INTELLIGENT BUILDING STRUCTURAL HEALTH MONITORING USING MACHINE LEARNING

Banoth Kiran Kumar, S. Gouthami
Department of Electronics and Communication Engineering
Jayamukhi Institute of Technological Sciences, Narsampet, Warangal, Telangana, India

Abstract
Structural health monitoring (SHM) is essential for assessing the safety, reliability, and serviceability of buildings exposed to ageing, operational loads, environmental effects, and natural hazards. Periodic visual inspection and manual testing may not continuously capture progressive or early-stage changes in structural behaviour. This paper presents an intelligent building SHM framework that combines sensor-based structural-response acquisition, signal preprocessing, feature extraction, feature selection, and conventional machine-learning classification. Accelerometers and vibration sensors are considered for dynamic response monitoring, while strain gauges and displacement sensors provide complementary measurements at critical locations such as beams, columns, slabs, and joints. The acquired signals are filtered, detrended, normalized, and divided into fixed-length segments. Nine representative time- and frequency-domain characteristics are described, including mean, variance, standard deviation, root mean square (RMS), peak amplitude, crest factor, dominant frequency, spectral energy, and bandwidth. The resulting feature vectors are processed using Logistic Regression, k-Nearest Neighbors (k-NN), Naïve Bayes, Decision Tree, and Support Vector Machine (SVM) classifiers to distinguish Healthy/Normal from Abnormal/Damaged conditions. The project report lists SVM as the best-performing classifier, with 96.10% accuracy, 95.80% precision, 96.40% recall, and 96.10% F1-score. A Raspberry Pi-based data-acquisition implementation using an ADS1115 analog-to-digital converter framework emphasizes interpretable feature-based learning and potential edge deployment for continuous building-condition assessment.

Keywords: Structural health monitoring; vibration-based damage detection; machine learning; feature extraction; accelerometer; Raspberry Pi; edge computing; building safety.

1. Introduction

Structural health monitoring is an engineering approach for evaluating the safety, integrity, serviceability, and durability of civil infrastructure. Buildings are exposed to ageing, repeated operational loads, earthquakes, strong winds, and environmental effects. Conventional inspection commonly relies on periodic visual assessment and manual testing. Although useful, these methods are time-consuming, subjective, and may not reliably reveal hidden or early-stage damage [1], [2]. Low-cost sensors and intelligent data-processing methods enable continuous measurement of structural response. Accelerometers and vibration sensors capture dynamic behaviour, while strain gauges and displacement sensors provide information about deformation and movement. Sensors can be installed at critical structural locations, including beams, columns, slabs, and joints. The measured responses contain information related to stiffness, damping, vibration characteristics, and possible structural deterioration. Continuous sensing generates substantial volumes of data. Manual interpretation is difficult, and fixed thresholds may not distinguish damage-related changes from normal variations caused by temperature, humidity, wind, traffic, occupancy, and other operational factors. Machine learning (ML) offers a data-driven means of learning relationships between extracted response features and known structural conditions. This work focuses on conventional ML algorithms with compact feature vectors to balance classification capability, interpretability, and computational requirements.

The problem addressed is the development of an automated and computationally efficient framework that processes sensor-derived structural-response data and classifies a building condition as Healthy/Normal or Abnormal/Damaged. The objectives are to acquire structural response from relevant sensors; preprocess and segment the signals; extract and select time- and frequency-domain features; train and compare five conventional ML classifiers; and support dashboard visualization and alert generation for abnormal behaviour.

2. Related Work

Research in SHM has progressed from visual inspection, non-destructive testing, and modal analysis toward sensor-based monitoring and data-

Doi:10.48047/ijiee.2026.16.2.69

driven damage assessment. Vibration-based approaches are important because changes in stiffness, mass distribution, damping, or structural connectivity can influence natural frequencies, mode shapes, and measured vibration responses. Avci et al. reviewed vibration-based damage detection from traditional methods to ML and deep-learning applications [6]. Machine learning has been applied to building and bridge monitoring for supervised damage classification and related condition-assessment tasks. Gomez-Cabrera and Escamilla-Ambrosio reviewed ML techniques for building and bridge SHM [8]. Conventional classifiers such as Logistic Regression, k-NN, Naïve Bayes, Decision Trees, and SVMs can operate on engineered statistical and frequency-domain features. Their relatively modest computational requirements make them candidates for embedded and edge-based implementations. Deep-learning methods, including convolutional and recurrent architectures, can learn representations from high-dimensional signals, but typically require more data and computational resources. Reviews of deep-learning SHM and unsupervised vibration-based monitoring also identify data availability, environmental variability, and generalization as important challenges [2], [9], [11]. Edge SHM research has emphasized low-power wireless sensing and local processing to reduce communication requirements [3], [12].

3. Proposed System Architecture

The system comprises a building structure, a sensor network, a data-collection unit, signal preprocessing, feature extraction and selection, ML classification, and a visualization and alert interface. The paper identifies accelerometers, vibration sensors, strain gauges, and displacement sensors as possible sensing devices. The system working diagram shows the sequence from building structure and sensor network through Raspberry Pi data collection, preprocessing, feature extraction, feature selection, ML classification, and dashboard/alert output.

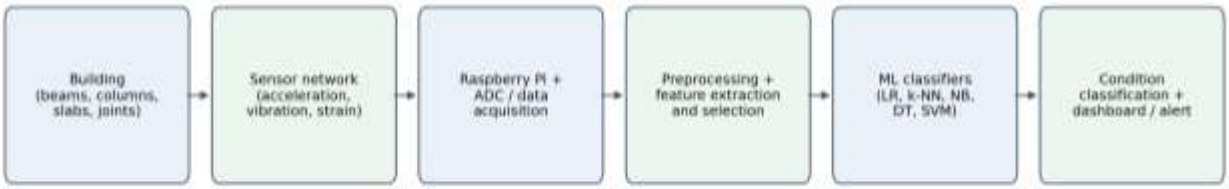


Fig. 1. Functional architecture system block diagram

3.1 Sensor configuration and data acquisition

Sensors are intended to be installed at structurally important locations such as beams, columns, slabs, and joints. Accelerometers measure acceleration and dynamic response; vibration sensors capture abnormal vibration; strain gauges measure structural strain and deformation; and displacement sensors measure structural movement. The response is represented as a time-dependent signal $x(t)$. For a sampling frequency f_s, the sampling interval is $T_s = 1/f_s$ and the discrete-time signal is $x[n] = x(nT_s)$.

The annexed acquisition code in the source report initializes an I²C interface on a Raspberry Pi, configures an ADS1115 ADC channel (A0/P0), reads ADC values and voltage, and logs timestamp, elapsed time, ADC value, and voltage to a CSV file. It sets SAMPLING_INTERVAL = 0.01 s, corresponding to a nominal 100 Hz acquisition interval.

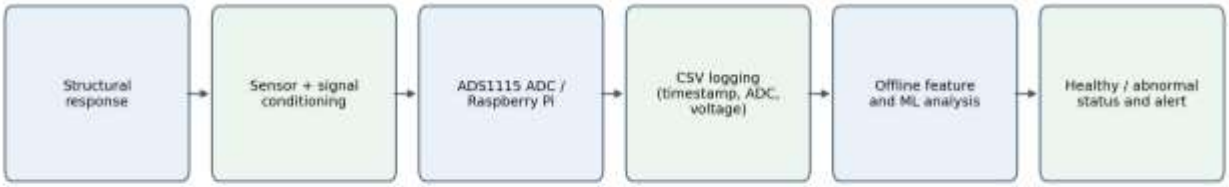


Fig. 2. Data-acquisition and analysis workflow consistent with the Raspberry Pi/ADS11.

4. Methodology

The proposed workflow follows sensor acquisition, signal preprocessing, segmentation, feature extraction, feature selection, dataset preparation, ML model training, condition classification, and visualization/alert generation. The same preprocessing and feature-extraction operations should be applied during training and online inference.

Doi:10.48047/ijiee.2026.16.2.69

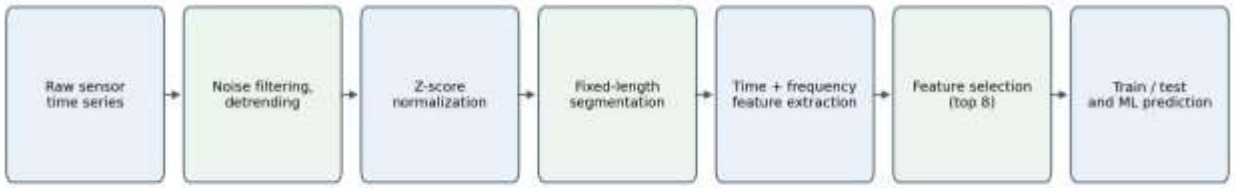


Fig. 3. Signal preprocessing, feature extraction, and classifier pipeline.

4.1 Signal preprocessing

Raw sensor signals may contain measurement noise, electrical interference, sensor drift, environmental effects, and unwanted trends. The source project identifies noise removal, detrending, normalization, and fixed-length segmentation. Filtering is represented as $xf[n] = F\{x[n]\}$, where F denotes the selected filtering operation. Detrending is represented as $xd[n] = xf[n] - T[n]$, where T[n] is the estimated trend. Z-score normalization is $zi = (xi - \mu)/\sigma$, where μ and σ are estimated from the relevant training data. The system additionally implements DC-offset removal, linear detrending, a fourth-order Butterworth band-pass filter, and Z-score normalization.The system use a sampling frequency FS = 1000 Hz and a fixed segment length of 2048 samples. It divides each recording into non-overlapping segments of that length. This setting is not consistent with the 100 Hz nominal acquisition interval in the Raspberry Pi acquisition script. The actual sampling frequency must be measured and reconciled before frequency-domain results or model performance can be reproduced.

4.2 Time-domain feature extraction

For a segment containing N samples x_i, the mean is $\mu = (1/N)\Sigma x_i$ and variance is $\sigma^2 = (1/N)\Sigma(x_i - \mu)^2$. Standard deviation is $\sigma = \sqrt{\sigma^2}$. Root mean square is $RMS = \sqrt{[(1/N)\Sigma x_i^2]}$. Peak amplitude is $Xpeak = \max|x_i|$, and crest factor is $CF = Xpeak/RMS$. These features characterize the average response, dispersion, amplitude, impulsiveness, and distribution of each signal segment.

4.3 Frequency-domain feature extraction

The Fast Fourier Transform converts the segment into a frequency-domain representation: $X[k] = \Sigma_{n=0}^{N-1} x[n]e^{(-j2\pi kn/N)}$. Dominant frequency is the frequency corresponding to the maximum spectral magnitude after ignoring the DC component. Spectral energy is calculated as the sum of squared FFT magnitudes. Spectral centroid and spectral bandwidth are also computed; mean frequency is calculated using the magnitude-weighted frequency average. These descriptors characterize the distribution and concentration of structural-response energy across frequency.

4.4 Feature vector and feature selection

The feature vector using mean, variance, standard deviation, RMS, peak amplitude, crest factor, dominant frequency, spectral energy, and bandwidth. The annex code extends this set with peak-to-peak amplitude, skewness, kurtosis, spectral centroid, and mean frequency. After feature extraction, invalid or infinite values are handled, missing feature values are replaced by the feature median, and feature scaling is performed using StandardScaler. SelectKBest with the ANOVA F-statistic (f_classif) retains up to eight features (TOP_FEATURES = 8).

4.5 Machine-learning classifiers

Five classifiers are evaluated: Logistic Regression, k-Nearest Neighbors, Gaussian Naïve Bayes, Decision Tree, and Support Vector Machine. The annex code specifies Logistic Regression with max_iter = 2000; k-NN with five neighbours; GaussianNB; a Decision Tree with max_depth = 8; and an RBF-kernel SVM with C = 1.0 and gamma = 'scale'. The data are divided using an 80:20 stratified train-test split with random_state = 42. The scaler and feature selector are fitted on the training subset and then applied to the test subset.

Classifier	Configuration in source code
Logistic Regression	max_iter = 2000; random_state = 42
k-NN	n_neighbors = 5
Naïve Bayes	GaussianNB
Decision Tree	max_depth = 8; random_state = 42
SVM	RBF kernel; C = 1.0; gamma = 'scale'

Doi:10.48047/ijiee.2026.16.2.69

4.6 Classification and alert generation

The trained classifier receives a feature vector extracted from a new sensor segment and outputs one of two structural conditions: Healthy/Normal or Abnormal/Damaged. The paper proposes displaying the result through a monitoring dashboard and generating an alert when abnormal or damaged behaviour is detected.

5. Experimental Setup and Results

The results describes a structural test environment with sensors, data acquisition, signal processing, feature extraction, ML training/testing, performance evaluation, and a monitoring/alert interface. The comparative performance values for the five classifiers shown below table.

Model	Accuracy (%)	Precision (%)	Recall (%)	F1-score (%)
Logistic Regression	91.80	90.90	92.60	91.74
k-NN	93.40	92.80	93.70	93.25
Naïve Bayes	88.90	87.60	89.80	88.69
Decision Tree	94.20	93.70	94.80	94.25
SVM	96.10	95.80	96.40	96.10

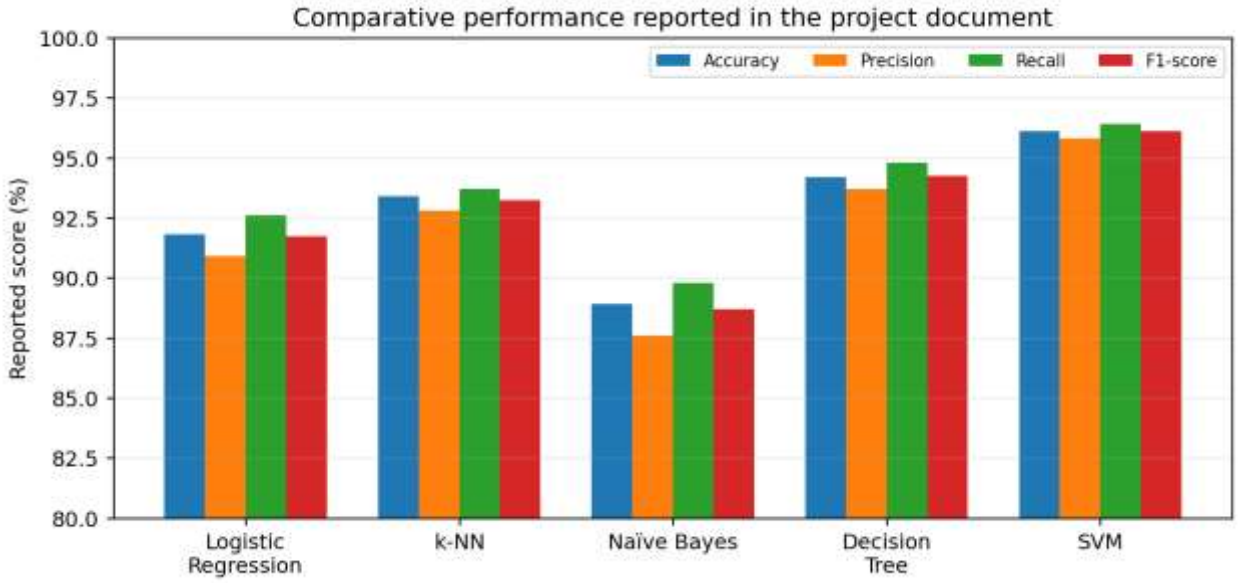


Fig. 4. Comparison of source-reported classifier metrics

Among the listed classifiers, SVM provides the highest reported accuracy (96.10%), precision (95.80%), recall (96.40%), and F1-score (96.10%). Decision Tree is the next strongest model by reported accuracy (94.20%), followed by k-NN (93.40%), Logistic Regression (91.80%), and Naïve Bayes (88.90%). These figures suggest SVM as the leading candidate within the reported comparison, subject to confirmation using the original dataset and reproducible evaluation.

5.2 Evaluation metrics

The project uses accuracy, precision, recall, F1-score, and a confusion matrix. For a binary classifier, accuracy = (TP + TN)/(TP + TN + FP + FN), precision = TP/(TP + FP), recall = TP/(TP + FN), and F1 = 2 × precision × recall/(precision + recall). Here, TP denotes damaged observations correctly identified as damaged; TN denotes healthy observations correctly identified as healthy; FP denotes healthy observations incorrectly labelled damaged; and FN denotes damaged observations incorrectly labelled healthy. False negatives are particularly important because they may delay inspection of a genuinely damaged structure.

5.3 Interpretation and reproducibility considerations

The code indicates an 80:20 stratified split, standardization, selection of the eight highest-scoring features, and evaluation of the five classifiers. Nevertheless, the source report does not include the raw sensor dataset or the generated CSV results file. It is therefore not possible to recalculate the reported metrics from the supplied material. In addition, the acquisition script's 100 Hz nominal sampling interval conflicts with the feature-extraction script's FS = 1000 Hz setting.

The proposed framework follows a compact feature-engineering approach to structural condition

Doi:10.48047/ijiee.2026.16.2.69

classification. Time-domain features describe amplitude and statistical variation, while frequency-domain features characterize the dynamic response. Feature selection reduces the input dimensionality before conventional ML classification. The comparison reported the RBF-SVM ahead of the other four classifiers, which is consistent with the ability of a nonlinear kernel to represent nonlinear decision boundaries in feature space. The use of a Raspberry Pi and an ADS1115 converter provides a low-cost acquisition route, and the time-stamped ADC readings and voltage values to CSV. The intended architecture can support local signal processing and classification, with a dashboard and alert interface for maintenance personnel.

8. Conclusion and Future Scope

This paper has presented a sensor-based intelligent building SHM framework using conventional machine-learning classifiers. The workflow integrates structural-response acquisition, signal filtering, detrending, normalization, fixed-length segmentation, time- and frequency-domain feature extraction, feature selection, and binary classification into Healthy/Normal and Abnormal/Damaged states. The paper considers Logistic Regression, k-NN, Naïve Bayes, Decision Tree, and SVM and reports the best performance for SVM, with 96.10% accuracy and 96.10% F1-score. The project also includes Raspberry Pi/ADS1115 acquisition code and proposes dashboard-based visualization and abnormal-condition alerts. The framework is attractive because it uses compact features and computationally efficient classifiers, making edge-based SHM a plausible deployment direction.

References

[1] O. Avci, O. Abdeljaber, S. Kiranyaz, M. Hussein, M. Gabbouj, and D. J. Inman, "A Review of Vibration-Based Damage Detection in Civil Structures: From Traditional Methods to Machine Learning and Deep Learning Applications," Mechanical Systems and Signal Processing, vol. 147, Art. no. 107077, 2021.

[2] G. Toh et al., "Review of Vibration-Based Structural Health Monitoring Using Deep Learning," Applied Sciences, 2020. Full bibliographic details should be checked.

[3] T. Buckley, B. Ghosh, and V. Pakrashi, "Edge Structural Health Monitoring (E-SHM) Using Low-Power Wireless Sensing," Sensors, vol. 21, no. 20, Art. no. 6760, 2021.

[4] Y. J. Cha et al., "Deep Learning-Based Structural Health Monitoring: Review," 2024. Full bibliographic details should be checked.

[5] A. A. S. R. de Sousa et al., "Multiclass Supervised Machine Learning Algorithms Applied to Structural Integrity," Springer, 2023. Full bibliographic details should be checked.

[6] O. Avci et al., "A Review of Vibration-Based Damage Detection in Civil Structures: From Traditional Methods to Machine Learning and Deep Learning Applications," Mechanical Systems and Signal Processing, vol. 147, Art. no. 107077, 2021.

[7] A. Gomez-Cabrera and P. J. Escamilla-Ambrosio, "Review of Machine-Learning Techniques Applied to Structural Health Monitoring Systems for Building and Bridge Structures," Applied Sciences, vol. 12, no. 21, Art. no. 10754, 2022.

[8] Y. Lei et al., "Data-Driven Structural Health Monitoring and Damage Detection through Deep Learning: State-of-the-Art Review," Sensors, 2020. Full bibliographic details should be checked.

[9] "Unsupervised Learning Methods for Data-Driven Vibration-Based Structural Health Monitoring: A Review," Sensors, vol. 23, no. 6, Art. no. 3290, 2023.

[10] "Intelligent Damage Diagnosis in Bridges Using Vibration-Based Monitoring Approaches and Machine Learning: A Systematic Review," Results in Engineering, 2022. Full bibliographic details should be checked.

[11] M. Numan, "Advancements in Structural Health Monitoring: A Review of Machine Learning Approaches for Damage Detection and Assessment," International Journal for Computational Civil and Structural Engineering, 2024. Full bibliographic details should be checked.

Doi:10.48047/ijiee.2026.16.2.69