International Institute of Information Technology, Hyderabad

Geometry Optimization Using Machine Learning Techniques

BTP Final Project Report

Team Members:

Manas Sachin Deshmukh	2022102040
Aditi Singh	2022101024
Faculty Advisor:	Jofin George

Submission Date: 4th September 2025

Contents

Abstract

This report presents a comprehensive, mathematically rigorous machine learning framework for optimizing masonry arch geometries. The study aims to minimize the probability of collapse under combined static and dynamic loading conditions, including gravity, external vertical loads, and seismic effects. By integrating classical limit analysis, virtual work principles, and multi-output neural network regression models, the framework predicts collapse factors and plastic hinge locations for diverse arch geometries. Tens of thousands of simulations generate a high-dimensional dataset encompassing variations in span, rise, thickness, material density, and block counts. The proposed multi-output ANN exhibits high prediction accuracy, validated with metrics such as MSE, MAE, and R^2. The framework serves as a computationally efficient pre-design tool, providing engineers with data-driven insights for safe and resource-efficient masonry arch design. Future extensions include integration with evolutionary optimization and CAD-based automated design pipelines.

1 Introduction

Background and Motivation

Masonry arches have historically been essential in both ancient and modern architecture. Their load-bearing behavior depends sensitively on geometry, material properties, and loading conditions. Improper design may lead to catastrophic failure due to hinge formation and collapse mechanisms. Classical design methods often rely on conservative empirical rules, resulting in overuse of materials and elevated costs. Finite element analysis (FEA) provides precise modeling but is computationally intensive, limiting its use for iterative optimization or parametric studies.

Machine learning offers a complementary approach, allowing engineers to capture nonlinear relationships between geometry, material properties, and structural performance. By training neural networks on high-fidelity simulated data, one can predict critical structural responses with minimal computational overhead. This approach enables rapid exploration of the design space, making it possible to optimize arch geometries for strength, stability, and material efficiency while considering complex loading scenarios that are analytically intractable.

Scope and Contributions

This project makes the following contributions:

- Development of a multi-output regression framework capable of predicting collapse factors and hinge locations simultaneously.
- Integration of classical mechanics and computational simulations to generate high-fidelity datasets.
- Detailed parametric analysis covering spans from 2 m to 9 m, thickness-to-span ratios from 0.08 to 0.16, and block counts from 8 to 60.
- Evaluation of model performance using MSE, MAE, R^2, and error distribution plots to assess robustness and generalization.

- Implementation of a pipeline combining machine learning predictions with evolutionary optimization strategies, enabling near-real-time structural optimization.

This BTP project demonstrates the feasibility of using data-driven approaches in conjunction with traditional mechanics to enhance the safety, efficiency, and sustainability of masonry structures.

2 Literature Review

Classical Structural Analysis

Limit analysis and the plastic theory of structures have long been employed to understand masonry arches. The concept of plastic hinges and collapse multipliers provides insight into failure mechanisms. Using virtual work principles, one can compute the critical load factor λ corresponding to the formation of a collapse mechanism. Classical formulas, however, are limited to simple shapes and homogeneous loading. In practice, masonry arches are often heterogeneous, and analytical solutions may fail to capture subtle variations in stress distribution.

Computational Methods

Finite element analysis (FEA) has enabled precise modeling of complex geometries and loadings. 3D meshing and nonlinear material modeling allow engineers to predict hinge formation and stress concentrations with high accuracy. However, FEA is computationally expensive for parametric studies, particularly when exploring tens of thousands of design variations.

Optimization Techniques

Optimization algorithms such as genetic algorithms (GA), particle swarm optimization (PSO), and simulated annealing have been used to minimize arch volume while satisfying stress and stability constraints. GAs evolve candidate designs iteratively, approaching a Pareto-optimal front balancing safety and material efficiency. While effective, these approaches often require repeated evaluations of expensive simulations, highlighting the need for surrogate models or predictive frameworks.

Emergence of Machine Learning in Structural Engineering

Machine learning models, particularly multi-output regression networks, have been successfully applied as surrogate models for structural optimization. These models can predict multiple output parameters simultaneously, capturing complex nonlinear dependencies. Recent studies demonstrate that ML models can outperform classical regression in predicting collapse factors, hinge locations, and load distribution patterns, particularly when large synthetic datasets are available for training.

3 Mechanical Principles and Structural Modeling

3.1 Arch Geometry and Nomenclature

The geometric properties of an arch determine stress paths and hinge locations. Important parameters include:

- **Span** s: Distance between supports.
- **Rise** h: Maximum vertical height.
- **Thickness** t: Radial cross-section depth.
- **Number of blocks** n: Discrete masonry elements.
- **Material density** ρ: Affects self-weight.

Typical arch shapes studied include semicircular, segmental, parabolic, and horseshoe. Complex shapes require careful consideration of stress distribution and hinge placement.

3.2 Loading Scenarios

Two main cases are considered:

1. Gravity and seismic loading: Vertical self-weight $F_2 = mg$ combined with horizontal seismic forces F_3, producing combined bending and axial stress.
2. Gravity with external vertical loads: Includes traffic, construction loads F_3, F_4 along with self-weight, generating concentrated stresses at critical points.

3.3 Limit Analysis and Virtual Work

Collapse occurs through the formation of plastic hinges. Using the principle of virtual work, the internal work W_i is equated to external work W_e at collapse:

$$\sum_i P_i \delta x_i = \lambda \sum_j F_j \delta y_j$$

For hinge indices b, c:

$$\text{work}_{ab} = \sum_{i=0}^{b} P_i (x_{c,i} - x_A),$$
$$\text{work}_{bc} = \sum_{i=b+1}^{c} P_i (x_{c,i} - x_{c,b}),$$
$$\text{work}_{cd} = \sum_{i=c+1}^{n} P_i (x_{c,i} - x_D)$$
$$\lambda = \frac{\text{Total internal work}}{W_\ell \Delta y_\ell}$$

The minimum positive λ identifies the critical failure mode.

4 Dataset Generation and Simulation

4.1 Parameter Space Definition

Dataset includes spans 2–9 m, thickness-to-span ratios 0.08–0.16, block counts 8–60, material densities 1900–2500 kg/m^3, and width/load scaling factors 5–8. This generates over 50,000 unique configurations, ensuring comprehensive coverage of practical scenarios.

4.2 Simulation Implementation

Python scripts compute hinge locations and collapse factors using vectorized operations. Missing or anomalous data are filtered, and results are normalized for ML training.

4.3 Data Storage and Format

Dataset stored in CSV format contains columns for all input parameters and outputs (collapse factor, hinge positions). Normalization ensures consistent scaling for efficient neural network training.

5 Machine Learning Model Design

5.1 Feature Engineering and Normalization

All numerical features are standardized to zero mean and unit variance. No categorical variables exist. This ensures uniform convergence rates during training.

5.2 Model Architecture

A multi-output feedforward neural network is implemented:

- Input layer: 8 features.
- Hidden layers: 256, 128, 64, 32, 16 neurons with ReLU activations.
- Dropout of 0.2 applied to prevent overfitting.
- Output layers: three heads predicting collapse factor α_1 and hinge positions B and C.

5.3 Training Details

Model trained using Adam optimizer, learning rate 10^{-3}, and MSE loss summed across outputs. Early stopping and validation splits prevent overfitting. Typical training epochs: 200–300.

6 Model Training and Results

6.1 Data Splits and Validation

Training: 50%, Validation: 20%, Test: 30%. Stratified random sampling preserves feature distribution across splits.

6.2 Epoch-Wise Losses

Table 1: Epoch-wise train and validation loss values for each output variable

Epoch	Train α_1	Train B	Train C	Val α_1	Val B	Val C
10	0.0031	4.0481	32.3394	0.0020	1.8778	14.8913
20	0.0012	1.9239	13.2914	0.0011	0.6272	4.5374
30	0.0009	1.1964	9.0722	0.0006	0.3019	3.0659
40	0.0007	0.7328	5.7236	0.0005	0.1953	2.9069
50	0.0006	0.4622	3.6567	0.0004	0.1103	1.2765
60	0.0005	0.3001	2.3191	0.0004	0.0970	1.1408
70	0.0005	0.2806	1.9807	0.0003	0.0874	0.7187
80	0.0005	0.2246	1.4647	0.0005	0.1328	0.6687

6.3 Final Performance

- Collapse factor (α_1) accuracy: 97.22%
- Hinge B: 98.90%
- Hinge C: 99.37%