

Development of a Rapid Screening Approach to  
Estimate the Seismic Capacity of Typical  
Buildings in Groningen

Rithu Maria

*(Page intentionally left blank)*

# Development of a Rapid Screening Approach to Estimate the Seismic Capacity of Typical Buildings in Groningen

by

Rithu Maria

*In partial fulfilment of the requirements for the degree of  
Master of Science in Structural Engineering at the  
Delft University of Technology*



*In collaboration with Royal HaskoningDHV*

An electronic version of this thesis is available at <http://repository.tudelft.nl/>.

*(Page intentionally left blank)*

# Graduation Committee

## **Chair and First Supervisor**

**Dr. F. Messali**

Department of Materials-Mechanics-Management & Design  
Faculty of Civil Engineering and Geo-Sciences  
Delft University of Technology

## **Second Supervisor**

**Dr. ir. Iuri B. C. M. Rocha**

Department of Materials-Mechanics-Management & Design  
Faculty of Civil Engineering and Geo-Science  
Delft University of Technology

## **External**

**Dr. ir. A. Tsouvalas**

Department of Hydraulic Engineering  
Faculty of Civil Engineering and Geo-Science  
Delft University of Technology

## **Company Supervisor**

**Dr. ir. Sander Meijers**

Lead Engineer | Industry & Buildings  
Royal HaskoningDHV

*(Page intentionally left blank)*

# Acknowledgement

This master's thesis represents the culmination of my studies in the Structural Engineering program at TU Delft. My journey has been challenging yet rewarding, and one which has helped me to grow personally and professionally. Throughout this journey, I have been fortunate to engage with a variety of professionals, academics, and practitioners whose insights have greatly enriched my understanding of the subject.

I would like to express my deepest gratitude to my thesis committee, whose guidance, expertise, and encouragement have been invaluable in shaping this work. In particular, I would like to thank a few remarkable individuals who have been a major part of my thesis journey: Dr. F. Messali, Dr. ir. Iuri B. C. M. Rocha and Dr. ir. Sander Meijers, for their continuous supervision, providing me with their valuable insights and feedback at every stage of my research. Our regular discussions and shared insights enriched the quality of this research. My sincere thanks to Dr. ir. A. Tsouvalas for his invaluable guidance and feedback, which was key to shaping this research. Additionally, I would like to thank Dr. ir. Sander Meijers once again for your help in collaborating with the Royal HaskoningDHV, and extend my gratitude to the VIIA team for their essential contributions and support, which were crucial to the success of this research.

A special mention goes to my sister, Riya Maria, who has been by my side every day during this journey. My parents (Dr. Cheriyaan EP and Laiza PP) and my brother (Dr. Roy Cheriyaan) for their constant love and support from the distance throughout my master's education. Lastly, I want to thank my friends, who have been a constant source of support throughout my master's journey.

*Rithu Maria*  
*Delft, 9<sup>th</sup> September 2024*

# Abstract

Gas extraction in the north-eastern Groningen province of the Netherlands has resulted in an increase in induced earthquakes, causing significant damage to homes and affecting residents' mental health. While reinforcement measures and compensations are being implemented, progress has been slow. The structural analysis for reinforcing these buildings follows various methodologies in accordance with Dutch standard NPR 9998, but these methods are time-consuming and repetitive.

This study investigates the use of machine learning techniques to optimize and accelerate the structural analysis of unreinforced masonry (URM) residential buildings. The traditional seismic analysis of URM buildings is computationally expensive, requiring extensive time for model development and assessment. To address this, a classification model and a surrogate model were developed using Deep Neural Network. The data required for both models are obtained from the existing database of Royal HaskoningDHV. Both models provided results indicating whether the building requires reinforcement measures or not. The classification model employs binary classification, using specific building features to categorize structures into those requiring reinforcement and those that do not. The model generates rapid, implementable outcomes, enabling efficient categorization of buildings. This swift assessment allows for prompt decision-making, particularly in identifying and prioritizing high-risk structures that need more detailed analysis. Surrogate models are efficient approximations of complex analysis, capable of predicting reliable structural responses. Here surrogate for SLaMA analysis is built. The SLaMA surrogate will serve as a proof of concept for developing surrogates of more complex and computationally demanding structural analyses.

The results show that, for the classification model, despite sufficient data points, the correlation between input features and the target variable remains weak, primarily due to changes in NPR regulations over the years. This did not allow for a consistent assessment procedure; hence, the same building might be compliant with a recent NPR and not with the older NPRs. For the surrogate model, two different input combinations were experimented with, the analysis provides insights into the impact of input parameters, the need for dimensionality reduction, and the number of data points required for computationally expensive models. The input combinations helped to analyze the curse of dimensionality. The results showed that as more parameters were added, the model required significantly more data to maintain performance. Additionally, the relevance of the selected reduced input features was observed, to see how well it captures the key aspects of the original analysis with less computational cost. It was observed that the selected features need to remain closely aligned with the original analysis to produce comparable results. In conclusion, while machine learning approaches offer significant potential for improving the efficiency of seismic analysis, further refinement and validation of the models are necessary to address their current limitations and enhance their applicability in real-world scenarios.

# Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
1.1	Research Context . . . . .	2
1.2	Research Objectives and Questions . . . . .	4
1.3	Research Scope . . . . .	4
1.4	Research Method . . . . .	4
<b>2</b>	<b>Literature Review</b>	<b>6</b>
2.1	Unreinforcement Masonry Buildings . . . . .	6
2.1.1	Mechanical Properties of Masonry . . . . .	6
2.1.2	Wall Material . . . . .	7
2.1.3	Building box behavior . . . . .	8
2.1.4	Piers and Spandrels . . . . .	8
2.2	Failure Mechanisms of URM walls . . . . .	9
2.2.1	In Plane Failure . . . . .	9
2.2.2	Out of Plane Failure . . . . .	10
2.3	Seismic Vulnerability Database by VIIA . . . . .	11
2.4	Simplified Lateral Mechanism Analysis (SLaMA) . . . . .	13
2.5	Application of the SLaMA-URM method . . . . .	15
2.5.1	Details on Overburden load and Wall Material . . . . .	15
2.5.2	Effect of flange on pier . . . . .	16
2.5.3	Pier capacity . . . . .	17
2.5.4	Yield Displacement of the Pier . . . . .	20
2.5.5	Calculation of Force vs Displacement of Pier capacity . . . . .	21
2.5.6	Transition from Pier Capacity to Wall Capacity or Building Capacity . . . . .	21
2.5.7	Converting to Sa-Sd Bilinear Curve . . . . .	22
2.6	Machine Learning Modelling Techniques . . . . .	23
2.6.1	Classification Model . . . . .	23
2.6.2	Surrogate Model . . . . .	24
2.6.3	Deep Neural Network . . . . .	25
<b>3</b>	<b>Surrogate Model for SLaMA Analysis</b>	<b>30</b>
3.1	Determination of input parameter for SLaMA analysis . . . . .	30
3.2	Development of Surrogate Model . . . . .	31
3.2.1	Data Augmentation for Pier Capacity . . . . .	31
3.2.2	Machine Learning Approach . . . . .	32
3.3	Results and Discussion . . . . .	38
3.3.1	Timber floor structures . . . . .	38
3.3.2	Concrete floor structures . . . . .	45
<b>4</b>	<b>Classification Model</b>	<b>53</b>
4.1	Data cleansing and Preprocessing . . . . .	53
4.2	Model Architecture and Complexity . . . . .	56

4.3	Training and Validating the Model . . . . .	57
4.4	Testing the Model . . . . .	57
4.5	Results and Discussion . . . . .	58
<b>5</b>	<b>Conclusion and Recommendation</b>	<b>68</b>
5.1	Conclusions . . . . .	68
5.2	Recommendation . . . . .	70

# List of Figures

1.1	(a) Groningen gas field map [2]; (b) Crack in the wall [23]. . . . .	2
1.2	Richter scale [12] . . . . .	3
1.3	An example house modelled in DIANA and analysed using NLTH analysis © RHDHV . . . . .	3
1.4	Report Framework . . . . .	5
2.1	Masonry wall [25] . . . . .	6
2.2	Masonry prisms compressed [18] . . . . .	7
2.3	(a) Clay Brick < 1945, (b) Clay Brick > 1945, (c) Calcium Silicate Brick > 1960 . . . . .	7
2.4	(a) wall-to-wall connection and no diaphragm, (b) good wall-to-wall connection with flexible diaphragm (c) good wall-to-wall connection and rigid diaphragm [21] . . . . .	8
2.5	Piers and Spandrels in Masonry wall . . . . .	9
2.6	In-Plane Failure Mechanisms [10] . . . . .	9
2.7	Different flanged wall [13] . . . . .	10
2.8	Out of Plane Failure Mechanisms . . . . .	10
2.9	Different Seismic Analysis Methods [11] . . . . .	12
2.10	Example of SLaMA analysis in the case of timber floor . . . . .	14
2.11	Example of SLaMA analysis in the case of concrete floor . . . . .	14
2.12	Representation of Linearly distributed moment case of (a) timber floor and (b) concrete floor . . . . .	15
2.13	Flange Effect © RHDHV . . . . .	17
2.14	Shear resistance due to rocking of pier [20] . . . . .	18
2.15	Force deformation relation in pier if bed joint sliding is governing [20] . . . . .	20
2.16	Example of Pier Capacity Curve . . . . .	21
2.17	Example of combining pier capacity to wall capacity . . . . .	22
2.18	Bilinearized pushover curve [20] . . . . .	22
2.19	A sample ADRS compliance check on a wall from VIIA reports © RHDHV . . . . .	23
2.20	Representation of Binary and Multi-class Classification . . . . .	24
2.21	Representation of Multi-label Classification . . . . .	24
2.22	SLaMA analysis Vs Surrogate Model . . . . .	25
2.23	Deep Neural Network Model . . . . .	25
2.24	Zooming in on a Single Neuron: Visualizing the Process Within Hidden Layers . . . . .	27
3.1	Comparison of random and Latin hypercube sampling (LHS) . . . . .	31
3.2	Sample Data Representation Using One-Hot Encoding . . . . .	33
3.3	Typical plot of L2 regularisation Vs Validation loss . . . . .	35
3.4	(a) Model is Overfitting, (b) Model is Underfitting, and (c) Model is a Good Fit. . . . .	36
3.5	Typical learning curve . . . . .	36
3.6	Validation loss vs L2 Regularization for 1 layer with different neurons for Xa input combination for buildings with timber floor . . . . .	38
3.7	Validation loss vs L2 Regularization for 2 layer with different neurons for Xa input combination for buildings with timber floor . . . . .	38
3.8	Validation loss vs L2 Regularization for 3 layer with different neurons for Xa input combination for buildings with timber floor . . . . .	38

3.9	Validation loss vs L2 Regularization for 4 layer with different neurons for Xa input combination for buildings with timber floor . . . . .	39
3.10	Validation loss vs L2 Regularization for 1 layer with 64 neurons for Xa input combination for buildings with timber floor . . . . .	39
3.11	Validation loss vs L2 Regularization for 1 layer with different neurons for Xb input combination for buildings with timber floor . . . . .	40
3.12	Validation loss vs L2 Regularization for 2 layer with different neurons for Xb input combination for buildings with timber floor . . . . .	40
3.13	Validation loss vs L2 Regularization for 3 layer with different neurons for Xb input combination for buildings with timber floor . . . . .	40
3.14	Validation loss vs L2 Regularization for 4 layer with different neurons for Xb input combination for buildings with timber floor . . . . .	41
3.15	Validation loss vs L2 Regularization for 2 layer with 8 neurons for Xb input combination for buildings with timber floor . . . . .	41
3.16	Epoch vs. Validation loss of the trained model with Xa input for buildings with timber floor . .	42
3.17	Epoch vs. Validation loss of the trained model with Xb input for buildings with timber floor . .	42
3.18	Train Size vs. Validation loss of the model with Xa input for buildings with timber floor . . . .	43
3.19	Train Size vs. Validation loss of the model with Xb input for buildings with timber floor . . . .	43
3.20	Scatter plot for Actual vs. Predicted values of output: (a) $S_y$ , (b) $u_y$ , and (c) $u_{max}$ . . . . .	44
3.21	Validation loss vs L2 Regularization for 1 layer with different neurons for Xa input combination for buildings with concrete floor . . . . .	45
3.22	Validation loss vs L2 Regularization for 2 layer with different neurons for Xa input combination for buildings with concrete floor . . . . .	46
3.23	Validation loss vs L2 Regularization for 3 layer with different neurons for Xa input combination for buildings with concrete floor . . . . .	46
3.24	Validation loss vs L2 Regularization for 4 layer with different neurons for Xa input combination for buildings with concrete floor . . . . .	46
3.25	Validation loss vs L2 Regularization for 2 layer with 64 neurons for Xa input combination for buildings with concrete floor . . . . .	47
3.26	Validation loss vs L2 Regularization for 3 layer with 64 neurons for Xa input combination for buildings with concrete floor . . . . .	47
3.27	Epoch vs. Validation loss of the trained model with Xa input for buildings with concrete floor .	48
3.28	Epoch vs. Validation loss of the trained model with Xb input for buildings with concrete floor .	48
3.29	Train Size vs. Validation loss of the model with Xa input combination, for buildings with concrete floor . . . . .	48
3.30	Train Size vs. Validation loss of the model with Xb input combination, for buildings with concrete floor . . . . .	49
3.31	Scatter plot for Actual vs. Predicted values of output: (a) $S_y$ , (b) $u_y$ , and (c) $u_{max}$ for concrete floor building. . . . .	49
3.32	Validation loss vs L2 Regularization for 1 layer with different neurons for Xc input combination for buildings with concrete floor . . . . .	50
3.33	Validation loss vs L2 Regularization for 2 layer with different neurons for Xc input combination for buildings with concrete floor . . . . .	50
3.34	Validation loss vs L2 Regularization for 3 layer with different neurons for Xc input combination for buildings with concrete floor . . . . .	51
3.35	Validation loss vs L2 Regularization for 4 layer with different neurons for Xc input combination for buildings with concrete floor . . . . .	51
3.36	Validation loss vs L2 Regularization for 1 layer with 128 neurons for Xc input combination for buildings with concrete floor . . . . .	51
3.37	Epoch vs. Validation loss of the trained model with Xc input for buildings with concrete floor .	52
3.38	fig:Train Size vs. Validation loss of the model with Xc input for buildings with timber floor . .	52
4.1	Example of constructing PC's . . . . .	55
4.2	PCA analysis for input parameters of Classification model . . . . .	56

4.3	Example of confusion matrix . . . . .	57
4.4	Validation loss vs L2 Regularization for 1 layer with different neurons for Classification model	58
4.5	Validation loss vs L2 Regularization for 2 layer with different neurons for Classification model	59
4.6	Validation loss vs L2 Regularization for 3 layer with different neurons for Classification model	59
4.7	Validation loss vs L2 Regularization for 4 layer with different neurons for Classification model	59
4.8	Validation loss vs L2 Regularization for 2 layer with 64 neurons for Classification model . . . .	60
4.9	Validation loss vs L2 Regularization for 3 layer with 32 neurons for Classification model . . . .	60
4.10	Validation loss vs L2 Regularization for 3 layer with 64 neurons for Classification model . . . .	60
4.11	Validation loss vs L2 Regularization for 3 layer with 128 neurons for Classification model . . .	61
4.12	Validation loss vs L2 Regularization for 4 layer with 16 neurons for Classification model . . . .	61
4.13	Train Vs Validation loss for Classification model with 2 layers, 64 neurons, L2=0.001 . . . . .	61
4.14	Train Vs Validation loss for Classification model with 3 layers, 32 neurons, L2=0.001 . . . . .	62
4.15	Train Vs Validation loss for Classification model with 3 layers, 64 neurons, L2=0.001 . . . . .	62
4.16	Train Vs Validation loss for Classification model with 3 layers, 128 neurons, L2=0.001 . . . . .	62
4.17	Train Vs Validation loss for Classification model with 4 layers, 16 neurons, L2=0.001 . . . . .	62
4.18	Confusion matrix for Classification model with 3 layers, 32 neurons, L2=0.001 . . . . .	63
4.19	Mutual Information score - Feature engineering . . . . .	64
4.20	Validation loss vs L2 Regularization for 2 layer with 64 neurons for Classification model . . . .	65
4.21	Epoch vs. Validation loss Train Vs Validation loss for Classification model with 2 layers, 64 neuron . . . . .	65
4.22	Train Size vs. Validation loss of the classification model . . . . .	65
4.23	Confusion matrix . . . . .	66
4.24	Class overlap of input variables in Classification model . . . . .	66

# List of Tables

2.1	Clay Brick (pre-1945)	16
2.2	Clay Brick (post-1945)	16
2.3	Calcium Silicate (Brickwork)(post-1960)	16
3.1	Input features for the surrogate model of a Timber floor building	32
3.2	Input features for the surrogate model of a Concrete floor building	33
3.3	Input features for the surrogate model of a Timber building	33
3.4	MAE, MSE, RMSE values for buildings with timber model	44
3.5	MAE, MSE, RMSE values for buildings with concrete model	50
4.1	Input features for the Classification model	54
4.2	Output features for the Classification model	54
4.3	Evaluation metrics Comparison of Model Architectures	63
4.4	Comparison of Training and Validation Loss and Accuracy	63

# List of Abbreviations

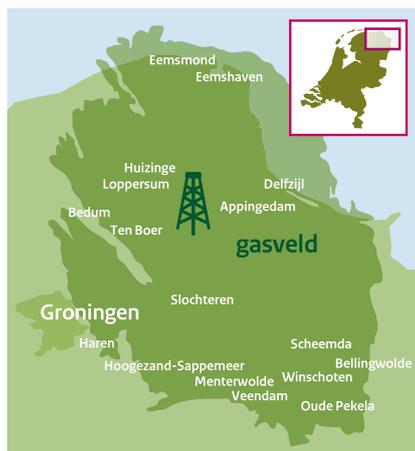
Abbreviation	Full form
PGA	Peak Ground Acceleration
NLPO	Non-Linear Pushover Analysis
URM	Unreinforced Masonry
Sa	Spectral Acceleration
Sd	Spectral Displacement
NSCE	Non seismic structural elements
PSSE	Primary seismic structural elements
MRS	Modal Response Spectrum Analysis
NLPO-SDF	Nonlinear Pushover Analysis-Single Degree of Freedom
NLPO-SLaMA	Nonlinear Pushover Analysis-Simplified Lateral Mechanism Analysis
NLTH	Nonlinear Time History Analysis
REF	Reference Approach
LF	Lateral Force Method
$\eta$	Damping correction
TB	Lower limit of the constant acceleration part of the spectrum
TC	Upper limit of the constant acceleration part of the spectrum
TD	lower limit of the constant spectral displacement part of the spectrum
p	acceleration spectral amplification factor
g	acceleration due to gravity
DNN	Deep Neural Network
ADRS	Acceleration Displacement Response Spectrum

# Chapter 1

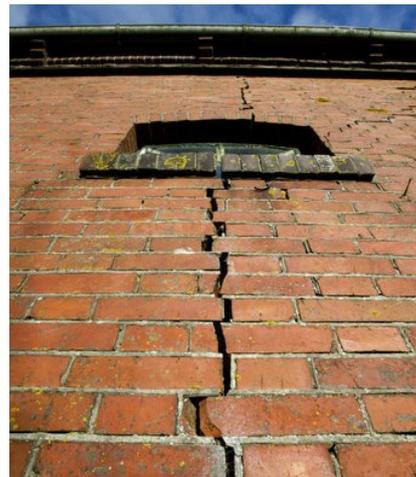
## Introduction

### 1.1 Research Context

The province of Groningen located in the North-East of the Netherlands has one of the largest gas fields in the world. Shell and Exxon Mobil own the gas fields in partnership with the Dutch Government. However, since 1986, the province has experienced more than 1000 seismic shocks due to the gas extraction. The earthquakes in the province have led to severe damage to the houses and the mental health of residents, see Figure 1.1. Huizinge earthquake in 2012, with a magnitude of 3.6 is one of the largest induced earthquakes the province has faced.



(a)



(b)

Figure 1.1: (a) Groningen gas field map [2]; (b) Crack in the wall [23].

Groningen is not traditionally an earthquake-prone area, hence the structural designs in the province have not incorporated seismic resistance measures. Majority of the buildings in this province are built of Unreinforced Masonry (URM), making them vulnerable to seismic shocks. So even though the magnitude of the earthquakes experienced by Groningen are within the lower range of Richter scale, see Figure 1.2, ranging between 0.5 and 3.0 magnitude [24] the structures still suffer damage.

Reinforcement for the houses and compensations are being provided for the residents, however, they are progressing slowly. The gas extraction is planned to be permanently ended by 2024 [22]; however, experts have stated that the earthquakes could continue [17].

To provide reinforcement to structures damaged by the induced earthquake, NPR9998 (Nederlandse Praktijkrichtlijn) a National practical guideline was developed and published by NEN (Nederlands Normalisatie Instituut) in 2015. The NPR focuses on the safety of the residence and not the prevention of cracks, cracks can occur on the structure even with the use of NPR. The NPR is scoped and tailored only for the North Eastern

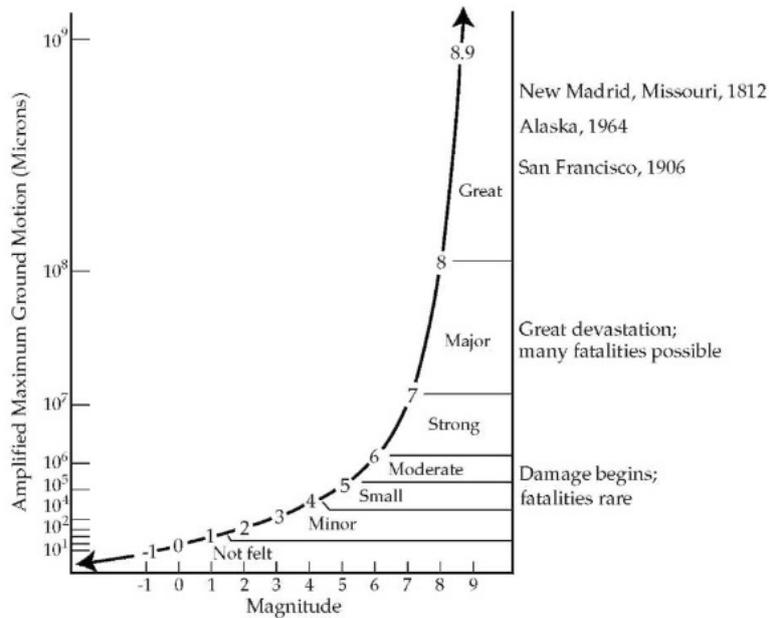


Figure 1.2: Richter scale [12]

region of the Netherlands as only that region is affected by induced seismic activity. Since 2015 the NPR have been revised three times and the latest version used for the assessment and safety of the buildings is NPR9998 2020 [20].

To implement structural reinforcement measures to the structures in Groningen, the National Coordinator Groningen (NCG) works on researching and strengthening buildings. The NCG outsources the reinforcement operations to a number of engineering company in Netherlands. One of the many companies is Royal HaskoningDHV (RHDHV). Since 2015 RHDHV, have initiated the VIIA project, within the Advance, Research and Technology department. The primary aim of the project is to make the structures within the earthquake area safer, by making sure the buildings are within the standard compliance. They conduct structural surveys and then analyse the data collected, by building a model of the structure in DIANA FEA and subjecting it to ground motions to provide the necessary strengthening measures for that structure, an example is shown in Figure 1.3. The structural analysis of the buildings is conducted using various methodologies, all adhering to the Dutch standard NPR 9998. However, each analysis method is time-consuming and repetitive. Initially, they started with the Model response spectrum, followed by non-linear time history, and later with non-linear pushover analysis. They have even tried to combine these procedures; however, the processes are not fast enough. Research is needed to investigate methods to reduce the time consumed to model and analyse the houses. All the buildings analyzed by the VIIA team since 2021 have been documented, and this information will be utilized in this research, this documentation is explained more in Section 2.3.

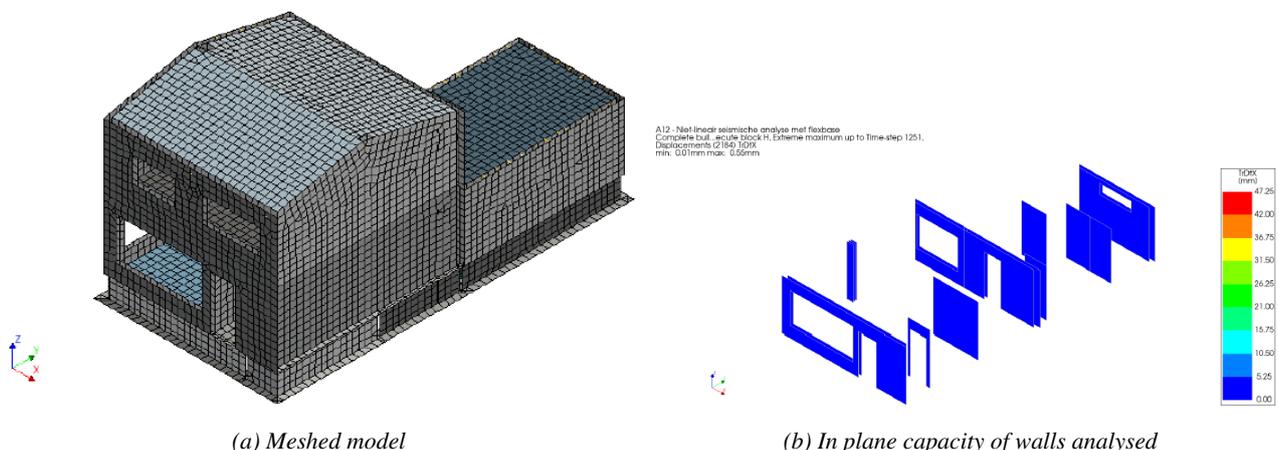


Figure 1.3: An example house modelled in DIANA and analysed using NLTH analysis © RHDHV

## 1.2 Research Objectives and Questions

The seismic effects on the Groningen building and associated retrofitting measures have been researched largely over the last few years. Investigations into the specific in-plane and out-plane load-bearing wall capacities have also been extensive. These researches have made use of time-consuming numerical analysis. However, there remains a gap in research regarding the application of machine learning algorithms to predict the stability of unreinforced masonry (URM) buildings in Groningen—a topic that has not yet been explored. The research aims to reduce the need for large analysis to the essential mechanism and essential findings and conclude with basic rules that can give us the fundamental limit of the seismic capacity of the building.

To achieve the goal of the study the primary research question is:

*“To what extent machine learning approaches can be leveraged, to optimise and accelerate the time required for modelling and analysing URM residential structures?”*

A few research questions that would help lead to the primary research question are;

- What are the effects of earthquakes on URM buildings, and what are the associated failure mechanisms?
- According to the existing database, what are the various seismic analysis methods currently employed to assess unreinforced masonry (URM) buildings?
- For the scope of this research, which types of machine learning models are suited for analysis?
- What are the potential input and output parameters that can be extracted from an existing database of characteristic parameters and analysis results [2.3], for developing a machine learning model for the seismic analysis of unreinforced masonry (URM) buildings?
- Is the available data sufficient to train, test, and validate a machine learning model?

## 1.3 Research Scope

The scope of this study is explicitly limited to analyzing the in-plane capacity of URM structures in Groningen, focusing on the overall structural response. While out-of-plane collapse mechanisms are acknowledged as potentially critical factors that can lead to progressive local failure and ultimately global structural collapse, they are not explicitly considered within the scope of this study. The primary objective here is to develop and evaluate two machine learning models: a classification model and a surrogate model, both aimed at assessing the need for reinforcement in URM buildings. The classification model is designed to categorize buildings based on whether reinforcement is necessary, while the surrogate model aims to approximate the outcomes of detailed seismic analysis to provide insights into structural integrity. It is important to note that the findings and results presented pertain specifically to Primary seismic structural elements (PSSE), with Non seismic structural elements (NSCE) structures being outside the scope of this investigation.

## 1.4 Research Method

The study will look into a combined assessment based on machine learning algorithms and analytical procedures. The research will utilise the results from previous numerical simulations and will enhance these findings with additional calculations. In the preliminary phase of the research, the focus was on understanding the behaviour of unreinforced masonry walls, how they are damaged due to earthquakes and understanding machine learning algorithms. Using Python software, a classification model and a surrogate model will be developed through machine learning algorithms. Both models will provide results indicating whether the building requires reinforcement measures or not. The classification model was selected because, it offers quick, actionable results to categorise buildings and make immediate decisions. It allows for the rapid identification of high-risk structures, saving time by filtering out lower-risk cases and prioritizing high-risk structures for further analysis. For the classification model, input and output parameters are sourced from the existing database maintained by RHDHV. The model is explained in detail in Chapter 4.

While the surrogate model approximates complex, computationally expensive simulations with simpler, faster models, enabling quick yet reliable predictions of structural behaviour. The surrogate model utilizes data generated from SLAMA analysis [2.4]. These results were used to build a surrogate model to assess the feasibility of a more computationally expensive approach. Since the SLAMA method is simpler than other analyses, it allows for quicker data generation and facilitates rapid iteration of different model complexities. This study explains how reduced input parameters help the model learn the relationship between input and target variables, and how well the model predicts outcomes. The impact of increasing input features, which leads to the need for more data points—often referred to as the curse of dimensionality—all are discussed in detail in Chapter 3.

Initially, this analysis is performed using Excel to create a preliminary dataset, which is then expanded and processed with Python to develop a more comprehensive dataset. The input for the preliminary dataset is derived from existing SLAMA reports within the RHDHV archive. The classification model and surrogate model are explained in more detail in Chapter 2.

The general structure of this report is outlined below:

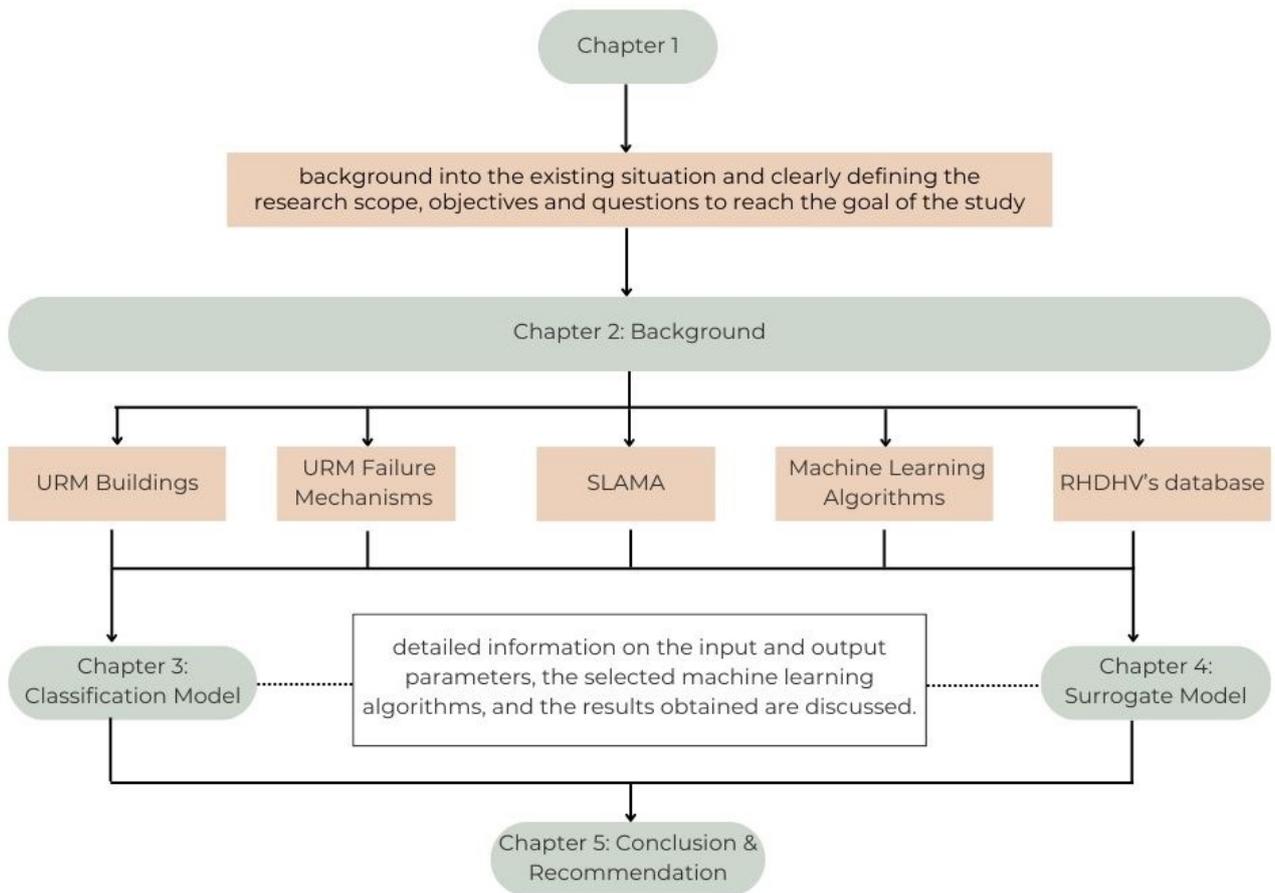


Figure 1.4: Report Framework

## Chapter 2

# Literature Review

### 2.1 Unreinforcement Masonry Buildings

Masonry is an anisotropic, two-phase composite material composed of mortar and other masonry units. It is a building material that mainly functions as a structural wall in various types of constructions. The anisotropy of masonry - meaning its mechanical properties vary with direction. - plays a crucial role in its behaviour under load. In this section, the focus will be on describing the masonry composites found in Groningen, which are clay-brick<1945, clay-brick>1945 and calcium-silicate>1960.

#### 2.1.1 Mechanical Properties of Masonry

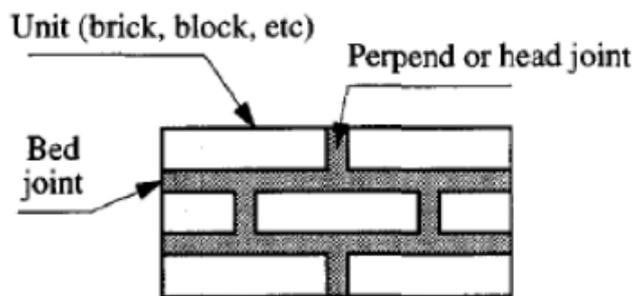


Figure 2.1: Masonry wall [25]

The tensile strength of masonry behaves differently based on their tensile force orientation to the bed-joints [27]. In Figure 2.1, when tensile strength is perpendicular to bed-joint, failure can occur at the interface of the bricks and mortar due to poor cohesive bond due to the difference in the manufacturing time of the constituents of the masonry structure. But, the tensile strength is observed to be higher when it is perpendicular to the head-joint. The cohesive bond in the head-joint and the cohesive strength in the bed-joints resist slipping of the bricks. The vertical loads, increases the frictional forces at the bed-joint, increasing the strength.

The compressive strength of the masonry depends on the strength of the masonry unit, mortar and the quality of the bond between the bricks and the mortar. According to Hilsdorf's investigations, compressive failure in masonry typically initiates from vertical cracks in the bricks [8] [16]. As the masonry is compressed, Poisson's effect causes both the bricks and mortar to attempt lateral deformation. However, due to the lower stiffness of the mortar compared to the bricks, the mortar deforms more laterally. This results in a tri-axial compressive state in the mortar and a state of vertical compression with lateral tension in the bricks. The tri-axial state is due to stress in the vertical direction from applied load, and stress in the horizontal direction due to Poisson's effect and constraints from the brick. This lateral tension in the bricks, caused due to Poisson's effect as the bricks are unable to expand freely and are instead subjected to tensile forces in the horizontal directions as they try to deform laterally, leads to vertical cracks in the brick, as seen in Figure 2.2.

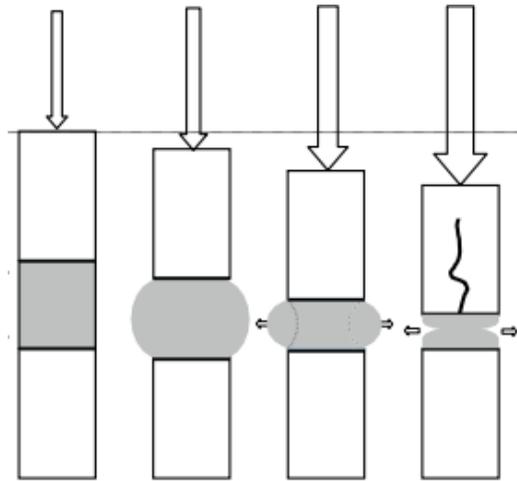


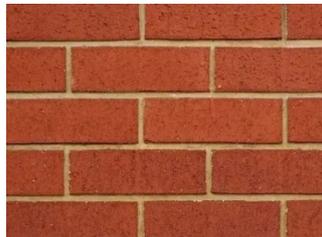
Figure 2.2: Masonry prisms compressed [18]

### 2.1.2 Wall Material

The distinction between the wall materials is related to their manufacturing processes, material properties, and performance characteristics. This study adopts the classification system proposed in NPR9998 [20].



(a)



(b)



(c)

Figure 2.3: (a) Clay Brick < 1945, (b) Clay Brick > 1945, (c) Calcium Silicate Brick > 1960

Clay Bricks manufactured pre-1945, Figure 2.3 (a), are often handmade or made using less mechanised kilns. The process would involve shaping clays into molds and firing them, resulting in varying shapes, colours and sizes. Their density and compressive strength are lower compared to modern bricks due to the less controlled firing process. They generally have high porosity, leading to increased moisture absorption, which can affect durability over time.

Clay Bricks manufactured post-1945, Figure 2.3 (b), allowed for more uniform production, with better control over the firing process in modern kilns. The density and compressive strength are higher due to more controlled firing and improved quality of raw materials. The advancements in manufacturing techniques and quality control reduced porosity compared to pre-1945 bricks.

Calcium Silicate bricks post-1960, Figure 2.3 (c), are manufactured from silica, lime and water. These bricks typically exhibit higher compressive strength compared to clay bricks, offering a more uniform appearance and superior thermal insulation. However, they are more brittle than clay bricks, making them prone to sudden failure when cracks develop, whereas clay bricks can endure larger deformations before failing [14]. Compared to clay bricks they have higher porosity [9], making the bricks susceptible to frost damage and reduced thermal performance.

The strength and durability of these wall materials make them suitable for both load-bearing and non-load-bearing applications. More about the material properties are discussed in section 2.4.

Mortar also plays a crucial role in masonry construction, acting as the binding agent that holds together bricks, stones, or blocks to create stable and durable structures. Its primary functions include bonding the masonry units to each other, distributing loads evenly across the structure, and providing resistance against environmental factors such as moisture and temperature fluctuations. Typically, mortar is composed of a mixture of cement, lime, sand, and water. The type and composition of mortar can significantly impact the strength, flexibility, and durability of the masonry.

### 2.1.3 Building box behavior

When a masonry house is subjected to an earthquake, the likelihood of failure depends on the performance of its individual structural elements. Poor connections between the walls and floors can lead to immediate out-of-plane failure, a localized type of structural failure. However, in this research, we assume that the connections between the walls and floors are strong or have been reinforced prior to analysis. This assumption results in what is known as the 'box behavior' of masonry houses, where strong connections allow the structure to act as a cohesive unit during seismic events.

The seismic performance of URM structures depends on the type of diaphragm (floor) used and the type of connection between the wall and the diaphragm. If the diaphragm is rigid, it will not bend or stretch easily, and because it is rigid it will effectively distribute the load to the walls it is connected to during a seismic activity. If the diaphragm is flexible, it will bend or stretch more, it does not distribute the lateral load uniformly creating a localized deformation, as shown in the Figure 2.4. Commonly used materials for diaphragm is timber which is flexible or concrete which is rigid. Timber floors result in the walls acting as cantilever walls, which increases their susceptibility to rocking failure. Conversely, concrete floors constrain the walls at both ends, making them more prone to diagonal tension or sliding failures, as they experience load from the top and, hence less prone to rocking failure.

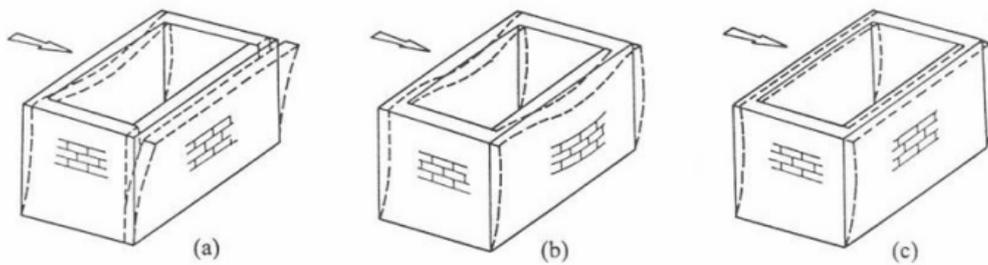


Figure 2.4: (a) wall-to-wall connection and no diaphragm, (b) good wall-to-wall connection with flexible diaphragm (c) good wall-to-wall connection and rigid diaphragm [21]

### 2.1.4 Piers and Spandrels

The masonry walls are conventionally divided into piers and spandrels by the door and window openings. The vertical elements, marked in blue in Figure 2.5, are piers, they are considered as the primary elements, responsible for resisting both gravity loads and lateral forces acting on the structure. The horizontal or secondary elements, marked in green in Figure 2.5, are spandrels. This division helps to reduce the complex behaviour of the wall during analysis. Depending on the opening size, if the piers are weaker than spandrels then the piers would fail first. The interaction between piers and spandrels is critical in determining the failure mechanism of the wall in the seismic environment. Most of the floors in Groningen are of concrete or timber. When floors are made of concrete, spandrels tend to be rigid. This rigidity provides a double-clamped condition for the piers, meaning the piers are supported at both ends, enhancing their stability and load-bearing capacity. With timber floors, the spandrels are more flexible. This flexibility allows the piers to act more like cantilever elements, where one end is fixed, and the other is free to move. This condition can lead to increased deformation and potential instability in the piers.

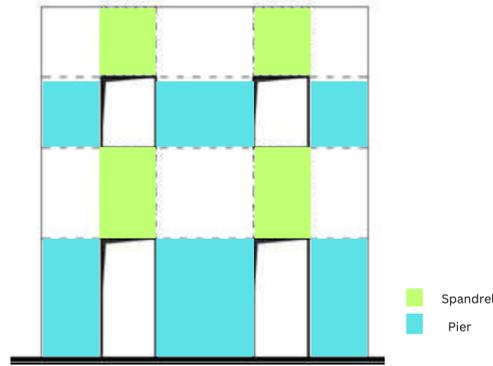


Figure 2.5: Piers and Spandrels in Masonry wall

## 2.2 Failure Mechanisms of URM walls

### 2.2.1 In Plane Failure

The in-plane seismic capacity of buildings is their ability to withstand and resist seismic forces acting within the plane of the building, typically along the horizontal direction. In seismic engineering, understanding a building's in-plane seismic capacity is crucial for assessing its structural performance and potential vulnerability during an earthquake.

In Figure 2.6 the in-plane failure mechanisms, which are diagonal tension, bed joint sliding, rocking failure and toe crushing, are represented respectively.

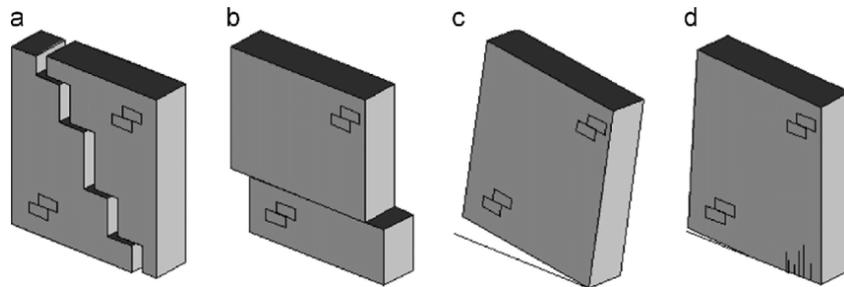


Figure 2.6: In-Plane Failure Mechanisms [10]

The rocking and bed joint sliding failure are deformation-controlled failure mechanisms, as the structure does not go into a sudden brittle failure but a controlled deformation. While toe crushing and diagonal tension are force-controlled failures because they result in a sudden brittle failure when the applied load is beyond the resistance of the structure.

Since in the latest NPR9998 2020 [20], detailed calculation methods and provisions are only provided for Rocking and bed joint sliding resistance, hence only they will be considered in the modelling. In Rocking failure, the wall/pier rotates about its base and it is mostly observed in slender piers. Bed joint sliding failure occurs due to a poor coefficient of friction between mortar and masonry or low vertical load on the wall/pier.

The in-plane walls are prone to instability and failure during an earthquake. The flange, which is the adjacent transverse wall to the in-plane wall, as shown in in Figure 2.7, increases the strength and displacement capacity of the wall when it's under the rocking or toe-crushing failure, this is called flange effect. The flanges can be of two groups: compression flanges and tension flanges [19].

A compression flange in a masonry wall is the part of the out-of-plane wall that resists compressive stresses from lateral loads. These flanges significantly influence the flexural capacity and stiffness of the adjacent pier. When the flange is in compression, it actively contributes to the structural integrity. However, if the load direction reverses, placing the flange in tension, can develop cracks contributing little to the pier response, as URM typically have low tensile strength [19].

Conversely, a tension flange is the part of the out-of-plane wall that resists uplift in the in-plane wall, typically

caused by local pier rocking or global overturning. Due to the low tensile strength of URM, these flanges contribute their weight to the in-plane wall's response. This weight helps resist overturning moments and increases the vertical stress on the piers, enhancing their stability [19].

Generally in URM buildings the wall-to-wall connections are either using steel ties or bricks embedded into adjacent walls [21]. These connections can sometimes be not enough, due to difference in stiffness between the in-plane and out of plane (or face loaded) walls, which can lead to loss of flange effect and softening of the structure.

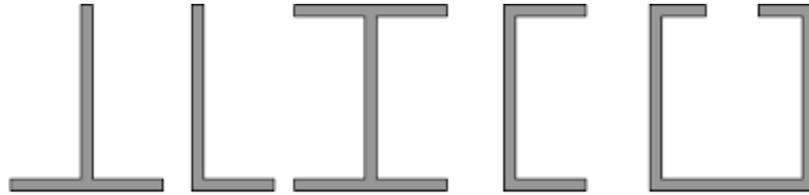


Figure 2.7: Different flanged wall [13]

### 2.2.2 Out of Plane Failure

The out-of-plane seismic capacity of buildings refers to their ability to withstand the force acting perpendicular to their plane. The out-of-plane failure of the structure during a seismic event could be due to large openings or long spanning walls or poor material.

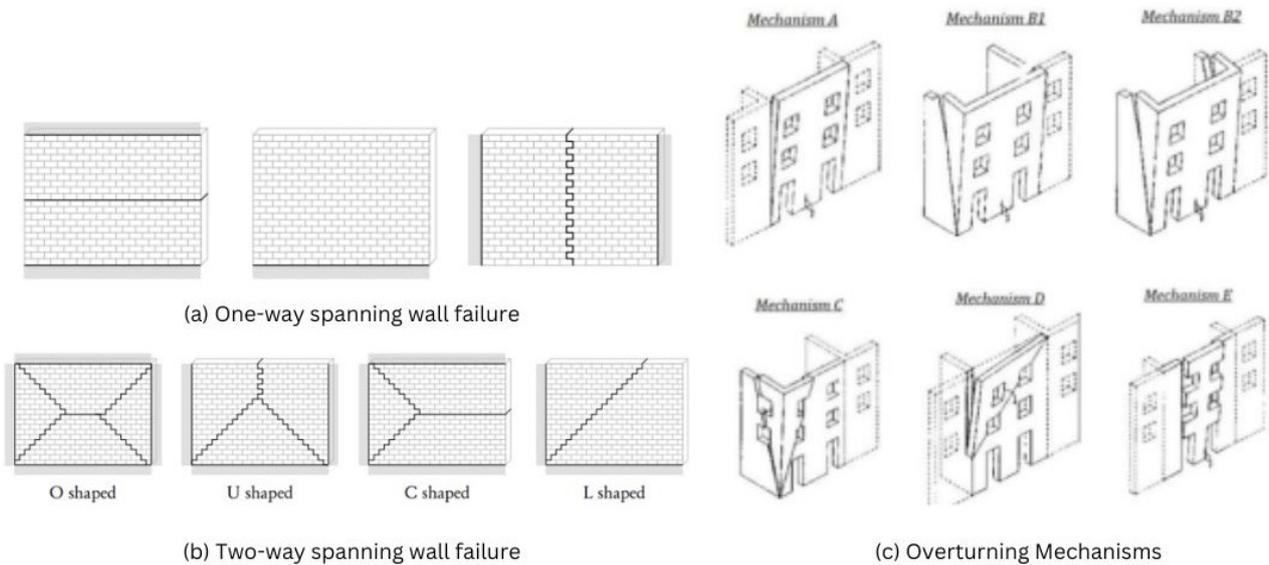


Figure 2.8: Out of Plane Failure Mechanisms

The wall behavior in the out of plane direction is usually governed by a flexural response, which develops in a different form for one-way and two-way spanning walls, as seen in Figure 2.8. In one-way spanning walls, the cracks run either horizontally or vertically, but in two-way spanning walls, the cracks could be a combination of horizontal, vertical or diagonal. During an earthquake, the connection between the wall and roof or wall and floor can affect the stability of the wall. If the connections are inadequate then the walls face an overturning mechanism, as seen in Figure 2.8 [3]. To attain the goal of the research, the focus will be on the in-plane capacity of masonry walls.

## 2.3 Seismic Vulnerability Database by VIIA

This research has utilised an existing database, provided by the VIIA team of Royal HaskoningDHV, and the database possesses a relational structure. This relational database is organised in tables within Excel, which was exported from Royal HaskoningDHV's web application, called MYVIIA, and is related through IDs provided in each sheet. This allows for efficient data management and data retrieval.

Each house is referred to as an 'object' with a unique ID. The houses are not always analysed as a whole, large objects are divided into sections and are referred to as 'parts' and each part is analysed separately and the analysis method used might not be the same for each part. The IDs for each object and their parts are unique. The objects are analysed using one of the following methods: Model Response Spectrum Analysis (MRS), Non-linear Pushover Analysis (NLPO), Non-linear Time History Analysis (NLTH), Lateral Force Method (LF) or Reference approach of NLTH or NLPO Analysis (REF).

- **NLPO:** It is a nonlinear static analysis method, where the response of the structure is estimated by gradually applying lateral force till the structure collapses. The NLPO analysis can be performed using numerical software or using an analytical approach called SLaMA, explained in Section 2.4. The database comprises analyses of 81 parts evaluated using numerical stimulation's and 81 parts analysed using analytical approach. After excluding components that were analyzed multiple times, the dataset includes 60 parts with SLaMA analysis and 66 parts analysed through numerical stimulation's.
- **NLTH:** It is a nonlinear dynamic analysis method, here a representative earthquake time history is required, which is an accurate representation of the dynamic behaviour of earthquakes. The NLTH method requires a 3D model of the structure to capture its intricate behaviour. The database comprises analyses of 282 parts evaluated using NLTH approach. After excluding components that were analyzed multiple times, the dataset includes 250 parts with NLTH analysis.
- **MRS:** Is a linear dynamic analysis, the maximum response of velocity, acceleration or displacement is found at each mode of vibration and then maximum responses are summed to get the overall structural response. The database comprises analyses of 305 parts evaluated using MRS approach. After excluding components that were analyzed multiple times, the dataset includes 266 parts with MRS analysis.
- **LF:** Also known as equivalent static analysis, it converts the dynamic forces to static and applies them laterally to the structure for analysis purposes, which makes this analysis the most conservative among the others described. The database comprises analyses of 1 part evaluated using LF approach.
- **REF:** this analysis method recommends reinforcement measures to a new object by extending the results from previously similar objects, which were assessed using NLTH or NLPO analysis. A good reference object is comparable based on peak ground acceleration (PGA) values, the connection between walls, wall thickness and primary seismic structural elements (PSSE) material. The reference approach is applied for NLTH and NLPO analysis. The database comprises analyses of 302 parts evaluated using REF-NLTH and 24 parts analysed using REF-NLPO. After excluding components that were analyzed multiple times in REF-NLTH approach, the dataset includes 290 parts with REF-NLTH approach.

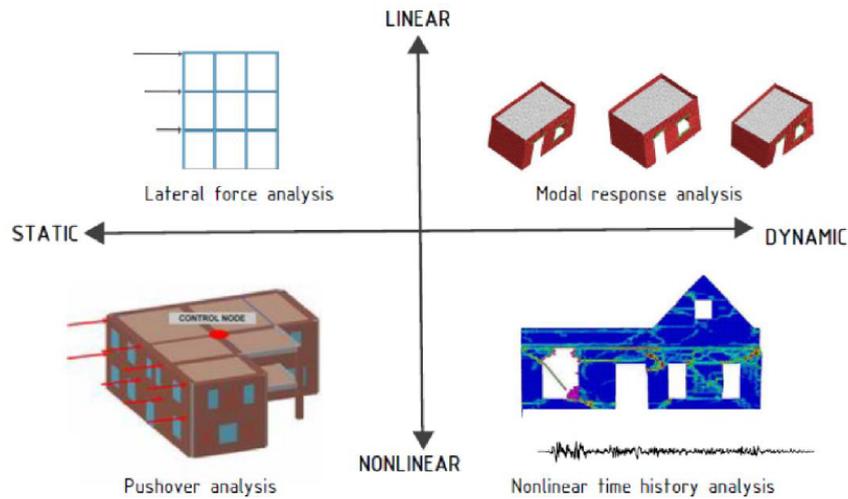


Figure 2.9: Different Seismic Analysis Methods [11]

Figure 2.9 shows a quick summary of each analysis mentioned above. Along with the method used to analyse the object, the database also gives engineering details such as the materials used for walls and floor, NPR9998 followed, length of the walls, building height, thickness of the wall, PGA value and many more.

Since the start of the VIIA project in 2015, the NPR's have been update. Below a clear overview of the NPR versions used, and shift in analysis methods are discussed.

NPR Versions Used:

- NPR9998:2015
- NPR9998:2017
- NPR9998:2018
- NPR9998:2018+C1+A1:2020
- NPR9998:2020

The latest NPR9998:2020 is made use mainly by the REF-NLTH, MRS and NLTH analysis methods.

Shift in analysis methods:

- The database includes data from 2015 to January 2024.
- From database it can be observed that initially MRS and NLPO approached was used and then shifted to MRS and NLTH.
- NLPO analysis: followed NPR9998:2018 or NPR9998:2018+C1+A1:2020.
- By 2022 the REF approach began along with NLTH and MRS analyses.
- By 2023, the use of the REF approach increased, particularly REF-NLTH, alongside NLTH and MRS analyses.
- As of January 2024, the analyses are primarily NLTH, with a few MRS analyses.

## 2.4 Simplified Lateral Mechanism Analysis (SLaMA)

There are several seismic analysis methods to evaluate the in-plane capacity of URM buildings. This paper makes use of a displacement-based assessment method called SLaMA. The guidelines for this analysis are prescribed within the Dutch NPR9998:2020 [20] and 2017 New Zealand Technical Guidelines for Seismic Engineering Assessments (Part C) [5]. The method focuses on determining the force-displacement (pushover) capacities of individual structural elements at a local level and subsequently evaluates the global failure of the structure. The analysis can be performed by hand or using Excel spreadsheets, hence it's found to be a cost-effective analysis method, that can be applied quickly for the initial feasibility studies.

The relative strength between piers and spandrels can be influenced by factors such as varying sizes of openings, floor types, and their connections to the walls. This variation causes the piers to experience in-plane failures, including rocking, bed joint sliding, toe crushing, and diagonal tension. Hence, the inelastic mechanism expected to form in the structures during the seismic shocks is the pier mechanism. In the pier mechanism, the inelastic behaviour will be concentrated within the pier [6].

The key steps for the SLaMA analysis are outlined below:

1. Acquire relevant data of the structure (wall and floor material, geometry, flange pier, overburden load, material properties and details) from its existing drawing plans and reports, at the pier level.
2. Determine the overburden load due to the flange effect for each pier that has a flange.
3. Calculate the shear resistance for in-plane failures (rocking and bed joint sliding), to identify the failure mechanism affecting the pier.
4. Evaluate the flexural and shear displacement of the pier to compute its yield displacement.
5. For the identified type of failure mechanism, compute the near-collapse displacement and force values. In this study, second-order effects for the piers are considered.
6. For buildings with timber floors, their compliance with safety is determined by evaluating the compliance of individual walls. Here, the structure operates as a series system, meaning that the entire structure requires reinforcement if one wall fails. Hence, when the floor material is timber, the capacity curve of the wall can be derived by integrating the capacities of the individual piers, as shown in Figure 2.10.
7. For buildings with concrete floors, their compliance with safety is determined by evaluating the compliance of the entire structure. In these scenarios, the structure operates as a parallel system, meaning that if one wall fails, the load can be redistributed to the adjacent walls. Hence, when the floor material is concrete, the capacities of the individual piers of all the walls are integrated to derive the capacity curve of the building, as shown in Figure 2.11.
8. The wall capacity curve or the building capacity curve, expressed in terms of base shear (kN) and deformation (mm), can be transformed into a bilinear  $S_a$  (g) and  $S_d$  (m) curve according to NPR Section G.4.2 [20]. This transformed curve can then be placed into the ADRS curve obtained through the location-specific seismic parameters to evaluate the compliance of the wall and determine if reinforcement is necessary.

In this paper, the SLaMA Analysis was first facilitated through Excel and then converted into a Python script to help generate more data.

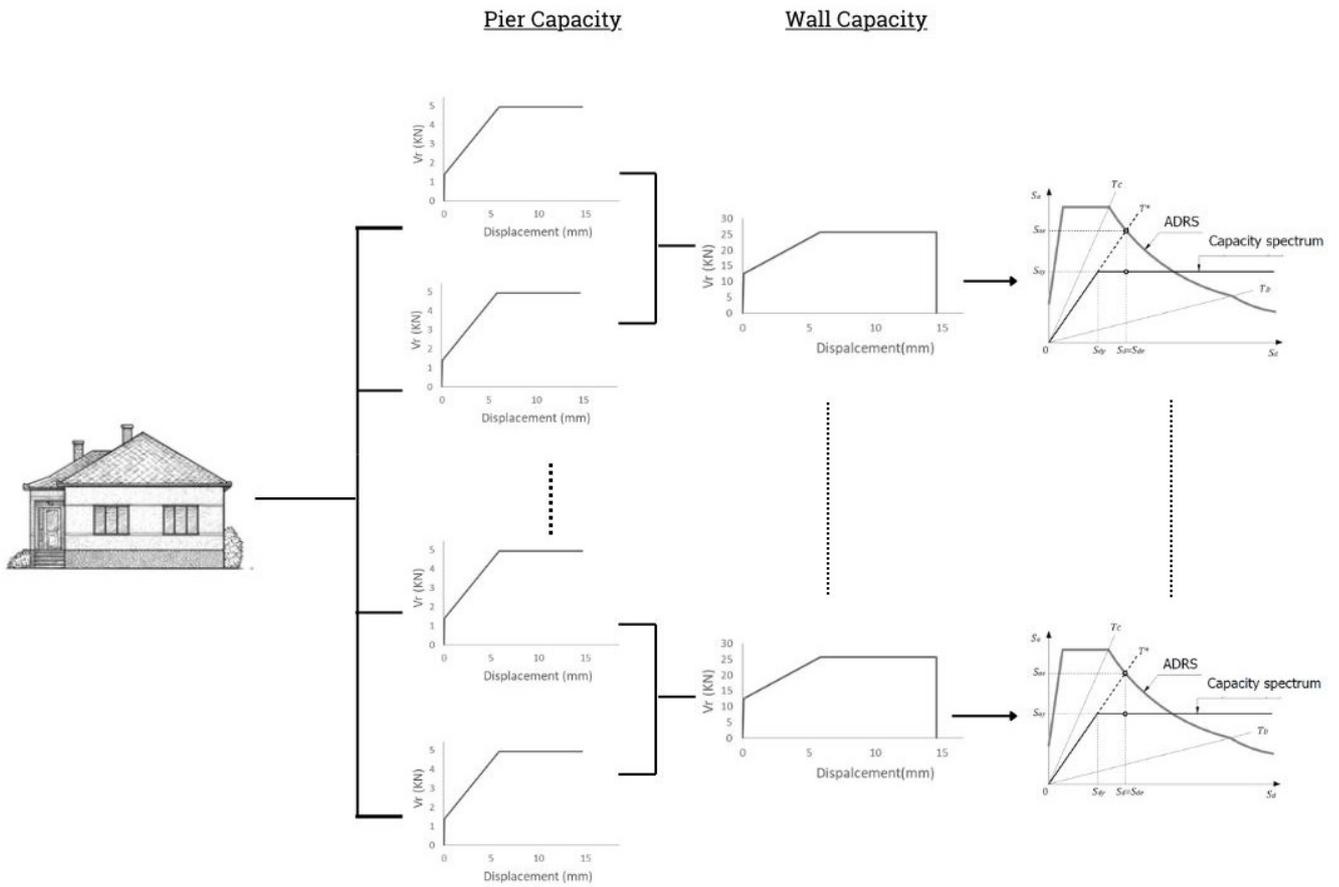


Figure 2.10: Example of SLAMA analysis in the case of timber floor

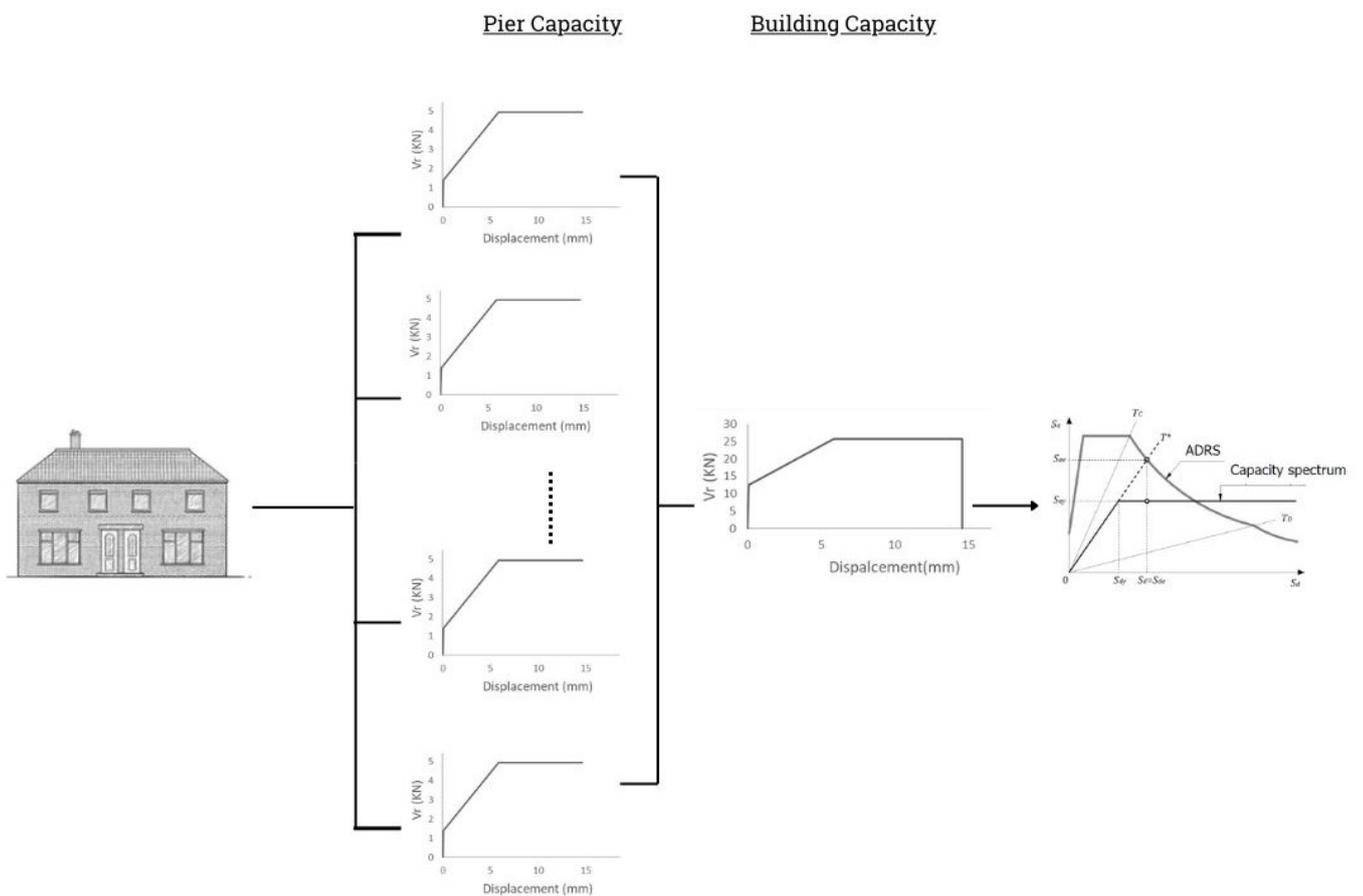


Figure 2.11: Example of SLAMA analysis in the case of concrete floor

## 2.5 Application of the SLaMA-URM method

This section will provide the steps followed to set up the SLaMA analysis. All the parameters and calculations done are in units of kilonewton and millimetres. The necessary input parameters for the SLaMA analysis such as the length, height, thickness of the piers and other parameters are obtained through the VIIA teams relational database, as specified in section 2.3.

### 2.5.1 Details on Overburden load and Wall Material

The VIIA reports provides a detailed description of how the load from the roof and floor is distributed among the walls based on the spanning direction of the floor. The equation used to calculate the overburden load on a wall is given below, equation 2.1, with the values for the parameter within the report. The axial overburden load is then equally distributed among the piers.

$$OverburdenLoad(kN) = A_{\text{floor, axial}} \times (g_{k,\text{floor}} + \varphi \times \psi_2 \times q_{k,\text{floor}}) + A_{\text{roof, axial}} \times (g_{k,\text{roof}} + \varphi \times \psi_2 \times q_{k,\text{roof}}) \quad (2.1)$$

where,

$A_{\text{floor, axial}}$  = the floor area contributing to the load ( $\text{m}^2$ )

$A_{\text{roof, axial}}$  = the roof area contributing to the load ( $\text{m}^2$ )

$g_k$  = unfactored permanent load ( $\text{kN}/\text{m}^2$ )

$q_k$  = unfactored variable load ( $\text{kN}/\text{m}^2$ )

$\varphi = 0.6$ , factor for variable load

$\psi_2 = 0.3$ , factor for variable load

Given the overburden loads, a portion of the masonry self-weight also influences the pier resistance depending on the floor material. As mentioned in section 2.1.3 timber floors result in the walls acting as cantilever walls, which increases their susceptibility to rocking failure. Conversely, concrete floors constrain the walls at both ends, making them more prone to diagonal tension or sliding failures, as they experience load from the top and, hence less prone to rocking failure. Therefore, for walls with concrete floors, self-weight from half the height of the pier is considered, whereas for timber floors, self-weight from the full height of the pier is considered, as shown in Figure 2.12.

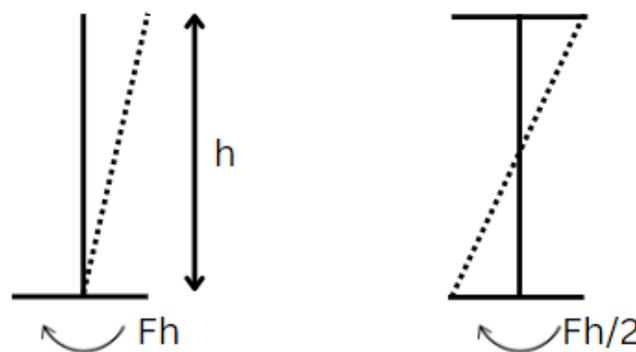


Figure 2.12: Representation of Linearly distributed moment case of (a) timber floor and (b) concrete floor

The material properties provided for Clay Brick (pre-1945), Clay Brick (post-1945), and Calcium Silicate (Brickwork)(post-1960) are provided in Table 2.1, 2.2, and 2.3 respectively.

Property	Symbol	Value
Density	$\rho$	1900 kg/m <sup>3</sup>
Masonry Strength	$F_{ma;m}$	8.5 MPa
Joint Strength	$f_j$	2 MPa
Cohesion	$F_{ma;v;0;m}$	0.3 MPa
Friction Coefficient	$\mu_f$	0.75
Modulus of Elasticity	E	5000MPa
Shear Modulus	G	2000MPa
Normalized mean compressive strength	$f_b$	20MPa

Table 2.1: Clay Brick (pre-1945)

Property	Symbol	Value
Density	$\rho$	1900 kg/m <sup>3</sup>
Masonry Strength	$F_{ma;m}$	10 MPa
Joint Strength	$f_j$	2 MPa
Cohesion	$F_{ma;v;0;m}$	0.4 MPa
Friction Coefficient	$\mu_f$	0.75
Modulus of Elasticity	E	6000MPa
Shear Modulus	G	2500MPa
Normalized mean compressive strength	$f_b$	20MPa

Table 2.2: Clay Brick (post-1945)

Property	Symbol	Value
Density	$\rho$	1850 kg/m <sup>3</sup>
Masonry Strength	$F_{ma;m}$	7 MPa
Joint Strength	$f_j$	2 MPa
Cohesion	$F_{ma;v;0;m}$	0.4 MPa
Friction Coefficient	$\mu_f$	0.75
Modulus of Elasticity	E	4000MPa
Shear Modulus	G	1650MPa
Normalized mean compressive strength	$f_b$	12MPa

Table 2.3: Calcium Silicate (Brickwork)(post-1960)

### 2.5.2 Effect of flange on pier

To account for the flange effect as discussed in section 2.2.1, the first step after collecting the structural data of the building is to calculate the load exerted on the piers that feature a flange.

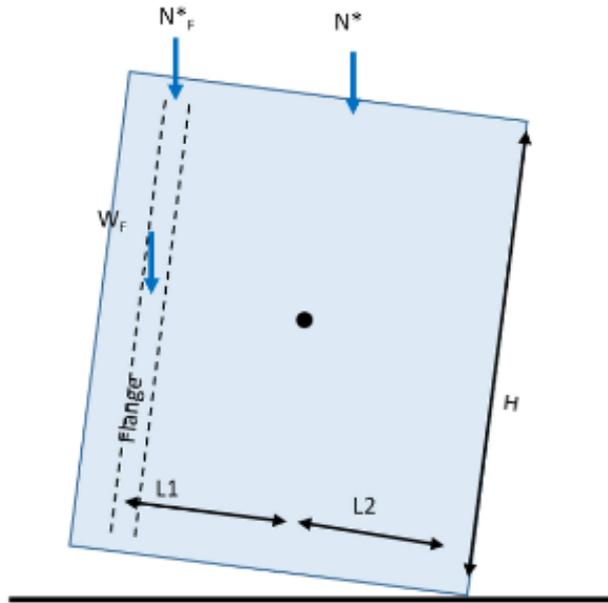


Figure 2.13: Flange Effect © RHDHV

Flanges provide additional capacity for Rocking and Toe Crushing failure mechanisms. Referring Figure 2.13, for each pier with flanges:

1. Name of the flange pier (exactly the same as the Pier Name) is determined.
2. The length ( $L_{\text{flange}}$ ), thickness ( $T_{\text{flange}}$ ) and height ( $H_{\text{flange}}$ ) of the flange pier are determined.
3. The effective length,  $L_{\text{eff}}$ , is calculated. It should be lesser of six times the flange thickness or the length of the flange [1].
4. Then determine  $L_2 = \frac{L_{\text{pier}}}{2}$
5. Computed  $L_1 = L_2 - \frac{T_{\text{flange}}}{2}$ , it is the length from the centre of the pier to the centre of the flange.
6. Overburden load on flange,

$$N_{\text{flange}}^* (\text{KN}) = \frac{N_{\text{flange}} \times L_{\text{eff}}}{L_{\text{flange}}}$$

7. The load due to flange effect,

$$N^* (\text{KN}) = (N_{\text{flange}}^* + W_{\text{flange}}) \times \frac{L_1 + L_2}{L_2}$$

, where  $W_{\text{flange}}$  is the self weight of the flange

### 2.5.3 Pier capacity

The governing failure mechanism for the pier is determined by calculating the shear resistance for rocking and bed joint sliding. The failure mechanism with the lower shear resistance is identified as the governing failure mode.

#### 2.5.3.1 Rocking Failure

The rocking mode of a pier often results in the crushing of its ends. Referring Figure 2.14, the resistance provided by a pier in the rocking mode based on flexural mechanism can be expressed by the following relationship [20]:

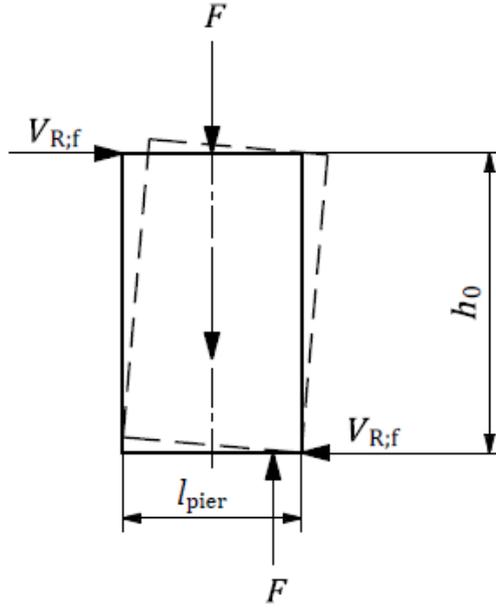


Figure 2.14: Shear resistance due to rocking of pier [20]

$$V_R = F \times \frac{l_{\text{pier}}}{2 \times h_o} \times \left( 1 - 1.15 \times \frac{\sigma_y}{F_{\text{ma;m}}} \right) \quad (2.2)$$

where:

- $V_R$  is the resistance force of the pier (KN),
- $F$  is the applied axial force, in combination with the dead load and the load due to the flange effect,
- $l_{\text{pier}}$  is the length of the pier,
- $\sigma_y$  is the average compressive stress over full cross-section of the pier  $\frac{F}{A_{\text{pier}}}$ ,
- $F_{\text{ma;m}}$  is the mean compressive strength of the masonry
- $h_o$  is the distance between the section where the shear capacity is achieved upon reaching the flexural resistance and the contraflexure point. This distance is based on the boundary conditions of the wall, as described in 2.12.

The following equation can determine the displacement capacity at near collapse for the rocking failure mode,

$$d_{NCr} = h_{\text{pier}} \times \theta_{\text{drift,r}} \quad (2.3)$$

$$\theta_{\text{drift,r}} = 0.0135 \times \left( 1 - 2.6 \times \frac{\sigma_y}{F_{\text{ma;m}}} \right) \times \frac{h_{\text{ref}}}{h_{\text{pier}}} \times \sqrt{\frac{h_{\text{pier}}}{l_{\text{pier}}}} \quad (2.4)$$

where,

- $d_{NCr}$  is the displacement at near collapse
- $h_{\text{pier}}$  is the height of the pier
- $h_{\text{ref}}$  is the reference height of the pier (=2400 mm)
- $\theta_{\text{drift,r}}$  is the drift limit near collapse

The corresponding shear resistance at near collapse is determined by equation 2.5. The equation considers the second-order effect for in-plane loads in masonry structures. The second-order effects, also known as P- $\Delta$  effects, arise from the lateral drift of the masonry pier. As the pier displaces laterally under applied loads, the load becomes eccentric due to the lateral movement. This eccentricity induces additional moments and stresses

on the pier, increasing the overall structural demand.

$$V_{\text{NCr}} = \left( \frac{(d_{\text{NCr}} - u_y) \cdot (-V_R)}{l_{\text{pier}} - u_y} \right) + V_R \quad (2.5)$$

Where:

- $V_{\text{NCr}}$  is shear at near collapse displacement
- $u_y$  is the yield displacement of the pier, its calculation is specified in section 2.5.4.

### 2.5.3.2 Bed Joint Sliding Failure

Bed joint sliding failure is a relatively stable mode of failure, that can undergo significant deformation even after initial cracking. The shear resistance of a URM pier is [20]:

$$V_R = l_c \times t_{\text{pier}} \times (F_{\text{ma};v;0;m} + \sigma_y \times \mu_f) \leq 0.1 \times f_b \times t_{\text{pier}} \times l_c \quad (2.6)$$

where:

- $V_R$  is the resistance force of the pier (KN) based on the shear mechanism,
- $l_c$  is the length of the compressed area at the end section of the pier. For a rectangular section, the  $l_c$  is computed as,

$$l_c = 3 \times \left( \frac{l_{\text{pier}}}{2} - \frac{M}{F} \right)$$

- $t_{\text{pier}}$  is the thickness of the pier
- $F_{\text{ma};v;0;m}$  is the initial mean shear strength of masonry
- $\mu_f$  is the mean masonry coefficient of friction
- $\sigma_y$  is the mean compressive stress in the compression area of the cross-section,  $\frac{F}{l_c \times t_{\text{pier}}}$
- $F$  is the axial force
- $f_b$  is the normalised mean compressive strength of the masonry units in the direction of the applied action effect. For clay bricks, it will be at least 20MPa and for calcium silicate bricks it will be at least 12MPa.
- $M$  is the moment in the cross-section, given by  $M = V_R \times h_o$ .

Magenes and Calvi [15] derived a simplified approach for the shear resistance due to bed joint sliding, using the Mohr-Coulomb formulation. The simplified formulation 2.7 has been used for the calculation of the shear resistance due to bed joint sliding.

$$V_R = l_p \times t_{\text{pier}} \times \left( \frac{1.5 \times F_{\text{ma};v;0;m} + \sigma_y \times \mu_f}{1 + \frac{3 \times F_{\text{ma};v;0;m} \times h_o}{\sigma_y \times l_p}} \right) \leq \frac{0.15 \times f_b \times t_{\text{pier}} \times l_p}{1 + \frac{3 \times F_{\text{ma};v;0;m} \times h_o}{\sigma_y \times l_p}} \quad (2.7)$$

where:

- $h_o$  is the effective height based on boundary condition 2.12.
- $l_p$  is the length of the pier

The residual shear resistance after  $F_{\text{ma};v;0;m}$  tends to zero under cyclic loading leaving only the frictional resistance can be calculated by setting  $F_{\text{ma};v;0;m} = 0$  in equation 2.7.

$$V_R = \mu_f \times F \quad (2.8)$$

If bed joint sliding failure governs then the Significant displacement capacity of the pier will have a drift limit  $\theta_{\text{SD}}$  of 0.003 and the drift limit value for a near collapse displacement would be 0.0075, see Figure 2.15.

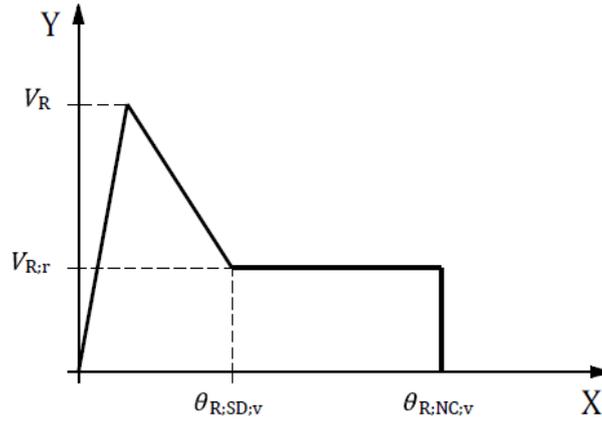


Figure 2.15: Force deformation relation in pier if bed joint sliding is governing [20]

#### 2.5.4 Yield Displacement of the Pier

Yield displacement refers to the point in the masonry pier when the pier undergoes plastic deformation. The total yield displacement of the pier is the sum of its flexural and shear displacement.

The flexural displacement of the masonry pier in the context of a concrete or timber floor can be computed using the following formulas.

$$u_{fd;c} = \frac{F \cdot h_{\text{pier}}^3}{24 \cdot E \cdot I} \quad (2.9)$$

$$u_{fd;t} = \frac{F \cdot h_{\text{pier}}^3}{3 \cdot E \cdot I} \quad (2.10)$$

where,

- $u_{fd;c}$  represents the flexural displacement specific to a concrete floor scenario, where the piers are fixed at both ends.
- $u_{fd;t}$  represents the flexural displacement specific to a timber floor scenario, where the piers are cantilevered.
- $I$  is the moment of inertia of the pier given by,

$$I = \frac{t_{\text{pier}} \cdot l_{\text{pier}}^3}{12} \quad (2.11)$$

- $F$  is the axial force
- $E$  is the Young's modulus

The shear displacement of the masonry pier can be quantified using the following formula,

$$u_{sd} = \frac{F \cdot h_{\text{pier}}}{G \cdot t_{\text{pier}} \cdot l_{\text{pier}} \cdot 0.87} \quad (2.12)$$

where,

- $u_{sd}$  represents the shear displacement of the masonry pier.
- $G$  is the shear modulus
- 0.87 is the safety factor. According to Eurocode 6 [4], The shear modulus is divided by a safety factor of 1.15, to ensure that the structure maintains its integrity and safety under all circumstances.

### 2.5.5 Calculation of Force vs Displacement of Pier capacity

After performing these necessary calculations, the force versus displacement values for the critical failure mechanism per pier can be determined. For constructing the capacity curve or force-displacement curve, we calculate three key points: A, B, and C. Point A represents the transition from elastic to plastic behaviour in the pier. Point B signifies a distinct change in stiffness, indicating significant structural degradation. Point C marks the ultimate capacity of the pier, beyond which it cannot sustain any additional load, see Figure 2.16.

The plot begins with an initial displacement and force at zero, then progresses to the yield displacement corresponding to the shear resistance of the governing failure mechanism. If the governing failure mechanism is rocking, Point B will have a displacement value calculated according to Equation 2.3 and a force value determined by Equation 2.5. Conversely, if the failure mechanism is bed joint sliding, the displacement at Point B will correspond to a drift limit of 0.003, with the force calculated using Equation 2.7. For Point C, if bed joint sliding governs the failure, the displacement will have a drift limit of 0.0075, and the force will be calculated as in Equation 2.7. If rocking governs the failure, the displacement will be calculated according to Equation 2.3, and the force will reduce to zero.

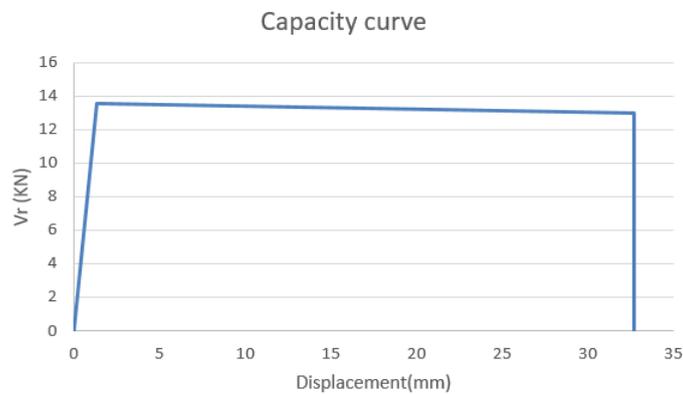


Figure 2.16: Example of Pier Capacity Curve

### 2.5.6 Transition from Pier Capacity to Wall Capacity or Building Capacity

The description and steps outlined above for the SLAMA analysis aim to provide the capacity curve at the pier level. To analyse at the building level, the pier capacity is aggregated to determine the wall capacity in the case of timber floors or the overall building capacity in the case of concrete floors, as specified in Section 2.4. To combine the pier capacities to the wall or building level, piecewise linear interpolation is used.

For each pier within a wall, forces are interpolated based on known displacement-force pairs using linear interpolation, see Figure 2.17. The `interp1d` function from the `scipy.interpolate` library is utilized for this purpose. This function generates a continuous piecewise linear interpolation of forces between known data points and can extrapolate beyond the defined range of displacements as needed. Interpolated forces from all piers are combined to derive the overall force profile of the wall or the building. This is achieved by summing the interpolated forces at a series of uniformly spaced displacement points spanning the maximum observed displacement among all piers. The combined force profiles are evaluated at the three key points A, B and C to determine the wall's or the whole building's capacity to withstand forces under given displacements. Forces at these points are computed by aggregating contributions from all piers using the interpolation functions developed for each.

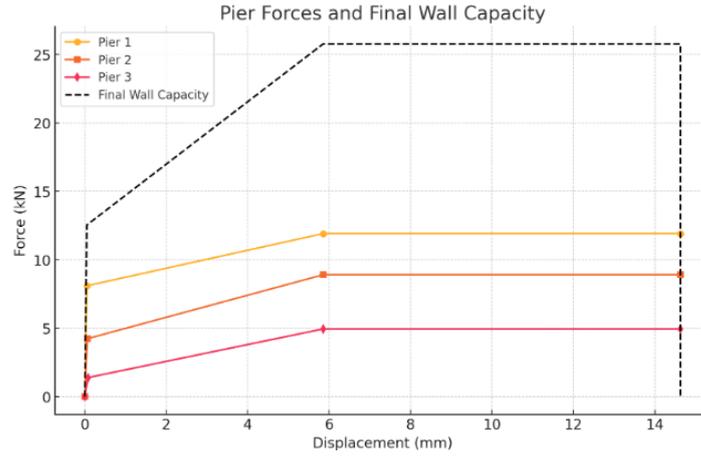


Figure 2.17: Example of combining pier capacity to wall capacity

### 2.5.7 Converting to Sa-Sd Bilinear Curve

Once the wall or building capacity curves are built, represented in terms of Force (kN) versus Displacement (mm), they are then transformed into Spectral Acceleration (Sa) in units of gravity (g) versus Spectral Displacement (Sd) in meters (m) and subsequently bilinearized according to NPR998:2020 [20], section G.4.2, see Figure 2.18.

To perform the conversion to Sa-Sd form, the force values are divided by the effective mass of the structure. In the case of a concrete floor, this effective mass is the sum of the effective masses of all the walls, floors, and roof of the structure. If the structure has a timber floor, the force is divided by the effective mass of the wall. Following this, the result is divided by the acceleration due to gravity ( $9.81m/s^2$ ). The displacement values are multiplied by  $10^{-3}$  to convert to meters.

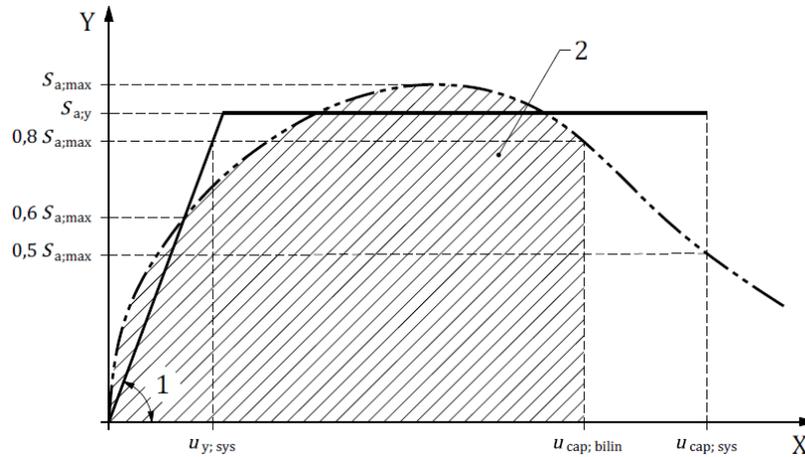


Figure 2.18: Bilinearized pushover curve [20]

where,

- $S_{amax}$  is the maximum spectral acceleration of the curve.
- $u_{cap,sys}$  is the maximum spectral displacement, which is calculated by considering 50% of the  $S_{amax}$ .
- $u_{cap,bilin}$  is the displacement that matches the point when  $S_{amax}$  drops to its 80%
- 1 is  $K_{init}/m_{eff}$
- 2 is  $E_m/m_{eff}$
- $K_{init}$  is the initial lateral stiffness
- $E_m$  deformation energy, calculated as the area under the Sa-Sd curve up to the displacement  $u_{cap,bilin}$ .
- $m_{eff}$  is the effective mass

- $S_y$  represents the spectral yield acceleration, which is determined using Equation 2.13. It is verified to fall within the range of  $S_{amax}$  and 80% of  $S_{amax}$  as shown in Figure 2.18.

$$S_y = (u_{cap,bili} \times K_{init}) - \sqrt{(u_{cap,bili} \times K_{init})^2 - 2 \times E_m \times K_{init}} \quad (2.13)$$

- $u_{y,sys}$  is the yield displacement. It is calculated by equation 2.14;

$$u_{y,sys} = \frac{S_y}{K_{init}} \quad (2.14)$$

These capacity curves can then be placed into location specific ADRS curves, which would tell whether the structure requires reinforcement.

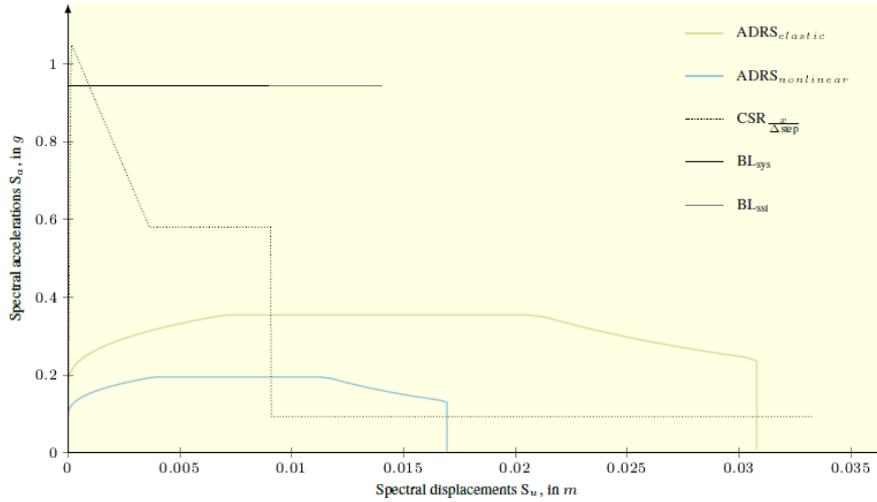


Figure 2.19: A sample ADRS compliance check on a wall from VIIA reports © RHDHV

In the Figure 2.19 the green line depicts the elastic ADRS curve and the blue the nonlinear site specific ADRS curve. The solid black line illustrates the bilinearized capacity curve of the wall which indicates that the wall is unsafe and would require reinforcement, as it falls outside the ADRS curve. The dotted lines represent the non-bilinearized wall capacity.

Once the capacity curves have been bilinearized, the dataset is prepared with the appropriate input and output parameters. These parameters are further detailed in Section 3.2.2, where constructing the surrogate model is discussed comprehensively.

## 2.6 Machine Learning Modelling Techniques

Machine learning can be categorised into supervised and unsupervised machine learning models. In supervised machine learning models the algorithm works with input data that has a labelled output data, essentially providing a correct answer for each input. In unsupervised machine learning the input data does not have a specific output label. The latter algorithm is used to identify patterns and trends in a dataset.

### 2.6.1 Classification Model

The present research makes use of supervised classification models. Classification is a machine-learning technique that predicts the correct labels for a given set of input data. It is a process of categorizing data into predefined categories based on specific characteristics.

Classification techniques are widely used in automated decision-making, pattern recognition, predictive analysis (predicting future outcomes based on existing data), cross-industry applications, automation of routine tasks, and many other areas. The technique can be applied not just to numerical data but to various data types including; categorical data, text data, image data, time series data, audio data and a combination of these data

types. Algorithms commonly used for building classification models include: Random Forest, support vector machine, neural network, Naive Bayes, and others.

There are different types of classification;

Binary Classification: Here the aim is to classify the given input into two categories, see Figure 2.20. For example 0 or 1, true or false, yes or no as the predictions of the model. In the context of this research we make use of the binary classification to predict whether the structure requires reinforcement (represented as 1) or not (represented as 0).

Multi-class Classification: Also known as multinomial classification, a classification technique involves classifying the input data into 3 or more mutually exclusive (classes that don't have anything in common with each other) classes, see Figure 2.20.

Multi-label Classification: The input can belong to multiple classes. For example, a text can contain multiple topics or a picture containing the different elements where each element is a class, see Figure 2.21.

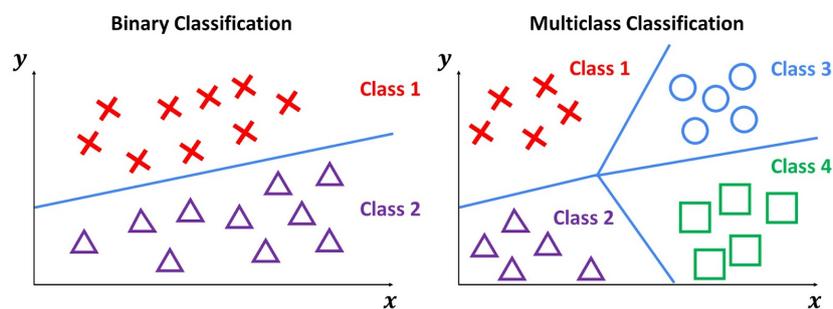


Figure 2.20: Representation of Binary and Multi-class Classification

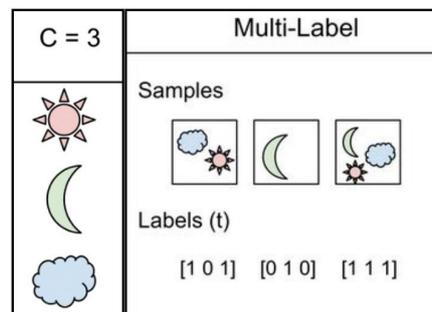


Figure 2.21: Representation of Multi-label Classification

### 2.6.2 Surrogate Model

Surrogate models or metamodels are machine learning models built to approximate the behaviour of complex process. The model provides an effective representation of the system being approximated, providing benefits like: Reduction of physical inputs required, Enhanced understanding of uncertainties associated with various design choices, Identification of the relative importance of different parameters and Insight into parameter behavior within the model.

Being an approximation of a system, there are risks that the model sometimes may not capture the intricate details of the original model, when the model is nonlinear or too complex in nature. The accuracy of the surrogate model also depends on the quality and quantity of data available. The input space for the surrogate model needs to be accurately represented, else if some parts of the original input are not represented, then the model will perform poorly. Though compared to the original process, the surrogate model takes lesser time, finding the right model complexity and architecture can be time consuming.

Popular surrogate modelling approaches are using neural networks, polynomial regression, support vector machines and Gaussian process. It is a data-driven approach, constructed using a limited number of selected input data points from the original process. In the context of this research a surrogate model for the SLaMA analysis is built using deep neural network, as shown in Figure 2.22.

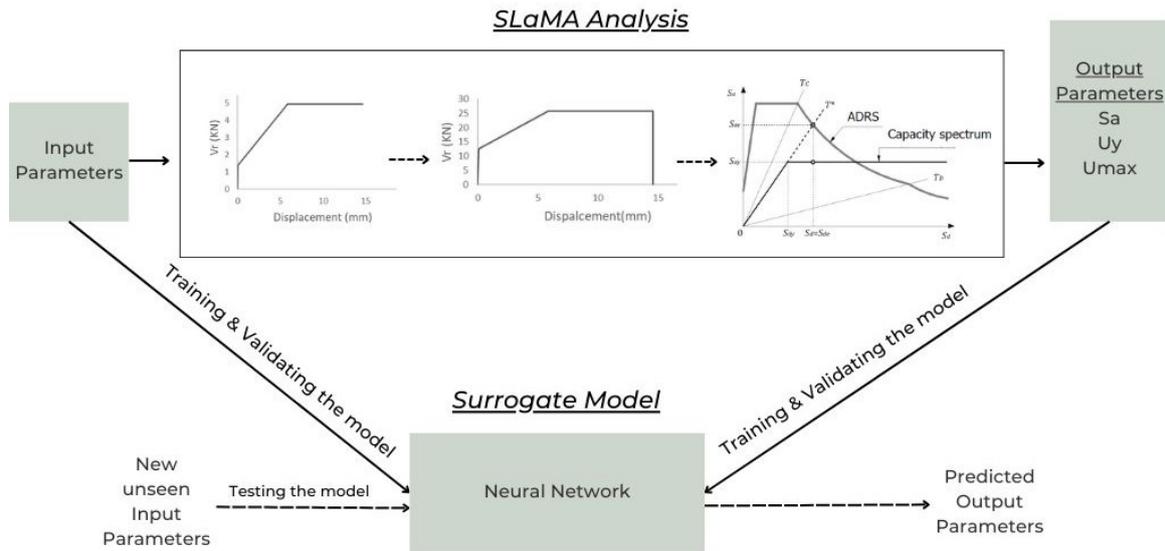


Figure 2.22: SLaMA analysis Vs Surrogate Model

### 2.6.3 Deep Neural Network

To achieve the goal of this research, a Deep Neural Network is used to model the classification and surrogate model. Deep Neural Network is a class of machine learning algorithms that works similar to how neurons in the human brain work. It consists of input layers (with the input parameters), two or more hidden layers and output layers (with the output parameters). Each layer consists of several nodes or neurons connected such that the output signal of one neuron is the input for another, see Figure 2.23 below. Within these hidden layers, the network learns the relationships between the input parameters and the output. This learning process allows the DNN to learn complex patterns and provide accurate predictions.

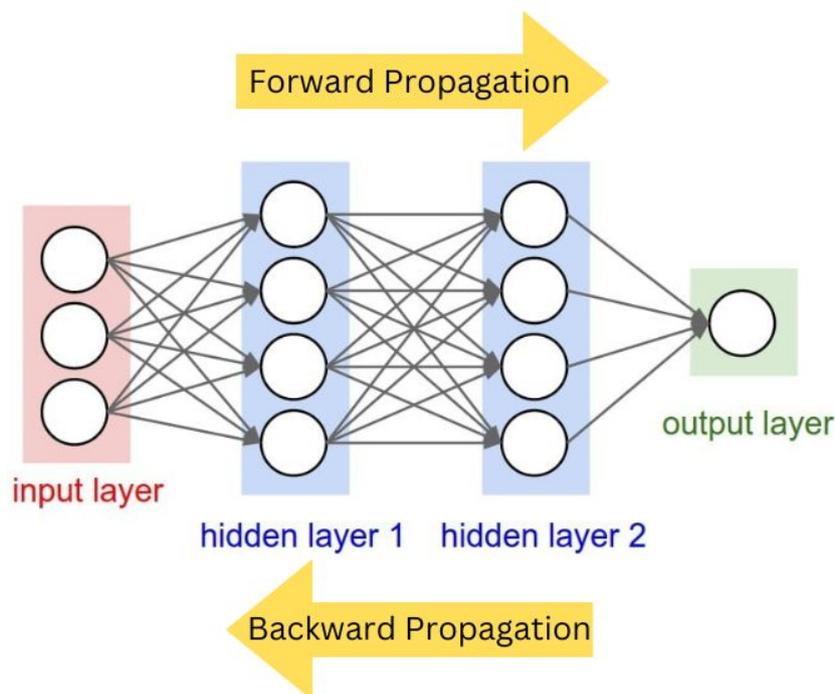


Figure 2.23: Deep Neural Network Model

The process of learning in a DNN starts with forward propagation, see Figure 2.23. During forward propagation, the network processes the input data through its layers, applying specific functions and weights to generate predictions. Once forward propagation produces predictions the network compares these predictions with the actual values to calculate the error. Based on this error, backward propagation is conducted, allowing the

network to learn from its mistakes and adjust its parameters to improve future predictions.

Each connection between neurons in adjacent layers has an associated weight. Weights in a neural network are crucial because they determine the strength and significance of the input features on the output. Initially, the weights are assigned random values. After forward propagation and once the error or loss is computed, the gradient of the loss with respect to the weights can be calculated. This weight gradient tells how much the weights need to be adjusted for the prediction to be closer to the actual value.

To see how the error or loss function is computed, for example in the surrogate model, Mean Square Error (MSE), which is the average of the square of the errors, is used as the loss function. If  $f$  is the loss function, for a single data point, it is given by:

$$f = \frac{1}{n} \cdot (y^* - y)^2 \quad (2.15)$$

where,

- $y^*$  is the predicted value. Consider a situation with a single neuron in the network; the output will be  $y^* = wx + b$ .
- $y$  is the actual value.
- $n$  is the number of data points

After the loss is computed, the loss guides the network in how much it needs to adjust its parameters. So through backward propagation, the gradient of the loss function with respect to the prediction is calculated to see how much the loss function changes in response to changes in the model's predictions:

$$\frac{df}{dy^*} = \frac{2}{n}(y^* - y) \quad (2.16)$$

Then, the gradient of the loss with respect to the weights is calculated, which tells how much each weight in the model should be adjusted in order to minimize the loss:

$$\frac{df}{dw} = \frac{df}{dy^*} \cdot \frac{dy^*}{dw} \quad (2.17)$$

Since  $y^* = wx + b$ , the derivative of  $y^*$  with respect to  $w$  is:

$$\frac{dy^*}{dw} = x \quad (2.18)$$

Using the computed gradients the weights can be adjusted by the optimiser. The optimiser updates the weights and biases using these gradients. The goal is to minimize the loss function, so the optimizer adjusts the parameters in a way that should reduce the loss. For the Classification and Surrogate model, Adam optimiser is used. Adam optimization is a stochastic gradient descent method that is based on adaptive estimation of first-order and second-order moments. Adam begins with an initial learning rate which is by default 0.001 in Tensorflow. Here the neural network is built using TensorFlow, an open-source software library for machine learning. The learning rate directly affects how quickly or slowly a model learns during training. For each weight, the optimiser calculates the moving average of the gradient, which is the first moment and the moving average of the squared gradient, which is the second moment. By using these moments, Adam can effectively adjust the learning rate: decreasing it for parameters with large or consistent gradients and increasing it for those with small or inconsistent gradients. This increase and decrease is done using the first and second-moment values. With these moments the weights are updated as shown below;

$$w_{t+1} = w_t - \eta \cdot \frac{m}{\sqrt{v} + \epsilon} \quad (2.19)$$

- $w_t$ : The current value of the weight.
- $\eta$ : The base learning rate (default 0.001 in TensorFlow).
- $m$ : The first moment estimate.
- $v$ : The second moment estimate.
- $\epsilon$ : A small constant added to avoid division by zero.

Combining all the information discussed so far, let's zoom in to observe what happens inside a single neuron of a hidden layer.

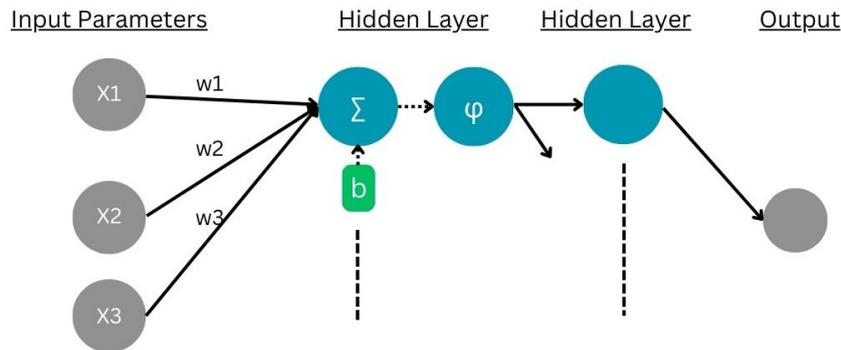


Figure 2.24: Zooming in on a Single Neuron: Visualizing the Process Within Hidden Layers

Observing Figure 2.24 the DNN training process has been described below:

- **Forward propagation:** Input data is passed through the network, focusing on what happens inside one neuron in the first hidden layer, the process begins with calculating the weighted sum of the inputs. Each input is multiplied by its respective weight, and these products are summed together. To this weighted sum, a bias term is added, which helps adjust the output and improve the model's flexibility. All of this is then passed into the activation function, as shown in equation 2.6.3. This function introduces non-linearity to the model, enabling the network to learn and capture complex patterns. Bias allows the activation function to be shifted to the left or right, which can be crucial in helping the network learn the underlying patterns in the data.

$$z = \phi \left( \sum (w_i \cdot x_i) + b \right) \quad (2.20)$$

where  $\phi$  is the activation function,  $w_i$  is the weight,  $b$  is bias and  $x_i$  is the input values. The process is continued through each layer till the output layer.

The activation function transforms the output of each neuron into a non-linear output. Different activation functions map outputs to specific ranges, which can be useful for various tasks. Used within the hidden layers of the network, for both models, ReLU (Rectified linear unit) activation function is used. It maps the outputs to a range of  $[0, \infty)$ . For the Classification model, the output layer uses a sigmoidal function, as it maps the outputs between 0 and 1, which is useful for binary classification. For surrogate models where the results are numerical, the softplus function is used. It is a smooth approximation of the ReLU function and maps the outputs to a range of  $(0, \infty)$  which is appropriate for tasks that require positive numerical outputs.

The output of one neuron serves as the input for the subsequent neuron, continuing through the network layers until predictions are made.

- **Loss Function:** The predictions are then compared to actual values using the loss functions. The function computes the discrepancy between the two values. As described before, the surrogate model makes use of MSE loss function. MSE is typically used in regression tasks where the goal is to predict continuous

values. While classification model uses Binary Cross-Entropy loss function, useful for binary classification, it measures the performance of a classification model whose output is a probability value between 0 and 1.

The equation to calculate the loss using the binary cross-entropy function is:

$$f = -[y \cdot \log(y^*) + (1 - y) \cdot \log(1 - y^*)] \quad (2.21)$$

where,

- $f$  is loss function.
- $y$  is the actual binary label (0 or 1).
- $y^*$  is the predicted probability of the positive class.

Then as discussed before the optimiser calculates the gradient of the loss with respect to the predictions is computed. Then the gradient of loss with respect to weights and bias is computed and the weights and bias are updated using the calculated gradients.

- *Backward propagation*: To minimise the error between the predicted and actual value, the values of weights and biases need to be adjusted, for this reason, backpropagation is done. Here the error obtained through the loss function is sent backwards into the network. Then from each layer it would determine the gradient for the weight and bias. Once the gradients are computed, the parameters in the network are subsequently updated using optimization algorithms.

Detailed explanations of the model selection, training, validating, testing and hyperparameter tuning are provided in Chapters 3 and 4 concerning the model built.

A few key terminologies that will be referenced frequently in the result section of Chapters 3 and 4 are discussed below;

- **Training loss**: It is the metric that tells how well the model learns from the training data.
- **Validation Loss**: This metric measures the model's performance on a validation dataset, which is separate from the training data. Validation loss is used to assess how well the model is likely to perform on unseen data during training. Most of the plots in the following chapters are plotted against validation loss. This is because it provides a clear and reliable way to evaluate and monitor the model's expected performance on new, unseen data.
- **Training Accuracy**: It represents the model's performance in terms of the number of correct predictions in the training process.
- **Validation Accuracy**: It represents the model's performance in terms of the number of correct predictions in the validation process.
- **Epoch**: This refers to the number of complete cycles the model goes through during the training process. If epoch=1, it means the whole dataset is processed once. If epoch=2, it means the model processes the dataset again after refining the weights to improve the predictions. A higher number of epochs generally allows the model to better generalize to unseen data, although too many epochs can lead to overfitting.
- **Model Complexity**: It refers to how intricate the model is depending on the number of input parameters, layers and neurons.
- **Model Architecture**: It refers to the structure of the model, including the arrangement and connections between layers, as well as the choice of activation functions applied within the network.
- **Batch Size**: It is the number of training samples processed in each iteration. It is useful when working with large datasets, as it can be challenging to process the entire dataset at once. Instead, the data is divided into batches, and one batch is processed at a time, updating the model weights based on the errors calculated in each batch.

Combining these concepts, during each iteration or epoch of the model training, the model makes predictions of the training dataset. After making predictions on the training set, it also predicts outcomes on the validation set. Both the predictions are then compared to the actual value. The difference between the actual and predicted data points is given by the loss functions, like mean-square-error (the loss function is different for different problem situations). The training loss is the average of the errors in the training data predictions, while the validation loss is the average of the losses in the validation set. As the training progresses the training loss is expected to reduce, indicating the model is predicting better. Monitoring both training and validation loss is important because if training loss is decreasing but validation loss is increasing, it may indicate overfitting.

## Chapter 3

# Surrogate Model for SLaMA Analysis

To achieve the objectives of this research, a surrogate model for the Simplified Lateral Mechanism Analysis (SLaMA) is proposed in this chapter. SLaMA, as previously discussed, is a computationally cost-efficient analysis method, with lesser accuracy compared to the analysis done through FEM or DEM software's for the prediction of the structural performance during a seismic shock. However, SLaMA's simplicity and the ability to easily generate large volumes of input and output data make it an attractive choice.

SLaMA will facilitate easy cross-verification of newly generated data for errors or inconsistencies. Generating data for computationally expensive analyses would be time-consuming, especially given the uncertainty around how many data points the machine learning model would require to learn effectively and generalize well to unseen data. Because numerical models are complex, they typically demand a larger dataset to achieve good generalization.

A surrogate model based on SLaMA can serve as a baseline to estimate the number of data points the surrogate model of a computationally expensive model would require. This approach can provide insights into the feasibility of developing a surrogate model for more complex, time-consuming, and higher-accuracy methods. The use of SLaMA allows for faster iterations when fine-tuning the surrogate model, making it easier to explore different model complexities and model architectures. This surrogate model for SLaMA can serve as a valuable tool for rapid assessments that meet the demands of early-stage design, contributing to efficient decision-making processes for URM masonry walls.

The surrogate model for SLaMA will predict  $S_a$  and  $S_d$  values at the wall level for timber floors and at the building level for concrete floors. To accommodate these different scenarios, two separate surrogate models are developed: one for timber floors and another for concrete floored buildings.

### 3.1 Determination of input parameter for SLaMA analysis

To perform SLaMA analysis as explained in section 2.5, information on the length, height, thickness, and other specifics of each pier within the walls of each house is required. To obtain the geometry and material data of the existing buildings in Groningen, the VIIA project's relational database is utilised, as explained in section 2.3. The existing reports for SLaMA analysis by VIIA follow an older version of the NPR, NPR 9998:2018 and NPR9998:2018+C1+A1:2020. Here the SLaMA analysis has been performed following the NPR guidelines, as mentioned in section 1.1, version NPR 9998:2020 [20]. Due to the database containing objects subjected to various analyses, including instances of multiple analyses per object, preliminary data cleansing was necessary to ascertain the count of objects that underwent SLaMA analysis.

Filtering the "object\_deel" sheet of the database for SLaMA analysis, designated by code 2, initially yielded 81 object parts. Subsequent filtering to exclude reanalyzed object parts reduced this number to 60. Among these, 29 object parts have recorded engineering data in the "engineering\_database" sheet. Detailed geometric information is essential for conducting SLaMA calculations. Consequently, the object IDs retrieved from the 60 parts will facilitate access to SLaMA reports and object drawings stored on the VIIA drive called MYVIIA.

Detailed information regarding each individual wall is extracted from the reports and drawings. This includes the pier name, height, length, and thickness (in millimetres) of each pier, the material composition of both the floor and walls, as well as the overburden load or axial load applied to each pier (in kilonewtons). Additionally, the name of the flange associated with each pier is noted to consider its effect. The dimensions of the whole walls, including their name, total height, length, and thickness, in meters. The floor materials consist of either timber or concrete, while the wall materials include Clay Brick (pre-1945), Clay Brick (post-1945), and Calcium Silicate (Brickwork)(post-1960). The wall material properties are in accordance with the NPR9998:2020 [20]. With the necessary input parameters in hand the SLaMA analysis was first performed in excel and then in python to generate more data points.

## 3.2 Development of Surrogate Model

### 3.2.1 Data Augmentation for Pier Capacity

The surrogate model requires a substantial number of data points to effectively learn and generalize. Consequently, data from 9 objects were extracted from MYVIIA, and the individual pier data for each object were manually collected. Among these, 7 objects had timber floors, and 2 had concrete floors. This process yielded 507 data points on pier capacities, which provided a general distribution of the dimensions and properties of the piers in Groningen buildings. With the general distribution in hand, we can generate additional data points using sampling techniques. The sampling technique employed here is Latin Hypercube Sampling (LHS). The SLaMA procedure described in Section 2.5 is coded in Python to facilitate the generation and storage of results as additional data becomes available.

Latin Hypercube Sampling (LHS) is a method for generating a well-distributed random sample for different parameters. Initially, the number of sample points is determined, and LHS ensures that the entire range of each variable is systematically explored while minimising the variance of estimated outputs. It divides each parameter range into intervals and selects one sample point from each interval. This ensures that the entire range is systematically covered without gaps. Unlike random sampling, which may miss parts of the range, LHS distributes sample points more uniformly, enhancing the representation of parameter space as shown in Figure 3.1. The sampled values are then shuffled to ensure that the sampling is randomised and not aligned along any particular dimension.

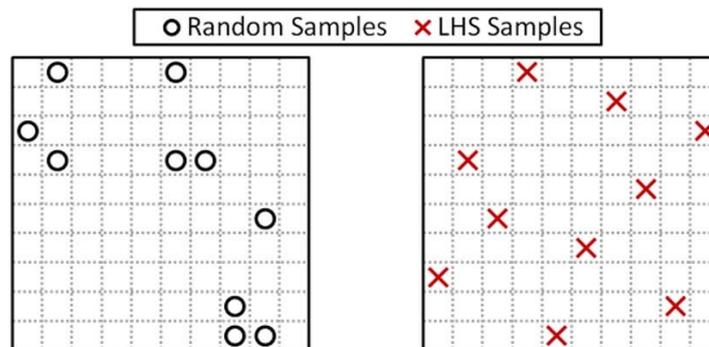


Figure 3.1: Comparison of random and Latin hypercube sampling (LHS)

The objects with timber and concrete floors are sampled separately. The parameters used for sampling include the length, height, thickness, and overburden load of the pier, the length and height of the wall and the height of the building. Since the newly generated data is random, a few constraints are applied to yield more reasonable samples. For both datasets, the newly sampled data is structured such that each new single-storey house is assumed to have 8 walls, with each wall containing 4 piers. The pier lengths and heights are verified to be below the total length and height of the walls, respectively. Additionally, the thickness of the walls is constrained to be either 100 mm or 210 mm. The wall material for each house was assigned randomly, alternating between three types: Clay Brick < 1945, Clay Brick > 1945, and Calcium Silicate > 1960. According to the database, each house has a minimum of 16 flanges. Therefore, for the newly generated houses, 16 flanges are assigned randomly per house, to include the effect of flange. The newly generated data assigns house numbers starting from 1. Each house has walls numbered from 1 to 8, labelled as W1\_house number to W8\_house number. The four piers for each wall are named PW\_wall number.1 to PW\_wall number.4.

Consequently, the 7 objects with timber floors initially had **505** piers and 145 walls, they are now sampled to have **2624** piers and 801 walls. Similarly, the objects with concrete floors initially had **56** piers and 26 walls and are now sampled to have **5604** piers and 1401 walls. The initial sample size was chosen randomly. For timber walls, a sample size of 801 showed good predictive performance. However, applying the same sample size to concrete-floored buildings did not yield accurate predictions, as the validation losses were high. To address this, the data size was increased, which led to an improvement in validation loss. Further increasing the data size beyond this point, however, did not result in any additional improvement.

### 3.2.2 Machine Learning Approach

This section describes the machine-learning techniques employed to develop a surrogate model using neural networks. The surrogate model serves as an efficient approximation of the SLAMA analysis.

#### 3.2.2.1 Features selected

The features required to run the seismic analysis of a building include the length, height, thickness, wall material, overburden load on each pier and identification of the flange pier if any. As explained in section 2.5, the seismic analysis of an entire wall is achieved by combining the individual analysis results of each pier within that wall. This approach accounts for the unique contribution of each pier to the overall seismic response of the wall.

Including all the input parameters will increase the dimensionality of the data. This can make it difficult for the model to learn patterns from such a high-dimensional space, reducing the performance of the model. Also, not all input parameters have a significant impact on the output variable.

In houses with timber floors, the wall capacity determines whether the house requires additional measures. The original SLAMA analysis assesses the capacity of each pier individually and then interpolates these results to determine the overall wall capacity. This means that the input parameters for the original analysis include the dimensions, material properties, and loads acting on each pier. To simplify the process, as the input space of the original analysis is high dimensional, the input parameters were reduced to wall-level parameters, which basically aggregate the effects of individual piers into a more manageable set of inputs. This reduction decreases the computational burden while still retaining sufficient accuracy to represent the original analysis process. While including all pier data in the surrogate model might increase the precision, it would also significantly increase the model's complexity, computational cost and need for more data points. Hence, the following parameters were considered for the surrogate model.

Input Variable	Description
Lwall(m)	Length of the wall in meters
Hwall(m)	Height of the wall in meters
Twall(m)	Thickness of the wall in meters
Wall_material	Clay Brick (pre-1945) or Clay Brick (post-1945) or Calcium Silicate (post-1960)
total_overburden_load(KN)	Sum of the loads acting on the piers of the wall in Kilonewton
total_pier_length(m)	Sum of the lengths of the piers of the wall in meters

*Table 3.1: Input features for the surrogate model of a Timber floor building*

So from the original analysis, the material of the wall is retained since all piers within a wall are made of the same material. Different objects can be made of any of the three materials, each of which affects the failure mechanism and the capacity of the object. The wall thickness is also kept because it remains consistent across all piers of the wall, which is important for accurately calculating the base shear. The lengths of the individual piers in the wall are summed together to give a total pier length. The overburden loads on each pier are combined into a single total load acting on the wall. Finally, the total length and height of the wall are used, simplifying the analysis by treating the wall as a single unit rather than focusing on individual pier dimensions.

In houses with concrete floors, determining the building's capacity helps decide if additional measures are needed. So as mentioned above to simplify the process, the input parameters have been reduced to building-level parameters, aggregating the effects of individual piers into a more manageable set of inputs. Hence the following were selected as the input parameters;

Input Variable	Description
Hbuilding(m)	Height of the building in meters
sum_Lwall(m)	sum of the lengths of the load-bearing walls in meters
Wall_material	Clay Brick (pre-1945) or Clay Brick (post-1945) or Calcium Silicate (post-1960)
total_overburden_load(KN)	Sum of the loads acting on the piers of the structure in Kilonewton
total_pier_length(m)	Sum of the lengths of the piers of the structure in meters
sum_Lwall_area	Sum of the area of the load-bearing walls

Table 3.2: Input features for the surrogate model of a Concrete floor building

From the original analysis, the material of the wall is retained assuming all walls in the house are of the same material. The lengths of the individual piers in the building are summed together to give a total pier length. Combining the loads to get the total load on the building. Instead of analyzing individual wall heights, the Hbuilding parameter was used to represent the total height of the building, focusing on the overall vertical dimension. The lengths of all load-bearing walls were summed to form the sum\_Lwall, capturing the total length of structural walls, while the sum\_Lwall\_area provided a combined measure of the total area of load-bearing walls.

It is important to also note that the reduction in input parameters, such as using the total height of the wall instead of the heights of individual piers, was also done with the goal of creating a surrogate model that could be easily integrated with more computationally expensive analysis methods. Even for computationally expensive models, there is a difference in analyzing timber-floored versus concrete-floored houses, due to differences in material behaviour. As mentioned, SLAMA provides a simplified analytical approach, which, while useful for preliminary assessments, lacks the accuracy of more detailed methods. For analyses like NLPO and NLTH, additional input parameters are crucial to fully capture structural behaviour. These include detailed seismic ground motions, time histories, damping ratios, material-specific details and refined mesh configurations in Finite Element Method (FEM) analyses. The reduced input features from SLAMA can facilitate preliminary evaluations or serve as a basis for more complex models, but fully understanding structural performance under seismic events requires incorporating these additional parameters to ensure accuracy and reliability in the analysis.

The surrogate model requires numerical data inputs, and Wall\_material is categorical. The Wall\_material input is converted into categorical by using one-hot encoding method. One-hot encoding technique turns categorical data into numerical by marking 1 in the presence of that category and 0 otherwise. A snapshot of a sample data representation using One-Hot Encoding is shown in Figure 3.2;

Wall_material_cal sil >1960	Wall_material_clay brick<1945	Wall_material_clay brick>1945	House No
0	1	0	1
1	0	0	2
0	0	1	3
0	1	0	4

Figure 3.2: Sample Data Representation Using One-Hot Encoding

The target variables for the surrogate model are,

Output Variable	Description
$S_y(g)$	Yield spectral displacement in units of g
$u_y(m)$	Yield spectral displacement of the wall in units of meters
$u_{max}(m)$	Maximum spectral displacement of the wall in units of meters

Table 3.3: Input features for the surrogate model of a Timber building

The input variables are assigned to a variable X and the output to a variable y. The timber surrogate model is tested for two different combinations of input parameters. Xa consists of the following parameters: Lwall (m), Hwall (m), Twall (m), total\_pier\_length (m), and total\_overburden (KN). Xb consists of the same parameters as Xa, with an additional parameter Wall\_material.

Similarly, the concrete surrogate model is tested for two different combinations of input parameters. Xa consists

of the following parameters: Hbuilding (m), sum\_Lwall (m), total\_pier\_length (m), and total\_overburden (KN). Xb consists of the same parameters as Xa, with an additional parameter: Wall\_material.

Normalizing or scaling input variables ensures that all input features contribute equally to the modelling process, else features with more variance might dominate the modelling. Additionally, the algorithm converges faster if all the features are on the same scale. Hence using MinMaxScaler from the sklearn.preprocessing module the input features can be normalised.

Once the input variables are scaled, they are then split into 80% of train data and 20% of test data. The train data is further split into 60% train data and 20% validation data. This approach ensures that the model is trained on a substantial portion of the data while also allowing for the evaluation of its performance on unseen data during training.

### 3.2.2.2 Model Architecture and Complexity

To identify the optimal model architecture and hyperparameters for minimizing validation loss, a systematic evaluation of model complexity was conducted. This evaluation focuses on determining the ideal number of layers, neurons per layer, and the appropriate L2 regularization values.

As discussed in Chapter 2, section 2.6.3, the network consists of a number of hidden layers and a number of neurons within those layers which are crucial in determining how effectively the DNN model can learn and generalize from the data, the relation between the input and output parameters. Along with selecting the right number of layers and neurons, it's essential to fine-tune the hyperparameters of the model. Hyperparameters are the parameters of a machine learning model that are set before the training process begins and control the learning process.

One of the most important hyperparameters is the L2 regularization value. L2 regularization adds a penalty to the loss function to discourage overly complex models. By encouraging the model to keep the weights small, it helps prevent the model from assigning too much importance to any single feature. This, reduces the risk of overfitting, which occurs when the model performs exceptionally well on the training data but fails to generalize to unseen data. The L2 regularisation is given to the loss function by  $\lambda \sum w_i^2$ . So new loss function is,

$$f_{\text{new}} = f + \lambda \sum w_i^2 \quad (3.1)$$

where,

- $w_i$  are the weights.
- $\lambda$  is the regularization strength or hyperparameter, controlling how much penalty is applied for weights.

To apply the lambda or regularisation term, the sum of the squared weights is added to the loss function to make sure that lambda grows bigger with larger weights. Squaring the weights makes sure that larger weights get a bigger penalty. This encourages the model to keep the weights small and balanced.

The right choice of L2 regularization is important as a too-high L2 value can make the model too simplistic and does not capture the underlying patterns in the data, while a too-low L2 value can cause overfitting, where the model learns the noise in the training data and fails to generalise to new unseen data.

To assess model complexity, various combinations of hidden layer neuron counts, the number of layers, and L2 regularization values were investigated. The different number of neurons in hidden layers of the model explored were 8, 16, 32, 64 or 128 neurons in each hidden layer. This approach started with a smaller number of neurons and progressively increased them to manage and assess the model's complexity. The different L2 regularisation values explored were 1e-10, 1e-09, 1e-08, 1e-07, 1e-06, 1e-05, 1e-04, 0.001, 0.01, 0.1, 1.0 and 10. The values are spaced exponentially, in logarithmic scale, to explore a broad range from 1e-10 which is almost no regularization, while 10 represents a very strong regularization.

Each combination of the neuron and L2 value is passed through a Sequential model built using TensorFlow. The analysis begins with a single-layer neural network model and systematically progresses to explore configurations with up to four layers. For example, the configurations range from a single layer with 8 neurons ([8])

to four layers each with 8 neurons ([8, 8, 8, 8]).

The DNN process as discussed in Chapter 2, section 2.6.3, is applied to both the training and validation datasets. Throughout all hidden layers, the activation function utilized is ReLU (Rectified Linear Unit), Relu introduces non-linearity in the model and helps the model to learn complex relations. The activation function for the output layer is SoftPlus, which makes sure the results are positive and have a smoother gradient. The loss function is computed using Mean Squared Error (MSE). MSE is commonly used in regression tasks and it computes the average squared difference between predicted and actual values. To optimize the weights during the training the Adam optimizer (Adaptive Moment Estimation) is used [26] and the model is run for 1000 epochs.

To determine the appropriate model complexity, the L2 regularization values were plotted against the validation loss for each layer configuration with different numbers of neurons. This resulted in four separate plots, one for each layer configuration from layer one to layer four. A typical plot is shown in Figure 3.3,

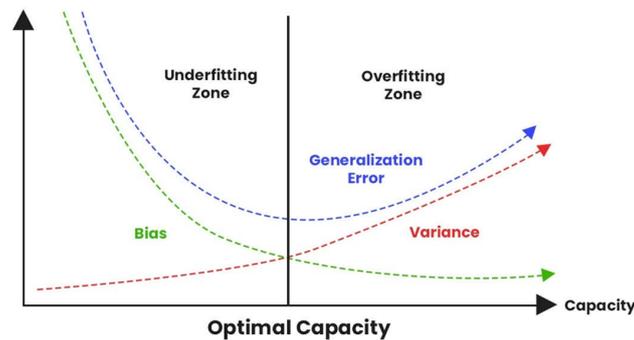


Figure 3.3: Typical plot of L2 regularisation Vs Validation loss

These plots typically exhibit a U-shape. The optimal L2 regularization value is identified as the point just before the curve starts to rise again, indicating overfitting. This point, marked by the black line on the generalization error curve in the graph, represents the balance between regularization and validation loss. The focus when selecting the model will be lower validation loss. This is because a low validation loss indicates that the model's predictions are close to the true values for the validation set. This typically means the model is effective at making accurate predictions and can be expected to perform well on new data. So, the best model configuration to select is the model that achieves a lower validation loss, reflecting a good balance between regularization and model performance.

The process is done for both combinations of input parameters  $X_a$  and  $X_b$ . From the analysis, for each combination, the suitable model architecture is selected and a comparison of the two combinations is done to choose the right input parameters combination.

### 3.2.2.3 Training and Validation of the Model

Following the evaluation of model complexity, for the derived model architecture the loss evaluation curves were generated to observe the progression of both training and validation losses. These Epoch vs. Validation Error curves are essential as they provide insights into how the model learns over epochs, highlighting any potential overfitting or underfitting tendencies. The purpose was to ensure that the model's performance on unseen validation data remains robust and that it generalizes well to new inputs. The graphs below, Figure 3.4 describe overfitting, underfitting and good fit model scenarios.

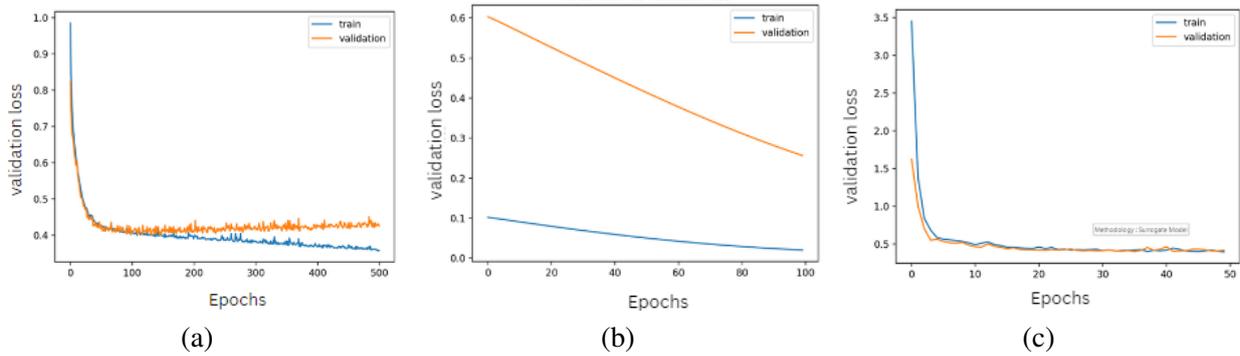


Figure 3.4: (a) Model is Overfitting, (b) Model is Underfitting, and (c) Model is a Good Fit.

From the graph,

**Overfitting:** The model performs exceptionally well on the training data but fails to generalize to the validation data. In the graph, Figure 3.4 (a), this is demonstrated by the training loss plot continuing to decrease steadily with more training, while the validation loss plot decreases to a certain point before beginning to rise again, demonstrating overfitting.

**Underfitting:** The model struggles to capture the underlying patterns in the data, leading to high prediction error on both the training and validation datasets. In the graph, Figure 3.4 (b), underfitting is shown by the training loss remains high and does not decrease, indicating that the model isn't learning effectively. The validation loss curve also shows little to no decrease over time as training progresses. It suggests that the model is too simple to fit the data properly.

**Good Fit:** The model strikes a balance between complexity and generalization. Both training and validation loss curves decrease and stabilize at low values, without significant divergence as shown in Figure 3.4.

After analysing the Epoch vs. Validation Error curves, the learning curves, Training Data Size vs. Validation Error curves, were generated, to analyse the model performance under varying data sizes. This allows to analyse how varying the size of the training dataset affects validation error. To achieve this, various training data sizes were tested, ranging from 10% to 99.99% of the training data, again for the same model architecture and 1000 epochs.

To understand how the size of the training data impacts the model's performance on unseen validation data, different proportions of training data, ranging from 10% to 99.99%, were tested. For each training size, the model was trained and evaluated on the validation dataset, and this process was repeated 20 times to obtain an average validation loss for that training size. This repetition helps to account for variability and ensures that the results are stable and reliable. To smooth out the fluctuations in the validation loss values and highlight the overall trend, the moving average of the validation loss is computed. By plotting the smoothed validation loss against the training data sizes, we can clearly observe how the model's performance changes with different amounts of training data. For generating the moving average a window size of 3 was selected. Window size refers to the number of data points used to calculate each average value in the smoothing process. A typical smoothed learning curve will look like Figure 3.5, which indicates that, if the validation loss decreases as the training size increases, it indicates that the model benefits from more data. More training data allows the model to learn better patterns and generalize well to unseen data.

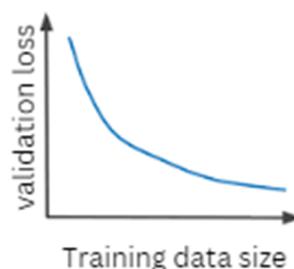


Figure 3.5: Typical learning curve

### 3.2.2.4 Testing of the Model

After training the model, predictions are made on the test data to evaluate the trained model's performance and compare it to the actual values. During testing, forward propagation is used to pass input data through the trained network to generate predictions. There is no backward propagation since the goal is to evaluate the model rather than to train it further. The predictions are made and the losses are computed now to assess the model's performance, not to update the weights or biases.

To assess model performance the evaluation metrics employed include scatter plots, mean absolute error (MAE), mean square error (MSE), and root mean square error (RMSE).

MAE measures the accuracy of the model by measuring the average absolute difference between the predicted values and the actual target values. Here equal weights are provided for all the errors.

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (3.2)$$

where,

- $n$  represents the number of data points
- $y_i$  represents the predicted value
- $\hat{y}_i$  represents the actual value

MSE also known as the L2 norm or the Euclidean distance measures the accuracy of the model, by measuring the average of the squares of the errors, which are the differences between the predicted values and the actual target values. Here more weight is given to larger errors, hence it penalises outliers more severely.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (3.3)$$

where,

- $n$  represents the number of data points
- $y_i$  represents the predicted value
- $\hat{y}_i$  represents the actual value

MAE and MSE are computed using functions from a library called `sklearn.metrics` in Python. RMSE measures the accuracy of the model, by taking the square root of the MSE value. It provides a measure of the spread of the errors.

$$\text{RMSE} = \sqrt{\text{MSE}} \quad (3.4)$$

The scatter plot provides a visual assessment of the model's performance. In this plot, actual values are represented on the x-axis, while predicted values are on the y-axis. A reference line  $y = x$  is included to denote a perfect model, where predicted values exactly match the actual values. Points that lie closer to this reference line indicate higher prediction accuracy, whereas points farther from the line indicate larger discrepancies between predicted and actual values.

### 3.3 Results and Discussion

#### 3.3.1 Timber floor structures

As specified in section 3.2.2.2, the results for finding the optimal model architecture and complexity by varying the number of neurons, layers and L2 regularization values for 1000 epochs for input variables combination Xa is given below;

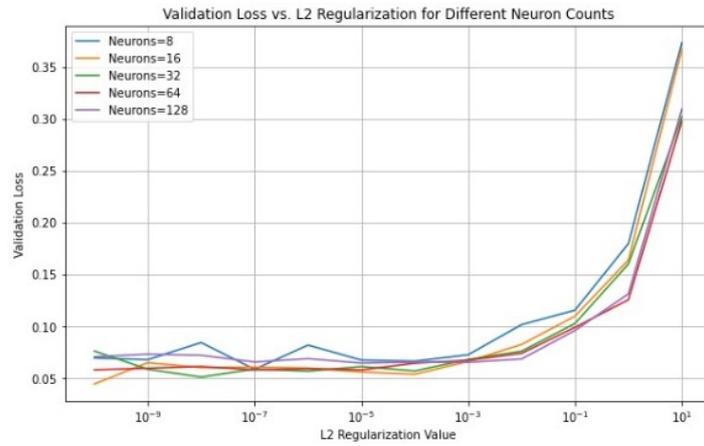


Figure 3.6: Validation loss vs L2 Regularization for 1 layer with different neurons for Xa input combination for buildings with timber floor

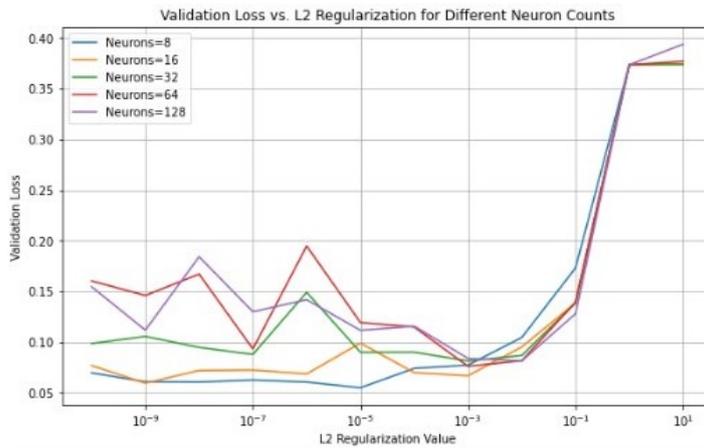


Figure 3.7: Validation loss vs L2 Regularization for 2 layer with different neurons for Xa input combination for buildings with timber floor

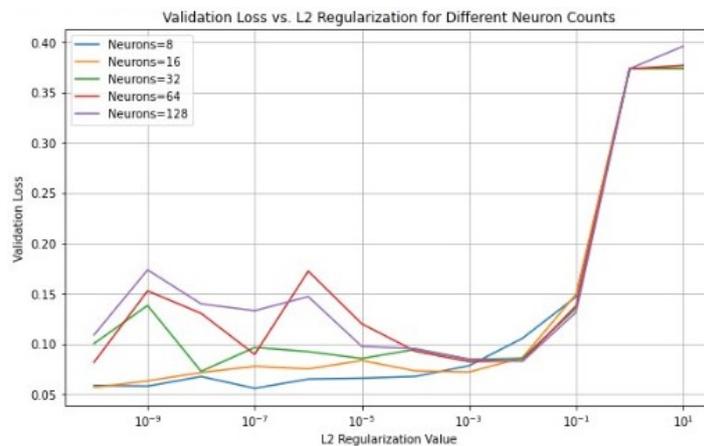


Figure 3.8: Validation loss vs L2 Regularization for 3 layer with different neurons for Xa input combination for buildings with timber floor

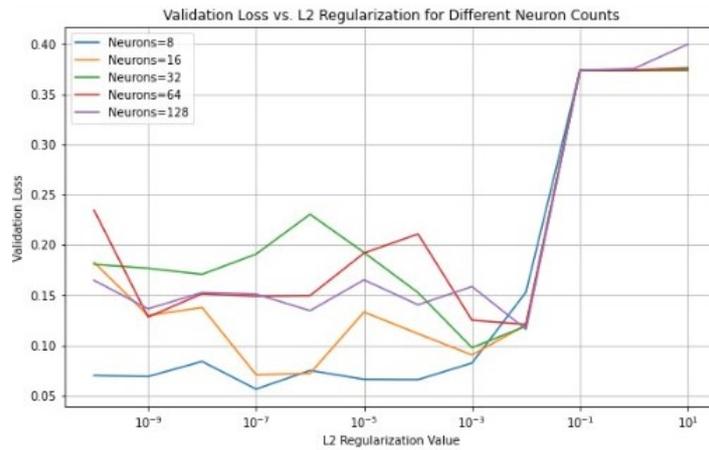


Figure 3.9: Validation loss vs L2 Regularization for 4 layer with different neurons for Xa input combination for buildings with timber floor

Close observation of the results for different neurons and L2 regularisation values in Layer 1 (Figure 3.6), Layer 2 (Figure 3.7), Layer 3 (Figure 3.8) and Layer 4 (Figure 3.9), analysing them as described in section 3.2.2.2, primarily focusing on the validation loss values, led to the selection of a single layer with 64 neurons as the model architecture. The L2 regularisation value is 0.0001, which is the point in the graph, Figure 3.10, just before the model starts to overfit.

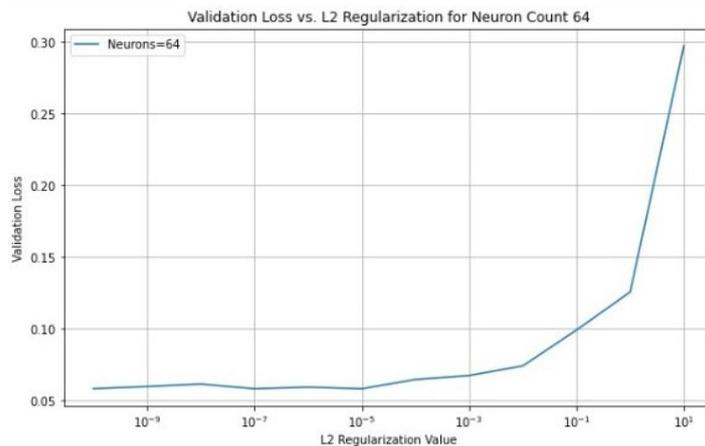


Figure 3.10: Validation loss vs L2 Regularization for 1 layer with 64 neurons for Xa input combination for buildings with timber floor

This configuration appears to strike an optimal balance between model capacity and regularization, effectively capturing the complexity of the data. While increasing the number of layers resulted in curves whose validation losses were consistently higher than that of the single-layer 64-neuron model. It's important to note that these validation losses were computed at the point where the values began to increase, indicating the onset of potential overfitting. So a model architecture of 1 layer with 64 neurons and L2 regularization of 0.0001 is selected. This choice provides a strong indicator that the model will generalize well to unseen data without overfitting, striking a balance between performance and robustness.

Similarly, the Validation loss vs L2 Regularization for different neurons was found for Xb input variable combination. The results were unsatisfactory because none of the configurations produced a smooth, U-shaped curve. Typically, a well-learning model will exhibit such a U-shape in its validation loss curve, indicating effective generalization. In contrast, the curves in this case were highly irregular and zigzagged, suggesting that the models were not learning effectively and did not generalize well to the data. To address this issue and improve the model's learning capacity, we decided to increase the dataset size to 1521 walls. This increase in the training dataset was implemented with the aim of providing the model with more diverse examples to learn from. The following are the results with the increased data points;

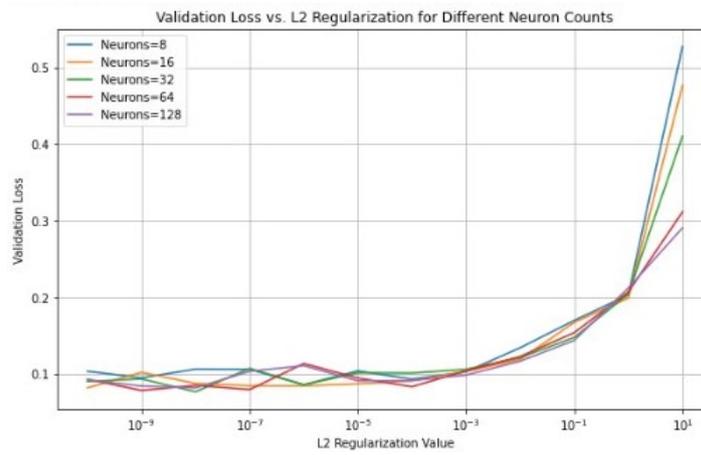


Figure 3.11: Validation loss vs L2 Regularization for 1 layer with different neurons for Xb input combination for buildings with timber floor

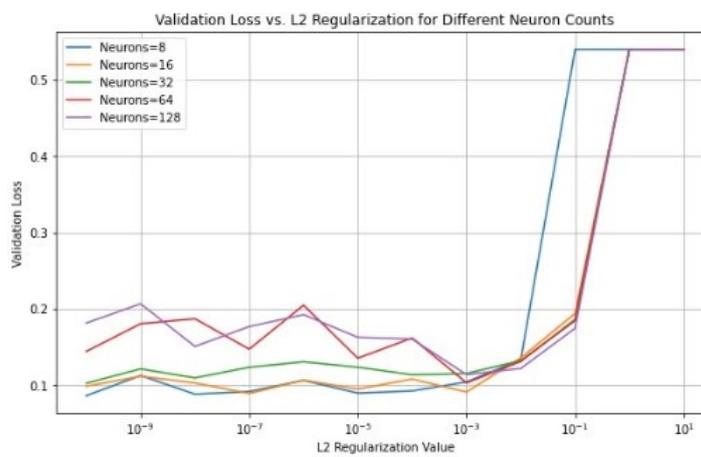


Figure 3.12: Validation loss vs L2 Regularization for 2 layer with different neurons for Xb input combination for buildings with timber floor

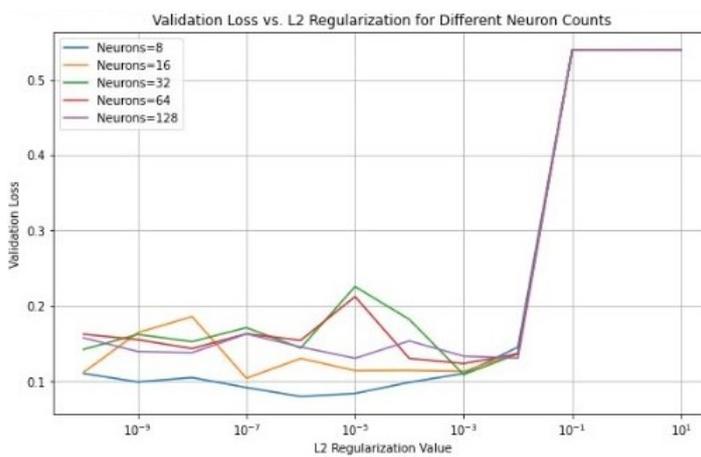


Figure 3.13: Validation loss vs L2 Regularization for 3 layer with different neurons for Xb input combination for buildings with timber floor

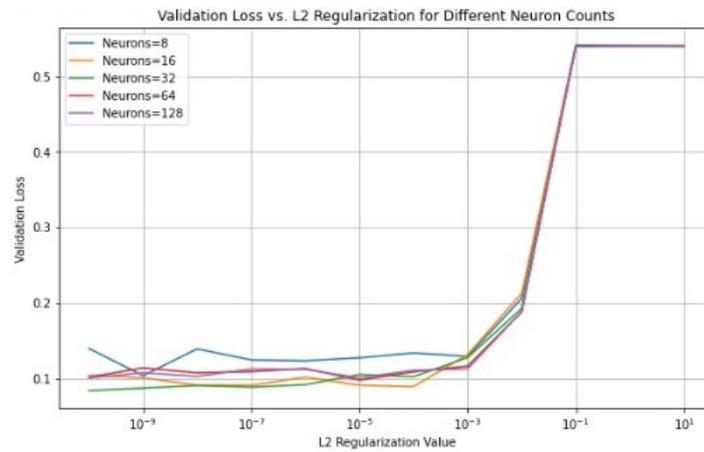


Figure 3.14: Validation loss vs L2 Regularization for 4 layer with different neurons for Xb input combination for buildings with timber floor

Observation of the results, for different neurons and L2 regularisation values in Layer 1 (Figure 3.11), Layer 2 (Figure 3.12), Layer 3 (Figure 3.13) and Layer 4 (Figure 3.14), analysing them as described in section 3.2.2.2, selection done as described in section 3.2.2.2, primarily focusing on the validation loss values, led to the selection of two layers with 8 neurons and L2 regularization value of 0.001 for our model architecture, which is the point in the Figure 3.15 just before overfitting. In comparison to other layer configurations and neuron counts, this particular model had a lower validation error, indicating it generalised better to the given data.

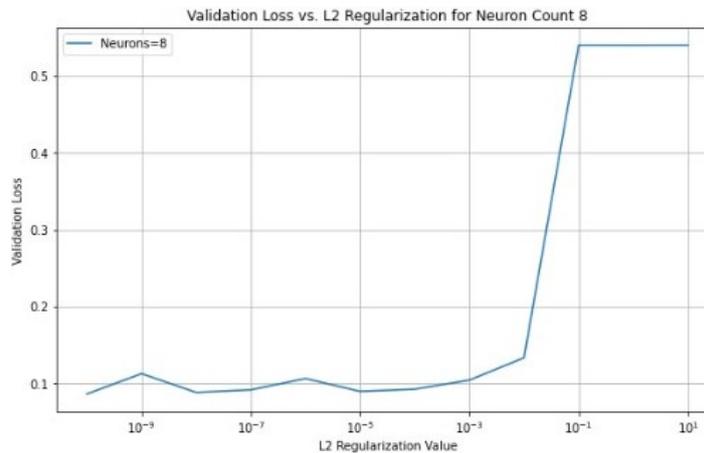


Figure 3.15: Validation loss vs L2 Regularization for 2 layer with 8 neurons for Xb input combination for buildings with timber floor

After the model architecture is set for both Xa and Xb the model is set to be trained and validated. From the graph below the difference between the validation and training loss is 0.0036 for Xa input and 0.0076 for Xb input. The Epoch vs. Validation loss curve for both input combinations is similar to the good fit model described in Figure 3.4. The minimal gap between training and validation loss curves suggests that the model performs consistently well on both the training data and unseen validation data. This indicates strong generalization capabilities.

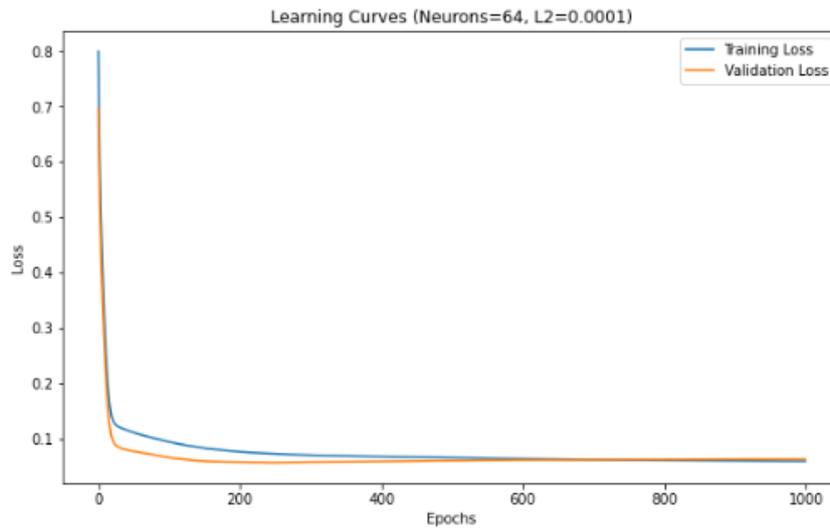


Figure 3.16: Epoch vs. Validation loss of the trained model with Xa input for buildings with timber floor

Epoch vs. Validation loss curve for Xa input combination shown in Figure 3.16. The model has a single layer of 64 neurons with L2 regularisation of 0.0001. Its validation loss is 0.063 and training loss is 0.059.

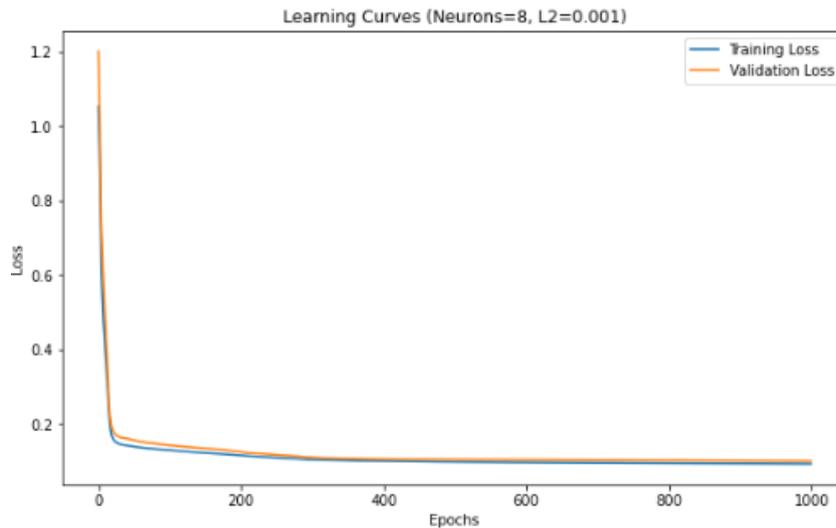


Figure 3.17: Epoch vs. Validation loss of the trained model with Xb input for buildings with timber floor

Epoch vs. Validation loss curve for Xb input combination, shown in Figure ???. The model has two layers of 8 neurons with L2 regularisation of 0.001. Its validation loss is 0.099 and training loss is 0.091.

The learning curves for both models were generated, as shown below.

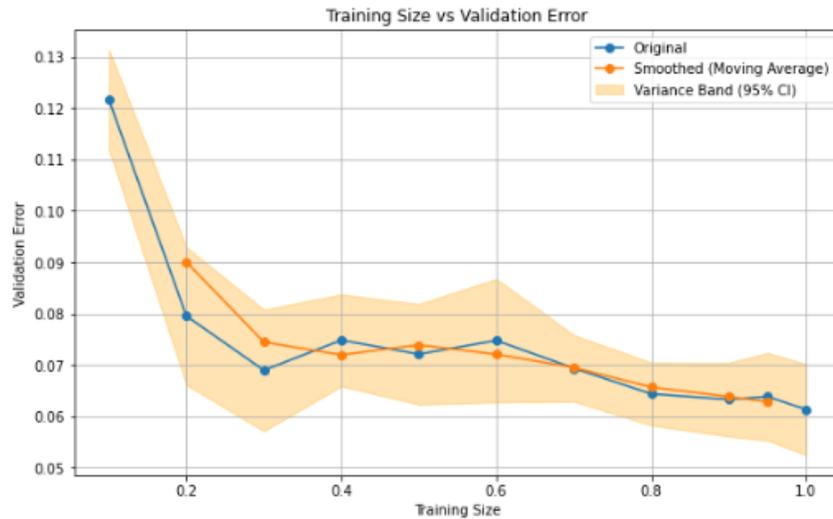


Figure 3.18: Train Size vs. Validation loss of the model with Xa input for buildings with timber floor

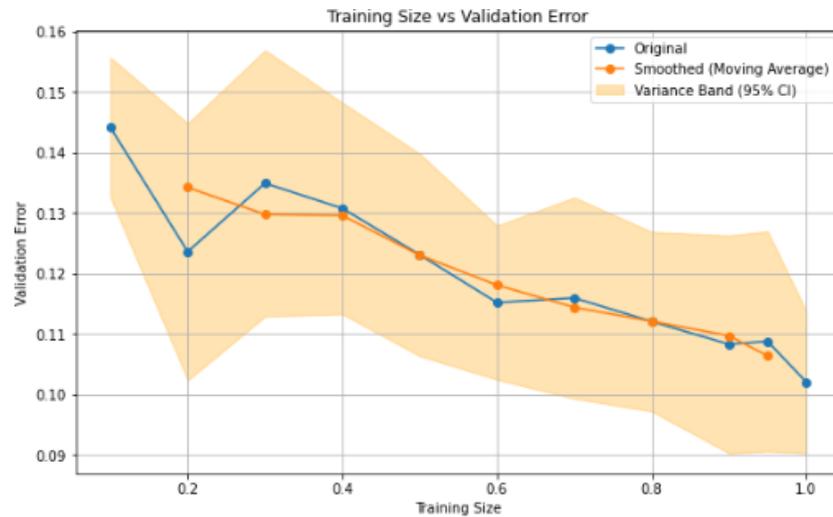


Figure 3.19: Train Size vs. Validation loss of the model with Xb input for buildings with timber floor

The orange shaded region represents the 95% confidence interval (CI) around the validation error curve. This interval provides a range within which the expected true validation error falls 95% of the time. The confidence interval gives a visual representation of the uncertainty or variability in the error estimates. The wider the shaded area, the greater the uncertainty in the model's performance. A narrow confidence interval suggests that the model's predictions are more consistent and reliable across different training runs. The 95% confidence interval is calculated for each training size using the mean validation error and its standard deviation across multiple runs. The interval is computed as the mean  $\pm 1.96$  times the standard deviation. This calculation creates lower and upper bounds for the confidence interval.

Observing the smoothed moving average curve of both models show a decrease in validation loss with increase in training sizes. But model 1, Figure 3.18 shows better performance. This is because the linear decline seen in model 2, Figure 3.19 suggests that this model is still learning, that it benefiting from additional data and requires more data. This can also be seen from the 95% CI, the wider CI represents uncertainty in the model's performance due to the lack of data points for the model to learn and produce more stable predictions. This is the curse of dimensionality, with the increase in the input space the data points become sparse and make it difficult for the model to learn patterns without the addition of more data to fill the space formed due to the expansion of the feature space.

For model 1, the significant drop in validation loss as the training size increases shows that the model is learning effectively from the given data. Also the 95% CI much narrower in model 1 than in model 2, indicating lesser variability in the models predictions. Hence, model 1's smooth decreasing curve with lower overall loss values indicates it's learning more effectively from the data. Therefore, model 1 is selected.

As mentioned before, to evaluate the selected model's predictive performance a scatter plot is generated. The plot aims to show how well the actual and predicted results align. Here the test data was utilised, which consists of data that the model has not encountered during its training and validation phases. This approach provides a crucial evaluation of the model's ability to generalize to entirely new, unseen data.

The model predicts three different output variables,  $S_{y(g)}$ ,  $u_{y(m)}$ , and  $u_{max(m)}$ , it is essential to assess the model's performance for each output independently. To do this, three separate scatter plots are generated, comparing the actual vs. predicted values for each output. In the graphs the blue dots are the predicted values and the red dashed line represents where the predicted values would perfectly match the actual values.

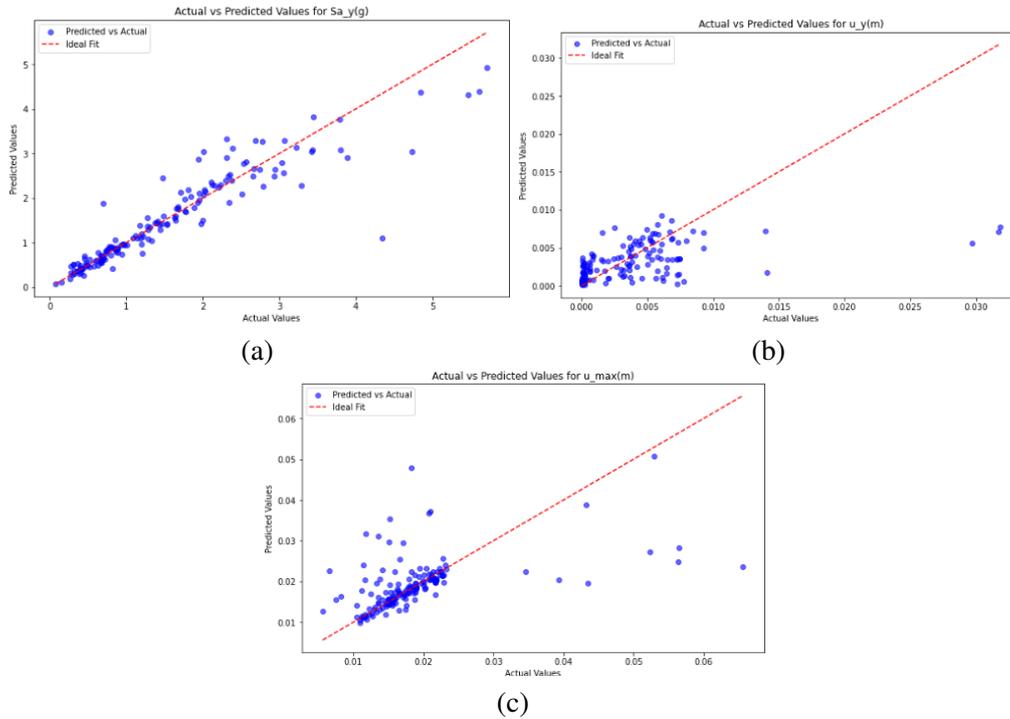


Figure 3.20: Scatter plot for Actual vs. Predicted values of output: (a)  $S_y$ , (b)  $u_y$ , and (c)  $u_{max}$ .

The outputs for  $u_y$ ,  $u_{max}$  and  $S_y$  are in the range of  $4.325E-06$  to  $0.01368$ ,  $0.000406$  to  $0.07$  and  $0.065$  to  $6.012$ . The evaluation metrics for them are;

Output Variable	MAE	MSE	RMSE
$S_y$	0.25377	0.2293	0.4789
$u_y$	0.002124	$1.77169e-05$	0.004209
$u_{max}$	0.004508	$6.428515e-05$	0.0080178

Table 3.4: MAE, MSE, RMSE values for buildings with timber model

The scatter plot for  $S_y$  shows that as the plot progresses the discrepancy is seen to increase indicating for a wider range the predictions are less accurate for  $S_y$ . The MAE value suggests that the model's predictions are off by approximately 0.25377 units from the actual values. Compared to the range of  $S_y$  the MAE suggests a moderate level of error. MSE squares the errors before averaging, giving more weights to larger errors and here MSE is 0.2293 which is low, suggesting that the model's predictions are generally close to the actual values, with few large errors. RMSE provides an error metric in the same units as the data and here RMSE value of 0.4789 is also relatively low compared to the data range. Therefore, overall the model performs well without any extreme outliers, which means the model is not making large, unexpected errors for data points.

Similarly, for  $u_y$  and  $u_{max}$ , the scatter plot shows that most of the points are near the actual value with a few discrepancies. For  $u_y$ , The MAE value suggests that the model's predictions are off by approximately 0.002124 units from the actual values. An RMSE of 0.004209 shows the typical magnitude of the prediction errors. Compared to the data range these values are small, indicating that large errors are uncommon. For  $u_{max}$ , MAE is 0.004508, MSE is  $6.428515e-05$ , and RMSE is 0.0080178. Given the range of actual values,

these metrics values are low, suggesting that the model’s predictions are generally accurate with small errors in prediction.

These discrepancies observed are due to the input parameters utilised because they might not fully capture the complex relationships in the data. Despite testing various model architectures with different neuron counts, layers, and L2 regularization, and training the best-performing model for 1000 epochs, some discrepancies persist. This suggests that the current set of input features may not fully capture all the complex relationships within the data. For a surrogate model, the input space needs to be the same as the original analysis procedure. However, due to the increased dimension of the input space, the input parameters were reduced to a few specific variables focusing at the wall level. This reduction in the input space may have hindered the surrogate model’s ability to capture the true relationships between the input and output parameters, leading to prediction errors. The reduction was necessary for computational efficiency and model simplicity but accounted for the observed deviations in some predictions. The reduction was also important because of the curse of dimensionality. Curse of dimensionality means, as the number of features or dimensions in a dataset increases, the amount of data needed to generalize accurately grows exponentially. In high dimensions more data is required and the training time increases along with the chances of overfitting. Hence reducing the input spaces becomes important.

Therefore, the results here show that there is room to improve the model by considering more input parameters, so the model can be a better approximation of the original analysis. This is easy for analysis like SLaMA where with the addition of an input parameter it can easily generate more data points. However, it will be difficult for computationally expensive models to generate more data with the addition of input features. To reduce the number of time-consuming analyses required to generate additional data, principal component analysis (PCA) can be used. PCA reduces the dimensionality of the input space while still preserving the essential information in the dataset. Based on the results of this analysis, another suggestion for computationally expensive models would be to maintain the original input space and apply PCA to improve efficiency.

### 3.3.2 Concrete floor structures

As specified in section 3.2.2.2, the results for finding the optimal model architecture and complexity by varying the number of neurons, layers and L2 regularization values for 1000 epochs for input variables combination Xa is given below;

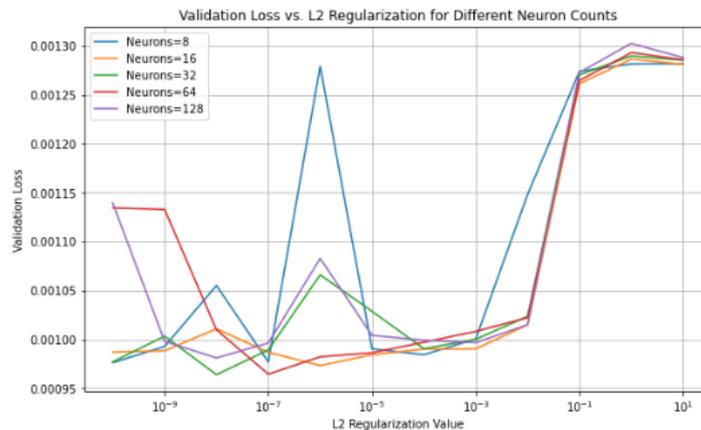


Figure 3.21: Validation loss vs L2 Regularization for 1 layer with different neurons for Xa input combination for buildings with concrete floor

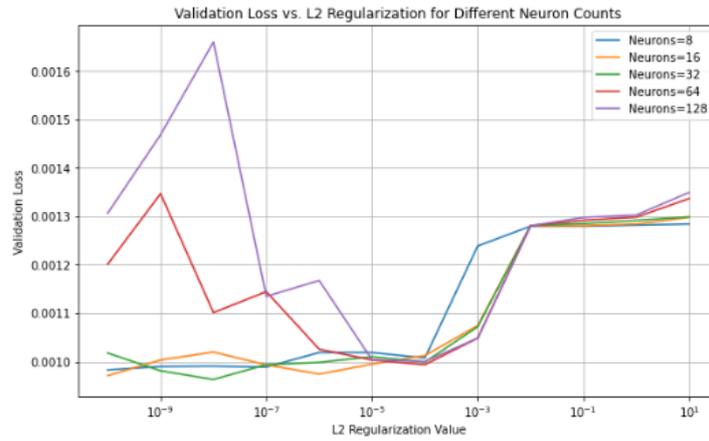


Figure 3.22: Validation loss vs L2 Regularization for 2 layer with different neurons for  $X_a$  input combination for buildings with concrete floor

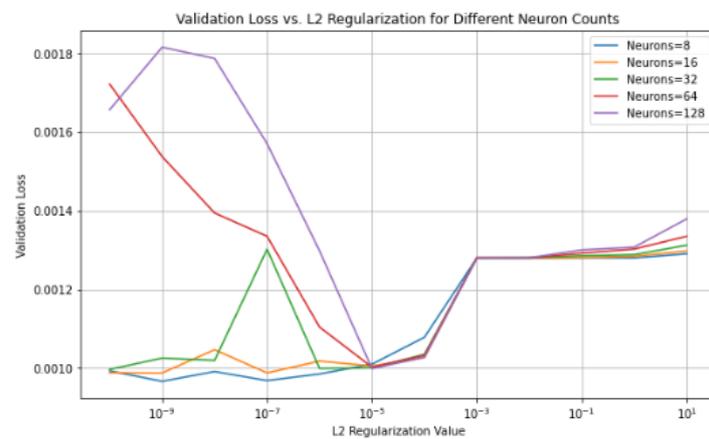


Figure 3.23: Validation loss vs L2 Regularization for 3 layer with different neurons for  $X_a$  input combination for buildings with concrete floor

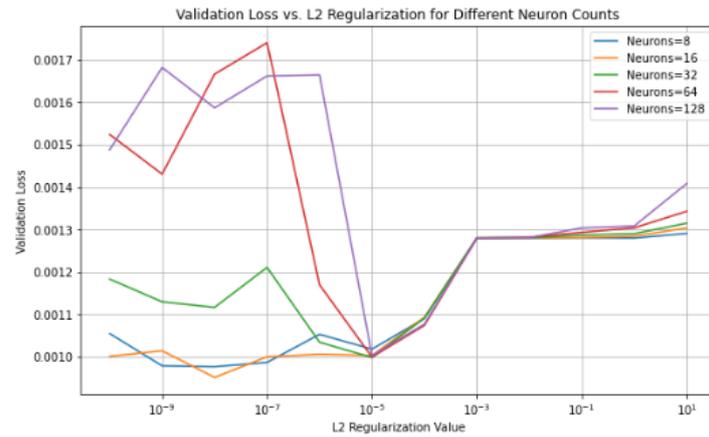


Figure 3.24: Validation loss vs L2 Regularization for 4 layer with different neurons for  $X_a$  input combination for buildings with concrete floor

Close observation of the results for different neurons and L2 regularisation values in Layer 1 (Figure 3.21), Layer 2 (Figure 3.22), Layer 3 (Figure 3.23) and Layer 4 (Figure 3.24), analysing them as described in section 3.2.2.2, primarily focusing on the validation loss values, led to the selection of a two layer with 64 neurons as the model architecture, with L2 regularisation value of 0.0001, which is the point in the graph, Figure 3.25, just before the model starts to overfit.

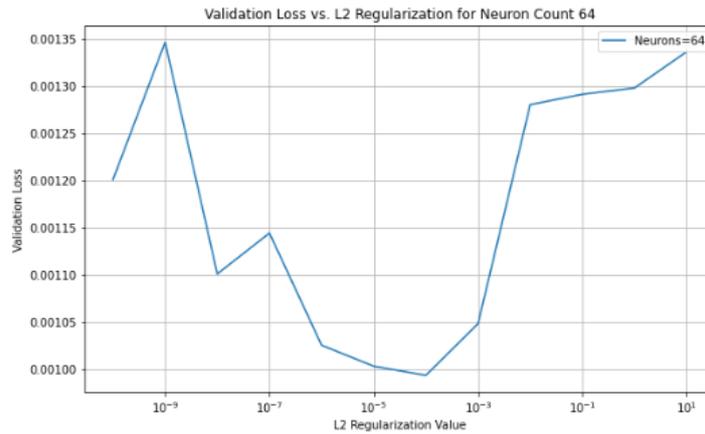


Figure 3.25: Validation loss vs L2 Regularization for 2 layer with 64 neurons for Xa input combination for buildings with concrete floor

Compared to the other layers, this model exhibited a more gradual U-shaped curve when transitioning from underfitting to overfitting. This indicates a smoother progression in model performance. Further increasing the neurons and number of layers did not show a smoother curve or lower validation loss. Hence, the model architecture selected for Xa input combination is two layers of 64 neurons with 0.0001 L2 regularisation.

Similarly, the Validation loss vs L2 Regularization for different neurons and layers was found for the Xb input variable combination. Observing the different layers and neurons, a model architecture of three layers of 64 neurons and an L2 regularisation of 0.0001 was selected, again, the point in the graph, Figure 3.26, just before the model starts to overfit.

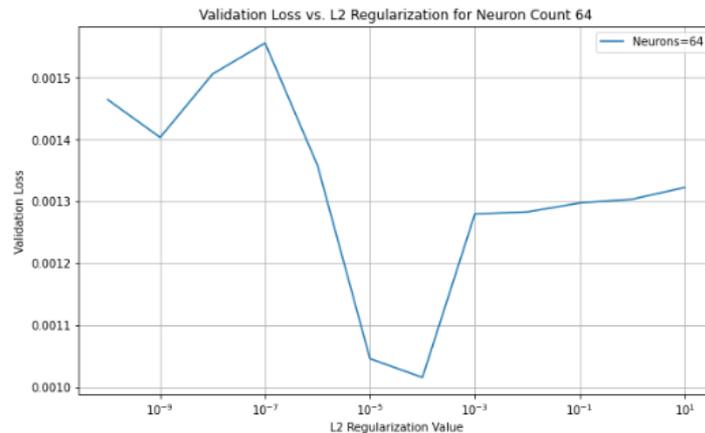


Figure 3.26: Validation loss vs L2 Regularization for 3 layer with 64 neurons for Xa input combination for buildings with concrete floor

After the model architecture for both the input combination are set, the models are trained and validated. From the graph below the difference between the validation and train loss for Xa input combination is 6.1465e-05 and 5.67e-05 for Xb combination. The Epoch vs. Validation loss curve for both input combinations is similar to the good fit model described in figure 3.4. The minimal gap indicated the model is performing well, its performs well on train and test data.

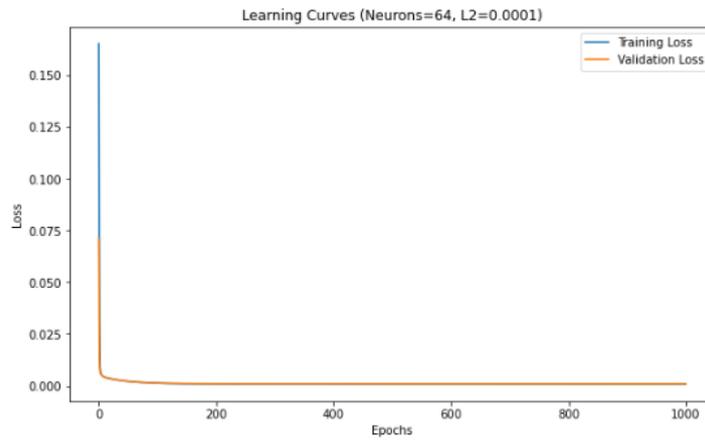


Figure 3.27: Epoch vs. Validation loss of the trained model with Xa input for buildings with concrete floor

Epoch vs. Validation loss curve for Xa input combination shown in Figure 3.27. The model has a two layers of 64 neurons with L2 regularisation of 0.0001. Its validation loss is 0.00099 and training loss is 0.00093.

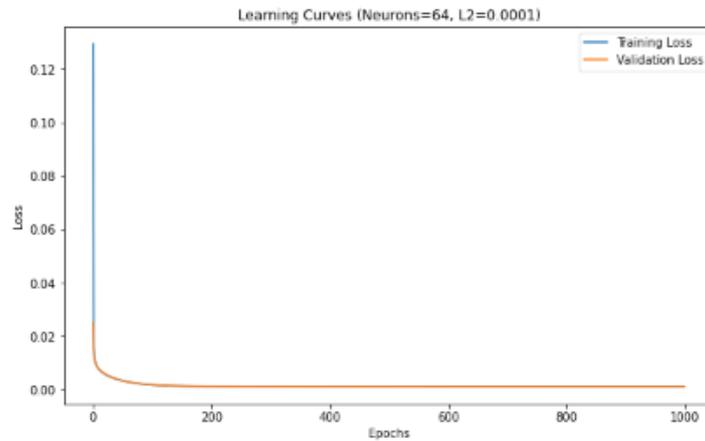


Figure 3.28: Epoch vs. Validation loss of the trained model with Xb input for buildings with concrete floor

Epoch vs. Validation loss curve for Xb input combination shown in Figure 3.28. The model has a three layers of 64 neurons with L2 regularisation of 0.0001. Its validation loss is 0.001 and training loss is 0.00096.

The learning curves for both models were generated, as shown below.

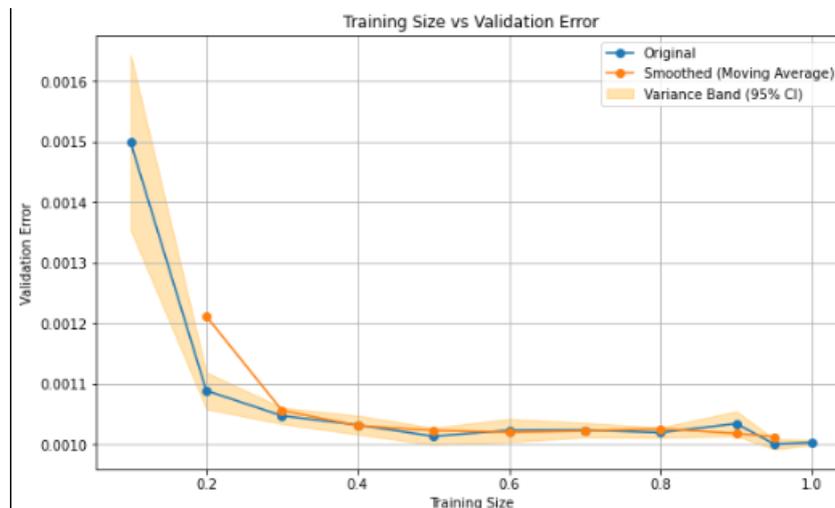


Figure 3.29: Train Size vs. Validation loss of the model with Xa input combination, for buildings with concrete floor

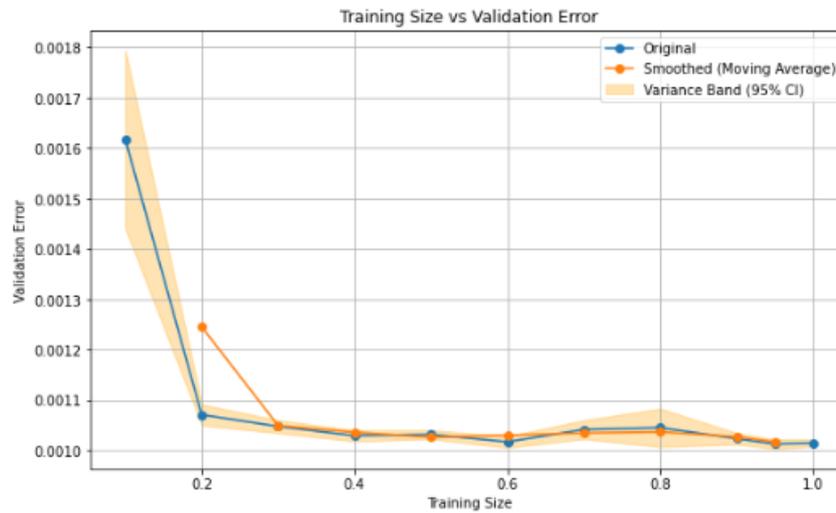


Figure 3.30: Train Size vs. Validation loss of the model with  $X_b$  input combination, for buildings with concrete floor

The orange shaded region represents the 95% confidence interval (CI) around the validation error curve, as explained in section 3.3.1. Observing the smoothed moving average curve of both models shows a decrease in validation loss with an increase in training sizes. Both models have smooth decreasing curves and the validation losses are also similar. Observing the 95% CI, both models have a narrow band, indicating that the validation error estimates are quite precise, with little variability. Both models show similar performance, so here model 1, in Figure 3.29, is selected over model 2, in Figure 3.30, because its more simple as it has lesser input features.

The scatter plots shows how well the model predicts to unseen data. The plot aims to show how well the predicted data matches the actual values. In the graphs the blue dots are the predicted values and the red dashed line represents where the predicted values would perfectly match the actual values.

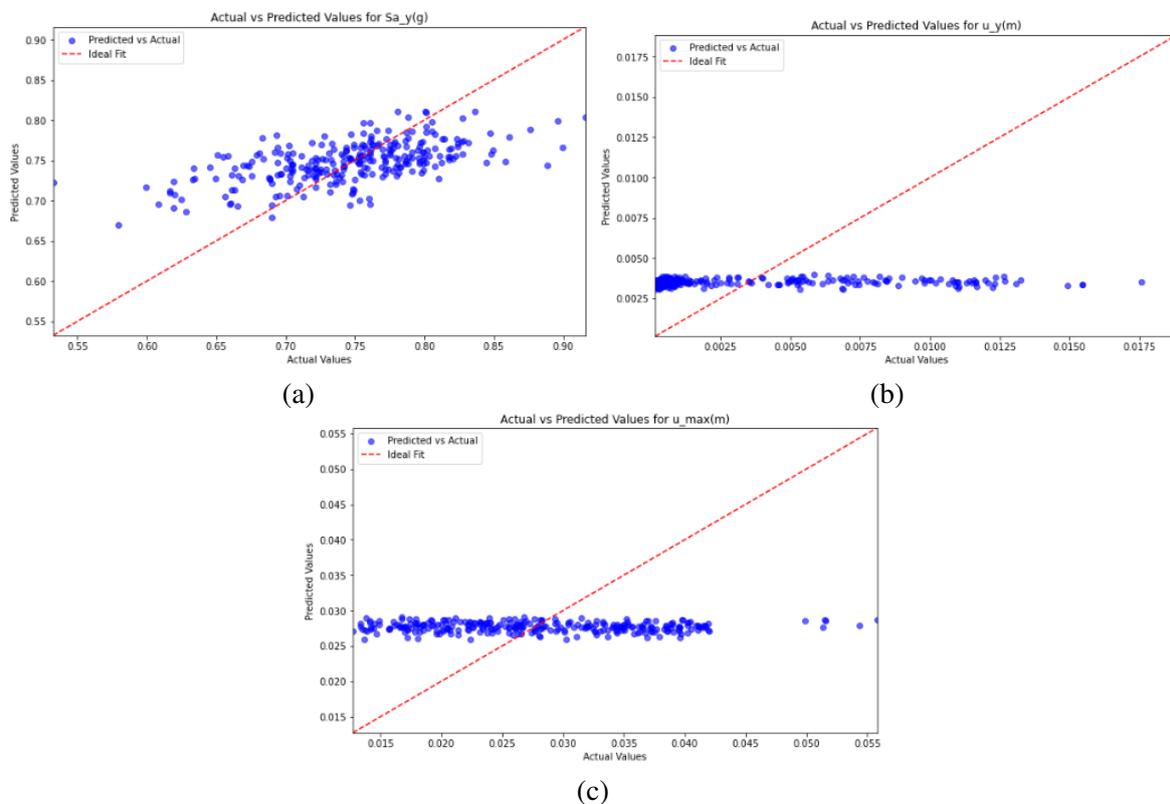


Figure 3.31: Scatter plot for Actual vs. Predicted values of output: (a)  $S_y$ , (b)  $u_y$ , and (c)  $u_{max}$  for concrete floor building.

The outputs for  $u_y$ ,  $u_{max}$  and  $S_y$  are in the range of  $8.84e-05$  to  $0.018$ ,  $0.000975$  to  $0.07$  and  $0.528$  to  $1.05$ . Observing the range of the outputs and the scatter plots, the data are not spread throughout the whole range. The clustering observed for the three outputs shows that the model is not capturing the underlying relationship

between the input and target variable. The model's performance is also evaluated using the MAE, MSE and RMSE metrics. The evaluation metrics for them are;

Output Variable	MAE	MSE	RMSE
S_y	0.0378	0.0023	0.048
u_y	0.00345	1.69e-05	0.0041
u_max	0.0076	8.023e-05	0.0089

Table 3.5: MAE, MSE, RMSE values for buildings with concrete model

For the three output parameters, the low error metrics suggest that the model is performing well on average, but the cluster suggests that it might not be generalising well with the whole range.

The horizontal lined clusters, specifically for  $u_y$ , Figure 3.31 (b) and  $u_{max}$ , Figure 3.31 (c) are due to the inability of the reduced input features to provide any meaningful pattern for the model to learn. When reducing the input space, some key features that define the true relationship between the input and target variables may have been omitted. Without these essential features, the model is unable to identify meaningful patterns in the data. As a result, it either picks up random fluctuations or fails to learn any pattern at all. This concern, that the current input features may not be sufficiently informative to differentiate between different output values effectively needs to be addressed. To test this, an additional input parameter,  $sum\_Lwall\_area$ , was introduced within the  $X_b$  input combination. So the new combination,  $X_c$ , tested for finding the optimal model architecture and complexity by varying the number of neurons, layers and L2 regularization values for 1000 epoch:

$X_c = ['Hbuilding(m)', 'sum\_Lwall(m)', 'total\_pier\_length(m)', 'total\_overburden(KN)', 'Wall\_material', 'sum\_Lwall\_area']$

The results are as follows:

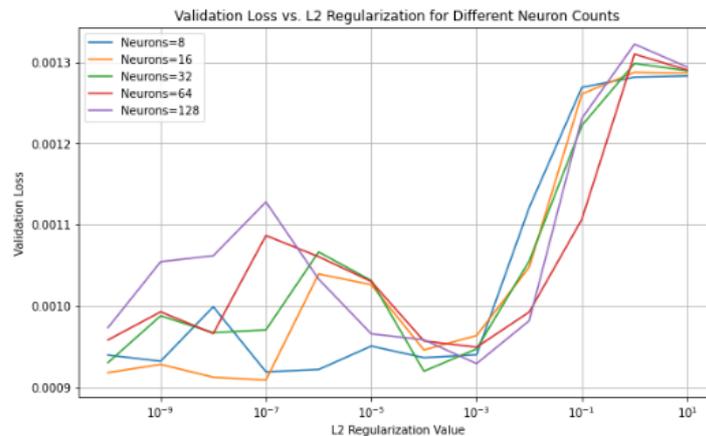


Figure 3.32: Validation loss vs L2 Regularization for 1 layer with different neurons for  $X_c$  input combination for buildings with concrete floor

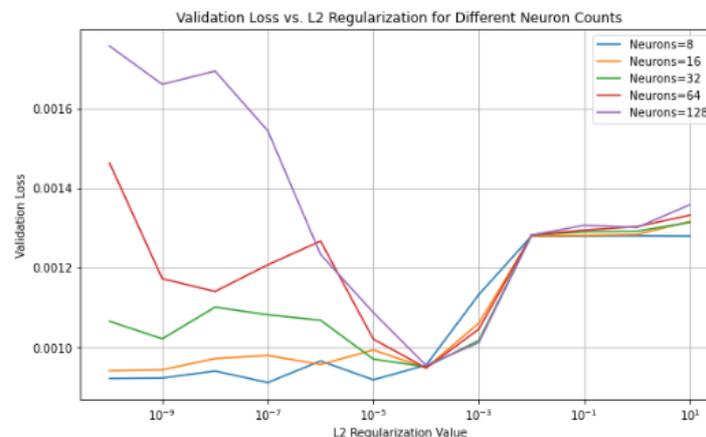


Figure 3.33: Validation loss vs L2 Regularization for 2 layer with different neurons for  $X_c$  input combination for buildings with concrete floor

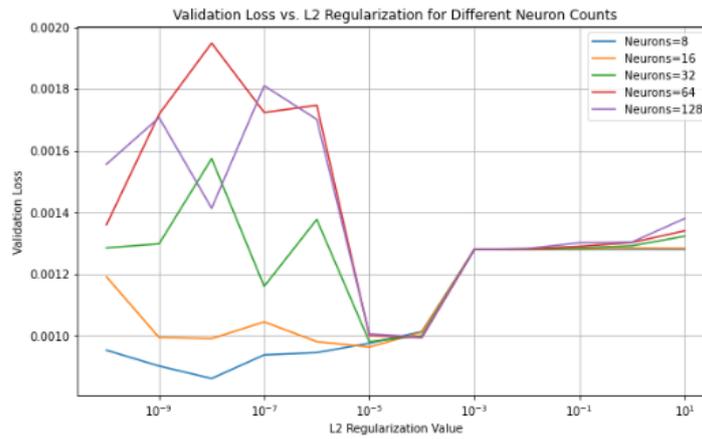


Figure 3.34: Validation loss vs L2 Regularization for 3 layer with different neurons for Xc input combination for buildings with concrete floor

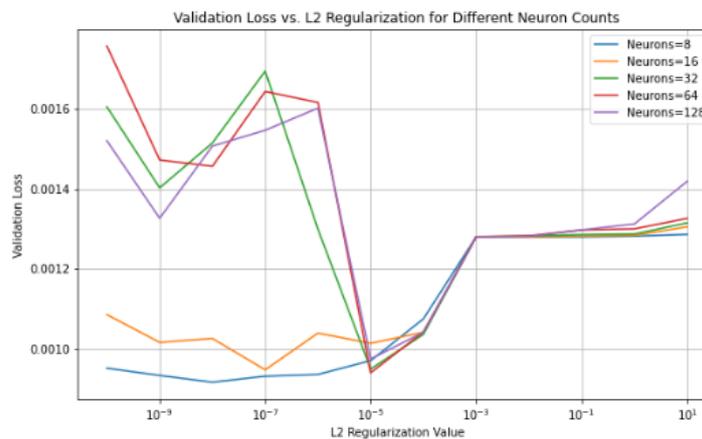


Figure 3.35: Validation loss vs L2 Regularization for 4 layer with different neurons for Xc input combination for buildings with concrete floor

Close observation of the results for different neurons and L2 regularisation values in Layer 1 (Figure 3.32), Layer 2 (Figure 3.33), Layer 3 (Figure 3.34) and Layer 4 (Figure 3.35), analysing them as described in section 3.2.2.2, primarily focusing on the validation loss values, led to the selection of a one layer with 128 neurons with L2 regularisation as 0.001 as the model architecture. Here, the point just before the model starts to overfit was selected from Figure 3.36. Compared to the other models architectures this model had a lower validation loss and smoother U-shaped gradient.

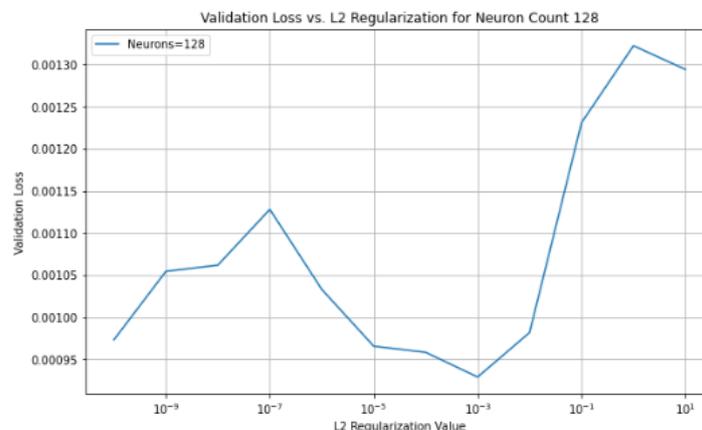


Figure 3.36: Validation loss vs L2 Regularization for 1 layer with 128 neurons for Xc input combination for buildings with concrete floor

With the model architecture selected, the model was trained and validated for the Xc input parameters. The difference between the validation and training loss is 4.88e-05 and The Epoch vs. Validation loss curve is

similar to the good fit model described in figure 3.4.

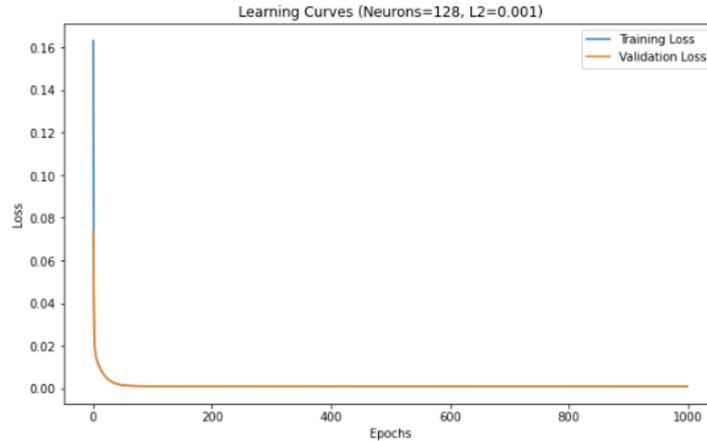


Figure 3.37: Epoch vs. Validation loss of the trained model with Xc input for buildings with concrete floor

Epoch vs. Validation loss curve for Xc input combination shown in Figure 3.37. The model has a one layer of 128 neurons with L2 regularisation of 0.001. Its validation loss is 0.00092 and training loss is 0.00088. Computing the models learning curve showed that, Figure 3.38, its smoothed moving average curve shows a linear decline indicating a gradual learning and its 95% CI is much wider than whats observed in Xa, Figure 3.18, and Xb, Figure 3.19. This graph, Figure 3.38, indicates that it requires more data points with the addition of an input parameter due to the curse of dimensionality as discussed before.

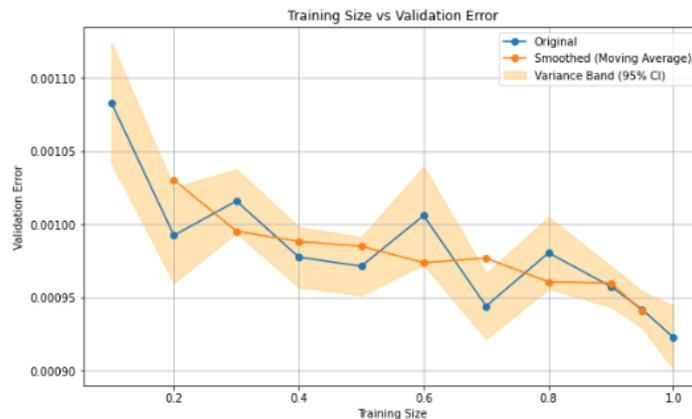


Figure 3.38: fig:Train Size vs. Validation loss of the model with Xc input for buildings with timber floor

Hence, for this analysis, it is crucial to identify input features that allow the model to effectively approximate the SLaMA analysis. The current reduced feature set fails to help the model capture the underlying relationships accurately. Therefore, when simplifying input features for this and other computationally expensive models, it's essential to ensure that valuable information is not lost. Careful feature selection or dimensionality reduction techniques, such as principal component analysis (PCA), can help maintain the balance between reducing complexity and preserving important information. It is also important to note that for the timber floor building analysis, the input variables were reduced to the wall level from the pier level, whereas for concrete floor buildings, the reduction was made at the building level. This creates a significant difference in the input space between the original analysis and the current surrogate model. Typically, for surrogate models to achieve results similar to the original analysis, the input space needs to remain consistent. When the input space is drastically reduced or altered, as in this case, the model may struggle to replicate the performance or accuracy of the original analysis. A recommendation for both timber and concrete floored models would be to normalise the output results during the training, validation, and testing phases. This is because, MAE, MSE and RMSE are all dimensional parameters, meaning they retain the same units as the target variable. This can be a limitation when comparing errors when the magnitude of the target variable varies. The larger errors typically associated with larger values of the output variable might dominate these evaluation metrics, leading to an overemphasis on accuracy for larger values. Hence using the evaluation metrics, on normalised output, will provide a fairer assessment as normalising transforms features to a common scale.

# Chapter 4

## Classification Model

The second model developed to achieve the goal of this research is the classification model. As mentioned in section 2.6.1, classification models are a type of machine learning algorithm designed to categorize input data into predefined classes. In this research, we specifically employ a binary classification model to address the problem at hand.

Binary classification models offer a simple and efficient approach in categorising and predicting discrete outputs. These models are useful in clearly providing whether the house requires measures, represented by 1 as the output, or does not require measure, represented by 0 as the output, based on the provided input data. This model can be used as an early stage analysis tool to check which buildings require to be prioritised first for further analysis.

### 4.1 Data cleansing and Preprocessing

The relational database from the VIIA team of RHDHV, as discussed in section 2.3, contains six Excel sheets. As previously noted, the data in each sheet is interconnected through the IDs provided to each part of an object.

The 'viiia\_objecten' sheet provides the actual name provided to the object in the MYVIIA. Each of these actual names is provided with an 'ID number'. The sheet also provides details on the archaeological values, asbestos presence, cluster number, conclusive analysis for the object, soil properties, time period, year of construction, PGA and NPR version. The next sheet 'object\_deel' provides details on the analysis method used to analyse the objects in 'viiia\_objecten'. The two sheets are connected by the 'ID number' in 'viiia\_objecten' and the 'viiia\_object\_id' in 'object\_deel'. This sheet also provides information on the consequence class, a description of whether the object was analysed as a whole or not, whether geotechnical analysis was done or not and conclusive analysis. Each of the 'viiia\_objecten' is provided with an ID number that is connected with the contents in the 'engineering\_database' sheet, by 'object\_deel\_id'. In this sheet information regarding each of the parts of the object is provided. This includes material of the floor and wall, gross floor area, number of storeys, the thickness of the wall, the sum of the length of the load-bearing wall, PGA value, NPR version, mass included and excluded, building height, building volume, building footprint, area of the load bearing wall, the area of the roof, base shear values and roof type. The 'object\_deel\_id' can be connected to the 'eng\_measures' sheet which provides information on the type of measure the object part requires. The 'rekenmethodiek' sheet provides information on the ID number provided for each analysis method. The analysis methods are represented by these numbers in 'object\_deel' sheet. The last sheet 'Spectra' provides location-specific details of each object. The details of each object can be related to sheet 'object\_deel', by the 'viiia\_object\_id' in both sheets. This sheet provides the GPS latitude and,  $T_B$ ,  $T_C$ , and  $T_D$ , PGA value and the reference period.

The output parameters for the classification model need to be in terms of 0 and 1. The 'eng\_measures' sheet contains measures from in-plane and out-of-plane failure measures, which are represented by identifiers. RHDHV maintains a comprehensive measures catalogue that details the measures provided for addressing both in-plane and out-of-plane failure situations. These measures are categorised and labelled from L1 to L9. Since the scope of the research is limited to in-plane analysis, L3 and L5, measures are considered to be 1, which indicates that

the object part requires measures. The rest of the measures are considered as 0, which suggests that the object part doesn't require measures.

As the next part in data cleansing, it was observed that within 'eng\_measures' each object part had multiple 0's and 1's which indicated that the part would have had in-plane and out-of-plane measures. To ensure that each object part is listed only once in the sheet, an object part with multiple entries of 0's and 1's is considered only for a single instance of 1. This approach assumes that if an object part has at least one 1 among multiple 0's, it indicates that the part requires measures. Similarly, for parts that have multiple 0's or 1's alone, they would appear for a single instance of 0 or 1.

After the output data has been preprocessed, next the input data points were preprocessed. The input points can be obtained from the 'engineering\_database' sheet. Both the 'engineering\_database' and 'eng\_measures' can be related by the 'object\_deel\_id'. The database is filled by the engineer after the numerical analysis has been conducted and results and measures are finalised. In some cases, the measures might not be recorded in the 'eng\_measures' database, despite the part being analyzed. Sometimes the 'engineering\_database' entries of the parts are not recorded even though the analysis was done for the part. Hence not all object parts present in the 'eng\_measures' database have corresponding entries in the 'engineering\_database' and not all objects in the 'engineering\_database' are present in the 'eng\_measures' database. To maintain uniformity, 'eng\_measures' sheet was first compared to 'engineering\_database' and the parts that were not there in 'eng\_measures' but present in 'engineering\_database' were removed from 'engineering\_database'. Similarly parts in 'engineering\_database' were compared to 'eng\_measures', and parts that were there in 'eng\_measures' but not in 'engineering\_database' were removed from 'eng\_measures'. The 'eng\_measures' sheet was merged to 'engineering\_database' sheet to consolidate the input and output parameters into a single sheet. Next, it was observed that some of the entries for the object part characteristics are 0, hence for the classification model to learn well these need to be removed.

The location-specific details provided in 'Spectra', would help to calculate the spectral displacement demand (Sd) value. To obtain the right Sd value for the right object parts in 'engineering\_database', first the right 'viia\_object\_id' needs to be extracted from the 'object\_deel' sheet by comparing the 'object\_deel\_id' in 'engineering\_database' to the ID number in 'object\_deel'. For all matching 'object\_deel\_id' and ID numbers, the corresponding 'viia\_object\_id' was extracted and added to the 'engineering\_database'. Comparing the 'viia\_object\_id' in 'engineering\_database' and 'Spectra', the Sd value for the object parts was extracted.

After the data cleansing process, there are 410 complete object parts, each with all the necessary data and measures fully recorded. From these, all numerical data are considered as input parameters. The list of these input and output parameters is provided in the table below. The input data are assigned to a variable X and the output to a variable y. All the features are in meters, kilograms and kilonewton, except nr\_levels.

<b>Input Variable</b>	<b>Description</b>
area_load_walls	area of all the load bearing walls, it excludes the opening area of the walls
area_roof	total area of the roof cover
building_footprint	area of the building measured to the external face of the external walls
building_height	height of the building
building_volume	volume of the building
gross_floor_area	total floor area within the building
length_load_walls	total length of the load bearing walls
mass_incl	total mass of the building
most_floor_height	the most common floor height in the building
most_thickness_load_walls	thickness of the walls of the building
nr_levels	the number of levels in the building
Displacement demand	spectral displacement demand, which is location specific

Table 4.1: Input features for the Classification model

<b>Output Variable</b>	<b>Description</b>
measure_type	It can be 0 if there is no reinforcement needed or 1 if reinforcement is needed

Table 4.2: Output features for the Classification model

Given the high dimensionality of the input data, we employ Principal Component Analysis (PCA) to reduce the number of dimensions. This approach is particularly suitable because PCA works effectively with numerical values and does not directly handle categorical data. Therefore, our focus on numerical input parameters aligns well with the requirements of PCA. PCA is a dimensionality reduction method, to reduce high dimensional data into smaller datasets, while still maintaining the significant patterns and trends.

In PCA, the new variables that are created are called principal components (PC's) which are a linear combination of the input variables and they are uncorrelated with each other. For the 12 input parameters, there will be 12 PC's, however, PCA is designed to concentrate the most relevant information in the first few PCs. This approach ensures that the majority of the important information is contained within the first few PCs, allowing for effective dimensionality reduction.

The construction of the PC's can be explained with the help of the figure 4.1;

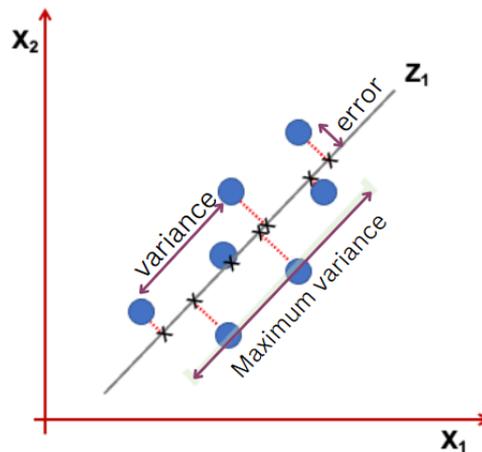


Figure 4.1: Example of constructing PC's

Consider the scatter plot of the data represented in the figure. The PC accounts for the largest possible variance. Hence the PC is oriented along the direction that accounts for the maximum variance in the data. While only one line is shown, in reality, there would be two PCs, as it is two dimensional, which will be orthogonal to the previous ones. The algorithm iteratively finds directions that maximize the variance of the projected data. Each subsequent PC captures the maximum remaining variance not accounted for by previous components. Simultaneously, as indicated in the figure, PCs are constructed to minimize the projection error, this involves finding the line that minimizes the perpendicular distances from the data points to the line.

To implement Principal Component Analysis (PCA) on our input data, first, the input variables are standardised, making sure the variance of all the variables is the same using StandardScaler from the sklearn.preprocessing. Then to apply PCA import PCA from sklearn.decomposition. Initially, we set the number of components to 12, matching our input variables. This allows us to observe how many reduced variables we can achieve while maintaining a 95% variance threshold. 95% makes sure that the majority of the data's variability is maintained. Using fit\_transform transform the X variable into PCA. Then plot the explained and cumulative variance, explained variance is how much each principal component contributes individually and cumulative explained variance explains how the total explained variance increases with more components. The plot shows that the 12 input variables can be reduced to 8 PC's, as shown in figure 4.2.

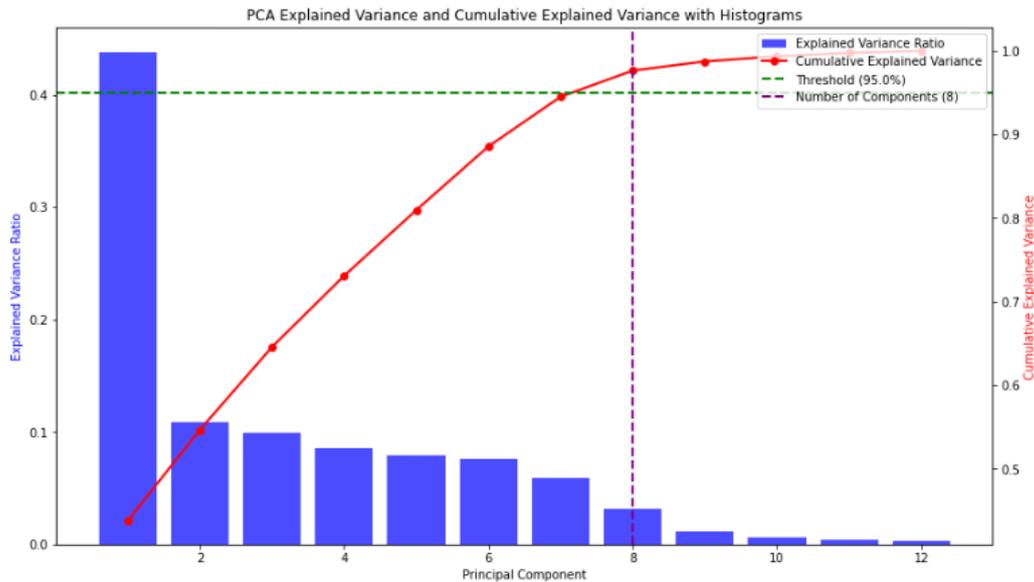


Figure 4.2: PCA analysis for input parameters of Classification model

The reduced input variables are then normalised using the MinMaxScaler from the sklearn.preprocessing module so that the DNN algorithm converges faster.

Once the input variables are normalised the input and output parameters are split into train, test and validation sets. Initially they are split into 80% of train data and 20% of test data. Then the training data is further divided into 60% train data and 20% validation data.

## 4.2 Model Architecture and Complexity

To identify the optimal model architecture and hyperparameters for minimizing validation loss, a systematic evaluation of model complexity was conducted. This evaluation focuses on determining the ideal number of layers, neurons per layer, and the appropriate L2 regularization values, similar to what was performed in section 3.2.2.2.

To evaluate the model complexity, different numbers of neurons in each hidden layer of the model explored were 8, 16, 32, 64 or 128 neurons, The different L2 regularisation values explored were  $1e-10$ ,  $1e-09$ ,  $1e-08$ ,  $1e-07$ ,  $1e-06$ ,  $1e-05$ ,  $1e-04$ , 0.001, 0.01, 0.1, 1.0 and 10. The values are spaced exponentially, in logarithmic scale, to explore a broad range from  $1e-10$  which is almost no regularization, while 10 represents a very strong regularization. Each combination of the neuron and L2 value is passed through a Sequential model built using TensorFlow, as mentioned in section 3.2.2.2 of the surrogate model. All the hidden layers utilise the ReLU (Rectified Linear Unit) activation function. Unlike the surrogate model, the output in this model is binary, hence, the activation function used in the output layer will be Sigmoid, which makes sure the results obtained are binary. The loss function used is binary crossentropy function, it is a loss function used to predict the loss between the actual binary label and the predicted binary label. To optimize the weights during the training the Adam optimizer (Adaptive Moment Estimation) is used and the model is run for 1000 epochs. This structure enables to assess how different levels of model complexity (through varying neuron counts and regularization strengths) affect the model's ability to learn and generalize from the data, to predict well on unseen data.

To determine the appropriate model complexity, the L2 regularization values were plotted against the validation loss for each layer configuration with different numbers of neurons. This resulted in four separate plots, one for each layer configuration from layer one to layer four. A typical plot, and how the plot is read is explained in section 3.2.2.2.

### 4.3 Training and Validating the Model

After assessing model complexity, loss evaluation curves were generated for the selected model architectures to track the progression of both training and validation losses. These Epoch vs. Validation Error curves are crucial for understanding the model’s learning trajectory across epochs and for identifying potential overfitting or underfitting issues. Figure 3.4, in section 3.2.2.3, provides details on underfitting, overfitting and good fit models.

After analysing the Epoch vs. Validation Error curves, the learning curves, Training Data Size vs. Validation Error curves, were generated, to analyse the model performance under varying data sizes. To achieve this, various training data sizes were tested, ranging from 10% to 99.99% of the training data, again for the same model architecture and 1000 epochs. Section 3.2.2.3 discusses how the curves are generated and and Figure 3.5 shows a typical learning curve.

### 4.4 Testing the Model

Once the model is trained it is necessary to observe its prediction on unseen data. To facilitate this, a confusion matrix is generated. The performance of the matrix is checked through Precision, Recall, F1 Score, and Accuracy. A confusion matrix provides how well a model performs by presenting a grid indicating how many times the prediction is accurate for binary output.

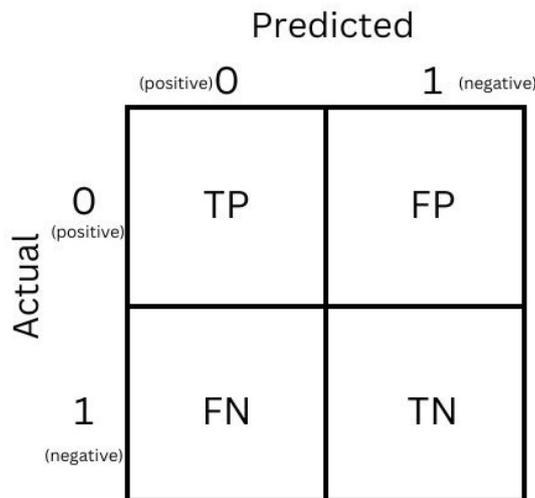


Figure 4.3: Example of confusion matrix

Where,

TP : True positive, where the model predicts the positive classes correctly.

TN : True negative, where the model predicts the negative classes correctly.

FP : False positive, where the model predicts the positive classes wrongly.

FN : False negative, where the model predicts the negative classes wrongly.

The following metrics are used to evaluate the confusion matrix performance for 0 and 1 binary output;

Precision: Precision measures how many of the instances the model has predicted as positive are actually positive. It tells how correctly the positive instances are predicted.

Recall: Recall measures how many of the positive instances the model has identified correctly. It tells how well the model finds the positive instances.

F1 score: is the harmonic mean of precision and recall. It is a single metric that tells how well the model identifies positive instances and the number of correct instances.

Accuracy: represents the number of correctly classified instances over the whole instances.

The metrics for Class 1 (positive class) are as follows:

$$\text{precision}_1 = \frac{TP}{TP + FP} \quad (4.1)$$

$$\text{recall}_1 = \frac{TP}{TP + FN} \quad (4.2)$$

$$\text{f1\_score}_1 = \frac{2 \times (\text{precision}_1 \times \text{recall}_1)}{\text{precision}_1 + \text{recall}_1} \quad (4.3)$$

The metrics for Class 0 (negative class) are as follows:

$$\text{precision}_0 = \frac{TN}{TN + FP} \quad (4.4)$$

$$\text{recall}_0 = \frac{TN}{TN + FN} \quad (4.5)$$

$$\text{f1\_score}_0 = \frac{2 \times (\text{precision}_0 \times \text{recall}_0)}{\text{precision}_0 + \text{recall}_0} \quad (4.6)$$

The overall Accuracy is calculated as:

$$\text{accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (4.7)$$

## 4.5 Results and Discussion

The results of the dataset passed through different number of layers, neurons and L2 regularisation values.

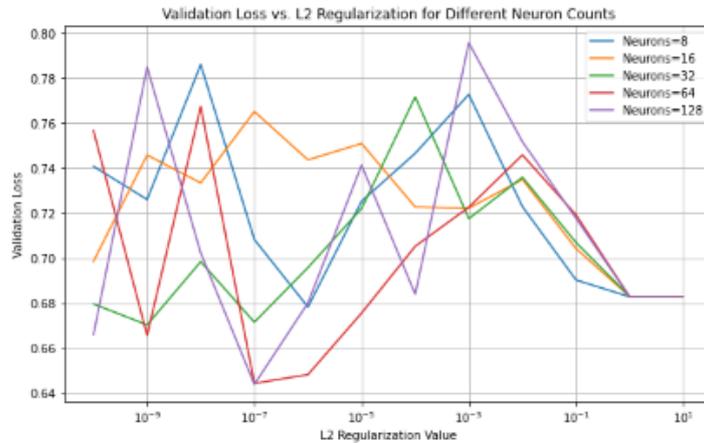


Figure 4.4: Validation loss vs L2 Regularization for 1 layer with different neurons for Classification model

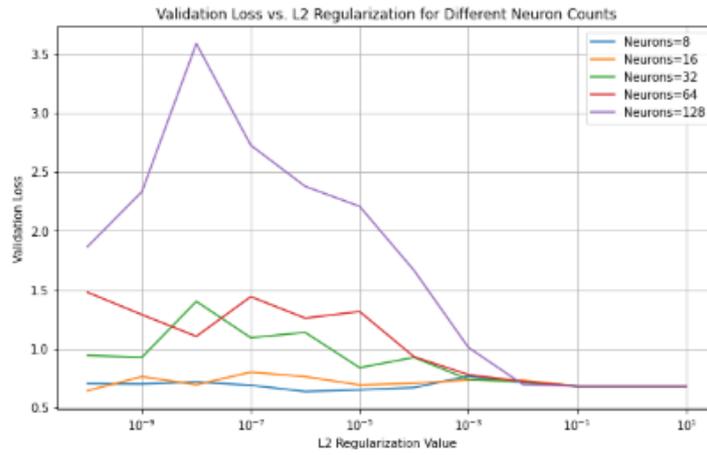


Figure 4.5: Validation loss vs L2 Regularization for 2 layer with different neurons for Classification model

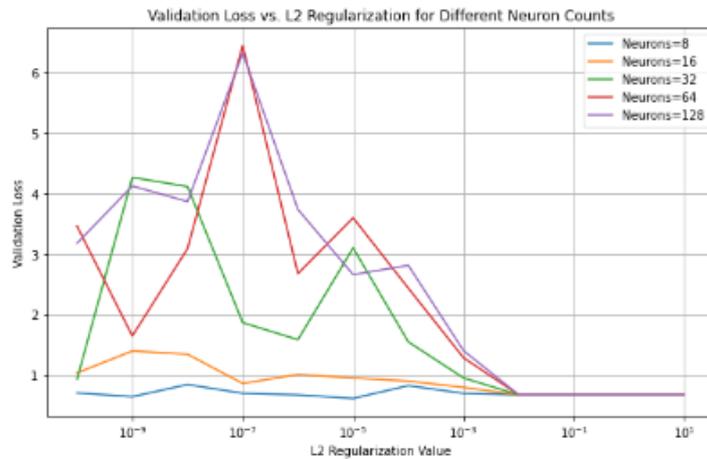


Figure 4.6: Validation loss vs L2 Regularization for 3 layer with different neurons for Classification model

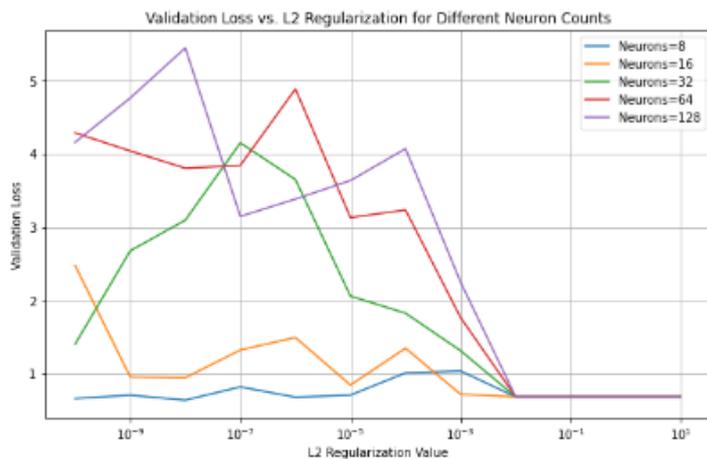


Figure 4.7: Validation loss vs L2 Regularization for 4 layer with different neurons for Classification model

Close observation of the results for different neurons and L2 regularisation values in Layer 1 (Figure 4.4), Layer 2 (Figure 4.5), Layer 3 (Figure 4.6) and Layer 4 (Figure 4.7), analysing them as described in section 4.2, primarily focusing on the validation loss values and curve smoothness, led to the selection of five promising model architectures, due to their consistently low and stable validation loss. The five models are represented in Figure 4.8, Figure 4.9, Figure 4.10, Figure 4.11 and Figure 4.12 ;

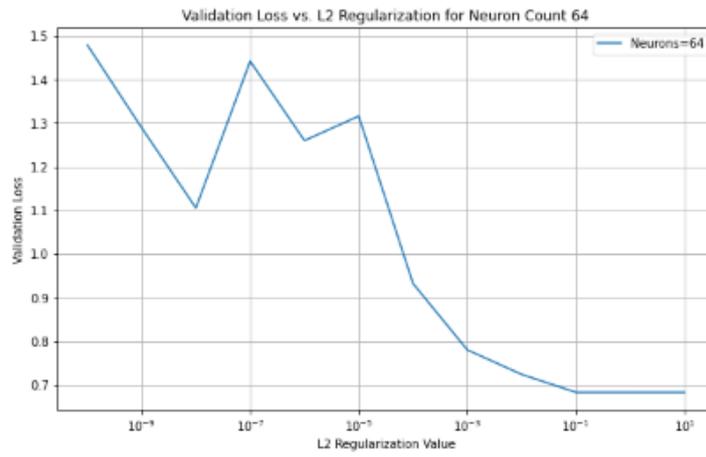


Figure 4.8: Validation loss vs L2 Regularization for 2 layer with 64 neurons for Classification model

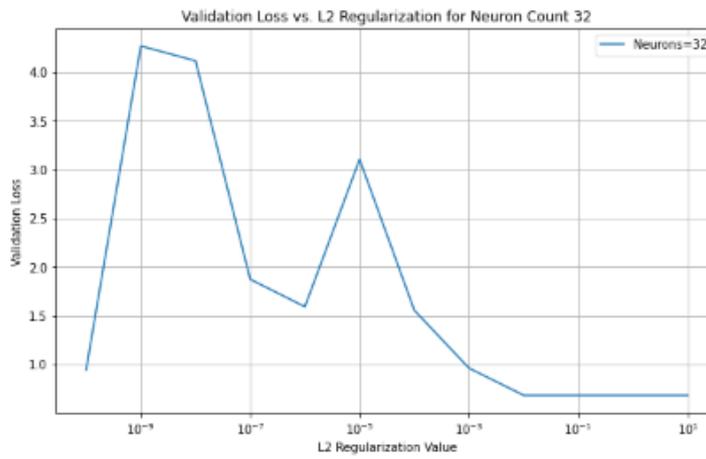


Figure 4.9: Validation loss vs L2 Regularization for 3 layer with 32 neurons for Classification model

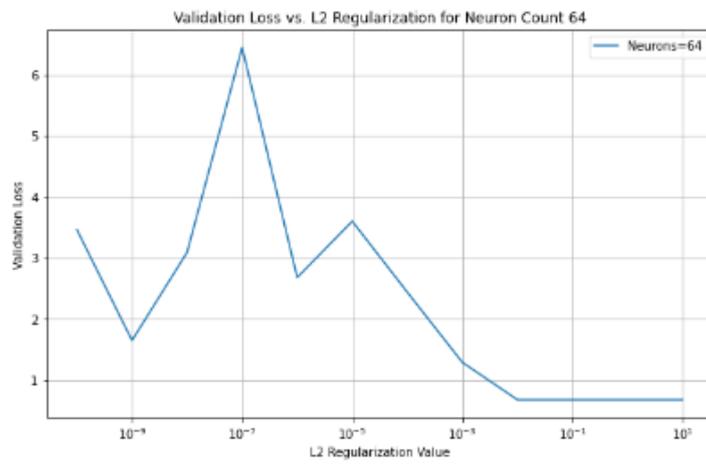


Figure 4.10: Validation loss vs L2 Regularization for 3 layer with 64 neurons for Classification model

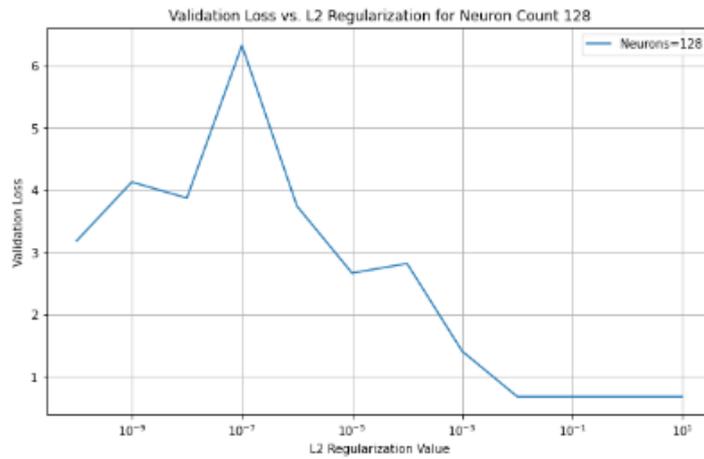


Figure 4.11: Validation loss vs L2 Regularization for 3 layer with 128 neurons for Classification model

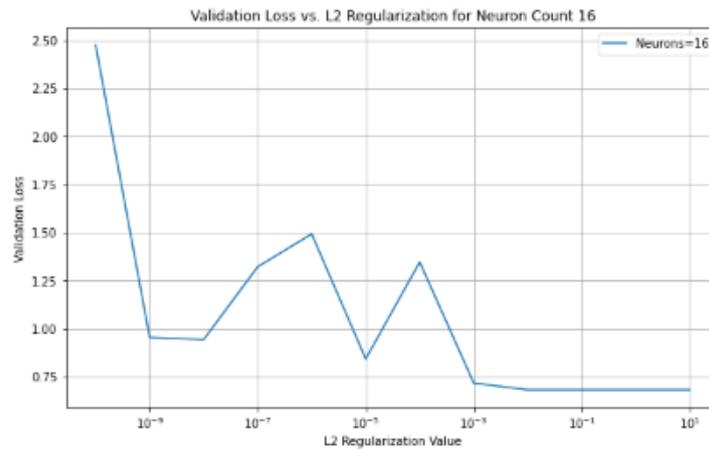


Figure 4.12: Validation loss vs L2 Regularization for 4 layer with 16 neurons for Classification model

Training and validating the models revealed that all architectures exhibited signs of overfitting, characterized by the Epoch vs. Validation loss curve being similar to the overfitting model described in figure 3.4. The figures below represent the Epoch vs. Validation loss curve for the five models.

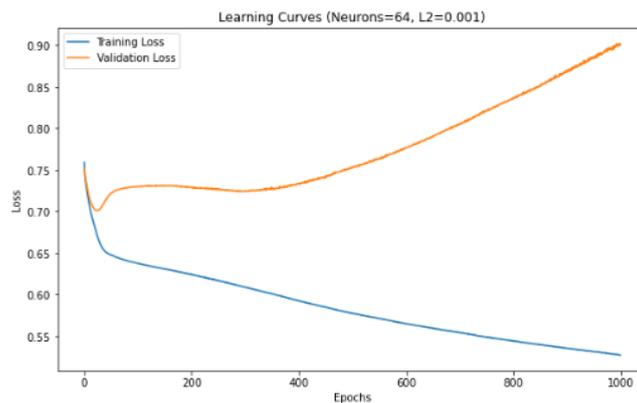


Figure 4.13: Train Vs Validation loss for Classification model with 2 layers, 64 neurons, L2=0.001

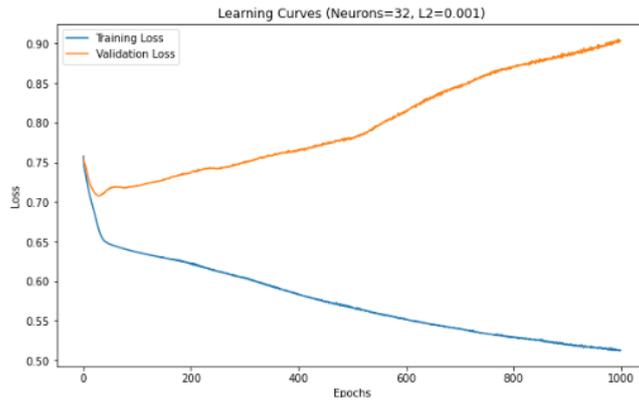


Figure 4.14: Train Vs Validation loss for Classification model with 3 layers, 32 neurons,  $L2=0.001$

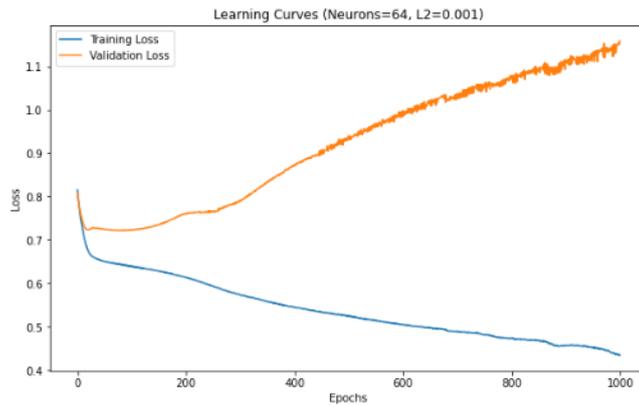


Figure 4.15: Train Vs Validation loss for Classification model with 3 layers, 64 neurons,  $L2=0.001$

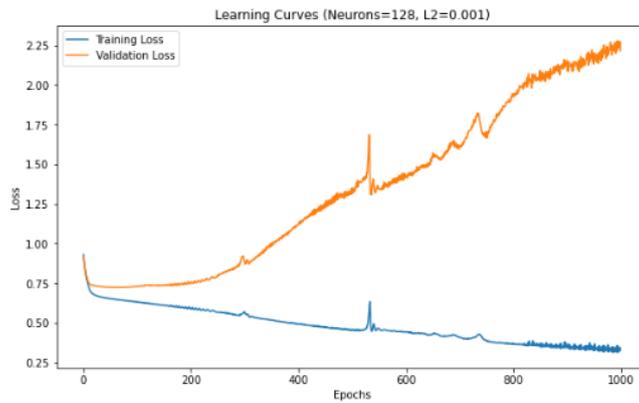


Figure 4.16: Train Vs Validation loss for Classification model with 3 layers, 128 neurons,  $L2=0.001$

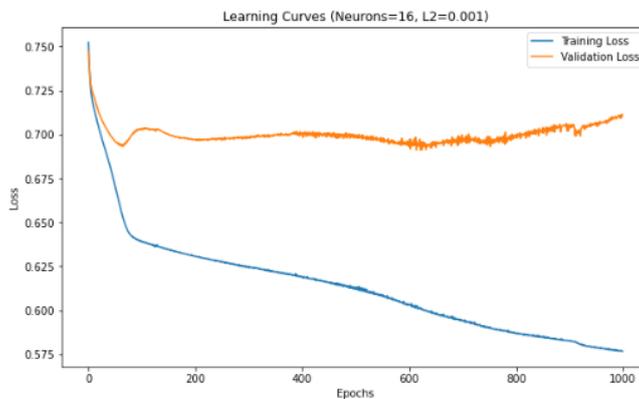


Figure 4.17: Train Vs Validation loss for Classification model with 4 layers, 16 neurons,  $L2=0.001$

This discrepancy indicates that while the models performed well on the training data, they struggled to generalize to unseen data. To address the overfitting the L2 regularization values were increased, which resulted in

an Epoch vs. Validation loss curve similar to the good fit model described in figure 3.4, showing the model is generalising well, but with this improvement, the model consistently predicted only the binary value 0 (which means no measures), on test data, despite previously being able to identify some positive cases. This can be because of an imbalance in the dataset. The dataset has 246 0's (which is 60.44% of the sample) and 161 1's (which is 39.56% of the sample), hence the imbalance ratio, calculated by dividing the majority class by the minority class, is 1.53:1. This means Class 0 is about 1.53 times more frequent than Class 1. This imbalance could be considered mild, as minority datasets are above 20% [7]. Hence the sole reason for the poor model performance cannot be the imbalance alone.

Observing the accuracy of the five models analysed above through the confusion matrices and the Precision, Recall, F1-score and Accuracy values of each of the models, see Table 4.3, the model with 3 layers, 32 neurons each performs better than the other four. The precision, recall, F1-score and accuracy of this model are higher compared to the other models.

Model Architecture	2 Layer, 64 Neurons	3 Layer, 32 Neurons	3 Layer, 64 Neurons	3 Layer, 128 Neurons	4 Layer, 16 Neurons
<b>PrecisionClass 1</b>	0.536	0.556	0.442	0.515	0.517
<b>RecallClass 1</b>	0.556	0.556	0.704	0.630	0.556
<b>F1 ScoreClass 1</b>	0.545	0.556	0.543	0.567	0.536
<b>PrecisionClass 0</b>	0.778	0.782	0.795	0.796	0.774
<b>RecallClass 0</b>	0.764	0.782	0.564	0.709	0.745
<b>F1 ScoreClass 0</b>	0.771	0.782	0.66	0.750	0.759
<b>Accuracy</b>	0.695	0.707	0.610	0.683	0.683

Table 4.3: Evaluation metrics Comparison of Model Architectures

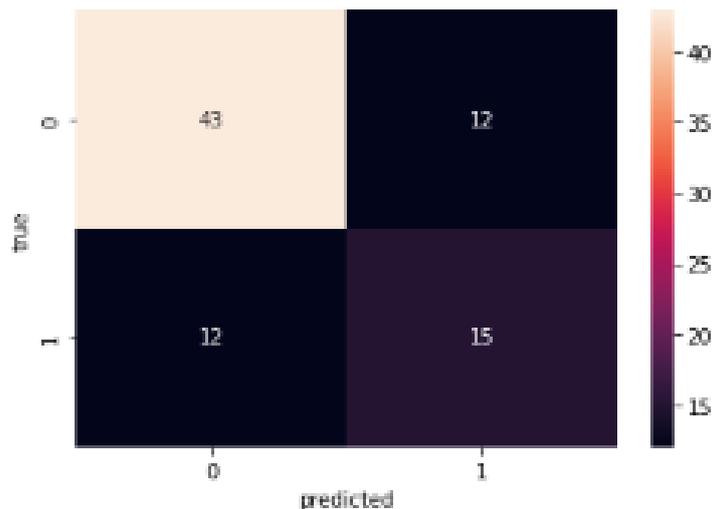


Figure 4.18: Confusion matrix for Classification model with 3 layers, 32 neurons, L2=0.001

The confusion matrix, Figure 4.18, as described in section 4.4 visually represents the frequency of correct and incorrect predictions for each class.

Metric	Training	Validation
<b>Loss</b>	0.5367	0.7738
<b>Accuracy</b>	0.7538	0.6307

Table 4.4: Comparison of Training and Validation Loss and Accuracy

As mentioned before the model does show overfitting, which can be understood by the increase in the validation loss as the training loss reduces, as well as the higher training accuracy than the validation accuracy. The confusion matrix's accuracy of 70.7% being close to the training accuracy also indicates that the model is overfitting. Hence, overfitting of the model does suggest in the end that the model has learned more of the training data, including the small details and noise, and hence is unable to generalise well with unseen data, showing less accuracy on the validation data than train data.

Given the observed overfitting of the PCA-based model, it is important to explore an alternative approach using all 12 original input features without dimensionality reduction. So for this input feature, to find the

optimal model architecture and complexity, the Validation loss vs L2 Regularization for different numbers of neurons, layers and L2 regularization values for 1000 epochs, were found. Observing the results for different layers and neurons, showed not much difference with the results observed for PCA-reduced input parameters. The validation losses remained within a similar range as observed in the PCA scenario, and the loss curves also exhibited signs of overfitting. Increasing the L2 regularization improved the model's performance, but it ultimately resulted in the model predicting only class 0, similar to the behaviour seen with the PCA approach.

This suggests that the selected features don't have a strong relationship with the target variable. With the current features in hand, a feature importance analysis was performed so that, by focusing on the most relevant features, the model performance can be improved and reduce overfitting. Mutual Information (MI) technique is used to identify the important features. MI is implement using sklearn.feature\_selection module, MI measures the dependency between the features and the target variable. Features with higher MI scores have a stronger relationship with the target variable, making them more important for prediction. Performing MI on the 12 input parameters showed the following results presented in figure 4.19.

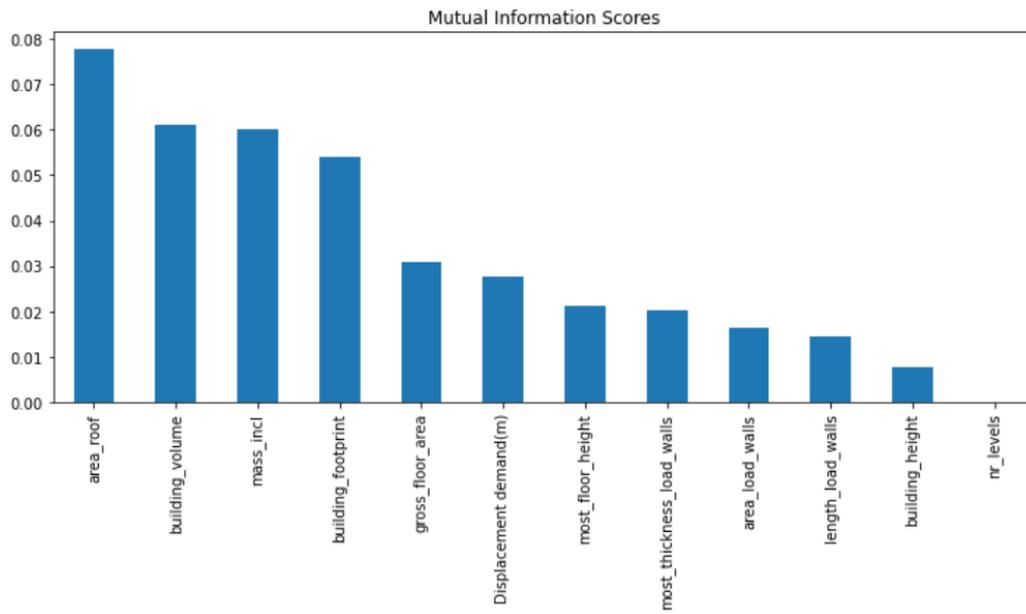


Figure 4.19: Mutual Information score - Feature engineering

The results show that 'area\_roof', 'building\_volume', 'mass\_incl' and 'building\_footprint' have the most relation to the target variable, while the other variables have a much smaller MI score. The MI scores can range from 0 to infinity and these scores are the dependency between a feature and the target variable. Hence it is important to note that the maximum score of 0.08 indicates that even the best feature has limited predictive power to explain the target variable. Since the scores are already small, 'area\_roof', 'building\_volume' and 'mass\_incl' can be considered as input parameters keeping 0.06 as the threshold and ignoring parameters below this score.

So for this input feature, to find the optimal model architecture and complexity, the Validation loss vs L2 Regularization for different numbers of neurons, layers and L2 regularization values for 1000 epochs, were found. Observing the results for different layers and neurons, analysing them as described in the section 3.2.2.2, primarily focusing on the validation loss values and curve smoothness, led to the selection of two layers with 64 neurons as the model architecture. The L2 regularisation value is 0.001, which is the point in the graph, Figure 4.20, just before the model starts to overfit.

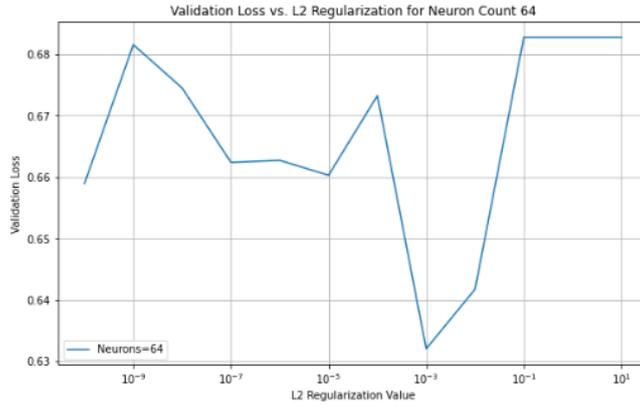


Figure 4.20: Validation loss vs L2 Regularization for 2 layer with 64 neurons for Classification model

With the model architecture selected, the models was trained and validated. The The Epoch vs. Validation loss curve, Figure 4.21, is similar to the good fit model described in figure 3.4, with the validation and training loss decreasing to stability. This indicates good generalization capabilities.

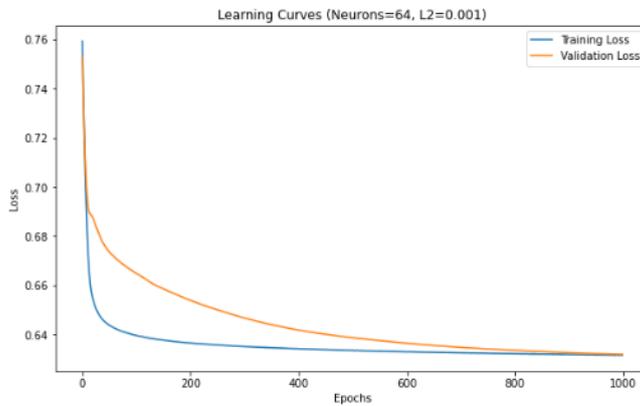


Figure 4.21: Epoch vs. Validation loss Train Vs Validation loss for Classification model with 2 layers, 64 neuron

The learning curves of the model, Figure 4.22, show a smooth decreasing curve, where the validation loss decreases with an increase in data size, indicating it's learning more effectively from the data. The variance band indicates that the predictions are consistent.

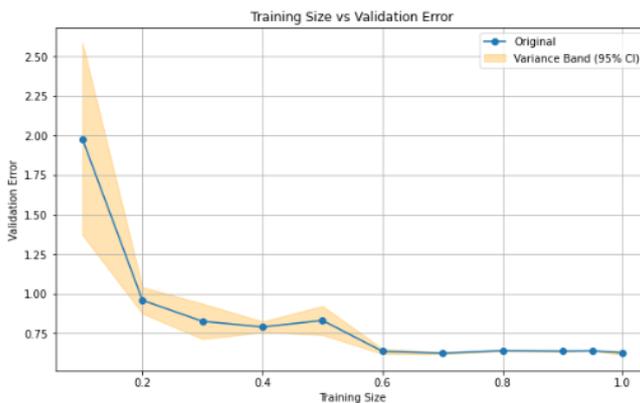


Figure 4.22: Train Size vs. Validation loss of the classification model

To observe how the model performs on the unseen data, the model is tested on unseen test data and a confusion matrix is generated.

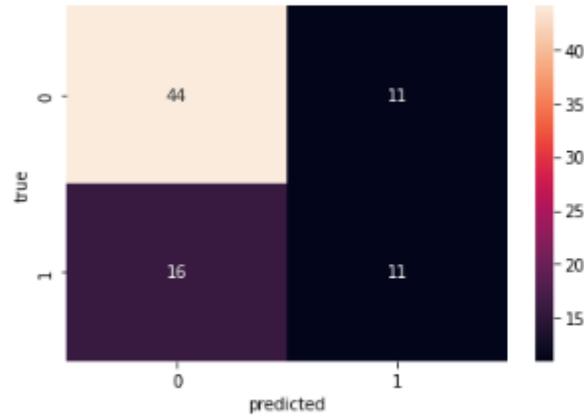


Figure 4.23: Confusion matrix

The confusion matrix as described in section 4.4 visually represents the frequency of correct and incorrect predictions for each class. For class 0, the model correctly predicts 44 instances, which is a good performance. However, it incorrectly classifies 11 instances of class 0 as class 1. For class 1, the model’s performance is poorer, correctly identifying only 11 instances, while misclassifying 16 instances as class 0. This indicates that the model is more accurate in identifying class 0 but struggles with class 1, suggesting a potential bias towards class 0 predictions.

However, it’s important to note that these predictions are more reliable compared to the earlier models. Previously, it was observed that predictions in class 1 were made only when the model was overfitting, which can be misleading. In contrast, this confusion matrix represents predictions from a model that generalizes well, as evidenced by its loss curve. The feature selection showed an improvement in the model but does not still fully capture the characteristics that distinguish class 1 from class 0.

In an overall observation, the model does show a better performance than before. It is performing well on the loss curve and learning curve (indicating the number of data points is sufficient), however, its prediction of the unseen data is poor. This issue, as previously discussed, stems from the weak relationship between the features and the target variable. There is notable overlap between classes 0 and 1, as illustrated in Figure 4.24.

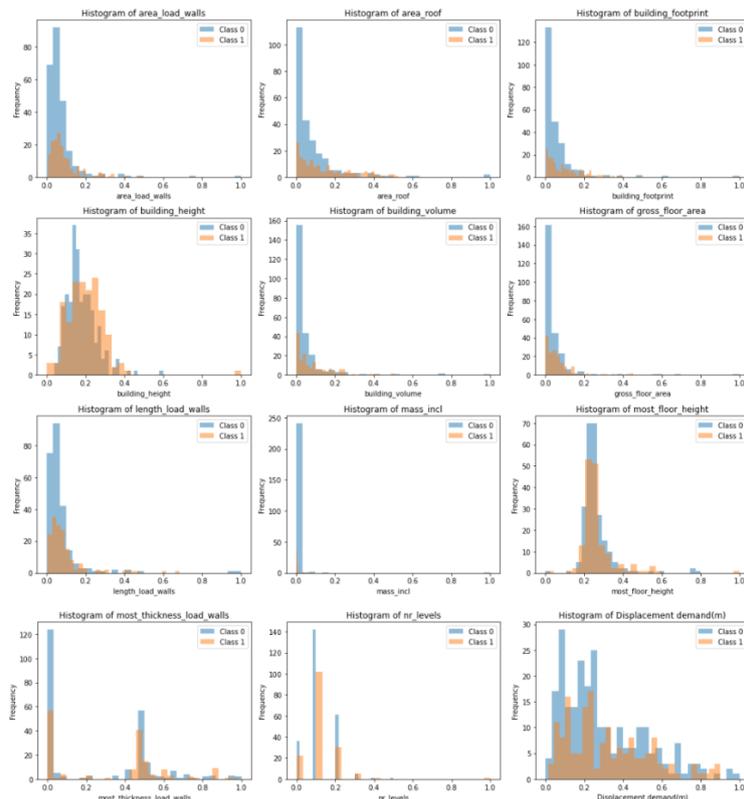


Figure 4.24: Class overlap of input variables in Classification model

While there are some features that show a significant overlap of class 0 and 1, which means the feature may not be very effective at distinguishing between the classes, there are features where one class is clearly distinct, which shows that the feature is highly effective for one class favouring that class more. Hence the reason for better predicting on class's 0. The future step can be to find input parameters that relate better with the target variable. However, more than finding more relevant features the inability of the model to predict well on unseen data could be because of the different NPR guidelines used over the years. Since 2015 the NPR versions have been changed. The dataset contains houses that have been analysed since 2015 and they have used different NPRs. Hence houses that might have required measures (or class 1) in 2015 according to the NPR then might not require measures today according to the current NPR.

Comparing the NPR versions from 2015 and 2020, the latest guidelines have reduced the margin for deviating interpretations and provide clearer classification of seismic and non-seismic structural elements. A new method for analyzing the risk of falling structural elements has been introduced, and the NLTH method has been revised. Additionally, a new setup for more effective use of pushover analyses (SLaMA) has been incorporated, along with criteria for assessing the out-of-plane resistance of masonry walls. The interpretation of masonry walls has also been updated, categorizing them as either non-seismic or as primary/secondary seismic elements depending on the direction of the force. Recommendations for soil investigations with a region-specific approach for existing buildings were also added. Furthermore, the ground motions used in the NPR9998 web tool were updated from version V4 to V6 between 2017 and 2020. This includes better subsurface characterization, the addition of magnitude and distance-dependent amplification factors, and modeling tremors along faults rather than at single points. Errors in formulas and references from previous versions were also corrected.

# Chapter 5

## Conclusion and Recommendation

This chapter summarizes the key conclusions of the thesis, starting with a discussion of the research problem, the methodology used, and the results obtained to achieve the thesis's objectives, followed by future research recommendations.

### 5.1 Conclusions

The primary aim of the thesis was to explore to what extent machine learning approaches can be leveraged, to optimise and accelerate the time required for modelling and analysing URM residential structures. This research was motivated by the ongoing seismic activity in Groningen, Netherlands, resulting from gas extraction. This activity has led to considerable damage to URM buildings, and traditional numerical analysis methods, while effective, are time-consuming. To address this, the thesis focused on developing faster, machine learning-based models as alternatives. This brings to the primary research question?

#### **To what extent can machine learning approaches be leveraged to optimise and accelerate the time required for modelling and analysing URM residential structures?**

Several sub-research questions were developed to address the main research question. A summary of the findings from these supporting questions is provided below:

- What are the effects of earthquakes on URM buildings, and what are the associated failure mechanisms?

Most of the buildings in Groningen are made of URM, which are vulnerable to seismic shocks. Hence a low-magnitude earthquake would damage the houses. The in-plane capacity of the walls, as mentioned in section 2.2.1 withstand seismic forces acting horizontally within the plane of the wall. The global capacity of the buildings is determined by the in-plane capacity of the walls. The in-plane failure mechanisms considered here are bed joint sliding and rocking failure. The out-of-plane seismic capacity of walls refers to their ability to withstand the force acting perpendicular to their plane. The local capacity of the building is influenced by the out-of-plane capacity of the walls, which depends on the ability of connections and diaphragms to transfer loads from walls subjected to out-of-plane forces. During a seismic event, this failure could be due to large openings or long spanning walls or poor material. The scope of this research is limited to in-plane failure focusing on the overall structural response.

- According to the existing database, what are the various seismic analysis methods currently employed to assess unreinforced masonry (URM) buildings?

An existing database, provided by the VIIA team of Royal HaskoningDHV has been utilised in this research. Within the database, the analysis methods currently employed to assess the seismic response of structures in Groningen are Model Response Spectrum Analysis (MRS), Non-linear Pushover Analysis (NLPO), Non-linear Time History Analysis (NLTH), Lateral Force Method (LF) or Reference approach of NLTH (REF-NLTH) or NLPO Analysis (REF-NLPO).

- NLPO: is a nonlinear static analysis method, where the response of the structure is estimated by gradually

applying lateral force till the structure collapses. It can be performed numerically and analytically.

- NLTH: is a nonlinear dynamic analysis method that uses a 3D model and an accurate earthquake time history to capture the structure's complex behaviour.
- MRS: is a linear dynamic analysis where the maximum response at each mode of vibration (velocity, acceleration, or displacement) is summed to obtain the overall structural response.
- LF: converts dynamic forces into static ones and applies them laterally to the structure, making it the most conservative analysis method.
- REF: uses results from similar previously analyzed objects (via NLTH or NLPO) to recommend reinforcement for a new object, based on comparable PGA values, wall connections, thickness, and PSSE materials.

The details on the number of houses analysed using each of these analyses are provided in depth in section 2.3.

- For the scope of this research, which types of machine learning models are suited for analysis?

A classification and surrogate model have been utilised in this research. Classification models offer quick, actionable results to categorise buildings and make immediate decisions to prioritise high-risk structures for further analysis. The surrogate model approximates complex, computationally expensive simulations with simpler, faster models, enabling quick yet reliable predictions of structural behaviour. Both the models are build using Deep Neural Network.

- What are the potential input and output parameters that can be extracted from an existing database of characteristic parameters and analysis results [2.3], for developing a machine learning model for the seismic analysis of unreinforced masonry (URM) buildings?

For the classification model, almost all the numerical data regarding the building characteristics were extracted and subjected to PCA analysis to reduce the high dimensionality of the input space. However due to the poor performance of the classification model feature selection was performed and 'area\_roof', 'building\_volume', and 'mass\_incl' were identified as the parameters most correlated with the target variable. Despite this, the overall correlation between all parameters in the database and the target variable was small, with the highest correlation score being only 0.08. For the surrogate model, object parts with SLaMA analysis were identified from the VIIA database and detailed reports on each of the object parts were extracted from MYVIIA. Pier details of seven object parts were manually extracted from the reports to conduct the SLaMA analysis. To generate more data, manually extracted data for buildings with timber and concrete floors were augmented separately using LHS sampling method to generate more data.

Including all the input parameters increases the precision of the model prediction, but it would also significantly increase the model's complexity, computational cost and need for more data points. Hence the input parameters were reduced to wall-level parameters for timber floored buildings: Lwall(m), Hwall(m), Twall(m), Wall\_material, total\_overburden\_load(KN) and total\_pier\_length(m). Similarly, for concrete floored buildings, the input parameters have been reduced to the building level parameters: Hbuilding(m), sum\_Lwall(m), Wall\_material, total\_overburden\_load(KN) and total\_pier\_length(m).

- Is the available data sufficient to train, test, and validate a machine learning model?

In the classification model, although there are sufficient data points and the dataset's original distribution is preserved without oversampling or undersampling, the features do not strongly correlate with the target variable. This is due to changes in NPR regulations over the years; for instance, buildings assessed in 2015 might not require the same measures under current NPR guidelines. As a result, there is an overlap between classes 0 and 1 within each feature, indicating a lack of consistency in the assessment procedures. Therefore, while the number of data points is adequate, their value would increase if they followed a consistent assessment standard, meaning if all the buildings used the same version of NPR.

For the surrogate model, the data points were sufficient for timber and concrete models when using lesser input parameters. However, to improve the model's accuracy and make it a better approximation of the SLaMA analysis, the addition of input features was explored. This increase in input dimensions led to the well-known

"curse of dimensionality" problem, where the model requires exponentially more data points to maintain its performance as the number of input features grows. The addition of input features to observe the model performance was easy in SLaMA due to the ease with which data can be generated but for computationally expensive models this can be challenging to run more time-consuming models to generate more data with an increase in input parameters. Hence an approach would be to use PCA analysis on their input space. PCA, reduces high dimensional data into smaller datasets, while still maintaining the significant patterns and trends. For the timber and concrete models, the input space was reduced initially, by aggregating the effects of individual piers into a more manageable set of inputs. This approach was taken to mitigate dimensionality issues and computational costs. However, this reduction must be done carefully to avoid the model learning noise or failing to extract meaningful patterns from the data. The surrogate model performs well when the input space is the same as the original analysis. Hence for computationally expensive models try to keep the same input space rather than aggregating parameters as initially done.

Finally, instead of aggregation, a better approach for complex models would be to apply techniques like PCA to the full set of original inputs. So that the model can make use of an input space similar to the original analysis but will not need more data points due to dimensionality reduction from PCA.

Additional key results from the two machine learning models built:

- From the above conclusions, using SLaMA analysis as a surrogate model was a promising approach. This SLaMA surrogate model provided valuable insights into the impact of input parameters, the required variability in these parameters, and the number of data points required for the model, which were easily met using SLaMA analysis. Additionally, it demonstrated the advantages of quicker analysis of the surrogate model, quickly testing and validating different parameter variations compared to starting with more computationally intensive simulations like NLTH.
- The SLaMa surrogate tool is a robust tool for verification before applying more detailed simulations. However, the current models are still in a preliminary stage and require further refinement. Specifically, additional analysis is needed to optimize the selection of input parameters to improve the model's generalization capabilities and performance on diverse datasets.
- The classification tool serves as a quick analysis method that can be utilized in the preliminary design stage to prioritize buildings. However, the current model is limited due to changes in NPR regulations over the years, resulting in a weak relationship between the input and target variables.
- A general comparison between both surrogate and classification models shows that surrogate models are generally more reliable as they provide approximations of seismic analysis methods, offering a more accurate representation of the underlying phenomena. In contrast, classification models categorize data into discrete classes, which can significantly depend on the balance of the dataset. Imbalances in class distribution can adversely affect the performance and accuracy of classification models, making surrogate models a more robust choice for reliable predictions.

This sub-research question led to the result of the final research question. While machine learning approaches show promise for improving seismic analysis efficiency, the current models are still in an immature state, and addressing their limitations mentioned earlier is crucial for enhancing their performance.

## 5.2 Recommendation

To enhance the performance and applicability of the models developed, several key areas for future work are essential:

- For the surrogate model, it is crucial to carefully select the most relevant input parameters to accurately capture the relationship between the input and output variables, ensuring the model generalizes well to unseen data. This can be achieved through machine learning techniques like Embedding. Embedding is a way to represent data in a lower-dimensional space, where important relationships between features are maintained. They can help the model better understand and learn from both categorical and numerical data, making it more effective than using raw data or one-hot encoding. Embedding reduces the

dimensionality of the input space without losing essential information.

- For the classification model a possible way to improve its performance would be to revise the analysis of the houses using the latest NPR version. Producing a new, consistent dataset based on the updated NPR guidelines would reduce the overlap of classes and allow the model to learn a better relation between the input and output parameters.
- The current surrogate model is for one-storey buildings for in-plane failures, future approaches can be for multi-storey buildings, including out-of-plane failures. Additionally, gathering more data on the different building types in Groningen, as identified by TNO, would help make the model more relevant to the region.
- A recommendation for surrogate models would be to normalise the output results during the training, validation, and testing phases. This is because, MAE, MSE and RMSE are all dimensional parameters, meaning they retain the same units as the target variable. This can be a limitation when comparing errors, when the magnitude of the target variable varies.
- To improve the accuracy of the surrogate model, combining high-fidelity NLPO analysis with low-fidelity SLaMA analysis can be effective. This approach allows the model to run at the speed of the SLaMA analysis while achieving the accuracy of the NLPO.

# References

- [1] Building Code Requirements for Masonry Structures (ACI 530-05/ASCE 5-05/TMS 402-05) ; Specification for Masonry Structures (ACI 530.1-05/ASCE 6-05/TMS 602-05) ; Commentary on Building Code Requirements for Masonry Structures (ACI 530-05/ASCE 5-05/TMS 402-05) ; Commentary on Specification for Masonry Structures (ACI 530.1-05/ASCE 6-05/TMS 602-05). URL <https://books.google.nl/books?id=GCORxyORHKEC&lpg=SL3-PA7&pg=PP2#v=onepage&q&f=false>.
- [2] Wikipedia contributors. Groningen gas field, 8 2024.
- [3] D. D’Ayala and E. Speranza. Definition of collapse mechanisms and seismic vulnerability of historic masonry buildings. *Earthquake spectra*, 19(3):479–509, 8 2003. doi: 10.1193/1.1599896. URL <https://doi.org/10.1193/1.1599896>.
- [4] Technical Committee CEN/TC 250 "Structural Eurocodes". Eurocode 6 - Design of masonry structures - Part 1-1: General rules for reinforced and unreinforced masonry structures . *The European Union*, 2005.
- [5] New Zealand Society for Earthquake Engineering. Assessment and improvement of the structural performance of buildings in earthquakes. Incl. Corrigenda 1 2. 63, 2017.
- [6] MJ Fox, BA Westeneng, CA Muir, AC Baird, WY Kam, PL Beazley, RD Jury, RD Sharpe, et al. Simple lateral mechanism analysis (slama) for the seismic assessment of unreinforced masonry structures. In *Proceedings of the 17th World Conference on Earthquake Engineering*. Japan Association for Earthquake Engineering Sendai, Japan, 2020.
- [7] Haibo He and Edward A. Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.
- [8] H.K. Hilsdorf. Investigation into the failure mechanism of brick masonry loaded in axial compression. *Designing, engineering and constructing with masonry products*, page 41, 1969.
- [9] Alexandr Ibragimov. little-known facts about silicate bricks and fires. *E3S Web of Conferences*, 138:01009, 1 2019. doi: 10.1051/e3sconf/201913801009. URL <https://doi.org/10.1051/e3sconf/201913801009>.
- [10] Fangming Jiang, Xiong Long, Likang Tian, Yan Tan, and Jiangtao Yu. Tensile strain-hardening cementitious composites and its practical exploration without reinforcement: A review. *Frontiers in Materials*, 9, 12 2022. doi: 10.3389/fmats.2022.1066796. URL <https://doi.org/10.3389/fmats.2022.1066796>.
- [11] Rajat Kapoor. Understanding structures and earthquakes in Groningen , 2018. URL <https://repository.tudelft.nl/record/uuid:25020a5a-094a-42d0-91ef-6269e81317c1>.
- [12] Zac Keeney. Earthquakes, magnitude and success, 11 2014.
- [13] Gregory R. Kingsley, P. Benson Shing, Thomas Gangel, National Institute of Standards, Technology,

- Gregory R. Kingsley, P. Benson Shing, and Thomas Gangel. Seismic design of special reinforced masonry shear walls. Technical report, 2014. URL <https://www.nehrp-consultants.org/publications/download/nistgcr14-917-31.pdf>.
- [14] Paul A. Korswagen, Michele Longo, and Jan G. Rots. Calcium silicate against clay brick masonry: an experimental comparison of the in-plane behaviour during light damage. *Bulletin of Earthquake Engineering*, 18(6):2759–2781, 2 2020. doi: 10.1007/s10518-020-00803-5. URL <https://doi.org/10.1007/s10518-020-00803-5>.
- [15] Guido Magenes and Gian Michele Calvi. In-plane seismic response of brick masonry walls. *Earthquake engineering and structural dynamics/Earthquake engineering structural dynamics*, 26(11):1091–1112, 11 1997. doi: 10.1002/(sici)1096-9845(199711)26:11. URL [https://doi.org/10.1002/\(sici\)1096-9845\(199711\)26:11<1091::aid-eqe693>3.0.co;2-6](https://doi.org/10.1002/(sici)1096-9845(199711)26:11<1091::aid-eqe693>3.0.co;2-6).
- [16] Christian Chukwudi Mathew. Investigation into the failure mechanism of masonry under uniaxial compression based on fracture mechanics and nonlinear finite element modelling, 2011. URL <https://www.ijserp.org/monograph/failure-mechanism-of-masonry.pdf>.
- [17] Shivam Mishra and Shivam Mishra. Dutch Government to permanently shut Groningen gas field. 6 2023.
- [18] G Mohamad, Fernando S Fonseca, At Ad Vermeltoort, and A Lubeck. Stiffness plasticity degradation of masonry mortar under compression: preliminar results. *Revista IBRACON de Estruturas e Materiais*, 11 (2):279–295, 4 2018. doi: 10.1590/s1983-4195201800020004.
- [19] Franklin L. Moon, Tianyi Yi, Roberto T. Leon, and Lawrence F. Kahn. Testing of a Full-Scale unreinforced masonry building following seismic strengthening. *Journal of Structural Engineering*, 133(9): 1215–1226, 9 2007. doi: 10.1061/(asce)0733-9445(2007)133:9(1215). URL [https://doi.org/10.1061/\(asce\)0733-9445\(2007\)133:9\(1215\)](https://doi.org/10.1061/(asce)0733-9445(2007)133:9(1215)).
- [20] NEN. *NEN (2020) - Assessment of the structural safety of buildings in case of erection, reconstruction, and disapproval - Induced earthquakes – Basis of design, actions and resistances. NPR9998:2020, NEN*. 1 2020. URL <https://www.nen.nl/npr-9998-2020-en-280873#:~:text=NPR%209998%3A2020%20geeft%20richtlijnen,na%20versterking%20voldoende%20aardbevingsbestendig%20zijn>.
- [21] NZGS NZSEE, SESOS. Guidelines - Seismic Assessment of Existing Buildings - Section C8: Unreinforced Masonry Buildings., 2017.
- [22] Robin Pascoe. Groningen gas fields to close permanently in October 2024 - DutchNews.nl, 6 2023.
- [23] Content Share and Mike Corder. Manmade quakes force Dutch to face future without gas, 1 2018.
- [24] Jesper Spetzler and Bernard Dost. Hypocenter estimation of induced earthquakes in Groningen. *Geophysical journal international*, page ggx020, 1 2017. doi: 10.1093/gji/ggx020. URL <https://doi.org/10.1093/gji/ggx020>.
- [25] Nicola Tarque, Guido Camata, Enrico Espacone, Humberto Varum, Catholic University of Peru, ROSE School-IUSS, Marcial Blondet, University of Aveiro, and University G. D’Annunzio. NUMERICAL MODELLING OF IN-PLANE BEHAVIOUR OF ADOBE WALLS. Technical report, 2010.
- [26] Keras Team. Keras documentation: Adam.
- [27] Zihao Wang Viktor Peterson. Cross-comparison of Non-Linear Seismic Assessment Methods for Unreinforced Masonry Structures in Groningen. *Concrete structures*, page 290, 12 2020.