature reviews through multiple search engines (Scopus, NUSearch, Google Scholar, etc.) using targeted keywords and Boolean operators. Notable references include (Giardina et al. (2015); Mair et al. (1996); Franzius (2003); Son and Cording (2007); Potts and Addenbrooke (1997)).
- Recognising correlations between parameters and physical constraints, such as those explored by Ghiassi and Milani (2019); Mair et al. (1993) for tunnel and material parameter interactions, and Hillerborg et al. (1976); Mair et al. (1993) for limits on allowable settlement (e.g., $S_{v,\max}$). Additionally, numerical stability considerations, including snap-back effects at the constitutive level, are addressed based on (Lourenco (1996); Feenstra (1993)).

3.2 Generating the Synthetic Dataset

Following the identification of input parameters, a model prototype (Model 0) is developed using FEM, incorporating an appropriate material model and optimising the FEM solver setup. Through Python scripting, the model becomes parametrically adjustable, accommodating changes in input parameters based on the sampled points. This enables the autonomous generation of multiple tunnel–soil–building configurations, essential for training ML algorithms, which require large datasets for learning and generalisation, further discussed in Section 7.

The procedure, including input identification and model generation, is illustrated in the schema in Figure 3.3, detailing each step in chronological order up to model execution.

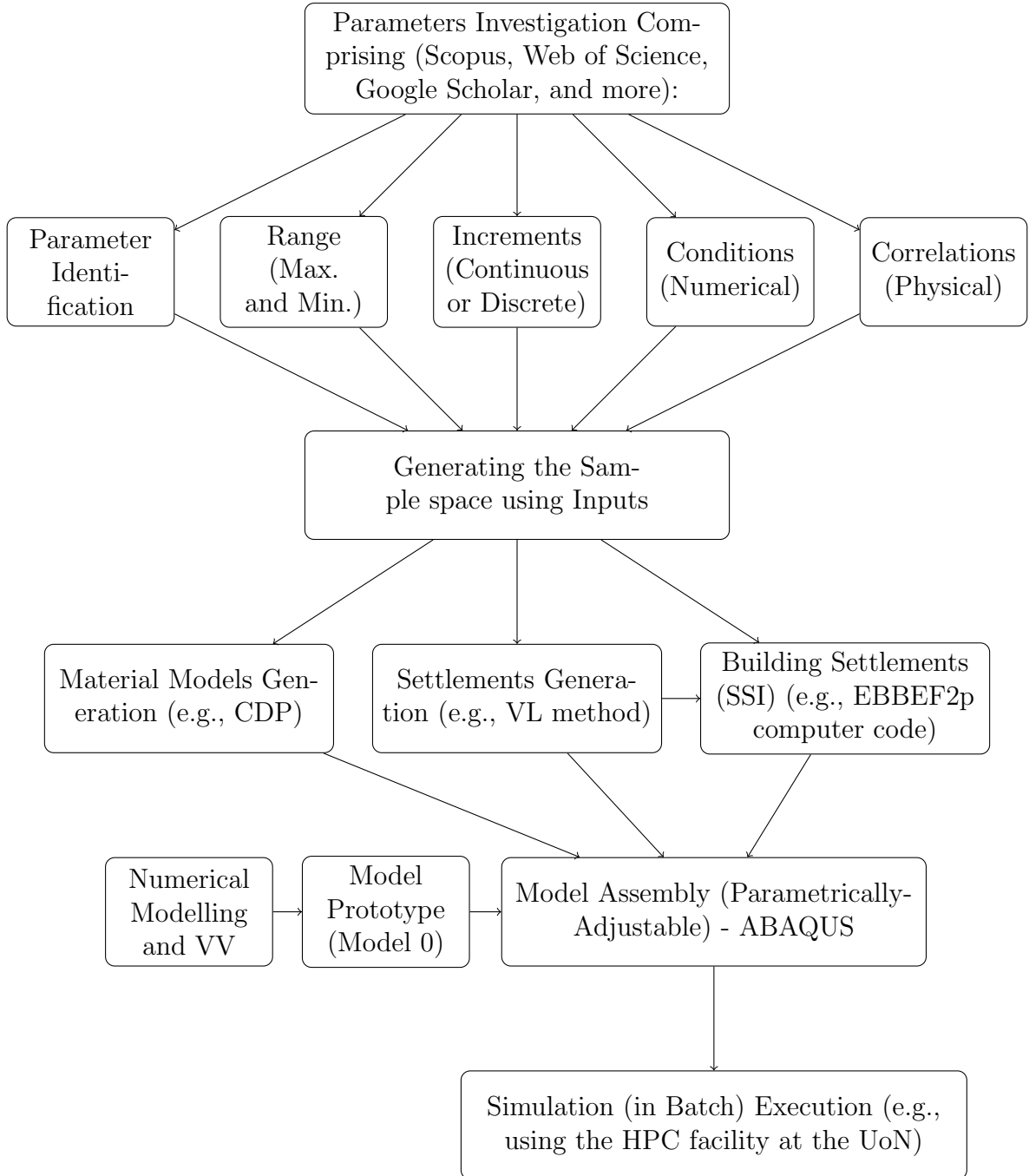


Figure 3.3: Preparation stage (pre-analysis) schema of the methodology.

The schema in Figure 3.3 summarises the preparation phase. A detailed breakdown of its components is provided below.

Parameter Investigation: This process involves collecting and coordinating the input parameters through the following tasks:

- **Parameter Identification:** Extracting relevant modelling parameters via research databases (e.g., Scopus, Google Scholar), guided by established works in the field.
- **Range (Max and Min):** Determining operational limits for each parameter.
- **Increments (Continuous or Discrete):** Establishing sampling resolution for parameters based on whether they are continuous or discrete.
- **Conditions (Numerical):** Ensuring numerical stability of models, referencing governing conditions (see Section 2.4.2).
- **Correlations (Physical):** Establishing logical relationships between parameters (e.g., physical or structural constraints) as will be discussed in Section 5.1.

Generating Sampling Space: By using LH sampling as the preferred sampling procedure (as discussed in Section 2.5.1), a broad range of different scenarios is generated. This ensures comprehensive coverage of building configurations with respect to tunnel and soil properties, facilitating the development of a reliable and robust dataset for ML model training.

Generating Numerical Model Components: The approach integrates empirical, analytical, and numerical methods, as described below:

- **Settlements Generation:** ($S_{v,\max}$ and $S_v(x)$) are computed using the methods described in Section 2.1.1, including the VL and/or COM approaches.

- **Material Model Generation:** Non-linear material models are created using the CDP model in ABAQUS to simulate the behaviour of masonry buildings, as detailed in Section 2.4.2.
- **Building Settlement (SSI) Generation:** SSI effects are calculated using the EBBEF2p computer code of Teodoru (2009b), as presented in Section 2.3.1.

Numerical Modelling and VV: In parallel with the generation of the sample space, a rigorous VV process is carried out for the CDP model. The selected models are based on key findings from the literature to ensure a comprehensive validation across fundamental aspects of the VV process. The simulation results are compared against benchmark data and experimental observations, as detailed below:

- **Mesh-Independence Studies:** In line with the recommendations of Lourenco (1996); Feenstra (1993), mesh objectivity is verified through post-failure analysis using multiple mesh sizes. This is achieved by correctly implementing the characteristic length, h , at the constitutive level, as discussed in Section 2.4.4.
- **Shear Wall under Combined Loading Conditions:** To validate the CDP model at a Macro level, a full-scale shear wall model is assessed under combined loading. Validation is performed through comparisons of global Force-Displacement (F-D) curves and visual damage patterns up to failure (Lourenco (1996)).
- **Scaled Façade Subject to Tunnelling-like Subsidence:** The experimental wall test of Giardina (2013) is used to validate the behaviour of a structure undergoing hogging-type settlement. This step serves as the closest verification scenario to tunnelling-induced damage in buildings. The model is evaluated in terms of global façade displacements and localised damage such as cracking.

3.2.1 Integration to Model Prototype

Model Prototype (Model 0): Following the previous steps, a base model is developed in ABAQUS, incorporating the CDP material model for the masonry façade and the EBBEF2p-derived settlement profile as the foundation boundary condition. Additionally, the model prototype includes geometric features such as door and window openings to enhance realism. This prototype serves as the foundation for generating numerous samples, with scenarios parametrically adjustable through Python scripting, including:

- Adjustable building-related parameters, such as geometry and material properties.
- Adjustable tunnel and soil parameters to account for varying applied settlements at the building foundation.

Assembling for a Unified Model: As previously mentioned, all components are integrated into a unified, parametrically adjustable model using the APDE environment in ABAQUS. This setup accommodates the full range of input parameters from the sample space, allowing for automated generation of various scenarios, as will be illustrated in Section 5.

Simulations Executions: The generated input files are executed in batch mode using the HPC facility at the UoN, which provides the computational power necessary to simulate a large number of models. This step is critical for building the extensive dataset of input–output relationships required for training the ML algorithms.

3.3 Post-Analysis Evaluation Using Damage Metrics, Machine Learning, and Building Information Modelling

Following the pre-analysis and execution stages, the evaluation of damage and subsequent post-analysis begins. This process is schematically represented in Figure 3.4. During this stage, various outputs from the FEM simulations are collected and linked to their corresponding sampled input models. These data are then used for postprocessing tasks, including SA and ML training.

The schema of Figure 3.4 illustrates the steps involved in the evaluation stage, including the preparation (pre-training phase) of ML models. A detailed description of each component is provided below:

Extraction of Field Outputs: Field outputs such as nodal displacements and maximum principal tensile strains are extracted from the simulations and utilised within the damage assessment framework, following strategies similar to those used in building damage studies (Giardina et al. (2015); Son and Cording (2005)).

Assigning Damage: Building damage is categorised based on the classification schemes presented in Section 2.2.3, allowing the assessment of both global and local damage levels.

ML Initiation: As a pre-training requirement, all simulation data are cleaned, standardised, and split into training and testing datasets, as discussed in Section 2.6.

Optimisation and Generalisation of ML: Model training is enhanced using hyperparameter optimisation methods. To assess the generalisation of each trained model, cross-validation techniques such as k_{fold} are used, providing insight into how well the model performs on unseen data (see Section 2.6).

Assessing Performance: The trained models are evaluated using metrics including the

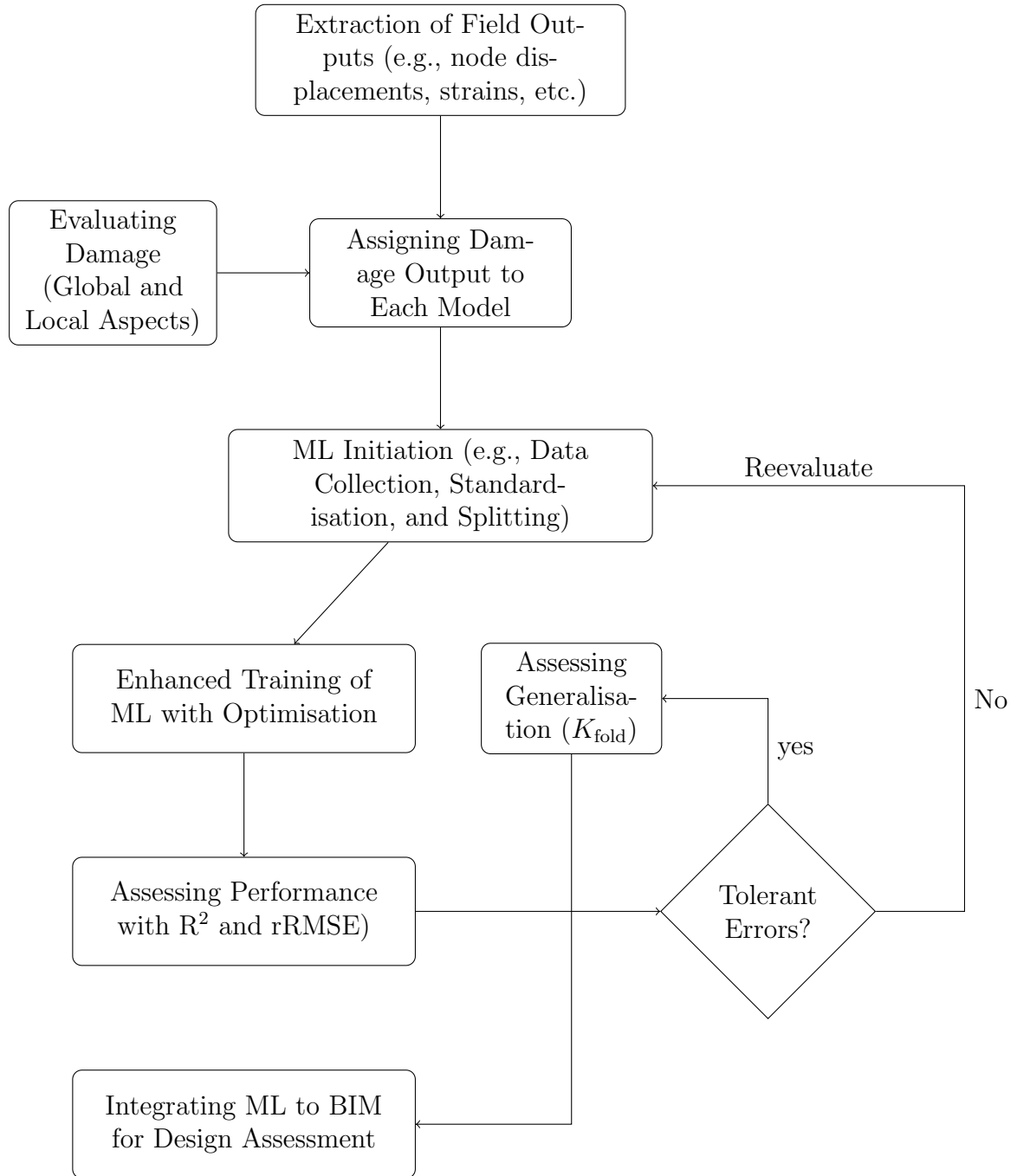


Figure 3.4: Assessment stage (post-execution) of FEM models (back-end of the methodology).

coefficient of determination (R^2) and the relative root mean square error (rRMSE), both described in detail in Section 2.6.4.

Integrating ML into BIM: Once the desired prediction accuracy is achieved, the trained ML model is integrated into a BIM environment (e.g., Revit) via Dynamo’s parametric scripting. This enables real-time assessment of tunnelling-induced building damage within the design model.

3.4 Summary

In summary, the methodology aims to develop a comprehensive framework for the assessment of building damage due to tunnelling using ML algorithms, implemented within a BIM platform. To summarise, the methodology is devised into:

- **Pre-execution phase**, including the generation of numerous FEM scenarios assembled using ABAQUS ABAQUS Python Development Environment (APDE), by means of:
 - Systematic Verification and Validation (VV) of numerical and analytical models used for subsequent dataset generation and damage analysis (presented in Chapter 4)
 - Effective sampling procedures of the identified input parameters, including considerations for parameter domain, increments, and physical and numerical restraints. (presented in Chapter 5)
 - A combination of empirical (e.g., VL), analytical (e.g., EBBEF2p), and numerical (e.g., CDP) models. (presented in Chapter 5)
 - Model batch executions using the HPC facility at the UoN. (presented in Chapter 5)

- **Post-execution phase**, including the assessment of building damage, training of ML algorithms, and integration into BIM, by means of:
 - Using a developed damage evaluation criterion to assess building damage on the various FEM models. (presented in Chapter 6)
 - Data processing of input–output correlations, model training, and evaluation. (presented in Chapter 7)
 - BIM implementation and in-tool scenario execution for near real-time predictions of building damage. (presented in Chapter 8)

These steps ensure the achievement of efficient and reliable analysis of building damage within the BIM platform, substituting the need for expert and advanced modelling techniques or complex integration of various components constituting this multidisciplinary problem. It also minimises potential human errors during data input, simulation execution, or manual damage assessments.

Chapter 4

Numerical Modelling

Numerical models play a vital role in predicting material non-linear behaviours. In the context of masonry structures, advanced numerical tools are essential for simulating building performance from the initial loading stages through to complete material failure, that is, from the onset of cracking to the total loss of strength. As previously discussed in Section 2.4, masonry can be modelled using Micro-modelling techniques, which involve representing individual units, mortar, and interfaces, capturing the material's heterogeneity at the cost of high computational demand. On the other hand, Macro-modelling assumes masonry as an anisotropic composite material, relating average strains and stresses. This approach offers a balance between accuracy and computational efficiency, making it more suitable for large-scale simulations. It is phenomenological and relies on material parameters derived from large-scale masonry tests conducted under homogeneous stress conditions.

Nonetheless, both Micro- and Macro-models face significant challenges due to the intrinsic properties of masonry, including the formulation of robust numerical algorithms and the satisfactory representation of inelastic behaviour (Lourenco (1996)). Modelling buildings subjected to tunnelling-induced settlements requires a focus on global structural behaviour Lourenco (1996); Giardina (2013); Yiu et al. (2017), making Macro-modelling

the most appropriate method in such cases.

In this section, the VV process based on benchmark data is conducted. As described in Section 3.2, the VV investigation begins with the verification of mesh dependency through characteristic length h , where the post-failure regime becomes objective under mesh refinement, ensuring consistent post-peak behaviour regardless of mesh size (see Section 2.4.4).

This is followed by the modelling of a large-scale experimental wall under excessive in-plane deformations. In this case, both tension and compression regimes are activated, along with the plasticity parameters of the CDP model. This broader range of input helps assess the model's ability to predict masonry behaviour under combined loading conditions. Lastly, an experimental masonry façade with openings is modelled to undergo settlements similar to those produced by tunnelling activities (hogging mode in this particular case). The model is validated using the CDP-calibrated parameters, and its accuracy is assessed based on both global and local measures of damage, using previously discussed criteria (Section 2.2.2) and newly developed benchmarks.

4.1 Mesh Independence Study

In the analysis of post-peak behaviour of strain-softening materials, achieving mesh-independent results is essential to ensure that numerical simulations yield objective and reliable outcomes. This requirement has led to the introduction of an equivalent length, h , into the stress-strain formulation, allowing the analysis to remain consistent under mesh refinement. The concept of incorporating h has been widely discussed in the literature since the work of Bazant and Oh (1983), where the importance of objectivity in relation to mesh size was first formally addressed. Achieving mesh independency ensures that computed material responses, such as fracture energy, $G_{f(c \text{ or } t)}$, remain constant regardless of mesh size, thereby enhancing the replicability and reliability of structural

simulations. To demonstrate the role of h in preserving energy consistency during mesh refinement, a simple bar loaded in uniaxial tension is presented, drawing parallels with studies that verified this behaviour in tension Crisfield (1982) and compression (Feenstra and Borst (1996); Lourenco (1996)).

Consider the bar depicted in Figure 4.1, which measures 80 mm thick, 240 mm wide, and 200 mm in height. The bar features a weakened central section with a strength reduced by 10% compared to the edge elements, to trigger localisation. The material properties for both the edge and core sections are as follows: the elastic modulus and Poisson's ratio are identical for both regions, with $E = 10000$ MPa and $\nu = 0.2$, respectively. The tensile strength, F_t , is 1 MPa for the edge and 0.9 MPa for the central section. The fracture energies, G_{ft} , are 0.029 and 0.026 N.mm/mm², respectively.

The bar's base is fully constrained against translational movement, while the top is restricted from horizontal displacement and subjected to a uniaxial tensile force along the top edge. Displacements are monitored at a designated point located at the top of the bar.

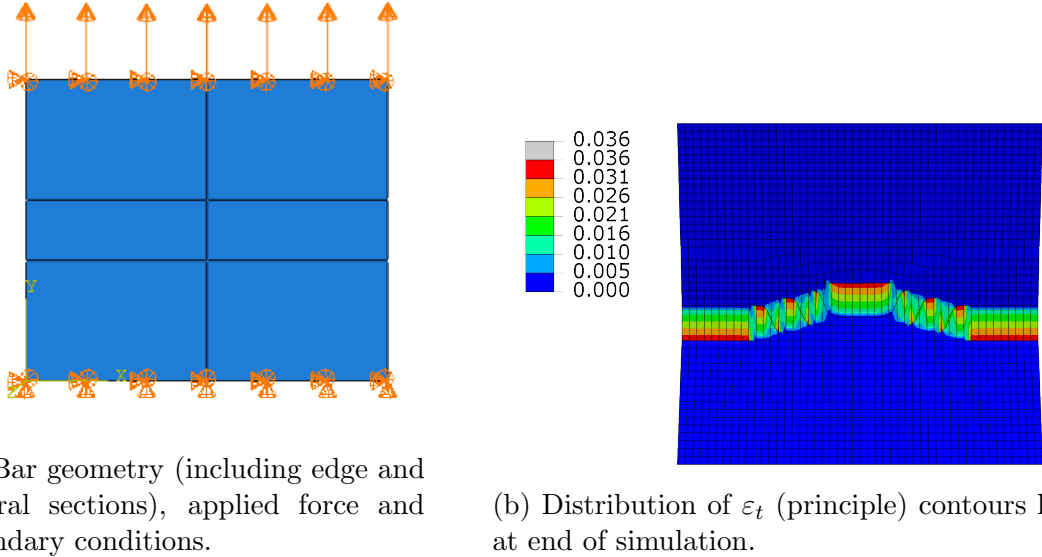


Figure 4.1: Assembly of bar in tension with an applied maximum displacement of 0.05 mm.

The analysis is performed following a two-phase procedure. Initially, the boundary con-

ditions are activated to ensure the stability of the model. Next, a tensile force is incrementally applied until complete failure of the bar. Shell elements (ABAQUS type S8R), using quadratic integration for increased accuracy, are employed throughout. Mesh sizes of 20, 10, and 5 mm are considered to evaluate the mesh independency of the post-peak responses.

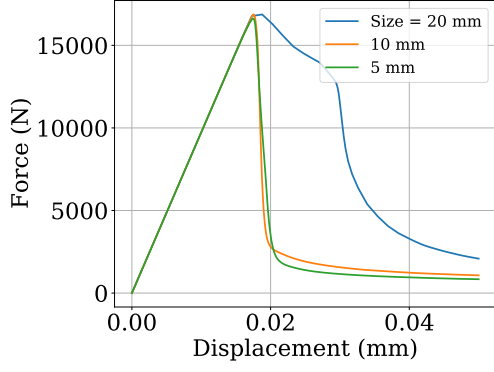
The first set of analyses is conducted without adjusting the equivalent length, h , in the stress-strain relationships, contrary to the recommendations provided in Table 2.5, to illustrate high mesh sensitivity. Subsequently, h -adjusted models are analysed (i.e., $h = \text{mesh size}$), using variable h in Equations 2.37 and 2.40. Lastly, ABAQUS's stress-displacement model is tested (as described in Section 2.4.4) to assess its built-in energy regularisation capability.

The analysis is carried out using a general static step with the following settings: NLgeom off, maximum number of increments = 100000, initial increment size = 0.0001, minimum = 1E-35, and maximum = 0.03.

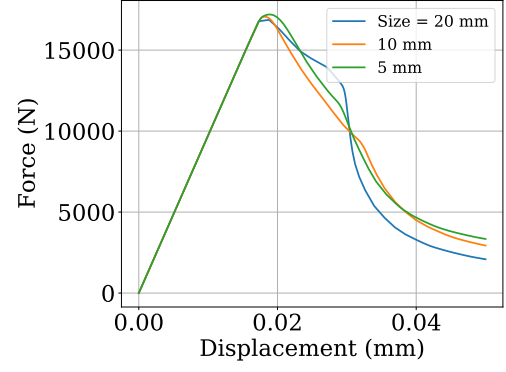
The energy-based regularisation comparative results are depicted in Figure 4.2. The results show the Force-Displacement (F-D) response of the bar to be almost entirely independent of mesh size when regularised. Specifically, in Figure 4.2b, where regularisation is applied (i.e., h is equivalent to the mesh size), the model maintains consistent post-peak behaviour across different mesh sizes.

In contrast, Figure 4.2a shows the response of the unregularised model using a consistent value of $h = 20$ mm. In this case, the response becomes noticeably more brittle as the mesh is refined. This occurs because h appears in the denominator of the regularisation equation, resulting in smaller values of G_{ft} being stored in smaller elements. Consequently, this leads to a notable reduction in the ductility of the post-peak response.

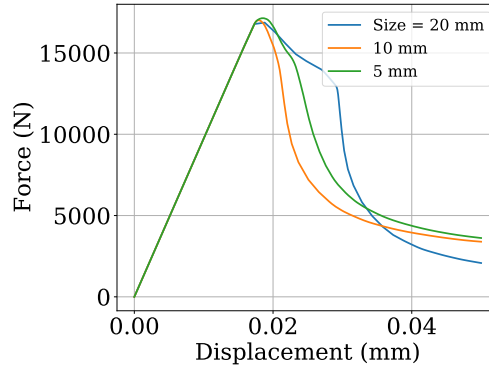
On the other hand, Figure 4.2c presents the results using a stress-displacement relationship instead of a stress-strain model. This approach does not involve h , and is solely



(a) Unregularised stress-strain (h is constant at 20 mm).



(b) Regularised stress-strain (h equals refined mesh size).



(c) Internally regularised stress-displacement (as a function in G_f).

Figure 4.2: F-D diagrams of the simple bar in tension using different constitutive approaches including (a) Unregularised stress-strain (b) Regularised stress-strain (c) and Internally Regularised stress-displacement.

based on the value of G_{ft} .

Although this model shows some discrepancies during the initial phase of softening, it clearly produces consistent curves that are comparable to those from the regularised stress-strain model in Figure 4.2b. This consistency marks a significant topic in numerical modelling, particularly in the context of post-peak behaviour in both tension and compression. In conclusion, the uniaxial bar test serves as a clear and effective benchmark to demonstrate the importance of using regularised constitutive laws. Either a stress-strain formulation with mesh-adjusted h , or a stress-displacement model based on fracture energy G_{ft} , must be adopted to ensure that simulation results remain objective and independent of mesh refinement.

4.2 Shear Wall and Confined Loading Conditions

Moving on to full-scale Macro-modelling of masonry structures, it is important to understand that this type of modelling involves approximations due to the complex failure mechanisms inherent to masonry material. These mechanisms cannot be directly replicated in models and are represented more broadly in a continuum. As mentioned before, Macro-models are applicable when the global response is the primary target. Ideally, the analysis of masonry structures should be performed using an anisotropic continuum model validated against experimental data from the literature.

In the past few decades, experiments have been conducted on masonry at various scales, including tests on small or single units to wallet-sized masonry triplets under uniaxial and combined loading Kaushik et al. (2007); Furtado et al. (2016), as well as small- to large-scale shear panels subjected to in-plane and out-of-plane cyclic or monotonic loading (Lourenco (1996); Manchego and Pari (2016)). These have become widely available in the literature. Since Macro-modelling is considered in this study, large-scale panels are often preferred as a basis for numerical models (Lourenco (1996); Yacila et al. (2019); Giardina et al. (2015)).

In this study, focus is placed on the large-scale tests conducted at ETH Zurich, as reported in the work of (Lourenco (1996)). The test comprises an unreinforced masonry wall made of hollow bricks, subjected to an initial vertical load followed by a horizontal force applied to the confining slab. The loading is applied monotonically until complete failure of the wall. These experiments are highly suited for model validation, as they not only involve large-scale structures with well-distributed cracking, but also provide most of the essential parameters needed to define the numerical model parameters that are obtainable from biaxial testing.

Figure 4.3 illustrates the geometry of the wall, consisting of a masonry panel with dimensions $3600 \text{ mm} \times 2000 \text{ mm} \times 150 \text{ mm}$, and two flanges with dimensions $150 \text{ mm} \times 2000$

mm \times 600 mm attached to the panel's ends. A concrete slab is placed on top of the specimen with dimensions 3600 mm \times 160 mm \times 1400 mm. The material properties of each component are found in Table 4.1. The constitutive material model is the orthotropic law of CDP, using the relationship in Equation 2.35 to represent the compressive response and Equation 2.40 for tension, similar to the bar validation in Section 4.1. The default set of plasticity parameters is used and given as follows: $\psi_{CDP} = 42^\circ$ (calibrated), $e_{CDP} = 0.1$, $\sigma_{ob}/\sigma_{0c} = 1.16$, $k_{CDP} = 0.67$, and $\mu_{CDP} = 0.0001$.

A thorough investigation using both regularised stress-strain and stress-displacement models was initially conducted across three mesh sizes (150, 75, and 50 mm). All models resulted in similar F-D curves, with the only difference being narrower crack bands observed with decreasing mesh size. Shell elements (ABAQUS type code S8R), with quadratic integration for increased accuracy, were used in the numerical simulations. Since both constitutive laws and various mesh sizes produced similar results, only the output from the regularised stress-strain model with 50 mm mesh is reported.

As briefly mentioned before, in the experimental setup, the concrete slab is reinforced and connected to additional steel beams for the application of the confinement load. Due to the absence of non-linear behaviour in the experimental results and the lack of information regarding the slab's behaviour, modelling the slab as linear elastic is considered an acceptable approach.

The total vertical load (P), evenly distributed on the wall, is 415 kN. A horizontal force (displacement) (h_f), is further applied to the concrete slab from one end, and the displacement is monitored from the other end. The base of the wall, including the panel and both flanges, is restrained in all directions, while the top of the slab is restricted only in the out-of-plane direction.

The analysis begins with the activation of boundary conditions. Vertical and gravitational loads are applied simultaneously as confining loads, followed by an incremental increase in h_f until complete failure of the specimen is observed, i.e., at 15 mm total h_f . A similar

analysis procedure to that of the bar validation is followed initially, using a general static procedure with the following settings: NLgeom off, maximum number of increments set to 100000, and initial, minimum, and maximum increment sizes of 0.0001, 1E-35, and 0.03, respectively.

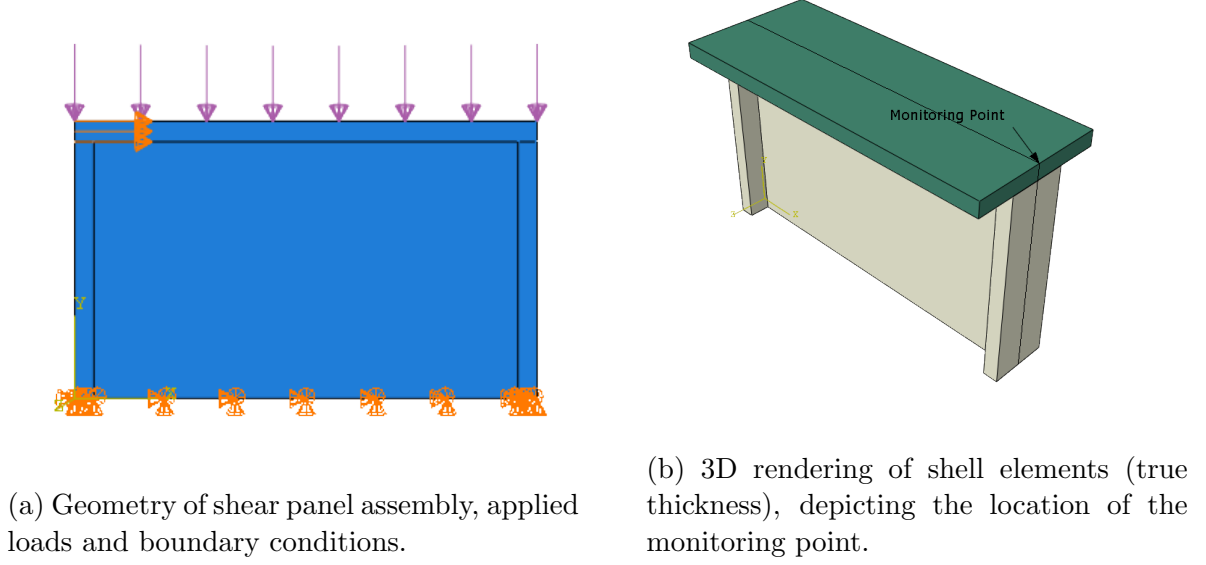


Figure 4.3: Macro-modelling of the shear panel.

Table 4.1: Model material properties and applied loads.

Properties	Slab	Panel	Flanges
ρ (kg/m ³)	2100	2100	2100
E (MPa)	30000	2460	2460
ν	0.20	0.18	0.18
f_t (MPa)	-	0.28	0.68
f_c (MPa)	-	1.87	9.50
G_{ft} (N.mm/mm ²)	-	0.02	0.02
G_{fc} (N.mm/mm ²)	-	5.00	5.00
ε_p	-	0.0008	0.0008
P (kN)	415	-	-
h_f (mm)	15	-	-

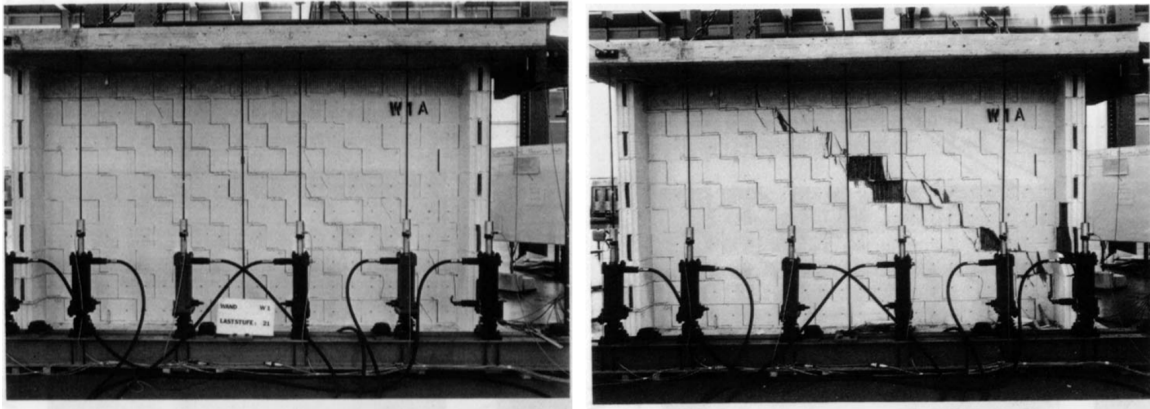
The comparison of cracking patterns between experimental and numerical results is shown in Figures 4.4 and 4.5, respectively. Each figure illustrates the wall's response at the peak load, reached at approximately $h_f = 2$ mm (i.e., subfigure (a)), and at complete failure, $h_f = 15$ mm (i.e., subfigure (b)).

While the visible onset of cracking is observed at $h_f = 2$ mm in the experimental results,

these were not accurately captured by the FEM, as expected, since Macro-modelling approaches typically struggle to reflect early-stage local non-linear deformations in the material. On the other hand, at $h_f = 15$ mm, better agreement in failure patterns is observed, despite the limitations inherent to the assumptions of the CDP model and the use of Macro-modelling techniques.

The FEM model, although not explicitly reproducing detailed local cracking, successfully captures the general trend of damage evolution under combined vertical and horizontal loading. Specifically, the model shows that cracks tend to initiate and propagate diagonally from one corner of the panel to the opposite, accompanied by flexural cracking at both flanges. This progression ultimately leads to the formation of a shear band across the specimen, connecting these regions of damage.

In conclusion, the model demonstrates the ability to capture the dominant, mature damage mechanisms observed in the experimental structural failure.



(a) At peak load (total displacement 2 mm). (b) At failure (total displacement 15 mm).

Figure 4.4: Failure patterns of experimental wall (After Lourenco (1996)).

A F-D diagram of base reaction versus displacement at the monitoring point is illustrated in Figure 4.6, comparing the experimental results, the FEM simulation, and the numerical results reported by (Lourenco (1996)). The combination of a relatively low initial vertical load and the confinement provided by the flanges led to extremely ductile behaviour. Lourenco (1996) reports that the unloading observed at approximately 2 mm is attributed

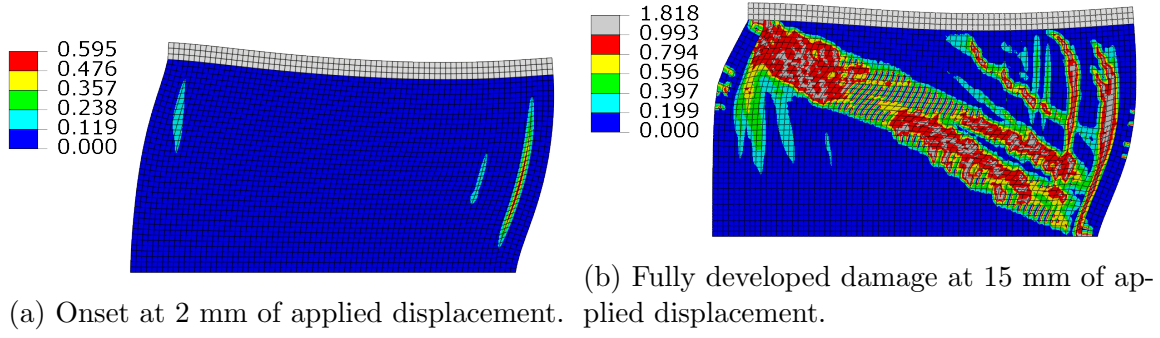


Figure 4.5: Tension damage distributions contours maps from FEM simulations (values > 0.99 indicate fully developed damage “red”).

to Mode I crack opening in the left flange, an effect also captured in the FEM model.

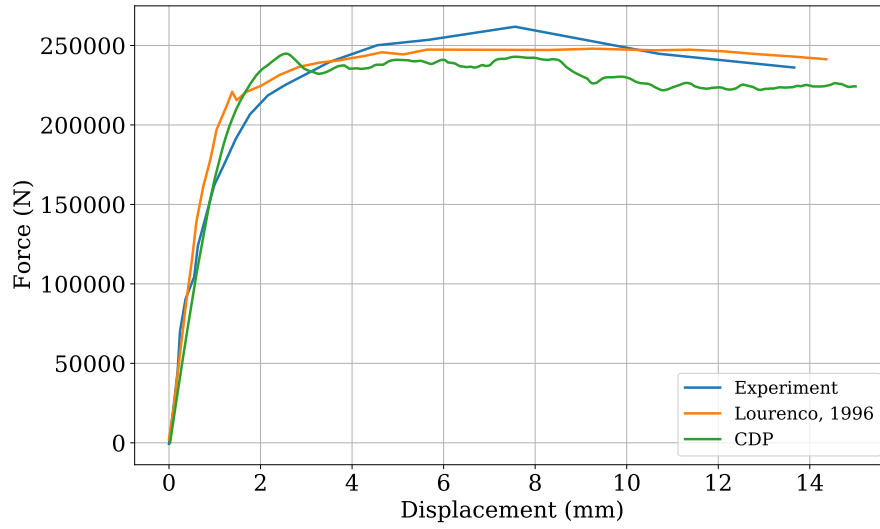


Figure 4.6: Comparison of diagrams between experiment curves and FEM output.

The general static step has been replaced with a dynamic implicit procedure, as it resulted in faster analysis and more stable convergence during simulations. An initial investigation showed that, on average, the dynamic procedure reduced analysis time by up to threefold compared to the general static procedure. Furthermore, memory usage was notably reduced, by up to tenfold, allowing for a wider range of incremental step monitoring. Additionally, several simulations that failed to converge using the general static approach were successfully completed using the dynamic procedure. Importantly, comparisons between static and dynamic analyses showed that the resulting damage patterns, displacements, and strains were identical, with no loss of accuracy. This conclusion was supported by monitoring the kinetic and internal energy histories: the kinetic energy

remained flat and within 5% of the internal energy, which is the accepted threshold for quasi-static conditions with negligible inertia effects (Demirci et al. (2018)). However, the total time assigned to the applied displacement h_f must still be carefully selected, as overly short periods can induce artificial inertia-driven damage (see Equation 2.27). To evaluate this, several trial periods were tested (0.01, 0.02, 0.1, 0.5, 1, 2, and 3 seconds). The results showed that periods of 1 second and longer maintained negligible inertia effects with no change in strain or displacement outcomes, while shorter periods introduced artificial dynamic amplification. Longer periods offered no additional accuracy benefit but increased computational time significantly. Therefore, a period of 1 second was selected for the analysis, balancing computational efficiency and numerical stability. It is noted that the optimal period may depend on the modelling scale, material properties, and boundary conditions, and should therefore be evaluated on a case-by-case basis.

With this, the ideal settings for the analysis become: a period of 1 second, NLgeom off, a maximum of 100000 increments, and initial, minimum, and maximum increment sizes of 0.0001, 1E-15, and 0.03, respectively.

Upon concluding the initial phase of investigations by achieving good agreement between the experimental results and the FEM output, alongside establishing a more efficient and reliable analysis procedure using the dynamic implicit method, the next step is to monitor the influence of CDP plasticity parameters on the output. Addressing the variability of these parameters is known to significantly affect results (Silva et al. (2021); Behnam et al. (2018); Rainone et al. (2023)). A systematic variation of these parameters is performed using common domains and sampled via the OAAT method, with all simulations conducted using the dynamic implicit approach based on the initial arrangement of values as previously described.

Figure 4.7a illustrates the F-D diagrams for different values of ψ_{CDP} (ranging from 20° to 52° in increments of 4). From the figure, the stiffness of the response appears largely unaffected by this parameter. This is because ψ_{CDP} , the dilation angle, governs the

volumetric plastic strain and primarily influences the non-linear response of the material. Therefore, the discussion now focuses on the peak and post-peak regions of the response. Increasing ψ_{CDP} raises the peak strength of the model, aligning it more closely with the experimental results. This is followed by a plateau phase, suggesting that ductility is not significantly influenced by this parameter. These results show good agreement for ψ_{CDP} values between 40 and 44 when compared to experimental data; hence, an average value of 42 is considered ideal in this case. The strength enhancement observed with increasing ψ_{CDP} is consistent with the findings of Rainone et al. (2023) and Silva et al. (2021), although the exact optimal value appears to be case-specific, depending on the modelling technique and applied loading conditions.

Furthermore, and in agreement with Silva et al. (2021), Figure 4.7b illustrates the impact of varying e_{CDP} , the flow potential eccentricity, which controls the shape of the plastic potential function. Variations in this parameter result in negligible changes in model response; hence, the default value of 0.1 recommended by ABAQUS is adopted. Similarly, altering σ_{0b}/σ_{0c} (Figure 4.7c), which defines the ratio of biaxial to uniaxial compressive strengths, results in only a slight reduction in the average force in the post-peak regime. Therefore, the ABAQUS default value of 1.16 is retained.

In contrast to the conclusions of Silva et al. (2021) and Behnam et al. (2018), increasing k_{CDP} (Figure 4.7d), the parameter that defines the shape of the yield surface in the deviatoric plane, produces negligible changes in the results. This is likely because wall behaviour in this case is predominantly governed by tension, with the compressive confinement being relatively low, while k_{CDP} primarily affects the compressive response. When $k_{CDP} = 0.5$, the analysis was prematurely terminated due to convergence issues; this case was not further investigated.

Finally, and in line with the findings of Silva et al. (2021), Behnam et al. (2018), and Rainone et al. (2023), increasing μ_{CDP} (Figure 4.7e), the viscosity parameter which controls viscoplastic regularisation, leads to higher peak strengths and more pronounced

post-peak softening. Lower values of μ_{CDP} result in more localised damage distributions. In this case, results for μ_{CDP} values of 0.0001 and below showed similar responses, while values above 0.001 significantly expanded the damaged area and led to an exponential increase in model strength, ultimately overestimating the wall's actual behaviour.

From these analyses, it is concluded that some plasticity parameters of the CDP model have significant impacts on the output, depending on the modelling technique and loading conditions applied. In this case, both ψ and μ were found to be highly influential, as shown in the parametric studies. The following values were determined for the plasticity parameters, which were used in the subsequent analyses of a masonry façade exposed to tunnelling-induced settlements. These analyses aimed to observe the model's behaviour and assess whether the results remained in good agreement with experimental data. The parameters are set as follows: $\psi_{CDP} = 42$, $e_{CDP} = 0.1$, $\sigma_{0b}/\sigma_{0c} = 1.16$, $k_{CDP} = 0.67$, and $\mu_{CDP} = 0.0001$. These values are not assumed to be universally optimal for all masonry scales; rather, they were calibrated specifically for this study through the previous experimental and numerical agreement. As such, they serve as a validated starting point for future simulations of masonry-like materials using the CDP model. Further adjustments may be necessary depending on the modelling scale and application.

4.3 Wall Exposed to Settlements

Following the modelling of the simple bar (mesh objectivity) and shear wall (CDP calibration), the next step involves testing the model's ability to capture the structural response of a scaled masonry façade subjected to settlements under experimental conditions. Utilising similar modelling procedures as before (dynamic, implicit), the numerical model is validated by examining both global and local responses of the façade and comparing them with experimental results. This ensures the suitability of CDP in accurately capturing the behaviour of large-scale building scenarios exposed to tunnelling-induced subsidence.

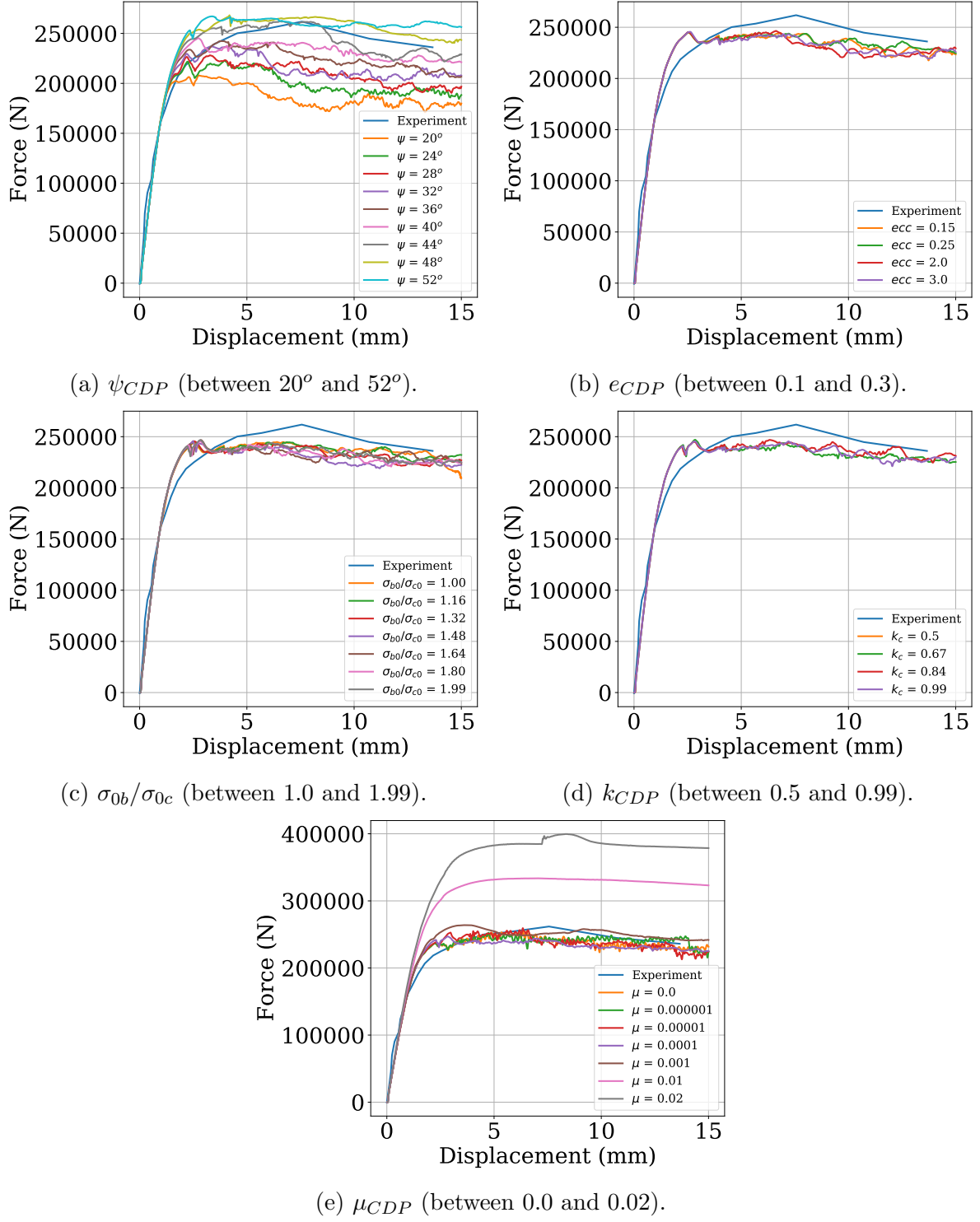


Figure 4.7: Parametric studies on CDP plasticity parameters (a) ψ_{CDP} (b) e_{CDP} (c) σ_{0b}/σ_{0c} (d) k_{CDP} (e) μ_{CDP} .

The experimental wall from Giardina (2013) is used. The model's global dimensions are 1428 mm in width and 1186 mm in height. The wall includes openings for doors measuring 126 mm x 326 mm, as well as several windows measuring 126 mm x 218 mm and 336 mm

x 218 mm for the wider ones. Additionally, small brick-sized openings are introduced at the locations of applied loads to compensate for scaling effects. Linear elastic lintels are inserted above the doors and windows. The base of the structure is initially restricted in both horizontal directions, leaving the vertical direction free to accommodate settlements. The settlement profile applied in this case mirrors that obtained from the experimental SSI results. The material properties of the wall are as follows: $\rho = 1900 \text{ kg/m}^3$, $E = 3000 \text{ MPa}$, $\nu = 0.2$, $f_t = 0.1 \text{ MPa}$, and $G_{f_t} = 0.01 \text{ N.mm/mm}^2$. The compressive behaviour of the material is assumed to be linear elastic, as compressive damage is seldom present due to the material's brittleness in tension (Giardina (2013); Lourenco (1996)). The lintel properties are: $\rho = 500 \text{ kg/m}^3$, $E = 11000 \text{ MPa}$, and $\nu = 0.15$. Details about the model's internal dimensions and the scaling-compensating vertical forces are provided in (Giardina (2013)).

As previously mentioned, the analysis procedure follows recommendations from earlier models. Here, a dynamic implicit analysis is used for the final step (applied settlements), employing the same settings as those used for the shear wall in Section 4.2. The simulation begins with boundary activation, followed by gravity and applied vertical loads. Lastly, 865 mm from the left-hand side of the wall's base boundary in the vertical direction is replaced with the applied settlements, in accordance with the experimental model results.

The settlements are incrementally applied while monitoring points A (at base), B, and C (located near the top-left edge of the wall) for global displacements. Figure 4.8 shows the model geometry, including applied loads, boundary conditions, and settlement profile.

The results in Figure 4.9 illustrate the damage patterns on the façade caused by subsidence. The figure shows the tension damage distribution contours from the FEM model ($d_t = 0.99$ indicating full damage), while also depicting the actual cracks observed in the experimental façade.

Close similarities are observed between the FEM results and the experiment. In particular, the numerical model closely replicates cracks 1, 2, 3, and 4, effectively splitting the

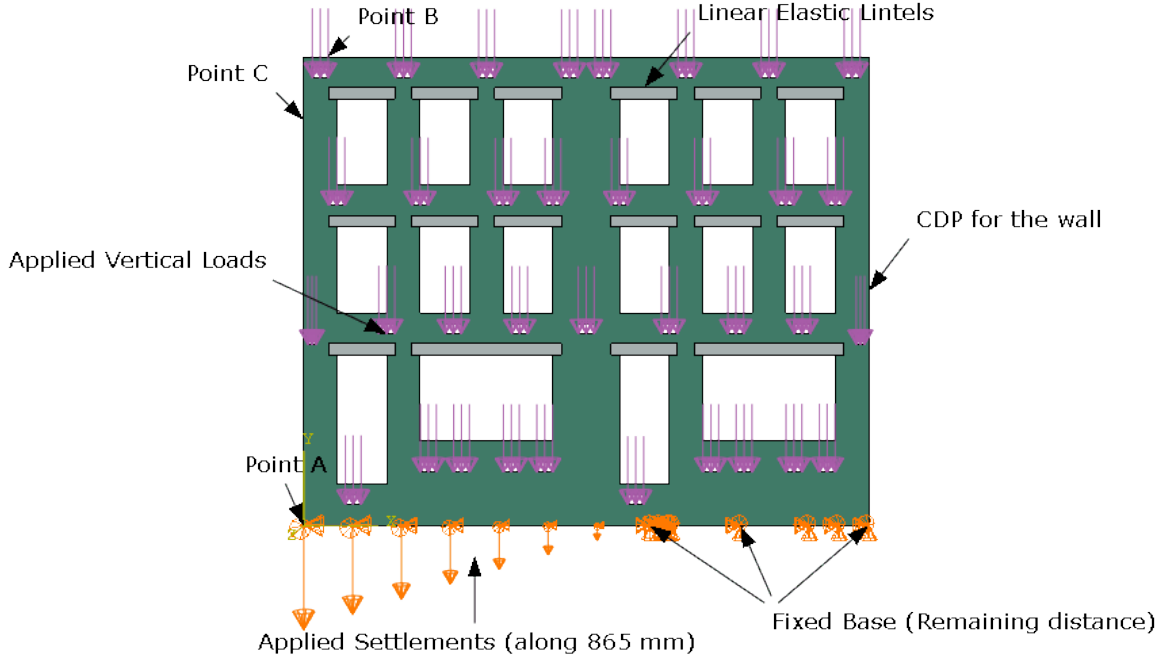


Figure 4.8: Numerical modelling of Giardina (2013) experiment of masonry façade exposed to settlements.

building into approximately two blocks. Crack 6 appeared prematurely and was underestimated, while crack 5 did not develop, likely due to stress redistribution near crack 3, which reduced stress concentrations in the upper right side of the wall. These observations are further quantified in Table 4.2, which compares actual crack widths and global deformations using the equations in Section 2.2.1.

The local aspect of damage (e.g., cracks) is focused on the deformation characteristics along the x-axis only, using scalar representations of ε_t^{ck} to monitor crack activation and measure crack width in a uniaxial direction. Assuming that ε_t along the x-axis is defined as the sum of elastic strain, ε_{0t}^{el} , and cracking strain, ε_t^{ck} , crack activation is then monitored using the criteria in Equation 4.1. Here, ABAQUS computes the strains by extrapolating and averaging the maximum principal ε_t values from the integration points of adjacent mesh elements into the nodes.

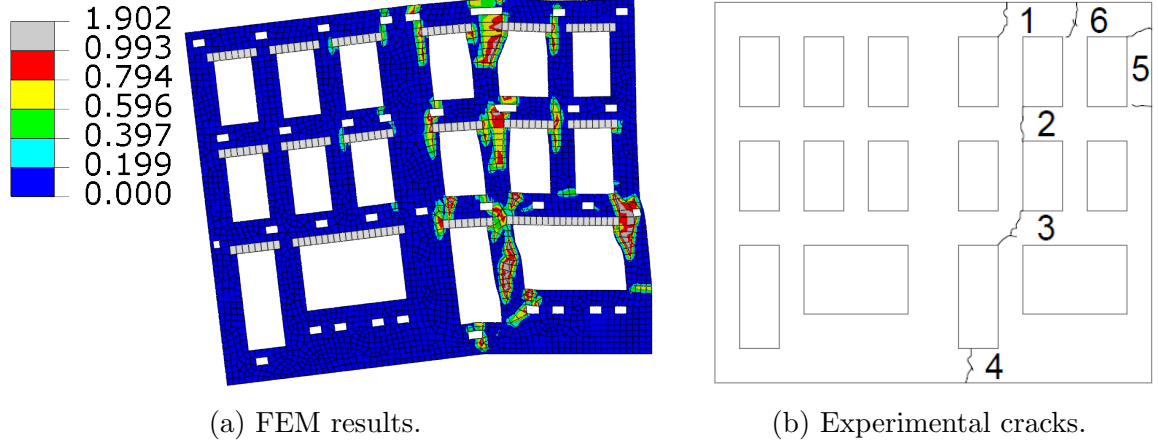


Figure 4.9: FEM simulation output of masonry façade under hogging mode of deformation, illustrating tension damage distributions (d_t).

$$\text{Single Crack Width} = \begin{cases} 0, & \text{for } \varepsilon_t \leq \varepsilon_{0t}^{el} \\ \varepsilon_t^{ck} \cdot l_e, & \text{for } \varepsilon_t > \varepsilon_{0t}^{el} \end{cases} \quad (4.1)$$

The cumulative crack width is then computed as:

$$\text{Total Crack Width} = \sum_i^j \text{Single Crack Width} \quad (4.2)$$

where i and j are the starting and ending indices of the sequence of consecutive values that satisfy the second condition in the case bracket of Equation 4.1.

The cracks are measured along a line of consecutive nodes anywhere in the wall. Crack width is evaluated for each consecutive node according to the condition described above. The total crack count is determined from the total number of sequences satisfying the same condition. In other words, if two elements (i.e., a line with two elements comprising three nodes) satisfy the second condition in the case bracket, then the summed values of both crack widths are considered a single crack. On the other hand, if an intermediate element does not satisfy the condition, then they are considered to be two separate cracks and are evaluated independently. This method has proven to be highly accurate,

as illustrated in this case and reported in Table 4.2.

The global movements of the structure are illustrated in Figure 4.10. Figure 4.10a shows the applied vs. actual monitored settlements between the FEM and the experimental model, also highlighting the location of structural splitting caused by the applied settlement and the appearance of vertical cracks at the piers and spandrels. On the other hand, Figure 4.10b illustrates the horizontal and vertical global displacements at monitoring points. These movements are well captured by the FEM, particularly the horizontal displacements at point C, where several cracks developing along the top edge of the façade contributed to the accumulated global horizontal movement at that location.

Table 4.2: Comparison between FEM and experimental results for the global and local assessments of building damage.

Criteria	Component	FEM Results	Experimental Results	Relative Difference (%)
Global	Top ε_t	0.0052	0.0058	10.3
	Bottom ε_t	0	0	0
	s	0.0067	0.0067	0
	t	0.0107	0.0114	6.1
	β	0.0041	0.0047	12.8
	Horizontal disp. (mm) ¹	10.5	10.5	0
	Vertical disp. (mm)	8.75	7.5	11.9
Local (mm) ²	Crack 1	5.56	5	11.2
	Crack 2	3.45	3.9	9
	Crack 3	0.63/3.48 ³	3.6	59.4/2.4
	Crack 4	1.1	0.85	5
	Crack 5	NA ⁴	1.8	NA
	Crack 6	0.49	3.1	52.2

¹ The divisor here uses the maximum global displacement value of 10.5 mm, both horizontal and vertical displacements are compared using this value.

² Similarly to ¹ the divisor is constant for all cracks based on the larger value (that is, 5 mm).

³ Crack 3 is singularly evaluated and summed of the accumulated cracks along the same height level due to the spread distribution in the FEM model.

⁴ Crack 5 did not develop and instead a new crack at right of crack 3 developed of 1.94 mm width absorbing the damage.

While the current model demonstrates good agreement with experimental results for a façade subjected to hogging-mode settlements, this single scenario alone does not permit generalisation to all tunnel-structure interaction cases. Broader applicability remains limited by the scarcity of comprehensive experimental datasets that include full material

4.4. SOIL-STRUCTURE INTERACTION USING THE EULER BERNOULLI BEAM ON A TWO PARAMETER ELASTIC FOUNDATION CODE

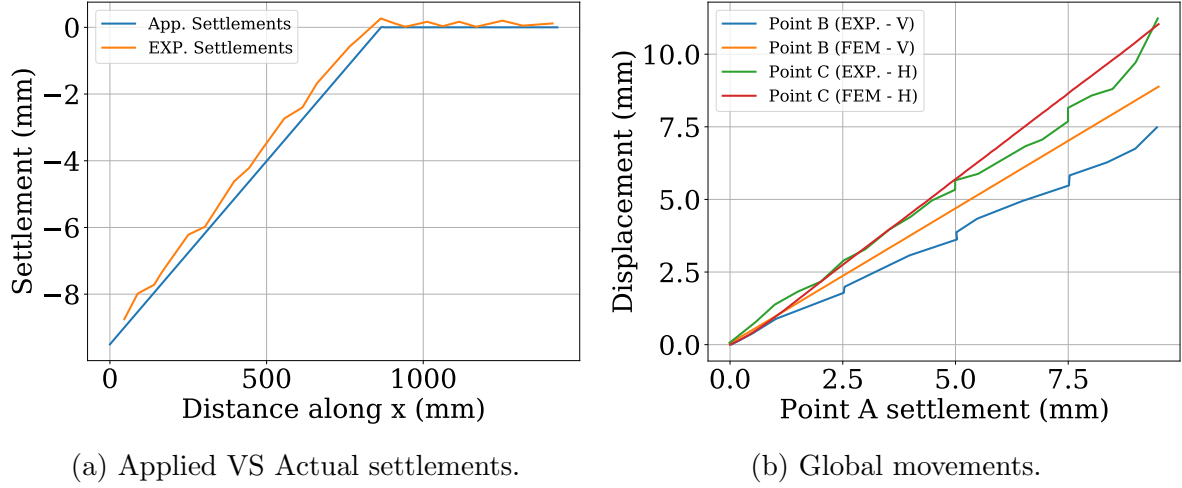


Figure 4.10: Applied settlements and global displacements of the masonry façade FEM vs Experimental.

properties, detailed geometry, and local damage characterisation. For instance, studies such as Son and Cording (2007) lack complete material or geometric details, while Yiu et al. (2017) adopt a perfectly plastic model without simulating damage, making it unsuitable for local damage assessment. Similarly, Frischmann et al. (1994) omits key soil and structural properties. As such, further calibration and validation would be necessary to extend this model to different tunnel-wall interaction scenarios.

4.4 Soil-Structure Interaction using the Euler Bernoulli Beam on a Two Parameter Elastic Foundation Code

The EBBEF2p extends the classical Winkler foundation model by incorporating both vertical and shear subgrade reactions, capturing soil continuity effects more realistically. It is based on the modified Vlasov foundation model, which introduces an additional parameter to account for shear interaction within the soil medium. This formulation leads to a two-parameter foundation model where the soil reaction is expressed not only as a function of local displacement but also of its second derivative, improving the represen-

tation of beam–soil interaction over traditional single-parameter models.

In this thesis, the EBBEF2p model is implemented iteratively to account for beam–soil compatibility under differential settlements, following the method introduced by (Teodoru (2009b)). The beam is discretised using standard finite difference techniques, and the model allows for updating SSI parameters based on relative stiffnesses. This results in an efficient and stable solution method that supports the simulation of a wide range of structural and soil stiffness scenarios with minimal computational demand.

Table 4.3: Input parameters for beam on modified Vlasov foundation, using the example (Teodoru (2009b)). H_s denotes the depth of the soil medium (m).

Euler-Bernoulli Beam				Elastic Foundation		
L (m)	B (m)	H (m)	E (MPa)	H_s (m)	E_s (MPa)	ν_s
20	0.5	1	27000	5	20	0.25

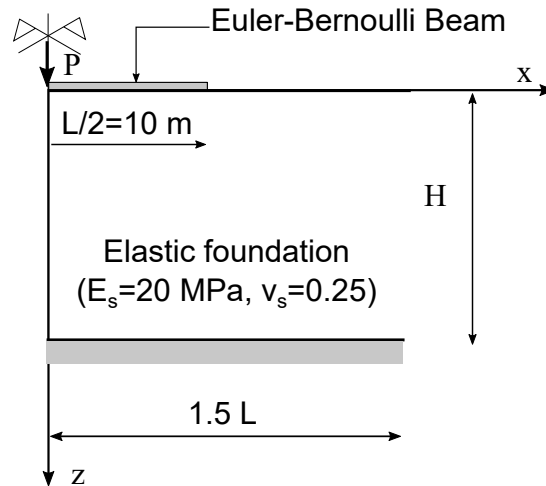


Figure 4.11: Beam on elastic foundation: half-plane geometry and medium property (Teodoru (2009b)).

Figure 4.12 shows the results from implementing EBBEF2p. The green line in Figure 4.12 represents the settlement results (Figure 4.12a), bending moments (Figure 4.12b), and shear forces (Figure 4.12c), compared against Teodoru (2009b)’s benchmark results: the 2D FEM modelling (orange line) and their implementation of the same EBBEF2p computer code (blue line).

The validation example uses the same beam model as in Figure 4.11, with a length $L = 20$ m, supported by an elastic foundation (soil medium), and subjected to a centric

load $P = 500$ kN. As mentioned earlier, due to axial symmetry, only half of the model is considered.

The benchmark results from Teodoru (2009b) show good agreement between the FEM model and their analytical code. The slight discrepancy between them is attributed to the assumption of full displacement compatibility at the beam–soil interface in the 2D model, whereas the Vlasov model assumes compatibility only in the vertical direction (Vallabhan and Das (1991)).

Additionally, the output from the adaptation showed close agreement with the analytical results of Teodoru (2009b), confirming the correctness of the implementation for validation purposes. This verification step not only demonstrates that the use of the EBBEF2p solution is a valid analytical method to replace 2D FEM models in capturing SSI effects accurately, but also highlights its efficiency by significantly reducing computational cost, making it suitable for various simulation needs.

A parametric investigation was also conducted to examine the influence of varying the beam E and soil E_s values on the settlement profile due to SSI using the EBBEF2p code. Figure 4.13 illustrates the changes in the differential settlement of the beam for E values ranging from 27 GPa (as used in the example by Teodoru (2009b)) to 1 MPa (see Figure 4.13a), and for E_s values varying between 200 MPa and 2 MPa (see Figure 4.13b).

The final step involves testing the model under a real-case scenario initially presented by Frischmann et al. (1994), of the Mansion House in London due to the construction of the Dockland Light Railway. In this case, a building is subjected to settlements resulting from excavation related to a railway extension. The building had previously experienced ground movements and cracking associated with earlier tunnelling activities, and the risk of structural damage was deemed unacceptable for the extension. The building is reported to be supported on piled foundations.

Rough estimates for the building’s material properties Frischmann et al. (1994) and ge-

4.4. SOIL-STRUCTURE INTERACTION USING THE EULER BERNOULLI BEAM ON A TWO PARAMETER ELASTIC FOUNDATION CODE

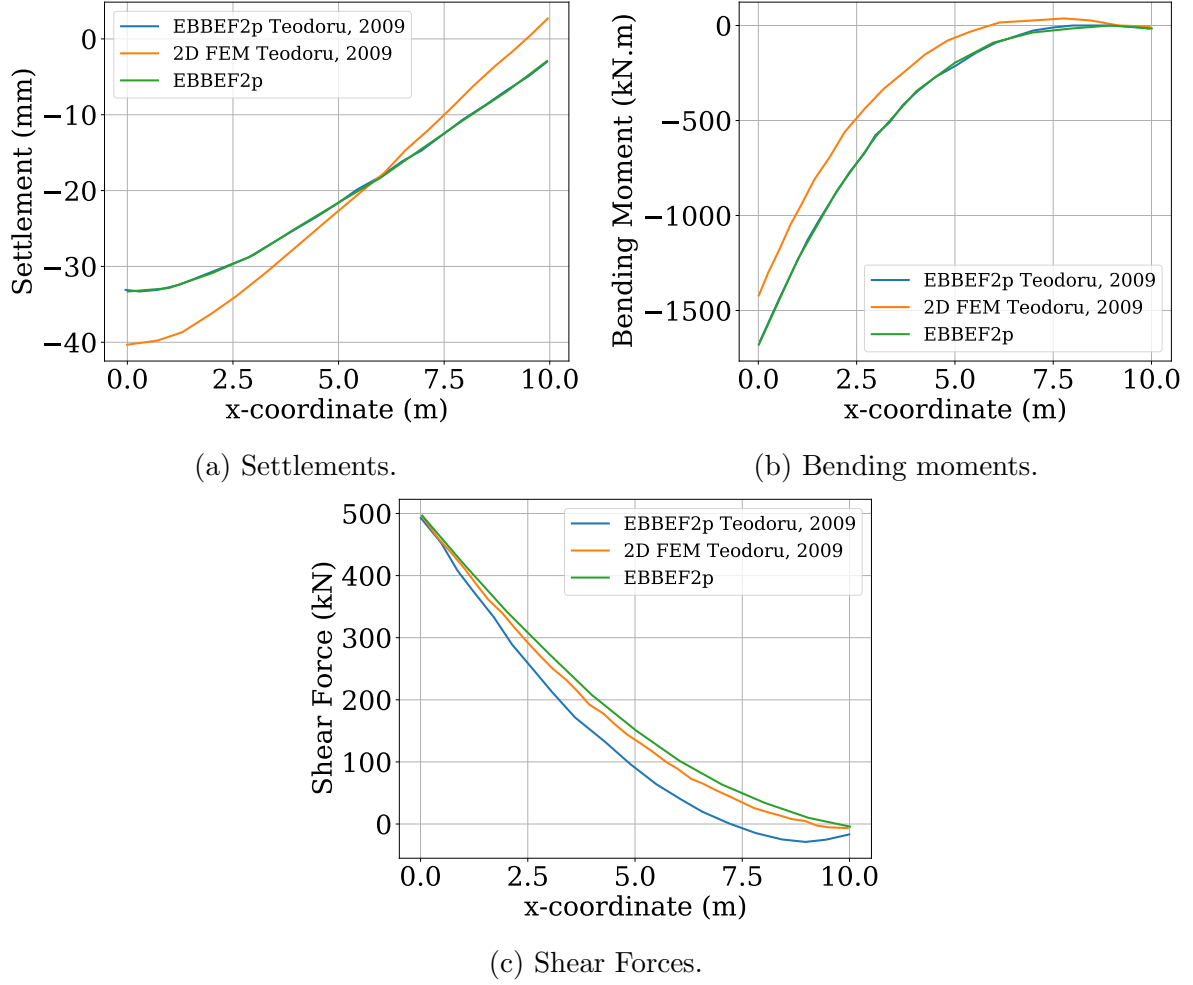


Figure 4.12: Comparing results using the EBBEF2p implementation on benchmark data from (Teodoru (2009b)).

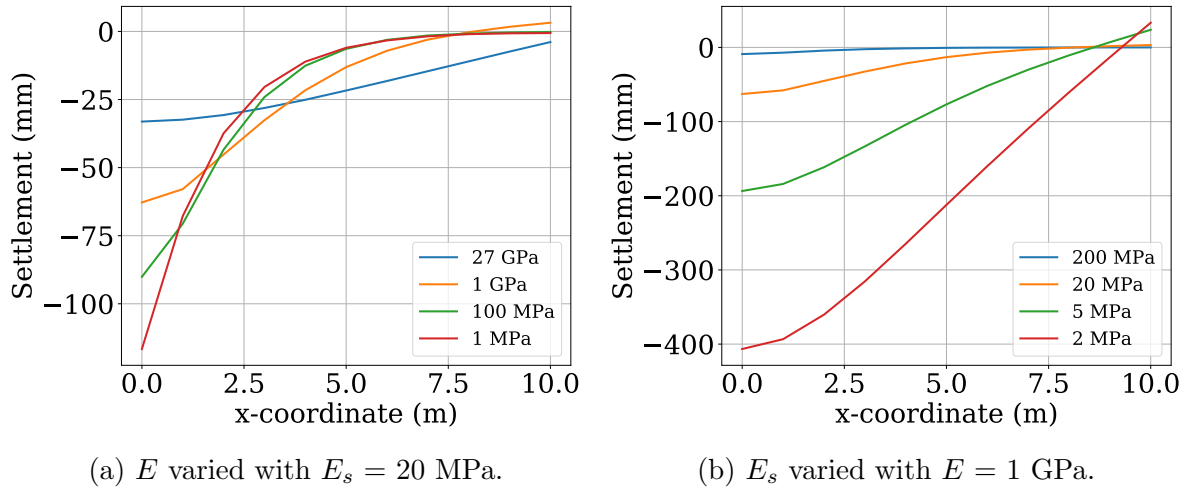


Figure 4.13: Comparison of settlement profiles with varying parameters values using the EBBEF2p code (a) E varied (b) E_s varied.

ometrical configuration Selby (1999) are provided and summarised below. It is equally important to note that the EBBEF2p model evaluates behaviour only within the linear elastic domain for both the soil and the building. For this purpose, the building's Young's modulus E is taken as 1 GPa, and Poisson's ratio ν as 0.2. The tunnel diameter D is 3.05 m, and the VL parameter range was approximated to be between 2% and 2.5% (with 2% selected). The depth to the tunnel axis Z_0 is 15 m, and the trough width parameter K is 0.334, which is considered quite narrow according to literature (see Section 2.1.1). The building length L is 50 m, and height H is 20 m.

As no detailed information is provided about the soil, Selby (1999) used an E_s of 50 MPa and ν_s of 0.49 for the same problem, indicating that these parameters are appropriate for an undrained analysis of stiff clay. Since a wall is being analysed, no openings are assumed. Lastly, the building is located at $e = 15$ m from the tunnel's centreline.

Regarding parameters related to the EBBEF2p code, the maximum number of iterations was set to 10, a medium (soil) depth (H_s) of 5 m, and an initial estimate of $\gamma = 0.5$. The settlement predictions for the Mansion House are presented in Figure 4.14.

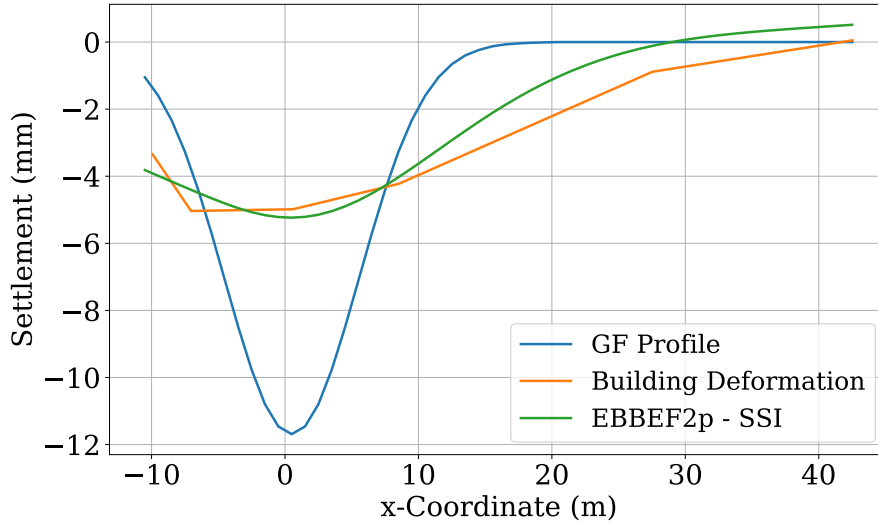


Figure 4.14: Actual and predicted settlements of the west wall due to tunnelling.

As observed in Figure 4.14, the settlement underneath the building is far more rigid than the initial GF settlements imposed if no building was placed. The numerical code

EBBEF2p converged after only two iterations with final values of $\gamma = 0.036$. The model demonstrated good agreement with field observations of ground movements at the foundation level. It is noted that a spring was introduced at the location of the left edge of the wall to compensate for the wall's presence in the in-plane direction. The amount of force was calibrated according to the closest possible similarity to given data as per (Frischmann et al. (1994)).

Several alternative methods exist for assessing SSI in tunnel–building problems, including single-parameter Winkler models, cracked beam models Acikgoz et al. (2021), design curves based on 2D numerical analyses Potts and Addenbrooke (1997); Franzius et al. (2004), and full 2D or 3D FEM simulations (Yiu et al. (2017); Cheng et al. (2007)). These methods were previously reviewed in Section 2.3. While advanced numerical models can offer detailed insights, particularly under highly non-linear or cracked conditions, they are computationally intensive and often impractical for extensive parametric or ML-based studies requiring hundreds of simulations.

The EBBEF2p model, on the other hand, provides a practical balance of accuracy and efficiency in the linear elastic domain. Its closed-form and iterative formulation allows for rapid settlement predictions while capturing essential SSI effects, making it particularly suitable for the large-scale simulation demands of this thesis. However, as discussed further below, it comes with limitations that should be considered when applying to more complex or damaged structural states.

4.4.1 Advantages, Limitations, and Implications of the EBBEF2p Model

The primary advantage of the EBBEF2p model lies in its efficiency. Compared to advanced 2D or 3D FEM approaches, this method significantly reduces computational time and memory demand, making it particularly suitable for large-scale parametric studies

and ML applications where hundreds or thousands of simulations may be required. It also offers reliable results within the linear domain, and, through its iterative nature, captures key stiffness interaction effects between soil and structure. In the FEM implementation used here, the model further allows for displacement compatibility between the structure and the ground surface, enabling partial decoupling that facilitates the observation of expansion and gap formation.

Nonetheless, the model has several limitations. It operates within a linear elastic framework and does not account for material nonlinearity, cracking, or plasticity effects. As such, it cannot simulate post-cracking behaviour or structural segmentation, effects that are often critical when buildings undergo significant settlement-induced damage. Additionally, the SSI is treated in a decoupled manner; ground settlements computed from elastic soil models are applied independently of the structural response, which may lead to inaccuracies when nonlinear interaction dominates. This also assumes negligible influence from neighbouring structures, which may not hold in dense urban settings.

Compared to more advanced modelling approaches such as full 2D/3D FEM or continuum-based damage models, the EBBEF2p solution trades off complexity for speed and generality. While this makes it unsuitable for high-fidelity modelling of damage mechanisms, its strengths lie in efficiently capturing first-order effects of SSI, especially in the early design stages or for probabilistic and data-driven assessments. Users of the model must therefore remain aware of its assumptions and avoid over-interpreting results in regimes where material or geometric nonlinearity becomes dominant.

4.5 Summary

This chapter fulfils the requirements of the first objective (Objective 1 in Section 1.5) by demonstrating the use of FEM models to simulate the non-linear response of masonry. It validates material properties, both global and local structural behaviour, and SSI through

the CDP and EBBEF2p models. The chapter is summarised in the following bullet points:

- A thorough investigation of mesh-independence studies via h -refinement was conducted using a simple bar in tension with a reduced-strength section at the centre. The study showcased different behaviours using: i) an unregularised stress–strain law, ii) a regularised stress–strain model, and iii) a stress–displacement model.
- A shear wall subjected to combined loading conditions was investigated using non-linear material properties and the concept of h -refinement. Additionally, a dynamic implicit step was employed. F-D curves were plotted, and extensive parametric studies were performed on the plasticity parameters of the CDP model.
- Numerical validation of a wall subjected to settlements was performed, including global displacements, local damage (developed during the simulation), and both local and global assessment metrics of building damage. The results were compared between the FEM and the experimental data, demonstrating good agreement between the two.
- Analytical VV of the

In conclusion, this chapter demonstrates the successful implementation and validation of numerical and analytical models capable of simulating the complex behaviour of masonry structures under settlement-induced damage. Through a combination of detailed finite element analyses using the CDP model and efficient elastic foundation modelling via the EBBEF2p framework, a robust platform has been established for assessing both local and global building responses. These models have been validated against experimental and benchmark cases, providing confidence in their use for generating reliable synthetic data. While certain limitations, such as the linear assumptions in the EBBEF2p model, remain, the methods presented here strike an effective balance between accuracy and computational feasibility, especially when considering the large-scale parametric and data-driven aims of this thesis. The next

chapter builds on this foundation to systematically generate a synthetic dataset for ML applications, drawing on the modelling approaches verified here.

Chapter 5

Generation of Synthetic Dataset

It is important to develop a robust and rich dataset to minimise errors and ensure reliable predictions in ML models. The quality of input, particularly in terms of quantity and balanced output distributions, has a significant influence on model performance, as will be demonstrated later in Chapter 7. In this section, the procedure for generating a synthetic dataset using FEM is presented, incorporating the previously Verification and Validation (VV) Concrete Damaged Plasticity (CDP) and Euler Bernoulli Beam on Two Parameter Elastic Foundation (EBBEF2p) models, for tunnelling-induced building damage. The process begins with the identification of input parameters. Since empirical, analytical, and numerical models are combined, only parameters related to the components of the selected modelling procedure are targeted.

For the prediction of settlements, the synthetic dataset is built using the VL method, as it requires fewer input parameters than the Chakeri & Onver method (COM) method for example. This allows a more detailed focus on building-related parameters such as geometry and material properties. Although it is possible to include both settlement methods, doing so significantly increases computational costs, approximately proportional to the number of input variables. The VL method requires four parameters to derive settlements, while the COM method requires ten. Us-

ing the latter would increase the dataset size by approximately 2.5 times, due to the inclusion of several additional variables, particularly when focusing on building parameters (given the wide adoption of this method). As the exact number of samples needed is currently unknown, and to remain within a realistic timeframe and available computational resources, a compromise was necessary. Therefore, the VL method was selected as the basis for settlement modelling. The overview structure of this chapter is illustrated in Figure 5.1.

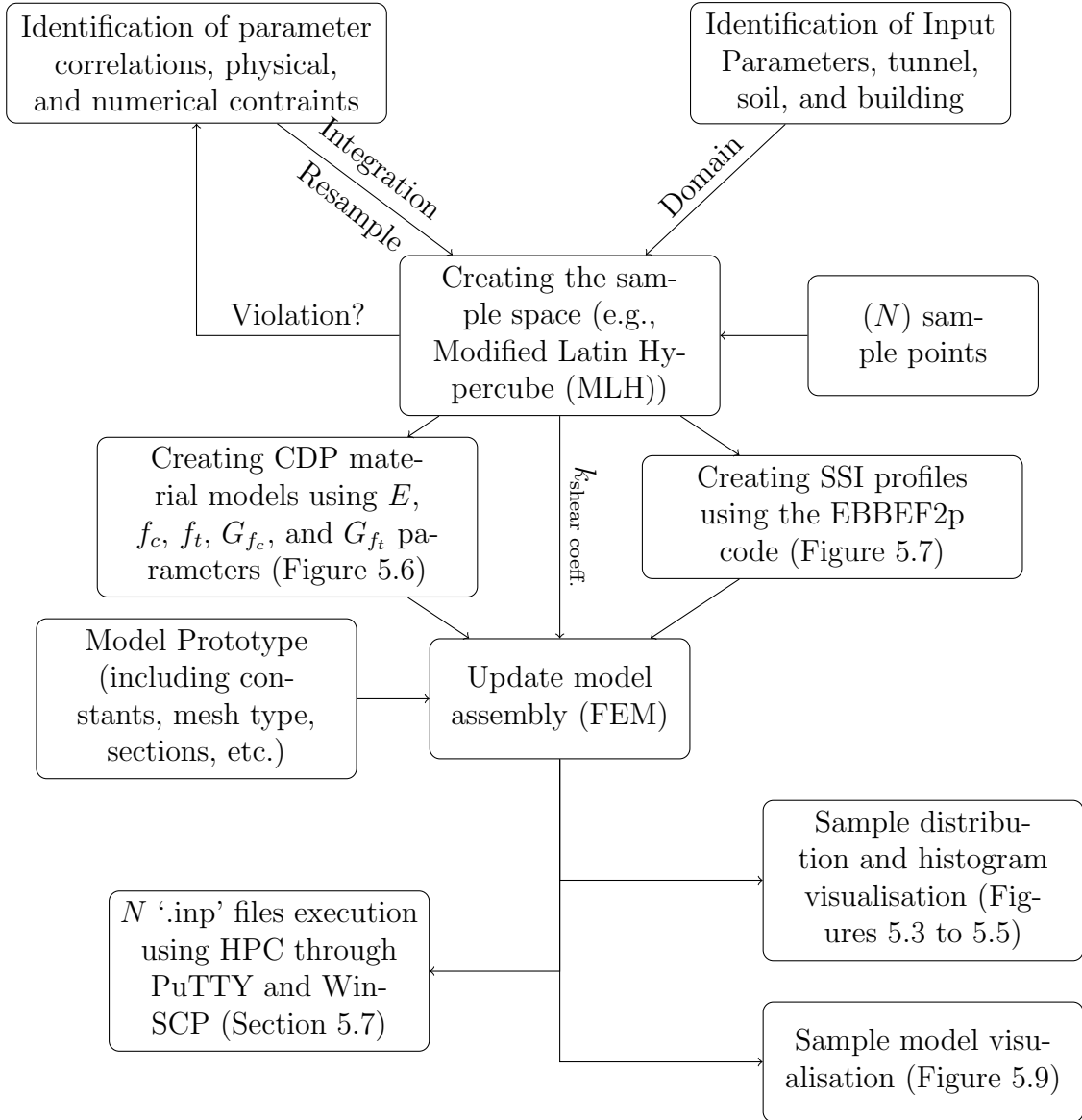


Figure 5.1: Generating synthetic dataset methodology overview.

5.1 Identification of Input Parameters

Figure 5.2 provides a schematic overview of the tunnel-soil-structure system and visually depicts the key input parameters presented in the following tables.

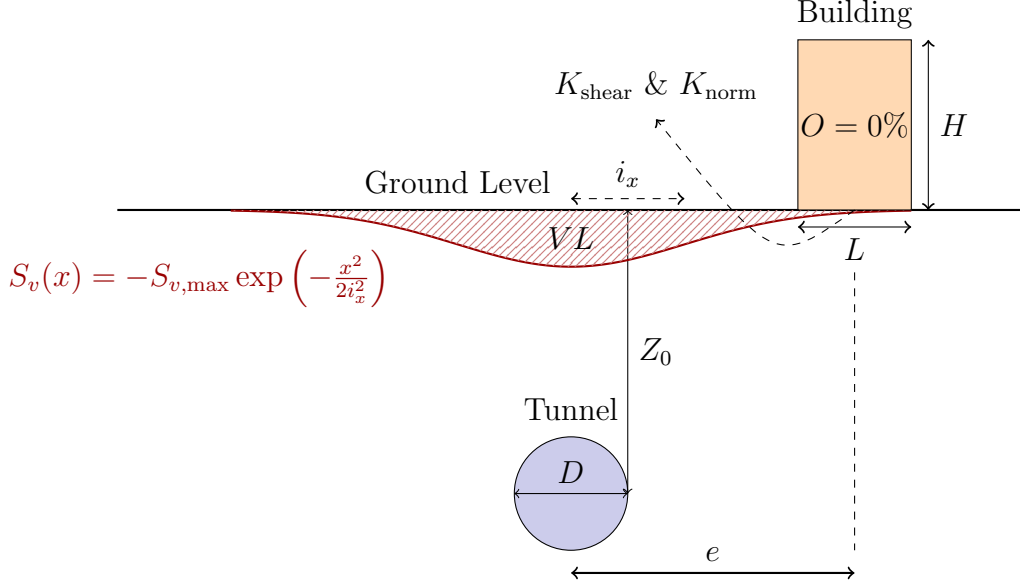


Figure 5.2: Schematic of tunnel-SSI showing key input parameters including tunnel depth (Z_0), diameter (D), offset (e), building height (H), length (L), and settlement trough width (i_x). The settlement profile follows a Gaussian curve (Equation 2.1). Hatched area indicates (VL). No openings are considered in the façade ($O = 0\%$). The soil-structure interface is represented by the normal and shear stiffness parameters (K_{norm}) and (K_{shear}), respectively.

Table 5.1 presents a list of the identified parameters related to the VL method, including their minimum and maximum values, as well as the increments used. These inputs are derived from a combination of experimental studies, numerical simulations, and field observations from previous tunnelling projects. As mentioned earlier in Section 2.1, the soil type assumed for these analyses is homogeneous and behaves linearly.

Providing these values (domains and increments) allows the identification of feasible value boundaries and offers insight into previous investigations and the observations made within these limits. These values also help in determining suitable domains for appropriate sampling procedures. It is further observed whether finite or discrete increments were used, either due to experimental constraints or numerical

Table 5.1: Input parameters relating to the VL method.

Parameter	Symbol (Unit)	Domain	Increment	Source
Volume Loss	VL (%)	0.2 — 3	$\approx^a 0.5$ — $^a 1$	(Mair et al. (1996))
		0.32 — 5	≈ 0.2 — 0.5	Marshall et al. (2012)
		0.39 — 3.3	≈ 0.05 — 0.5 - 1	Burland et al. (2001b)
Cover/Diameter	C_T/D (-)	—	1.3, 2.4 and 4.4 ^b	Marshall et al. (2012)
Depth	Z_0 (m)	—	20 and 34 ^b	Potts and Addenbrooke (1997)
		9 — 24 _(To Crown) ^c	≈ 1 — 6	Burland et al. (2001a)
		15 — 37	≈ 2 — 3.5	Burland et al. (2001b)
		—	90	High Speed Two (HS2) Ltd (2017)
		4.85 — 11.8	≈ 0.75	Burland et al. (2001b)
Diameter	D (m)	—	17.63	Chan et al. (2021); Bao et al. (2021)
Inflection Point	i_x (m)	6.35 — 16	≈ 0.4 — 4.5	Burland et al. (2001b)

^a \approx and — symbols represent the approximate values and investigated range, respectively.

^b These are specific values used in the analysis, not increments.

^c “To Crown” refers to the distance from ground surface to tunnel crown, which corresponds to C_T .

efficiency requirements, and an assessment is made regarding whether reducing the increments between values using the synthetic strategy would result in any benefit (e.g., capturing non-linear relationships) or not.

Similarly, Table 5.2 lists the input parameters for the soil medium used in the EBBEF2p model. While these parameters are sufficient to describe a linear response, they were considered appropriate for establishing a SSI model that captures some degree of non-linear behaviour, as previously demonstrated in the Mansion House case study in Section

Furthermore, Table 5.3 contains a list of building material property parameters.

Table 5.2: Input parameters related to soil while using the EBBEF2p method

Parameter	Symbol (Unit)	Domain	Increment	Source
Trough Width Par. (At Surface Level)	K (-)	—	0.25 and 0.5	Mair et al. (1996)
		0.25 — 0.63	≈ 0.2 — 0.5	Marshall et al. (2012)
		0.45 — 0.53	≈ 0.02 — 0.03	Burland et al. (2001b)
Deformation Modulus	E_s (MPa)	2 — 250	≈ 10 — 150	Bowles (1996)
Poisson's Ratio of Soil	ν_s (-)	0.1 — 0.5	≈ 0.05 — 0.2	Bowles (1996)

Parameters E , f_c , f_t , and G_{f_t} are included, all of which contribute to developing the CDP model. Other parameters, including Poisson's ratio (ν), are not included. To the best of the author's knowledge, no investigations have been conducted into the influence of ν on global building behaviour using Macro-modelling. Therefore, a constant value of 0.2 is used throughout. As buildings exposed to settlements are more prone to fail in tension Lourenco (1996); Giardina (2013), the material compressive fracture energy (G_{f_c}) is not explicitly considered in this study. Instead, G_{f_c} is taken as a function of the material's compressive strength (F_c), as given by (Ghiassi and Milani (2019)):

$$G_{f_c} = 15 + 0.43 f_c - 0.0036 f_c^2 \quad (5.1)$$

Similarly, the density of the material (ρ), is taken from Giardina (2013), and is given as 1900 kg/m³.

Additionally, Table 5.4 contains a list of input parameters related to building geometry, including the building length (L), height (H), and the percentage of openings (O). It is important to note that in this context, openings refer to those created by the presence of windows and doors. In modelling these parameters, variation in the percentage of openings reflects changes in spacing between openings rather than

Table 5.3: Input parameters related to the building material using the CDP model.

Parameter	Symbol (Unit)	Domain	Increment	Source
Elastic Modulus	E (MPa)	—	1000, 3000 and 9000	Giardina et al. (2015)
		666 — 6035	≈ 100 — 3000	Pluijm (1992)
Compressive Strength	f_c (MPa)	0.5 — 8	≈ 0.1 — 1	Ghiassi and Milani (2019)
		2.3 — 48.2	≈ 3 — 5	Kaushik et al. (2007)
Tensile Strength	f_t (MPa)	—	0.1, 0.3 and 0.9	Giardina et al. (2015)
		0.05 — 0.58	≈ 0.01 — 0.1	Schubertl (1994)
		0.13 — 1.47	≈ 0.1 — 0.5	Pluijm (1992)
Tensile Fracture Energy	G_{ft} (N.m/m ²)	—	10, 20, and 10000	Giardina et al. (2015)
		4 — 30	≈ 3 — 8	Pluijm (1992)
Axial Stiffness ^a	EA (kN/m)	3.450×10^6 — 1.725×10^7	3.450×10^6	Potts and Addenbrooke (1997)
Bending Stiffness ^a	EI (kN.m ² /m)	2.208×10^8 — 1.553×10^{10}	2.208×10^8 and 3.105×10^9	Potts and Addenbrooke (1997)

^a The axial and bending stiffness values are derived from the contribution of the slab to framed structures of 1 to 5 floors in height. Although framed structures are beyond the scope of the current study, their inclusion provides valuable insights for future applications considering frame buildings.

changes in their size, as was the case in (Giardina (2013)). This approach ensures more realistic modelling of buildings with varying O values. The dimensions of the openings are defined as 1200 mm \times 1500 mm for windows and 1200 mm \times 2000 mm for doors. Furthermore, as wall thickness was found to have negligible effects in both the EBBEF2p and numerical models when assessing in-plane behaviour, it is assumed constant at 250 mm.

Lastly, Table 5.5 provides input parameters related to the contact area between the building foundation and the soil surface, including the parameter e , which defines the offset from the tunnel centreline. In this investigation, buildings are assumed to be supported by shallow strip footings, which simplify the modelling of SSI (Giardina et al. (2015); Yiu et al. (2017); Cao et al. (2022)). Accordingly,

Table 5.4: Input parameters relating to building façade geometry (2D).

Parameter	Symbol (Unit)	Domain	Increment	Source
Percentage of Openings	O (%)	0 — 30	10	Son and Cording (2007); Giardina et al. (2015)
		16 — 60	≈ 5 — 10	Potts and Addenbrooke (1997)
Length	L (m)	12.5 — 40	≈ 0.5 — 7	Burland et al. (2001b)
		8 — 29	≈ 1 — 8	Charles and Skinner (2004)
Height	H (m)	12 — 20	4	Burland et al. (2001a)
		—	16	Burland et al. (2001b)

only the vertical and horizontal resistances, represented by the normal and shear friction coefficients, (K_{norm} and K_{shear} or $K_{\text{shear coeff.}}$), are considered.

Table 5.5: Input parameters related to the contact area between building and soil.

Parameter	Symbol (Unit)	Domain	Increment	Source
Interface Shear Behaviour	K_{shear} (MPa)	—	Smooth and Rough ^a	Giardina et al. (2015)
		0.33 — 0.6	≈ 0.1	SupercivilCD
		—	0.3	Yiu et al. (2017)
Interface Normal Behaviour	K_{norm} (N/m ³)	0.7×10^7 — 0.7×10^9	Domain $\times 10$	Giardina et al. (2015)
Position	e (m)	—	Sagging and Hogging	Giardina et al. (2015)
		0 — 28	≈ 4 — 12	Potts and Addenbrooke (1997)

^a Smooth and Rough interactions can otherwise be defined as frictionless and semi-/total adherence with the ground.

To summarise the relevant parameter space described in Tables 5.1 to 5.5, the complete set of input variables used to generate the synthetic dataset via FEM

modelling is presented in Table 5.6. In addition to ranges, this table indicates the type of incrementation used: either continuous (finite) or discrete. Although all parameters are initially sampled as continuous, they are later adjusted for implementation in FEM, requiring specific increments. For example, an increment of 175 mm is used for L to ensure consistent mesh sizing across the wall, foundation, and soil layers. A smaller increment of 5.5% is selected for O , as values below this threshold caused errors in the model, preventing successful simulation. These errors arise during the automated generation of facade openings: because door and window dimensions are fixed, any reduction in O below approximately 5.5% results in insufficient wall space to accommodate even a single door and window combination, which causes the script to fail during geometry creation. The height parameter (H) is constrained between 2.8 m and 4.2 m to reflect typical residential or commercial building heights and to ensure compatibility with the EBBEF2p model.

Table 5.6: Summary of input parameters in FEM implementation.

Parameter	Min. Value	Max. Value	Increment (As used in FEM)
E (MPa)	666.0	9000.0	Continuous
f_t (MPa)	0.05	1.47	Continuous
G_{ft} (N.mm/mm ²)	0.004	0.03	Continuous
H (m)	3.2	20.0	2.8 to 4.2
L (m)	8.0	60.0	0.175
O (%)	0.0	30.0	5.5
e (m)	0.0	45.0	Continuous
E_s (MPa)	2.0	250.0	Continuous
ν_s (-)	0.1	0.49	Continuous
i_x (m)	3.6	45.0	Continuous
K_{shear} coeff. (-)	0.0001	0.6	Continuous
VL (%)	0.2	5.0	Continuous
Z_0 (m)	7.3	90.0	Continuous
D (m)	4.9	17.6	Continuous
f_c (MPa)	0.5	48.2	Continuous

During the preliminary investigation, it was found that several input parameters exhibit strong correlations, with some parameters acting as dependent variables and others as independent. These relationships were particularly evident in material properties of masonry Ghiassi and Milani (2019), the influence of soil type

on settlement troughs Mair et al. (1993), and the influence of offset distance relative to tunnel diameter (D) (Peck (1969)). These dependencies are summarised in Table 5.7, which maps dependent variables to their associated independent parameters. In addition to these dependencies, physical constraints Mair et al. (1993) and the requirement for numerical stability in CDP models Lourenco (1996); Feenstra (1993) are critical considerations. These aspects must be addressed when modelling scenarios involving multiple interdependent parameters. Table 5.8 outlines these conditions along with their corresponding mathematical formulations.

Table 5.7: Correlations between dependent and independent parameters.

Parameter	Symbol (Unit)	Dependency	Formula	Source
Trough Width	i_x (m)	K and Z_0	$K \times Z_0$	Mair et al. (1993)
Elastic Modulus	E^* (MPa)	f_c	$(300 - 700) f_c$	Ghiassi and Milani (2019); Kaushik et al. (2007)
Tensile Fracture Energy	G_{f_t} (N.m/mm ²)	f_t	$(0.005 - 0.05) f_t$	Ghiassi and Milani (2019); Giardina et al. (2015); Hillerborg et al. (1976)
Distance	e (m)	D	$(0 - 3) D$	Peck (1969)

* Parameter E can also be indirectly and/or directly dependent on parameter f_t as well, as f_t was found to range between (10-20)% the value of f_c (Ghiassi and Milani (2019)).

In Table 5.7, the dependency of the trough width parameter (i_x) on Z_0 via the variable K is determined from several field measurements by plotting i_x against Z_0 , the depth of the tunnel axis below ground level. This relationship was originally proposed as $i_x = 0.43 Z_0 + 1.1$ O'Reilly and New (1982), and, for practical purposes, is often approximated as $i_x = K \times Z_0$ (Mair et al. (1993)). Dependencies of building material properties such as E and G_{f_t} are established through observations of several experiments on individual masonry components or combined units under various loading conditions. These studies provide a range of possible values for each parameter as a function of f_c and f_t . In many cases, f_t can also be expressed as a

function of f_c , such that $f_t = (10\text{--}20)\% f_c$ (Ghiassi and Milani (2019)). However, this correlation is not included in the current work since the focus is primarily on the behaviour of building material in tension. Lastly, Peck (1969) observed that parameter e values exceeding $3 \times D$ result in negligible soil movements around tunnel excavations. Therefore, structures located beyond this limit are assumed to remain unaffected by tunnelling activities.

Table 5.8: Constraints based on physical and numerical integrity.

Component	Condition (Unit)	Formula	Source
Physical	Maximum Settlement ^a , $S_{v,\max}$, in GF (mm)	$\sqrt{\frac{\pi}{2}} \cdot \frac{VL \cdot D^2}{4i_x} < 60$ or 150	Mair et al. (1993); Burland et al. (2001b)
	Minimum Overburden Height (m)	$Z_0 > 1.5 \times D$	Mair et al. (1993)
	Elastic Modulus ^b , E , (MPa)	$E < 9000$	Giardina (2013)
	Tensile Strength ^c , f_t , (MPa)	$f_t \geq \frac{E}{10000}$	Hillerborg et al. (1976)
Numerical	Characteristic Length, h , (mm)	$h \leq \frac{k_{soft} \cdot G_{f_t} \cdot E}{f_t^2}$	Feenstra (1993)
	Snap-Back in Compression (–)	$\varepsilon_m \geq \frac{f_c}{E} + \varepsilon_p$	Lourenco (1996)

^a In Mair et al. (1993), it is reported that some tunnelling processes induced $S_{v,\max}$ of up to 150 mm.

^b E is given an upper limit based on Giardina et al. (2015) to avoid scenarios in which higher stiffness contributes to negligible differential settlement at the foundation level.

^c f_t is restricted using the limit of Hillerborg et al. (1976), avoiding misinterpretation of crack initiation of buildings with high E and low f_t .

5.2 Creating the Sample Space

With the input parameters and their corresponding properties familiarised, the sample space is generated using the information in Table 5.6, accounting for correlations and constraints described in Tables 5.7 and 5.8. For an effective sampling distribution, the LH method is adopted, as it maximises the exploration of the search space while requiring fewer samples than traditional techniques, an important advantage when facing significant computational demands (Saltelli et al. (2008)). To this end,

the SALib library in Python is used (Herman et al. (2022)). Scripting is initiated by importing the necessary modules, such as the `Latin` module. The initial code structure is detailed in Listing A.1.

The code in Listing A.1 is populated with the number of parameters, `num_vars` (i.e., 15), a list of `names` (e.g., E , f_t , G_{f_t} , etc.), and the corresponding `bounds` for each parameter, as provided in Table 5.6.

Functions are then defined to represent the correlations between dependent and independent variables, based on the information in Table 5.7, and implemented into the code using the code provided in Listing A.2.

A random value is selected between `low` and `high`, as given in Table 5.7, and is multiplied by the independent parameter to compute the value of the dependent parameter. Since four correlations were previously defined, this process is repeated four times, once for each dependent parameter.

Functions are also defined to implement the physical and numerical constraints described in Table 5.8. An example is provided in Listing A.3.

This example demonstrates how to define the maximum allowable settlement threshold (e.g., $S_{v,\max}$), ensuring that no combination of input parameters violates this value (e.g., 60 mm or 150 mm). Similar logic is applied to all other constraints listed in Table 5.8, with the `Numpy` library used to handle the required mathematical operations. With the parameter ranges, correlations, and conditions defined, the LH method (specifically, the MLH variant) is applied to generate N samples. Any sample that violates the predefined conditions is filtered out using Listing A.4.

The code then checks each generated sample using an `if` statement. If a violation is detected, the `else` clause is executed to produce a replacement sample. This ensures that all samples in the final dataset comply with the required constraints.

Through the MLH method, a sample space is obtained that respects both parameter correlations and physical/numerical conditions. The following plots illustrate

the distribution of $N = 500$ sample points generated using this method. Figure 5.3 shows the distribution of material parameters, along with corresponding histograms on the diagonal. The bounds for each sample are scaled by their associated independent parameter, as defined in Table 5.7. Consequently, a larger value of the independent parameter produces a wider possible range for the dependent parameter, and vice versa. If correlations were ignored, the histograms would show a more uniform distribution, as is typical of standard LH sampling. Figure 5.4, on the other hand, presents the distributions of tunnel and soil parameters, which are mainly constrained by the conditions in Table 5.8. These include physical or numerical limits that indirectly shape the sample distributions. One exception is e , a dependent parameter based on D , where the upper bound of e is restricted as D increases. This reflects the assumption that values of e beyond a certain threshold are non-critical.

Finally, Figure 5.5 illustrates the distributions of other building and soil parameters that are not directly governed by any defined correlation or constraint. These appear more evenly distributed, though minor unevenness may result from the rejection of samples that indirectly violated conditions. Across all three figures, the diagonal histograms help identify the number of values sampled for each parameter, offering insight into regions with dense or sparse data. The sampled values are stored in plain ‘.txt’ files and are used to generate the various components of the model, including the CDP material model, the EBBEF2p-based SSI settlements, and parametric adaptations of the prototype model (Model 0).

5.3 Generating Concrete Damaged Plasticity Material Models

The CDP material models are now developed using relevant parameters from the sample points generated in the MLH sample space, for a total of N samples. To

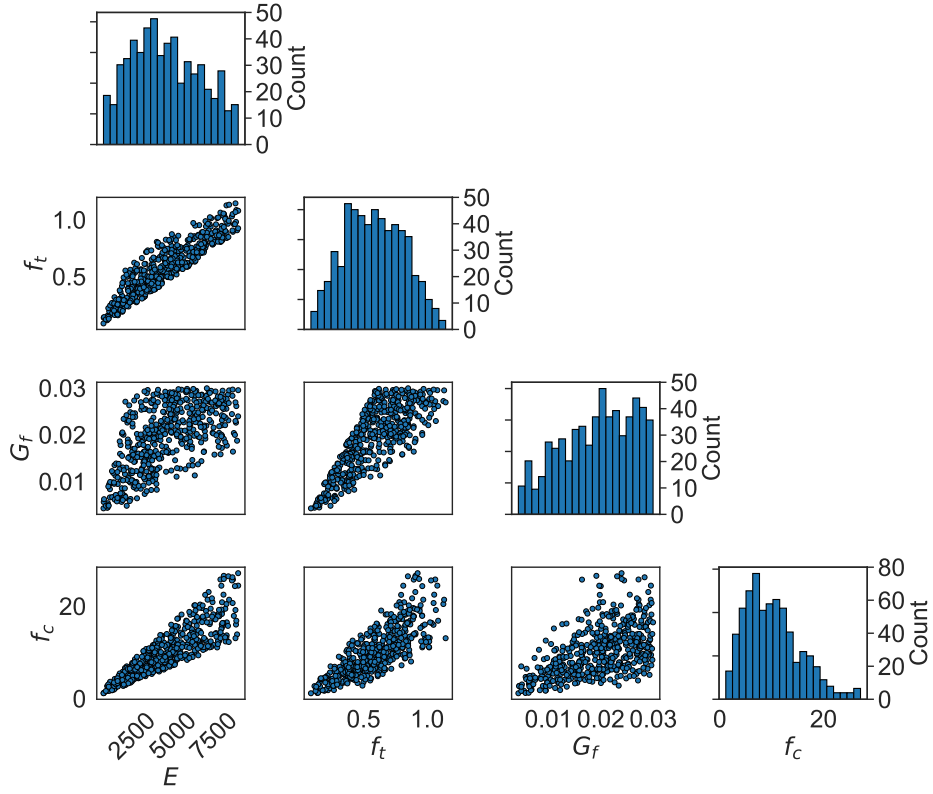


Figure 5.3: Sample distributions of material parameters.

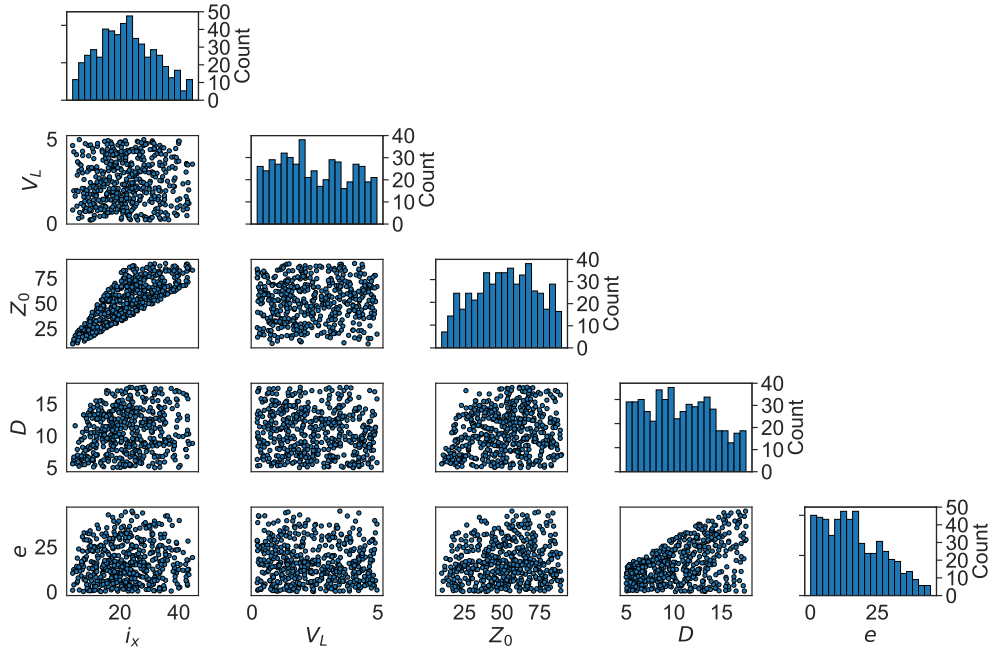


Figure 5.4: Sample distributions of soil and tunnel parameters.

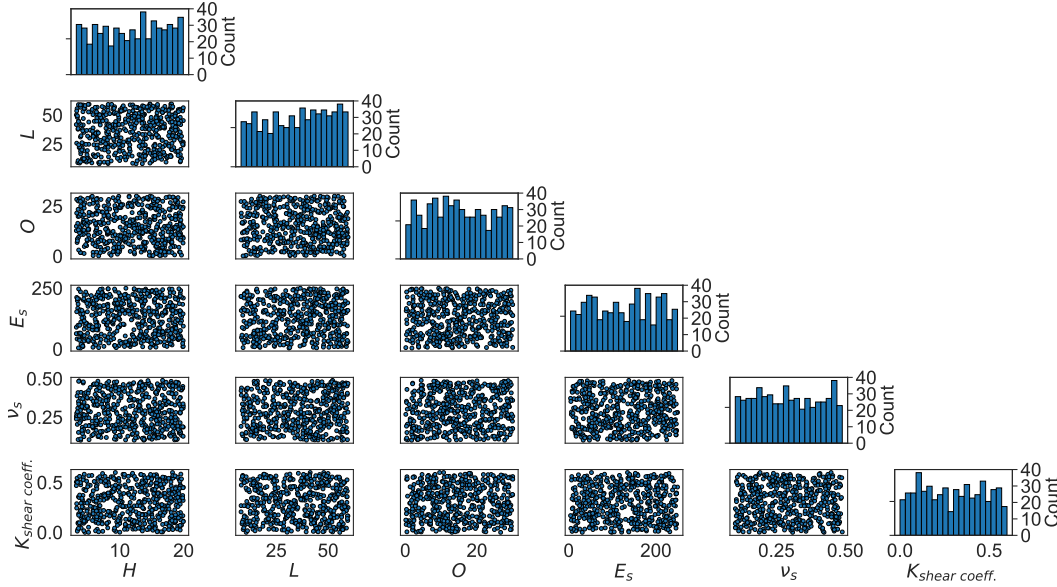


Figure 5.5: Sample distributions of additional soil and building parameters.

illustrate this process, three randomly selected samples, whose properties are reported in Table 5.9, are plotted to demonstrate:

- The material's compressive behaviour using Equation 2.35 (Figure 5.6a).
- Tensile behaviour using a parabolic curve from Equation 2.40 (Figure 5.6c).
- The corresponding stiffness degradation in both compression and tension using Equations 2.30 and 2.31 (Figures 5.6b and 5.6d, respectively).

The area under the curves represents the value of $G_{f(c \text{ or } t)}$ in the tension and compression regions of the material. These values are also presented in Table 5.9.

Each individual CDP model requires the definition of the above-mentioned parameters (curves). Other parameters not explicitly listed are considered constant and are assigned values consistent with those obtained from the validation models. This includes the CDP plasticity parameters, specific analysis settings, and model-related inputs such as wall thickness and ρ . These plots and the associated data are generated using Python scripting and are then used to develop the FEM Model Prototype within the material definition section of the ABAQUS interface.

Table 5.9: Material properties for CDP development for a random selection of input samples.

Sample Point	E (MPa)	f_c (MPa)	f_t (MPa)	G_{f_t} (N.mm/mm ²)
1	6,271.6	18.25	0.69	0.0284
2	1,370.6	2.58	0.32	0.0135
3	5,797.6	11.61	0.62	0.0173

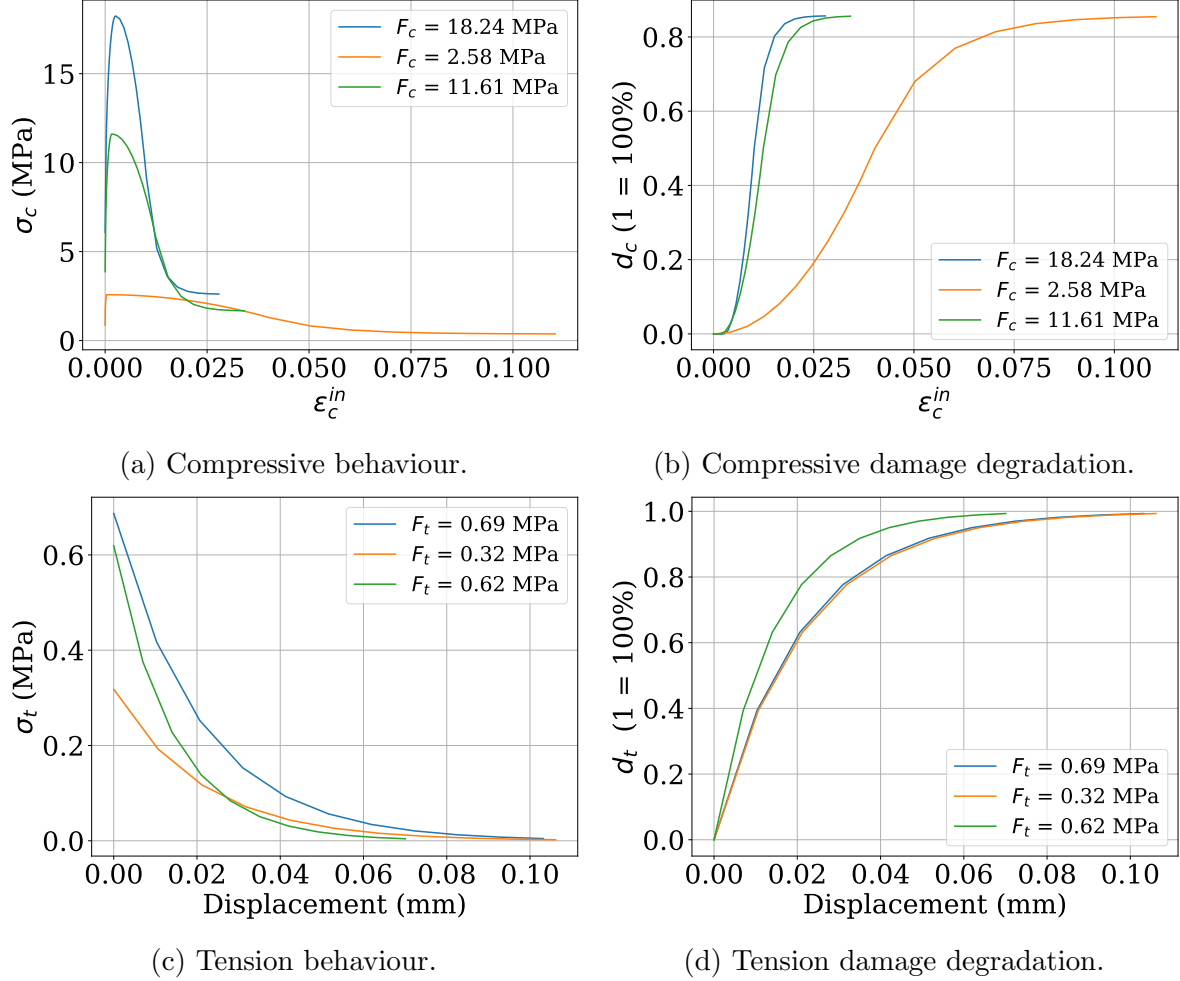


Figure 5.6: CDP material behaviour in tension, compression and corresponding damage degradation, respectively.

5.4 Generating Soil-Structure Interaction Using the Euler Bernoulli Beam on a Two Parameter Elastic Foundation

In conjunction with the development of CDP models, SSI-induced settlement profiles are similarly generated using the iterative procedure described in Section 2.3.1, incorporating relevant inputs from the MLH search space. The same sample points used earlier to illustrate the CDP models are employed here to demonstrate the corresponding SSI-induced settlements. The input values for these cases are listed in Table 5.10. It is noted that material non-linear parameters (e.g., f_t , f_c , G_{ft}) and $k_{\text{shear coeff.}}$ are not considered in the prediction of building deformation using the EBBEF2p model. This exclusion stems from the fact that the EBBEF2p code operates based on linear elastic assumptions and does not accommodate non-linear material behavior. Consequently, such parameters are only applicable within the FEM model, which is capable of capturing material non-linearities.

Table 5.10: Relevant parameters for the evaluation of SSI using EBBEF2p model for the same sample points of the CDP.

Parameter	Model 1	Model 2	Model 3
E (MPa)	6271.5	1370.6	5797.6
H (m)	18.0	5.2	3.2
L (m)	14.2	30.2	50.1
O (%)	8.5	25.9	14.9
e (m)	4.9	13.4	4.3
E_s (MPa)	29.7	47.0	89.6
ν_s	0.48	0.19	0.25
i_x (m)	5.3	38.5	39.4
VL (%)	1.02	3.94	0.51
Z_0 (m)	15.9	61.9	79.9
D (m)	5.7	8.9	15.5

Figure 5.7 illustrates the SSI effects for the selected sample points using the EBBEF2p code. The variation in settlement response is significant. For instance, Figure 5.7a demonstrates notable building resistance to deformation, attributed to high values

5.4. GENERATING SOIL-STRUCTURE INTERACTION USING THE EULER BERNOLLI BEAM ON A TWO PARAMETER ELASTIC FOUNDATION

of E combined with relatively low E_s , among other factors. In contrast, Figures 5.7c and 5.7b show strong adherence of the building to the ground surface (GF) deformation profile, indicating reduced structural stiffness and a heightened potential for damage.

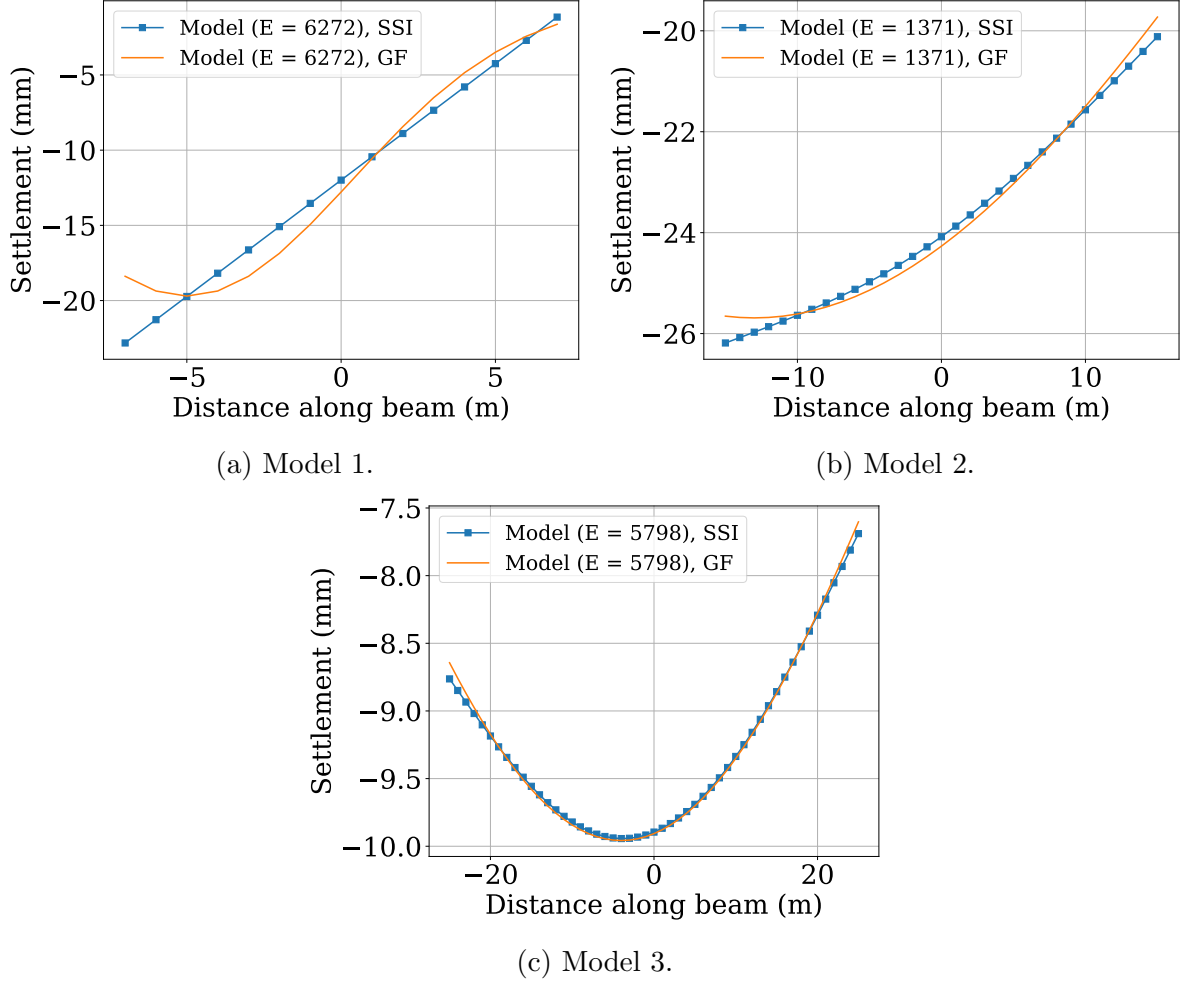


Figure 5.7: SSI of buildings with various levels of resistance to deformation.

Applying settlements as point displacements at the foundation level in ABAQUS requires a range of sometimes complex and stepwise considerations. ABAQUS allows for the use of analytical field functions, enabling the user to apply any non-linear displacement profile over a specified region by defining equations in terms of spatial coordinates. Therefore, as an additional step, polynomial fits were evaluated for each SSI model to identify the best-fitting settlement curve. Although polynomial orders up to 16 were initially considered during exploratory testing, in practice,

polynomials of degree 3 or 4 were sufficient for the vast majority of cases. The resulting function is continuous and directly applied to the FEM model. Across all models, the polynomial fitting achieves high accuracy, with an average R^2 value close to 1 and a minimum of $R^2 = 0.998$.

5.5 Model Prototype (Model 0)

With both CDP material models and SSI settlement profiles for N samples from the MLH search space now available, the step-by-step development of a parametric FEM prototype model, referred to here as Model 0, is described. This model includes fields for variables corresponding to each sampled input parameter, which are updated autonomously to simulate different structural conditions. This approach eliminates the need for significant manual input during model assembly, thereby minimising potential errors.

The model prototype is shown in Figure 5.8. The wall is modelled using 2D shell elements (ABAQUS S8R) and is assigned CDP material properties. It is noted that the use of 2D shell elements for the wall is consistent with earlier analyses and validation efforts, where the nonlinear wall behaviour was also modelled using shell elements in a 3D space. This modelling choice was made primarily to reduce computational demands. Moreover, the EBBEF2p model, which informs settlement inputs, is based on simplified 2D structural representations, further limiting the necessity or feasibility of full 3D modelling at this stage. However, this approach comes with certain limitations. Specifically, shell elements do not capture out-of-plane or torsional behaviours, which may be relevant in real structures. Additionally, structural components such as in-plane return walls, which could contribute significantly to the stiffness at façade edges, are not modelled. As a result, the simulated stiffness and deformation responses may be conservative or less representative of buildings with more complex three-dimensional configurations.

A foundation modelled using 3D solid elements (ABAQUS C3D20R) is further added, applying the same material properties as those used for the wall. In this case, the characteristic length h is calculated as $\sqrt[3]{\text{Volume}}$, assuming a volumetric aspect ratio of 1. Solid elements are required for the foundation because the implementation of contact elements between the building and soil surface necessitates one of the interacting bodies to be a solid (ABAQUS C3D20R) element. The soil is modelled as an elastic plate extending 2 m from each edge of the building footprint, allowing horizontal expansion in response to vertical cracking at the foundation. The plate is 10 mm thick and adopts soil properties that vary by sample (e.g., E_s , ν_s). However, due to the de-coupled nature of the modelling approach, these soil plate properties do not directly influence the output of the FEM model in terms of SSI.

To model the interaction between the foundation and the soil plate, a frictional contact interface is implemented using ABAQUS's penalty method. This was defined between the bottom face of the foundation and the top face of the soil plate using the parameter $k_{\text{shear coeff.}}$, which was sampled from the MLH space. This setup allows for sliding resistance via friction as well as separation through hard contact, thereby enabling both relative translations and gap formation between the interacting surfaces.

Because contact interactions between the building and soil plate are used, both vertical and horizontal interaction properties are defined. These include vertical stiffness (k_{norm}), which allows hard contact with potential for gap formation, and a penalty-based horizontal friction coefficient ($k_{\text{shear coeff.}}$), which is sampled directly from the MLH space. This setup results in a semi-coupled model: while the settlements are externally imposed based on EBBEF2p outputs, the interaction behaviour during displacement is influenced by the defined contact properties.

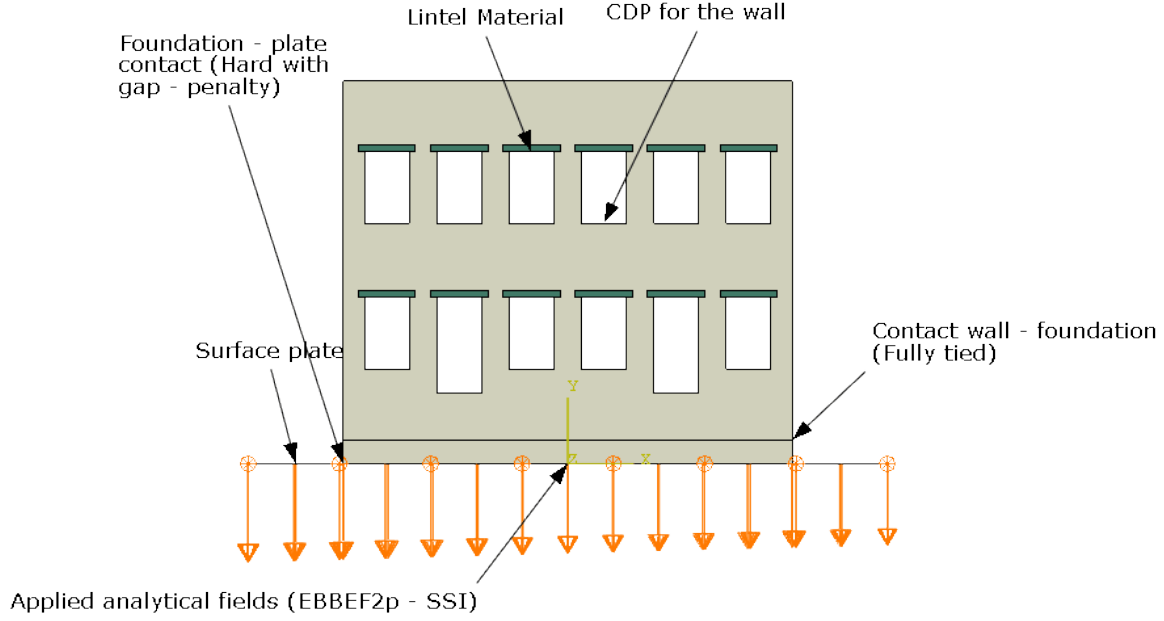


Figure 5.8: Model prototype with a basic definition and properties of the various components, including the application of load and area of the contact interface.

5.6 Automating Model Generation

With all components now prepared and ready to be assembled into a unified model (i.e., one FEM model per sample point), the model generation procedure begins. This step is executed within the ABAQUS Python Development Environment (APDE) environment, using a Python script that generates and exports input file ‘.inp’, with each file representing one FEM model. In total, N such files are created. This section provides an overview of the main code components used in the automation process, with relevant snippets explained where appropriate. These steps demonstrate how model assembly is managed between the APDE environment and the ABAQUS interface. The process begins by importing the essential ABAQUS libraries, presented in Listing A.5.

These imports enable the creation and management of geometry, materials, boundary conditions, and output controls. Constants used across all models are then defined, including window and door dimensions, material density (ρ), wall thickness, mesh sizing, and door frequency. Next, the values of the input parameters

from the MLH sample space are imported, which includes $N = 500$ samples for each parameter.

The model assembly process begins with a **for** loop iterating through each sample. Input parameters are adjusted according to the model constraints described previously in Table 5.6. All parameter values are also converted into International System of Units (SI) units (N and mm) to ensure consistency with the CDP and SSI definitions.

The major modelling steps are summarised below:

- **Determining the number of floors** based on the total building height H , presented in Listing A.6.
- **Calculating the number of virtual lines** corresponding to the centrelines of windows and doors. Each floor is assumed to contain the same number and placement of openings, evenly and symmetrically distributed. Only the ground floor contains door-sized openings, arranged such that two windows are placed between every two doors, following the approach in (Giardina (2013)). The number of columns of openings is determined using the Listing presented in A.7.
- **Defining the geometry** of the wall, foundation, and soil. Partitioning is applied to the soil to align with the mesh used for the wall and foundation, presented in Listing A.8.
- **Translating and assembling the model components** in the ABAQUS assembly module, presented in Listing A.9.
- **Partitioning the wall** to define window and door openings, as well as lintel zones. These are implemented through several **for**-loops and conditions. Due to their complexity, they are not detailed here.
- **Loading CDP material curves** for compression, tension, and their respective degradation values. These are applied to the wall and foundation based

on the current sample point, detailed in Listing A.10.

These properties are not explicitly reassigned in each model, as the naming conventions and constants are predefined and automatically linked to the relevant model parts.

- **Assigning the SSI settlement profile** as a vertical displacement field using the best-fit polynomial expression, presented in Listing A.11.
- **Introducing the contact stiffness value** $k_{\text{shear coeff.}}$ using a penalty-based tangential behaviour definition, presented in Listing A.12.

The remaining boundary conditions (described later in this section) are then defined, section properties are assigned, the mesh is generated, and field and history output requests are defined. The script then proceeds to generate the model for the next sample point ($i + 1$), continuing until all N models have been produced.

Component-specific settings for boundaries, mesh types, and section assignments are summarised in Table 5.11. Once complete, the generated ‘.inp’ files are passed to the execution phase using HPC facilities, as described in Section 5.7.

Table 5.11: Other model component specifications.

Component	Wall	Foundation	Soil Layer
Material Property	CDP	CDP	Elastic
Section Type	Shell (2D)	Solid (3D)	Shell (2D)
Thickness (mm)	250	250	10
Assembly	Top	Between	Bottom
Interaction	Tie-	Tie-Contact	-Contact (penalty with gap)
Mesh Type	S8R	C3D20R	S8R
Seed (mm)	175	175	175

The analysis procedure for the various FEM models is identical and is defined as a Dynamic, Implicit step with quasi-static application over a total execution time of 2.5 seconds to minimise momentum effects. The numerical step settings are: Maximum Number of Increments = 1000000; Initial Increment Size = 0.001; Minimum Increment Size = 1.5×10^{-15} ; and Maximum Increment Size = 0.025.

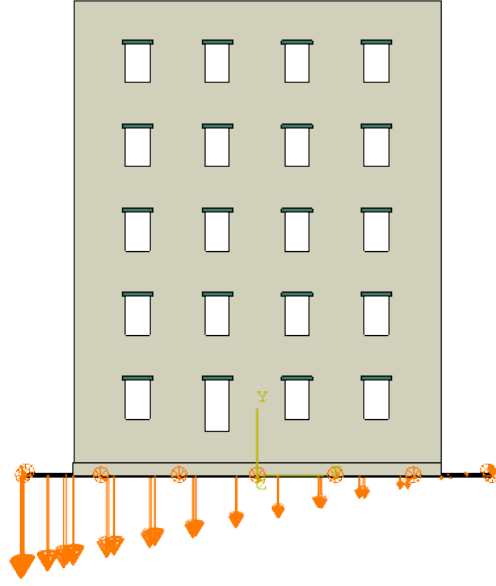
The field outputs requested for each model include: “Stresses”, “Strains”, and “Displacements”, along with “DAMAGEC” (d_c) and “DAMAGET” (d_t) contours, representing zones of inelasticity due to compressive and tensile failures, respectively. While “Displacements” and “Strains” are used for the evaluation of structural damage, the “Stresses”, “DAMAGEC”, and “DAMAGET” are used for illustration purposes only.

Additionally, history outputs are requested, including values over time for: Total Energy, Internal Energy, Kinetic Energy, Artificial Energy, and the nodal displacements of the foundation. Boundary conditions are applied as follows: both horizontal movements of the soil layer (panel) are restricted, allowing only vertical deformation to match the shape of the applied settlement trough; the wall is constrained in the out-of-plane direction to prevent twisting or bending along its thickness. Although the horizontal movement of the soil layer (panel) is constrained to maintain the shape of the settlement trough, the building foundation remains free to slide horizontally relative to the soil layer through the defined frictional contact interaction, enabling simulation of sliding, separation, and gap formation during settlement-induced deformation.

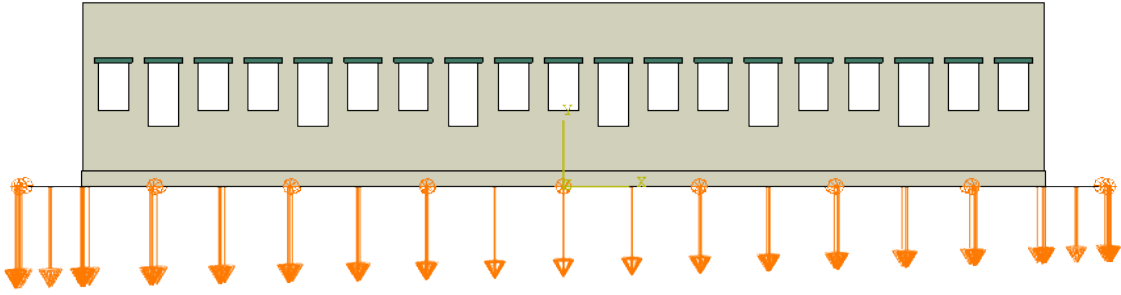
The same three sample points previously used for illustrating the CDP material models (Section 5.3) and the SSI profiles (Section 5.4) are shown in Figure 5.9, where the finalised FEM models are ready for execution.

5.7 High Performance Computing for ‘.inp’ File Execution

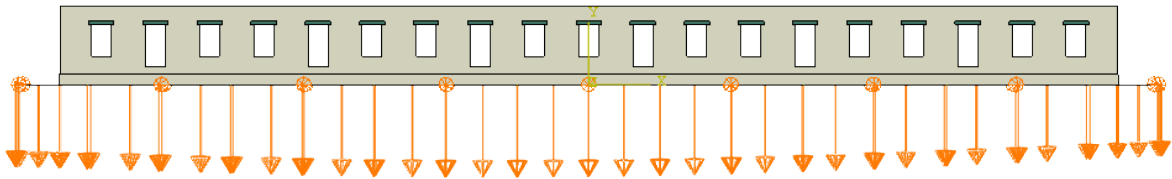
A complete set of N FEM-based input files ‘.inp’ is now prepared for execution. For this task, the HPC facility at the UoN is used. Both PuTTY (Figure 5.10a) and WinSCP software are utilised (Figure 5.10b). The user must be connected either via a WLAN cable (when on-site) or through a VPN connection (when off-site). A



(a) Model 1, 5 floors and located on hogging region.



(b) Model 2, 1 floor and located (centered) on inflexion point.

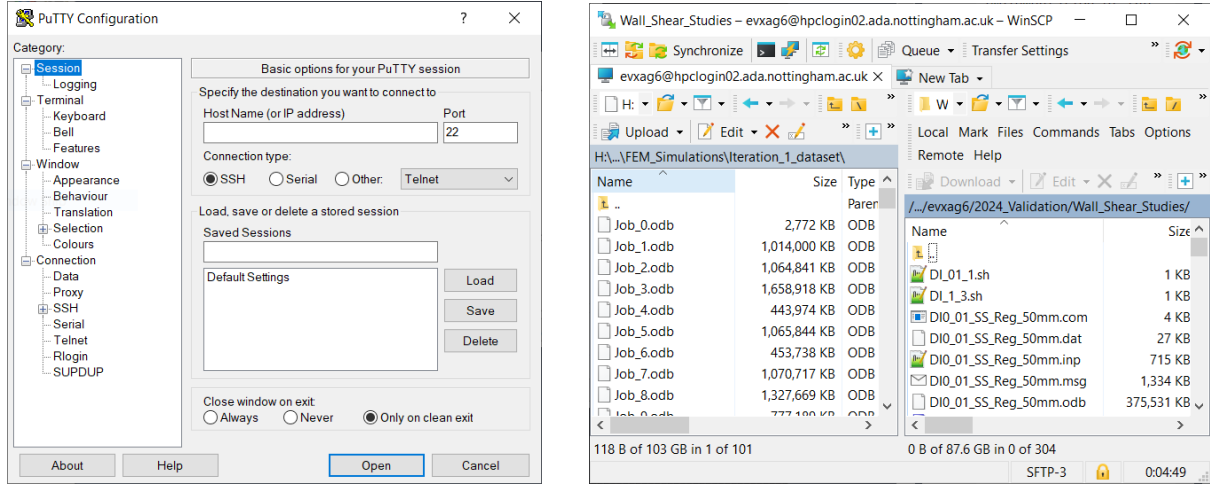


(c) Model 3, long structure, located on sagging region.

Figure 5.9: Model assembly in FEM, comprising geometry, material (CDP), applied settlements SSI (boundaries), and numerical settings.

good working knowledge of Linux and its terminal commands is also required.

Let Job_i denote the i -th input file, where $i \in [1, N]$. These jobs are executed using a Bash shell script ‘.sh’ i.e., ‘Jobs.sh’, containing a loop to submit a batch of jobs within a given range. The script is submitted using the code presented in Listing A.13.



(a) PuTTY login window configuration.

(b) WinSCP configuration (online with HPC).

Figure 5.10: Software to transfer, access, and execute ABAQUS ‘.inp’ files.

Upon execution, each job is placed into a queue depending on the number of available ABAQUS licenses at that moment.

Once the simulations are complete, results stored in the output database (‘.odb’) files are transferred using WinSCP from the remote HPC repository (right window) to the local machine (left window) for post-processing. Furthermore, by dividing the total jobs into multiple batches (e.g., 20 jobs per script), several ‘.sh’ scripts can be executed in parallel, significantly increasing the number of simulations run concurrently. This is particularly useful for time-constrained tasks requiring a large number of simulations.

5.8 An Iterative Procedure for Data Rebalancing

From the literature, it is concluded that the potential for output overfitting or unbalanced distributions can be inevitable in certain circumstances. Several studies in geotechnical engineering have proposed the use of Active Learning (AL) approaches as an effective way to address imbalanced output distributions. These methods selectively query the most informative and under-represented data points, enabling the model to train and test on a more diversified dataset, thereby improving the predictions of ML models (Qu et al. (2023)). However, one limitation of AL is its difficulty in handling data with high levels of noise or fluctuations Settles (2012), which is typical in tunnelling-induced damage scenarios in brittle materials.

Alternatively, Saadallah et al. (2019) applied under-sampling techniques to address data imbalance. They found that reducing the amount of majority-class data by up to 60% improved the prediction accuracy of tunnelling-induced settlements by up to 20%. Nonetheless, the effectiveness of under-sampling can vary depending on dataset characteristics. Oversampling techniques such as those proposed by Kubat and Matwin (2000); Luís et al. (2013) generate synthetic samples by interpolating between minority-class instances. While effective in some cases, these methods may produce unrepresentative samples when the minority target values are widely scattered.

To address these limitations, this section introduces an iterative approach aimed at developing robust datasets and improving the quality of synthetically generated outputs in the presence of skewed data. The proposed method adapts to learned statistical patterns by refining the input space over several iterations, potentially enhancing ML prediction performance.

The iterative procedure consists of the following steps:

- i) Identification of influential parameters via SA;
- ii) Statistical analysis of input parameter distributions;

- iii) Identification of high-impact parameter regions;
- iv) Resampling of refined input space.

A schematic representation of this approach is shown in Figure 5.11.

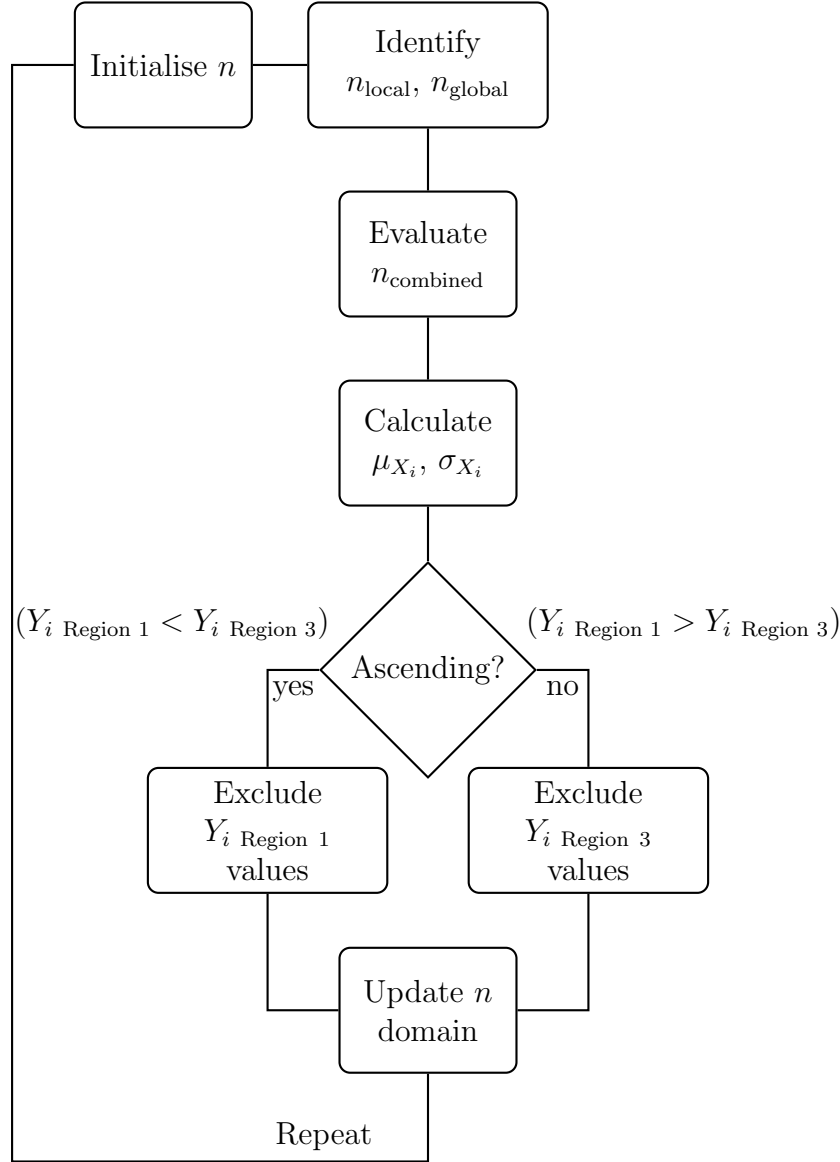


Figure 5.11: Flowchart of the iterative procedure including feature selection, statistical exploration, and range adjustment.

Step 1: Identification of Influential Parameters

The process begins by identifying influential input parameters that significantly affect system responses. Let the dataset inputs be denoted as $n = \{X_1, X_2, \dots, X_n\}$,

shared across all output variables. Using a RFR model, the importance $I(Y_i)$ of each input X_i is computed for each output variable (Menze et al. (2009); Antoniadis et al. (2021)).

A threshold of $\phi = 0.05$ is used to identify significant features:

$$n_{\text{local}} = \{X_i \in n \mid I(Y_i)_{\text{local}} \geq \phi\} \quad (5.2)$$

$$n_{\text{global}} = \{X_i \in n \mid I(Y_i)_{\text{global}} \geq \phi\} \quad (5.3)$$

The final set of important features combines both:

$$n_{\text{combined}} = n_{\text{local}} \cup n_{\text{global}} \quad (5.4)$$

A parameter is deemed influential if it exceeds the threshold for at least one output variable.

Step 2: Statistical Exploration

The statistical distribution of each selected parameter $X_i \in n_{\text{combined}}$ is analysed by computing its mean μ_{X_i} and standard deviation σ_{X_i} . The domain is divided into three regions:

$$\text{Region 1: where } X_i < \mu_{X_i} - \sigma_{X_i} \quad (5.5)$$

$$\text{Region 2: where } \mu_{X_i} - \sigma_{X_i} \leq X_i \leq \mu_{X_i} + \sigma_{X_i} \quad (5.6)$$

$$\text{Region 3: where } X_i > \mu_{X_i} + \sigma_{X_i} \quad (5.7)$$

Step 3: Identifying Regions of Interest

For each region, the average output value is computed: $Y_{i, \text{Region 1}}$, $Y_{i, \text{Region 2}}$, and

$Y_{i \text{ Region } 3}$. The direction of impact is determined by comparing $Y_{i \text{ Region } 1}$ and $Y_{i \text{ Region } 3}$:

- If $Y_{i \text{ Region } 1} < Y_{i \text{ Region } 3}$, the trend is ascending (increased values produce more damage).
- If $Y_{i \text{ Region } 1} > Y_{i \text{ Region } 3}$, the trend is descending (decreased values produce more damage).

Depending on the trend, low-impact regions are excluded from the sampling space in the next iteration. If conflicting trends exist for a parameter across outputs, the steeper gradient is prioritised.

Through this process, it is possible to: i) identify influential parameters, ii) evaluate their distributions, and iii) target impactful regions. This results in increasingly damage-prone FEM models per iteration, improving dataset quality and reducing computational waste. Further implementation is discussed in Section 6.4.1.

5.9 Summary

This chapter presented the methodology for identifying input parameters, their correlations, and interdependencies, while accounting for both physical and numerical constraints. A MLH sampling procedure was employed to generate a diverse set of input configurations. These configurations were then used to automatically assemble FEM models via the APDE tool and executed on the UoN HPC facility through batch processing.

This chapter fulfilled the second and third objectives of the thesis (Objectives 2 and 3 in Section 1.5). In addition, it introduced an iterative refinement procedure aimed at improving output balance and dataset representativeness. The implementation and impact of this procedure are further explored in Chapters 6 and 7.

Chapter 6

Evaluation of Local and Global Damage to Buildings - Case Study of Numerical Simulations

From the analysis performed on the numerous synthetic FEM models, the research question is addressed: How do various local and global damage assessment methods, as found in the literature (Section 2.2), perform in determining and understanding building damage in numerical case studies? By employing these methods, the differences in damage understanding based on the specific approaches used are showcased. Hence, both “local” and “global” damage metrics are assessed, providing key insights into building behaviour in response to tunnelling. This chapter begins with the extraction of model node displacements, as well as ε values from the ‘.odb’ files produced by the executed simulations. This is followed by the assessment of local (e.g., Maximum Crack Width), global (e.g., $\varepsilon_{h,\max}$), analytical-global (i.e., the LTSM method), and a numerical local-global approach (i.e., Total Damaged Area A_{damaged} and values of ε_t). An evaluation of overall distributions and correlation studies concludes the chapter. A flowchart of the chapter logic is presented in Figure 6.1.

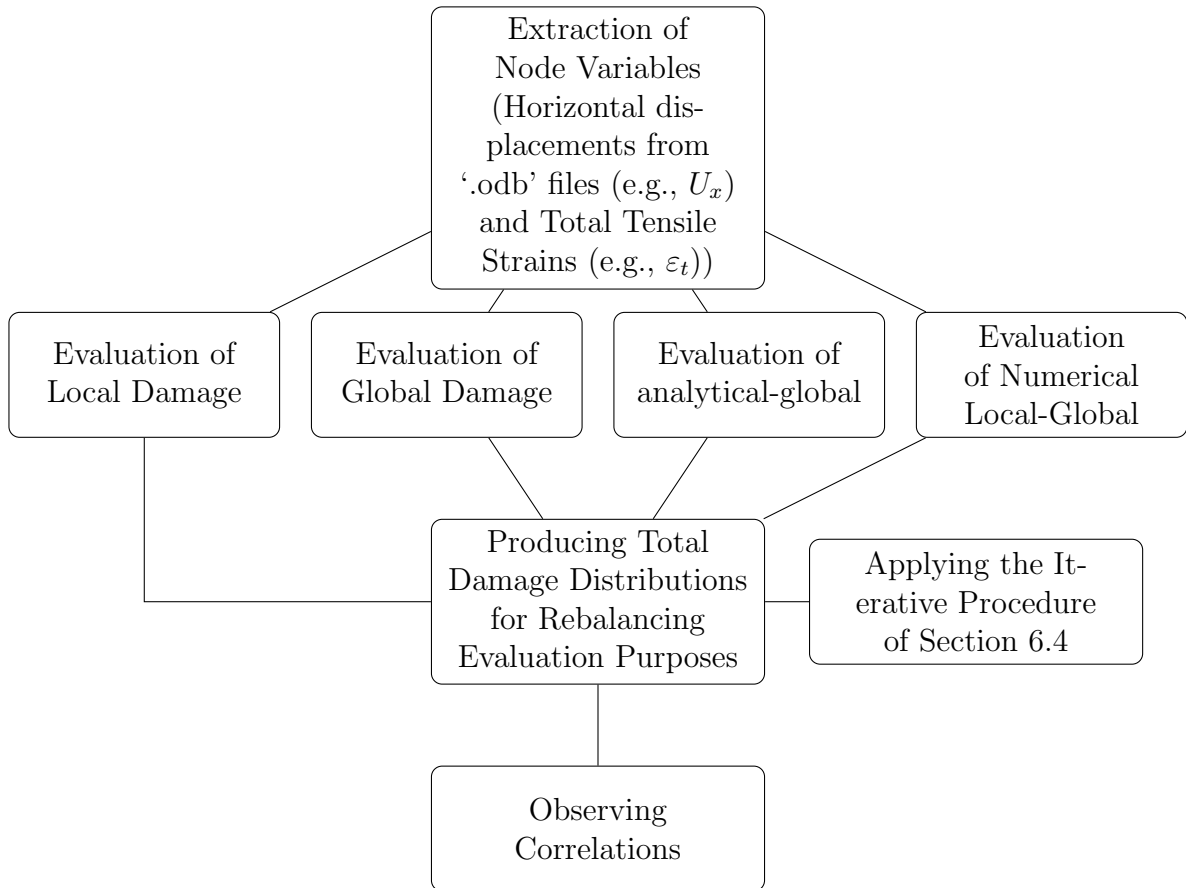


Figure 6.1: Flowchart illustrating the process of damage evaluation and processing pipeline.

When considering local aspects of building damage, the focus is on examining each individual element in the model to identify extreme strain values indicative of damage. Due to the Macro-modelling nature of the mesh, this cannot be classified as a discontinuity but rather as a smeared damage within a continuum. Various approaches exist to assess local damage, including the measurement of bending and shear in individual elements, as used innovatively in the LTSM method (Section 2.2.4), or by evaluating crack width magnitude and count.

However, in this chapter, local damage assessment refers exclusively to the Maximum Crack Width and the Total Number of Cracks. Due to technical challenges in automating damage evaluation, particularly during the meshing of the building model, both criteria are currently measured only at the extreme fibres (i.e., the top and bottom wall fibres). When modelling the building, the meshing process does not enforce horizontal partitioning between levels; instead, the mesh is generated randomly within the building's main body. In contrast, the extreme fibres are typically fixed and vertically aligned, which greatly facilitates their automated identification and subsequent calculations. This approach represents a limitation of the present methodology. The variable Maximum Crack Width is defined as the highest crack value detected along the extreme fibres, while Total Number of Cracks is the cumulative count of all cracks identified within these regions. The inability to partition the building horizontally means that cracks occurring between these extreme regions, including variations in crack depth with height or cracks near openings, may not be captured accurately. Addressing this would require the building to be partitioned into distinct horizontal layers at the desired levels prior to meshing, an enhancement that was not implemented in the current study. The method used for identification follows the same approach as in Section 4.3.

Global damage assessment methods, on the other hand, pertain to the behaviour of the building as a whole or as partial segments, typically measured in the continuum. The key global metrics of interest include s , t , β , and $\varepsilon_{h,\max}$, measured either at

the top fibre (hogging mode) or the bottom fibre (sagging mode), depending on deformation patterns. These metrics, in conjunction with the local metrics, can be categorised into damage levels as defined in Section 2.2.3, offering comprehensive insights into the extent of building damage.

Traditionally, global metrics are applied to entire structures or visually identified damaged zones. Son and Cording (2005) adopted such methods effectively, leveraging manual inspections to locate the most critical segments. However, this is feasible only when analysing a small number of buildings. In this study, where hundreds of buildings with varied properties are analysed, a scalable alternative is proposed: the “screening over the façade” approach.

To capture potential critical damage occurring at different scales, the building is iteratively partitioned into progressively larger segments: $1/8^{\text{th}}$, $1/6^{\text{th}}$, $1/4^{\text{th}}$, $1/3^{\text{rd}}$, $1/2^{\text{nd}}$, $2/3^{\text{rd}}$, and the entire building. At each segmentation level, global damage metrics $(s, t, \beta, \varepsilon_{h,\max})$ are computed for each segment, and the maximum value within each partition is retained. The final reported metric is then selected as the highest value obtained across all segmentation levels, ensuring that both localised and distributed damage patterns are adequately identified.

Figure 6.2 shows an example segment from a randomly selected building model, extracted from an executed ‘.odb’ file in ABAQUS. It presents the damage in tension (DAMAGET) distribution (where 1 indicates full tensile damage) with the reference corner points A, B, C, and D marked. These reference points are used to evaluate node displacements for subsequent global metric calculations.

6.1 Evaluating Building Damage using Various Criteria

Building damage is evaluated systematically and automatically; thus, each individual FEM model is not inspected manually. Using Python scripting, the extraction

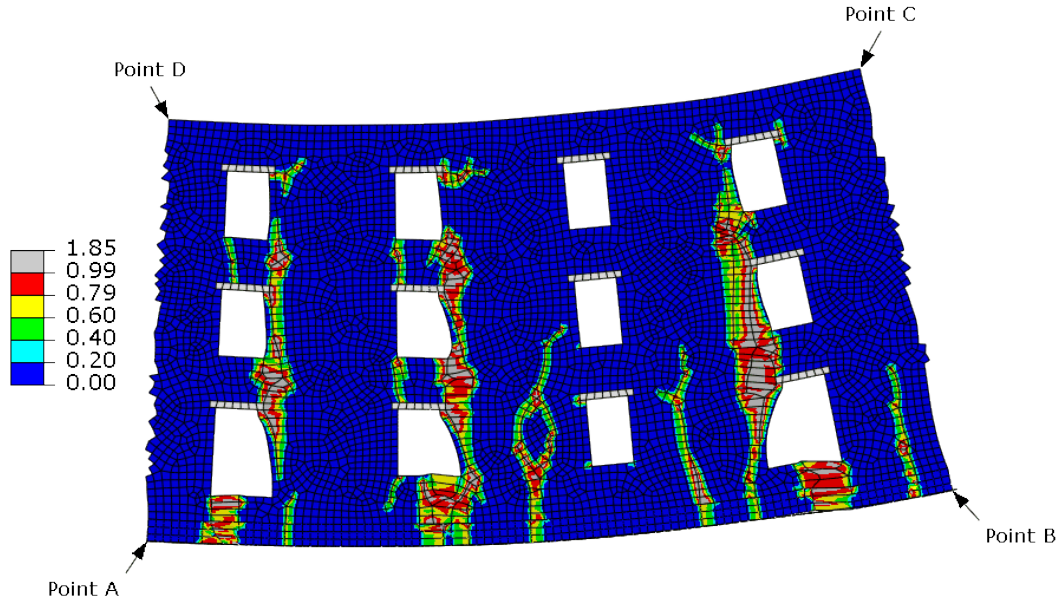


Figure 6.2: Damage distributions (DAMAGET) of building segment exposed to sagging mode of deformation.

of field variables (e.g., U_x and ε_t) is automated by calling the ‘.odb’ files within the ABAQUS interface in background mode. A code snippet demonstrating this logic is shown in Listing A.14.

Each output is stored with a unique reference name. The script processes each ‘.odb’ file and exports ‘.txt’ files containing nodal displacements and strain values at element centroids, along with DAMAGET contour plots in ‘.png’ format for each FEM model. It is worth noting that ABAQUS Python scripts can either be written from scratch using the user manual Dassault Systèmes Simulia Corp (2021) or generated automatically via ‘.rpy’ recordings from manual operations, which can then be refined, looped, and parameterised to suit different model development needs.

For this damage evaluation, the focus is placed solely on the façade and foundation; the lintels and soil plate are neglected.

6.1.1 Case Studies - Local Damage Evaluation

In this section, local building damage metrics are evaluated using highly impacted FEM case studies. Figure 6.4 presents a selection of buildings subjected to varying degrees of damage, focusing specifically on Maximum Crack Width and Total Number of Cracks. Figure 6.4a shows a building located near the inflection point of the deformation curve, exhibiting widespread DAMAGET values due to high ε_t , which is linked to relatively low tensile properties: $f_t = 0.372$ MPa and $G_{f_t} = 0.00174$ N.mm/mm². Additionally, the low stiffness of the building ($E = 1930$ MPa) and the relatively high soil stiffness ($E_s = 204$ MPa) resulted in pronounced SSI behaviour, forcing the structure to conform more closely to the ground deformation (GF) profile, contributing further to induced damage.

In this scenario, damage is mostly concentrated in the hogging and sagging zones, a typical pattern in such cases Burland (1997); Burland et al. (1977); Boscardin and Cording (1989), with one prominent crack initiating near the inflection point. These cracks generally begin at the extreme fibres and can propagate vertically along the height (H), particularly near wall openings due to their weakened cross-sectional resistance, an observation consistent with the experimental validation from Section 4.3 and earlier findings by (Son and Cording (2005)).

Figure 6.4b illustrates a second building under hogging deformation. Vertical cracks are observed developing in the upper fibre and propagating downward across openings. In this and the previous case, when cracking propagates significantly along the height (H), the building is expected to split vertically into separate structural blocks, initiating at the top and progressing downwards, particularly near the inflection point (see Figure 4.10a of Section 4.3). However, due to the decoupling of SSI in the EBBEF2p model, this effect cannot fully develop due to the settlement profile being fixed, contributing to further supporting the structure even after complete loss of strength, leading to an underestimation of the Maximum Crack Width.

Nevertheless, the predicted crack locations and their vertical propagation patterns are consistent and show good agreement with the observations made by (Son and Cording (2005)). The main exception is that the cracking in Son and Cording (2005) followed shear patterns (zig-zagging), which can again be attributed to the additional support provided by the predetermined SSI settlement profile. In other words, as non-linear behaviour introduces discontinuities in the continuum, the redistribution of stresses between the building and the ground is absent. Additionally, because the model is vertically restricted by the imposed settlement profile, only horizontal movements are permitted through frictional properties defined at the soil–foundation interface. This, in turn, can increase axial strain along the wall cross-section.

Lastly, Figure 6.4c shows a slightly damaged building under the sagging mode of deformation. In this model, cracks propagate along the height of the building from the bottom fibre, and the length of these cracks along H is considerable. It is therefore reasonable to assume that the building has segmented into three parts, with limited connection between them.

The segmentation approach, where different segments are considered as separate structures, is more evident when fully coupled analyses are performed, particularly when examining discontinuities in the SSI settlement profile.

However, as explored, this is also highly dependent on the length of the cracks along H . Whether segmentation should be considered when cracking reaches 25%, 50%, or 75% of H , or based on the magnitude of change in s between segments, is not clearly defined in the literature.

Figure 6.3 illustrates the distribution of input parameters used for the selected case studies. In this figure, the extended bars represent the parameter ranges, while the I-shaped markers indicate the value of each parameter for each model and its relative position within the total range. Furthermore, Table 6.1 summarises the local damage metrics: Total Number of Cracks, Maximum Crack Width and its

location, as well as the assigned damage category in accordance with Table 2.1 of Section 2.2.3.

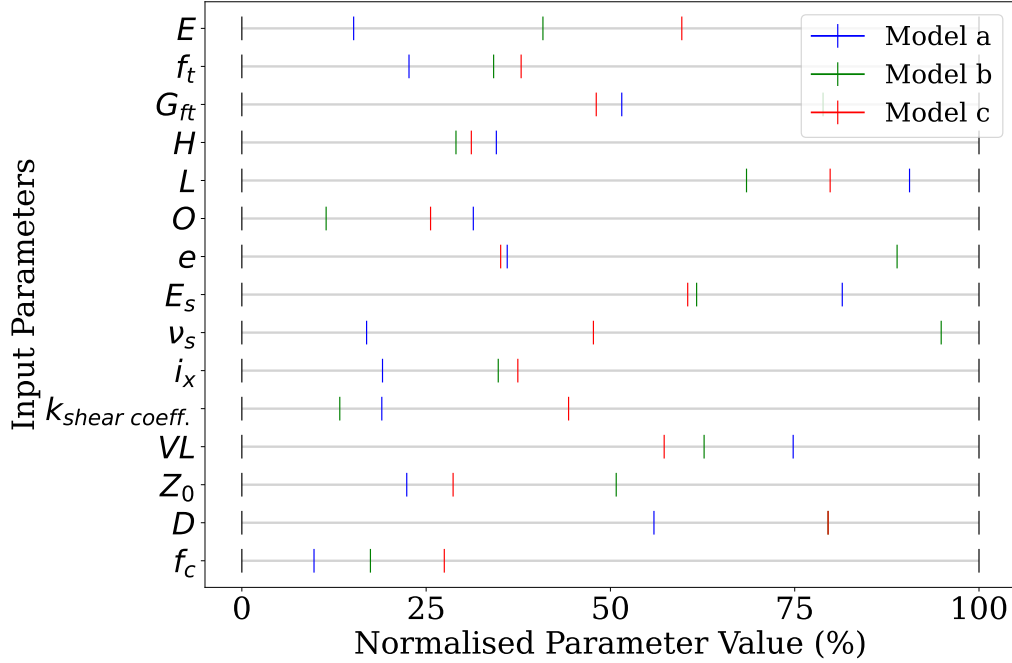


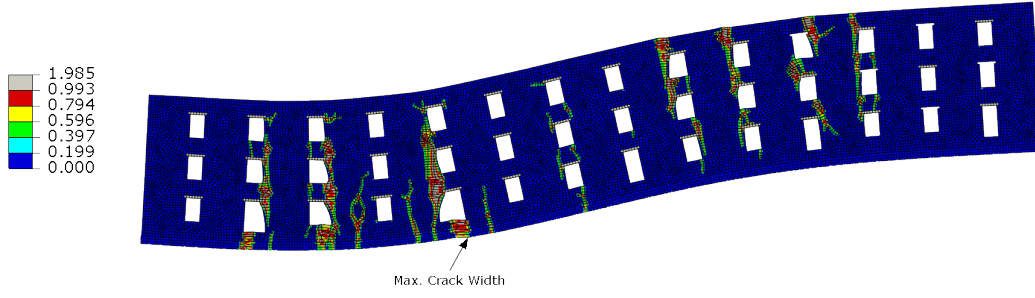
Figure 6.3: Input parameter distributions (in normalised terms) for the models used for the local damage evaluation of the selected case studies in Figure 6.4.

Table 6.1: Local building damage evaluation metrics of Figure 6.4.

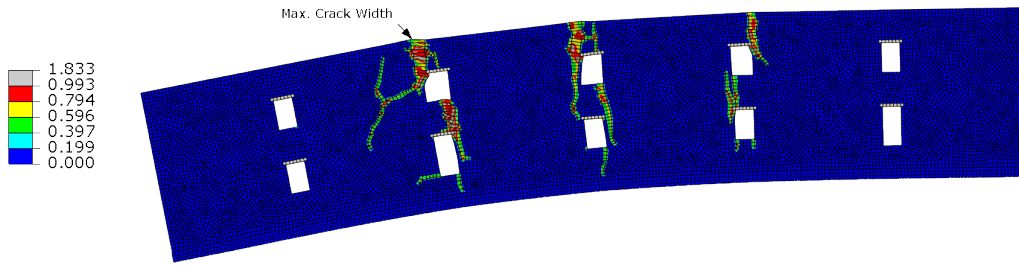
Damage Evaluation	Model a	Model b	Model c
Tot. No. Cracks	15	3	2
Max. Crack Width (mm)	16	9.5	3.5
Location (% from left edge)	33.9	31.3	27.3
Category of Damage	Moderate	Slight	Slight

As seen from Table 6.1, evaluating local damage through scripting does not include the identification of cracks between the extreme fibres of the wall, nor does it estimate their depth along H . This limitation is due to the complex and unique mesh arrangement generated within the wall boundaries and across different models. While this can be partially addressed by first partitioning the wall along H , that approach is not applied in the current analysis.

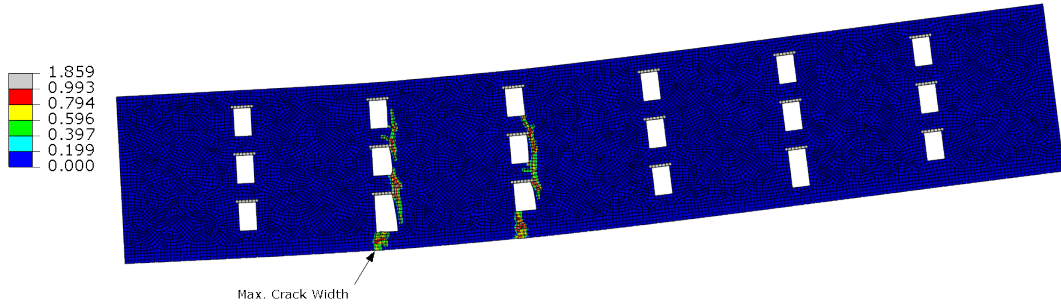
The above observations on façade cracking correspond to the final (mature) crack state under the theoretical GF values of $S_{v,max}$. The foundation’s history outputs



(a) Considerable building local damage “At inflection”.



(b) Building local damage “At hogging”.



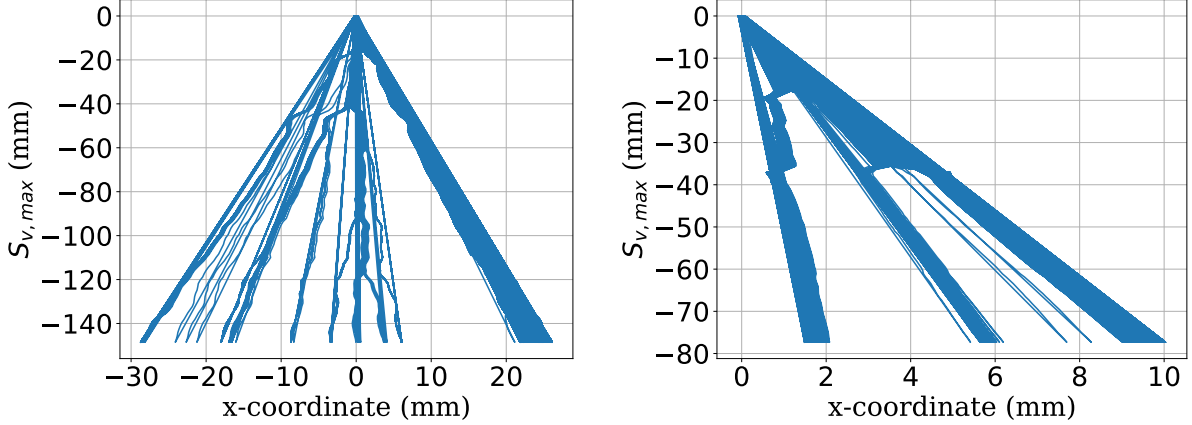
(c) Slight building local damage “At sagging”.

Figure 6.4: Several categories of building damage are observed for the various configurations (DAMAGET contours). In addition to presenting the location of Maximum Crack Width.

of nodal displacements were monitored throughout the application of load, allowing me to identify the moment and location of crack initiation and propagation. This is illustrated in Figure 6.5. In this figure (Figures 6.5a and 6.5b), the relative positions of foundation nodes are normalised from zero and plotted horizontally

from a reference point, showing progression up to $S_{v,\max}$. Clusters of points indicate sections of the building that move as unified blocks without cracking in between, thus, the gaps between clusters represent cracks.

Figure 6.5a demonstrates the development of 11 cracks at the foundation level due to sagging deformation (as seen in Figure 6.5a), with an additional 4 cracks, totalling 16 as shown in Table 6.1, forming above the hogging region and not captured during load application. Notably, the Maximum Crack Width of 16 mm initiated at approximately $S_v(x) = 15$ mm and continued to widen thereafter. Conversely, Figure 6.5b depicts the crack initiation and propagation for the wall shown in Figure 6.4c. The initial crack formed to the left of the eventual Maximum Crack Width, which emerged around $S_v(x) = 18$ mm. This crack subsequently released tension from the first crack, causing the earlier crack to close at about $S_v(x) = 32$ mm. This segmentation behaviour persisted as the newer crack propagated to its final width. A further crack was then initiated at approximately $S_v(x) = 35$ mm.



(a) Node displacements (model of Figure 6.4a). (b) Node displacements (model of Figure 6.4c).

Figure 6.5: Foundation nodes displacements with increased values of $S_v(x)$ until reaching $S_{v,\max}$.

Figure 6.6 illustrates the initiation and increasing crack width values of the Maximum Crack Width for each model presented above, up to $S_{v,\max}$. The purpose of this plot is to provide a deeper understanding of crack development and propagation as settlement increases, offering further insights into key aspects of building

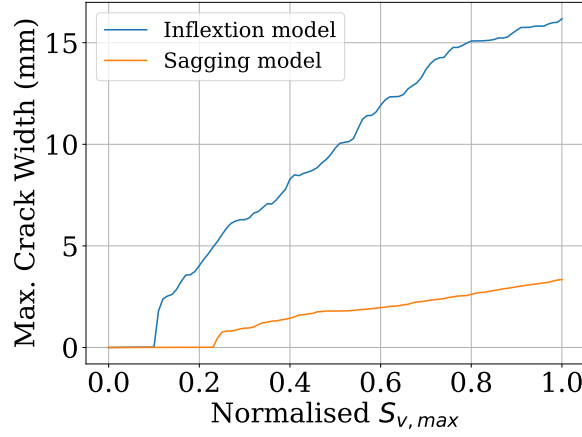


Figure 6.6: Initiation and propagation of Maximum Crack Width values with increased values of normalised $S_v(x)$.

damage.

Figure 6.6 illustrates the initiation and increasing crack width values of the Maximum Crack Width for each model presented above, up to $S_{v,max}$. The two curves represent different building positions along the settlement trough: the “Inflexion model” corresponds to a building located near the inflection point of the settlement curve, whereas the “Sagging model” represents a building located within the sagging (concave) zone. The purpose of this plot is to provide a deeper understanding of crack development and propagation as settlement increases, offering further insights into key aspects of building damage.

It is noted that categorising damage solely based on local aspects of building performance may limit the understanding of the overall potential risk. Nonetheless, valuable insights can still be drawn from local assessments, although additional evaluation criteria are typically recommended (Burland et al. (1977)).

6.1.2 Case Studies - Global Damage Evaluation

Similarly, building damage in highly impacted FEM case studies is reassessed according to global metrics. The examination begins by illustrating two buildings believed to experience significant global deformations, as shown in Figure 6.7.

It is immediately evident that the building shown in Figure 6.7a did not exhibit material non-linearity, as indicated by its $f_t = 0.51$ MPa and $G_{f_t} = 0.0021$ N.mm/mm². Yet, from a global perspective, the building undergoes significant deformations, as seen in the values of t and s . This response can be attributed to a combination of $E = 4180$ MPa, $L = 20.3$ m, $e = 14.8$ m, $i_x = 10.9$ m, and a VL value of 3.76%. The relatively high stiffness and volume loss values, together with the proximity between e and i_x (i.e., placing the building at the inflection zone where s is maximised), and a comparatively small building length all contribute to elevated global deformation metrics, resulting in a damage category that exceeds negligible.

In contrast, Figure 6.7b presents a building displaying significant non-linear behaviour, with increased values of β (attributable to larger θ) and $\varepsilon_{h,\max}$ (due to horizontal expansion from crack development), consistent with findings reported in (Son and Cording (2005); Giardina et al. (2015)). The observed increase in β is partly attributed to the limited building height ($H = 3.72$ m), coupled with relatively high soil stiffness ($E_s = 172$ MPa), which makes the structure more compliant with ground deformations. Similarly, the elevated $\varepsilon_{h,\max}$ is associated with a low fracture energy ($G_{f_t} = 0.0015$ N.mm/mm²) and low structural height, both contributing to increased horizontal expansion due to the development of multiple cracks across the building segments.

Figure 6.8 illustrates the distribution of input parameters used in these selected case studies. In addition, Table 6.2 summarises the global damage metrics, including s , t , β , and $\varepsilon_{h,\max}$. The corresponding damage categories are assigned using the classification criteria in Tables 2.2, 2.3, and 2.4, as well as the damage curves presented in Figure 2.4.

It is important to highlight that, depending on the metric used, a building may fall into different damage categories. As observed in Table 6.2, the same model may be classified with slight damage based on s , but severe damage according to β . This is in addition to the variations observed from local damage metrics. Hence,

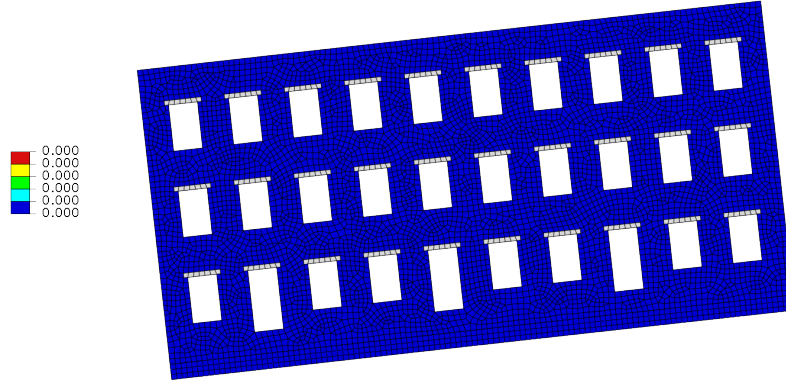
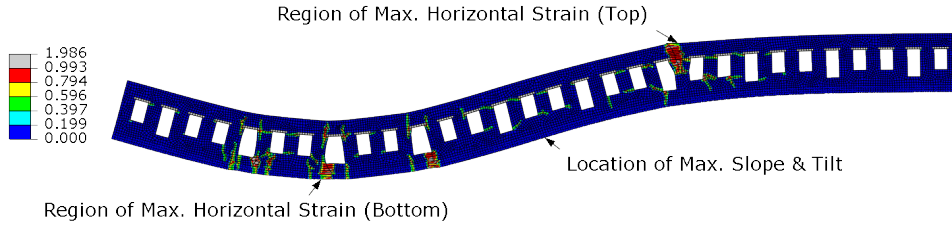

 (a) Structure undergoing large values of t and s .

 (b) Structure undergoing large values of β and ε_h .

Figure 6.7: Example structures with significant damage according to global metrics (illustration of DAMAGET contours).

Table 6.2: Global building damage evaluation metrics of the case studies presented in Figure 6.7.

Damage Evaluation	Model a	Model b
s	0.006604 (1/151)	0.004907 (1/204)
Category for s	3/4 (Moderate)	2/4 (Slight)
t	0.006550 (1/153)	0.004582 (1/218)
Category for t	3/5 (Monitoring)	3/5 (Monitoring)
β^a	5.4×10^{-5}	3.162×10^{-3}
Category for β	0/4 (Negligible)	5/5 (Severe)
$\varepsilon_{h,\max}$	2.5×10^{-5}	2.776×10^{-3}
Category for $\varepsilon_{h,\max}$	0/5 (Negligible)	3/5 (Moderate)

^a Damage categorisation using the variable β must be used in conjunction with the corresponding value of $\varepsilon_{h,\max}$ in Figure 2.4 to determine the level of induced damage.

it is crucial to include a combination of multiple metrics, capturing both localised and global structural responses, to ensure a comprehensive evaluation of building damage. Moreover, depending on the specific use and requirements of the structure,

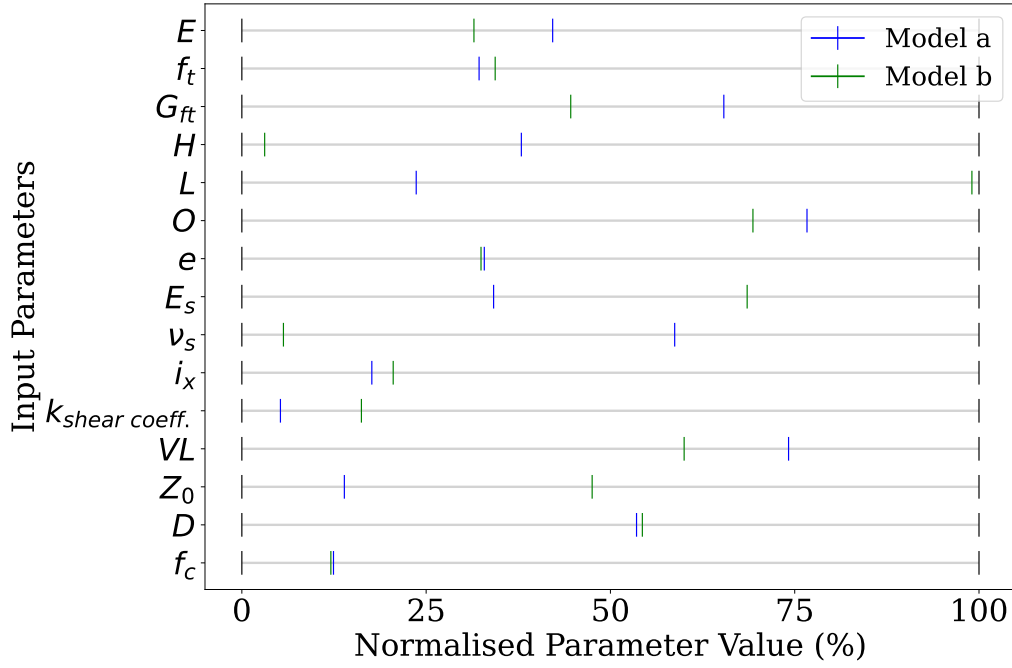


Figure 6.8: Input parameter distributions (in normalised terms) for the models used for the global damage evaluation of the selected case studies in Figure 6.7.

it may be appropriate to prioritise some metrics over others.

6.1.3 Case Studies - Limiting Tensile Tensile Method Assessment

For comprehensiveness, the analytical-global LTSM method is also applied to evaluate building damage in case studies identified as highly impacted with respect to this criterion. This method is widely adopted for assessing the global behaviour of buildings (Burland et al. (1977)). Although simplistic, it serves as a useful addition to the overall assessment framework when used alongside the previously described local and global methods.

The results presented here are derived from the outputs of the analytical model, which incorporates linear input variables. Specifically, the calculated values are obtained by applying equations 2.12 to 2.16 as defined in Section 2.2.4.

However, the method is limited, as it assumes linear elastic behaviour and idealises

the structure as a simple beam, which may not fully capture complex structural responses. Nonetheless, the building does not follow a GF settlement profile; instead, it resists ground movement based on the relative stiffness between the structure and the soil. Although SSI effects are indirectly considered through this interaction, the simplifications may still lead to an under- or overestimation of the induced damage in certain scenarios. To evaluate these metrics, ε_b , ε_d , and ε_h are measured to determine the failure type and damage category using the equations presented in Section 2.2.4, specifically equations 2.12 to 2.16. The assessment framework is implemented in Python and does not require any FEM outputs, as it is a fully analytical approach that relies solely on the input parameters defined at each sampled point. Two examples are used to demonstrate the outcomes of this method, with the corresponding input parameters provided in Table 6.3.

In the LTSM calculations, the shear stiffness G was derived using the standard elastic relationship:

$$G = \frac{E}{2(1 + \nu)} \quad (6.1)$$

However, both E and G were adjusted to account for the presence of openings in the façade or wall layout. These reductions were based on empirical relations derived using the reduction assessments from Son and Cording (2007), which provide equivalent elastic properties suitable for representing openings in masonry. This modification enables a more realistic representation of the building's flexural and shear behaviour than assuming a solid section.

Figure 6.9 illustrates the building's location with respect to the tunnel centreline and the settlement trough, incorporating both vertical and horizontal ground displacements. This is a key distinction from the previously described EBBEF2p approach used for the FEM models, which considers only vertical settlement. The inclusion of horizontal displacements in the LTSM method provides a more com-

prehensive representation of SSI, especially for estimating deformations in hogging and sagging regions. Dashed lines in different colours indicate the corresponding building sections within each ground deformation zone.

Table 6.3: Models input parameters for LTSM metrics.

Parameter / Model of	Figure 6.9a	Figure 6.9b
E (MPa)	3,762.5	5,791.9
H (m)	11.6	4.1
L (m)	58.0	40.5
O (%)	15.0	24.6
e (m)	11.4	36.3
i_x (m)	17.7	13.2
VL (%)	3.46	2.51
Z_0 (m)	34.9	25.7
D (m)	11.5	12.7

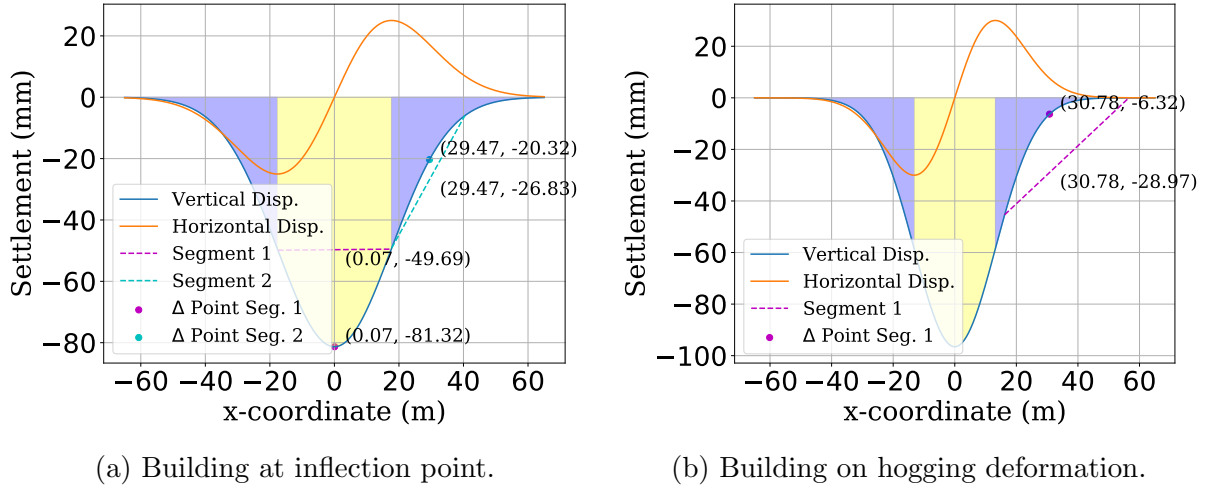


Figure 6.9: Building location WRT settlement trough for the assessment of building damage according to the LTSM method.

Specifically, Figure 6.9a illustrates a building located in the region between sagging and hogging modes of deformation, crossing through the inflection point (similar to the case in Figure 6.4a). Along each segment of the building (before and after the inflection point), the values of Δ are measured, and subsequently the DR parameter. Using the equations for the LTSM method, predictions are made for each segment for the values of ε_b and ε_d , also considering ε_h . These values are then compared with ε_{lim} across the segments to determine the dominant mode of

failure and its magnitude, thereby reflecting the level of induced damage based on the categorisation described in Table 2.4.

On the other hand, Figure 6.9b shows a building positioned exclusively in the hogging region. The entire building is considered as a single segment, with the corresponding calculations performed similarly to the previous example. The results for both models are summarised in Table 6.4.

Table 6.4: Results for LSTM models presented in Figure 6.9.

Model	DR	ε_b	ε_d	Failure Mode	Damage Category
Figure 6.9a	9.00E-04	2.62E-03	1.48E-03	Bending	Moderate
Figure 6.9b	5.62E-04	-5.21E-05	8.96E-07	Bending	Negligible

Although a simplistic method, the inclusion of the LSTM in the assessment framework is beneficial, as it enriches the initial understanding of building damage using considerably fewer computational resources. Its outputs can be directly compared with those from more advanced methods for further investigations, such as comparisons of direct damage categorisation, correlation studies, and more.

6.1.4 Case Studies - Overall Damage and ε Over the façade

This method involves calculating the total area of damaged elements across the façade and plotting the corresponding ε_t values. In some cases, ε_t at 99% is used as a representative metric, commonly referred to as the “characteristic strain.” Unlike the definition in Yiu et al. (2017), where characteristic strain is defined as the strain below which a certain percentage of the wall area does not exceed, here it is calculated by first sorting all ε_t values from highest to lowest and excluding the top 1% of extreme values. The 99th percentile value is then taken from the remaining distribution to avoid over-representing highly localised or spurious peaks. This approach enables a more stable assessment of global façade damage, while still aligning with the benchmark categorisation shown in Table 2.4, as used also for the LSTM method.

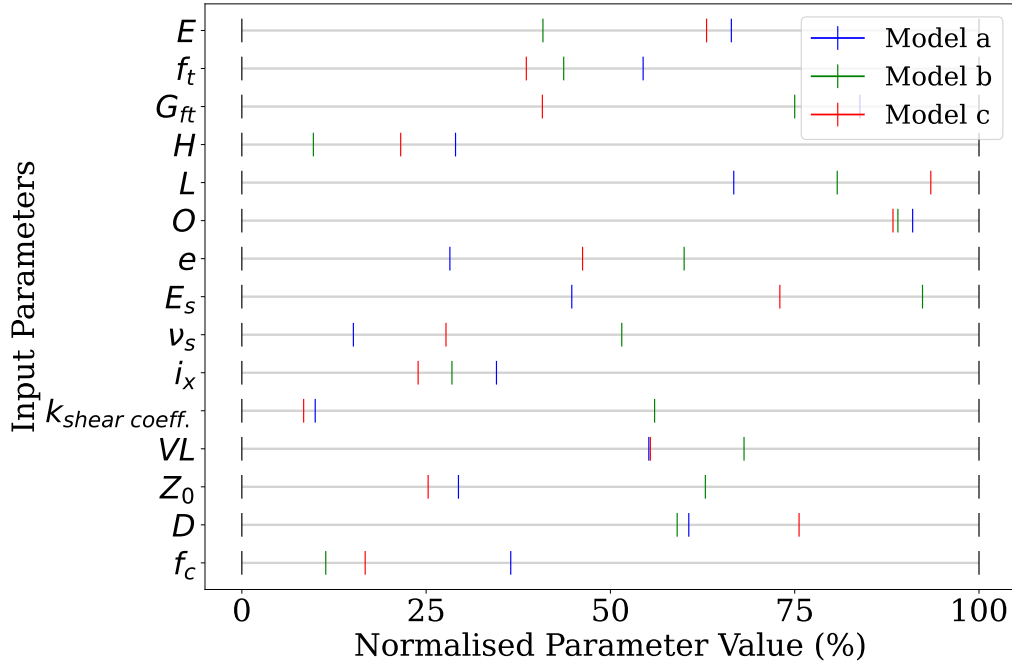


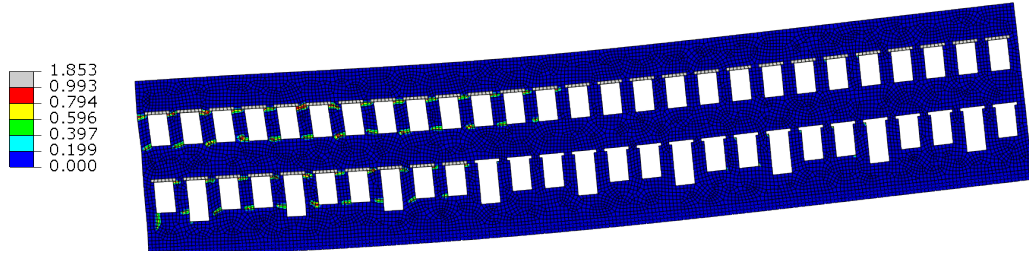
Figure 6.10: Input parameter distributions (in normalised terms) for the models used for the global damage evaluation of the selected case studies in Figure 6.11.

Below is an example of three selected structures with varying levels of overall damage, where values of ε_t^{ck} are used to identify elements undergoing non-linear behaviour. This value is measured at the element centroid, as required during the early stages of ‘.odb’ file postprocessing (Section 6.1). Figure 6.11 illustrates the distributed damage across the wall façades. Figure 6.11a corresponds to minimal or no damage, Figure 6.11b represents moderate damage, and Figure 6.11c shows significant or considerable damage. The exact strain values, percentages of damaged elements, and corresponding model input parameters are provided in Figure 6.10 and Table 6.5, respectively.

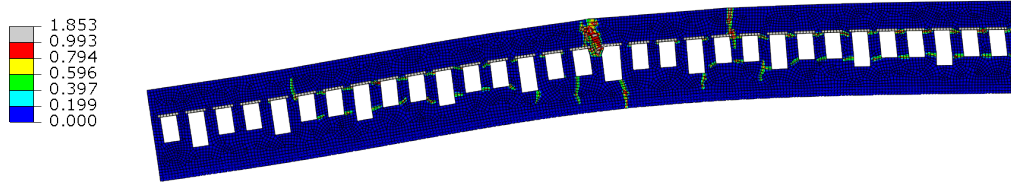
Table 6.5: % of damaged areas, and ε_t values of Figure 6.11 case studies.

Damage Evaluation	Model a	Model b	Model c
Damaged Area (%)	1.93	4.62	9.56
ε_t (Category)	0.0058 (Severe)	0.0407 (Severe)	0.0442 (Severe)
ε_t at 99% (Category)	0.0012 (Slight)	0.0042 (Severe)	0.0074 (Severe)
ε_t at 95% (Category)	0.00007 (Negligible)	0.00014 (Negligible)	0.00138 (Slight)

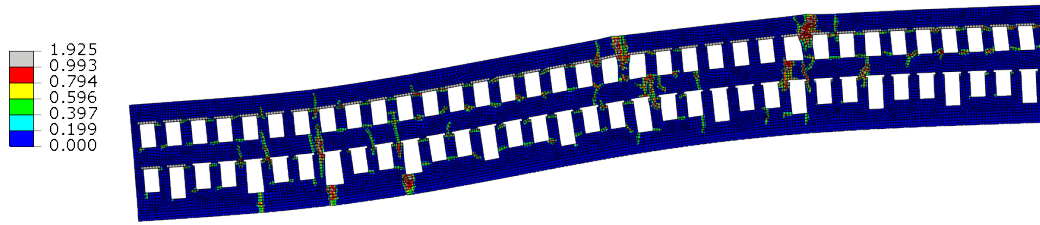
The area percentages correspond to a weighted average, calculated by multiplying the damage variable by the element’s area. Since the areas of the elements compris-



(a) Low to no damaged elements.



(b) Moderate numbers damaged elements (with clustering).



(c) Significant numbers of damaged elements including both clustering and spread.

Figure 6.11: Evaluating damage according to damaged elements, illustrating DAMAGET contours on building façade.

ing the wall are not all identical, this approach provides more accurate calculations than a simple averaging of ε_t values. This method can be considered a global approach to building damage assessment, as it does not explicitly identify the location of the damaged areas or their spread but rather provides an overall estimate for the entire façade, unless additional improvements are implemented.

In this section, the difference between local and global damage methods is examined, each describing a different aspect of model deformation. Initially, each metric appears to provide significantly different levels of damage from one another, even for the same model. This is particularly evident when damage is categorised according to criteria in the literature. For example, a building that incurs high values

of s or t may not necessarily exhibit severe discontinuities in the building material, and therefore, no cracking may be observed.

Ideally, this necessitates the use of a holistic damage assessment framework (similar to Giardina (2013)), in which considerations for all of the above metrics are combined or correlated. Depending on the function and use of the building, prioritisation of one method over another may also be necessary. Nonetheless, simply illustrating and evaluating each metric of building damage independently still enables the development of a broad, rich-data, and diverse outlook on the risk induced to buildings exposed to tunnelling.

6.2 Summary of Methods For All Case Studies

While the previous observations focused on identifying buildings with the highest damage values relative to their respective damage criteria (i.e., selecting the most severely damaged building based on its specific damage model, local, global, or otherwise), a brief presentation is provided of the different levels (categories) of all damage variables for each of the previously presented models, compiled into one comprehensive table. A summary of these results is presented below in Table 6.6 for the local and global damage assessment metrics, and Table 6.7 for the analytical-global and numerical local-global damage metrics.

It is noted that some consistencies and contradictions are observed across the damage classifications of various models. For instance, model (a) from the local assessment section exhibited significant levels of induced damage across all damage variables, which is consistent across the different methods. In contrast, model (b) from the LSTM method presents a notable contradiction. While local, global, and numerical local-global evaluations indicate substantial damage, the LSTM method classifies it as negligible. This discrepancy may stem from the limited input parameters and the underlying assumption of linear behaviour in the LSTM method,

Table 6.6: Categories of damage for the models presented earlier as case studies for each damage criteria in comparison to their global categorisation of damage (Part 1).

Model	Max. Crack Width	s	t	β	$\varepsilon_{h,\max}$
Local a	Very Severe	Moderate	Moderate	Very Severe	Very Severe
Local b	Moderate	Slight	Very Slight	Moderate	Moderate
Local c	Slight	Slight	Very Slight	Very Slight	Very Slight
Global a	Very Slight	Moderate	Moderate	Negligible	Negligible
Global b	Moderate	Slight	Slight	Moderate	Moderate
LTSM a	Moderate	Slight	Very Slight	Moderate	Moderate
LTSM b	Moderate	Slight	Very Slight	Moderate	Moderate
Area-Strain a	Very Slight	Slight	Very Slight	Negligible	Negligible
Area-Strain b	Moderate	Slight	Very Slight	Slight	Slight
Area-Strain c	Moderate	Slight	Slight	Slight	Slight

Table 6.7: Categories of damage for the models presented earlier as case studies for each damage criteria in comparison to their global categorisation of damage (Part 2).

Model	$\varepsilon_{bor\ d}$	Damaged Area	ε_t	ε_t @ 99%	ε_t @ 95%
Local a	Very Severe	6.51	Very Severe	Very Severe	Slight
Local b	Very Severe	2.93	Very Severe	Very Severe	Negligible
Local c	Very Severe	0.92	Very Severe	Negligible	Negligible
Global a	Very Severe	0	Negligible	Negligible	Negligible
Global b	Very Severe	6.87	Very Severe	Very Severe	Slight
LTSM a	Moderate	2.48	Very Severe	Very Severe	Negligible
LTSM b	Negligible	3.28	Very Severe	Moderate	Negligible
Area-Strain a	Very Severe	1.93	Very Severe	Slight	Negligible
Area-Strain b	Very Severe	4.62	Very Severe	Very Severe	Negligible
Area-Strain c	Very Severe	9.56	Very Severe	Very Severe	Slight

which can overlook the effects of highly non-linear material responses. In particular, although the structure may not undergo pronounced bending or displacement, the material could still experience significant damage due to high tensile strains, captured by advanced non-linear FEM but missed by simplified approaches. These findings underscore the importance of a comprehensive damage assessment framework that incorporates multiple evaluation methods to achieve a more accurate and reliable understanding of tunnelling-induced building damage.

To address the variability observed across different damage metrics, a structured decision-making approach can be adopted. One effective strategy would involve defining clear selection criteria based on the project's specific requirements and

objectives. These criteria might include the sensitivity of metrics to different damage mechanisms, the comprehensiveness of capturing local versus global effects, the ease of metric interpretation, and the compatibility with available data inputs. For example, if the primary concern is local structural integrity, metrics sensitive to crack widths and strain distributions would be prioritized. Conversely, if broader structural stability or serviceability is of greater importance, global or numerical local-global metrics may be more relevant. Additionally, employing weighted scoring or Multi-Criteria Decision Analysis (MCDA) Valerie and Theodor (2002) could systematically quantify the relative importance of each metric, aiding in the selection of the most appropriate damage assessment method.

6.3 Distribution of Damage Variables

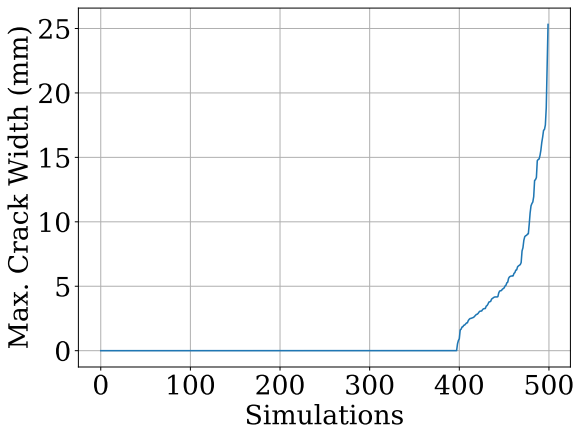
Since the purpose of the FEM models is the training of ML algorithms, a fundamental question must be addressed first: how well are the output values distributed across the full range of each damage variable? In other words, do the output variables (i.e., local, global, etc.) sufficiently cover the significant regions corresponding to different levels of building damage? Is the data distribution adequate for effective training of ML algorithms, particularly in light of clustering and overfitting concerns?

In this section, the distribution of damage values across the entire set ($N = 500$) of FEM models is examined, considering both local and global damage metrics. The natural clustering (or skewing) and highly imbalanced distribution of data are then highlighted. A novel technique is proposed to redistribute the data, thereby reducing training bias and mitigating the risk of overfitting in ML models. This approach is described in detail in Section 6.4, from which it is concluded that it directly improves predictions for lightly non-linear outputs and also provides recommendations to increase the number of samples in cases involving more highly

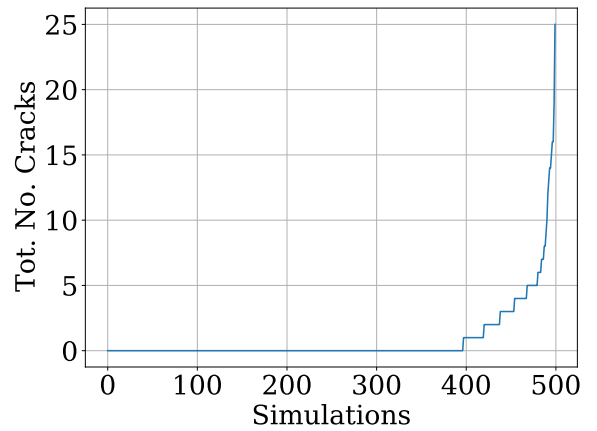
non-linear outputs.

In Figure 6.12, the distribution of local output variables (i.e., Maximum Crack Width and Total Number of Cracks) is plotted against the number of total simulations, referred to here as the original dataset composed of $N = 500$ samples, generated as previously discussed in Section 5.2, and arranged in ascending order. Initially, this excludes the dataset iterations addressed later in Section 6.4. It is clearly visible that a large number of simulations did not exhibit any non-linear response based on the local criteria.

In fact, only around 20% of the data (≈ 100 simulations) produced output values other than zero for the damage variables. This means that only one out of five combinations of input parameters in the original MLH sampled dataset (Section 5.2) resulted in any measurable damage, i.e., an output that is not zero or null. These non-zero outputs are inherently necessary for effective ML training and for conducting input parameter SA studies aimed at evaluating system response. Therefore, the remaining 80% (approximately four out of five samples) converged to a single “null” output, i.e., no damage. Nonetheless, a “null” or zero output still holds value, provided that the ratio of such samples is balanced relative to the total number of simulations, as it helps identify combinations of input parameters that do not lead to damage.



(a) Max. Crack Width.



(b) Tot. No. Cracks.

Figure 6.12: Original distributions of local damage metrics with simulations.

Moving on to the global assessment metrics, namely, s , t , β , and $\varepsilon_{h,\max}$, the distribution of these output variables across simulations is illustrated in Figure 6.13. The highly non-linear outputs, such as β (Figure 6.13c) and $\varepsilon_{h,\max}$ (Figure 6.13d), exhibit greater skewness in their data distributions compared to s (Figure 6.13a) and t (Figure 6.13b). In particular, a noticeable increase in the values of β and $\varepsilon_{h,\max}$ is observed around the 20% (1:5) mark, as previously seen in the case of local metrics. This comes as no surprise, as both parameters are linked to the initiation and propagation of cracks, leading to their exponential growth. The segmentation of walls into separate blocks contributes to higher θ values and significantly increases ΔL , both of which are key contributors to increasing β and $\varepsilon_{h,\max}$, respectively.

In contrast, s and t are found to be less affected by crack initiation, as these outputs are more dependent on the shape of the settlement trough from SSI, influenced by parameters such as e , VL, D , and Z_0 , rather than the building's non-linear properties. Similarly, initial observations of these trends suggest the need for rebalancing techniques. Although not all output variables indicate a sharp increase in value, the primary interest is in the highly non-linear outputs. In these cases, rebalancing is expected to significantly improve model performance, whereas its effect on less non-linear variables is minimal. This will be further illustrated later in this chapter.

Furthermore, Figures 6.14 and 6.15 illustrate the output distributions of $\varepsilon_{(b \text{ or } d), r}$ from the LTSM method and of the damaged elements of the wall alongside ε_t values, plotted against simulation numbers, respectively. It is immediately apparent that the distribution of $\varepsilon_{(b \text{ or } d), r}$ values resembles those of the global variables s and t . This similarity arises because the LTSM model is also based solely on the material's elastic properties and is fully analytical, requiring no FEM outputs.

Still, a notable increase in the rate of change is observed after approximately 70–80% of the simulations. This is attributed to the natural exponential rise of $S_{v,\max}$ when the results are reordered in ascending order against the sample points (Figure 6.14b). This pattern also applies to all other damage metrics. On the other

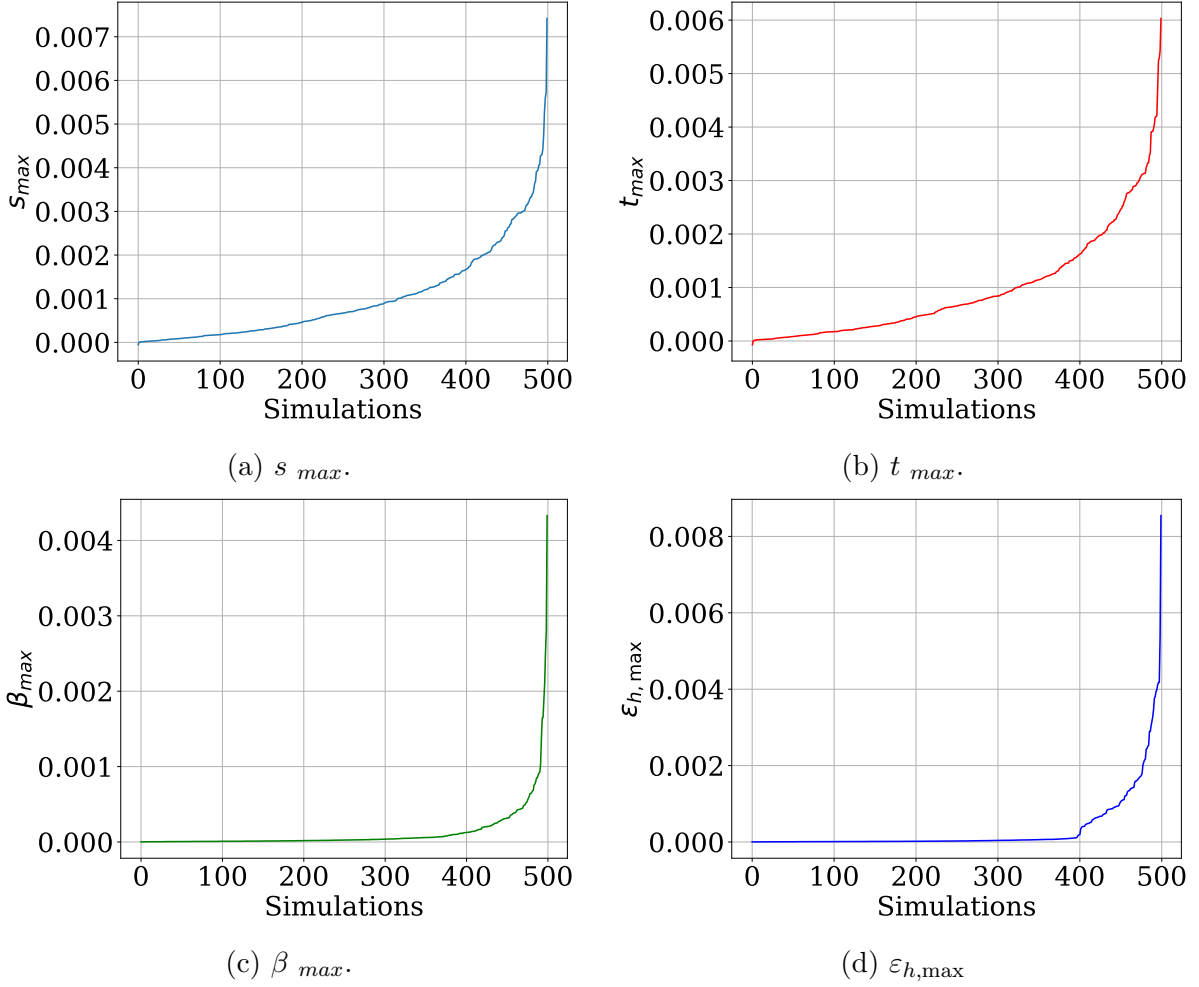
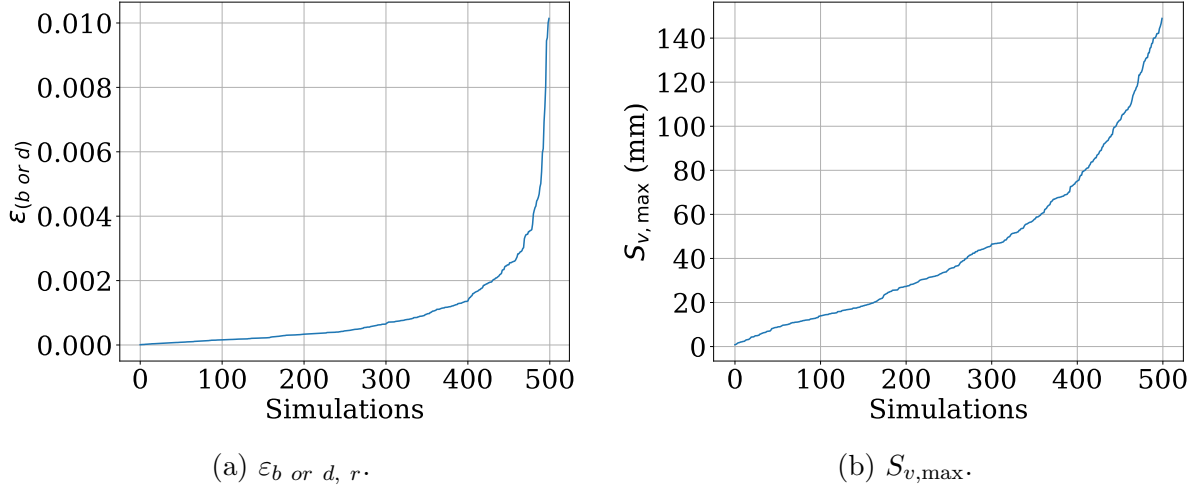
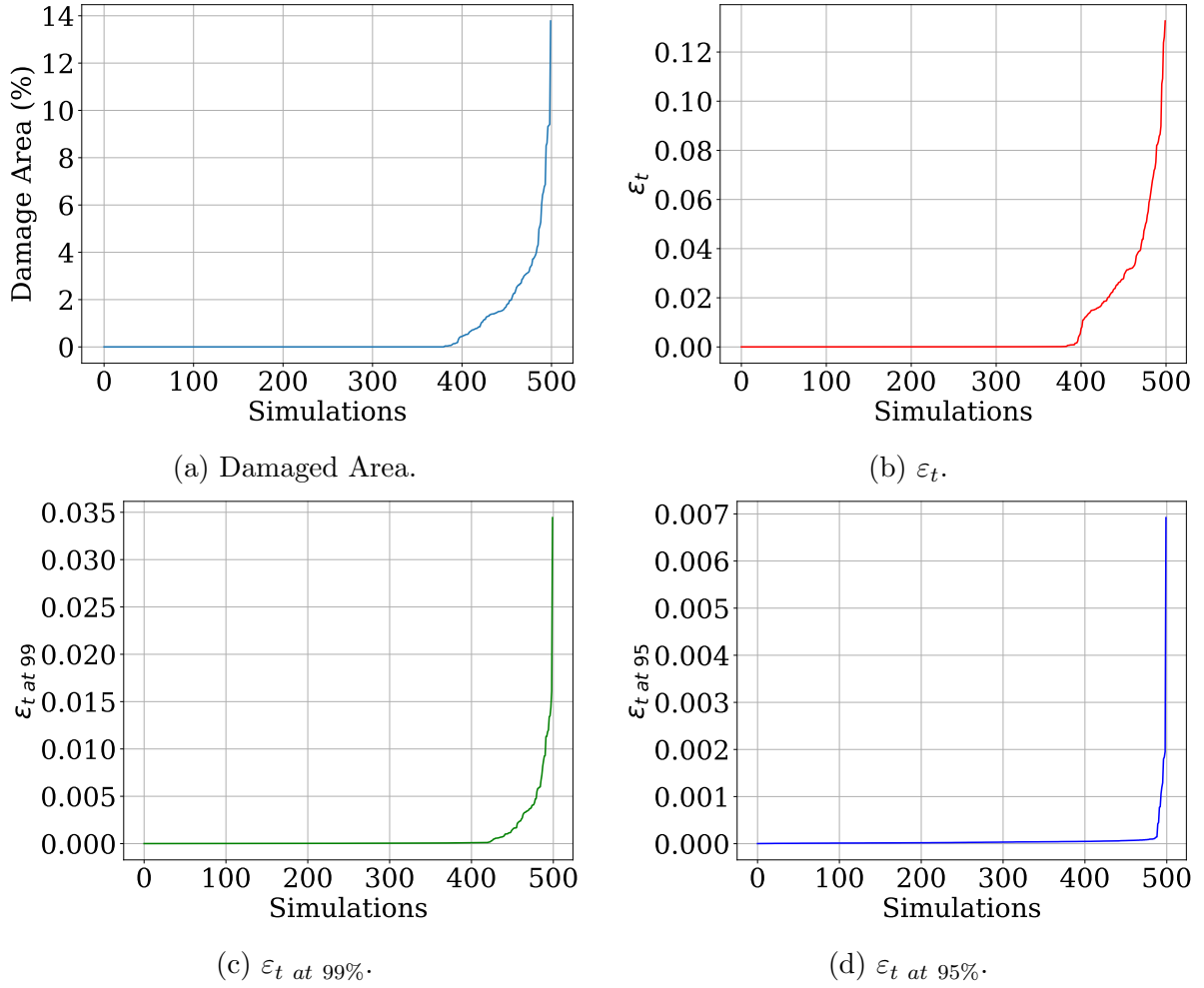


Figure 6.13: Initial distributions of global metrics with simulations.

hand, the strain values (Figures 6.15b to 6.15d) and the damaged elements of the façade (Figure 6.15a) show a distribution similar to that observed in the local metrics, namely, Maximum Crack Width and Total Number of Cracks. Both the damaged elements and ε_t follow a comparable trend, showing an approximate 1:5 ratio of damaged models. These metrics are typically triggered and recorded as soon as cracking occurs, regardless of the crack magnitude. A similar pattern is observed in the distribution of ε_t at 99%. In contrast, ε_t at 95% appears in only 2% of the total simulations. While this value carries no specific technical meaning, it simply indicates how few buildings experience substantial cracking, both in magnitude and frequency, beyond a 5% threshold derived from ε_t .

Even a 50-50 use of Null combined with meaningful data (i.e., more useful for the


 Figure 6.14: Initial distributions of LSTM damage criteria and $S_{v,\max}$.

 Figure 6.15: Initial distributions of additional metrics (Damaged Area and ε_t) with simulations.

training of ML), Even with a 50-50 mix of “null” and meaningful data (i.e., data more suitable for the training of ML models), adopting the current distribution state means that a significant portion of simulations may not contribute effectively. In fact, they could negatively impact the overall training process due to the previously discussed issues of bias and overfitting. Additionally, at this stage, it is still unknown which input parameters have the greatest influence on the outputs. Current assumptions are based primarily on engineering judgement and existing literature.

Hence, in the following section, the SA of input parameters is carried out as a first step toward identifying the most influential factors. This step guides the novel approach in generating a more targeted sample space that focuses on highly non-linear output variables.

6.4 An Iterative Procedure of Data Rebalancing

From the previous section, it was concluded that the distribution of output variables based on the current MLH samples demonstrates significant skewness in data, resulting in major imbalance. It was observed that approximately 1 in 5 simulations of highly non-linear metrics were useful, while the remaining 80% of total simulations resulted in similar outcomes, consuming significant computational resources and time, providing little benefit to the overall usefulness of these results due to similarity in values.

6.4.1 Implementation of the Iterative Procedure

To improve the distribution and representation of damaged cases in the dataset, the iterative procedure is implemented on the synthetic datasets produced by the FEM models, as described in Section 6.1. This approach aims to increase the proportion of simulations that result in meaningful local and global damage outputs, thereby

reducing the dominance of majority-class (i.e., undamaged) cases and avoiding unnecessary computations caused by highly clustered output distributions.

The RFR ensemble model from `sklearn` in Python and the `RandomForestRegressor` model, with `n_estimators` = 1000 to improve the model's stability and identify the most influential parameters through feature importance for the SA. A threshold $\phi = 5\%$ is used to distinguish important parameters from non-important ones. The analysis is performed on an initial set ($N = 500$) of data points, using two ensemble models trained separately: one with local output values and the other with global output values. The selection of output variables for the SA is based on the highest expected levels of non-linearity. Specifically, $\varepsilon_{h,\max}$ is used for the global assessment, and Maximum Crack Width is used for the local assessment. The SA results of both models are shown below in Figure 6.16. From this figure, the most important parameters can be identified. Nonetheless, significant error bars are initially observed, likely due to the initial number of simulations being insufficient coupled with the presence of high levels of data clustering. Applying techniques of over- or under-sampling could potentially enhance the initial estimates of SA, though this was not further investigated.

By performing SA, parameters that would benefit from improved dataset quality can be identified by further analysing their statistical metrics, regions of interest, and then resampling for each iteration. Less influential parameters remain unchanged. Notably, for each iteration, SA is performed on the current dataset along with previous ones, including the original dataset. This approach avoids over-reliance on a single dataset and makes decisions based on the accumulated sample points, providing higher accuracy in SA predictions due to the increased number of simulations used and improved population of the minority classes of output variables.

Figure 6.16 illustrates the parameters with the highest impact using the sample points from the initial dataset. Figure 6.25a illustrates the SA of the global metric, indicating the most impactful parameters as follows (in order of importance): i_x ,

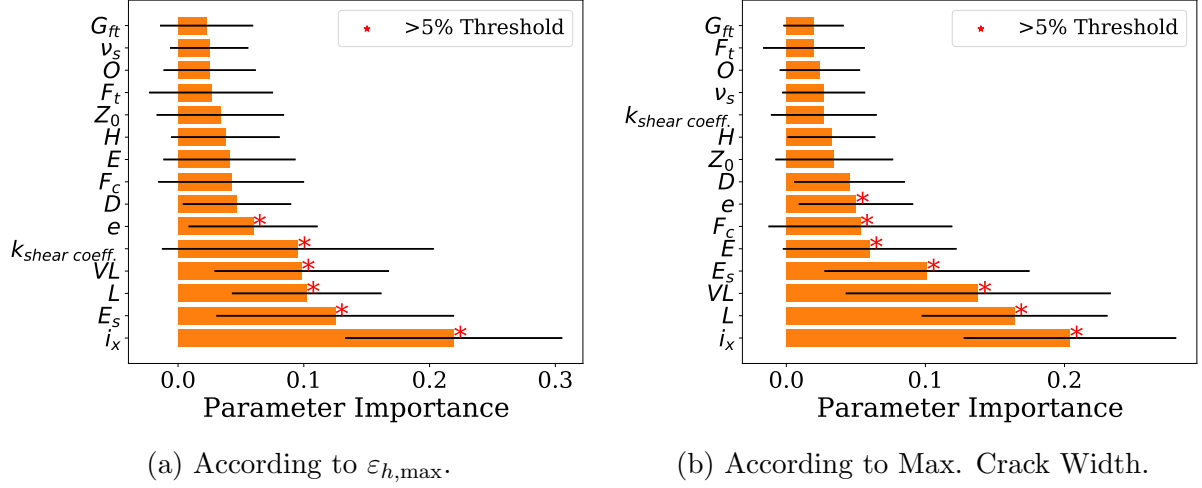


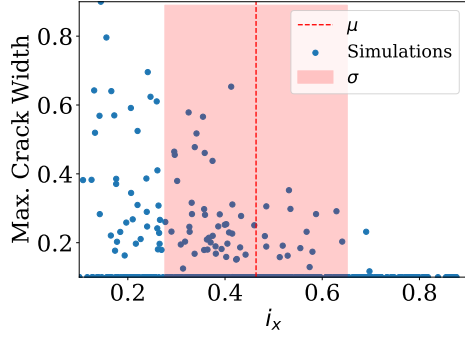
Figure 6.16: Importance score of input parameters using the RFR, totalling a sum of 1.

E_s , L , VL , $k_{shear\ coeff.}$, and e . In contrast, Figure 6.25b illustrates the SA of the local metric, revealing the most influential parameters in order of importance as: i_x , L , VL , E_s , E , f_c , and e . Notably, some parameters are important in both datasets, indicating their broad influence across different outputs.

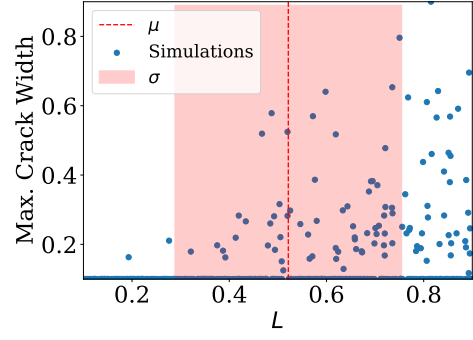
Step 2 - Statistical Exploration using μ_{X_i} and σ_{X_i} : Once the most important parameters are identified using the original dataset μ_{X_i} and σ_{X_i} are determined for the input values of each influential parameter. This is a foundational step in determining separate regions of parameter distributions (Step 3). The output distributions with respect to influential parameters are also examined, visually illustrating how outputs vary with increased values of influential parameters.

Figure 6.17 illustrate the distributions of Maximum Crack Width along ranges of values of the most influential parameters and for the original dataset. The axis values in these graphs are normalised between 0.1 and 0.9 for consistency. From these figures, clear trends of increasing or decreasing behaviour in damage with increased parameter values can be observed, depending on the parameter in question. For example, parameters i_x (Figure 6.17a), f_c (Figure 6.17f), and E (Figure 6.17e) are all generally following a similar trend, in which a decline in damage values is observed with increased parameter values (negative correlation), indicating that damage values associated with higher values of such parameters are of lower

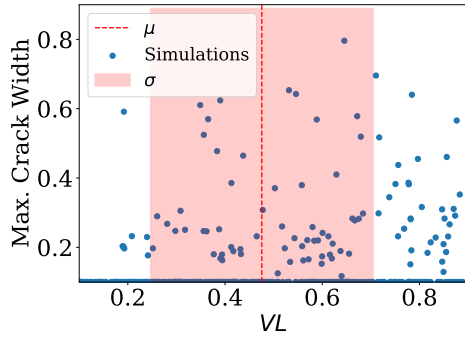
magnitude and vice versa. On the contrary, parameters such as L (Figure 6.17b), VL (Figure 6.17c), and E_s (Figure 6.17d) illustrate an increase in damage values at higher values of input parameters (positive correlation), indicating an opposite effect, in which lower parameter values introduced fewer Maximum Crack Widths and vice versa. Finally, observations of parameters such as e (Figure 6.17g) (although influential as per step 1) initially provide unclear behaviour (trend) on whether increasing or decreasing this input will introduce more damage. Nonetheless, a slight decrease in damage with increased input values is still observed, indicating a negative correlation between the two. This is further clarified with increased iterations as average determinations of Step 3 are applied, where regions of influential parameters are statistically determined by the μ_{X_i} measure.



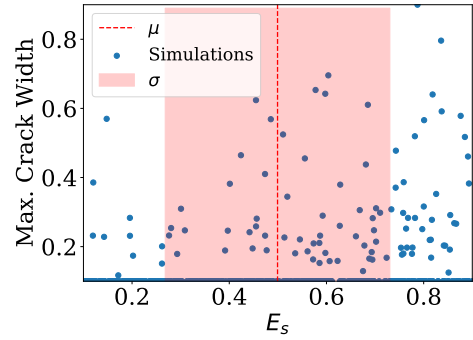
(a) i_x parameter.



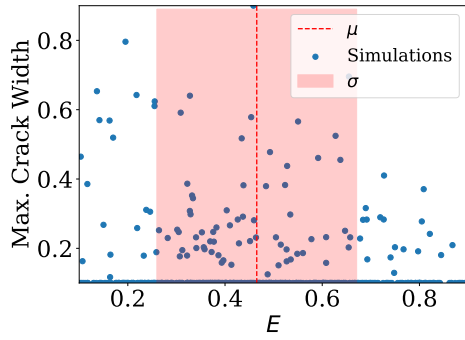
(b) L parameter.



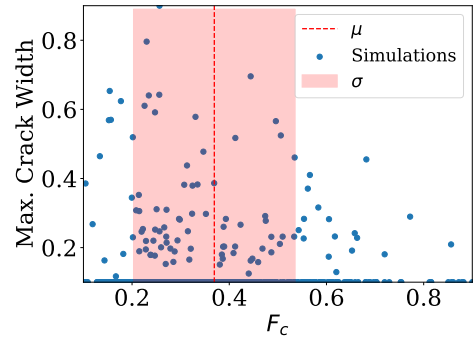
(c) VL parameter.



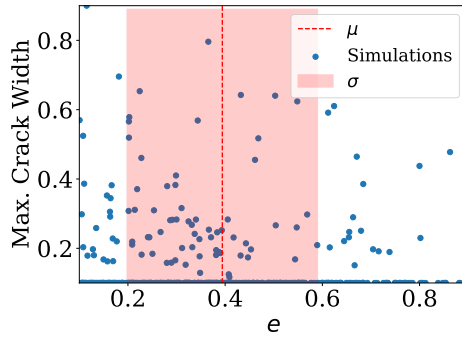
(d) E_s parameter.



(e) E parameter.



(f) f_c parameter.



(g) e parameter.

Figure 6.17: Damage distributions in normalised values of the Max. Crack Width against the most influential parameters and for the original dataset.

The results from the global output variable $\varepsilon_{h,\max}$ are presented in Figure 6.18, where the distributions of global damage are plotted against the most influential input parameters. Similarly to the local results, axis values are normalised for consistency. Consistent with the trends observed in the local output results, parameter i_x (Figure 6.18a) shows a decline in global damage values with increasing input values. This suggests that larger values of i_x , which contribute to the widening of the settlement trough, significantly reduce global building damage, indicating a negative correlation.

On the contrary, parameters such as E_s (Figure 6.18b), L (Figure 6.18c), VL (Figure 6.18d), and $k_{\text{shear coeff.}}$ (Figure 6.18e) demonstrate increased damage with higher input values. These show a positive correlation similar to their local counterparts, except for $k_{\text{shear coeff.}}$, which was not influential in the local results but shows a clear positive correlation here. Lastly, the e parameter (Figure 6.18f) also shows a negative correlation with global damage, more clearly than in the local case.

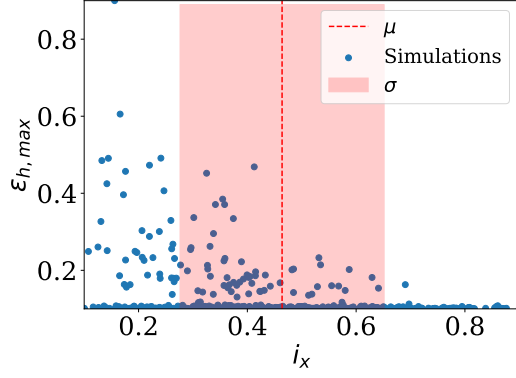
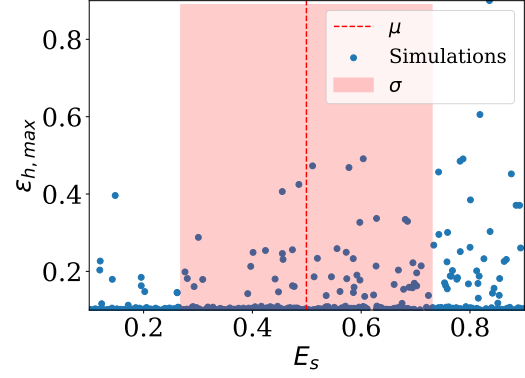
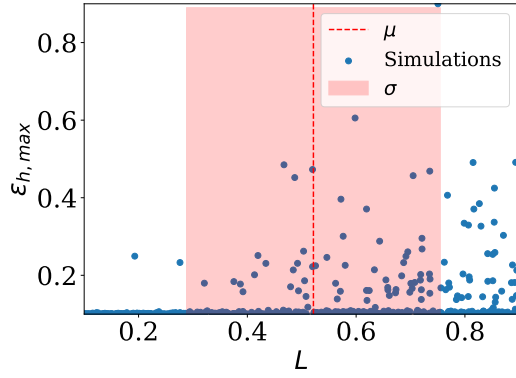
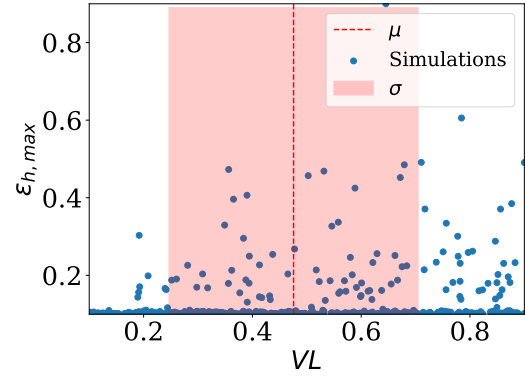
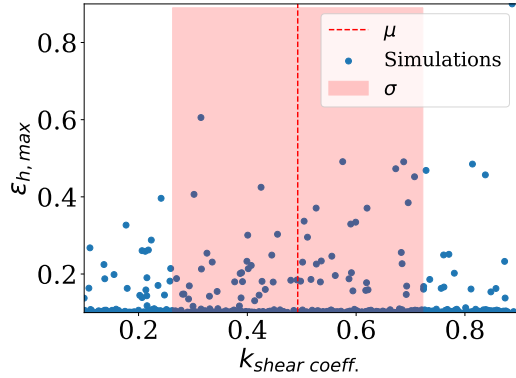
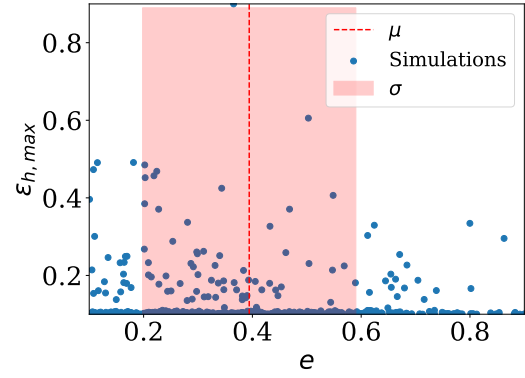

 (a) i_x parameter.

 (b) E_s parameter.

 (c) L parameter.

 (d) VL parameter.

 (e) $k_{shear\ coeff.}$ parameter.

 (f) e parameter.

 Figure 6.18: Damage distributions in normalised values of $\varepsilon_{h,max}$ against the most influential parameters and for the original dataset.

The mixed response between parameters over local and global damage metrics makes this investigation highly beneficial in understanding each parameter's overall impact on a selected output. In addition to helping identify regions of high influence, it aids in determining whether increasing or decreasing values of a given

parameter have a positive or negative correlation to building damage. This paves the way for refining the input space and incrementally targeting more influential regions of input parameters per iteration.

Step 3 - Identifying Regions of Interest: The final step before resampling is the identification of regions of interest. In Step 2, the distribution of parameters along with their respective μ_{X_i} and σ_{X_i} was presented. These two statistical measures are now used to divide the range of input values into three distinct regions:

- Below $\mu_{X_i} - \sigma_{X_i}$,
- Between $\mu_{X_i} - \sigma_{X_i}$ and $\mu_{X_i} + \sigma_{X_i}$,
- Above $\mu_{X_i} + \sigma_{X_i}$.

In a normally distributed curve, approximately 68% of all values fall within the central region (i.e., between $\mu_{X_i} - \sigma_{X_i}$ and $\mu_{X_i} + \sigma_{X_i}$), while the remaining 32% are split between the lower and upper ends. Although the sample points are not strictly normally distributed, they exhibit similar distribution patterns depending on the relative position of a given parameter, including the effects of correlations and constraints previously discussed.

To this end, the average output value $Y_{i \text{ Region (1 and 3)}}$, that is, the selected local metric “Maximum Crack Width” or global metric “ $\varepsilon_{h,\max}$ ”, is calculated for each of the three defined regions. Based on these average values, an ascending or descending trend is determined. By comparing these regional averages, the region of lower influence can be identified and then excluded from the input space domain of the relevant input parameter for the MLH resampling in the subsequent iteration. This strategy effectively targets input value regions with higher potential to induce building damage for the selected output metric. For both metrics, the influential parameters are combined into a single table. The average output for each region of each input parameter is evaluated, such that a single representative value per region per parameter is ultimately obtained.

Table 6.8: Average values (normalised) for each of $Y_{i \text{ Region } 1, 2 \text{ or } 3}$ of input parameter using μ_{X_i} and σ_{X_i} with % of total values falling within each region.

Parameter	Reg. 1	Reg. 2	Reg. 3	% Reg. 1	% Reg. 2	% Reg. 3
E	0.164	0.143	0.127	19.98	51.42	28.61
L	0.103	0.137	0.177	23.58	58.33	18.09
e	0.147	0.136	0.131	12.24	49.06	38.70
E_s	0.112	0.133	0.174	20.97	57.75	21.29
i_x	0.207	0.127	0.102	22.10	46.88	31.02
$k_{shear \text{ coeff.}}$	0.124	0.134	0.130	20.40	57.49	22.11
VL	0.110	0.136	0.164	18.42	57.23	24.35
f_c	0.150	0.145	0.134	12.86	41.61	45.53

As mentioned above, Table 6.8 illustrates the average normalised output values for both local and global damage metrics across the three regions defined by μ_{X_i} and σ_{X_i} for the most influential parameters identified in the first cycle of the iterative procedure. Additionally, the table shows the percentage of data points falling within each region. From this table, the following conclusions are drawn:

- Increasing the values of E , e , i_x , and f_c contributes to reducing the overall damage induced to buildings (negative correlation). Since the goal of the iterative procedure is to target parameter ranges more likely to result in damage, the region $Y_{i \text{ Region } 3}$ is removed for these parameters. Therefore, sampling in the next iteration will only include values from $Y_{i \text{ Region } 1}$ and $Y_{i \text{ Region } 2}$.
- In contrast, increasing the values of L , E_s , $k_{shear \text{ coeff.}}$, and VL results in greater building damage (positive correlation). Accordingly, $Y_{i \text{ Region } 1}$ is excluded from the input space for the next iteration to focus the sampling on regions more prone to inducing damage.

It is worth noting that the material parameter f_c is often overlooked in similar studies Giardina et al. (2015); Lourenco (1996), with the reasoning that buildings in such loading conditions do not undergo significant compressive stress. However, in this work, f_c emerged as influential due to its high correlation with other critical parameters, particularly E , which is widely acknowledged as an influential factor (Giardina et al. (2015)). Therefore, it is concluded that SA is clearly sensi-

tive to parameter correlations introduced by the MLH sampling approach, although this relationship was not further explored in this study. By repeatedly applying this iterative procedure, the proportion of simulations that result in building damage is significantly increased. This is achieved by systematically removing the least influential parameter regions, as determined by SA, thereby favouring the generation of non-linear behaviour and reducing unnecessary computational overhead through a structured and statistically grounded method.

Summary of Results from the Iterative Procedure.

The iterative process was repeated five times. Although no universal rule exists to determine the optimal number of iterations, five cycles were deemed appropriate based on improved distribution of output variables observed during the process. Table 6.9 presents the minimum value ranges of each input parameter across all iterations. If a parameter's minimum remained unchanged or slightly increased, it indicates that the parameter had little to no influence on the output damage metrics and was likely below the importance threshold ($\phi < 0.05$) identified in Step 1. Thus, these parameters remained unaltered in subsequent iterations.

Conversely, Table 6.10 displays the maximum value ranges per iteration. A significant reduction in maximum values for a parameter indicates its association with lower damage outputs, warranting its upper range to be trimmed to encourage more damage-prone configurations. If both the minimum and maximum bounds for a parameter remained unchanged, it suggests the parameter did not contribute meaningfully to the outcome and was therefore excluded from targeted refinement. However, if both bounds shifted across iterations, this indicates a changing trend in correlation caused by evolving sample populations, necessitating dynamic adjustment during the rebalancing process.

Tables 6.9 and 6.10 are further illustrated in Figure 6.19 using radar charts. These charts are normalised to ensure consistency in visualisation across iterations. They provide a visual representation of the gradual reduction in the range of values for

Table 6.9: Minimum values of input parameters used throughout iterations.

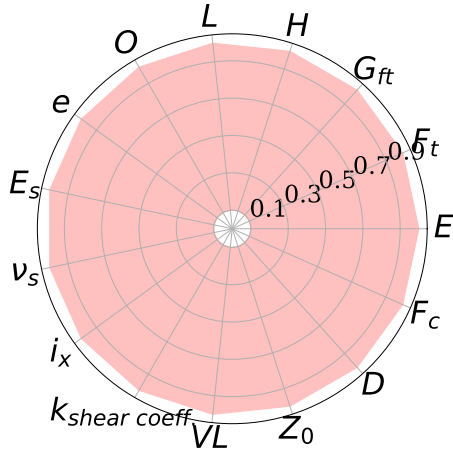
Parameter	Original Dataset	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Iter. 5
E	666	666	666	666	666	666
f_t	0.05	0.05	0.05	0.05	0.05	0.05
G_{f_t}	0.004	0.004	0.004	0.004	0.004	0.004
H	3.2	3.2	3.2	3.2	3.2	3.2
L	8.0	<u>20.3</u>	<u>27.0</u>	<u>32.8</u>	<u>37.6</u>	37.6
O	0	0	0	0	0	0
e	0.0	0.0	0.0	0.0	0.0	0.0
E_s	2.0	<u>54.0</u>	<u>82.4</u>	<u>108.3</u>	<u>131.4</u>	131.4
ν_s	0.1	0.1	0.1	0.1	0.1	0.1
i_x	3.6	3.6	3.6	3.6	3.6	3.6
$k_{shear\ coeff.}$	0.0001	<u>0.123</u>	<u>0.188</u>	0.188	<u>0.27</u>	0.27
VL	0.20	<u>1.08</u>	<u>1.61</u>	<u>2.04</u>	<u>2.29</u>	2.29
Z_0	7.3	7.3	7.3	7.3	7.3	7.3
D	4.9	4.9	4.9	4.9	<u>6.1</u>	6.1
f_c	0.5	0.5	0.5	0.5	0.5	0.5

Table 6.10: Maximum values of input parameters used throughout iterations.

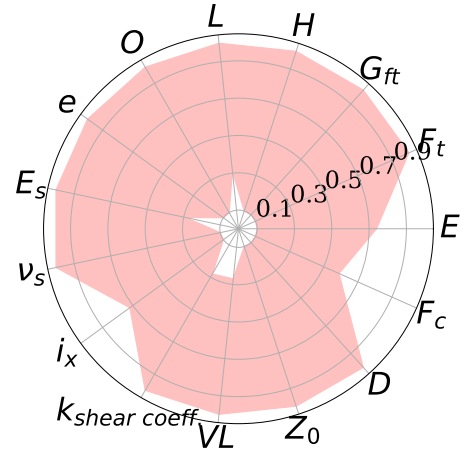
Parameter	Original Dataset	Iter. 1	Iter. 2	Iter. 3	Iter. 4	Iter. 5
E	9000	<u>6616</u>	6616	6616	6616	<u>4841</u>
f_t	1.47	1.47	1.47	1.47	1.47	1.47
G_{f_t}	0.03	0.03	0.03	0.03	0.03	0.03
H	20.0	20.0	20.0	20.0	20.0	20.0
L	60.0	60.0	60.0	60.0	60.0	<u>55.5</u>
O	30	30	30	30	<u>24.585</u>	<u>19.91</u>
e	45.0	<u>27.6</u>	27.6	27.6	27.6	27.6
E_s	250.0	250.0	250.0	250.0	250.0	250.0
ν_s	0.49	0.49	0.49	0.49	0.49	0.49
i_x	45.0	<u>32.2</u>	<u>26.3</u>	<u>22.9</u>	<u>20.4</u>	<u>19.4</u>
$k_{shear\ coeff.}$	0.60	0.60	0.60	<u>0.51</u>	0.51	0.51
VL	5.00	5.00	5.00	5.00	5.00	<u>4.06</u>
Z_0	90	90	90	90	90	90
D	17.6	17.6	<u>14.7</u>	14.7	14.7	<u>11.6</u>
f_c	48.2	<u>26.48</u>	<u>15.77</u>	15.77	15.77	<u>8.03</u>

each input parameter, starting from the original dataset (Figure 6.19a) up to the input space defined in iteration 5 (Figure 6.19f). As the parameter ranges shrink, it is important to note that the reduced area under the radar chart does not represent any objective or physical truth. Instead, the purpose of these charts is to visually communicate how the search space is progressively reduced through each iteration,

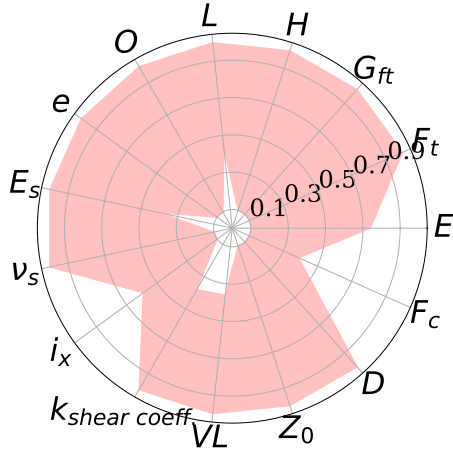
highlighting both the direction of reduction (i.e., whether the range is being trimmed from the minimum or maximum side) and the magnitude of this reduction (i.e., the proportion of the range that has been removed).



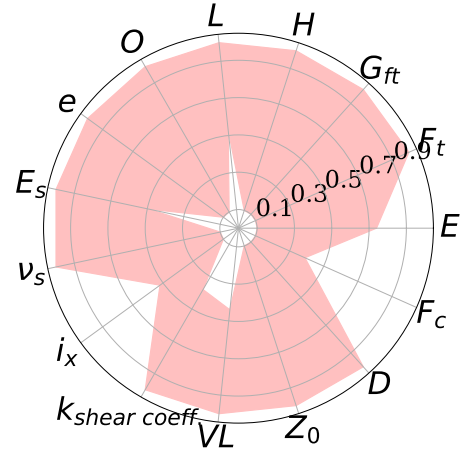
(a) Original Dataset ($N \approx 500$).



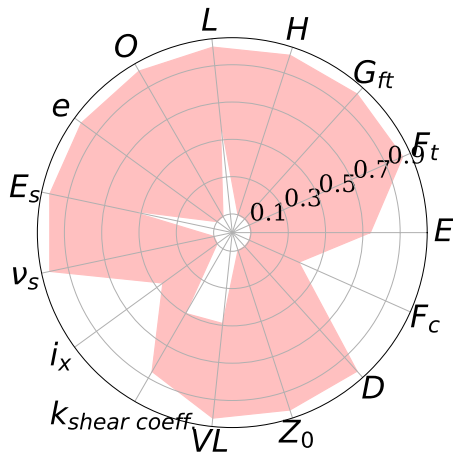
(b) Iteration 1 dataset ($N \approx 100$).



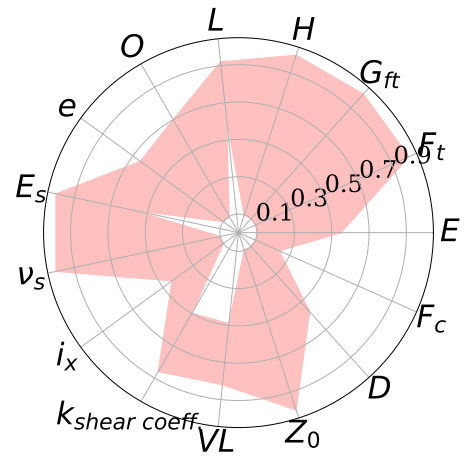
(c) Iteration 2 dataset ($N \approx 100$).



(d) Iteration 3 dataset ($N \approx 100$).



(e) Iteration 4 dataset ($N \approx 100$).



(f) Iteration 5 dataset ($N \approx 100$).

Figure 6.19: Distribution of input parameters between iterations using radar charts.

The progressive increase in building damage per simulation with increased iterations is illustrated in Figures 6.20 and 6.22. With each iteration, the rate of simulations undergoing significant non-linear behaviour increased, indicating the successful outcome of applying the iterative procedure. Figure 6.20 shows the distributions of the local metrics: Maximum Crack Width (Figure 6.20a) and Total Number of Cracks (Figure 6.20b) across different iterations. In both figures, the number of simulations exhibiting non-zero crack initiation and total crack counts increased by approximately 5–20% per iteration, corresponding to an additional 5 to 20 simulations per cycle, higher increases being observed in earlier iterations. By iteration 5 (not executed), more than 97% of simulations showed significant damage with at least one identified crack. Compared to the original dataset in Figure 6.12, a clear improvement is observed, with a notable reduction in the most populated region of the output variable, namely, the undamaged building cases.

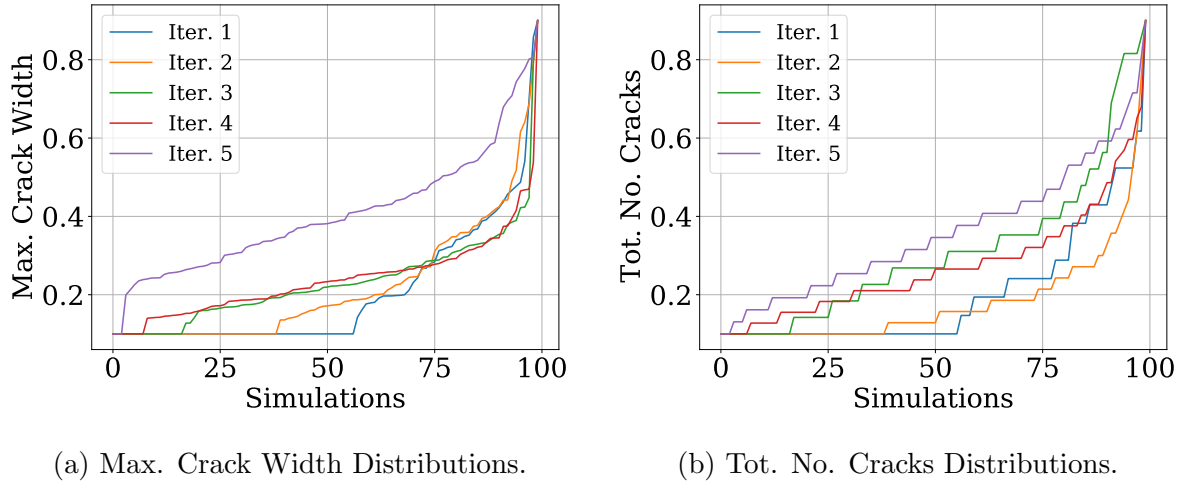


Figure 6.20: Distributions of local metrics with increased iterations (y-axis normalised between 0 and 1).

Similarly, Figure 6.21 illustrates the improved distributions of global damage metrics. In particular, Figure 6.21d, which shows the distribution of $\varepsilon_{h,\max}$, reveals a trend comparable to that of the Maximum Crack Width metric, with a steady increase in the number of simulations exhibiting damage. This similarity is expected, as $\varepsilon_{h,\max}$ is strongly associated with the initiation and propagation of cracks and is

the only global metric that primarily reflects the model's non-linear behaviour. In contrast, Figures 6.21a through 6.21c, which depict the metrics s , t , and β , exhibit less noticeable improvements with the iterative procedure. This is because these metrics depend largely on the shape of the settlement trough, which is governed by the elastic properties of the soil and the building. Since the iterative approach is specifically designed to enhance non-linear FEM-based responses, its influence on metrics driven by elastic behaviour is naturally limited.

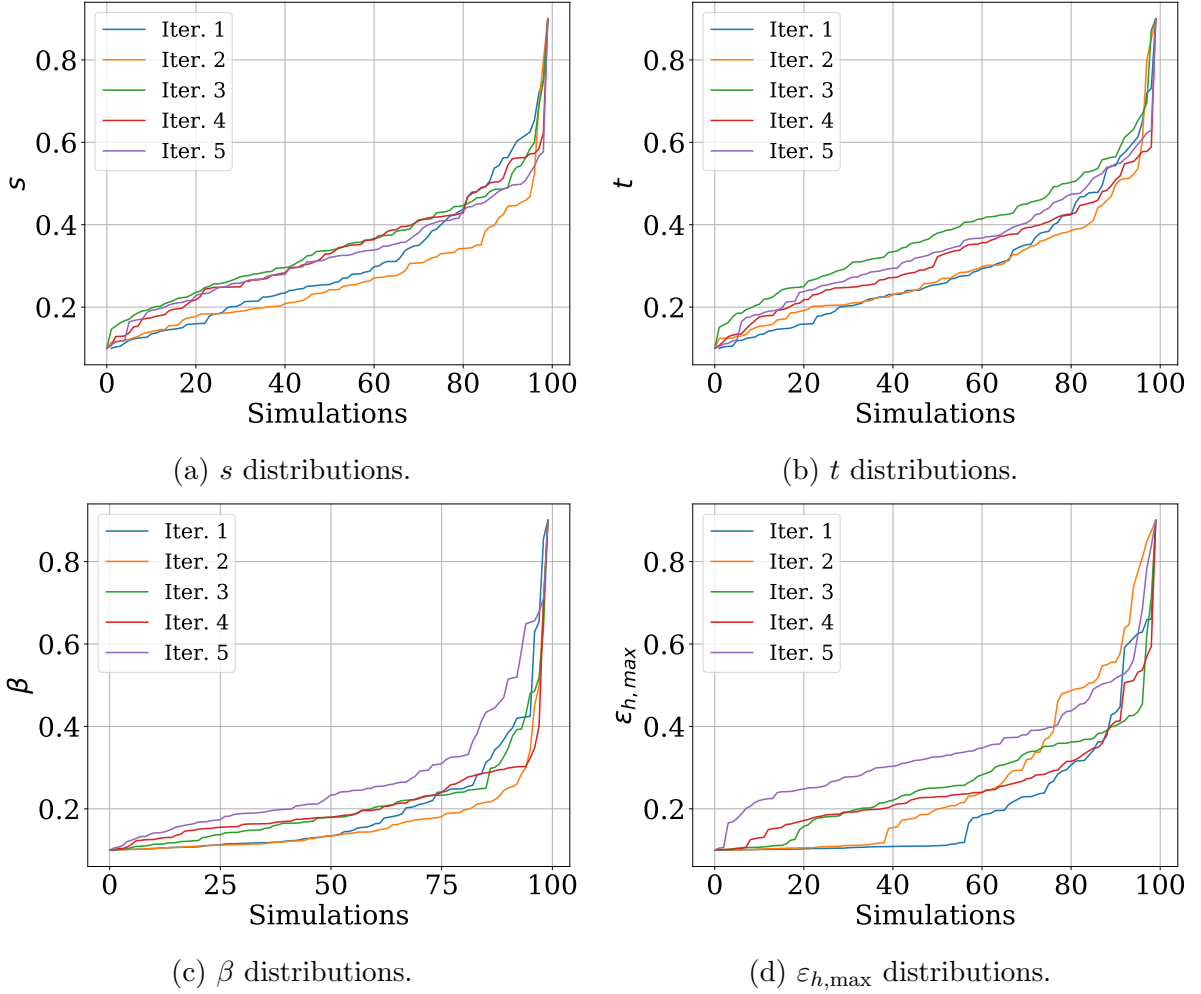


Figure 6.21: Distributions of local metrics with increased iterations (y-axis normalised).

Lastly, Figure 6.22 illustrates the distribution of other global and local damage metrics, including the LTSM method (Figure 6.22a). A noticeable increase in the rate of induced damage can be observed. Although the point of damage initiation remains unchanged, the overall increase in values, despite not involving FEM-based

outputs, is attributed to the screening-over-the-façade approach. Specifically, the increased values of parameter L , as influenced by the iterative procedure, allow for a broader selection of segments across the settlement trough. This enhances the likelihood of higher damage calculations, especially when considered in conjunction with other impactful parameters. On the other hand, Figure 6.22b shows the distribution of damaged elements based on the area metric. The damage initiation here mirrors that of previously discussed non-linear metrics, such as Maximum Crack Width and $\varepsilon_{h,\max}$. A similar trend is observed for the strain metric ε_t in Figure 6.22c. However, the improvements in damage distribution are less prominent for the ε_t at 99% metric, as shown in Figure 6.22d. The models generated in each iteration were not discarded but were instead cumulatively added to the original dataset. This growing pool of simulations, encompassing both early low-damage and later high-damage cases, served as the full input set for training the ML models in Chapter 7. As such, the iterative approach not only refined the input space but also ensured a diverse and well-balanced dataset for supervised learning.

While the iterative procedure was designed to enrich the dataset with more simulations showing non-linear behaviour, this does not introduce bias in the traditional sense. The goal was not to reflect the real-world distribution of damaged vs undamaged cases, but to enhance the ML model's ability to generalise across a wider range of outputs, especially rare but critical failure cases. As these improved distributions are observed as a result of the iterative procedure, potential benefits are anticipated during the training of ML models. These aspects are further explored in Chapter 7.

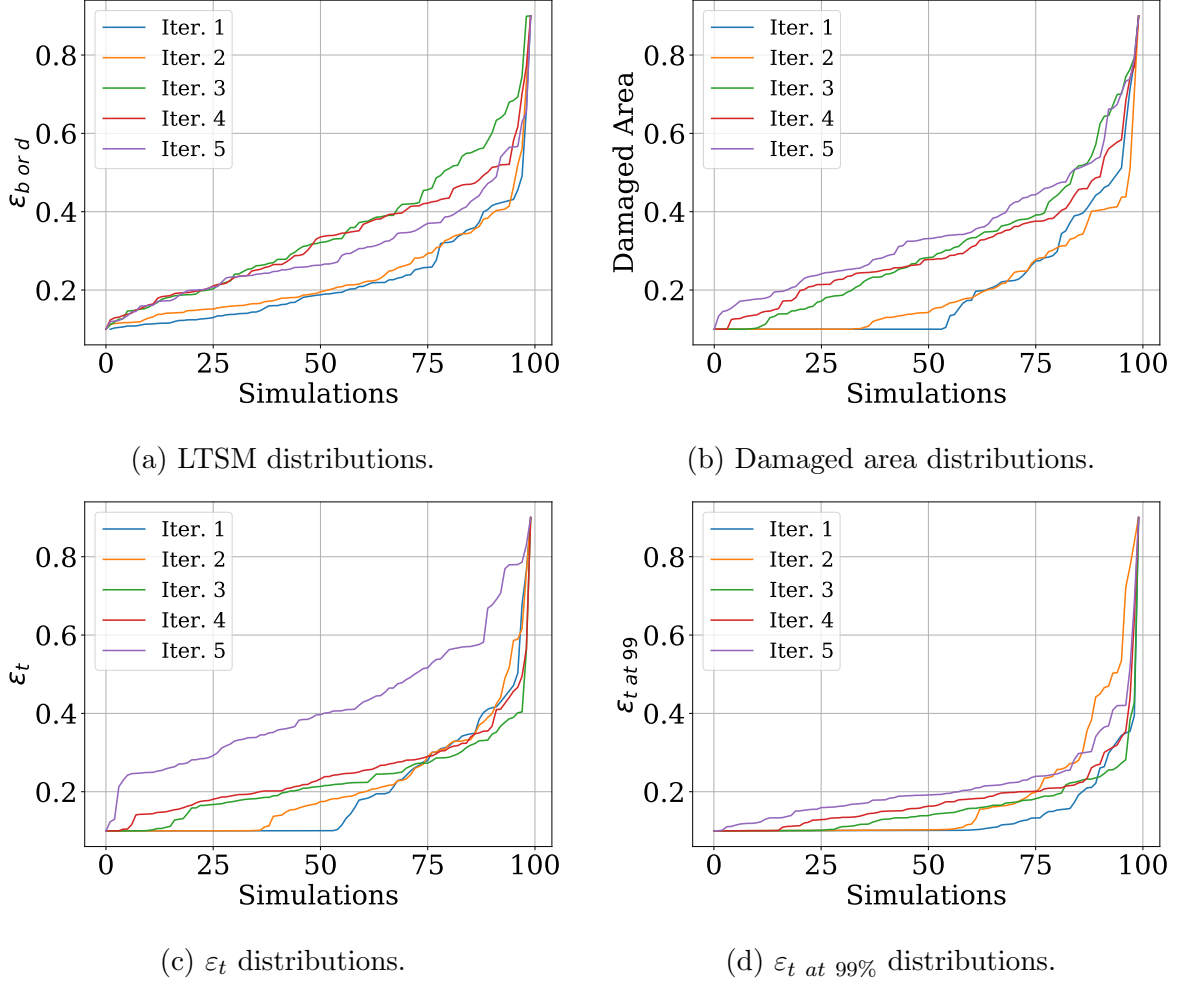


Figure 6.22: Distributions of additional global and local metrics with increased iterations (y-axis normalised).

Figures 6.23 and 6.24, on the other hand, present a comparison of input parameter distributions between the original dataset and the complete dataset (i.e., original plus iterations 1 to 5). These figures include only the parameters identified as highly influential (e.g., $\phi > 0.05$) during the SA of iteration 5. The comparison is made using both the local metric (Maximum Crack Width) and the global metric ($\varepsilon_{h, \max}$). Both μ_{X_i} and σ_{X_i} in these figures serve as strong indicators of parameter behaviour across iterations. Notably, μ_{X_i} shows meaningful trends. For example, parameter i_x , as shown in Figures 6.23a and 6.24a, exhibits a clear shift in μ_{i_x} toward lower input values. This trend is driven by the progressive reduction of the upper bound of the parameter range throughout the iterative process. Additionally,

a narrower spread of values is confirmed by the decrease in σ_{i_x} compared to the original dataset. Conversely, parameters such as L (Figures 6.23b and 6.24c) and VL (Figures 6.23c and 6.24d) show a noticeable shift in μ_{X_i} toward higher values. This behaviour results from a consistent reduction of the lower bound of their input ranges. Although these parameters also exhibit a reduced σ_{X_i} , their standard deviations remain slightly larger than that of i_x , reflecting the nature and extent of their iteration-driven range adjustments. These observations are similarly valid across the remaining influential parameters, where each μ_{X_i} and σ_{X_i} reflect the overall shift in parameter central tendency and the narrowing of value ranges before and after applying the iterative procedure.

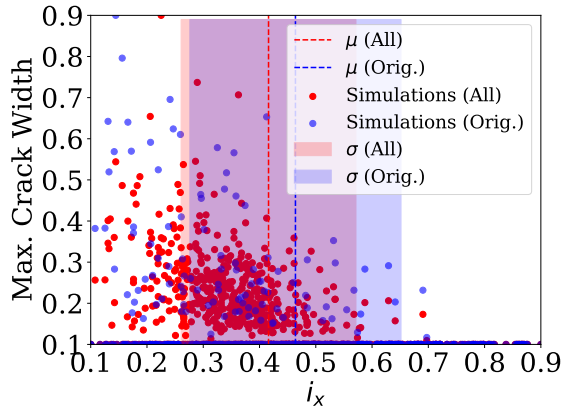
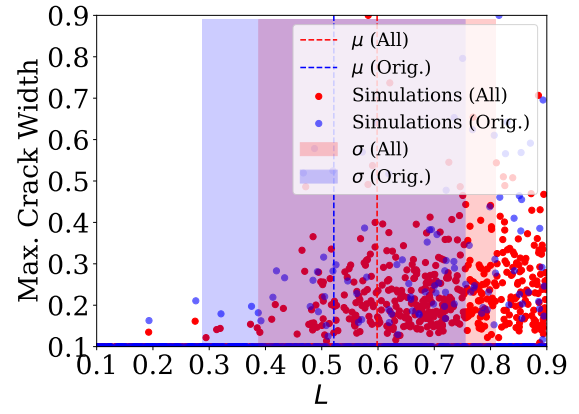
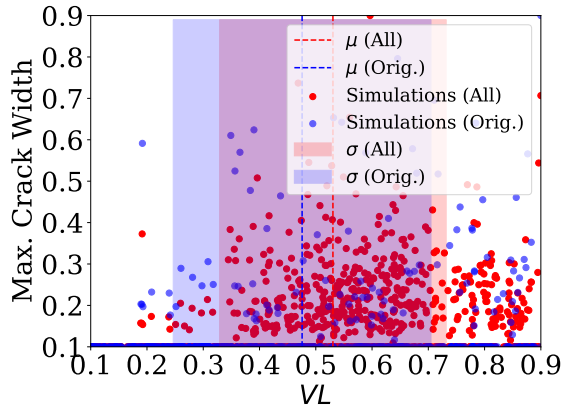
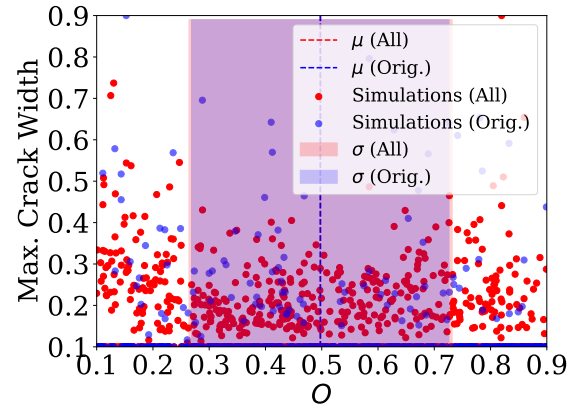
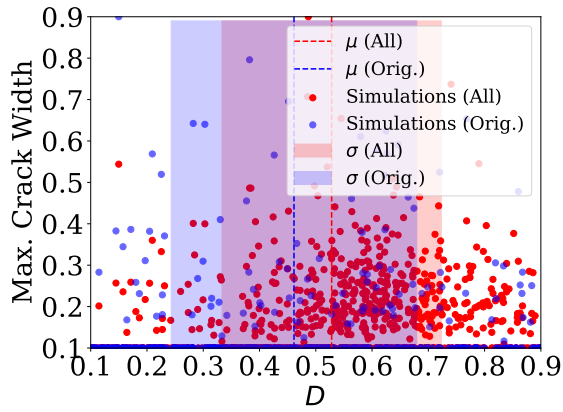
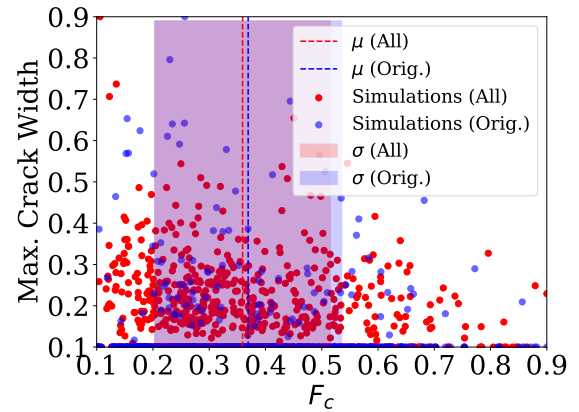

 (a) i_x parameter.

 (b) L parameter.

 (c) VL parameter.

 (d) O parameter.

 (e) D parameter.

 (f) f_c parameter.

Figure 6.23: Damage distributions in normalised values of the Max. Crack Width against the most influential parameters and for the entire dataset.

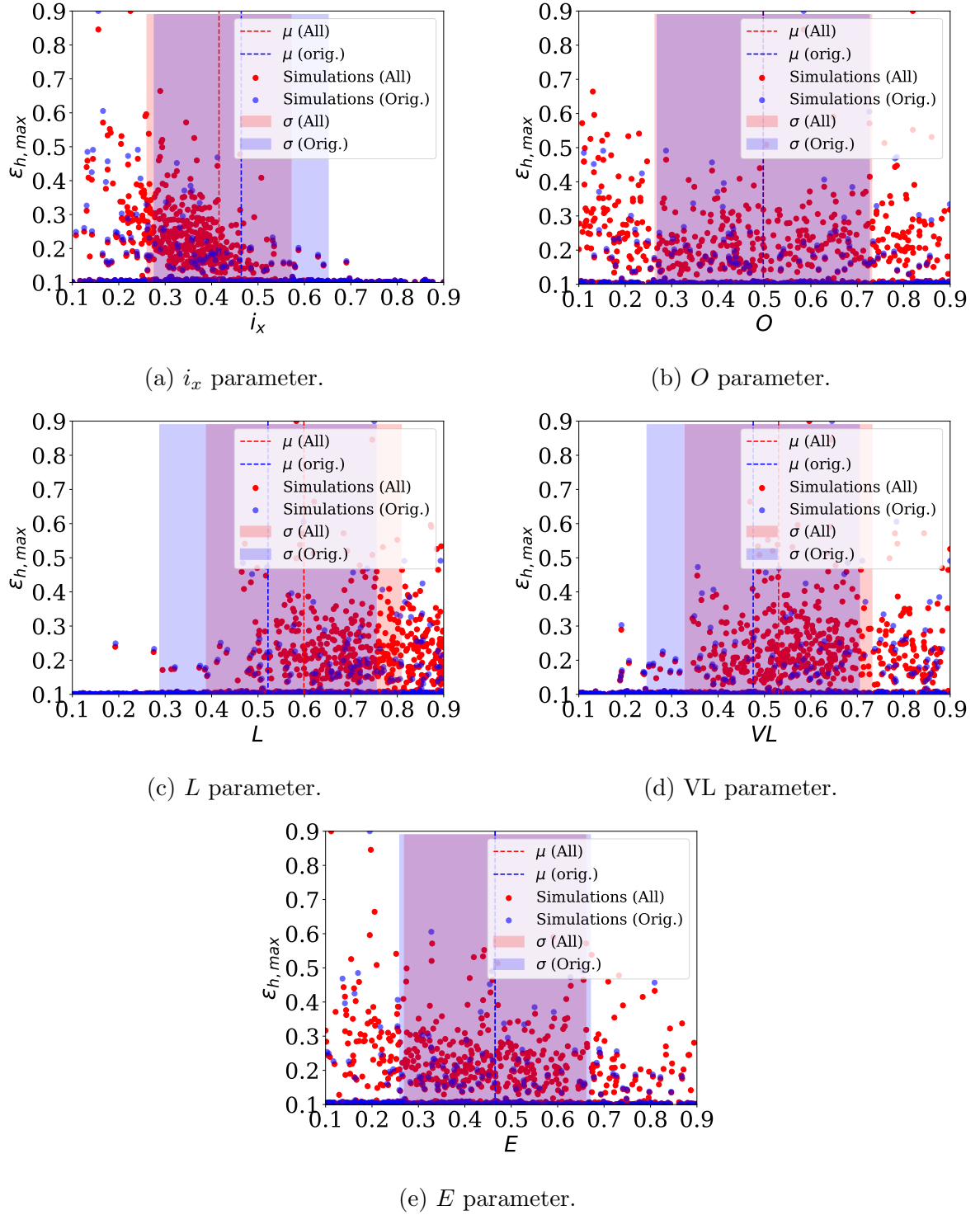


Figure 6.24: Damage distributions in normalised values of $\varepsilon_{h,max}$ against the most influential parameters and for the entire dataset.

The final SA performed on the entire dataset (i.e., the original dataset plus iterations 1 to 5) is illustrated in Figure 6.25. From this figure, it is concluded that after five iterations of model refinement using the iterative procedure described above, the list of the most influential parameters can be summarised as follows:

- For the local metric (Maximum Crack Width), in order of importance: i_x , L , VL, O , f_c , and D , respectively.
- For the global metric ($\varepsilon_{h,\max}$), in order of importance: i_x , L , VL, O , and E , respectively.

Parameters i_x , L , VL, and O consistently appear as the top four influential inputs across both metrics, taking first, second, third, and fourth place, respectively. In contrast, parameters such as f_c and D appear to be influential exclusively in the local metric after including all iterations, while E is found to be influential only in the global metric. Nonetheless, it is common for some parameters to fall below the importance threshold in certain iterations. The results shown in Figure 6.25 reflect only the outcome from iteration 5 of the SA. Therefore, these rankings may differ if the SA were conducted using earlier iteration results, such as iteration 2 or 3.

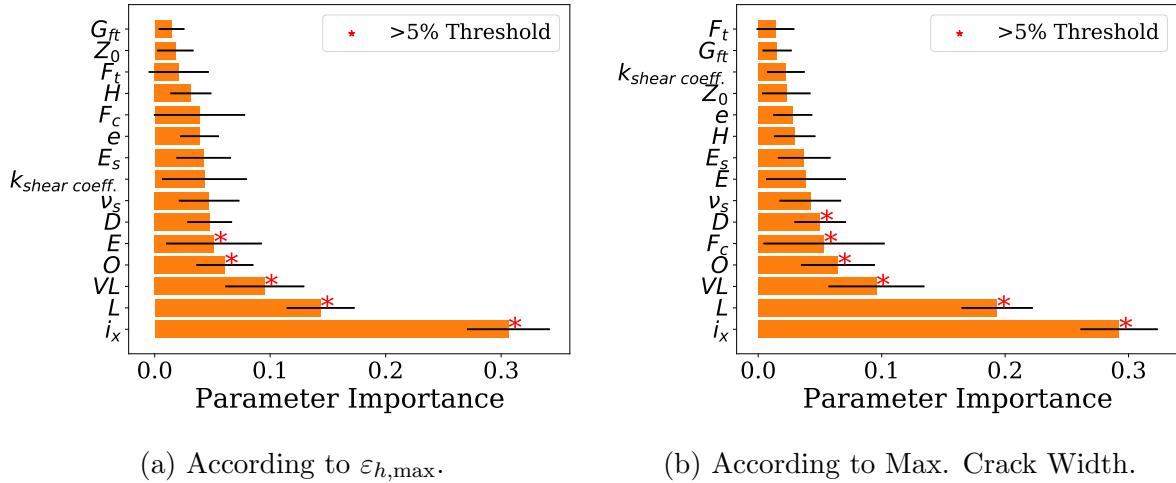


Figure 6.25: Importance score of input parameters using the RFR for the entire dataset.

6.5 Correlation Between Damage Metrics Across the Entire Set of Simulations

As mentioned earlier and outlined in the objectives of this thesis, this section briefly explores the correlations between various damage metrics, focusing on both potential correlations and similarities across the spectrum of output variables. For this purpose, a total of eight pairs of variables were analysed, and the four most highly correlated pairs (based on the R^2 criterion) as well as the four least correlated pairs were identified. The damage categories for each metric, along with the corresponding levels of damage, are also presented in Table 6.11. These categories align with the thresholds recommended in the literature discussed in Chapter 2.2.3. The analysis includes all simulations, incorporating both the original dataset and the results from iterations 1 to 5.

Table 6.11: Categories of damage of metrics according to literature.

Parameter / level of damage	1	2	3	4	5
Max. Crack Width	V. Sl.	Sl.	M.	Se.	V. Se.
Tot. No. Cracks ^a	Sl.	Sl.	Se.	Se.	Se.
s	Ne.	Sl.	M.	H.	H.
t	D. Lim.	No.	Mo.	R. A.	U. Lim.
β	Ne.	V. Sl.	Sl.	M.	Se.
ε_b or d, r	Ne.	V. Sl.	Sl.	M.	Se.
ε_t	Ne.	V. Sl.	Sl.	M.	Se.
Damaged Area ^a	Sl.	Sl.	Se.	Se.	Se.

^a The Tot. No. Cracks are assumed based on the recommendations of Table 2.1. As no criteria is yet developed for the Damaged Area metric, this followed suite with the Tot. No. Cracks, using a 2.5% values as a threshold.

^{Note} The abbreviations are given as following: V. = Very; Sl. = Slight; M. = Moderate; Se. = Severe; Ne. = Negligible; H. = High; D. Lim. = Design Limit; No. = Noticeable; Mo. = Monitoring; R. A. = Remedial Action; U. Lim. = Ultimate Limit.

Figure 6.26 illustrates the best correlating pairs of damage metrics. Colour coding is used to indicate damage levels, with green, yellow, orange, red, and pink corresponding to levels 1 through 5, as defined in Table 6.11. The strongest correlation is observed between t and s (Figure 6.26b), with an R^2 score of 0.991. This is closely followed by the correlation between $\varepsilon_{h,\max}$ and Maximum Crack Width

6.5. CORRELATION BETWEEN DAMAGE METRICS ACROSS THE ENTIRE SET OF SIMULATIONS

(Figure 6.26a), which has an R^2 value of 0.965. Both correlations demonstrate consistency across various data regions, including those with fewer instances of higher damage levels. Additionally, the correlation between Maximum Crack Width and $\varepsilon_{h,max}$ (Figure 6.26c), as well as β and Total Number of Cracks (Figure 6.26d), show a generally strong relationship. However, these correlations exhibit greater variability as damage levels increase, suggesting a reduced correlation at higher damage values.

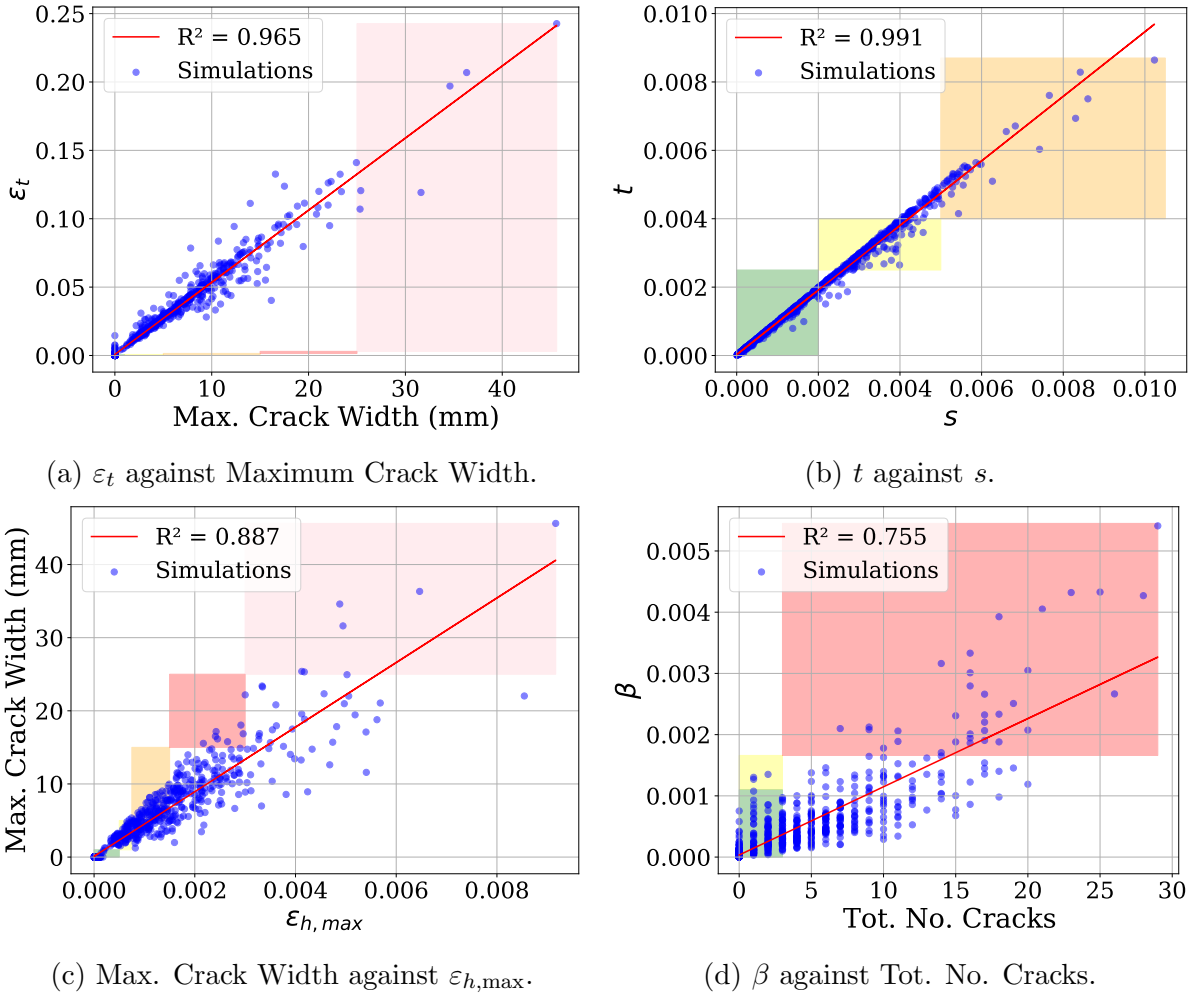


Figure 6.26: Closest 4 correlations found between pairs of metrics using the entire dataset.

Conversely, Figure 6.27 illustrates the best correlating pairs between other output variables. The results indicate that the variable most closely correlated with LTSM is s (Figure 6.27a), with an R^2 score of 0.479. Meanwhile, the Damaged Area shows a somewhat stronger correlation with the Total Number of Cracks (Figure 6.27b),

6.5. CORRELATION BETWEEN DAMAGE METRICS ACROSS THE ENTIRE SET OF SIMULATIONS

achieving an R^2 score of 0.722, though this correlation weakens as damage levels increase. On the other hand, despite the iterative procedure, both ε_t at 99% and ε_t at 95% (Figures 6.27c and 6.27d) show some degree of data clustering towards the lower end of values, with poor performance according to the R^2 score. This is reasonable, given that ε_t affects only a small percentage of the façade area, even amidst high levels of non-linearity. Ignoring the top 1% of strain values significantly reduces the potential of finding high damage values, and hence, reducing an additional 4% further increases data skewness, as depicted in Figures 6.15c and 6.15d.

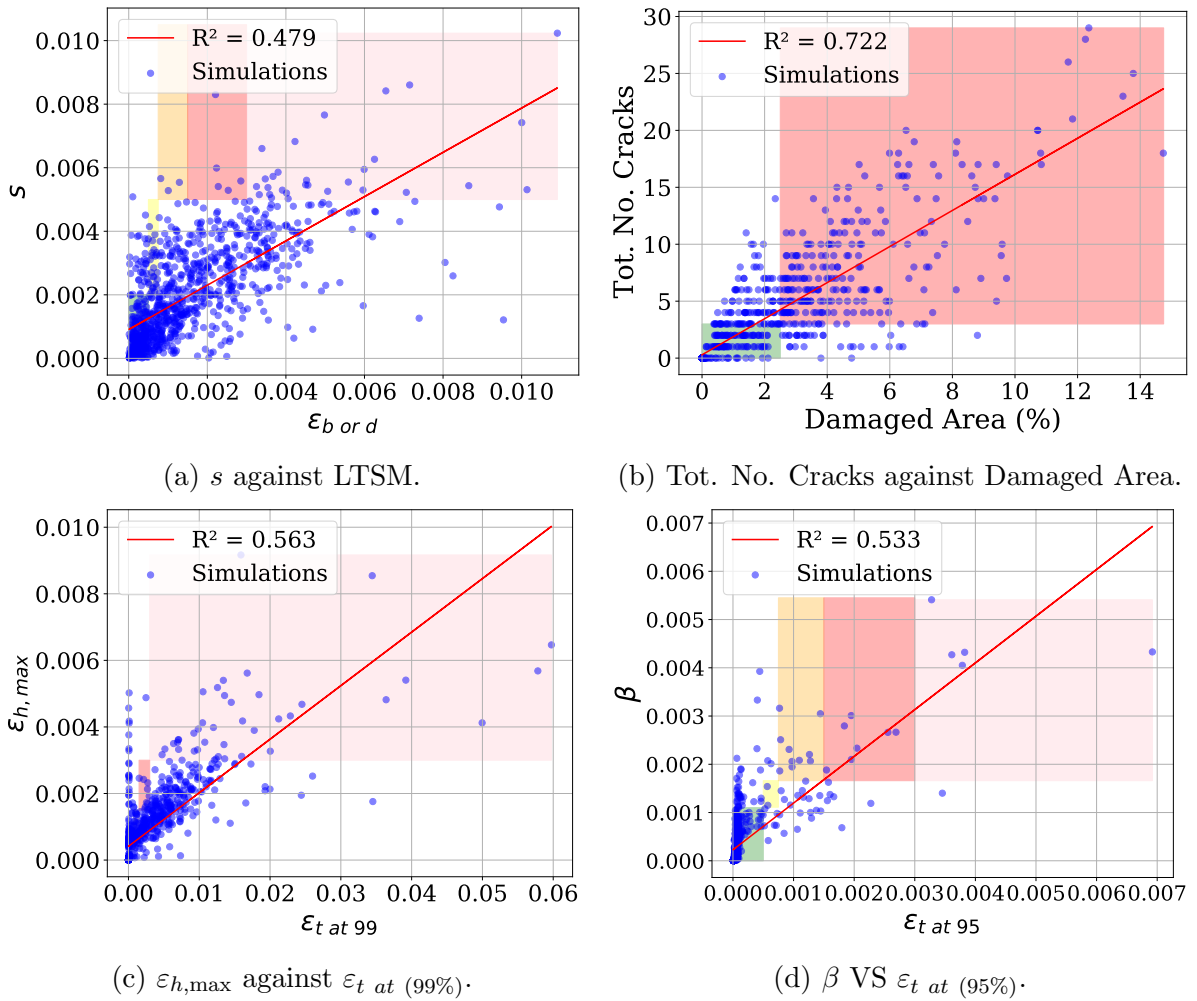


Figure 6.27: Furthest 4 correlations found between pairs of metrics using the entire dataset.

It is noted that in both Figures 6.26 and 6.27, a non-diagonal distribution of damage classifications is observed (as indicated by the colour boxes bounds). This occurs

because not all metrics are designed to assess damage in the non-linear domain; therefore, some values are forced to fall outside of the designated areas. For example, Figure 6.26a shows a very limited extension of the first four categories on the y-axis when considering traditional strain damages (hence, the other metric is used to assign the damage level). This is because damage, according to LTSM, yields very small values when considering a linear beam. Similar patterns can be observed in Figures 6.27a and 6.27c.

6.6 Summary

This chapter fulfils the fourth and fifth objectives of the thesis (Objective 4 and 5 in Section 1.5), by examining several aspects of local and global building damage metrics, investigating the total distributions of damage with simulations, proposing an iterative procedure, and identifying meaningful correlations between the damage metrics. During the investigation, the following observations were made:

- Building damage category is highly influenced by the specific metric used for its assessment, depending on the required output of the specific case study. For example, a building may indicate severe damage based on a global metric while simultaneously showing negligible damage using a local metric, and vice versa. Therefore, the inclusion of multiple metrics provides a holistic view of the various damage levels, enriching the understanding of building damage.
- The distributions of damage and all related metrics showed varying levels of data skewness or imbalance (with some being very severely skewed), suggesting that using input data produced by the initial sampling of a selected search space is not necessarily sufficient to produce high-quality datasets (i.e., evenly populated output classes). Therefore, rebalancing techniques such as over- and under-sampling, or the proposed iterative procedure (depending on the nature of the data), must be used in accordance with design requirements to

achieve this goal.

- The suggested iterative procedure effectively addressed this issue by producing better distributions of output variables, especially in the case of highly non-linear outputs such as Maximum Crack Width and $\varepsilon_{h,\max}$ (which were used as targets in this case). Nonetheless, this method necessitates further improvements. Severe limitations include the lack of optimisation in different framework components such as: the number of samples in the initial and iteration datasets, the total number of iterations, validation of the value of ϕ used for the SA threshold, and determining an optimal range of reduction (rather than using a fixed $1 \sigma_{X_i}$ for all parameters), depending on the rate of change in influence between $Y_{i \text{ Region 1}}$ and $Y_{i \text{ Region 3}}$.
- Correlation studies indicated that the strongest relationships occurred between s and t , Maximum Crack Width and ε_t , $\varepsilon_{h,\max}$ and Maximum Crack Width, and Total Number of Cracks and β , respectively. Conversely, the weakest correlations were found between ε_b or d , r and s , ε_t at 95% and β , ε_t at 99% and $\varepsilon_{h,\max}$, and Damaged Area and Total Number of Cracks, respectively.

A key step in producing a high-quality synthetic dataset is proposed in this chapter. These improved datasets are used in Chapter 7 for the training of ML algorithms, with the goal of determining whether improved model assessment metrics in ML can be achieved. This potentially reduces the need for advanced ML model manipulation or added complexity to generate better model predictions.

Chapter 7

Machine Learning Training and Evaluation

Using the datasets from the previous chapter, this chapter assesses the performance of several common ML algorithms. These algorithms are tested using various segments of the dataset to evaluate the impact of the iterative procedure, utilising the original dataset, the improved datasets (iterations 1 to 5), and a combination of both. The primary output variable used for training the models is the same as that used in the iterative framework, namely, the Maximum Crack Width as the local metric and $\varepsilon_{h,\max}$ for the global metric. Each output variable is trained using a separate ML model, with a single output neuron per model. Additionally, other global and local metrics are also used to train and assess the performance of the ML models.

In this investigation, the ANN, SVR, and RFR ML models, which are specific to regression tasks, are used and evaluated. These models are implemented using Python's `scikit-learn` library, which allows manipulation of model hyperparameters, selection of appropriate activation functions where applicable, and other adjustments depending on the model used. Each model is trained using the 15 input parameters outlined in Table 5.6 to predict the local and global metrics described

earlier. Training is performed on the raw data rather than using damage classifications based on damage categories. As a result, the output variables are treated as continuous, making this a regression task that requires appropriate evaluation metrics such as R^2 or rRMSE. Conversely, if damage categories were used, classification metrics such as accuracy and precision would be applied. The predictive performance of each model is then compared, and the best-performing model is selected for making predictions. These models were chosen due to their capability in addressing complex geotechnical engineering challenges, as demonstrated in studies such as (Cao et al. (2022); Mahmoodzadeh et al. (2022); Hou et al. (2022)). Additionally, the impact of applying several techniques commonly used in the literature to enhance the models' predictive capabilities is explored. These are described below:

- **Feature Selection:** Used to identify key parameters and exclude less relevant ones during training, potentially increasing model training performance (Tang and Na (2021));
- **Advanced Optimisation Techniques:** Such as BO to find optimal model hyperparameters instead of relying on fixed values, which may limit performance, as highlighted by (Shreyas and Dey (2019));
- **Data Transformation:** Including input-output standardisation (Equation 2.48), normalisation (Equation 2.47), and power transformation scaling methods as suggested by Hou et al. (2022), or a combination of both;
- **Cross-validation (k_{fold}):** Used to assess model generalisation to unseen data and evaluate prediction bias and overfitting risk (Mahmoodzadeh et al. (2022)).

Common bounds of hyperparameter domains indicated in the literature are used to define the search space for the BO algorithm. These bounds are listed in Table 7.1. To fine-tune the hyperparameters, BO is employed, which operates by constructing a probabilistic surrogate model of the objective function. The process begins with n_{init} randomly selected samples to initialise the surrogate function, which is then

explored efficiently over n_{iter} iterations using an acquisition function. This function identifies candidate samples expected to yield improvement, which are then evaluated on the actual objective function.

Table 7.1: Hyperparameter limits for each model.

Model	Hyperparameter	Minimum Range	Maximum Range
ANN	Number of layers (<code>n_layers</code>)	3	4
	Number of nodes (<code>n_nodes</code>)	32	128
	Log learning rate (<code>log_lr</code>)	-4	-1
RFR	Number of estimators (<code>n_estimators</code>)	50	1000
	Maximum depth (<code>max_depth</code>)	2	30
	Minimum sample split (<code>min_samples_split</code>)	2	10
SVR	Log C (<code>log_C</code>)	0	3
	Epsilon (<code>epsilon</code>)	0.01	1.0
	Kernel numeric (<code>kernel_numeric</code>) ¹	0	3

¹ Kernel numeric: 0 is linear, 1 is polynomial, 2 is Radial Basis Function (RBF), and 3 is sigmoid.

Once data is transformed according to its specific needs (e.g., using standardisation or normalisation), it is further divided into training and testing sets using an 80/20 split. Subsequently, a 5-fold cross-validation (k_{fold}) is applied to evaluate the generalisation ability of the model. This method divides the dataset into five equal parts. Each subset is used once as a validation set, while the remaining four subsets are used for training. This process is repeated five times so that each subset is used for validation exactly once. The cross-validation procedure is performed after the initial training stage, once the ML model structure has been determined.

7.1 Training Performance on Different Datasets

The training performance of each model across different datasets is summarised in Table 7.2. From the table, it is observed that the best-performing model overall, across various datasets and both local and global output metrics, is the ANN, with SVR and RFR yielding comparable but slightly inferior results. In particular, on the combined dataset (original plus iterations 1 to 5), the ANN achieved a low

relative root mean square error (rRMSE) of 4.51% and a high coefficient of determination (R^2) of 0.94, both indicating strong predictive performance. However, a closer examination reveals that significant improvements across datasets were not achieved for certain highly non-linear outputs, such as Maximum Crack Width, Total Number of Cracks, and $\varepsilon_{h,\max}$. Conversely, more linearly behaving output variables (e.g., s , t , and β) showed consistent improvement with each iteration of the dataset.

This behaviour can be attributed to the differences in output redistribution and their respective levels of non-linearity. While the iterative procedure improved the data distribution and reduced the dominance of underrepresented regions, this refinement was more effective for linearly responsive outputs. These benefited from the increased number and diversity of samples, enabling the model to better approximate the regression line. For non-linear outputs, the improvements were more modest, suggesting that the number of new samples was insufficient to enhance predictive accuracy significantly.

Although the distributions were significantly improved in both cases, helping to reduce potential biases in the original dataset predictions, this improvement also revealed a key limitation. The previously better training performance of the original dataset was largely due to the concentration of clustered outputs near the lower end of the value range, which were closer to the regression line. However, this clustering did not represent the full range of damage scenarios and thus did not reflect the model's true predictive ability. Nonetheless, since the prediction metrics were comparable across all three datasets, especially for the non-linear outputs, the second (iterated) and third (combined) datasets are considered to provide more reliable predictive performance. These datasets enable a more balanced assessment across the full output domain.

To better illustrate this phenomenon, the actual versus predicted outputs are presented in Figures 7.1 and 7.2. In these figures, clustering of data points is clearly

Table 7.2: Performance metrics for different models across various damage metrics.

Model	Metric	Original	Iterations	Combined	Hyperparameters ^b
ANN	Max. Crack Width	(6.26, 0.83) ^a	(12.24, 0.68)	(7.80, 0.75)	3/116/0.0165
	$\varepsilon_{h,\max}$	(8.98, 0.78)	(11.64, 0.68)	(9.00, 0.77)	3/85/0.0001
	Tot. No. Cracks	(5.67, 0.81)	(7.32, 0.86)	(8.74, 0.82)	3/99/0.0001
	s	(8.83, 0.87)	(6.39, 0.92)	(4.78, 0.93)	3/124/0.0115
	t	(8.57, 0.87)	(6.47, 0.92)	(5.19, 0.94)	3/86/0.0279
	β	(3.82, 0.91)	(6.18, 0.90)	(4.51, 0.93)	3/119/0.0367
RFR	Max. Crack Width	(11.17, 0.45)	(13.78, 0.59)	(9.57, 0.62)	565/17/2
	$\varepsilon_{h,\max}$	(13.04, 0.54)	(13.31, 0.58)	(8.90, 0.78)	50/13/7
	Tot. No. Cracks	(7.74, 0.65)	(10.17, 0.73)	(11.07, 0.72)	423/22/2
	s	(12.03, 0.76)	(9.88, 0.81)	(7.07, 0.85)	641/29/4
	t	(11.48, 0.76)	(9.82, 0.82)	(8.27, 0.84)	509/24/2
	β	(6.99, 0.70)	(11.48, 0.64)	(8.14, 0.77)	137/11/2
SVR	Max. Crack Width	(9.01, 0.64) ^c	(12.66, 0.66)	(8.90, 0.67)	1.58/0.281/RBF
	$\varepsilon_{h,\max}$	(11.98, 0.61)	(11.60, 0.68)	(9.18, 0.76)	2.3/0.106/RBF
	Tot. No. Cracks	(8.00, 0.62)	(8.24, 0.82) ^c	(9.53, 0.79)	3.61/0.37/RBF
	s	(11.57, 0.77)	(7.80, 0.88)	(6.46, 0.88)	9.46/0.024/RBF
	t	(11.00, 0.78)	(8.11, 0.87)	(7.49, 0.87) ^c	5.79/0.079/RBF
	β	(7.56, 0.65)	(7.44, 0.85) ^c	(6.62, 0.85)	839/0.118/RBF

^a These values represent the rRMSE in % and R^2 , respectively.

^b The model structure (hyperparameters) is representative of the model trained on the combined dataset, it represents the number of layers, nodes per layer, and Learning Rate (LR), for the ANN model, respectively.

^c Total number of BO iterations had to be reduced from 110 to 85 due to convergence difficulties.

visible in the original dataset (Figures 7.1a and 7.2a), which contributes to prediction bias (Ψ), resulting in deceptively low values of rRMSE and high values of R^2 . By increasing the spread of data points and reducing the dominance of clustered regions, the model becomes capable of making predictions in less populated areas of the output space. However, due to the high variability typically associated with non-linear outputs, this increase in predictive scope may come at the cost of slightly higher error values in rRMSE and R^2 , as seen in Figures 7.1d and 7.2d.

Nevertheless, such models are considered more reliable, as they are capable of making informed predictions in output regions that were previously underrepresented or unseen in the original dataset, an important consideration for generalising across a broader damage domain. Finally, the combined dataset (original + iterations) demonstrates the model's ability to adapt to the expanded distributions (Figures 7.1c and 7.2c). Notably, the model trained on this combined dataset performs comparably to the original in terms of Ψ -biased accuracy metrics (as shown

in Table 7.2), while also achieving a far more balanced and comprehensive spread across the entire output domain. This results in more robust and generalisable predictions of building damage across a wider range of cases.

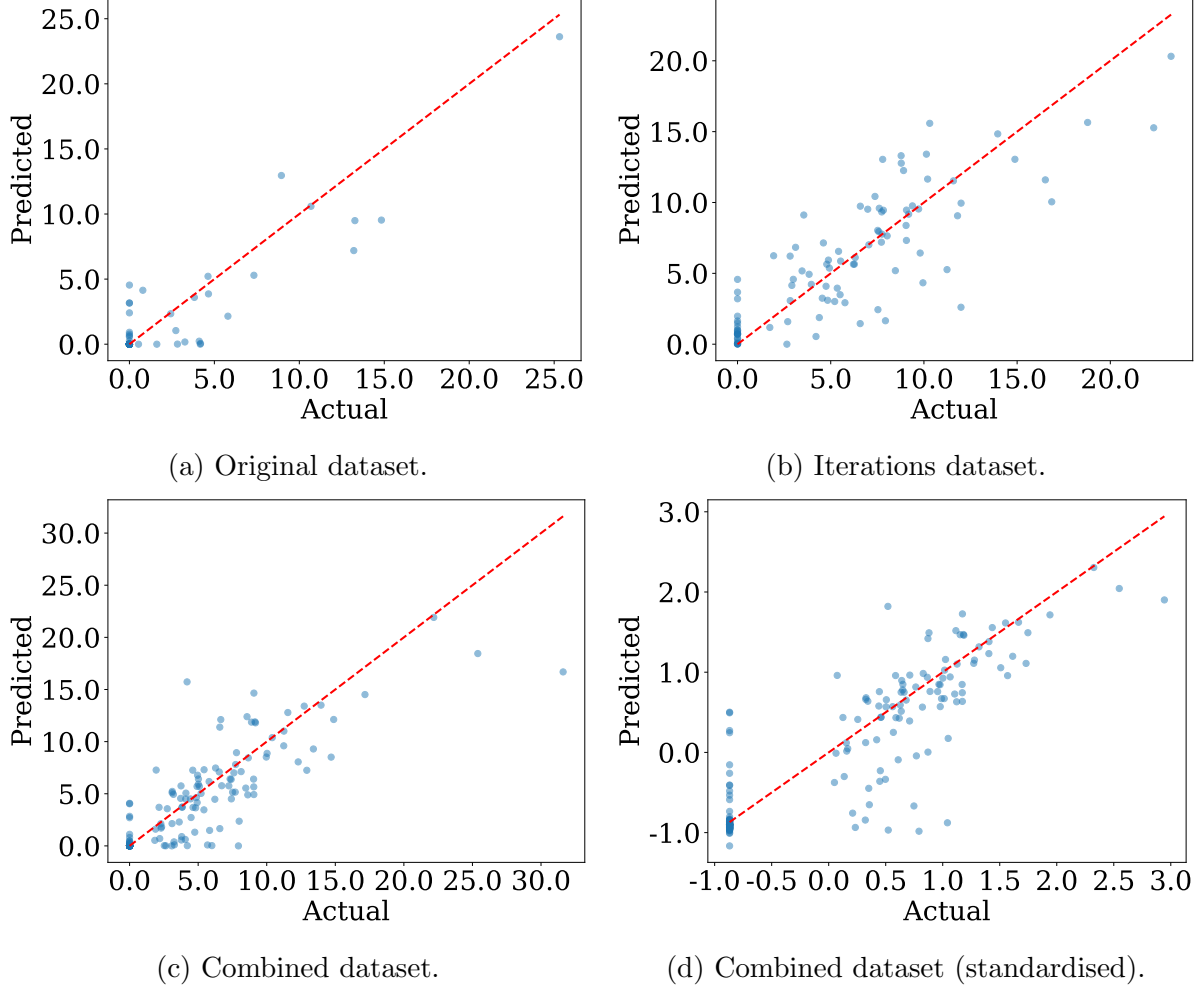


Figure 7.1: Actual vs predicted values using the Max. Crack Width metric. Illustrating standardised values too (sometimes helping in visualising data better).

Model training is terminated once the loss function converges to a predefined low value for the selected criterion, in this case, MSE for the ANN, and shows no improvement over several consecutive iterations. Figure 7.3 illustrates the evolution of the loss function during the training process of the ANN model for the global metric (Figure 7.3a) and the local metric (Figure 7.3b). The model is programmed to identify convergence when the MSE drops below a tolerance threshold of 0.0001 and remains consistently below this value for 10 consecutive iterations. From Fig-

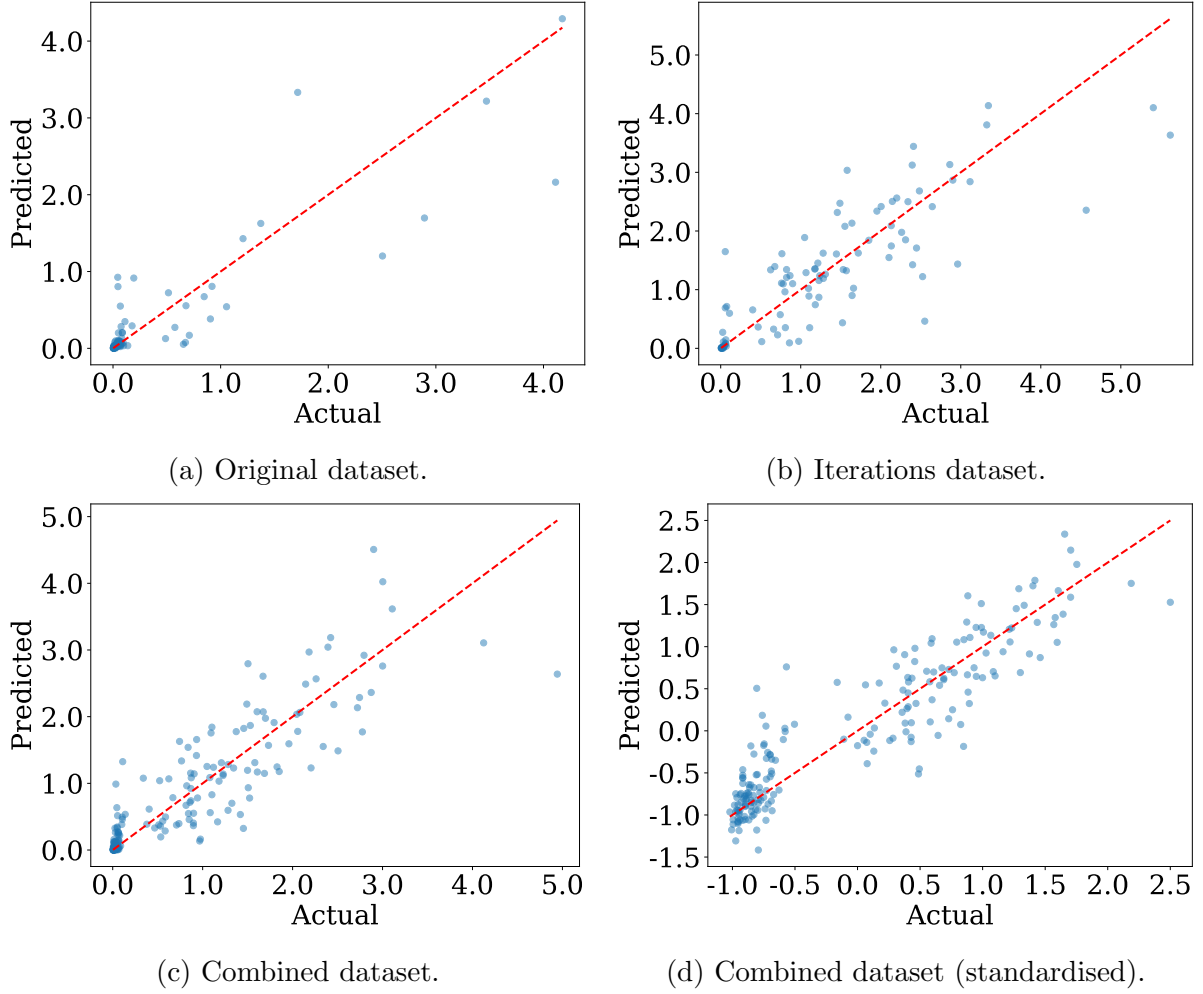


Figure 7.2: Actual vs predicted values using the $\varepsilon_{h,\max} \times 10^{-3}$ metric.

Figure 7.3a, a relatively smooth, parabolic decline in MSE values is observed for the global metric, with convergence achieved at approximately 420 iterations out of a total of 2500 maximum iterations defined. In contrast, the model trained on the local metric exhibits a much steeper reduction in MSE, reaching very low values within the first 30–40 iterations and achieving convergence at approximately 68 iterations (Figure 7.3b).

Both models demonstrate stable and efficient convergence, which indicates that further improvements in predictive accuracy, measured by rRMSE and R^2 , could be achieved by refining the model’s hyperparameter space (e.g., expanding the search bounds or increasing the number of BO iterations), or by enhancing the size and diversity of the training dataset, particularly for outputs associated with high

non-linearity.

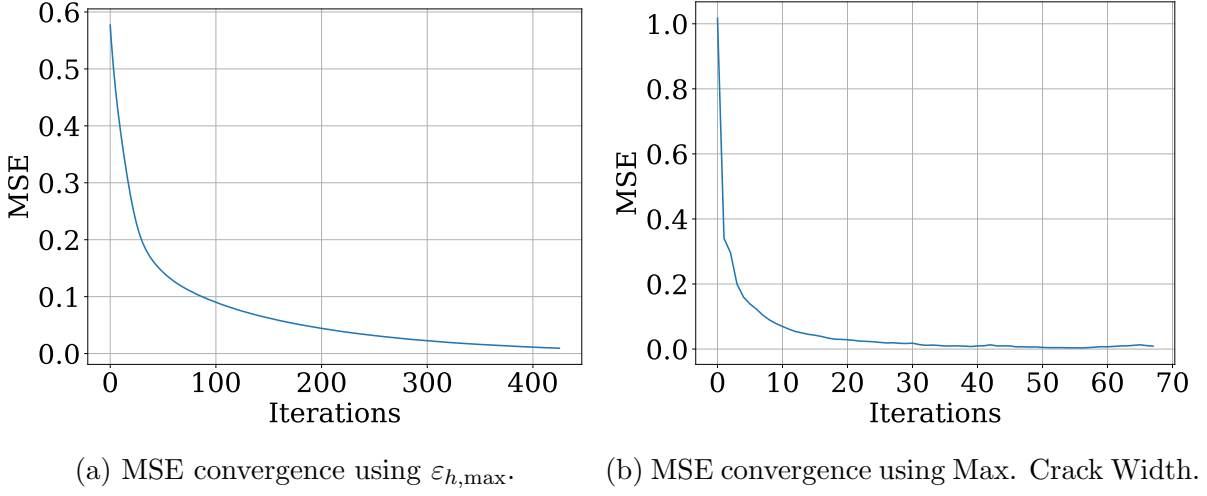


Figure 7.3: Loss function (MSE) convergence for ANN with increased iterations.

Having identified the best-performing model, namely, the ANN, and confirmed its convergence to a stable solution based on the MSE criterion, the model is subsequently subjected to a rigorous cross-validation procedure to evaluate its generalisation capability. The results of this evaluation, using a 5-fold cross-validation ($k_{fold} = 5$), are presented below. Each fold uses an 80/20 data split, ensuring each subset of the data is used once for validation while the remaining serve as training sets. Table 7.3 presents the rRMSE and R^2 values obtained for each fold using the Maximum Crack Width as the output variable. The average performance across all five folds yields an rRMSE of 10.15% with a standard deviation $\sigma_{rRMSE} = 1.83$, and an R^2 value of 0.65 with $\sigma_{R^2} = 0.043$.

Similarly, Table 7.4 shows the results for the global metric $\varepsilon_{h,\max}$. The average values across folds are 9.76% for rRMSE with $\sigma_{rRMSE} = 1.62$, and 0.71 for R^2 with $\sigma_{R^2} = 0.043$. Each table is accompanied by plots illustrating actual versus predicted values for each fold, highlighting the spread and alignment of predictions relative to the ideal diagonal fit line. While minor fluctuations are observed across folds, the relatively low values of σ_{rRMSE} and σ_{R^2} indicate that the model structure established during combined dataset training does not overfit the training data. These results confirm the model's robust generalisation performance and reliability.

when applied to unseen data.

Table 7.3: Cross-validation using the Max. Crack Width as output variable and a total of 5 folds with a 20-80 split.

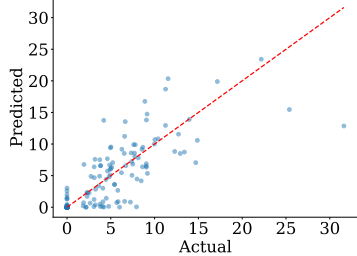
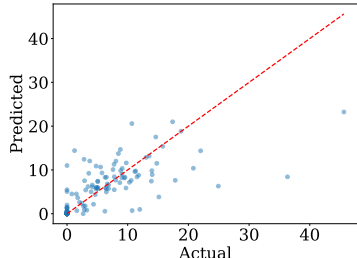
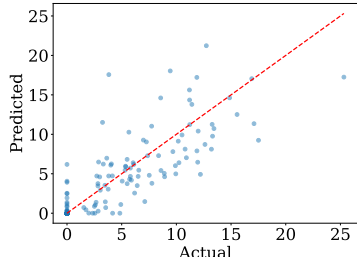
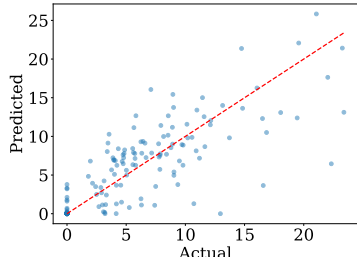
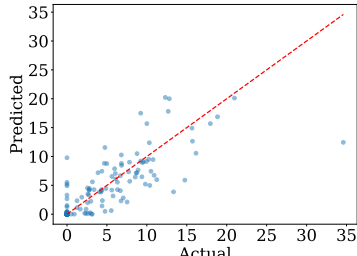
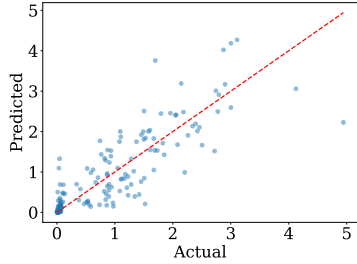
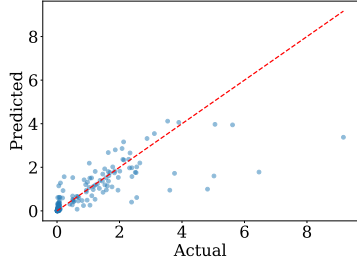
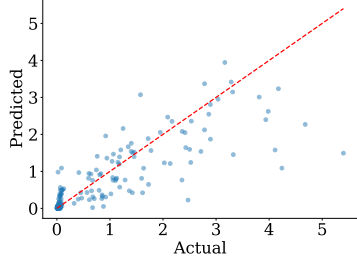
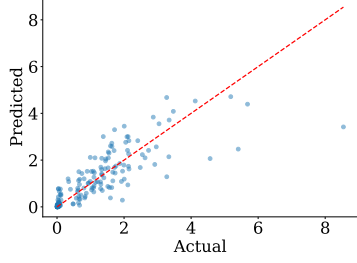
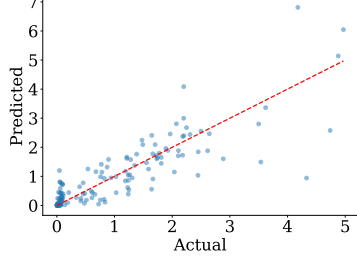
Fold Number	(rRMSE (%), R^2)	Actual vs Predicted
1/5	9.23, 0.65	
2/5	9.18, 0.58	
3/5	10.39, 0.70	
4/5	13.58, 0.67	
5/5	8.36, 0.67	

Table 7.4: Cross-validation using $\varepsilon_{h,\max}$ as output variable.

Fold Number	(rRMSE (%), R^2)	Actual vs Predicted
1/5	9.11, 0.77	
2/5	8.76, 0.63	
3/5	11.75, 0.69	
4/5	7.63, 0.73	
5/5	11.56, 0.71	

7.2 Impact of Feature Engineering and Split Ratio on Performance

In the case of skewed output variable distributions, as encountered in this study, data transformation is a common and effective strategy that can improve the performance of ML models (Hou et al. (2022)). In this investigation, the models are trained using transformed datasets; however, the outputs are returned to their original scale when making predictions and evaluating performance. This section assesses the performance of the ANN model using three different transformation techniques applied to the output variable only: the square root, logarithmic, and inverse transformations. The original (untransformed) output distribution is also included for comparison. To isolate the effects of the transformation techniques, the optimised hyperparameters used in Table 7.2 are fixed and applied accordingly for each respective output variable. The focus is placed on the local output metric, Maximum Crack Width, and the global output metric, $\varepsilon_{h,\max}$, using the entire combined dataset for training and evaluation.

In addition to transformation, all training is performed on the standardised dataset. The sequence of data preprocessing follows these steps: transformation of the output variable, followed by standardisation of the full dataset (inputs and transformed outputs) before training. During the prediction and evaluation stages, the inverse process, destandardisation and then detransformation, is applied to obtain results in the original physical scale. The resulting distributions of the output variables after each transformation are presented in Figures 7.4 and 7.5. Each figure illustrates the frequency distribution using 20 bins. Only the original distribution plots (Figures 7.4a and 7.5a) display the true physical values of the output variable along the x-axis, with appropriate units included. The remaining transformed distributions (Figures 7.4b to 7.4d and 7.5b to 7.5d) illustrate the spread of values in transformed space. These are not physically interpretable in terms of building damage but are

7.2. IMPACT OF FEATURE ENGINEERING AND SPLIT RATIO ON PERFORMANCE

useful for identifying whether clustering in the output data has been mitigated, thereby improving the model's training potential.

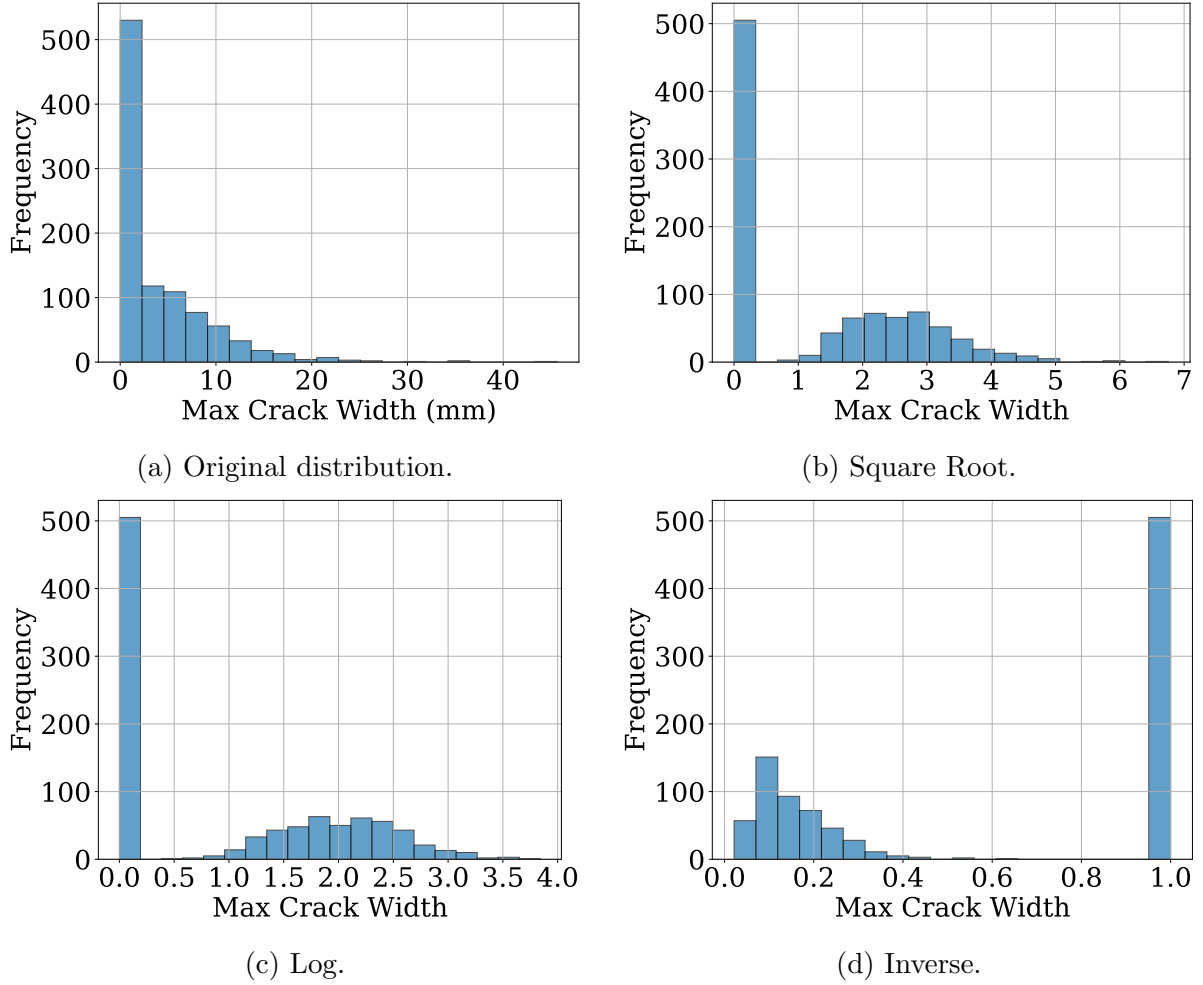


Figure 7.4: Transformed local output (Max. Crack Width) variable using different transformation techniques.

Training performance for both output variables, evaluated in terms of rRMSE and R^2 , across the various transformation techniques is presented in Table 7.5. For the local output metric, the performance differences among the transformation techniques are relatively minor. Both the square root and logarithmic transformations yield slightly better predictive accuracy compared to the original dataset. In contrast, the inverse transformation produced unrealistic results, with abnormal values for both rRMSE and R^2 ; this outcome was not further investigated. For the global output metric, performance improved more notably with the application of the square root transformation, followed closely by the logarithmic and

7.2. IMPACT OF FEATURE ENGINEERING AND SPLIT RATIO ON PERFORMANCE

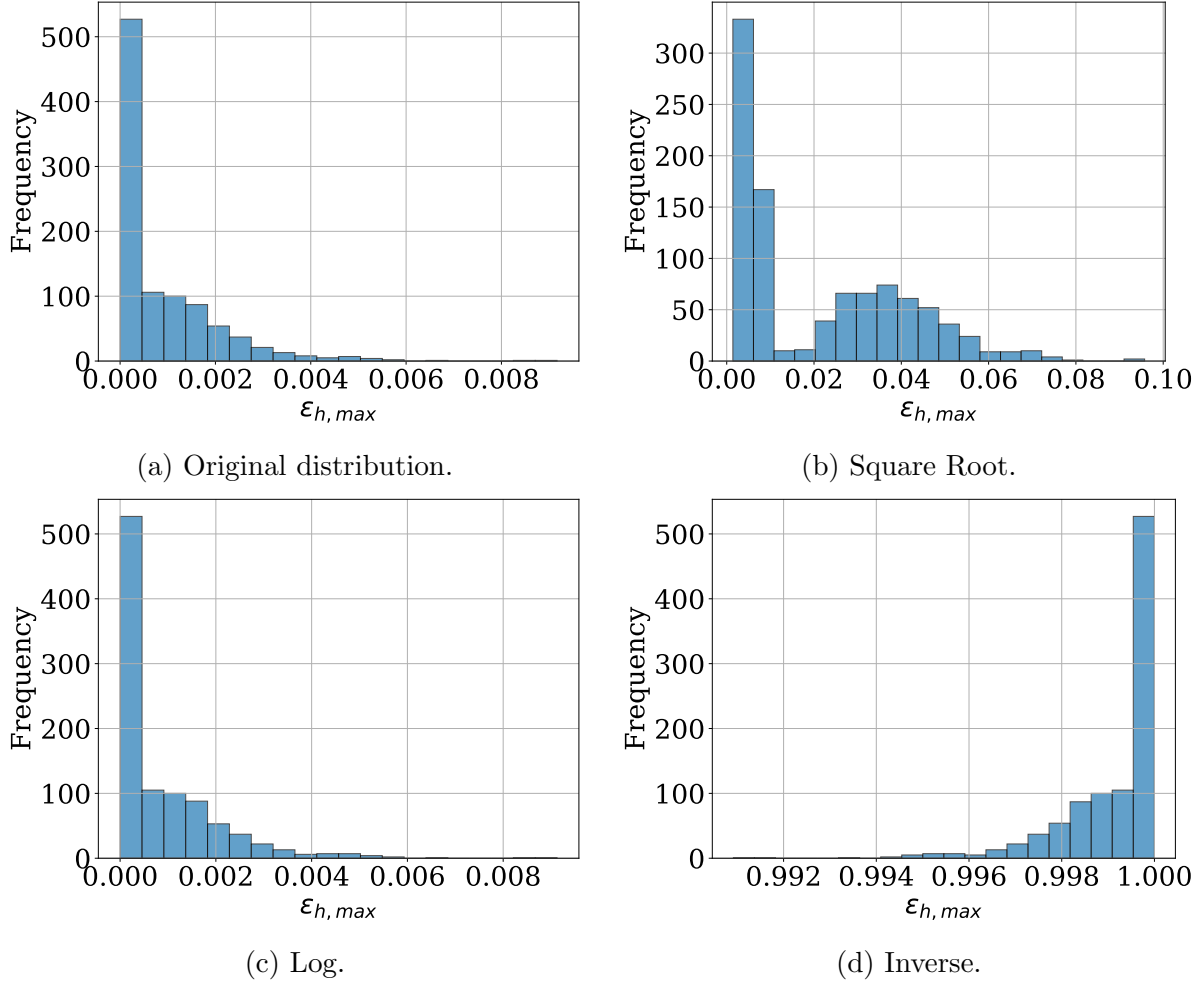


Figure 7.5: Transformed global output ($\varepsilon_{h,max}$) variable using different transformation techniques.

original (untransformed) datasets, which yielded very similar results. The inverse transformation again resulted in the poorest performance.

From these findings, it is concluded that appropriate data transformation can improve model prediction accuracy, even when preceded by a rebalancing procedure such as the iterative approach. Transformations help to further alleviate skewness in the output distributions, making the training process more effective. In this investigation, only a limited selection of transformation techniques was applied. However, many other approaches exist and could potentially enhance model training depending on the dataset characteristics, model structure, and domain expertise.

The influence of feature selection is further investigated. In the previous models,

7.2. IMPACT OF FEATURE ENGINEERING AND SPLIT RATIO ON PERFORMANCE

Table 7.5: ANN performance with different transformation techniques on local and global outputs.

Transformation	Local		Global	
	rRMSE	R ²	rRMSE	R ²
Original	8.74	0.68	10.4	0.7
Square Root	8.57	0.69	9.11	0.77
Log	8.58	0.7	10.39	0.7
Inverse	NA	NA	11.7	0.61

all 15 input parameters were used for training, regardless of their individual impact on the output. However, excluding irrelevant parameters during model training can enhance predictive performance (Tang and Na (2021)). To explore this, model performance is assessed using the standardised square-root-transformed output, applying five different thresholds for ϕ : i) no threshold, ii) 1%, iii) 3%, iv) 5%, and v) 10%, based on the most influential parameters identified using the RFR model previously applied in the SA. The outcomes of this investigation are presented in Table 7.6.

Although the literature generally supports the use of feature selection techniques to improve model performance, this was not observed in the present case. In fact, reducing the number of input parameters resulted in a decline in predictive accuracy. This may be attributed to complex, higher-order interactions between parameters, which the model may depend on for improved learning and performance. Consequently, for this specific case, retaining all input parameters, i.e., applying no threshold for ϕ , proved to be the most effective approach. Nonetheless, it is recommended that this process be applied routinely across different datasets and output variables, as varying levels of parameter interaction may produce differing effects on model performance.

Finally, three data-splitting ratios, as described in Section 2.6, are investigated: 10/90, 20/80, and 30/70 splits. The goal was to determine which split provides the best balance between model training quality and prediction performance. This investigation was carried out using both the local (Maximum Crack Width) and global

7.2. IMPACT OF FEATURE ENGINEERING AND SPLIT RATIO ON PERFORMANCE

Table 7.6: Performance metrics for different thresholds across global and local criteria.

Importance	global		local	
	rRMSE	R ²	rRMSE	R ²
No threshold (15)	9.11	0.77	8.6	0.69
1% (15)	9.11	0.77	8.6	0.69
3% (9/7) ^a	11.22	0.65	11.78	0.42
5% (4)	10.96	0.66	11.84	0.42
10% (3)	11.56	0.62	11.53	0.45

^a The values between the brackets indicate the number of influential parameters above the threshold, using the / indicate different number of influential parameters for different outputs (9 for local and 7 for global in this case) using the same threshold.

($\varepsilon_{h,\max}$) output variables, with the corresponding results presented in Table 7.7.

It is observed that model accuracy, as measured by rRMSE, improved for both output variables as the training portion of the split increased. However, at the 30/70 split, a decline in R² scores was noted, indicating potential overfitting, where the model learns the training data too closely and performs poorly on unseen data. Therefore, it is concluded that a 20/80 split achieves an optimal balance between training performance and generalisation capability. This trade-off is commonly encountered in ML, as larger training sets can improve learning but may simultaneously reduce the model's ability to generalise, depending on the algorithm and dataset characteristics.

Table 7.7: Performance metrics for different data splits across global and local outputs.

Split ratio	global		local	
	rRMSE	R ²	rRMSE	R ²
10/90 split	9.94	0.73	9.49	0.62
20-80 split	9.11	0.77	8.6	0.69
30-70 split	7.44	0.64	7.73	0.63

7.3 Artificial Neural Network with Particle Swarm Optimisation

The performance of ML training is further assessed by implementing an ANN optimised using a PSO meta-model, applied to the combined dataset. This model is briefly described in Section 2.6. An 80/10/10 split is used for training, testing, and validation, respectively. The dataset is normalised using Equation 2.47, and model accuracy is evaluated using the rRMSE metric (expressed in %).

The training and optimisation process is illustrated in Figure 7.6. The figure outlines the iterative training approach used to optimise the model's architecture and learning rate. Following the dataset split, the PSO is initialised with defined numbers of swarms and particles, including their initial positions and velocities. For each particle, an ANN is trained iteratively using the training subset, and the fitness function, based on the testing subset, is computed. Upon reaching the maximum training iterations, the prediction error, i.e., the rRMSE, is evaluated using the validation subset. This validation error is then used to update particle positions, thereby optimising the network weights (w) and biases (Ψ).

The configuration used in this study includes 30 PSO particles and 15 optimisation iterations for velocity and position updates. The ANN is trained with a learning rate (LR) of 0.001 and up to 3000 maximum iterations. Figure 7.7a presents the actual vs. predicted outputs for the global metric $\varepsilon_{h,\max}$, while the convergence of the PSO for this metric is shown in Figure 7.7b, indicating convergence after just two iterations and reaching a plateau of performance thereafter. In contrast, Figure 7.7c illustrates the actual vs. predicted performance for the local metric (Maximum Crack Width). The corresponding convergence curve in Figure 7.7d indicates that the model reached a stable solution after seven iterations, slightly more than the global metric. Finally, the predictive performance in terms of rRMSE for both output metrics is summarised in Table 7.8.

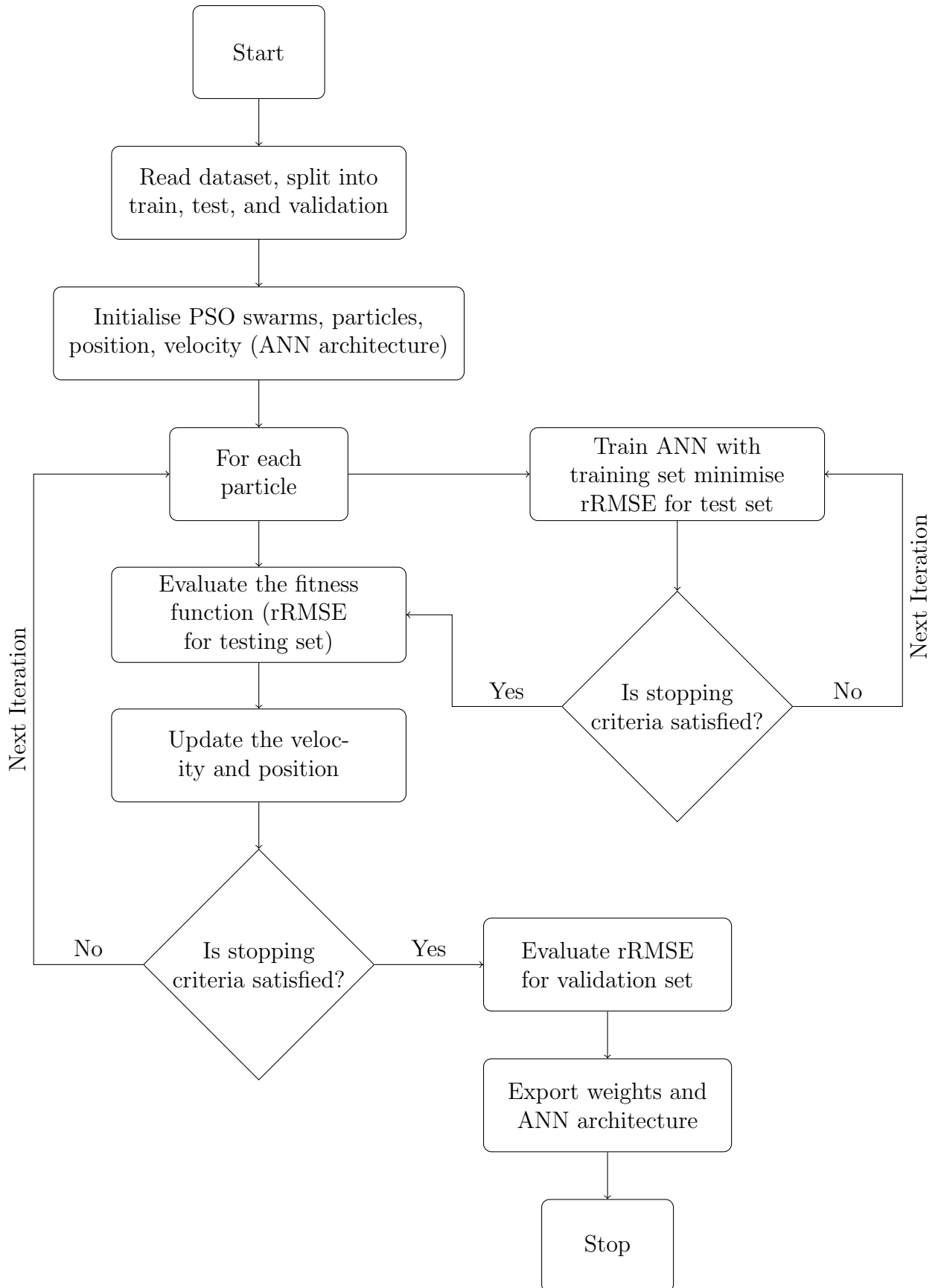
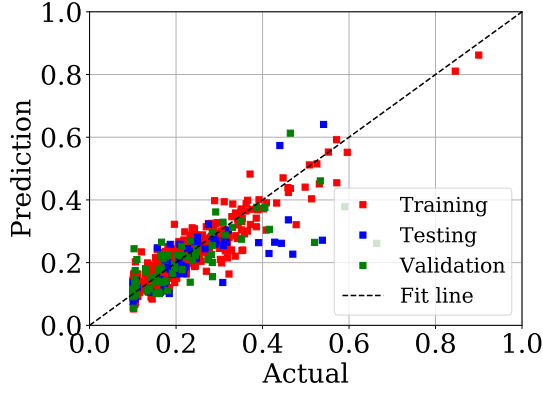
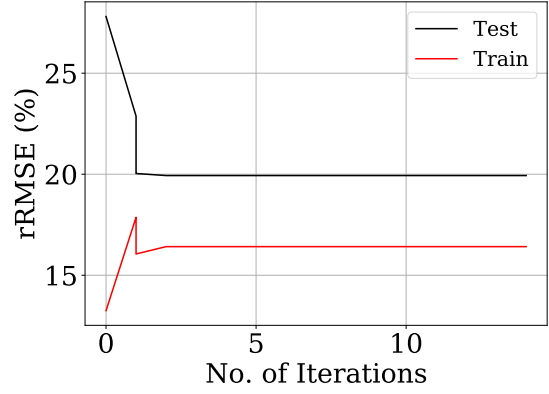


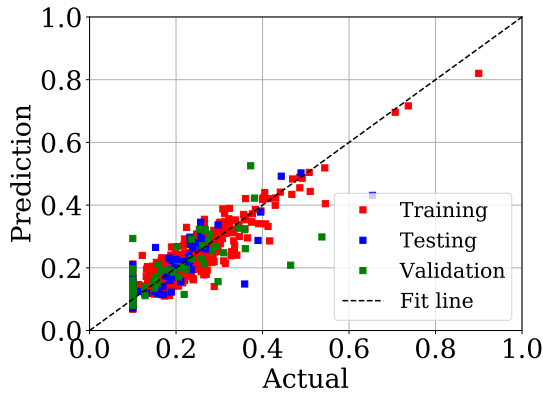
Figure 7.6: Schema for training of ANN optimised by PSO.



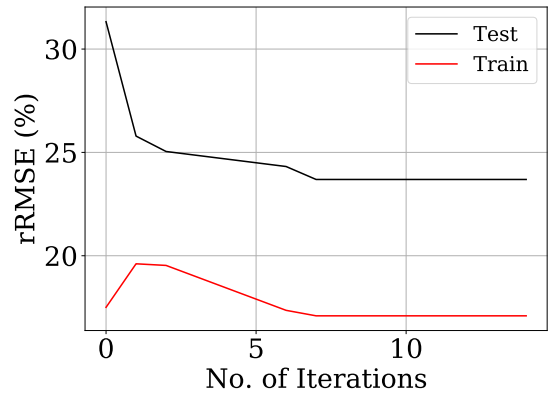
(a) Actual vs predicted values global output.



(b) PSO convergence.



(c) Actual vs predicted values local output.



(d) PSO convergence.

Figure 7.7: Actual vs predicted values plots in addition to the convergence performance of the PSO.

Table 7.8: Results of ANN-PSO performance on local and global outputs.

rRMSE (%)	$\varepsilon_{h,\max}$	Max. Crack width
Train	16.42	17.09
Test	19.93	23.69
Validation	27.3	30.65

It is quite noticeable that the performance metrics achieved by the ANN-PSO models are significantly less accurate compared to the results obtained from the models presented in Section 7.1. When comparing performance on the test set, it is observed that the ANN optimised via BO achieved approximately 4.5 times better predictive accuracy for the local output and around 2.3 times better performance for the global output compared to the ANN optimised with PSO. A more direct and fair comparison would involve using PSO to optimise the model's hyperparameters,

similar to the approach used with BO, rather than only optimising weights (w) and biases (Ψ) while keeping the hyperparameters fixed. However, this aspect was not further investigated.

7.4 Principal Component Analysis NU-Support Vector Regression Approach on Dataset

In this section, a novel framework recently proposed by Izonin et al. (2024) is implemented, which integrates unsupervised and supervised learning via the combination of Principal Component Analysis (PCA) and Nu Support Vector Regression (NuSVR), forming a unified PCA-NuSVR model. Their approach includes algorithms for training and application, employing 5-fold cross-validation, and evaluating results using standard performance metrics. Individually, PCA is a statistical method used to transform original data into a new coordinate system by replacing the original features with uncorrelated variables called principal components (Shakhovska et al. (2020)). A detailed mathematical overview of this method, including its advantages and limitations, is available in (Jolliffe and Cadima (2016)). In the context of this framework, PCA is applied to both the input and output variables simultaneously, creating a single hyperbody, a multidimensional structure representing the dataset. This allows the prediction of all six output attributes simultaneously (presented below in Table 7.9) and also permits inverse predictions of input features, if required, based on the hyperbody structure.

To perform these predictions, the NuSVR algorithm is used. NuSVR, a variation of the standard SVM, is tailored for regression tasks and constructs a regression function by controlling the fraction of data points used as support vectors through the Nu parameter. This method typically requires fewer optimisation parameters compared to classical SVR with an RBF kernel, thus potentially lowering computational complexity. Its main advantages include high modelling efficiency, reduced

sensitivity to overfitting, and the ability to generalise well on smaller datasets. However, it is known to be sensitive to outliers and still requires careful parameter tuning.

By combining these two techniques, the PCA-NuSVR framework effectively addresses the regression problem defined in this thesis. The full structure of the framework is shown in Figure 7.8, adapted from (Izonin et al. (2024)). To simplify the presentation, the term “ML System” is introduced to refer to the collection of eight independent NuSVR models, each responsible for predicting one of the output damage metrics.

The framework is divided into three main components: the preparation block, the training block, and the application block. For convenience, the preparation and training blocks are combined into a single mode referred to as the “training mode”, while the third operates as the “application mode”, used for making predictions on unseen data.

Table 7.9: Values of the performance metrics for predicted local-global damage indicators via the proposed PCA-NuSVR framework for the assessment of building damage due to tunnelling.

Error Metric	Tot. No. Cracks	t	s	$\varepsilon_{h,\max}$	Max. Crack Width	β
MaxError	8.06	0.0024	0.0024	0.0026	9.51	0.0010
MedError	0.52	0.0002	0.0002	0.0002	0.97	4.9E-08
MAE	1.03	0.0003	0.0003	0.0003	1.59	0.0001
MSE	2.72	1.96E-07	2.04E-07	2.26E-07	5.45	2.8E-08
RMSE	1.65	0.0004	0.0004	0.0005	2.14	0.0002
rRMSE (%)	3	1.5	1.5	3.5	3.7	9.2
R^2	0.776	0.893	0.897	0.707	0.7	0.8
Optimal Nu value	0.26	0.89	0.99	0.70	0.2	0.8
Optimal C value	0.297	0.94	2.38	0.841	0.187	0.53

The results of the proposed PCA-NuSVR framework for both global and local output metrics, based on the combined dataset, are summarised in Table 7.9. This table presents a comprehensive evaluation using several performance metrics, including the average values derived from a 5-fold cross-validation. In particular, the rRMSE and R^2 values for both the Maximum Crack Width and $\varepsilon_{h,\max}$ are used to

7.4. PRINCIPAL COMPONENT ANALYSIS NU-SUPPORT VECTOR REGRESSION APPROACH ON DATASET

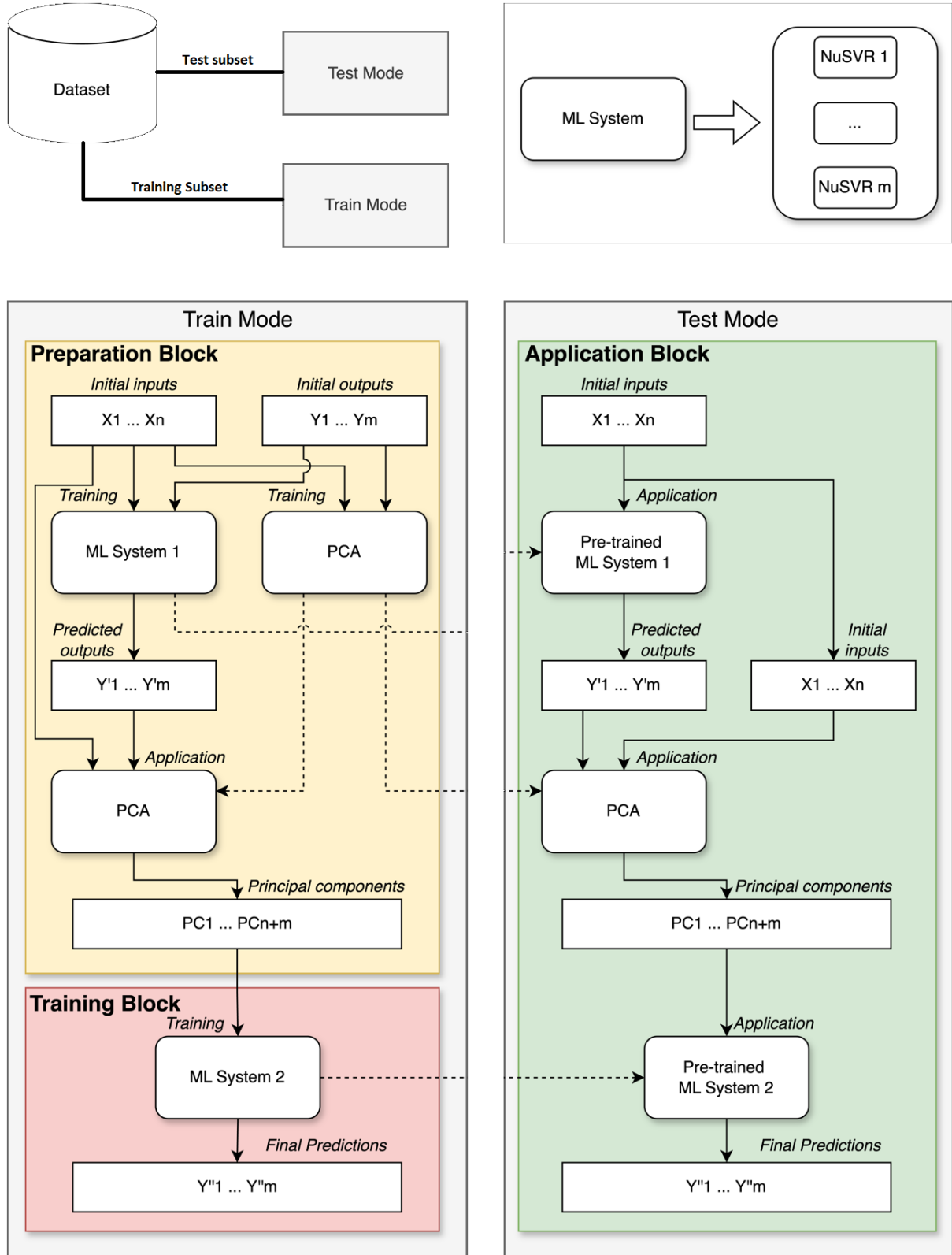


Figure 7.8: Flowchart of the proposed PCA-NuSVR framework (Izonin et al. (2024)).

assess the predictive accuracy of the model. From the results, a marked improvement in the rRMSE values is observed compared to previous models, including

the ANN results reported in Table 7.2. While the improvements in R^2 scores are less pronounced, the PCA-NuSVR approach demonstrates comparable or superior predictive reliability.

Additionally, the actual versus predicted values for the testing sets are illustrated in Figure 7.9. These include the Maximum Crack Width (Figure 7.9a) and $\varepsilon_{h,\max}$ (Figure 7.9b). Similar to prior methods, a noticeable clustering of data is observed near the lower end of the value spectrum, particularly in the Maximum Crack Width results, due to the cracking activation criteria. However, this effect is less pronounced in the $\varepsilon_{h,\max}$ distribution. Nevertheless, thanks to the iterative data rebalancing procedure introduced earlier, the model benefits from a more representative and diversified dataset. As a result, the PCA-NuSVR model is capable of reliably predicting across a wider range of output values, including those in less common or sparsely populated regions of the output domain.

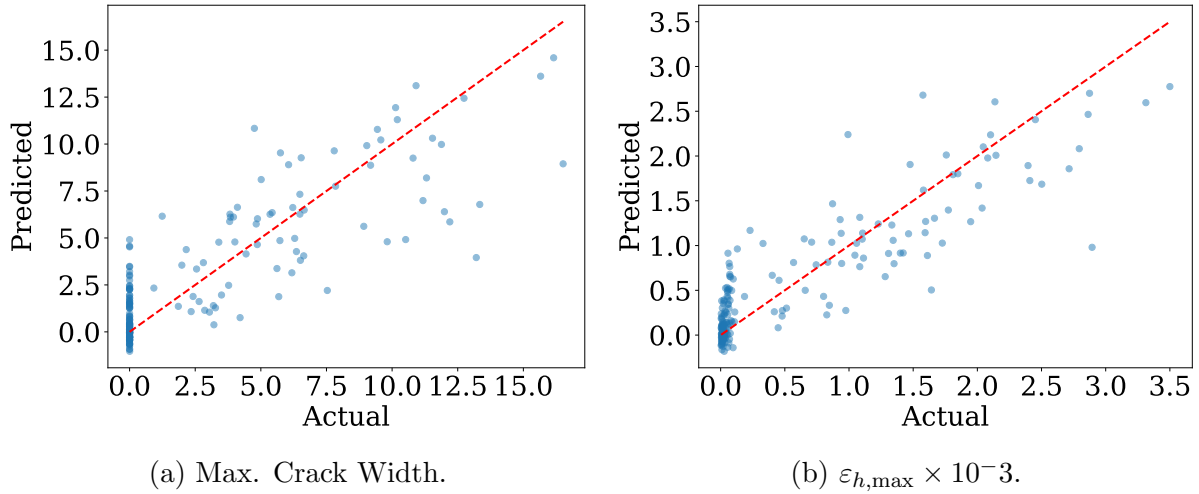


Figure 7.9: Actual vs predicted values using the PCA-NuSVR model on testing sets for the (a) local output (b) global output.

Additionally, to provide a more comprehensive comparison of the accuracy of the proposed PCA-NuSVR framework, a selection of widely used ML methods were also evaluated. These include: the baseline NuSVR model, which serves as the foundation of the proposed framework; the classical SVR with a radial basis function (RBF) kernel, which is methodologically similar; and a range of ensemble-based

7.4. PRINCIPAL COMPONENT ANALYSIS NU-SUPPORT VECTOR REGRESSION APPROACH ON DATASET

regression models including the RF, Gradient Boosting (GB), eXtreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machine (LGBM) regressors. Table 7.10 summarises the comparative results for all investigated methods based on their R^2 scores, which are used to assess the accuracy of predictions for both local and global output indicators of building damage. This comparative analysis further supports the robustness and predictive strength of the proposed framework relative to conventional regression techniques.

Table 7.10: R^2 -values of the local and global indicators for the assessment of building damage using different ML-based regressors.

Damage Metric	Proposed Framework	NuSVR	SVR-RBF	LGBM	XGBoost	RFR	GB
Crack Width	0.697	0.664	0.639	0.619	0.572	0.575	0.546
Tot. Cracks	0.776	0.730	0.700	0.699	0.648	0.639	0.630
$\varepsilon_{h,\max}$	0.707	0.668	0.643	0.635	0.598	0.579	0.556
β	0.799	0.764	0.739	0.691	0.705	0.682	0.670
s	0.897	0.889	0.875	0.830	0.853	0.831	0.864
t	0.893	0.883	0.868	0.822	0.849	0.823	0.852

As evident from Table 7.10, the ensemble methods such as RF, GB, and XGBoost demonstrated relatively unsatisfactory prediction accuracy for the local indicators. Slightly improved results were observed with the classical SVR using an RBF kernel and the LGBM regressor. Notably, significantly higher accuracy was achieved by the basic NuSVR, which forms the foundation of the proposed framework. However, the proposed PCA-NuSVR framework yielded the lowest prediction errors for the local indicators associated with tunnelling-induced building damage. It achieved a 5% improvement in R^2 values for the Maximum Crack Width over the next best-performing model, NuSVR. Similar performance trends were observed across the global indicators. In particular, while NuSVR consistently showed strong results, the proposed PCA-NuSVR model delivered a further 5.8% improvement in the R^2 value for $\varepsilon_{h,\max}$, again outperforming all tested methods.

7.5 Summary

This chapter addresses the sixth objective of the thesis (Objective 6 in Section 1.5) by exploring the application of various ML models for regression-based prediction of tunnelling-induced building damage, assessed using both local and global output metrics. Throughout the chapter, the influence of multiple techniques to improve model performance is evaluated, including data transformation, feature selection, hyperparameter optimisation, and model-specific training strategies. The following key findings are highlighted:

- The ANN-BO model consistently outperformed both the RFR-BO and SVR-BO models across all local and global damage metrics, as optimised using the bounds provided in Table 7.1.
- The effect of dataset composition was found to be output-dependent: while linear outputs showed incremental improvement with each iteration, highly non-linear outputs (e.g., Maximum Crack Width, $\varepsilon_{h,\max}$) did not demonstrate the same degree of enhancement. This suggests that further data augmentation may be required to improve predictions for such variables.
- The iterative rebalancing procedure successfully enabled broader prediction coverage across the full output domain, reducing model bias (Ψ) and mitigating overfitting by increasing the reliability of performance metrics such as rRMSE and R^2 .
- Cross-validation demonstrated that the chosen ANN structure generalises well to unseen data, with an inter-fold variation of 18% in rRMSE and 6.6% in R^2 .
- Among the tested data transformation methods, the square root transformation yielded the best prediction accuracy. Meanwhile, feature selection based on importance thresholds did not enhance model performance, and the full input set proved most effective. Nonetheless, both approaches should be considered on a case-by-case basis.

- The ANN-PSO model underperformed relative to earlier approaches. Future enhancements may include using PSO for hyperparameter tuning or a hybrid ANN-PSO-BO framework.
- The proposed PCA-NuSVR of Izonin et al. (2024) was so far the best predictive model with notably low values of rRMSE in comparison to the best performing model (ANN-BO), while similar values of R^2 scores were obtained with no significant improvement. The model is further compared to other conventional and ensemble models which it outperformed.

Overall, this chapter has demonstrated the value of combining advanced machine learning models with tailored optimisation and preprocessing strategies for the reliable prediction of complex structural responses. It is considered best practice to evaluate multiple models, transformations, and training techniques, especially when dealing with highly non-linear data, to achieve generalisable and accurate predictions.

Chapter 8

Integration into Building Information Modelling

BIM, a collaborative platform, has emerged as a transformative technology in the Architecture, Engineering, and Construction (AEC) industries. It provides an effective means to create and manage digital representations of the physical and functional characteristics of infrastructure, enabling the modelling, analysis, and communication of complex interdependencies (International Tunnelling and Underground Space Association (2022); Ninić and Meschke (2015)). As discussed previously, predicting building damage induced by tunnelling activities is a critical challenge. The integration of ML-based predictive models into BIM facilitates seamless representation and analysis of tunnelling-induced damage. BIM enables efficient parameter management for scenario analysis, supporting more informed and proactive (near real-time) decision-making. Additionally, it provides a platform for visualising the spatial relationships between the ground, tunnels, and buildings (Ninić et al. (2019, 2017b)). This chapter aims to effectively address the thesis’s objective 7 from Section 1.5, where a step-by-step description of the tool’s key development steps is illustrated.

The first step in the implementation is to develop a fully parametric information

Table 8.1: Parameters for the description of geometric-semantic BIM for tunnelling.

Tunnel	Soil	Building
D (m)	Geometry ($m \times m \times m$)	H (m)
Z_0 (m)	E_s (kPa)	B or W (m)
L_t (m)	c (kPa)	L (m)
e_t (m)	ν_s (-)	E (kPa)
y_{face} (m)	φ ($^\circ$)	ν (-)
σ_τ (KPa)	γ_s (kN/m^3)	e or x_b (m)
Deformation scale factor (-)	K (-)	y_b (m)

model for tunnelling within a suitable BIM environment, providing a user-friendly interface, easy control of the design parameters, and therefore efficient investigation of design scenarios. The state-of-the-art Autodesk BIM design software Revit, alongside the add-on Dynamo, is selected for this purpose (Autodesk (2019)). Dynamo is a built-in graphical algorithm editor in Revit, introducing programming aspects into the environment, which are necessary for integrative design, allowing developers to directly access the software’s Application Programming Interface (API). A fully parametric tunnel model is created, considering ground, tunnel, and building components described with the 22 parameters listed in Table 8.1. While a “dummy” tunnel geometry can be modelled directly in Dynamo, the model would not carry semantic parameters, such as the D parameter, which are required for analytical studies. To tackle this issue, the tunnel and structure are created using a Revit “family”, and Dynamo is used instead to provide and control the required parameters. A family in Revit is a class with parametric definitions and constraints, allowing the definition of specific family attributes for individual family instances (Revit objects). The design model is then created by inserting instances of the family with the assigned corresponding geometric-semantic parameters. This means that multiple instances can also be inserted in the same model (e.g., for multiple buildings). Figure 8.1(a) shows the parametric tunnel model with all geometric parameters being retrieved and controlled in Dynamo using integer sliders.

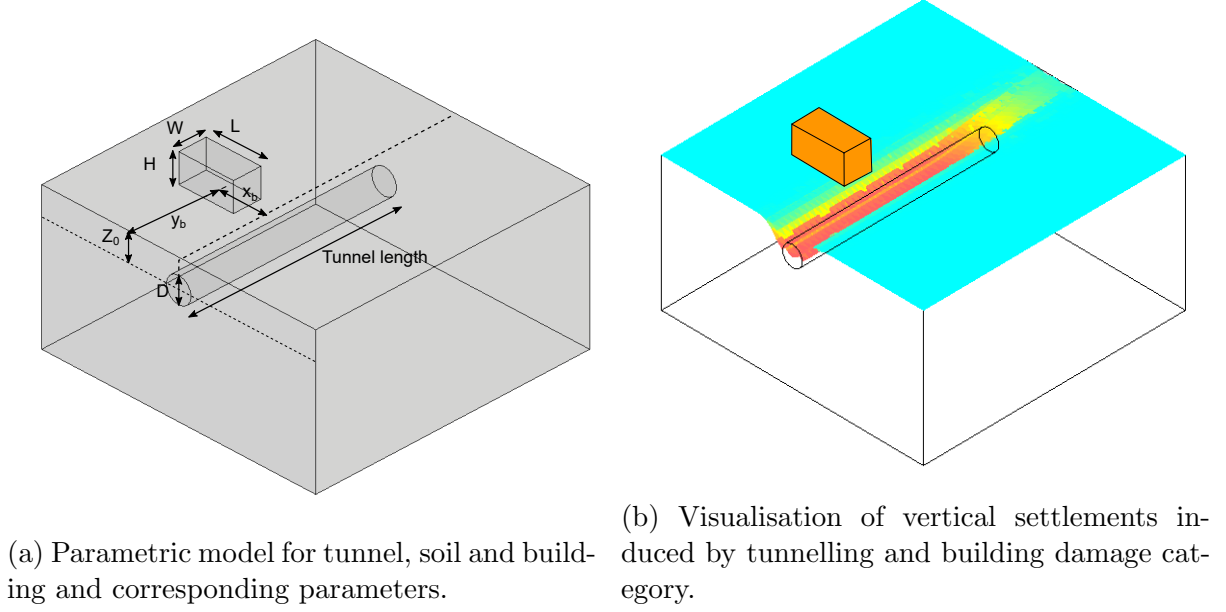


Figure 8.1: Assembly of the whole tunnel-soil-building interaction model in BIM using dynamo in Revit.

8.1 Algorithmic Implementation

The algorithms for calculations of building damage due to tunnelling-induced settlements have been implemented into Dynamo as Python scripts, as shown in Figure 8.2. The parameters used for the generation of the tunnel model are also used as input variables for the implemented algorithms. Therefore, each change in the design model will automatically be applied to the analysis. Moreover, a clear visualisation of the analysis results within the design model is developed (Figure 8.2, “assembly on the right”). The ground settlement profile is defined with coordinates, and a smooth surface is formed by joining the points of coordinates to provide a clearer visual appearance. Then, the surfaces are associated with colours that indicate the ground settlement value (see Figure 8.1b). Likewise, the building damage category is visualised using the colour code shown in Table 8.2, using the case of $\varepsilon_{h,\max}$ global metric for this example.

Figure 8.3 shows the flowchart of the overall implemented procedure for model generation in BIM, including the calculation of settlements from Section 2.1.1, SSI using the EBBEF2p model from Section 2.3.1, and prediction of damage category

Table 8.2: Classification of building damage with the colour code for visualisation using the LSTM damage classification system of (Burland et al. (1977)).

Damage category	Degree of severity	$\varepsilon_{h,\max}$ (%)	Colour code
0	negligible	0 – 0.05	Green
1	very slight	0.05 – 0.075	Yellow
2	slight	0.075 – 0.15	Orange
3	moderate	0.15 – 0.3	Red
4	severe	≥ 0.3	Magenta

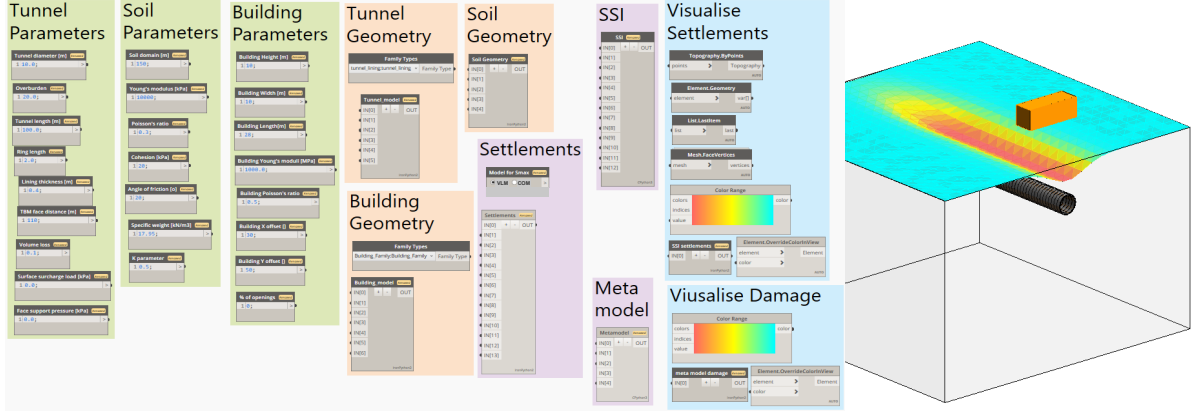


Figure 8.2: User interface for input of design parameters (left) and tunnel model with visualised settlements and damage category (right).

from Section 6 using simulation-based meta-models in Section 7. The BIM model begins generation when settlements are calculated based on the selected empirical model (in this case, the VL model, as per the trained data). The SSI, which depends on soil, tunnel, and building parameters, is internally evaluated as previously incorporated into the FEM models and is used solely for the analysis phase. Hence, model visualisation is restricted to GF settlements, but actual soil deformation beneath the building is computed with SSI effects considered during damage evaluation. Finally, the building damage category is predicted based on the trained meta-models and visualised as outlined in Figure 8.3.

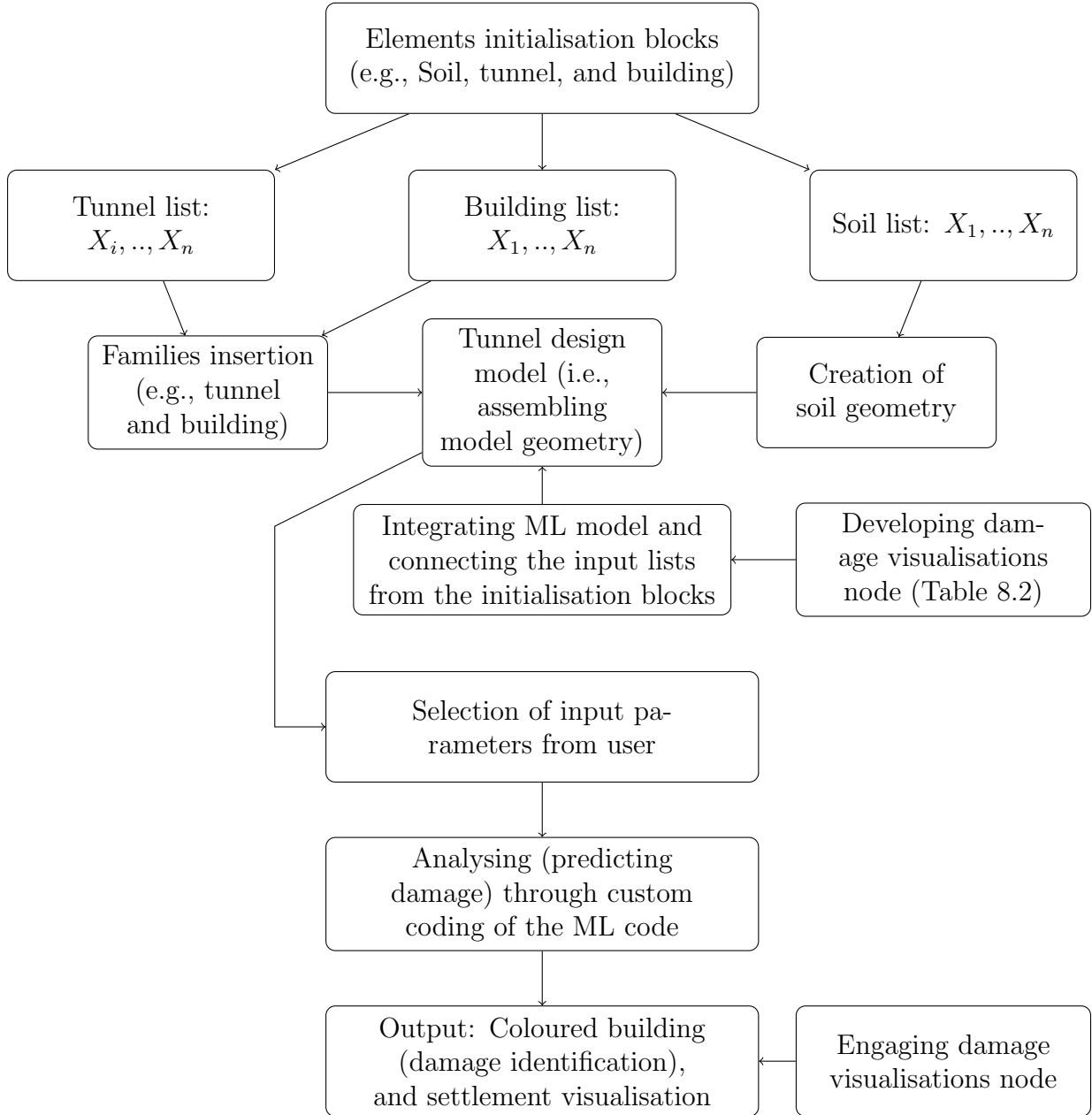


Figure 8.3: Algorithm for model generation and visualisation of settlements and damage category.

8.2 Prediction of Damage Risk in Building Information Modelling Using Meta-Models

As the training of ML models to predict damage based on the input space of Table 5.6, a custom node is implemented using the Dynamo API. The node reads the trained synaptic weights and ANN architecture and performs forward calculations to predict the damage risk based on the model input parameters required by the user.

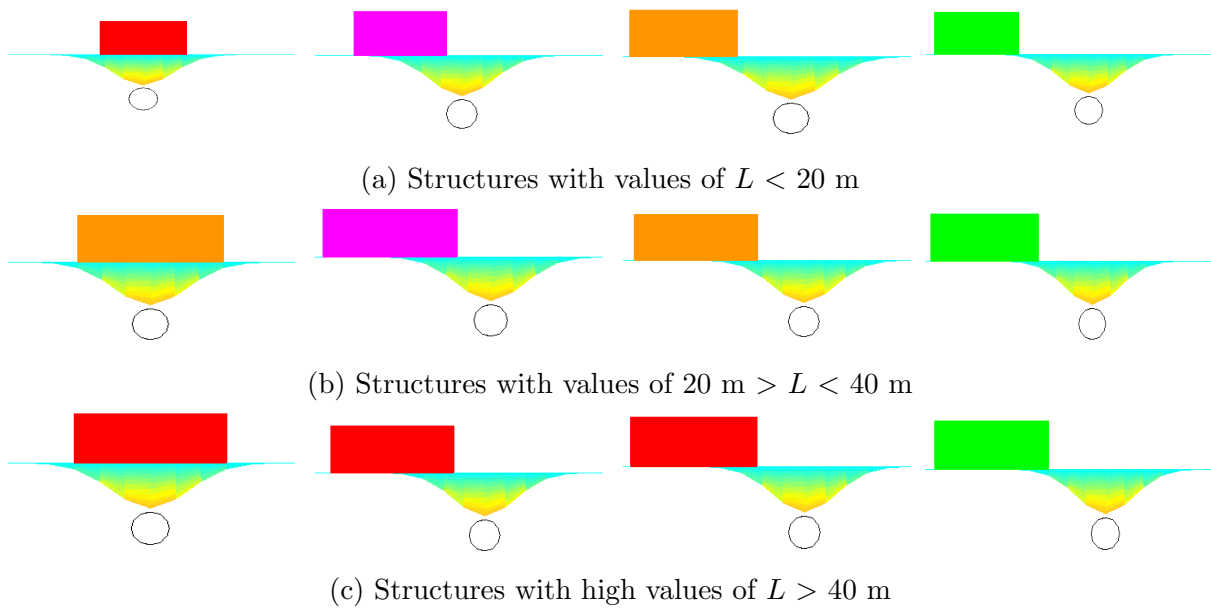


Figure 8.4: Prediction of tunnelling-induced damage category based on meta-model for different values of e (these are $0 D$, $1.5 D$, $2.25 D$, $3 D$, respectively). “left to right in figures” and same for a), b), and c). D is the μ value obtained from the sampling procedure after the iterative procedure and given as 6.6 m.

Figure 8.4 illustrates examples of meta-model predictions of building damage, illustrated in the colour coding defined in Table 8.2. In this example, structures of different L values are illustrated against various locations away from the tunnel centreline. Figure 8.4a illustrates structures with L values smaller than 20 m (anywhere between 8 m and 20 m), Figure 8.4b illustrates structures with L values between 20 m and 40 m, and Figure 8.4c illustrates structures with L values above 40 m (anywhere between 40 m and 60 m, which is the longest building used). It is

observed how damage increases with increased values of L between the three figures. This is also in good agreement with damage distribution plots for both the correlation between parameter L and the global metric $\varepsilon_{h,\max}$ of the original dataset in Figure 6.18c, and the combined dataset in Figure 6.24c. A similar pattern is observed for parameter e , where increasing its values generally leads to reduced damage levels (as evidenced by the reduced range in smaller values in Table 6.10 of the iterative procedure). However, this is less evident due to a carefully selected localised range of values for this parameter.

In these visualisations, the values of all input parameters other than L and e are fixed at their final mean values obtained from the iterative sampling procedure. For instance, the VL parameter is held constant at its converged mean μ value of around the 2.25%. This approach allows for the isolated examination of the influence of L and e on the predicted damage outcomes.

These predictions, which accurately capture output variables similarly achieved through FEM simulation, demonstrate the BIM model's capability to represent the complex deformation mechanisms of non-linear structural behaviour caused by tunnelling. Hence, with a fully parametric model for tunnelling (including the soil, tunnel, and building) and an automatic workflow for assessing damage risk based on the developed meta-model, it is possible to generate design models, assess multiple buildings simultaneously, and visualise the results in the design environment, as shown in Figure 8.5.

8.3 Summary

This chapter fulfils the seventh and final objective of the thesis (Objective 7 in Section 1.5). The procedure presented in this chapter is executed in near real-time (within seconds), while accounting for complex interactions and the non-linear response of masonry structures to settlements using FEM. This demonstrates how the

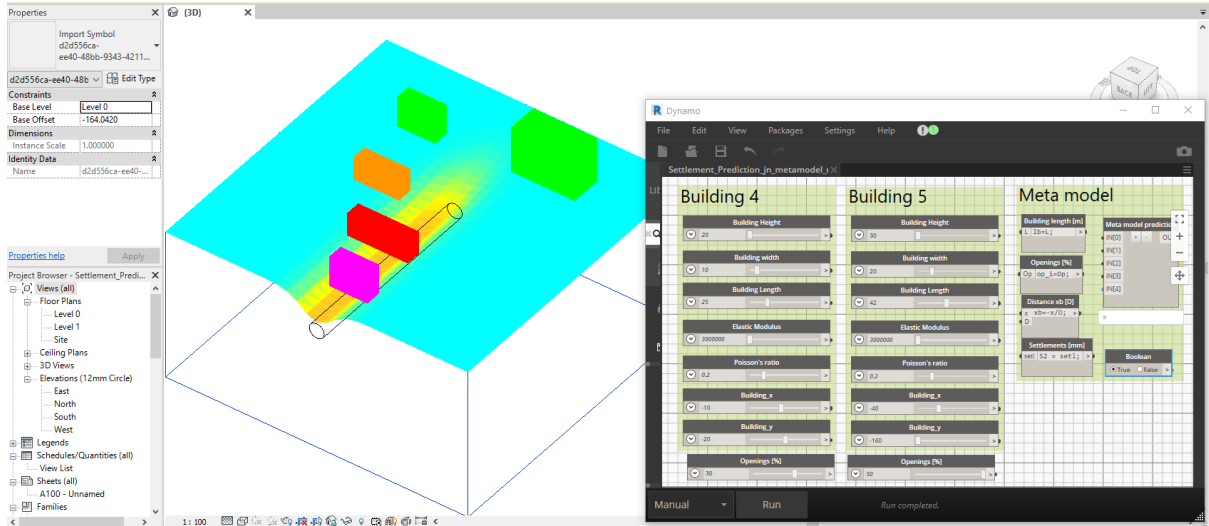


Figure 8.5: Real-time prediction of tunnelling-induced damage risk for multiple buildings in the design environment.

combination of empirical, analytical, and numerical models within the meta-model training stage and its implementation into BIM, along with parametric modelling through Dynamo, is a powerful tool for the quick and efficient investigation of tunnelling design alternatives directly within the design tools. The user-friendly environment further enables designers to easily investigate, evaluate, and optimise a design without needing expert knowledge of the underlying methodology. The visualisation of settlements and the associated risk category is intuitive, making it effective for presenting results to non-experts and supporting the decision-making process.

Chapter 9

Conclusion, Study Limitations, and Future Work

In conclusion, this thesis proposes a comprehensive framework for BIM-based design that seamlessly integrates advanced empirical, analytical, numerical, digital modelling, and ML techniques. This integration enables the prediction of building damage due to tunnelling in near-real time (within seconds) using ML algorithms trained on several non-linear FEM-based numerical simulations.

The benefits of such a method are numerous, particularly during the design stage of tunnelling projects, where fast, accurate, and highly reliable predictions of building damage are essential for assessing the overall impact and design economics of various tunnel design alternatives. This framework not only provides rapid and robust predictions, but also offers a user-friendly interface that simplifies the application of its complex underlying methodologies. As a result, it becomes a widely accessible tool that can be effectively utilised by a broad range of individuals from diverse professional backgrounds, regardless of their expertise in ML, BIM, or programming, which are the core tools employed in its development.

In addition to these contributions, the following key observations were made:

- Several methods are used in the literature to estimate tunnelling-induced set-

lements, address SSI effects, and evaluate building damage in both its local and global forms. Additionally, various methods exist for modelling building material, as well as several SA and ML algorithms which can be used and adopted within this framework, each with its benefits and limitations. Therefore, the results obtained from the analysis performed using the proposed framework are directly tied, in terms of accuracy and reliability, to the strengths and weaknesses of the aforementioned methods. For example, this framework uses a semi-coupled approach that integrates empirical, analytical, and numerical methods. This approach enables the execution of a large number of numerical simulations. In contrast, a fully coupled method, while offering more direct results, significantly increases the required computational effort, which can become prohibitive for large-scale simulations. This would result in fewer data points available for training and analysis, thereby undermining one of the primary objectives of this thesis: generating a sufficiently large and diverse simulation dataset. Similarly, regarding the selected modelling strategy, a macro-modelling approach is used here due to its favourable trade-off between computational cost and accuracy. However, this comes at the expense of reduced precision in evaluating building damage, particularly with respect to local damage indicators. Depending on the scope and objective of a given study, the selection of methods and modelling strategies must carefully weigh such trade-offs. This framework prioritises the ability to evaluate both global and local metrics efficiently. Therefore, high-fidelity methods like micro-modelling, although more accurate, were not considered optimal due to their computational intensity and difficulty in integration. A semi-coupled and macro-modelling strategy was ultimately adopted to ensure computational feasibility while still capturing the necessary behaviours for the large-scale parametric study.

- A thorough investigation of several FEM-based simulations across different

testing scales is deemed necessary, primarily to address the various aspects of numerical inconsistencies. This includes the process of *model validation*, which involves comparing the numerical results against real-world structural performance obtained from laboratory experiments conducted under controlled conditions and/or well-documented field data from similar materials and application contexts. In addition to validation, numerical *verification* is essential to ensure that the implementation of the model is accurate and that the code produces results consistent with known benchmarks, analytical solutions, or previously validated numerical models. Verification confirms that the computational methods used are functioning correctly, regardless of the specific modelling strategies. Depending on the simulation approach, such as implicit (“general”) or explicit (“dynamic”) schemes, each method must be independently verified to ensure the robustness and reliability of the underlying algorithms. It is also critical to maintain consistency within the chosen modelling strategies. For instance, switching from one numerical scheme to another while retaining other modelling assumptions may introduce significant inconsistencies. If not handled carefully, such changes can lead to contradictory outcomes and dangerous misinterpretations. Another key aspect requiring careful consideration is the issue of *mesh sensitivity*. This inherent limitation must be addressed through proper mesh convergence studies to ensure the stability and accuracy of the solution, regardless of mesh resolution. This becomes particularly important in non-linear modelling scenarios involving strain softening and damage degradation behaviours, where mesh refinement can strongly influence the outcome.

- Creating such a comprehensive framework for the functional evaluation of building damage, considering a wide range of input parameters and the integration of complex tunnel-soil-building interactions, undoubtedly involves a vast list of model variables. These parameters and their boundary values

were identified through a comprehensive literature review, and further analyses were conducted to establish dependencies among them. Some parameters are interdependent to varying degrees, and in certain configurations, assigning specific values to one parameter can render others incompatible, preventing their simultaneous inclusion within a valid model. Therefore, employing a traditional OAAT (One-At-A-Time) sampling approach becomes impractical. Instead, *conditional sampling* becomes a fundamental necessity to properly capture these complex interrelationships. Advanced sampling strategies, such as the MLH (Maximin Latin Hypercube) modelling technique presented in this thesis, are essential to ensure thorough coverage of the input domain. This prevents the omission of critical but less frequent model scenarios, while maintaining full adherence to the physical realism and constraints of the modelling framework. Additionally, due to the high volume of samples required to sufficiently train ML algorithms, automating both the model assembly and execution procedures is not merely a convenience but a necessity. In this context, code-friendly platforms such as ABAQUS are highly recommended, as they allow direct model manipulation via scripting. This stands in contrast to other software packages that lack such programmability, making automation tedious or infeasible. Lastly, high-performance computing (HPC) resources are crucial for executing the large number of FEM simulations. The outputs of these simulations, typically stored and processed through automated scripts parsing ‘.txt’ files, form the datasets used to train and evaluate the predictive ML models.

- The evaluation of building damage, whether categorised as “local” or “global,” is rarely explicitly defined in the literature. Moreover, a comprehensive application of a significant range of these methods within a single study is still missing. Although an attempt to develop a damage-risk framework has previously been made Giardina (2013), it still lacks complete adoption across vari-

ous modelling approaches and scenarios. While a unified framework describing damage holistically through both local and global aspects was not pursued in this thesis, as it lies beyond its defined scope, a thorough examination of a wide range of damage assessment methodologies was conducted through simulations. Insightful and broad conclusions regarding damage assessments were presented, highlighting how results differ depending on the chosen method. Furthermore, the method-based distribution of damage outcomes across all simulated samples was monitored. This comprehensive exploration proved particularly useful in identifying which damage criteria are more representative of the dataset as a whole. Moreover, these data-rich distributions revealed a gap in the existing literature, which led to the development of a novel iterative procedure for data rebalancing. By back-analysing the simulation data, strategies were identified to improve the reliability of model outputs by increasingly populating previously underrepresented areas in the output space. This yielded more informative outputs that, in turn, were used to better train ML algorithms. As demonstrated, model accuracy improves not only with the total number of sample points but also with the uniformity of their distribution across the spectrum of potential outputs.

- From the analysis presented in this thesis, the application of multiple SA and ML algorithms on the same dataset is recommended. These should be compared and evaluated based on the accuracy of their predictions relative to the given input-output relationships. The best-performing models are then selected for further analysis, SA for the iterative procedure and ML for integration into the BIM framework. Furthermore, in addition to using basic ML algorithms (i.e., those with fixed structure and no additional tuning), which can be applied for either regression (predicting a continuous value, as done in this thesis) or classification (predicting an output class/category, outside the scope of this thesis), optimisation techniques, data transformation, and cross-

validation are all essential for improving model reliability and performance. It is therefore strongly advised to incorporate at least a few of these techniques when conducting any form of ML training, especially in cases involving highly non-linear input-output relationships, such as those examined in this work. These techniques assist in identifying hidden inconsistencies, such as data overfitting and output biases, which are common in advanced statistical models. Additionally, they help to enhance model accuracy by allowing flexibility in determining the most suitable structure for the dataset under consideration.

- The implementation of the developed ML model (best performing) into the BIM platform simplifies its complex, code-based nature, which is typically challenging to operate directly from the source. Additionally, the integration of a parametric modelling framework, allowing users to select input values, within the platform, along with colour-coded visualisation of damage levels, effectively transforms the process from a monotonic, expert-exclusive tool into a rich, informative, colour-driven, and user-friendly framework. This makes it accessible to a broad audience, including individuals with little to no understanding of ML techniques or advanced code-based tools such as FEM or Python.

9.1 Study Limitations

Some notable limitations of this work include:

- The current state of the framework integrates rather traditional methods, including the VL and EBBEF2p methods for calculating settlements and SSI, respectively. Integrating additional methods such as the COM or a non-linear or elastic perfectly plastic SSI allows for a more versatile and rich tool that can be used with various input elements. However, this is reflected in an

exponential increase in numerical samples, added complexity of modelling, additional parameters that are not straightforward to determine, and increased computational demand, which can still be a challenging task to date.

- Although using a semi-coupled approach (due to the EBBEF2p SSI model) has proven to be computationally effective, and sufficiently accurate based on the validated models, it still involves trade-offs in precision compared to fully coupled methods. The accuracy of SSI predictions can be limited due to the potential negligence of a more severe non-linear behaviour between the soil and the building interface. This plays an important role in identifying the accurate damage induced to the building without restrictions caused by the foundation's constraints.
- As mentioned before, the iterative procedure, although effective in providing a rebalanced dataset, lacks a rigorous optimisation procedure for parameters such as θ , σ_{X_i} , the number of samples, and the total number of iterations, which directly impacts the effective increase in the accuracy of both SA and ML training. Especially when increasing the number of samples or changing values of θ or σ_{X_i} could potentially present valuable insights that might diverge the ultimate solution towards a different direction; however, this was not further investigated and could be the scope of future work.
- The prediction accuracy of ML is still considered to be notably high, particularly in the case of highly non-linear output variables such as Maximum Crack Width and $\varepsilon_{h,\max}$. This is a direct result of two factors: the total number of samples used for testing and training the ML model, and the quality of output distribution along the sample points. This can be addressed by increasing the total number of samples while closely observing the output distributions and adjusting accordingly using methods such as the iterative procedure in the case of highly skewed data output.
- Lastly, the current modelling strategy lacks the integration of 3D modelling

aspects of buildings, as well as considerations for other types of buildings such as concrete-filled framed structures and steel structures. In addition, different types of foundations, such as isolated footing or pile foundations, were not addressed, which can alter the overall impact of tunnelling due to the different horizontal resistance to ground subsidence. Lastly, different tunnel configurations (such as curvature, twin tunnelling, incline and decline tunnelling) were not considered, nor were long-term settlements due to consolidation effects or time-dependent tunnelling activities such as when tunnels are actively passing underneath a building, during which the building might experience tension release phenomena during excavation operations.

9.2 Generalizability and Computational Implications

Another notable limitation relates to the generalizability of this study's findings when applied to different geotechnical and structural scenarios. Extending the current framework to accommodate alternative foundation types, such as pile foundations instead of the currently modelled shallow foundations, introduces significant implications. Specifically, this would require adjustments to the FEM input space, incorporating additional parameters such as pile number, length, diameter, spacing, and stiffness characteristics. The integration of these parameters would exponentially increase the number of simulations needed, proportional to the number of additional inputs multiplied by the required numerical samples. Moreover, if a fully coupled SSI approach is adopted, explicit modelling of piles within the surrounding soil strata would be mandatory, substantially increasing computational complexity, runtime, and memory usage. Conversely, employing a decoupled SSI approach similar to that presented in this study would mitigate some computational overhead by integrating only settlement outputs from SSI analyses directly onto building

façades, thereby avoiding explicit pile modelling. Nevertheless, changing foundation type inherently demands a full rerun of FEM simulations and subsequent retraining and revalidation of ML models. The associated computational costs, modelling complexity, and required analysis time would significantly increase, potentially limiting the broader and immediate applicability of the current results to these more complex or fundamentally different geotechnical scenarios.

9.3 Future Work

To address these limitations, the following tasks are recommended as future work of this thesis.

- Creating a new set of data using a fully coupled tunnel-soil-structure interaction model, using a 2-step approach such as the case of Yu et al. (2025), or even by modelling the tunnel while adopting a non-linear material model for the soil, and, if possible, the non-linear tunnel-soil interface interaction. Although this might still be a challenging task to date, even by adopting a linear model for the soil and tunnel-soil interfaces, this would still provide useful insights into the soil-building interaction, as well as building damage, which can be directly compared to the current set of available data. This would provide better quality data (due to coupling) for the training of ML models and SA studies and provide definitive answers about the extent to which the EBBEF2p no longer (if so) presents accurate predictions of building damage and becomes ultimately unreliable for increased non-linearities.
- A rigorous optimisation strategy (e.g., gradient-based, variationally consistent, or AL) for the iterative procedure must be devised to determine the most effective values of the method parameters (e.g., θ and σ_{X_i} values), as well as the total number of iterations and samples for each iteration needed. This ensures the iterative procedure is maximised for effectiveness without risking

missing important insights drawn from individual iterations, which directly impact the results of the SA studies and dictate the direction of resampling for individual parameters in subsequent iterations.

- As mentioned before, in addition to using a fully coupled modelling approach to model interaction, increasing the number of samples will provide a larger set of data from which better prediction performance of different ML algorithms can be achieved. Furthermore, in relation to ML algorithms, further improvements can be made, such as increasing the search space of model hyperparameters, attempting dropout techniques, or applying early stopping criteria, etc.
- The current implementation of the framework in BIM is limited to a set of only a handful of parameters (precisely, the L , e , O , and $S_{v,\max}$ which further comprises D , VL, and i_x) with only one damage output variable to be predicted per building (that is, $\varepsilon_{h,\max}$). This can further be improved by incorporating the additional parameters already presented in this thesis, as well as any additional parameters from further development of the model as previously discussed. Furthermore, the output layer of the ML can be increased to more than one node (output), accounting for the various local and global damage output values, providing a more informative and flexible framework depending on the level of detail required or design use.

Bibliography

- S Acikgoz, A Franza, M J DeJong, and R Mair. Cracked equivalent beam models for assessing tunneling-induced damage in masonry buildings. *Journal of Geotechnical and Geoenvironmental Engineering*, 147(2):04020167, 2021. doi: 10.1061/(ASCE)GT.1943-5606.0002443.
- D Addessi, S Marfia, and E Sacco. A plastic nonlocal damage model. *Computer Methods in Applied Mechanics and Engineering*, 191(13):1291–1310, 2002. ISSN 0045-7825. doi: [https://doi.org/10.1016/S0045-7825\(01\)00325-5](https://doi.org/10.1016/S0045-7825(01)00325-5). URL <https://www.sciencedirect.com/science/article/pii/S0045782501003255>.
- A Alsahly, F Hegemann, M König, and G Meschke. Integrated bim-to-fem approach in mechanised tunnelling. *Geomechanics and Tunnelling*, 13(2):212–220, 2020. doi: 10.1002/geot.202000002.
- A Antoniadis, S Lambert-Lacroix, and J M Poggi. Random forests for global sensitivity analysis: A selective review. *Reliability Engineering System Safety*, 206:107–312, 2021. ISSN 0951-8320. doi: <https://doi.org/10.1016/j.ress.2020.107312>. URL <https://www.sciencedirect.com/science/article/pii/S0951832020308073>.
- P B Attewell and J P Woodman. Predicting the dynamics of ground settlement and its derivatives caused by tunnelling in soil. *Ground Engineering*, 15:13, 1982.

- P B Attewell, J Yeates, and A R Selby. *Soil movements induced by tunnelling and their effects on pipelines and structures*. Blackie, 1986.
- Autodesk. Autodesk Revit, 2019. URL <http://www.autodesk.co.uk/products/revit-family/>.
- H X H Bao, J P Larsson, and V Wong. Light at the end of the tunnel:the impacts of expected major transport improvements on residential property prices. *Urban Studies*, 58:2971–2990, 11 2021. ISSN 1360063X. doi: 10.1177/0042098020967308.
- G Baptiste, M Bertrand, and S P Philippe. Correlation and variable importance in random forests. *Statistics and Computing*, 27, 05 2017. doi: 10.1007/s11222-016-9646-1.
- K J Bathe, E Ramm, and E L Wilson. Finite element formulations for large deformation dynamic analysis. *International Journal for Numerical Methods in Engineering*, 9(2):353 – 386, 1975. doi: 10.1002/nme.1620090207. URL <https://www.scopus.com/inward/record.uri?eid=2-s2.0-0016451238&doi=10.1002%2fnme.1620090207&partnerID=40&md5=8ae6e76bc961a249b4f804b254f6a792>.
- Z P Bazant and B H Oh. Crack band theory for fracture of concrete. *Materials and Structures*, 93(16):155–177, 1983.
- H Behnam, J S Kuang, and B Samali. Parametric finite element analysis of rc wide beam-column connections. *Computers Structures*, 205:28–44, 2018. ISSN 0045-7949. doi: <https://doi.org/10.1016/j.compstruc.2018.04.004>. URL <https://www.sciencedirect.com/science/article/pii/S004579491830004X>.
- D Boldini, N Losacco, S Bertolin, and A Amorosi. Finite element modelling of tunnelling-induced displacements on framed structures. *Tunnelling and Underground Space Technology*, 80:222 – 231, 2018. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2018.06.019>.

- M D Boscardin and E J Cording. Building response to excavation-induced settlement. *Journal of Geotechnical Engineering*, 115(1):1–21, 3 1989. ISSN 0733-9410. doi: 10.1061/(ASCE)0733-9410(1989)115:1(1).
- J E Bowles. *Foundation analysis and design*. McGraw-Hill, fifth edition edition, 1996. ISBN 0079122477.
- B Broms and H Bennermark. Stability of clay at vertical openings. *Journal of Soil Mechanics & Foundations Div*, 93:71–94, 1967.
- H-G Bui and G Meschke. A parallelization strategy for hydro-mechanically coupled mechanized tunneling simulations. *Computers and Geotechnics*, 120:103378, 2020. ISSN 0266-352X. doi: <https://doi.org/10.1016/j.compgeo.2019.103378>.
- J Burland and C Wroth. Settlement of buildings and associated damage. In *Conference on Settlement of Structures*. Proc. of the Conference on Settlement of Structures, 1975.
- J B Burland. Assessment of risk of damage to buildings due to tunnelling and excavation, 1997.
- J B Burland and C P Wroth. Settlement of buildings and associated damage. pages 611–654. Pentech Press, 1974.
- J B Burland, B B Broms, and V F B de Mello. Behaviour of foundations and structures. pages 495–546, 3 1977.
- J B Burland, J Standing, and F M Jardine. *Building response to tunnelling : case studies from construction of the Jubilee Line Extension, London. V1*, volume 2. T. Telford, 2001a. ISBN 0727730177.
- J B Burland, J Standing, and F M Jardine. *Building response to tunnelling : case studies from construction of the Jubilee Line Extension, London V2*. T. Telford, 2001b. ISBN 0727730177.
- J B Burland, J R Standing, and F M Jardine. Assessing the risk of building damage due to tunnelling - lessons from the jubilee line extension, london. In

Proceedings of the 2nd International Conference on Soil Structure Interaction in Civil Engineering, pages 11–38, Zurich, 2002.

- C Camós, O Špačková, D Straub, and C Molins. Probabilistic approach to assessing and monitoring settlements caused by tunneling. *Tunnelling and Underground Space Technology*, 51:313–325, 2016. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2015.10.041>.
- B T Cao, M Obel, S Freitag, P Mark, and G Meschke. Artificial neural network surrogate modelling for real-time predictions and control of building damage during mechanised tunnelling. *Advances in Engineering Software*, 149, 3 2020a. ISSN 18735339. doi: [10.1016/j.advengsoft.2020.102869](https://doi.org/10.1016/j.advengsoft.2020.102869).
- B T Cao, M Obel, S Freitag, L Heußner, G Meschke, and P Mark. Real-time risk assessment of tunneling-induced building damage considering polymorphic uncertainty. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 8, 3 2022. ISSN 23767642. doi: [10.1061/ajrua6.0001192](https://doi.org/10.1061/ajrua6.0001192).
- L Cao, D Zhang, and Q Fang. Semi-analytical prediction for tunnelling-induced ground movements in multi-layered clayey soils. *Tunnelling and Underground Space Technology*, 102:103446, 2020b. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2020.103446>.
- H Chakeri and B Ünver. A new equation for estimating the maximum surface settlement above tunnels excavated in soft ground. *Environmental Earth Sciences*, 71:3195–3210, 04 2013. doi: [10.1007/s12665-013-2707-2](https://doi.org/10.1007/s12665-013-2707-2).
- A W Y Chan, C H C Yeung, A J Westmoreland, S W Fok, C C W Ng, and F Guedon. Construction of the tuen mun–chek lap kok link sub-sea tunnels in hong kong. *HKIE Transactions Hong Kong Institution of Engineers*, 28: 139–153, 2021. ISSN 1023697X. doi: [10.33430/V28N3THIE-2019-0043](https://doi.org/10.33430/V28N3THIE-2019-0043).
- J A Charles and H D Skinner. Settlement and tilt of low-rise buildings. *Geotechnical Engineering*, 157:65–75, 11 2004.

- C Y Cheng, G R Dasari, Y K Chow, and C F Leung. Finite element analysis of tunnel–soil–pile interaction using displacement controlled model. *Tunnelling and Underground Space Technology*, 22:450–466, 2007. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2006.08.002>.
- M A Crisfield. Accelerated solution techniques and concrete cracking. *Comp. Meth. Appl. Mech. Engrg.*, 33:585–607, 1982.
- J Daller, M Zibert, C Exinger, and M Lah. Implementation of BIM in the tunnel design – engineering consultant’s aspect. *Geomechanics and Tunnelling*, 9: 674–683, 2016.
- A M D’Altri, V Sarhosis, G Milani, et al. Modeling strategies for the computational analysis of unreinforced masonry structures: Review and classification. *Archives of Computational Methods in Engineering*, 27(4):1153–1185, 2020. ISSN 1134-3060.
- Dassault Systèmes Simulia Corp. *ABAQUS Documentation*. Dassault Systèmes, 2021. Available from Dassault Systèmes, <https://www.3ds.com/products-services/simulia/products/abaqus/>.
- M J DeJong, G Giardina, B Chalmers, D Lazarus, D Ashworth, and R J Mair. Impact of the crossrail tunnelling project on masonry buildings with shallow foundations. *Proceedings of the Institution of Civil Engineers-Geotechnical Engineering*, 172(5):402–416, 2019.
- H Demirci, S Bhattacharya, D Karamitros, N Alexander, and R Singh. Finite element model of buried pipelines crossing strike-slip faults by abaqus/explicit. 06 2018.
- C Eastman, P Teicholz, R Sacks, and K Liston. *BIM handbook: A guide to Building Information Modeling for owners, managers, designers, engineers, and contractors*. Wiley & Sons, 2008. doi: 10.1002/9780470261309.
- S Fabozzi, S A Biancardo, R Veropalumbo, and E Bilotta. I-bim based approach for geotechnical and numerical modelling of a conventional tunnel excavation.

- Tunnelling and Underground Space Technology*, 108:103723, 2021. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2020.103723>.
- V Fargnoli, C G Gragnano, D Boldini, and A Amorosi. 3d numerical modelling of soil–structure interaction during epb tunnelling. *Géotechnique*, 65(1):23–37, 2015. doi: 10.1680/geot.14.P.091.
- P H Feenstra. Computational aspects of biaxial stress in plain reinforced concrete. 3 1993.
- P.H. Feenstra and R. De Borst. A composite plasticity model for concrete. *Int. J. Solids Structures*, 33(5):707–730, 1996.
- M Feurer and F Hutter. *Hyperparameter Optimization*, pages 3–33. 05 2019. ISBN 978-3-030-05317-8. doi: 10.1007/978-3-030-05318-5_1.
- A Franza and A M Marshall. Empirical and semi-analytical methods for evaluating tunnelling-induced ground movements in sands. *Tunnelling and Underground Space Technology*, 88:47–62, 2019. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2019.02.016>.
- A Franza, S Ritter, and M J Dejong. Continuum solutions for tunnel-building interaction and a modified framework for deformation prediction. *Géotechnique*, 70:108–122, 2020. doi: 10.1680/jgeot.17.P.279.
- J Franzius, D M Potts, T I Addenbrooke, and J B Burland. The influence of building weight on tunnelling-induced ground and building deformation. *SOILS AND FOUNDATIONS*, 44:25–38, 2004. doi: 10.3208/sandf.44.25.
- J N Franzius. Behaviour of buildings due to tunnel induced subsidence. 2003.
- S Freitag, B T Cao, J Ninić, and G Meschke. Recurrent neural networks and proper orthogonal decomposition with interval data for real-time predictions of mechanised tunnelling processes. *Computers & Structures*, 207:258 – 273, 2018. ISSN 0045-7949. doi: <https://doi.org/10.1016/j.compstruc.2017.03.020>. CIVIL-COMP 2017.

- W W Frischmann, J E Hellings, S Gittoes, and C Snowden. Protection of the mansion house against damage caused by ground movements due to the docklands light railway extension. *Proceedings of the Institution of Civil Engineers - Geotechnical Engineering*, 107:65–76, 1994.
- A Furtado, H Rodrigues, A Arêde, and H Varum. Experimental evaluation of out-of-plane capacity of masonry infill walls. *Engineering Structures*, 111:48–63, 2016. ISSN 0141-0296. doi: <https://doi.org/10.1016/j.engstruct.2015.12.013>. URL <https://www.sciencedirect.com/science/article/pii/S0141029615007774>.
- A Gamra, J Ninić, and B Ghiassi. Optimising synthetic datasets for machine learning-based prediction of building damage due to tunnelling. *Tunnelling and Underground Space Technology*, 152:105961, 2024. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2024.105961>. URL <https://www.sciencedirect.com/science/article/pii/S0886779824003791>.
- A Genikomsou and M A Polak. Finite element simulation of concrete slabs with various placement and amount of shear bolts. *Procedia Engineering*, 193:313–320, 2017. ISSN 1877-7058. doi: <https://doi.org/10.1016/j.proeng.2017.06.219>. URL <https://www.sciencedirect.com/science/article/pii/S1877705817327832>. International Conference on Analytical Models and New Concepts in Concrete and Masonry Structures.
- A S Genikomsou and M A Polak. Finite element analysis of punching shear of concrete slabs using damaged plasticity model in abaqus. *Engineering Structures*, 98:38–48, 2015. ISSN 0141-0296. doi: <https://doi.org/10.1016/j.engstruct.2015.04.016>.
- B Ghiassi and G Milani. *Numerical modeling of masonry and historical structures*. Matthew Deans, 2019. ISBN 978-0-08-102440-9.
- G Giardina. Modelling of settlement induced building damage. 3 2013.
- G Giardina, M Hendriks, and J G Rots. Sensitivity study on tunnelling induced

- damage to a masonry facade. *Engineering Structures*, 89:111–129, 3 2015. ISSN 1873-7323. doi: 10.1016/j.engstruct.2015.01.042.
- G Giardina, M J DeJong, B Chalmers, B Ormond, and R J Mair. A comparison of current analytical methods for predicting soil-structure interaction due to tunnelling. *Tunnelling and Underground Space Technology*, 79:319 – 335, 2018. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2018.04.013>.
- K H Goh and R J Mair. Building damage assessment for deep excavations in singapore and the influence of building stiffness. *Geotechnical Engineering*, 42: 1–12, 2011.
- V Guglielmetti, P Grasso, A Mahtab, S Xu, and L Soldo. *Mechanized Tunnelling in Urban Areas: Design methodology and construction control*. 3 2008. ISBN 9780203938515.
- M Hajjar, A N Hayati, M M Ahmadi, and S A Sadrnejad. Longitudinal settlement profile in shallow tunnels in drained conditions. *International Journal of Geomechanics*, 15(6):04014097, 2015. doi: 10.1061/(ASCE)GM.1943-5622.0000447.
- J Herman, W Usher, and Others. Salib documentation release 1.4.6.1, 2022.
- High Speed Two (HS2) Ltd. A guide to tunnelling costs, 2017. Available at: <https://www.britishtunnelling.org.uk/>.
- A Hillerborg, M Modeer, 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*, 6:773–782, 1976.
- S Hou, Y Liu, and Q Yang. Real-time prediction of rock mass classification based on tbm operation big data and stacking technique of ensemble learning. *Journal of Rock Mechanics and Geotechnical Engineering*, 14:123–143, 2 2022. ISSN 16747755. doi: 10.1016/j.jrmge.2021.05.004.
- M Huang, J Ninić, and Q Zhang. Bim, machine learning and computer vision

- techniques in underground construction: Current status and future perspectives. *Tunnelling and Underground Space Technology*, 108, February 2021. ISSN 0886-7798. doi: 10.1016/j.tust.2020.103677.
- International Tunnelling and Underground Space Association. *BIM in Tunnelling – Guideline for Bored Tunnels – Vol 1*. ITA-AITES, 2022. URL <https://about.ita-aites.org/publications/wg-publications/2127/bim-in-tunnelling-guideline-for-bored-tunnels-vol-1>.
- I Izonin, A Gamra, O Boychuk, J Ninić, R Tkachenko, and S Mitoulis. PCA-NuSVR Framework for Predicting Local and Global Indicators of Tunnelling-induced Building Damage. In *1st International Conference on Smart Automation & Robotics for Future Industry (SMARTINDUSTRY 2024)*, pages 32–46. CEUR-WS.org, 2024.
- O Jenck and D Dias. Analyse tridimensionnelle en différences finies de l’interaction entre une structure en béton et le creusement d’un tunnel à faible profondeur: 3d-finite difference analysis of the interaction between concrete building and shallow tunnelling. *Géotechnique*, 54:519–528, 2004. doi: 10.1680/geot.2004.54.8.519.
- I T Jolliffe and J Cadima. Principal component analysis: a review and recent developments. *Phil. Trans. R. Soc. A.*, 374(2065):20150202, Apr 2016. doi: 10.1098/rsta.2015.0202.
- H B Kaushik, D C Rai, and S K Jain. Stress-strain characteristics of clay brick masonry under uniaxial compression. *Journal of Materials in Civil Engineering*, 19:728–739, 2007. doi: 10.1061/(ASCE)0899-1561(2007)19:9(728).
- C Koch, A Vonthron, and M König. A tunnel information modelling framework to support management, simulations and visualisations in mechanised tunnelling projects. *Automation in Construction*, 83:78–90, 2017.
- R Kohavi. A study of cross-validation and bootstrap for accuracy estimation and

- model selection. In *International Joint Conference on Artificial Intelligence*, 1995. URL <https://api.semanticscholar.org/CorpusID:2702042>.
- M Kubat and S Matwin. Addressing the curse of imbalanced training sets: One-sided selection. *Fourteenth International Conference on Machine Learning*, 06 2000.
- S Lagomarsino, A Penna, A Galasco, and S Cattari. Tremuri program: An equivalent frame model for the nonlinear seismic analysis of masonry buildings. *Engineering Structures*, 56:1787–1799, 2013. ISSN 0141-0296. doi: <https://doi.org/10.1016/j.engstruct.2013.08.002>. URL <https://www.sciencedirect.com/science/article/pii/S0141029613003696>.
- N Loganathan and H G Poulos. Analytical prediction for tunneling-induced ground movements in clays. *Journal of Geotechnical and Geoenvironmental Engineering*, 124:846–856, 1998. doi: 10.1061/(ASCE)1090-0241(1998)124:9(846).
- N Loganathan, H G Poulos, and D P Stewart. Centrifuge model testing of tunnelling-induced ground and pile deformations. *Géotechnique*, 50:283–294, 2000. doi: 10.1680/geot.2000.50.3.283. URL <https://doi.org/10.1680/geot.2000.50.3.283>.
- P B Lourenco. Computational strategy for masonry structures. 3 1996.
- P B Lourenço, J G Rots, and J Blaauwendraad. Two approaches for the analysis of masonry structures: Micro and macro-modeling. *Heron*, 40 (4), 1995. ISSN 0046-7316. URL <http://resolver.tudelft.nl/uuid:3c98b29a-3c76-47b4-9c6b-b2f6b178ff1f>.
- T Luís, R P Rita, P Bernhard, and B Paula. Smote for regression. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 977–985, Prague, Czech Republic, 2013. Springer. doi: 10.1007/978-3-642-40669-0_33.
- A Mahmoodzadeh, H R Nejati, M Mohammadi, H H Ibrahim, S Rashidi, and B F Ibrahim. Forecasting face support pressure during epb shield tunneling

- in soft ground formations using support vector regression and meta-heuristic optimization algorithms. *Rock Mechanics and Rock Engineering*, 55:6367–6386, 10 2022. ISSN 1434453X. doi: 10.1007/s00603-022-02977-7.
- R J Mair, R N Taylor, and A Bracegirdle. Sub-surface settlement profiles above tunnels in clays. *Geotechnik*, 43(2):315–320, 1993.
- R J Mair, R N Taylor, and J B Burland. Prediction of ground movements and assessment of risk of building damage due to bored tunnelling. pages 713–718, 1996.
- J Manchego and S Pari. *Análisis experimental de muros de albañilería confinada en viviendas de baja altura en Lima, Perú*. PhD thesis, MS thesis, Dept. of Civil Engineering, Pontificia Universidad Católica del Perú, 2016.
- A M Marshall, R Farrell, A Klar, and R Mair. Tunnels in sands: The effect of size, depth and volume loss on greenfield displacements. *Geotechnique*, 62:385–399, 5 2012. ISSN 00168505. doi: 10.1680/geot.10.P.047.
- B H Menze, B M Kelm, R Masuch, U Himmelreich, P Bachert, W Petrich, and M E Ladd. A comparison of random forest and its gini importance with standard chemometric methods for the feature selection and classification of spectral data. *BMC Bioinformatics*, 10(1):213, 2009. doi: 10.1186/1471-2105-10-213.
- G Meschke. From advance exploration to real time steering of tbms: A review on pertinent research in the collaborative research center “interaction modeling in mechanized tunneling”. *Underground Space*, 3(1):1 – 20, 2018. ISSN 2467-9674. doi: <https://doi.org/10.1016/j.undsp.2018.01.002>. Computational Methods in Mechanized Tunneling.
- P Milillo, G Giardina, M J DeJong, D Perissin, and G Milillo. Multi-temporal In-SAR structural damage assessment: The London crossrail case study. *Remote Sensing*, 10(2):287, 2018.
- S Miliziano and A de Lillis. Predicted and observed settlements induced by the mechanized tunnel excavation of metro line c near s. giovanni station in rome.

- Tunnelling and Underground Space Technology*, 86:236 – 246, 2019. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2019.01.022>.
- M D Morris. Factorial sampling plans for preliminary computational experiments. *Technometrics*, 33(2):161–174, 1991.
- J Ninić and G Meschke. Model update and real-time steering of tunnel boring machines using simulation-based meta models. *Tunnelling and Underground Space Technology*, 45:138 – 152, 2015. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2014.09.013>.
- J Ninić, J Stascheit, and G Meschke. Beam-solid contact formulation for finite element analysis of pile-soil interaction with arbitrary discretization. *International Journal for Numerical and Analytical Methods in Geomechanics*, 38(14): 1453–1476, 2014.
- J Ninić, S Freitag, and G Meschke. A hybrid finite element and surrogate modelling approach for simulation and monitoring supported tbm steering. *Tunnelling and Underground Space Technology*, 63:12–28, 2017a. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2016.12.004>.
- J Ninić, C Koch, and J Stascheit. An integrated platform for design and numerical analysis of shield tunnelling processes on different levels of detail. *Advances in Engineering Software*, 112:165–179, 2017b.
- J Ninić, C Koch, and W Tizani. Meta models for real-time design assessment within an integrated information and numerical modelling framework. *Lecture Notes in Computer Science*, 10836, 07 2018.
- J Ninić, H-G Bui, C Koch, and G Meschke. Computationally efficient simulation in urban mechanized tunneling based on multilevel bim models. *Journal of Computing in Civil Engineering*, 33, 05 2019. doi: 10.1061/(ASCE)CP.1943-5487.0000822.
- J Ninić, C Koch, A Vonthron, W Tizani, and M König. Integrated parametric multi-level information and numerical modelling of mechanised tunnelling projects.

- Advanced Engineering Informatics*, 43:101011, 2020. ISSN 1474-0346. doi: <https://doi.org/10.1016/j.aei.2019.101011>.
- J Ninić, A Alsahly, A Vonthron, H-G Bui, C Koch, M König, and G Meschke. From digital models to numerical analysis for mechanised tunnelling: A fully automated design-through-analysis workflow. *Tunnelling and Underground Space Technology*, 107:103622, 2021. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2020.103622>.
- J Ninić, A Gamra, and B Ghiassi. Real-time assessment of tunnelling-induced damage to structures within the building information modelling framework. *Underground Space (China)*, 14:99–117, 3 2024. ISSN 24679674. doi: [10.1016/j.undsp.2023.05.010](https://doi.org/10.1016/j.undsp.2023.05.010).
- M Obel, M A Ahrens, and P Mark. Metamodel-based prediction of structural damages due to tunneling-induced settlements. *ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering*, 6, 3 2020. ISSN 23767642. doi: [10.1061/ajrua6.0001092](https://doi.org/10.1061/ajrua6.0001092).
- M P O'Reilly and B M New. Settlements above tunnels in the united kingdom - their magnitude and prediction. pages 55–64. The Institution of Mining and Metallurgy, 1982.
- R B Peck. Deep excavations and tunnelling in soft ground. *Geotextiles and Geomembranes*, 1:311 – 375, 1969. ISSN 02661144. doi: [10.1016/0266-1144\(84\)90012-8](https://doi.org/10.1016/0266-1144(84)90012-8).
- F Pinto and A J Whittle. Ground movements due to shallow tunnels in soft ground. i: Analytical solutions. *Journal of Geotechnical and Geoenvironmental Engineering*, 140(4):04013040, 2014. doi: [10.1061/\(ASCE\)GT.1943-5606.0000948](https://doi.org/10.1061/(ASCE)GT.1943-5606.0000948).
- R V D Pluijm. Material properties of masonry and its components under tension and shear. In *Proceedings 6th Canadian Masonry Symposium*, pages 675–686, Saskatoon, Canada, 15–17 June 1992. University of Saskatchewan.
- D M Potts and T I Addenbrooke. A structure's influence on tunnellinginduced

- ground movements. *Proceedings of the Institution of Civil Engineers: Geotechnical Engineering*, 125(2):109–125, 1997. doi: 10.1680/igeng.1997.29233.
- D M Potts and L Zdravković. *Finite Element analysis in geotechnical engineering: application*, volume 2. Thomas Telford, 2001.
- A J Powderham. An overview of the observational method: development in cut and cover and bored tunnelling projects. *Géotechnique*, 44(4):619–636, 1994. doi: 10.1680/geot.1994.44.4.619.
- S Providakis, C D F Rogers, and D N Chapman. Predictions of settlement risk induced by tunnelling using BIM and 3D visualization tools. *Tunnelling and Underground Space Technology*, 92:103049, 2019. ISSN 0886-7798. doi: <https://doi.org/10.1016/j.tust.2019.103049>.
- B Pulatsu, E M Bretas, and P B Lourenço. Discrete element modeling of masonry structures: Validation and application. *Earthquakes and Structures*, 11(4): 563–582, 2016. doi: 10.12989/eas.2016.11.4.563.
- T Qu, S Guan, Y T Feng, G Ma, W Zhou, and J Zhao. Deep active learning for constitutive modelling of granular materials: From representative volume elements to implicit finite element modelling. *International Journal of Plasticity*, 164:103576, 2023. ISSN 0749-6419. doi: <https://doi.org/10.1016/j.ijplas.2023.103576>. URL <https://www.sciencedirect.com/science/article/pii/S0749641923000621>.
- L S Rainone, V Tateo, S Casolo, and G Uva. About the use of concrete damage plasticity for modeling masonry post-elastic behavior. *Buildings*, 13(8), 2023. ISSN 2075-5309. doi: 10.3390/buildings13081915. URL <https://www.mdpi.com/2075-5309/13/8/1915>.
- W J Rankin. Ground movements resulting from urban tunnelling: predictions and effects. pages 79–92. The Geological Society, 1988.
- Y R Rashid. Analysis of prestressed concrete pressure vessels. *Nuclear Engineering and Design*, 7(4):334–344, 1968.

- S Ritter, M J Jong, G Giardina, and R Mair. The effect of surface structures on soil deformations due to tunnelling in sand. *Rivista Italiana di Geotecnica*, 51, 3 2017. doi: 10.17863/CAM.9342.
- P. Roca, M. Cervera, G. Gariup, and L. Pelà. Structural analysis of masonry historical constructions. classical and advanced approaches. *Archives of Computational Methods in Engineering*, 17(3):299–325, September 2010. doi: 10.1007/s11831-010-9046-1. URL <https://doi.org/10.1007/s11831-010-9046-1>.
- J G Rots. *Computational modeling of concrete fracture*. Dissertation, Delft University of Technology, Delft, The Netherlands, 1988.
- J G Rots and J Blaauwendraad. Crack models for concrete, discrete or smeared? fixed, multi-directional or rotating. *Heron*, 34(1), 1989. URL <http://resolver.tudelft.nl/uuid:0a401939-2988-4f9d-a396-b6a865b22bd9>.
- A Saadallah, A Egorov, B T Cao, S Freitag, K Morik, and G Meschke. Active learning for accurate settlement prediction using numerical simulations in mechanized tunneling. *Procedia CIRP*, 81:1052–1058, 2019. ISSN 2212-8271. doi: <https://doi.org/10.1016/j.procir.2019.03.250>. URL <https://www.sciencedirect.com/science/article/pii/S2212827119305554>. 52nd CIRP Conference on Manufacturing Systems (CMS), Ljubljana, Slovenia, June 12-14, 2019.
- S Saloustros, L Pelà, P Roca, and J Portal. Numerical analysis of structural damage in the church of the poblet monastery. *Engineering Failure Analysis*, 48:41–61, 11 2014. doi: 10.1016/j.engfailanal.2014.10.015.
- A Saltelli, S Tarantola, F Campolongo, and M Ratto. *Sensitivity analysis in practice: a guide to assessing scientific models*. John Wiley & Sons, 2004.
- A Saltelli, M Ratto, T Andres, F Campolongo, J Cariboni, D Gatelli, et al. *Global sensitivity analysis: The primer*. John Wiley & Sons, 2008.
- P Schubertl. Tensile and flexural strength of masonry-influences, test methods, test results, 1994.

- A Selby. Tunnelling in soils – ground movements, and damage to buildings in workington, uk. *Geotechnical and Geological Engineering*, 17:351–371, 1999. doi: 10.1023/A:1008985814841.
- B Settles. *Active Learning*. Synthesis Lectures on Artificial Intelligence and Machine Learning. Springer Cham, 1 edition, 2012. ISBN 978-3-031-00432-2. doi: 10.1007/978-3-031-01560-1.
- N Shakhovska, V Yakovyva, and N Kryvinska. An improved software defect prediction algorithm using self-organizing maps combined with hierarchical clustering and data preprocessing. In S. Hartmann, J. Küng, G. Kotsis, A.M. Tjoa, and I. Khalil, editors, *Database and Expert Systems Applications*, volume 12391 of *Lecture Notes in Computer Science*, pages 414–424. Springer International Publishing, Cham, 2020. doi: 10.1007/978-3-030-59003-1_27.
- S K Shreyas and A Dey. Application of soft computing techniques in tunnelling and underground excavations: state of the art and future prospects. *Innovative Infrastructure Solutions*, 4, 12 2019. ISSN 23644184. doi: 10.1007/s41062-019-0234-z.
- L Silva, A Christoforo, and R Carvalho. Calibration of concrete damaged plasticity model parameters for shear walls. *Matéria (Rio de Janeiro)*, 26, 01 2021. doi: 10.1590/s1517-707620210001.1244.
- H Skinner and J Charles. Settlement and tilt of low-rise buildings. *Proceedings of The Institution of Civil Engineers-geotechnical Engineering - PROC INST CIVIL ENG-GEOTECH E*, 157:65–75, 3 2004. doi: 10.1680/geng.2004.157.2.65.
- S Smith. Building information modelling: moving Crossrail, UK, forward. *Proceedings of the Institution of Civil Engineers - Management, Procurement and Law*, 167:141–151, 2014.
- M Son and E J Cording. Estimation of building damage due to excavation-induced ground movements. *Journal of Geotechnical and Geoenvironmental Engineer-*

- ing*, 131:162–177, 3 2005. ISSN 1090-0241. doi: 10.1061/(asce)1090-0241(2005)131:2(162).
- M Son and E J Cording. Evaluation of building stiffness for building response analysis to excavation-induced ground movements. *Journal of Geotechnical and Geoenvironmental Engineering*, 133:995–1002, 11 2007. doi: 10.1061/ASCE1090-02412007133:8995.
- SupercivilCD. Coefficient of friction. URL <http://www.supercivilcd.com/FRICTION.htm>.
- X Tan. A tool for optimal planning of urban tunnelling based on bim and effective 3d visualization, 2019.
- L Tang and S H Na. Comparison of machine learning methods for ground settlement prediction with different tunneling datasets. *Journal of Rock Mechanics and Geotechnical Engineering*, 13:1274–1289, 12 2021. ISSN 16747755. doi: 10.1016/j.jrmge.2021.08.006.
- I B Teodoru. Beams on elastic foundation the simplified continuum approach. *Buletinul Institutului Politehnic Din Iași. Sectia Constructii, Arhitectura*, 55: 37–45, 2009a. ProQuest Central; SciTech Premium Collection.
- I B Teodoru. Ebbef2p - a computer code for analyzing beams on elastic foundations. *Intersections*, 6:28–44, 01 2009b.
- I B Teodoru and V Mușat. Beam elements on linear variable two-parameter elastic foundation. *Buletinul Institutului Politehnic din Iași*, 54 (58):69–78, 2008.
- S Timoshenko. *Strength of materials - Part I*. D. Van Nostrand Co., Inc., 1957.
- B Valerie and J S Theodor. *Multiple Criteria Decision Analysis: An Integrated Approach*. Springer Science & Business Media, 2002.
- C V G Vallabhan and Y C Das. Modified vlasov model for beams on elastic foundations. *Journal of Geotechnical Engineering*, 117:956–966, 1991. ISSN 0733-9410.

- A Verruijt and J R Booker. Surface settlements due to deformation of a tunnel in an elastic half plane. *Géotechnique*, 46:753–756, 1996. doi: 10.1680/geot.1996.46.4.753. Cited by: 543.
- V Z Vlasov and N N Leont’ev. *Beams, plates and shells on elastic foundations*. Israel Program for Scientific Translations, 1966.
- E Winkler. *Theory of elasticity and strength*. H. Dominicus, 1867.
- J Yacila, G Camata, J Salsavilca, and N Tarque. Pushover analysis of confined masonry walls using a 3d macro-modelling approach. *Engineering Structures*, 201:109731, 2019. ISSN 0141-0296. doi: <https://doi.org/10.1016/j.engstruct.2019.109731>. URL <https://www.sciencedirect.com/science/article/pii/S0141029619314385>.
- W N Yiu, H J Burd, and C M Martin. Finite-element modelling for the assessment of tunnel-induced damage to a masonry building. *Géotechnique*, 67:780–794, 2017. ISSN 17517656. doi: 10.1680/jgeot.sip17.P.249.
- Y Yu, A Franza, B Ghiassi, L Neves, and A Marshall. Tunnelling-induced wall damage: An appraisal of elastoplastic constitutive models for masonry. *Tunnelling and Underground Space Technology*, 156:106240, 01 2025. doi: 10.1016/j.tust.2024.106240.
- J Zhao and M DeJong. Three-dimensional probabilistic assessment of tunneling induced structural damage using monte-carlo method and hybrid finite element model. *Computers and Geotechnics*, 154:105122, 2023. ISSN 0266-352X. doi: 10.1016/j.compgeo.2022.105122.

Appendices

Appendix A

Code Listings and Implementation Details

```
problem = {'num_vars': <param count>, # Integers
'names': [<param names>], # Strings
'bounds': [[<param 1 range>], # Floats, format: [min,max]
..., [<param n range>]],}
```

Listing A.1: Definition of the problem dictionary used in the SA.

```
def parameter_corr_1(ind. param (\minor{e.g.,} F_c)):
    value = np.random.uniform(<low>, <high>) #Random values
    dep. param = value * ind. param
    return dep. param
```

Listing A.2: Example function showing the correlation between an independent and a dependent parameter.

```

def condition_1(vars):
    value = compute_value_1(vars[index_1], ..., vars[index_n]) # Compute
    value from list
    return value < threshold (S_v, max) # Check threshold
def compute_value_1(param_1, ..., param_n):
    const_1 = np.pi / 2
    derived_constant = np.sqrt(const_1)
    value = 1000 * derived_constant * (VL / 100) * (D ** 2) / (4 * i_x)
    return value

```

Listing A.3: Python functions for evaluating conditions (e.g., condition_1 presented) and computing the corresponding value based on input variables.

```

param_samples_LH = latin.sample(problem, N)
i = 0
while i < N:
    parameters = param_samples[i, :] # Extracting values
    if (condition_1(param) and ... and condition_n(param)):
        i += 1
    else:
        new_row = latin.sample(problem, 1)
        dep_par_1 = param_correlation_1(ind_param_n)
        dep_par_1_arr[i] = dep_par_1 # Update arrays
        new_row[0, index_dep_par_1] = dep_par_1
        # ... repeated for each parameter
        param_samples[i, :] = new_param # Update list with new values
np.savetxt('MLH_samples.csv', param_samples, delimiter=',') # Store

```

Listing A.4: Python implementation of MLH sampling with conditional checks and parameter correlation adjustments.

```

from abaqus import *
from abaqusConstants import *
from caeModules import *

```

Listing A.5: Python imports required for running Abaqus CAE scripting and modules.

```

num_rows = [1, 2, 3, 4, 5] # Floors number

```

Listing A.6: Python code snippet defining the number of building floors.

```

col = int(round((L * H * 0) / ((width - 2 * flange) * w_depth)))

```

Listing A.7: Python code calculating the number of required columns based on geometric parameters.

```

s1.rectangle(point1=(-L/2, 0), point2=(L/2, H)) # Foundation
p.BaseSolidExtrude(sketch=s1, depth=thickness)

y1 = L/2 + 2000 # Soil extent
s.rectangle(point1=(y1, 0), point2=(-y1, 3*thickness))
s.Line(point1=(-y1, -thickness/2), point2=(y1, -thickness/2)) # Partition

s.rectangle(point1=(-L/2, 0), point2=(L/2, H)) # Wall

```

Listing A.8: Python code for creating the geometry of the foundation, soil, partition line, and wall in Abaqus.

```

a.translate(instanceList=('Wall', ), vector=(D0, 0.0 , 0.0))
a.translate(instanceList=('Foundation', ), vector=(D0, 0.0 , 0.0))
a.translate(instanceList=('Soil Layer', ), vector=(D0, 0.0 , 0.0))

```

Listing A.9: Python code translating the wall, foundation, and soil layer instances in the Abaqus assembly.

```

comp = np.loadtxt("Path/Compression_" + str(i) + ".txt", float)
comp_d = np.loadtxt("Path/Compression_damage_" + str(i) + ".txt", float)
tens = np.loadtxt("Path/Tension_" + str(i) + ".txt", float)
tens_d = np.loadtxt("Path/Tension_damage_" + str(i) + ".txt", float)

mdb.models['M_' + str(i)].Material(name='CDP')
mdb.models['M_' + str(i)].Material(name='Soil')

```

Listing A.10: Python code for loading material data from text files and creating material definitions in Abaqus for each model iteration.

```

Displacement = np.loadtxt("Path/analytical_fields_"+str(i)+".txt",str) #
    String value
mdb.models['M_'+str(i)].analyticalFields['vertical_
displacements'].setValues(expression= str(Displacement))

```

Listing A.11: Python code for loading displacement data and assigning it to an analytical field in Abaqus.

```

mdb.models['M_'+str(i)].interactionProperties['IntProp-1']
.tangentialBehavior.setValues(formulation=PENALTY, directionality=
ISOTROPIC, table=((friction, ), ), maximumElasticSlip=FRACTION,
fraction=0.005)

```

Listing A.12: Python code for setting the tangential behavior properties of the interaction in Abaqus using a penalty formulation.

```
#!/bin/bash (The first 5 SBATCH lines must be within given limits)
#SBATCH --partition=defq
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=4
#SBATCH --mem=64g
#SBATCH --time=108:00:00
#SBATCH --mail-user=ali.gamra@nottingham.ac.uk
#SBATCH --mail-type=FAIL,TIME_LIMIT

unset SLURM_GTIDS

cd ${SLURM_SUBMIT_DIR}

module load abaqus-uoneasy/2022

for i in $(seq {start} {end});
do
    abq2022 job=Job_$i input=Job_$i.inp cpus=4 scratch=${TMDIR}
    interactive
done
```

Listing A.13: SLURM batch script for submitting multiple Abaqus jobs with resource specifications and email notifications.

```
# Example logic: loop through .odb files and extract nodal displacements
for file in odb_files:
    session.openOdb(name=file)
    odb = session.odbs[file]
    step = odb.steps['Step-1']
    frame = step.frames[-1]
    displacement = frame.fieldOutputs['U']
    strain = frame.fieldOutputs['LE']
    ...
    # Save output
    write_to_txt(displacement, strain)
    generate_png(DAMAGET)
```

Listing A.14: Python script for automating post-processing of ABAQUS output databases, extracting nodal displacements and element strains for each .odb file.

Appendix B

Dynamo Script for the Evaluation of Building Damage due to Tunnelling, Snapshots



Figure B.1: Nodes describing the input parameters (identified from the pink background in Dynamo) for the tunnel, soil and building including the additional geometry-related parameters.

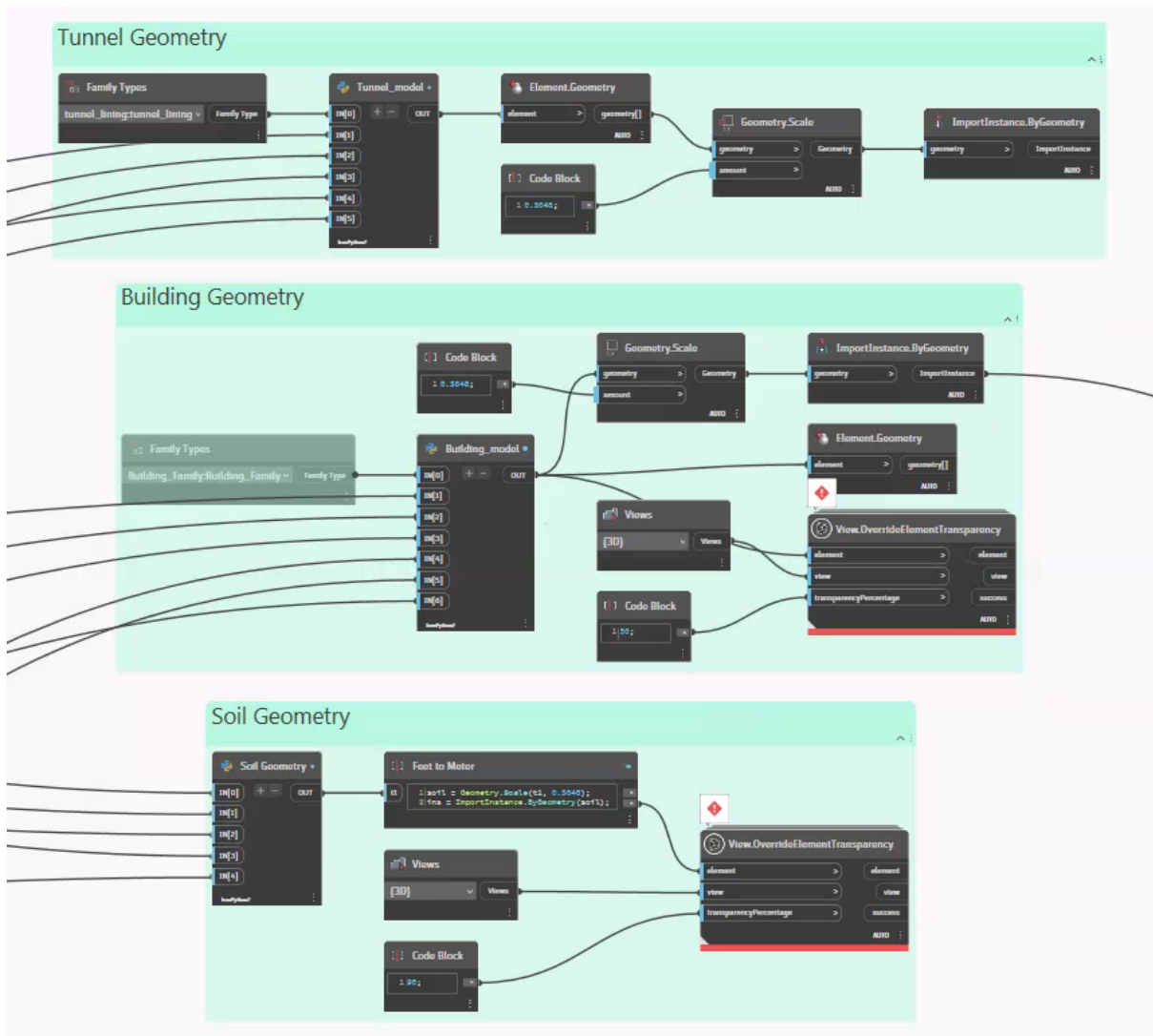


Figure B.2: Nodes describing the actions (identified from the greenish cyan background in Dynamo) for the geometry of the former input elements. Including a view command and a transparency percentage allowing the effective visualisation of tunnel geometry, soil and building (including position).

Appendix C

Source Code for Model Sampling Using the Modified Latin Hypercube Method

```
import os

directory = "F:/Paper - first author/Pre_Process/1.Sampling"
folder_name = "Iteration_5_dataset"

folder_path = os.path.join(directory, folder_name)

if not os.path.exists(folder_path):
    os.makedirs(folder_path)
else:
    print("Folder already exists")

print(f"Folder '{folder_name}' created in '{directory}'")

from SALib.sample import morris, latin
import numpy as np
import csv
```

```

import pandas as pd

problem = {
    'num_vars': 15,
    'names': ['E', 'Ft', 'Gf', 'Height', 'Length', 'OpeningRate',
              'e', 'E_Soil', 'Poissons', 'ix', 'Friction',
              'VL', 'Z', 'D', 'Fc'],
    'bounds': [
        # original dataset
        [666, 9000], # 0 Elastic modulus
        [0.05, 1.47], # 1 Tensile Strength
        [0.004, 0.03], # 2 Fracture Energy in Tension (upper bound selected
to avoid linear behaviour)
        [3.2, 20], # 3 Height of Building (starting at 1-floor height)
        [8, 60], # 4 Length of Building
        [0, 30], # 5 Opening rate
        [0, 45], # 6 Eccentricity (skewed literature available)
        [2, 250], # 7 Soil Stiffness
        [0.1, 0.49], # 8 Soil's Poisson's
        [3.6, 45], # 9 Trough width (poor literature available; distance 0
to 4X avg diameter, i.e., 0 to 45m)
        [0.0001, 0.6], # 10 Friction Coefficient
        [0.2, 5], # 11 Volume Loss
        [7.3, 90], # 12 Tunnel depth (poor literature available, starting at
1.5 min D, i.e., 7.275)
        [4.9, 17.6], # 13 Tunnel diameter
        [0.5, 48.2] # 14 Compressive Strength
    ],
}

# fracture energy selected upper bound because i dont want linear behaviour.
# height started at 3.2 because i want to start rom 1 floor height.
# distance starting from 0 to 4X avg diameter (i.e., 0 to 45m).
# inflection point k = 0.5 between zmin and zmax (i.e., 3.638 and 45).

```

```

# tunnel depth (z) starts at 1.5 min D (i.e., 7.275)

n = 500

param_samples_latin = latin.sample(problem, N=n) #Initial trial
np.savetxt("param_samples_latin.txt", param_samples_latin)

mesh = 175
gfm = 0
h = 0
kp = 0
Fc_arr = np.zeros(n)
elastic_modulus_arr = np.zeros(n)
tensile_fracture_energy_arr = np.zeros(n)
e_arr = np.zeros(n)
trough_width_arr = np.zeros(n)

# =====CORRELATIONS=====

# Correl1
# Define the correlation between elastic_modulus and Ft
def elastic_modulus_correlation(Fc):
    n = np.random.uniform(300, 700) # 500
    elastic_modulus = n * Fc
    return elastic_modulus

# Corre2
# Define the correlation between tensile_fracture_energy and Ft
def tensile_fracture_energy_correlation(Ft):
    n = np.random.uniform(0.005, 0.05) # 0.0275
    tensile_fracture_energy = n * Ft
    return tensile_fracture_energy

# Corre3

```

```

# Define the correlation between e and D
def e_correlation(D):
    n = np.random.uniform(0, 3) # 1.5
    e = n * D
    return e

# Corre4
# Define the correlation between ix and Z
def trough_width_correlation(Z):
    m = np.random.uniform(0.25, 0.63) # 0.44
    trough_width = m * Z
    return trough_width

# =====CONDITIONS=====

# Cond1 (max settlement)
def condition1(vars):
    value = compute_value1(vars[9], vars[11], vars[13])
    return value < 150 # 150 max settlement

def compute_value1(var10, var12, var14):
    pi_half = np.pi/2
    sqrt_pi_half = np.sqrt(pi_half)
    value = 1000 * sqrt_pi_half * (var12 / 100) * (var14 ** 2) / (4 * var10)
    return value

# Cond2(tens. snap-back)
def condition2(vars):
    value2 = compute_value2(vars[0], vars[1], vars[2])
    return value2 > mesh

def compute_value2(var1, var2, var3):
    value2 = var1 * var3 / (var2 ** 2) # all this * 2 if linear decay is
    chosen.

```

```

    return value2

# Cond3(comp. snap-back)
def condition3(vars, gfm, h, kp):
    cj = 0.27 / (15.2 ** 0.25)
    em = cj * vars[14] / (vars[0] ** 0.7)
    kp = em - vars[14] / vars[0]
    gfm = 15 + 0.43 * vars[14] - 0.0036 * vars[14] ** 2
    h = mesh
    value3 = compute_value3(vars[0], vars[14], kp)
    return value3 < 75 / 67 * gfm / (h * vars[14]) + kp

def compute_value3(var1, var15, kp):
    value3 = var15 / var1 + kp
    return value3

# Cond4 (overburden height)
def condition4(vars):
    value4 = compute_value4(vars[12], vars[13])
    return value4

def compute_value4(var13, var14):
    value4 = var13
    return value4 >= 1.5 * var14

# Cond5 (Ft should not be less than E/10000)
def condition5(vars):
    return vars[1] >= vars[0]/10000

def condition_a(vars):
    return (vars[0] >= 666 and vars[0] <= 9000) and \
           (vars[1] >= 0.05 and vars[1] <= 1.47) and \
           (vars[2] >= 0.004 and vars[2] <= 0.03) and \
           (vars[3] >= 3.2 and vars[3] <= 20) and \

```

```

        (vars[4] >= 8 and vars[4] <= 60) and \
        (vars[5] >= 0 and vars[5] <= 30) and \
        (vars[6] >= 0 and vars[6] <= 45) and \
        (vars[7] >= 2 and vars[7] <= 250) and \
        (vars[8] >= 0.1 and vars[8] <= 0.49) and \
        (vars[9] >= 3.6 and vars[9] <= 45) and \
        (vars[10] >= 0.0001 and vars[10] <= 0.6) and \
        (vars[11] >= 0.2 and vars[11] <= 5) and \
        (vars[12] >= 7.3 and vars[12] <= 90) and \
        (vars[13] >= 4.9 and vars[13] <= 17.6) and \
        (vars[14] >= 0.5 and vars[14] <= 48.2)

# =====GENERATING SAMPLES=====

param_samples_mod_latin = latin.sample(problem, n)
np.savetxt('F:/Paper - first
           author/Pre_Process/1.Sampling/'+folder_name+'/Samples.csv',
           param_samples_mod_latin, delimiter=',')

# Apply the correlation between elastic_modulus and Ft
for i in range(n):
    Fc = param_samples_mod_latin[i, 14]
    elastic_modulus = elastic_modulus_correlation(Fc)
    param_samples_mod_latin[i, 0] = elastic_modulus

# Apply the correlation between tensile_fracture_energy and Ft
for i in range(n):
    Ft = param_samples_mod_latin[i, 1]
    tensile_fracture_energy = tensile_fracture_energy_correlation(Ft)
    param_samples_mod_latin[i, 2] = tensile_fracture_energy

# Apply the correlation between e and D
for i in range(n):
    D = param_samples_mod_latin[i, 13]

```



```

    e = e_correlation(D)

    param_samples_mod_latin[i, 6] = e

# Apply the correlation between i and Z
for i in range(n):
    Z = param_samples_mod_latin[i, 12]

    trough_width = trough_width_correlation(Z)

    param_samples_mod_latin[i, 9] = trough_width

# Check if the condition is violated and resample if necessary
i = 0
while i < n:
    vars = param_samples_mod_latin[i, :]

    if (condition1(vars) and condition2(vars) and condition3(vars, gfm, h,
kp) and
        condition4(vars) and condition5(vars) and condition_a(vars)):
        i += 1
    else:
        new_row = latin.sample(problem, 1)

        # Generate new values for e, D, i and Z
        Fc = new_row[0, 14]

        elastic_modulus = elastic_modulus_correlation(Fc)

        Ft = new_row[0, 1]

        tensile_fracture_energy = tensile_fracture_energy_correlation(Ft)

        D = new_row[0, 13]

        e = e_correlation(D)

        Z = new_row[0, 12]

        trough_width = trough_width_correlation(Z)

    # Update the generated values arrays

```

```

        elastic_modulus_arr[i] = elastic_modulus
        tensile_fracture_energy_arr[i] = tensile_fracture_energy
        e_arr[i] = e
        trough_width_arr[i] = trough_width

        # Apply the new values to the i-th row
        new_row[0, 0] = elastic_modulus
        new_row[0, 2] = tensile_fracture_energy
        new_row[0, 6] = e
        new_row[0, 9] = trough_width
        param_samples_mod_latin[i, :] = new_row

np.savetxt('F:/Paper - first
           author/Pre_Process/1.Sampling/'+folder_name+'/Samples.csv',
           param_samples_mod_latin, delimiter=',')

input_file = 'F:/Paper - first
             author/Pre_Process/1.Sampling/'+folder_name+'/Samples.csv'
output_file = 'F:/Paper - first
              author/Pre_Process/1.Sampling/'+folder_name+'/Samples.txt'

with open(input_file, 'r', newline='', encoding='utf-8') as csv_file:
    reader = csv.reader(csv_file)

    with open(output_file, 'w', encoding='utf-8') as txt_file:
        for row in reader:
            line = ' '.join(row) # Customize the delimiter if necessary
            txt_file.write(line + '\n')

print(f"CSV file '{input_file}' has been converted to a TXT file
      '{output_file}'.")

```

Appendix D

Source Code for Model Assembly in ABAQUS Using the ABAQUS Python Development Environment and Model Prototype

```
### -*- coding: mbcs -*-  
  
###  
  
### Abaqus/CAE Release 2019 replay file  
  
### Internal Version: 2018_09_24-19.41.51 157541  
  
### Run by evxag6 on Thu Jul 21 11:24:26 2022  
  
from abaqus import *  
from abaqusConstants import *  
from caeModules import *  
import numpy as np  
import time  
import multiprocessing as mp
```

```

from multiprocessing import Pool

#from part import BoundingBox

session.journalOptions.setValues(replayGeometry=COORDINATE,
    recoverGeometry=COORDINATE)
Mdb()
openMdb(pathName='D:/ABAQUS_Models/2D_latin_models/FEM.cae')
import os
os.chdir(r"D:\ABAQUS_Models\2D_latin_models")

seed = 175

thickness = 250          #WALL THICKNESS
f_height = 500           #FOUNDATION HEIGHT
width = 1200             #FLANGE WIDTH
flange = 130             #FLANGE OFFSET
height = 150             #FLANGE HEIGHT
w_depth = 1500           #WINDOWS DEPTH
d_depth = 2000           #DOOR DEPTH
m_poisson = 0.2          #MASONRY POISSON'S
m_density = 2.1e-09      #MASONRY DENSITY
floor_height = 3200

filename = r"F:/Paper - first
    author/Pre_Process/1.Sampling/Original_dataset/Samples.txt"

e_building = np.loadtxt(filename,float)[: ,0]
tensile_strength = np.loadtxt(filename,float)[: ,1]
tensile_fracture_energy = np.loadtxt(filename,float)[: ,2]
floor_height = np.loadtxt(filename,float)[: ,3]
building_length = np.loadtxt(filename,float)[: ,4]
opening_rate = np.loadtxt(filename,float)[: ,5]
building_offset = np.loadtxt(filename,float)[: ,6]
e_soil = np.loadtxt(filename,float)[: ,7]

```

```

soil_poissons = np.loadtxt(filename,float)[: ,8]
trough_width = np.loadtxt(filename,float)[: ,9]
friction_coefficient = np.loadtxt(filename,float)[: ,10]
volume_loss = np.loadtxt(filename,float)[: ,11]
tunnel_depth = np.loadtxt(filename,float)[: ,12]
tunnel_diameter = np.loadtxt(filename,float)[: ,13]
compressive_strength = np.loadtxt(filename,float)[: ,14]

k1 = 3 # door frequency
k2 = 2 # door initial positioning
D0 = 0
D1 = 0

for iii in range(30,35):
    e_b = e_building[iii] # as is
    ft = tensile_strength[iii] # as is
    gft = tensile_fracture_energy[iii] # as is
    f_height = 1000 * floor_height[iii]
    b_length = 1000 * round(building_length[iii] / 0.175) * 0.175
    op_rate = 0.01 * round(opening_rate[iii] / 5.5) * 5.5
    b_offset = 1000 * building_offset[iii]
    e_s = e_soil[iii] # as is
    s_poissons = soil_poissons[iii] # as is
    t_width = 1000 * trough_width[iii] # as is
    v_loss = 0.01 * volume_loss[iii] # as is
    t_depth = 1000 * tunnel_depth[iii] # as is
    t_diameter = 1000 * tunnel_diameter[iii] # as is
    fc = compressive_strength[iii] # as is
    friction = friction_coefficient[iii] # as is

    #columns and rows for the openings
    #col = int((b_length * floor_height * op_rate) / ((width - 2 * flange) *
    w_depth))

```

```

#col = int(round((b_length * floor_height * op_rate) / ((width - 2 *
flange) * w_depth)))

# Define the height ranges and the number of rows for each range
height_ranges = [(0, 5600), (5600, 8400), (8400, 12600), (11200, 16800),
(14000, 21000)]
num_rows = [1, 2, 3, 4, 5]

# Calculate the row based on the height of the floor
for i, (min_height, max_height) in enumerate(height_ranges):
    if min_height <= f_height < max_height:
        row = num_rows[i]
        break

## columns and rows for the openings
##col = int((b_length * floor_height * op_rate) / ((width - 2 * flange)
* w_depth))
col = int(round((b_length * floor_height * op_rate) / ((width - 2 *
flange) * w_depth)))

#else:
    # If the floor height is outside of all defined ranges, use a
default row value of 1
    #row = f_height / floor_height

#row = f_height / floor_height

# facade domain and spacing
h_spacing_f = (b_length - col * width) / (col + 1)    #MIN. SPACING --
(i.e 100 -col* i.e 20)/(col+1) = i.e 5 to find mnimum number of columns
left = h_spacing_f + width - b_length / 2
right = b_length / 2
v_spacing_f = (f_height - row * (height + w_depth)) / (row + 1)
down= v_spacing_f + 0.5 * height + w_depth

```

```

up = f_height

# foundation geometry
s1 = mdb.models['M'+ '_' +str(iii)].ConstrainedSketch(name='__profile__',
sheetSize=2000.0)

g, v, d, c = s1.geometry, s1.vertices, s1.dimensions, s1.constraints
s1.setPrimaryObject(option=STANDALONE)
s1.rectangle(point1=(-b_length/2, 0), point2=(b_length/2, f_height))
p = mdb.models['M'+ '_' +str(iii)].Part(name='Part-2',
dimensionality=THREE_D,
    type=DEFORMABLE_BODY)
p.BaseSolidExtrude(sketch=s1, depth=thickness)
s1.unsetPrimaryObject()
session.viewports['Viewport: 1'].setValues(displayedObject=p)
del mdb.models['M'+ '_' +str(iii)].sketches['__profile__']

# soil layer geometry
s = mdb.models['M'+ '_' +str(iii)].ConstrainedSketch(name='__profile__',
sheetSize=2000.0)

g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.setPrimaryObject(option=STANDALONE)
y1 = b_length/2 + 2000
s.rectangle(point1=(y1, 0), point2=(-y1, 3*thickness))
p = mdb.models['M'+ '_' +str(iii)].Part(name='Part-3',
dimensionality=THREE_D,
    type=DEFORMABLE_BODY)
p.BaseShell(sketch=s)
s.unsetPrimaryObject()
p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']
session.viewports['Viewport: 1'].setValues(displayedObject=p)
del mdb.models['M'+ '_' +str(iii)].sketches['__profile__']

# partitioning soil layer
p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']

```

```

p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']
f, e, d = p.faces, p.edges, p.datums
t = p.MakeSketchTransform(sketchPlane=f.findAt(coordinates=(-b_length /
4, f_height / 2, 0.0), normal=(0.0, 0.0, 1.0)),
sketchUpEdge=e.findAt(coordinates=(
    y1, 1.5 * thickness / 2, 0.0)), sketchPlaneSide=SIDE1, origin=(0.0,
1.5 * thickness, 0.0))
s = mdb.models['M'+ '_' +str(iii)].ConstrainedSketch(name='__profile__',
sheetSize=88012.78,
    gridSpacing=2200.31, transform=t)
g, v, d1, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.setPrimaryObject(option=SUPERIMPOSE)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']
p.projectReferencesOntoSketch(sketch=s, filter=COPLANAR_EDGES)
s.Line(point1=(-y1, -thickness / 2), point2=(y1, -thickness / 2))
s.HorizontalConstraint(entity=g.findAt((0.0, -thickness / 2)),
addUndoState=False)
s.Line(point1=(-y1, thickness / 2), point2=(y1, thickness / 2))
s.HorizontalConstraint(entity=g.findAt((0.0, thickness / 2)),
addUndoState=False)
s.Line(point1=(-y1 + 2000, -thickness / 2), point2=(-y1 + 2000,
thickness / 2))
s.VerticalConstraint(entity=g.findAt((-y1 + 2000, 0.0)),
addUndoState=False)
s.Line(point1=(y1 - 2000, -thickness / 2), point2=(y1 - 2000, thickness
/ 2))
s.VerticalConstraint(entity=g.findAt((y1 - 2000, 0.0)),
addUndoState=False)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']
f = p.faces
pickedFaces = f.findAt(((b_length / 4, f_height / 2, 0.0), ))
e1, d2 = p.edges, p.datums
p.PartitionFaceBySketch(sketchUpEdge=e1.findAt(coordinates=(y1, 1.5 *
thickness / 2,

```



```

0.0)), faces=pickedFaces, sketch=s)

s.unsetPrimaryObject()

del mdb.models['M'+ '_' +str(iii)].sketches['__profile__']

# walls and lintels (had to be last cause next code block operates with
this command)

s = mdb.models['M'+ '_' +str(iii)].ConstrainedSketch(name='__profile__',
    sheetSize=2000.0)

g, v, d, c = s.geometry, s.vertices, s.dimensions, s.constraints
s.setPrimaryObject(option=STANDALONE)
s.rectangle(point1=(-b_length / 2, 0), point2=(b_length / 2, f_height))
p = mdb.models['M'+ '_' +str(iii)].Part(name='Part-1',
dimensionality=THREE_D,
    type=DEFORMABLE_BODY)
p.BaseShell(sketch=s)
s.unsetPrimaryObject()

p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
del mdb.models['M'+ '_' +str(iii)].sketches['__profile__']

f, e, d1 = p.faces, p.edges, p.datums
t = p.MakeSketchTransform(sketchPlane=f[0], sketchUpEdge=e[2],
    sketchPlaneSide=SIDE1, sketchOrientation=TOP, origin=(0,0, 0.0))
s1 = mdb.models['M'+ '_' +str(iii)].ConstrainedSketch(name='__profile__',
    sheetSize=2828.42, gridSpacing=70.71, transform=t)
g, v, d, c = s1.geometry, s1.vertices, s1.dimensions, s1.constraints
s1.setPrimaryObject(option=SUPERIMPOSE)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
p.projectReferencesOntoSketch(sketch=s1, filter=COPLANAR_EDGES)

# initializing assembly stage
if iii == 1:
    a = mdb.models['M'+ '_' +str(iii)].rootAssembly
    a.translate(instanceList=('Part-1-1', ), vector=(D0, 0.0 , 0.0))
    a.translate(instanceList=('Part-2-1', ), vector=(D0, 0.0 , 0.0))
    a.translate(instanceList=('Part-3-1', ), vector=(D0, 0.0 , 0.0))

```

```

else:
    a = mdb.models['M'+ '_' +str(iii)].rootAssembly
    a.translate(instanceList=('Part-1-1', ), vector=(D1-D0, 0.0 , 0.0))
    a.translate(instanceList=('Part-2-1', ), vector=(D1-D0, 0.0 , 0.0))
    a.translate(instanceList=('Part-3-1', ), vector=(D1-D0, 0.0 , 0.0))

# lintels by partition
if op_rate != 0:
    for i in np.arange(left, right, h_spacing_f + width):
        for j in np.arange(down, up, v_spacing_f + w_depth + height):
            for a, b, c, d, e, f in [(i, j, j + height, i - width, i -
width + flange, i - flange)]:
                s1.Line(point1=(e, b), point2=(d, b))
                s1.HorizontalConstraint(entity=g[6], addUndoState=False)

                s1.Line(point1=(d, b), point2=(d, c))
                s1.VerticalConstraint(entity=g[7], addUndoState=False)
                s1.PerpendicularConstraint(entity1=g[6], entity2=g[7],
addUndoState=False)

                s1.Line(point1=(d, c), point2=(a, c))
                s1.HorizontalConstraint(entity=g[8], addUndoState=False)
                s1.PerpendicularConstraint(entity1=g[7], entity2=g[8],
addUndoState=False)

                s1.Line(point1=(a, c), point2=(a, b))
                s1.VerticalConstraint(entity=g[9], addUndoState=False)
                s1.PerpendicularConstraint(entity1=g[8], entity2=g[9],
addUndoState=False)

                s1.Line(point1=(a, b), point2=(f, b))
                s1.VerticalConstraint(entity=g[5], addUndoState=False)
                s1.PerpendicularConstraint(entity1=g[4], entity2=g[5],
addUndoState=False)

```

```

p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
f = p.faces
pickedFaces = f.getSequenceFromMask(mask=('[#1 ]', ), )
e1, d2 = p.edges, p.datums
p.PartitionFaceBySketch(sketchUpEdge=e1[2], faces=pickedFaces,
    sketchOrientation=TOP, sketch=s1)
s1.unsetPrimaryObject()

# windows and doors opening
del mdb.models['M'+ '_' +str(iii)].sketches['__profile__']
count2 = k1      #space between doors
place = k2      #starting point 0 is the first 1& 2 is the second
place fo a count of 3
for i in np.arange(left, right, h_spacing_f + width):
    if place % count2 == 0:
        for j in np.arange(down, up, v_spacing_f + w_depth + height):
            if j % down == 0:
                for e, b, f, g in [(i - width + flange, j, i -
flange, j - d_depth)]:
                    e1 = float(e)
                    b1 = float(b)
                    f1 = float(f)
                    g1 = float(g)
                    p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
                    s = p.features['Shell planar-1'].sketch

mdb.models['M'+ '_' +str(iii)].ConstrainedSketch(name='__edit__',
    objectToCopy=s)

s2 =
mdb.models['M'+ '_' +str(iii)].sketches['__edit__']
g, v, d, c = s2.geometry, s2.vertices,
s2.dimensions, s2.constraints
s2.setPrimaryObject(option=SUPERIMPOSE)

```

```

        p.projectReferencesOntoSketch(sketch=s2,
                                     upToFeature=p.features['Shell planar-1'],
filter=COPLANAR_EDGES)

        s2.rectangle(point1=(e1, b1), point2=(f1, g1))
        s2.unsetPrimaryObject()

        p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
        p.features['Shell planar-1'].setValues(sketch=s2)
        del

mdb.models['M'+ '_' +str(iii)].sketches['__edit__']

        p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
        p.regenerate()

    else:
        for e, b, f, g in [(i - width + flange, j, i -
flange, j - w_depth)]:

            e1 = float(e)
            b1 = float(b)
            f1 = float(f)
            g1 = float(g)

            p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
            s = p.features['Shell planar-1'].sketch

mdb.models['M'+ '_' +str(iii)].ConstrainedSketch(name='__edit__',
objectToCopy=s)

            s2 =

mdb.models['M'+ '_' +str(iii)].sketches['__edit__']

            g, v, d, c = s2.geometry, s2.vertices,
s2.dimensions, s2.constraints

            s2.setPrimaryObject(option=SUPERIMPOSE)
            p.projectReferencesOntoSketch(sketch=s2,
                                     upToFeature=p.features['Shell planar-1'],
filter=COPLANAR_EDGES)

            s2.rectangle(point1=(e1, b1), point2=(f1, g1))
            s2.unsetPrimaryObject()

            p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']

```

```

        p.features['Shell planar-1'].setValues(sketch=s2)
        del
mdb.models['M'+ '_' +str(iii)].sketches['__edit__']
        p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
        p.regenerate()

    else:
        for j in np.arange(down, up, v_spacing_f + w_depth + height):
            for e, b, f, g in [(i - width + flange, j, i -
flange, j - w_depth)]:
                e1 = float(e)
                b1 = float(b)
                f1 = float(f)
                g1 = float(g)
                p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
                s = p.features['Shell planar-1'].sketch

mdb.models['M'+ '_' +str(iii)].ConstrainedSketch(name='__edit__',
objectToCopy=s)

                s2 =
mdb.models['M'+ '_' +str(iii)].sketches['__edit__']
                g, v, d, c = s2.geometry, s2.vertices,
s2.dimensions, s2.constraints
                s2.setPrimaryObject(option=SUPERIMPOSE)
                p.projectReferencesOntoSketch(sketch=s2,
                    upToFeature=p.features['Shell planar-1'],
filter=COPLANAR_EDGES)
                s2.rectangle(point1=(e1, b1), point2=(f1, g1))
                s2.unsetPrimaryObject()
                p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
                p.features['Shell planar-1'].setValues(sketch=s2)
                del
mdb.models['M'+ '_' +str(iii)].sketches['__edit__']
                p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
                p.regenerate()

```

```

        place += 1

#=====ASSIGING MATERIAL PROPERTIES=====

    # ASSIGING MATERIAL PROPERTIES USING THE FC&FT MATERIAL PROCESSING CODE
    (PYTHON)

    comp = np.loadtxt("F:/Paper - first
author/Pre_Process/3.Material_CDP/Original_dataset/Compression_"+str(iii)+
".txt",float)    #interchangable

    comp_d = np.loadtxt("F:/Paper - first
author/Pre_Process/3.Material_CDP/Original_dataset/Compression_damage_"
+str(iii)+".txt",float)

    tens = np.loadtxt("F:/Paper - first
author/Pre_Process/3.Material_CDP/Original_dataset/Tension_exp_"
+str(iii)+".txt",float)

    tens_d = np.loadtxt("F:/Paper - first
author/Pre_Process/3.Material_CDP/Original_dataset/Tension_damage_exp_"
+str(iii)+".txt",float)

#=====ASSIGING MATERIAL PROPERTIES=====

    mdb.models['M'+ '_' +str(iii)].Material(name='Material')

    mdb.models['M'+ '_' +str(iii)].materials['Material'].Density(table=((m_density,
), ))

    mdb.models['M'+ '_' +str(iii)].materials['Material'].Elastic(table=((e_b,
m_poisson), ))

    mdb.models['M'+ '_' +str(iii)].materials['Material'].
ConcreteDamagedPlasticity(table=((
40.0, 0.1, 1.16, 0.67, 0.0001), ))

    mdb.models['M'+ '_' +str(iii)].materials['Material'].
concreteDamagedPlasticity.ConcreteCompressionHardening(
table=(comp))

    mdb.models['M'+ '_' +str(iii)].materials['Material'].
concreteDamagedPlasticity.ConcreteTensionStiffening(
table=(tens), type=DISPLACEMENT)

```

```

mdb.models['M'+ '_' +str(iii)].materials['Material'].
concreteDamagedPlasticity.ConcreteCompressionDamage(
    table=(comp_d))
mdb.models['M'+ '_' +str(iii)].materials['Material'].
concreteDamagedPlasticity.ConcreteTensionDamage(
    table=(tens_d), type=DISPLACEMENT)

# soil elastic properties
mdb.models['M'+ '_' +str(iii)].Material(name='Soil')
mdb.models['M'+ '_' +str(iii)].materials['Soil'].Elastic(table=((e_s,
s_poissons), ))

# masonry wall section
mdb.models['M'+ '_' +str(iii)].HomogeneousShellSection(name='masonry_sec',
preIntegrate=OFF,
material='Material', thicknessType=UNIFORM, thickness=thickness,
thicknessField='', nodalThicknessField='', idealization=NO_IDEALIZATION,
poissonDefinition=DEFAULT, thicknessModulus=None, temperature=GRADIENT,
useDensity=OFF, integrationRule=SIMPSON, numIntPts=5)

mdb.models['M'+ '_' +str(iii)].HomogeneousSolidSection(name='foundation_sec',
    material='Material', thickness=None)

#=====ASSIGING SETTLEMENT PROFILE=====

# ASSIGING SETTLEMENT PROFILE USING THE EBBEF2p PROCESSING CODE (PYTHON)
# applying the analytical fields of applied settlements.
apllied_vertical_displacements = np.loadtxt("F:/Paper - first
author/Pre_Process/2.EBBEF2p/Original_dataset/analytical_fields_"+str(iii)
+".txt",str)
mdb.models['M'+ '_' +str(iii)].analyticalFields['vertical_displacements'].
setValues(
    expression= str(apllied_vertical_displacements))

#=====ASSIGING SETTLEMENT PROFILE=====

```

```

# applying different friction coefficients
mdb.models['M'+ '_' +str(iii)].interactionProperties['IntProp-1'].
tangentialBehavior.setValues(
    formulation=PENALTY, directionality=ISOTROPIC,
slipRateDependency=OFF,
    pressureDependency=OFF, temperatureDependency=OFF, dependencies=0,
table=((
    friction, ), ), shearStressLimit=None, maximumElasticSlip=FRACTION,
    fraction=0.005, elasticSlipStiffness=None)

# applying boundary conditions.
a = mdb.models['M'+ '_' +str(iii)].rootAssembly
e1 = a.instances['Part-3-1'].edges
edges1 = e1.findAt(((0.0, 0.0, 3 * thickness / 2), ), ((0.0, 0.0, -3 *
thickness / 2), ))
f2 = a.instances['Part-1-1'].faces
faces2 = f2.findAt(((0.0, f_height + 1, 0.0), ))
region = regionToolset.Region(edges=edges1, faces=faces2)
mdb.models['M'+ '_' +str(iii)].boundaryConditions['z-constraint'].
setValues(region=region)

a = mdb.models['M'+ '_' +str(iii)].rootAssembly
region = a.sets['bottom']
mdb.models['M'+ '_' +str(iii)].DisplacementBC(name='applied_settlements',
    createStepName='Step-2', region=region, u1=UNSET, u2=1.0, u3=UNSET,
    ur1=UNSET, ur2=UNSET, ur3=UNSET, amplitude=UNSET, fixed=OFF,
    distributionType=FIELD, fieldName='vertical_displacements',
localCsys=None)
a = mdb.models['M'+ '_' +str(iii)].rootAssembly
region = a.sets['bottom']
mdb.models['M'+ '_' +str(iii)].EncastreBC(name='initial_constraints',
    createStepName='Initial', region=region, localCsys=None)
mdb.models['M'+ '_' +str(iii)].boundaryConditions['initial_constraints'].
deactivate(

```



```

'Step-2')

# assigning sections walls & lintels
if op_rate != 0:
    for i in np.arange(left, right, h_spacing_f + width):
        for j in np.arange(down, up, v_spacing_f + w_depth + height):
            for a, b, c, d, e, f in [(i, j, j + height, i - width, i -
width + flange, i - flange)]:
                p = mdb.models['M'+str(iii)].parts['Part-1']
                f = p.faces
                faces = f.findAt(((a - width + 1, b + 1, 0.0), ))
                region = p.Set(faces=faces,
name='Lintels'+str(int(a))+str(int(b)))
                p = mdb.models['M'+str(iii)].parts['Part-1']
                p.SectionAssignment(region=region,
sectionName='Lintel_section', offset=0.0,
                                offsetType=MIDDLE_SURFACE, offsetField='',
                                thicknessAssignment=FROM_SECTION)

            # regenerating the mesh
            elemType1 = mesh.ElemType(elemCode=S8R,
elemLibrary=STANDARD)
            elemType2 = mesh.ElemType(elemCode=STRI65,
elemLibrary=STANDARD)
            p = mdb.models['M'+str(iii)].parts['Part-1']
            pickedRegions =(faces, )
            p.setElementType(regions=pickedRegions,
elemTypes=(elemType1, elemType2))
            p = mdb.models['M'+str(iii)].parts['Part-1']
            p.seedPart(size=seed, deviationFactor=0.1,
minSizeFactor=0.1)
            p = mdb.models['M'+str(iii)].parts['Part-1']
            p.generateMesh()

```

```

p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
f = p.faces
faces = f.findAt((((b_length / 2) + 1, 1, 0.0), ))
region = p.Set(faces=faces, name='Wall')
p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
p.SectionAssignment(region=region, sectionName='masonry_sec', offset=0.0,
                    offsetType=MIDDLE_SURFACE, offsetField='',
                    thicknessAssignment=FROM_SECTION)

# regenerating the mesh
elemType1 = mesh.ElemType(elemCode=S8R, elemLibrary=STANDARD)
elemType2 = mesh.ElemType(elemCode=STRI65, elemLibrary=STANDARD)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
pickedRegions =(faces, )
p.setElementType(regions=pickedRegions, elemTypes=(elemType1, elemType2))
p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
p.seedPart(size=seed, deviationFactor=0.1, minSizeFactor=0.1)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-1']
p.generateMesh()

# assigning sections foundation
p = mdb.models['M'+ '_' +str(iii)].parts['Part-2']
c = p.cells
cells = c.findAt(((b_length / 2, f_height / 2, thickness / 2), ),
                ((b_length / 2, f_height / 2, thickness / 2), ))
region = regionToolset.Region(cells=cells)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-2']
p.SectionAssignment(region=region, sectionName='foundation_sec',
                    offset=0.0,
                    offsetType=MIDDLE_SURFACE, offsetField='',
                    thicknessAssignment=FROM_SECTION)

# regenerating the mesh
p = mdb.models['M'+ '_' +str(iii)].parts['Part-2']

```

```

elemType1 = mesh.ElemType(elemCode=C3D20R, elemLibrary=STANDARD)
elemType2 = mesh.ElemType(elemCode=C3D15, elemLibrary=STANDARD)
elemType3 = mesh.ElemType(elemCode=C3D10, elemLibrary=STANDARD)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-2']
c = p.cells
cells = c.findAt(((b_length / 2, f_height / 2, thickness / 2), ))
pickedRegions =(cells, )
p.setElementType(regions=pickedRegions, elemTypes=(elemType1, elemType2,
    elemType3))
p = mdb.models['M'+ '_' +str(iii)].parts['Part-2']
p.seedPart(size=seed, deviationFactor=0.1, minSizeFactor=0.1)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-2']
p.generateMesh()

# assigning sections soil layer
p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']
f = p.faces
faces = f.findAt(((y1 - 1, 583.333333, 0.0), ), ((y1 - 1,
    416.666667, 0.0), ), ((y1 - 1, 166.666667, 0.0), ), ((-y1 + 1,
    333.333333, 0.0), ), ((0, 333.333333, 0.0), ))
region=regionToolset.Region(faces=faces)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']
p.SectionAssignment(region=region, sectionName='Soil_section',
offset=0.0,
    offsetType=MIDDLE_SURFACE, offsetField='',
    thicknessAssignment=FROM_SECTION)

# regenerating the mesh
p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']
elemType1 = mesh.ElemType(elemCode=S8R, elemLibrary=STANDARD)
elemType2 = mesh.ElemType(elemCode=STRI65, elemLibrary=STANDARD)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']
f = p.faces
#faces = f.findAt((( -1.05 * b_length / 2, thickness, 0.0), ))

```

```

faces = f.findAt((((b_length / 2) + 1999, 583, 0.0), ), (((b_length / 2)
+ 1999,
416, 0.0), ), (((b_length / 2) + 1999, 166, 0.0), ), ((-(b_length / 2) -
1999,
333, 0.0), ), ((0.0, 333, 0.0), ))
pickedRegions =(faces, )
p.setElementType(regions=pickedRegions, elemTypes=(elemType1, elemType2))
p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']
p.seedPart(size=seed, deviationFactor=0.1, minSizeFactor=0.1)
p = mdb.models['M'+ '_' +str(iii)].parts['Part-3']
p.generateMesh()

# creating a set using boundingbox - modifying node numbers.
nodes =
mdb.models['M'+ '_' +str(iii)].parts['Part-2'].nodes.getByBoundingBox(-((b_length
/ 2) + 0.1 * (b_length / 2)), 0.0 - 10, thickness - 10, ((b_length / 2)
+ 0.1 * (b_length / 2)), 0.0 + 10, thickness + 10)
mdb.models['M'+ '_' +str(iii)].parts['Part-2'].Set(nodes=nodes,
name='seedsnodes')
nodes_list = str(list(nodes))
list_of_chars = ['+', '[', ']', ',', ' ', '']
for character in list_of_chars:
    nodes_list = nodes_list.replace(character, '')
result =
nodes_list.split("mdb.models'M_'+str(iii)+''.parts'Part-2'.nodes",
1000)[1::]
"mdb.models'M_30'.parts'Part-2'.nodes"
xt = [int(xt) for xt in result]
final_lists = []
for i in range(int((b_length / seed) + 1)):
    final_lists.append(xt[i] + 1)

# creating its history output.

```

```

regionDef=mdb.models['M'+ '_' +str(iii)].rootAssembly.allInstances['Part-2-1'].
sets['seedsnodes']

mdb.models['M'+ '_' +str(iii)].historyOutputRequests['H-Output-2'].setValues(
    region=regionDef)

# creating new models.
mdb.Model(name='M'+ '_' +str(iii+1), objectToCopy=mdb.models['M_30'])
#del mdb.models['M'+ '_' +str(iii)]

# writing job and submission.
mdb.Job(name='Job'+ '_' +str(iii), model='M'+ '_' +str(iii), description='',
type=ANALYSIS, atTime=None,
    waitMinutes=0, waitHours=0, queue=None, memory=90,
memoryUnits=PERCENTAGE,
    getMemoryFromAnalysis=True, explicitPrecision=SINGLE,
    nodalOutputPrecision=SINGLE, echoPrint=OFF, modelPrint=OFF,
    contactPrint=OFF, historyPrint=OFF, userSubroutine='', scratch='',
    resultsFormat=ODB, multiprocessingMode=DEFAULT, numCpus=6,
numDomains=6,
    numGPUs=0)
mdb.jobs['Job'+ '_' +str(iii)].writeInput(consistencyChecking=OFF)
#   mdb.jobs['JobM'+ '_' +str(iii)].submit(consistencyChecking=OFF)
#   mdb.jobs['JobM'+ '_' +str(iii)].waitForCompletion()

    iii += 1
#   return

```