International Journal on Engineering Technology (InJET)

www.kec.edu.np/kec-journal/
Volume 2, issue no. 1, Dec 2024, Page 135-150
Received Date: Aug 12, 2024
Accepted Date: Oct 23, 2024
ISSN: 3021-940X (Print)
DOI: https://doi.org/10.3126/injet.v2i1.72562

Leveraging Machine Learning for Structural Response Characterization of Brick Masonry

Suyogya Dahal[1], Rabindra Adhikari[2,*], Dipendra Gautam[3]

[1]*Thapathali Campus, Institute of Engineering, Kathmandu, Nepal, er.suyogyadahal@gmail.com*
[2] *Thapathali Campus, Institute of Engineering, Kathmandu, Nepal, rabindraadhi@gmail.com*
[3]*University of Iceland, Iceland, dig17@hi.is*

Abstract

Advanced techniques like data-driven approaches and machine learning are crucial for understanding and designing resilient masonry buildings to seismic and other hazards. The study highlights the potential of machine learning for low-cost, fast structural assessment of buildings, which will significantly improve the existing vulnerability assessment procedures and increase the reliability of the results at lower investments. This study presents the utilization of machine-learning models to estimate the building's structural response, which is helpful for the vulnerability characterization of the building. Considering the natural period of vibration of the building as an essential structural response parameter, it is predicted using the standard building features collected in post-disaster surveys. This study further analyzes the importance of building features, such as different geometric configurations and material properties, for a building's time-period response. Seven different machine-learning models were trained and evaluated for prediction accuracy using model evaluation metrics such as MAE, MSE, RMSE, and R^2, of which seven models, which gave an R^2 value of more than 0.5, were considered for detailed study. Among various models, the model with the CatBoost algorithm was the best and had the highest model accuracy.

Keywords: Brick masonry, natural time-period, PyCaret, Machine learning, building vulnerability

1. Introduction

Masonry buildings have been the primary building type in Nepal, comprising structures from single units laid and bound with binding mortar. Brick and stone are the most common materials used in masonry construction. During the Gorkha earthquake of 2015, significant damage to such structures was observed. Though recent urban construction prefers steel and RC structures, more than three-quarters of the existing building stocks belong to masonry. With increasing awareness of the vulnerability of different categories of buildings, interest is rising from both government and private sectors to evaluate the vulnerability of the existing buildings and detailed designs for the new buildings, even for the rural masonry buildings. However, detailed engineering analysis of individual buildings is costly in terms of human resources, physical resources for design, and time. There is a trend of conducting field surveys to collect information on building damage and related data after hazards such as earthquakes, which can be used in conjunction with ML models to improve structures' structural response. However, those are focused on aid distribution and not effectively utilized for the future planning of resilient structures.

Masonry is a popular construction technique in Nepal due to its many advantages. However, like with any construction system, the performance of masonry structures to various types of loads, including seismic loads, depends on several structural parameters such as masonry unit type, mortar type, and geometry of the structure, among others. Among others, the seismic performance is mainly dependent on the dynamic characteristics of the building, such as the natural period of vibrations, mode shapes, damping, etc. This study has considered the fundamental period of vibration as a measure of structural response. This study intends to

**Corresponding Author*

apply the benefits of the machine learning method for the prediction of structural response, and hence, once any of the parameters are predicted, this method can be extended for any other response parameters as well.

This research is based on the fact that machine learning is a new concept in the field of earthquake engineering, and only a few works have been carried out to predict the dynamic characteristics and vulnerability of masonry structures using machine learning algorithms. Machine learning, being a highly flexible tool to predict the behavior of any structure based upon previously established behavior, can help to predict even the seismic behavior of any building structure without rigorous engineering analysis.

Some literature introduces machine learning (ML) methods like supervised, unsupervised, and reinforcement learning. There is some literature based on the behavior of steel structures, but barely any literature that focuses on masonry structures' machine learning seismic vulnerability. So, this study will be very useful in establishing a machine learning algorithm that can efficiently determine the seismic vulnerability of existing masonry structures and facilitate the preliminary design of such structures. (Tang et al., 2022) proposed a machine learning-based fast seismic risk assessment framework to ease the computational burden in estimating the potential earthquake-induced loss of a building during its intended life. The hazard parameters of sites and the structural parameters of buildings were incorporated as inputs. For outputs for the regression and classification tasks of supervised learning, the continuous risk values and discrete risk levels were used. A study by Tang et al. (2022) found that the artificial neural networks achieved the lowest root mean square error of 0.0051 for regression and the highest accuracy of 96.8% for classification. (Mangalathu and Jeon, 2018) studied beam-column joints to identify the response mechanism, including the classification of failure mode and the prediction of associated shear strength of beam-column joints with machine learning techniques. In their study, the efficiency of various machine learning techniques was evaluated using extensive experimental data from 536 experimental tests, all of which exhibited either non-ductile joint shear failure prior to beam yielding or ductile joint shear failure after beam yielding. They concluded that lasso regression has better efficiency and reasonable accuracy in classification and prediction.

Machine learning (ML) algorithms can analyze vast structural data, including material properties, load conditions, and environmental factors. By learning from training data, ML models can identify patterns, correlations, and anomalies that might not be apparent through traditional methods. ML enables predictive modeling for structural behavior, such as predicting deflections, stresses, and failure modes. Hence, ML algorithms can optimize structural designs by exploring configurations and selecting the most efficient ones. This leads to cost savings, reduced material usage, and improved safety. Hence, ML can empower engineers to make informed decisions, optimize designs, and enhance structural safety. As research and adoption continue, ML’s potential impact on structural engineering remains substantial.

A study by (Gautam et al., 2016) outlines the commonly observed failure patterns in Nepal's buildings after the 2015 M_w7.8 Gorkha (Nepal) earthquake. Several types of damage patterns were observed for reinforced concrete buildings and for unreinforced masonry and adobe houses during the reconnaissance survey performed immediately after the earthquake. Several field visits in the affected districts including, non-engineered buildings, middle and high-rise buildings, commercial complexes, administrative buildings, schools, and other critical facilities, were conducted, and associated failure/damage patterns were identified and analyzed. The construction and structural deficiencies were identified as the major causes of failure. However, the structural characteristics have remained the most important parameter determining its response in seismic events.

According to the National Planning Commission (2015), the total number of fully damaged buildings in the Gorkha Earthquake 2015 was determined to be 4,98,852, with the number of partially damaged buildings being 2,56,697. Among them, low-strength masonry buildings accounted for 95% of the fully damaged buildings (4,74,025) and 67.7% of the partially damaged buildings (1,73,867), and Cement based masonry buildings accounted for 3.7% of the fully damaged buildings (18,214) and 25.6% of the partially damaged buildings (65,859) with 8790 fatalities, 22,300 injuries. They affected 8 million people from 31 out of 75 districts in Nepal (NPC, 2015). This shows that Un-reinforced (URM) structures are the most vulnerable during earthquakes. It is interesting to note that significant effort was made to conduct the field surveys and generate the above data. However, the optimum utilization of those data can be very useful in future disaster management activities, and predicting the response of such buildings in future earthquakes is the most important one. This can be facilitated by leveraging the use of machine-learning techniques.

Building inspections by Dizhur et al. (2016) during the Gurkha Earthquake have shown that Clay brick URM construction practice was widely variable amongst inspected buildings. In Nepal, mud mortar was typically manually mixed on-site using shovels, hands, and feet. 'Raw' clay bricks (adobe, sun-dried only) were used in URM construction to reduce material cost and improve thermal properties.

During the majority of the field visits, visual inspection and some short data entry are preferable for identifying probable damages. While this alone may not be sufficient to develop a proper action plan for the local authority or the government, with the help of machine learning models, we can get important insight into the dynamic behavior of buildings from such data entries.

Data from earthquakes in Nepal shows that un-reinforced masonry (URM) structures are the most vulnerable during earthquakes. Studying the types of masonry structures and their performance has considerable scope and is needed in Nepal. So, it is important to access data and take precautional measures for the seismic vulnerability of many URM structures. So, to analyze such a vast number of structures, only employment of FEM modeling for detailed engineering analysis for characterization of structures will take a significant number of workforce and resources. Since it is very time consuming, tedious, and impractical to carry out such detailed analysis on all of the structures, this is the area where machine learning algorithms can help us to predict the structural response of these building structures, which can further be extended to assess the vulnerability of such structures during a large earthquake or other hazards.

2. Materials and Methods

2.1. Machine learning model

According to (Mitchell, 1997), “a computer program is said to learn from experience “E” with respect to some task “T” and performance measure “P”, if its performance on “T”, as measured by “P”, improves with experience “E””. Machine learning model is an algorithm that can predict defined parameters, based on a set of inputs, the prediction is based on the training of the model with the known set of inputs-outputs. Regression is a simple, common method to establish the relationship between inputs and outputs employed in machine learning models.

2.1.1. Regression analysis

Regression analysis is a statistical method to estimate relationships between a dependent variable and one or more independent variables. It helps us understand the strength of the relationship between variables and model the relationship between them. There are many variations of regression analysis, such as linear, multiple linear, and nonlinear analysis.

2.1.2. Gradient Boosting Algorithm

Gradient Boosting is a robust boosting algorithm that combines several weak learners into strong learners, in which each new model is trained to minimize the loss function, such as mean squared error or cross-entropy of the previous model using gradient descent. In each iteration, the algorithm computes the gradient of the loss function with respect to the predictions of the current ensemble and then trains a new weak model to minimize this gradient. The new model's predictions are then added to the ensemble, and the process is repeated until a stopping criterion is met.

There is a technique called Gradient-boosted Trees, whose base learner is CART (Classification and Regression Trees). Figure 1 explains how gradient-boosted trees are trained for regression problems.

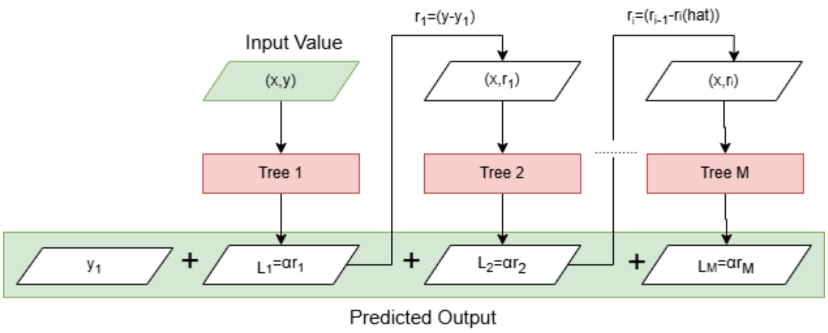


Figure 1. Gradient Boosted Trees for Regression

The ensemble consists of M trees. Initial tree is trained using the feature matrix x and the labels y. The predictions labeled y_1(hat) are used to determine the training set residual errors r_1. The second tree is then trained using the feature matrix x and the residual errors r_1 of the initial tree as labels. The predicted results r_1(hat) are then used to determine the residual r_2. The process is repeated until all the ensemble's M trees are trained. Prediction of each tree in the ensemble is shrunk after it is multiplied by the learning rate (η), which ranges between 0 and 1. There is a trade-off between η and the number of estimators; a decreasing learning rate provides higher accuracy by increasing estimators in order to reach a certain model performance. Each tree predicts a label, and the final prediction is given by equation (1).

$$y(pred) = y_1 + (\eta * r_1) + (\eta * r_2) + \ldots\ldots + (\eta * r_N) \quad \text{(Equation 1)}$$

where,
y(pred) – initial prediction
y_1 – initial prediction
η – learning rate
r_N- N^{th} residual

Steps involved in Gradient Boosting Algorithm

Step 1: Let x and y be the input and target with N no of samples. To obtain the function f(x) that maps the input features x to the target variables y. The difference between the actual and the predicted variables is shown by equation (2), which is also known as the loss function.

$$L(f) = \sum_{i=1}^{N} L(y_i, f(x_i)) \quad \text{(Equation 2)}$$

where,
L(f) – loss function
y_i – target variable
x_i – input variable
f(x)- a function that maps input to the target
N- number of samples

Step 2: Loss function L(f) is minimized with respect to f, using equation (3).

$$\widehat{f_0}(x) = \underset{f}{\operatorname{argmin}}\, L(f) = \underset{f}{\operatorname{argmin}} \sum_{i=1}^{N} L(y_i, f(x_i)) \quad \text{(Equation 3)}$$

where,
y_i – target variable
x_i – input variable
f(x)- a function that maps input to the target
N- number of samples

Considering the gradient boosting algorithm in M stages, then, to improve the F_m, the algorithm can add some new estimator h_m as having $1<m<M$, as shown in equation 4.

$$\hat{y}_i = F_{m+1}(x_i) = F_m(x_i) + h_m(x_i) \qquad \text{(Equation 4)}$$

where,
y_i – target variable
x_i – input variable
m- stage in gradient boosting algorithm
h_m- estimator

Step 3: Steepest Descent

The steepest descent $h_m = -\rho_m.g_m$, where ρ_m is constant and known as step length, and g_m is the gradient of loss function L(f), defined by equation (5).

$$g_{im} = -\left[\frac{\partial L(y_i, f(x_i))}{\partial f(x_i)}\right]_{f(x_i)=f_{m-1}(x_i)} \qquad \text{(Equation 5)}$$

where,
y_i – target variable
x_i – input variable
m- stages in gradient boosting algorithm
g_m- gradient of loss function L(f)

Step 4: Solution

The gradient similarity for M trees can be expressed by equation 6.

$$f_m(x) = f_{m-1}(x) + \left(\underset{h_m \in H}{\operatorname{argmin}} \left[\sum_{i=1}^{N} L\left(y_{i,} f_{m-1}(x_i) + h_m(x_i)\right)\right]\right)(x) \qquad \text{(Equation 6)}$$

where,
y_i – target variable
x_i – input variable
m- stage in gradient boosting algorithm
h_m- estimator

The solution will be expressed by equation (7).

$$f_m = f_{m-1} - \rho_m g_m \qquad \text{(Equation 7)}$$

where,
ρ_m is constant
g_m- gradient of loss function L(f)

2.1.3. XGBoost Algorithm

XGBoost is a popular and efficient open-source implementation of the gradient-boosted trees algorithm. Gradient boosting is a supervised learning algorithm that attempts to accurately predict a target variable by combining the estimates of a set of simpler, weaker models (Chen and Guestrin, 2016). According to them "Among the machine learning methods used in practice, gradient tree boosting is one technique that shines in many applications. Tree boosting has been shown to give state-of-the-art results on many standard classification benchmarks".

When using gradient boosting for regression, the weak learners are regression trees, and each regression tree maps an input data point to one of its leaves. The training proceeds iteratively, adding new trees that predict the residuals or errors of prior trees, which are then combined with previous trees to make the final prediction. It's called gradient boosting because it uses a gradient descent algorithm to minimize the loss when adding new models. Mathematically, XGBoost can be represented by equation (8).

$$F_m(x) = F_{m-1} + \alpha_m h_m(X, r_{m-1}) \qquad \text{(Equation 8)}$$

where,
α_i -regularization parameters and residuals computed with the i^{th} tree
τ_i - residuals computed with the i^{th} tree
h_i - function trained to predict residuals

To compute α_i, we use the residuals computed, r_i, and compute the following in equation (9)

$$\underset{\alpha}{\operatorname{argmin}} = \sum_{i=1}^{m}(L(y_i, F_{i-1}(X_i) + \alpha h_i(X_i, r_{i-1})) \quad \text{(Equation 9)}$$

where,
L(y,f(x)) – differentiable loss function

2.1.4. CatBoost Algorithm

CatBoost is an open-source boosting library developed by Yandex, designed for use on problems with many independent features.Catboost is a variant of gradient boosting that can handle both categorical and numerical features. It does not require any feature encoding techniques like One-Hot Encoder or Label Encoder to convert categorical features into numerical features. It also uses a symmetric weighted quantile sketch (SWQS) algorithm, which automatically handles the missing values in the dataset to reduce overfitting and improve its overall performance.

Given a training dataset with N samples and M features, where each sample is denoted as (x_i, y_i), as x_i is a vector of M features and y_i is the corresponding target variable, CatBoost aims to learn a function F(x) that predicts the target variable y. Mathematically, CatBoost can be represented by equation (8).

$$F(x) = F_0(x) + \sum_{i=1}^{M}\sum_{i=1}^{N} f_m x(i) \quad \text{(Equation 10)}$$

where,
F(x) - overall prediction function.
$F_0(x)$ - initial guess or the baseline prediction
$\sum_{m=1}^{M}$. - summation over the ensemble of trees. M - number of trees in the ensemble.
$\sum_{n=1}^{N}$. - summation of the training samples. N - number of training samples.
$f_m(x_i)$ - prediction of the m-th tree for the i-th training sample.

According to the equation, the overall prediction F(x) is obtained by summing up the initial guess $F_0(x)$ with the predictions of each tree $f_m(x_i)$ for each training sample. This summation is performed for all trees (m) and all training samples (i).

2.2. Selection of ML-models

There are various ML-models being used in civil engineering ML-predictions. Preliminary evaluation of twenty open-source popular ML algorithms were done considering R-square as model evaluation metrics. Among them, seven best algorithms are used of detailed evaluation using more metrices that are defined in the following sections.

2.3. Training for ML-model

2.3.1. Training data Set

A training data set is a data set used during the learning process and is used to fit the parameters. For regression analysis, a learning algorithm looks at the training data set to determine the optimal combinations of variables that will generate a reasonable predictive model.

2.3.2. Test data set

A test data set is a data set that is used to access the performance of a regression model which are independent of the training data set but are taken from same data set and split into training and test data, usually 70% of data are taken as train set and 30% of data are taken as test set. This ensures that test data follows the same probability distribution as the training set.

2.3.3. Training methodology

This study is based on a parametric variation of one-, two---, and three-story unreinforced masonry buildings based upon various building parameters like mortar type, opening, slab system aspect ratio, partition type, etc., considered as input parameters of analysis. The Pycaret library is used for machine learning, which is an open-source, low-code machine learning library in Python that automates machine learning workflows.

508 sample masonry buildings were analyzed in the FEM-based software SAP2000. Modal analysis of buildings was carried out to determine the fundamental time period of vibration of individual structures, which are considered output parameters of the analysis. Then, the input parameters and output parameters were grouped to form a data set for each analysis. These data sets were recorded in tabular format and fed to Pycaret to find the best machine-learning algorithm to predict the response of the URM based on input parameters.

Various machine algorithms were compared using Pycaret to get the best result based on the model performance as measured by Mean absolute error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), and Coefficient of determination (R^2). The best model was evaluated based on residual error, prediction error, feature importance, etc. If the performance is unsatisfactory, the training data must be increased to further train the model. Finally, the best model was used to create the final model, which was used to predict the structural response of various URM based on parametric variations only. This process is depicted in Figure 2.

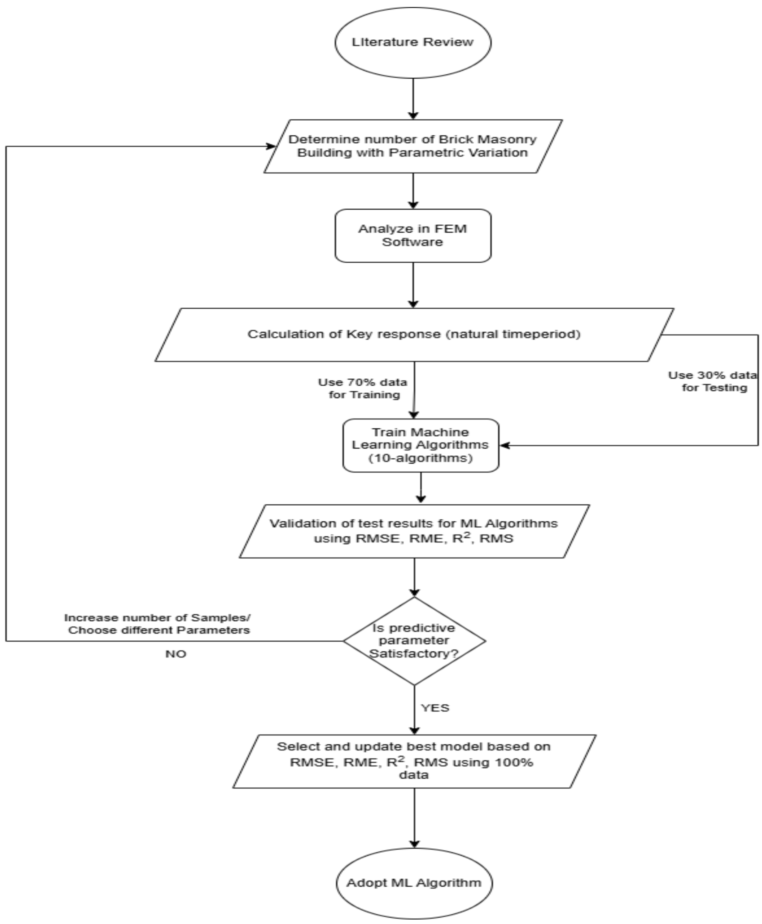


Figure 2. Flowchart of Methodology

2.3.4. Model evaluation

The MSE, MAE, RMSE, and R-squared metrics are mainly used to evaluate the prediction error rates and model performance in regression analysis. The major metrics used in model evaluation are listed as follows:

- MAE (Mean absolute error) represents the difference between the original and predicted values extracted by averaging the absolute difference over the data set.
- MSE (Mean Squared Error) represents the difference between the original and predicted values, which is obtained by squaring the average difference over the data set.
- RMSE (Root Mean Squared Error) is the error rate by the square root of MSE.
- R-squared (Coefficient of determination) represents the coefficient of how well the values fit compared to the original values. The values from 0 to 1 are interpreted as percentages. The higher the value is, the better the model is.

The above metrics can be mathematically expressed in equations 11 to 14.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} |y_i - \widehat{y}| \qquad \text{(Equation 11)}$$

$$MSE = \frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2 \qquad \text{(Equation 12)}$$

$$RMSE = \sqrt{MSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N} (y_i - \hat{y})^2} \qquad \text{(Equation 13)}$$

$$R^2 = 1 - \frac{\sum (y_i - \hat{y})^2}{\sum (y_i - \bar{y})^2} \qquad \text{(Equation 14)}$$

where,
$\hat{y} - predicted\ value\ of\ y$
$\bar{y} - mean\ value\ of\ y$

2.4. Database of structural characteristics

The accuracy of ML algorithms is highly dependent upon data availability for training and testing purposes. Since ML uses previous experience with data to predict results, the performance of such algorithms highly depends on the availability of prior data that it uses to train itself. The performance of ML models increases with the increase of data for training and testing purposes, so high-quality data is essential for training accurate ML models. The data set in this study consists of input data (building materials and geometry-related data) and output data (period of vibration of the building). Various research paper was studied for consideration of the material properties of buildings in this study, of which Kaushik, Rai, and Jain, 2007 were found to be the most relevant data source for our research as they considered various grades of brick cement mortar, which fulfilled our requirement of mortar variation. Further, based on site visits, related research papers, study of images and data from multiple online resources, and author’s experience in building design and construction industry, different geometric configuration and material properties of the buildings are defined for this study that reflects the actual construction practices thus generating high-quality and most relevant data to machine learning algorithms for high accuracy of ML models.

2.4.1. Building input parameters

508 sample masonry buildings were analyzed in the FEM-based software SAP2000. Modal analysis was carried out to determine the fundamental time period of vibration of the building for individual structures, which were considered the output parameters of the analysis.

A total of 508 building models were prepared for the detailed analysis of the data generation with different material and geometric configurations defined by input parameters. The plan configurations adopted in the models are presented in Table 1. Structural analysis of the all buildings was carried out in FEM software for each model to record the output of each building. Parameters for each building type were arranged in row and column format. Length, Breadth, Aspect Ratio, Slab Type, opening percentage, Regularity, no of stories, and Mortar type were considered as input parameters, and the natural time period was set as the target parameter in PyCaret (PyCaret is a simple, easy-to-learn, low-code machine learning library in Python). PyCaret was set up in Python Jupyter Notebook to find the best predictive model. The parameter types of all input parameters are as presented in Table 2.

Table 1. Plan Configuration of various building types

SN	Length(m)	Breadth(m)	Aspect Ratio
1	6	6	1
2	8	6	1.33
3	8.22	7.92	1.03
4	10	8	1.25
5	7.92	7.62	1.03
6	12.8	7.31	1.75
7	9.14	5.48	1.66
8	9.14	7.01	1.30
9	12	6	2
10	14.63	9.14	1.59
11	5.18	3.96	1.30
12	6.70	3.96	1.69
13	11.88	4.26	2.78
14	11.88	5.18	2.29

Table 2. Parameters used as input for ML models

SN	Parametric Variations	Parameter Type
1	Length	Numerical
2	Breadth	Numerical
3	Aspect Ratio	Numerical
4	Minimum number of bays	Numerical
5	Maximum opening	Categorical
6	Opening Percentage	Numerical
7	Plan Irregularity	Categorical
8	Storey	Numerical
9	Slab type	Categorical
10	Band	Categorical
11	Mortar Thickness	Numerical
12	Modulus of Elasticity	Numerical

2.4.2. Building response parameters

This study takes the fundamental natural time period of vibration “T_n” as the building response or output parameter. The time period of vibration in a multi-degree freedom system such as multi-storied buildings is the time required for one cycle of oscillation in one of these natural modes. The fundamental time period is the time period for the mode of vibration with the largest period of vibration. The period of vibration can be expressed as a cyclic frequency or circular frequency according to equations 13 and 14.

$$\omega_n = \frac{2\pi}{T_n}$$ (Equation 15)

$$f_n = \frac{1}{T_n}$$ (Equation 16)

where,
T_n - natural period of vibration
ω_n - natural circular frequency of vibration
f_n - natural cyclic frequency of vibration

The fundamental natural time period of vibration is considered one of the most important dynamic characteristics of any building. It determines its response to dynamic loads such as earthquakes and gives valuable insight into the building's seismic performance.

2.4.3. Generation of data sets

The data sets required for model training and modal performance evaluation are prepared by combining input parameters and response parameters that are determined as discussed in the preceding sections. Response (output) parameters for the data sets are prepared through detailed FEM analysis of each case of the building models using commercial FEM software SAP2000v23 by CSI.

Material properties were defined for brick masonry with mud and cement mortar using the parameters mentioned below in Table 3. The material property for a rigid floor was approximated using M20 concrete, and the property of wooden material was approximated for a flexible floor joist with wooden flooring.

Table 3. Modulus of Elasticity for Brick Cement Mortar and Brick Mud Mortar

Mortar Type	**Modulus of Elasticity**
Cement Sand Mortar (1:6)	230 MPa
Cement Sand Mortar (1:3)	380 MPa
Cement Sand Mortar (1:0.5:4.5)	420 MPa
Brick in Mud Mortar	29.145MPa

Source: (Kaushik, Rai and Jain, 2007)

Shear Modulus (Gm) is calculated using relation,

$$G_m = \frac{E_m}{2(1+\vartheta_m)}$$ (Equation 17)

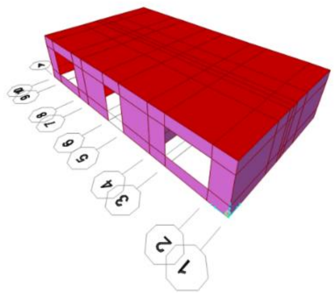
where,
G_m - Shear modulus
E_m - elastic modulus
ϑ_m - poisson's ration

In the FEM model, the thin-shell element was adopted to model for masonry walls, and rigid floor systems were also modeled using thin-shell elements. Flexible floors were modeled using membrane elements. The base was fixed at the plinth level. To confirm the integrity of the system, shell meshing was done in 0.5mx0.5m size.

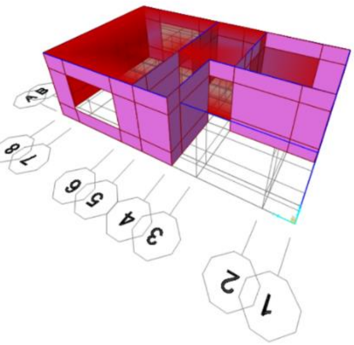
The time period was calculated using modal analysis of the building in which a total of 12 modes were recorded. On the floors, 2 kN/m^2 of area load was considered for live load, and 1.5 kN/m^2 of live load was considered. The time period obtained for maximum mass participation was recorded along with the drift of the structures. Table 3.3 shows the plan configuration of buildings used for analysis in SAP2000. Table 3.5 shows various machine learning models used for comparison of time periods.

Table 4. Live load on the building

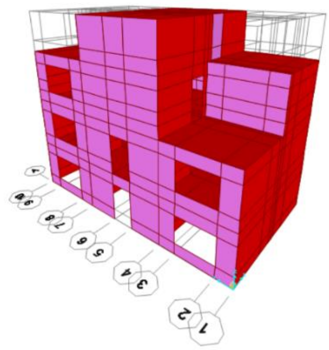
Description of Load	**Load (kN/m2)**
Roof Live Load	1.5
Floor Live Load	2



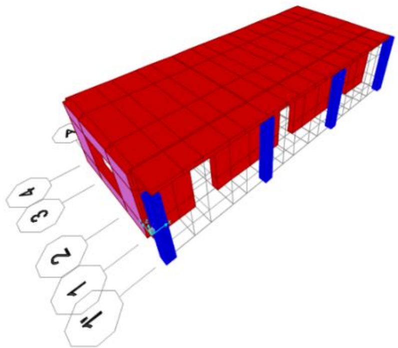
CEMENTLOWS006



CEMENTFLEXS009



CLAY008



CEMENTLOWS018

Figure 3. Few FEM models of buildings used for analysis

3. Results and Discussions

Twenty different ML algorithms were used to train ML models using training datasets, among with best seven models (based on evaluation metrics) are shown in Table 5. Four model evaluation matrices, MAE, MSE, RMSE, and R^2, were used to validate the accuracy of the ML models. Further, the models were used to predict the output from test data sets as well for the model evaluation. The results are compared and discussed in the following sections.

Table 5. Machine learning models used for prediction

SN	ML Models
1	CatBoost Regressor
2	Extreme Gradient Boosting
3	Light Gradient Boosting Regressor
4	Gradient Boosting Regressor
5	Random Forest Regressor
6	Extra Tree Regressor
7	Decision Tree Regressor

3.1. Comparative performance of various ML models

Each machine learning algorithms were used to predict the time period of masonry buildings, and the results thus obtained were verified using corresponding output data from the datasets. This study found that the predicted values by various machine learning algorithms differ to various degrees. CatBoost Regressor, Extreme Gradient Boosting, Light Gradient Boosting Regressor, and Gradient Boosting Regressor were some of the best-fitting algorithms for this study, according to the value of model evaluation metrics as presented in Table 6. The first two models have MAE less than 0.5 and R^2 more than 0.95, indicating the best models. Catboost regressor was then selected as the best ML model for further assessment.

Table 6. Model evaluation metrics of various ML models for the natural time period

SN	ML Models	MAE	MSE	RMSE	R2

1	CatBoost Regressor	0.041	0.005	0.070	0.965
2	Extreme Gradient Boosting	0.045	0.006	0.079	0.953
3	Light Gradient Boosting Regressor	0.057	0.008	0.088	0.945
4	Gradient Boosting Regressor	0.059	0.009	0.095	0.937
5	Random Forest Regressor	0.058	0.011	0.099	0.927
6	Extra Tree Regressor	0.058	0.014	0.109	0.911
7	Decision Tree Regressor	0.066	0.018	0.124	0.885

3.2. Evaluation of CatBoost Regressor model

The CatBoost Regressor was selected as the best ML model based on the model evaluation metrics. As the evaluation is satisfactory, this model's efficiency is further improved by using the test data for training purposes, which increases the training datasets. Accordingly, the prediction errors were reduced significantly with an increased R^2 of 0.992, as depicted in Table 7, which is a very good fit.

Table 7. Model evaluation metrics of CatBoost Regressor for the time period after model finalization

SN	ML Models	MAE	MSE	RMSE	R2
1	CatBoost Regressor	0.015	0.001	0.034	0.992

3.2.1. Residuals

Residuals are the deviation of the predicted value from the expected value in training datasets. Figure 5 shows the residuals for the natural time period of train and test data, where the time period is plotted on the x-axis as the predicted value and the residuals on the y-axis. The distribution chart on the right shows an excellent fit of the data with insignificant residuals for most of the data.

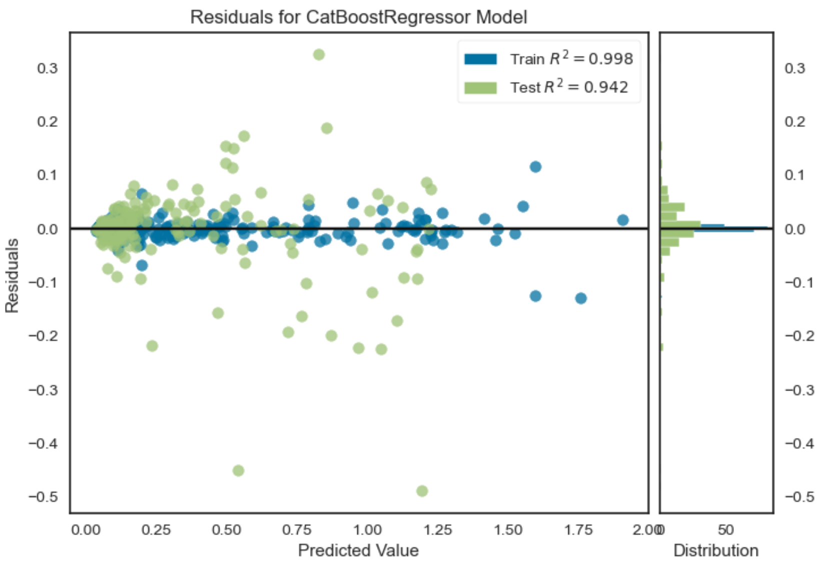


Figure 4. Residuals for Natural Time Period Prediction of Masonry Buildings

3.2.2. Prediction errors

Prediction error represents the correlation between the predicted value by the ML model from the expected value (obtained from FEM analysis). It is presented as a plot of these two values and compared with the identified line; the closer the data are to the identity line, the better the approximation and the fit is determined by R^2. The R^2 value of 1 corresponds to a complete correlation, whereas an R^2 value of 0 shows no correlation. Figure 6 shows the prediction error for a natural time period of test data, y representing the expected time period and ŷ representing the predicted time period. As evaluated by R^2, the fit is 0.942 for the test data, 0.998 for the train data, and 0.992 for the overall data.

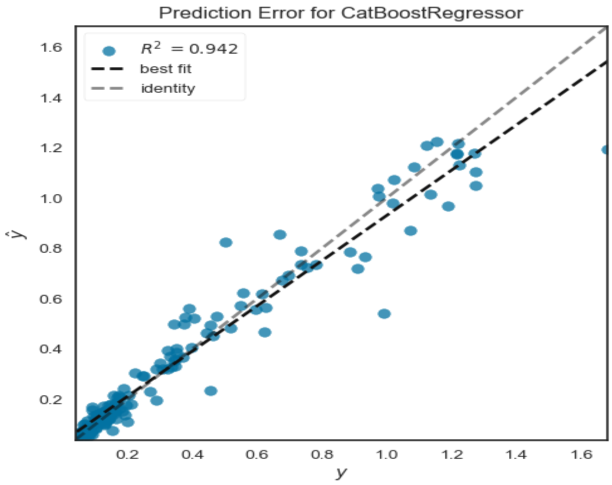


Figure 5. Prediction error in natural time-period

Figures 7 to 10 present a plot of predicted vs. expected value (obtained from detailed engineering analysis) for natural time periods.

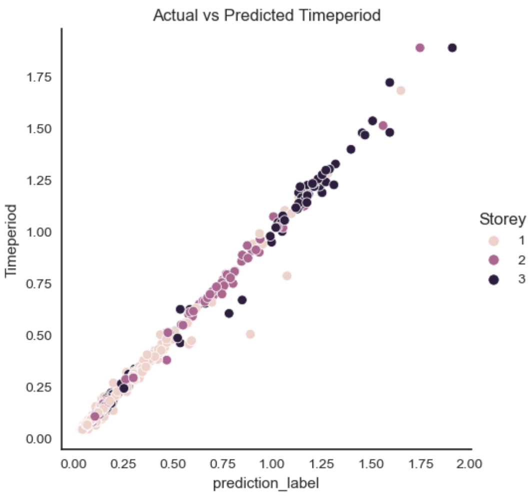


Figure 6. Prediction result comparison for the natural time period for a different story

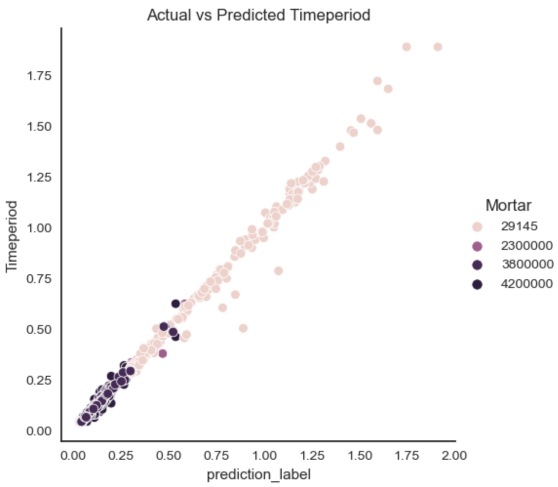


Figure 7. Prediction result comparison for the natural time period for different mortar type

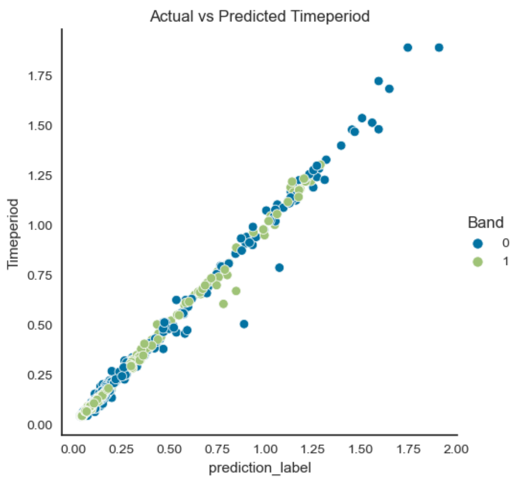


Figure 8. Prediction result comparison for the natural time period for different band types

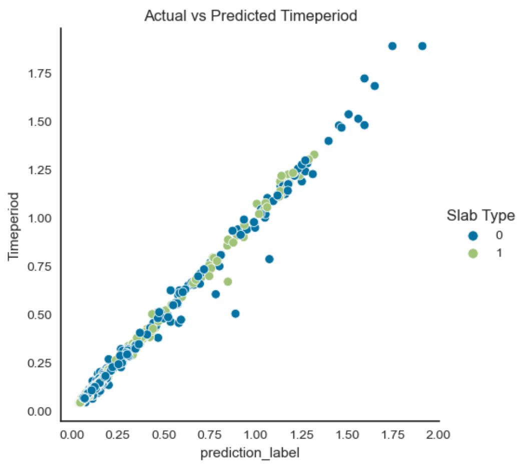


Figure 9. Prediction result comparison for the natural time period for the different slab type

The figures showed that, except for a few limited datasets, most data fit the identity line well, meaning that the prediction is pretty good. Further, table 7 indicated that the CatBoost algorithm predicted the time period of the masonry building with an RMSE of 0.034 and an R^2 value of 0.992 for URM with provided geometric and material properties. Thus, if the geometric and material properties of any URM buildings are within the limits of this study, the time period of such buildings can be predicted with good accuracy.

3.2.3. Feature importance

One of the study's most important insights is the feature importance plot for the predicted value, i.e., natural time period. It provides information on the most important parameters affecting the output and determines the weightage of individual input parameters or features.

As shown in Figure 7, a brick masonry structure's natural time period depends on mortar type, number of stories, and the presence of band and slab types in decreasing order. The dependence of natural time period is most evident on mortar type because there is a significant change in the stiffness of the building with cement and mud mortar types.

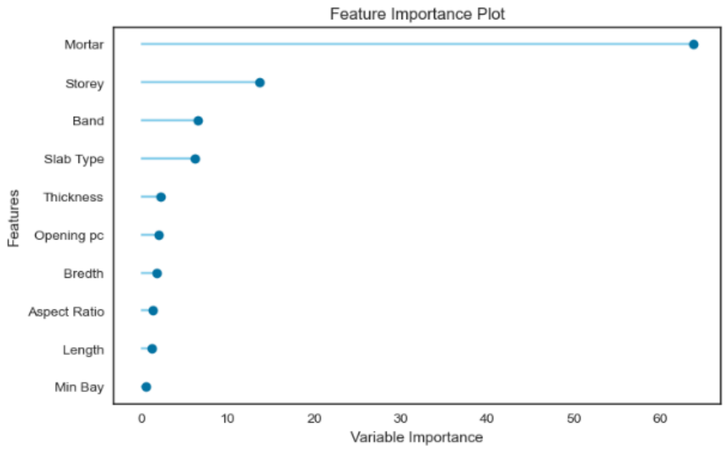


Figure 10. Feature importance for time period prediction

Similarly, the number of stories is the second most important feature, which is well justified in line with several empirical relations for natural periods dependent on the height of the building, such as in NBC105:2020 and IS1893:2016, among others.

4. Conclusions

This study presented the effective utilization of a machine learning algorithm for predicting the structural response parameter of brick masonry buildings. Detailed structural analysis of structures has always been sought as a most engaging and challenging job requiring significant expertise, time, and effort. Thus, such analysis is very limited in the context of low-cost housing or mass surveys. This attempt to use ML models successfully predicts the structural response, which has massive potential to expand to other necessary parameters of structural response meaningful for vulnerability evaluations or structural designs.

This study found that the trained ML model with the CatBoost algorithm has a very good prediction capability for the natural period of the masonry building. About 400 training datasets provided descent training to the model. Seven ML models were trained and evaluated with different evaluation metrics, and six of them performed well with R^2 above 90%. The CatBoost algorithm was the best as measured by evaluation metrics, with an RMSE of 0.034 and an R^2 value of 0.992.

Hence, CatBoost Regressor can be effectively utilized to predict structural response. The prepared model is able to predict the natural time period of vibration of brick masonry buildings with decent accuracy. This model can be easily extended by training other important structural response parameters to predict those parameters. Current study is limited to predict a single output, the natural period of the building which is an important dynamic property of the building based on limited number of inputs. This good agreement of the results supports that this method can be expanded to predict other features of the buildings as well.

Further, evaluating the feature importance for the trained model, the time period of URM was primarily dependent upon mortar type, followed by the number of stories of the building, then the presence of bands, slab types, and so on. Hence, ML is helpful in predicting the output parameters, and the determined feature importance gives advantageous insights on the structural characteristics' sensitivity to different buildings' parameters.

Various kinds of literature have shown the time period's dependency on the building's stiffness and mass. Further empirical formula from building codes also relates the building's plan configuration, height, and building type for the calculation of the fundamental time period of the building. This study not only confirms the dependency of fundamental time period on configuration, height, and type of building but also predicts the time period of masonry buildings, an important dynamic property of building, with high accuracy and close to FEM models. So, we can conclude that ML models can be used for the prediction of the dynamic response of masonry buildings. ML models can also be useful in predicting other responses of civil engineering structures with ease if properly deployed, which can be of huge importance as a research tool in the context of civil engineering.

Though ML can help us characterize building response, it cannot justify its result on its own and needs some human interpretation. Since ML uses previous experience with data to predict results, the performance of such algorithms highly depends on the availability of prior data that it uses to train itself, so we need to provide it with sufficient volume of high quality data for the training process and which may be considered as a limitation of ML. Depending on the selected output, several input parameters are important to properly define the output; if we are not able to properly identify the input parameter for the prediction of output parameters, there can be inconsistency in the predicted output, which is also a limitation of ML and hence good judgement knowledge is required for defining the inputs and outputs for the optimum performance of ML.

Due to the flexibility of ML, it has huge potential in every civil engineering discipline. Further research can be carried out using ML to characterize the structural response of other structures or any part of a structure. Custom ML models other than already available can also be developed for further enhancing the output.

Acknowledgment

The authors acknowledge Thapathali Campus for providing the necessary support to conduct this study as a Master's degree thesis in earthquake engineering.

References

Chen, T. and Guestrin, C. (2016) ‘XGBoost: A Scalable Tree Boosting System’, *Knowledge Discovery and Data Mining*, 13-17-August-2016, pp. 785–794. Available at: https://doi.org/10.1145/2939672.2939785.

Dizhur, D. *et al.* (2016) ‘Performance of unreinforced masonry and infilled RC buildings during the 2015 Gorkha, Nepal earthquake sequence’, *Brick and Block Masonry: Trends, Innovations and Challenges - Proceedings of the 16th International Brick and Block Masonry Conference, IBMAC 2016*, pp. 2399–2408. Available at: https://doi.org/10.1201/B21889-314.

Gautam, D. *et al.* (2016) ‘Common structural and construction deficiencies of Nepalese buildings’, *Innovative Infrastructure Solutions*, 1(1). Available at: https://doi.org/10.1007/s41062-016-0001-3.

GeeksforGeeks (2020). Gradient Boosting in ML. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/ml-gradient-boosting/?ref=lbp [Accessed 28 October 2024].

GeeksforGeeks (2021). CatBoost in Machine Learning. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/catboost-ml/?type=article&id=545413 [Accessed 28 October 2024].

Kaushik, H.B., Rai, D.C. and Jain, S.K. (2007) 'Stress-Strain Characteristics of Clay Brick Masonry under Uniaxial Compression', *Journal of Materials in Civil Engineering*, 19(9), pp. 728–739. Available at: https://doi.org/10.1061/(ASCE)0899-1561(2007)19:9(728).

Mangalathu, S. and Jeon, J.S. (2018) 'Classification of failure mode and prediction of shear strength for reinforced concrete beam-column joints using machine learning techniques', *Engineering Structures*, 160, pp. 85–94. Available at: https://doi.org/10.1016/j.engstruct.2018.01.008.

Mitchell, T. (1997) 'Machine Learning (McGraw-Hill International Editions Computer Science Series): Tom M. Mitchell: 9780071154673', p. 414.

Tang, Q. *et al.* (2022) 'Machine Learning-Based Fast Seismic Risk Assessment of Building Structures', *Journal of Earthquake Engineering*, 26(15), pp. 8041–8062. Available at: https://doi.org/10.1080/13632469.2021.1987354.