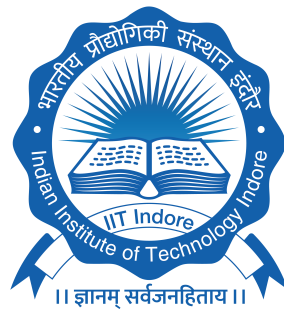


AI-Based Vibration Monitoring for Structural Damage Detection in Bridges and Roads

Ph.D. Thesis

by

REVANTH DUGALAM



**DEPARTMENT OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE
SEPTEMBER 2025**

AI-Based Vibration Monitoring for Structural Damage Detection in Bridges and Roads

A Thesis

*submitted in partial fulfillment of the
requirements for the award of the degree*

of

DOCTOR OF PHILOSOPHY

by

REVANTH DUGALAM

(2101104004)



**DEPARTMENT OF CIVIL ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY INDORE**

September 2025

Dedication


*To my mother Smt. Padmavati and father Shri
Yedukondalu.*



INDIAN INSTITUTE OF TECHNOLOGY INDORE

I hereby certify that the work that is being presented in the thesis entitled "**AI-based Vibration Monitoring for Structural Damage Detection in Bridges and Roads**" in partial fulfilment of the requirements for the award of the degree of Doctor of Philosophy and submitted in the Department of Civil Engineering, Indian Institute of Technology Indore, is an authentic record of my work carried out during the time period from January 2022 to July 2025 under the supervision of Dr. Guru Prakash, Assistant Professor, Department of Civil Engineering, Indian Institute of Technology Indore.

The matter presented in this thesis has not been submitted for the award of any other degree of this or any other institute.

 01/09/2025
Signature of the student with date

REVANTH DUGALAM

This is to certify that the above statement made by the candidate is correct to the best of my/our knowledge.

 01/09/2025

Signature of Thesis Supervisor with date

(Dr. GURU PRAKASH)

.....
REVANTH DUGALAM has successfully given his Ph.D. Oral Examination held on...27/11/2025.....

 27/11/2025

Signature of Thesis Supervisor with date

(Dr. GURU PRAKASH)

Acknowledgements

First and foremost, I would like to thank almighty **God** for granting me countless blessings and strength throughout this journey.

I would like to express my heartfelt gratitude and deep appreciation to my thesis supervisor, **Dr. Guru Prakash**, Assistant Professor, Department of Civil Engineering, Indian Institute of Technology Indore, for his unwavering support, guidance, and mentorship throughout my doctoral journey. His constant encouragement, visionary ideas, and constructive feedback have been instrumental in shaping the direction and quality of my research. I am especially thankful for the trust he placed in me to undertake this novel topic. His enthusiasm, meticulous planning, and rigorous reviews inspired discipline and clarity in my work. His thoughtful suggestions on technical content and writing greatly enhanced the quality of manuscripts and publications.

I am sincerely thankful to **Prof. Manish Kumar Goyal**, and **Dr. Kaustav Bakshi** for their valuable insights and suggestions during the Comprehensive Evaluation and Research Progress (CERP) seminars as my PSPC members. Their technical inputs have motivated me to refine and strengthen my work. I also express my gratitude to **Dr. Abhishek Rajput**, Head of the Department of Civil Engineering, for his continuous support and words of encouragement.

I respectfully acknowledge the support and institutional vision of **Prof. Suhas Joshi**, Director, Indian Institute of Technology Indore, for providing an excellent research environment and opportunities for academic growth.

I would also like to thank **Mr. Tapes Parihara**, Manager, Academic Office, and his team for their consistent administrative support throughout the course of my Ph.D.

My sincere thanks to the staff of the Estate Office, Infrastructure Development Office, Housekeeping, and Security teams at IIT Indore for their timely assistance

and support. I am grateful to the technical and administrative staff of the Department of Civil Engineering, including *Ms. Rinki Kukreja*, *Mr. Ajay Malviya*, and *Mr. Amit Jadhav*, for their constant cooperation during the course of my research.

I extend special thanks to my seniors *Dr. Gyanesh Patnaik*, and *Dr. Anshul Kaushik* for their mentorship, encouragement, and camaraderie. I also thank my colleagues *Mr. Manish Yadav*, *Mr. Sumedh Limaye* and *Mr. Gourav Agarwal* for their collaborative spirit, engaging discussions, and for contributing to a vibrant and energetic lab atmosphere.

I am profoundly indebted to my family for their unconditional love, support, and sacrifices. My deepest gratitude to my mother **Smt. Padmavati Dugalam**, my father **Shri Yedukondalu Dugalam** and my brother **Mr. Eswar Dugalam**, whose constant encouragement has been the backbone of this journey. I also extend my heartfelt thanks to my extended family and relatives for their blessings and motivation. Finally, I remember and honor my grandfather, **Late Shri Subba Rao Dugalam**, who was my first teacher and an enduring source of inspiration.

Abstract

Timely detection of structural damage in civil transportation infrastructure is critical to protect public safety, maintain serviceability, and extend the service life of structural systems. Conventional manual inspection is unsuitable for large, spatially distributed systems such as bridge superstructures and road networks; procedures are labor-intensive, require traffic control, and full coverage is rarely achievable. Structural Health Monitoring (SHM) offers a data-driven alternative, using in situ measurements to localize and quantify deterioration in such structures. In particular, vibration-based monitoring uses sensor measurements of ambient and operational responses to extract dynamic parameters that serve as damage-sensitive indicators. Artificial Intelligence (AI) models help to transform sensor measurements and their derived dynamic parameters into decision-ready outputs of damage presence, location, and severity. Despite these advances, notable gaps in practice remain: for bridge-beam components, the literature has essentially treated localization and quantification as separate problems, and end-to-end models that deliver both simultaneously are scarce. For roads, robust multi-defect classification remains limited, and integrated frameworks that carry network-level condition assessment through segment-level repair scoping with transparent, unit-rate cost estimates are not yet well established.

The objectives are threefold. First, develop a hybrid multi-input, multi-output (MIMO) regression that fuses experimentally measured and finite-element-simulated modal frequencies to detect damage and simultaneously estimate location and depth in bridge-beam components. Second, extend this component-level analysis with multi-output deep regression to predict location, width, and depth from frequency-domain inputs, benchmarking FNN, MLP, RNN, LSTM, and 1D CNN-GRU-ResNet with EEFO tuning. Third, design a road-network assessment workflow that couples segment-level IRI with supervised classification from vehicle-mounted accelerometers (binary and multi-class) and links these outputs to unit-rate cost coefficients for transparent repair scopes—supporting prioritized, budget-aware interventions.

This thesis develops data-driven workflows for bridge-beam components and road networks. For bridge beams, a hybrid multi-input, multi-output (MIMO) regression framework fuses experimentally measured and finite-element-simulated modal frequencies to detect damage and simultaneously estimate its location and depth. Four learners—Support Vector Regression, Linear Regression, Gradient Boosting, and Random Forest (RF)—were developed and benchmarked within this MIMO formulation. After hyperparameter tuning, the ensemble model RF fit the data best and stayed accurate on the new testing dataset. The framework is extended with deep learning on the same frequency-domain inputs to estimate damage width. Five architectures (FNN, MLP, RNN, LSTM, and a 1D CNN-GRU-ResNet hybrid) are tuned using Electric Eel Foraging Optimization (EEFO). LSTM yields the lowest overall errors, with the 1D CNN-GRU-ResNet close behind and particularly strong for width and depth. SHAP analysis indicates that lower-order modes are most informative for location, while higher-order modes contribute more to width and depth.

Extending the same measurement-to-decision workflow to road networks, overall ride quality and localized defects are combined by computing IRI at 50-m spacing along a 13-km road section and training supervised binary classifiers on signals from vehicle-mounted accelerometers to flag pothole presence. Among the tested models, SVM delivered the best binary performance (81% accuracy; precision 0.67, recall 0.92), outperforming alternatives on the same vibration features. The workflow is then extended to multi-class classification (potholes, rutting, speed humps) using a sliding-window feature extractor with window-size tuning; a tuned Random Forest achieved 97% accuracy in simulation and 85% on field data, exceeding SVM, KNN, Naïve Bayes, and Decision Tree baselines. Finally, segment-level IRI ratings and defect labels are linked to unit-rate cost coefficients to produce transparent repair scopes and an overall maintenance estimate for the pilot stretch, addressing the gap between condition assessment and prioritized, budget-aware interventions.

Overall, the study provides unified, data-driven workflows that integrate detection, localization, and quantification for both bridge-beam components and road networks. These contributions strengthen the practical applicability of AI-enabled

SHM and road-condition assessment, supporting more efficient, scalable, and cost-informed infrastructure maintenance and management.

List of Publications

(Total impact factor of published articles: 25.4 &
citations: 122+)

S. No	Details of the Publication	Impact Factor	Quartile Ranking
Thesis-Related Research Articles			
1	Dugalam, R., & Prakash, G. (2025). Frequency-based deep learning model for beam damage detection. <i>European Journal of Environmental and Civil Engineering</i> , 1-27. https://doi.org/10.1080/19648189.2025.2495134	2.3	Q2
2	Dugalam, R., & Prakash, G. (2024). Development of a random forest based algorithm for road health monitoring. <i>Expert Systems with Applications</i> , 251, 123940. https://doi.org/10.1016/j.eswa.2024.123940	7.5	Q1
3	Dugalam, R., & Prakash, G. (2024, February). A hybrid multiple input multiple output (MIMO) model for simultaneous localization and quantification of structural damage in beam structures. In <i>Structures</i> (Vol. 60, p. 105879). Elsevier. https://doi.org/10.1016/j.istruc.2024.105879	4.3	Q1

Other Contributed Research Articles

- 5 Prakash, G., **Dugalam, R.**, Barbosh, M., & Sadhu, A. (2022, October). Recent advancement of concrete dam health monitoring technology: A systematic literature review. In *Structures* (Vol. 44, pp. 766-784). Elsevier. <https://doi.org/10.1016/j.istruc.2022.08.021> 4.3 Q1
- 6 **Dugalam, R.**, & Prakash, G. (2025). Damage detection of Carbon Fibre Reinforced Polymer (CFRP) tubes subjected to low-velocity transverse impact. *Next Materials*, 9, 101355. <https://doi.org/10.1016/j.nxmte.2025.101355> - Scopus
- 7 Maganti, T. R., Kandikuppa, C. S., Gopireddy, H. K., **Dugalam, R.**, & Boddepalli, K. R. (2025). Enhanced compressive strength and impact resistance in hybrid fiber reinforced ternary-blended alkali-activated concrete: An experimental, weibull analysis and finite element simulation. *Composites Part C: Open Access*, 100629. <https://doi.org/10.1016/j.jcomc.2025.100629> 7 Scopus

Conference articles published

1. **Dugalam, R.**, & Prakash, G. (2022, December). An Integrated Approach for Damage Detection, Localization and Quantification in Beam Like Structure Using Vibration Measurements. In *Structural Integrity Conference and Exhibition* (pp. 511-523). https://doi.org/10.1007/978-981-97-6367-2_40 Springer - Nature

2. **Dugalam, R.,** Ram S.S., & Prakash, Springer -
 G. Pavement and Road Health Monitoring Nature
 using Random Forest Technique. *Recent
 Trends in Computational Mechanics and
 Simulation: Select proceedings of ICCMS
 2022.* "Accepted"
-

S. No	Details of the Conferences	Format	Organizer & Place
International Conferences			
1	Structural Integrity Confer- ence and Exhibition	Oral Pre- sentation	Indian Institute of Technology Hyder- abad, Telangana, In- dia.
2	International Congress on Computational Mechanics and Simulation	Oral Pre- sentation	Indian Institute of Technology Indore, Madhya Pradesh, In- dia.

Contents

Acknowledgements	vii
Abstract	ix
List of Publications	xiii
Nomenclature	xxiv
Acronyms	xxvi
1 Introduction	1
1.1 Background	1
1.2 Motivation	4
1.3 Objectives	6
1.4 Thesis organization	7
2 Literature Review	11
2.1 Introduction	11
2.2 Machine learning	11
2.2.1 Regression-based ML models for bridge components	12
2.2.2 Classification-based ML models for road damage detection	16
2.3 Deep learning	21
2.3.1 Regression-based models for bridge components	21
2.4 Gaps in literature	22
3 Background : Formulation of AI Models in SHM	25
3.1 Introduction	25
3.2 Machine learning models	26
3.2.1 Prediction function expressions	28

3.2.2	Model evaluation metrics	35
3.3	Deep learning models	37
3.3.1	Architectures	38
3.3.2	Mathematical formulations	39
3.3.3	Loss functions and optimization	43
3.3.4	Evaluation and inference	44
3.4	Summary	45
4	Proposed Methodology for Damage Assessment	47
4.1	Introduction	47
4.2	Overall framework	48
4.3	Data acquisition	49
4.4	Regression-based damage assessment	50
4.4.1	Feature extraction	50
4.4.2	AI model development	51
4.4.3	Optimization and hyperparameter tuning	52
4.4.4	Inference and decision support	52
4.5	Classification-based damage assessment	53
4.5.1	Feature extraction	53
4.5.2	AI model development	53
4.5.3	Optimization and hyperparameter tuning	54
4.5.4	Inference and decision support	55
4.6	Summary	55
5	Bridge Beam Damage Assessment	57
5.1	Introduction	57
5.2	Experimental setup and data collection	58
5.2.1	Numerical simulation of modal parameters and validation	62
5.2.2	Hybrid dataset construction	65
5.3	Implementation using ML: MIMO regression model	66
5.3.1	Problem formulation as MIMO regression	67
5.3.2	Model development and evaluation	68
5.3.3	Hyperparameter tuning	71

5.3.4	Inference and testing	72
5.4	Implementation using DL: EEFO - optimized model	73
5.4.1	Extended output parameter	74
5.4.2	Model architectures development	74
5.4.3	Performance analysis	77
5.5	Comparative summary implementations	83
5.6	Conclusion	84
6	Road Damage Assessment	87
6.1	Introduction	87
6.2	Field study setup and data collection	88
6.2.1	Data acquisition system	90
6.3	Model 1: Binary classification of road damage and estimation of IRI	93
6.3.1	Determination of IRI	96
6.3.2	Model development	99
6.3.3	Maintenance cost assessment	104
6.4	Model 2: Multi-class classification of road damages	109
6.4.1	Data processing and feature extraction	110
6.4.2	Model development	114
6.4.3	Comparison of the proposed classifier with other classifiers	118
6.5	Cost estimation for rutting	119
6.6	Conclusion	121
7	Conclusions and Recommendations	123
7.1	Significant contributions	123
7.2	Conclusions	124
7.3	Recommendations for future work	125

List of Figures

1.1	Damages on I-40 bridge superstructure	1
-----	---------------------------------------	---

1.2	Damages on road network	2
1.3	Enhancing the essential infrastructure monitoring using AI	3
1.4	Damages on a) Humen pearl river bridge b) Diefenbaker bridge	4
1.5	Different types of damages on road networks	5
1.6	Thesis Organization	8
2.1	Beam Damage Detection Experimental Setup [1].	14
2.2	Road condition monitoring based on AI techniques [2]	17
2.3	Accelerometer placement on vehicle: (a) overview of sensor installations; (b) direction of sensor box installation; and (c) placement of sensor box [3]	17
3.1	Categorization of AI technologies	25
3.2	ML model development process	27
3.3	Working principle of random forest regressor	29
3.4	Working principle of random forest classifier.	34
3.5	Confusion matrix in classification	37
3.6	DL model development process	38
3.7	FNN structure for prediction of damage characteristics	40
3.8	RNN structure for damage detection using frequencies	41
3.9	Damage characteristics detection using LSTM	41
3.10	Damage characteristics detection using 1D CNN–GRU–ResNet	43
4.1	Proposed methodology for AI-based structural damage assessment	48
4.2	Generalized data acquisition from dynamically excited structural systems	49
4.3	Feature extraction for regression task	50
5.1	Cantilever beam	58
5.2	Inducing damage on the beam using an angle grinder with 4-inch metal cutting blade	59
5.3	Damaged portions at different locations with varying widths and depths	60
5.4	Experimental setup performing modal analysis on the damaged cantilever beam using accelerometer and a data acquisition system.	60
5.5	Vibration data of a damaged beam a) time domain b) frequency domain.	61
5.6	Change in frequencies with varying widths of damage.	61

5.7	Numerically modelled cantilever beam	62
5.8	Cantilever beam with damage at a distance of 785 mm from fixed end. . . .	63
5.9	Frequency variation with changing damage location and depth, (a) for second mode and (b) fifth modes	64
5.10	Six modes of the beam	65
5.11	Flowchart for the implementation of proposed methodology using ML	66
5.12	Regression performance plots of all MIMO models for damage location and depth, (a) SVR (b) LR (c) GBM (d) RF	69
5.13	Flowchart of the proposed frequency-based DL model	73
5.14	Evolution of best scores across generations for DL models	75
5.15	Variation of best optimized epochs across generations for DL models	75
5.16	Model learning curves	76
5.17	Inference time comparison across DL models	77
5.18	Calibration plots (a) LSTM, (b) 1D CNN-GRU-ResNet	80
5.19	Best model: SHapley additive explanations (a) location, (b) width, (c) depth	82
6.1	Route map of pilot study	89
6.2	Different road damages and their respective vibration signals.	89
6.3	Installation of accelerometer and data monitoring.	90
6.4	Sensors placement on the test vehicle.	91
6.5	Monitoring of vibration response during field study	91
6.6	Acceralation data from the field monitoring	91
6.7	A typical acceleration data collected from the dashboard accelerometer. . . .	92
6.8	A schematic diagram illustrating the binary classification integration with IRI	93
6.9	Typical raw acceleration data and filtered signal	94
6.10	Sliding window technique	95
6.11	(a) The velocity in the Z direction is derived from the filtered acceleration and (b) The displacement in the Z direction is derived from the velocity . . .	96
6.12	Segment-wise IRI values for every 50 m interval along the test road	98
6.13	Distribution of IRI	99
6.14	Pothole detection using SVM	100

6.15	Testing dataset confusion matrices for the four models	102
6.16	Precision-recall curve for SVM classification	103
6.17	Localization using SVM	103
6.18	A flowchart detailing the sequential steps for the process of multiple road anomaly classification.	110
6.19	Feature extraction for classification task	111
6.20	Importance or weight of various features.	111
6.21	Important features used in the field study	113
6.22	Appropriate selection of window size.	114
6.23	Convergence study for classifier metrics	115
6.24	Precision-recall curve for the classification.	117
6.25	ROC curve for the classification.	117

List of Tables

2.1	Review of regression-based ML models for vibration-based damage de- tection	15
2.2	Limitations of existing literature for road damage detection	23
5.1	Material Properties of Steel	58
5.2	Mesh convergence by RMS percentage error	63
5.3	Comparison of experimental and numerical modal frequencies for differ- ent damages	64
5.4	Training parameters of MIMO regression models.	68
5.5	Learned coefficients of LR model	68
5.6	Feature importance of modal frequencies (RF model).	70
5.7	Various test statistics for the trained models.	70
5.8	Grid of Parameters for Hypertuning in MIMO Regression Models	71
5.9	Improved performance metric after hyperparameter tuning.	72
5.10	Actual and predicted damage location and depth of new testing dataset.	73

5.11	EEFO-optimized hyperparameters for the deep learning models	77
5.12	Performance Metrics of Different Models	78
5.13	Comparison of ML and DL-based approaches for damage assessment	84
6.1	Details of car and accelerometer	90
6.2	FHWA IRI thresholds for Highways [4].	99
6.3	Model parameters estimates of various ML models	100
6.4	Model Evaluation Metrics for Different Classes and Averaging Methods . .	101
6.5	Estimation of segment-wise IRI-based maintenance Cost	106
6.6	IRI range based maintenance cost coefficient (α) (30 km/hr)	107
6.7	Estimation of pothole repair cost	107
6.8	Cost details of the material used, equipment and labour	108
6.9	Kilometre-wise IRI-based and pothole-based maintenance cost estimation (in million INR)	108
6.10	Features used in field study.	112
6.11	Model parameters	115
6.12	Model performance for classification of various damages.	116
6.13	Classification Metrics for Different Classifiers	119
6.14	Various unit cost (in INR) used in the cost estimation.	120
6.15	The estimated cost (in INR) for repairing of typical rutting damage. . . .	121

Nomenclature

$x(t)$: Acceleration signal (time domain)
\mathbf{F}_{reg}	: Feature vector for regression task
$\mathcal{F}[x(t)]$: Fourier transform of $x(t)$
$\mathbf{x} = [x_1, x_2, \dots, x_n]$: Input feature vector
\hat{y}	: Predicted output / class label
y	: Ground-truth value / class label
N	: Number of training samples
β_0, β_i	: Intercept and regression coefficients
\mathbf{w}, b	: Weights and bias in support vector regression
ξ_i, ξ_i^*	: Slack variables for margin violations (SVR)
ε	: Epsilon-insensitive margin threshold (SVR)
C	: Regularization parameter (SVR)
$h_t(\mathbf{x})$: Output of the t^{th} decision tree
T	: Total number of trees
η	: Learning rate in boosting algorithms
$\mathcal{N}_k(\mathbf{x})$: Set of k nearest neighbors to \mathbf{x}
$\mathcal{L}_{\text{class}}$: Categorical cross-entropy loss
\mathcal{L}	: Loss function (generic)
C_k	: Class k (Naive Bayes)
$P(C_k \mathbf{x})$: Posterior probability of class C_k given \mathbf{x}
$P(C_k)$: Prior probability of class C_k
$P(x_j C_k)$: Likelihood of feature x_j given class C_k
f_{binary}	: Binary classification decision function
f_{multi}	: Multi-class classification decision function
$\mathbf{y}_i \in \{0, 1\}$: Binary class labels
$\mathbf{y}_i \in \{1, 2, \dots, K\}$: Multi-class labels
N_k	: Defect count / distribution for class k
K	: Total number of classes
β, \mathbf{w}, b	: Model parameters (generic)
$\mathbb{I}(\cdot)$: Indicator function
$P(\cdot)$: Probability estimate
$F_k(\mathbf{x})$: Boosted output score for class k
p_k	: Class proportion at a decision node
$\mathbf{W}^{(l)}$: Weights for layer l
$\mathbf{b}^{(l)}$: Bias terms for layer l
σ	: Activation function
$\mathbf{h}^{(l)}$: Hidden output at layer l
t	: Time-step index

\mathbf{h}_t	: Hidden state at time t
C_t	: LSTM cell state at time t
f_t, i_t, o_t	: LSTM gates (forget, input, output)
\tilde{C}_t	: LSTM candidate state
\mathbf{f}_{CNN}	: CNN feature vector
\mathbf{x}_{skip}	: Skip (residual) input in ResNet

Acronyms

ACO ant colony optimization. [xxv](#), [15](#)

Adam adaptive moment estimation. [xxv](#), [49](#)

AI artificial intelligence. [xxv](#), [3](#), [4](#), [5](#), [6](#), [7](#), [8](#), [11](#), [16](#), [25](#), [26](#), [45](#), [47](#), [48](#), [49](#), [55](#), [57](#), [87](#), [88](#), [123](#), [124](#), [125](#)

CNN convolutional neural network. [xxv](#), [16](#), [21](#), [42](#), [77](#), [78](#)

DL deep learning. [xxv](#), [8](#), [11](#), [16](#), [21](#), [22](#), [25](#), [37](#), [38](#), [39](#), [44](#), [45](#), [58](#), [63](#), [64](#), [66](#), [67](#), [73](#), [74](#), [77](#), [83](#), [84](#), [85](#), [124](#)

DT decision tree. [xxv](#), [19](#), [33](#), [34](#), [118](#), [119](#), [125](#)

EEFO electric eel foraging optimization. [xxv](#), [44](#), [49](#), [58](#), [75](#), [76](#), [77](#), [83](#), [84](#), [123](#), [124](#)

EV explained variance. [xxv](#), [36](#), [70](#), [72](#)

FFT fast fourier transform. [xxv](#), [13](#), [18](#), [50](#)

FN false negative. [xxv](#), [36](#), [37](#)

FNN feedforward neural network. [xxv](#), [19](#), [38](#), [39](#), [40](#), [76](#), [77](#), [78](#), [79](#), [80](#), [124](#)

FP false positive. [xxv](#), [36](#), [37](#)

GBM gradient boosting machine. [xxv](#), [14](#), [15](#), [19](#), [22](#), [30](#), [34](#), [35](#), [52](#), [54](#), [68](#), [70](#), [71](#), [72](#), [84](#), [124](#)

GRU gated recurrent unit. [xxv](#), [38](#), [39](#), [42](#), [77](#), [78](#)

IRI international roughness index. [xxv](#), [5](#), [18](#), [19](#), [88](#), [92](#), [94](#), [96](#), [97](#), [98](#), [99](#), [104](#), [105](#), [106](#), [107](#), [108](#), [109](#), [123](#), [124](#), [125](#)

KNN k-nearest neighbors. [xxv](#), [19](#), [31](#), [33](#), [118](#), [119](#), [125](#)

LR linear regression. [xxv](#), [14](#), [15](#), [28](#), [68](#), [70](#), [84](#), [124](#)

LSTM long short-term memory. [xxv](#), [21](#), [38](#), [41](#), [42](#), [76](#), [77](#), [78](#), [79](#), [80](#), [81](#), [82](#), [83](#), [84](#), [124](#), [125](#)

MAE mean absolute error. [xxv](#), [36](#), [43](#), [44](#), [70](#), [71](#), [72](#), [76](#), [77](#), [78](#), [84](#)

MAPE mean absolute percentage error. [xxv](#), [36](#), [70](#), [72](#)

MedAE median absolute error. [xxv](#), [36](#), [70](#), [72](#)

MIMO multi-input multi-output. [xxv](#), [57](#), [66](#), [67](#), [68](#), [72](#), [73](#), [74](#), [84](#), [123](#), [124](#), [125](#)

ML machine learning. [xxv](#), [8](#), [11](#), [12](#), [13](#), [16](#), [19](#), [21](#), [22](#), [25](#), [26](#), [27](#), [31](#), [37](#), [44](#), [45](#), [57](#), [58](#), [66](#), [73](#), [83](#), [84](#), [85](#), [99](#), [100](#)

MLP multilayer perceptron. [xxv](#), [38](#), [40](#), [76](#), [77](#), [78](#), [79](#), [84](#), [124](#)

MSE mean squared error. [xxv](#), [30](#), [36](#), [39](#), [42](#), [43](#), [44](#), [52](#), [68](#), [70](#), [71](#), [72](#), [76](#), [77](#), [78](#), [84](#), [125](#)

MSLE mean squared logarithmic error. [xxv](#), [36](#), [70](#), [72](#)

NDT non-destructive testing. [xxv](#), [2](#)

OLS ordinary least squares. [xxv](#), [28](#), [52](#)

PSO particle swarm optimization. [xxv](#), [16](#), [21](#)

R² coefficient of determination. [xxv](#), [36](#), [70](#), [71](#), [72](#), [84](#)

RBF radial basis function. [xxv](#), [31](#), [32](#), [68](#), [71](#), [100](#)

ReLU rectified linear unit. [xxv](#), [39](#)

ResNet residual network. [xxv](#)

RF random forest. [xxv](#), [14](#), [15](#), [16](#), [19](#), [21](#), [23](#), [29](#), [34](#), [35](#), [52](#), [54](#), [68](#), [70](#), [71](#), [72](#), [84](#), [99](#), [100](#), [101](#), [102](#), [111](#), [112](#), [113](#), [114](#), [115](#), [116](#), [118](#), [119](#), [124](#), [125](#), [147](#)

RMSE root mean squared error. [xxv](#)

RMSProp root mean square propagation. [xxv](#), [44](#)

RNN recurrent neural network. [xxv](#), [38](#), [40](#), [76](#), [77](#), [78](#), [79](#), [80](#), [84](#), [124](#)

ROC receiver operating characteristic. [xxv](#), [118](#)

SDD structural damage detection. [xxv](#), [12](#), [13](#), [14](#), [21](#)

SHM structural health monitoring. [xxv](#), [2](#), [4](#), [8](#), [11](#), [12](#), [14](#), [15](#), [21](#), [22](#), [25](#), [26](#), [27](#), [28](#), [31](#), [37](#), [44](#)

SVM support vector machine. [xxv](#), [19](#), [23](#), [30](#), [32](#), [53](#), [54](#), [99](#), [100](#), [101](#), [102](#), [103](#), [104](#), [107](#), [108](#), [118](#), [119](#), [125](#)

SVR Support vector regression. [xxv](#), [30](#), [31](#), [68](#), [70](#), [71](#), [72](#), [84](#), [124](#)

TN true negative. [xxv](#), [36](#), [37](#)

TP true positive. [xxv](#), [36](#), [37](#)

VBM vibration-based monitoring. [xxv](#), [2](#), [3](#), [11](#), [12](#), [15](#), [125](#)

Introduction

1.1 Background

Transportation infrastructure includes large networks of bridges, roads, railways, airports, and tunnels that are very important for modern life. These infrastructures —help the economy grow by allowing goods and services to move easily, connect people socially by making travel and communication possible, and are essential for everyday activities, ensuring smooth travel for everyone. This vast network includes everything from expansive railway lines and highways that cross entire countries to major bridges and tunnels that link remote areas. These structures involve huge investments of money and resources and are vital for —the country’s wealth and development, keeping people safe while traveling. While transportation infrastructure is multifaceted, this thesis focuses specifically on bridges and roads, given their fundamental role in regional connectivity, urban development, and everyday transportation.



Figure 1.1: Damages on I-40 bridge superstructure

However, this essential infrastructure is continually exposed to numerous challenges, such as nature’s impact, including extreme weather events, floods, earthquakes, and other natural events, which can damage these structures over time. Together with the cumulative effects of aging, material fatigue, and rising traffic loads, this leads to a gradual, hard-to-

detect degradation (Fig. 1.1 & 1.2). This degradation often begins subtly and is difficult to detect in its early stages. If these issues go unnoticed and unaddressed, they can escalate into major problems—resulting in catastrophic economic losses, significant disruptions to transportation systems, and, most importantly, heightened risks to public safety. There is a need to regularly check and monitor the condition of these structures proactively, ensuring their long-term integrity and operational safety.



Figure 1.2: Damages on road network

Traditionally, damage assessment of infrastructures like bridges and roads has relied heavily on periodic visual inspections and manual non-destructive testing (NDT) techniques. These approaches are important but have several drawbacks. They usually only find problems after some damage has already happened, requiring much time and effort from workers. The results can also depend on who is doing the inspection, making them sometimes less reliable. They are limited in their ability to detect hidden or internal damage, provide continuous monitoring, or offer quantitative insights into the severity and progression of deterioration. Because of these limitations, problems are sometimes only discovered when they become serious, leading to expensive emergency repairs, longer shut-downs, and maintenance resources not being used to their best.

To overcome the limitations of traditional inspection methods, the field of structural health monitoring (SHM) has become increasingly important. The main aim of SHM is to provide continuous, real-time assessment of structural integrity, enabling a shift from reactive maintenance to proactive, condition-based management. One of the most effective SHM techniques is vibration-based monitoring (VBM). This technique preserves the struc-

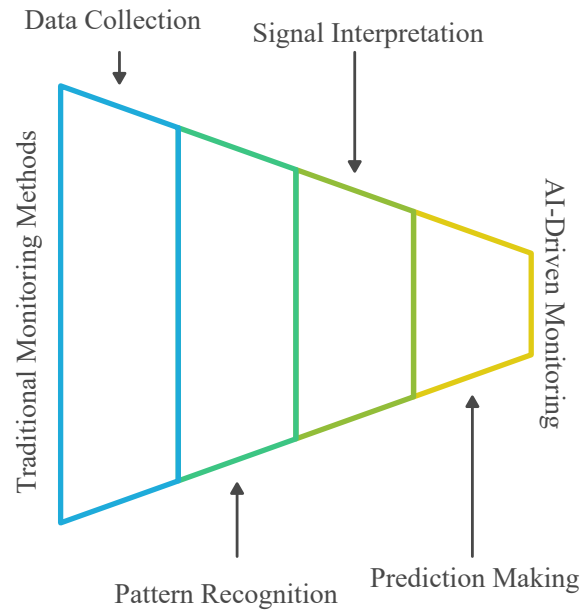


Figure 1.3: Enhancing the essential infrastructure monitoring using AI

tural integrity while offering high sensitivity to minor changes in dynamic response, which may indicate early-stage damage. VBM also provides information about the overall health of the entire structure, not just one specific area. Even though VBM have many advantages, they produce a large and complex amount of vibration data that can be difficult to manage. Traditional analysis methods are often not robust or flexible enough to handle this data effectively. This is where artificial intelligence (AI) becomes very useful. artificial intelligence (AI) can find patterns in large datasets, understand complicated vibration signals, and make smart predictions about a structure's condition. In VBM, AI can automatically detect signs of damage in bridges and roads, leading to more accurate results, reducing the chance of human error, and helping keep these structures safer for longer. Fig. 1.3 visually illustrates the transition from traditional to AI-driven monitoring. It shows how the process evolves from simple data collection and signal interpretation to advanced pattern recognition and prediction-making, highlighting how AI enables more comprehensive and proactive infrastructure monitoring.

This thesis conceptualizes the health assessment of transportation infrastructure—particularly bridges and roads—as a shared challenge that can benefit from a common methodological approach. While these two structures have different designs, behaviors, and damage patterns, the fundamental steps involved in vibration-based monitoring

and AI-driven analysis offer a consistent framework that can be adapted to each structure. This research seeks to bridge the gaps in intelligent infrastructure damage assessment, paving the way for a more resilient, efficient, and safer transportation future.

1.2 Motivation

Even though SHM is becoming more popular and vibration-based methods are improving, there are still significant challenges in making damage detection fully automatic, reliable, and valuable for all types of transportation infrastructure. The existing literature so far addresses a specific task at a time, leading to fragmented solutions that do not fully leverage the potential of integrated AI-driven approaches for transportation networks. Several critical gaps remain in current research, motivating the need for more comprehensive and multi-task approaches in this thesis.

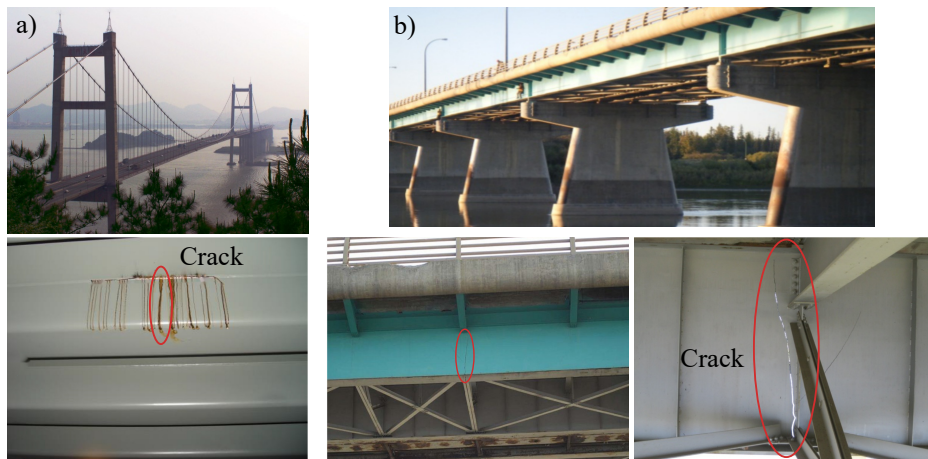


Figure 1.4: Damages on a) Humen pearl river bridge b) Diefenbaker bridge

A key challenge for structural elements such as bridge beams remains the concurrent localization and precise damage quantification using vibration data. Existing techniques can often detect or localize damage. However, reliably extracting multiple damage parameters—such as location, depth, and width—in a single step with high accuracy is still limited by current algorithmic capabilities. Most approaches yield only one damage metric per analysis, requiring multiple iterations to assemble a complete damage profile. This complicates and slows maintenance decision-making, for example, in large-scale and aging infrastructure such as the I-40 Mississippi river bridge, Humen pearl river bridge and Diefenbaker bridge where comprehensive inspections are logistically complex and resource-intensive.

In such cases, teams of structural engineers are often trained and required to physically climb, using ropes and harnesses, to inspect every critical component, sometimes under hazardous conditions. Knowing only the location of damage, without understanding its severity (depth or width), is insufficient for prioritizing repairs or planning targeted interventions.

Early studies—such as the review by Alvandi and Cremona (2006) [5]—focused on basic vibration-based methods that could confirm if damage was present and roughly where it was located (see Fig. 1.4). However, these approaches generally stopped short of giving detailed measurements about the damage’s severity or exact size. Recent studies, including the work by Chinka et al. (2021) [6], have taken important steps forward in the field by successfully identifying and quantifying specific damage characteristics—such as the exact position and depth of cracks—in laboratory beam experiments. Their results, often shown as scatter plots comparing predicted and actual values, demonstrate strong agreement for these parameters. Still, predicting a wider range of damage details—including crack width or dealing with more complicated types of damage—all at once using a single AI model remains a significant challenge. This gap highlights the need for more advanced solutions for effective, focused maintenance planning.



Figure 1.5: Different types of damages on road networks

Most existing road monitoring systems focus either on overall ride quality such as the international roughness index (IRI) [7, 8] —or on detecting specific localized defects like potholes [9, 10] , but seldom address both aspects together. This limitation is visually illustrated in the Fig. 1.5, which shows a section of road featuring several types of surface problems: roughness, potholes, rutting, and speed bumps. The illustration makes it

evident that these different forms of damage often appear together on the same stretch of pavement, and each brings its own set of challenges when it comes to identifying and categorizing them. Because of this, it's important to move beyond approaches that focus on just one aspect, like overall ride quality or single types of defects, offering a comprehensive view of road health and directly linking these insights to maintenance strategies and costs. Furthermore, while AI has made significant progress, accurately classifying different types of road damage from vibration data—such as distinguishing between potholes, rutting, and speed bumps—remains a complex task. Effective maintenance depends on this level of detail, but the complex and varied vibration patterns produced by different defects require advanced AI models capable of robust multi-class classification.

A broader critical gap in this field is proving that AI-based vibration monitoring systems can adapt and scale to many parts of the transportation infrastructure. To tackle this, the thesis develops and validates a new AI-driven vibration monitoring framework designed to work across various structural types. What sets this framework apart is its flexibility—it applies the same core principles of AI-based vibration analysis whether the structure is a bridge or a road. The practical value of this framework is shown through two primary case studies: one involves controlled laboratory tests on bridge beams. At the same time, the other uses real-world data gathered from road networks. By this framework, the thesis aims to show how intelligent and proactive maintenance strategies can be developed for all transportation infrastructure.

1.3 Objectives

Building upon the critical gaps identified in current infrastructure monitoring practices, the overarching aim of this thesis is to develop and rigorously validate a comprehensive AI-based vibration monitoring framework. This framework is designed for structural damage detection across diverse transportation infrastructure components, with a specific focus on bridge beams and road networks, ultimately enabling more proactive, intelligent, and cost-effective maintenance strategies for the entire transportation system. To achieve this overarching aim, the specific objectives of this research are articulated as follows:

1. To develop and validate advanced AI-based models capable of accurately identifying the location and estimating the depth of damage in bridge beam components, using

machine learning techniques.

2. To extend this work by developing deep learning models that can simultaneously predict the location, depth, and width of damage in bridge beams, enabling a more detailed and comprehensive damage assessment.
3. To establish a reliable AI-based framework for assessing road networks by combining localized damage detection (such as identifying potholes) with the evaluation of overall ride quality, and linking these assessments to cost-related models for effective maintenance planning.
4. To implement and evaluate multi-class classification algorithms that can accurately distinguish between different types of road surface damage (such as potholes, rutting, and speed bumps) using vibration data collected from vehicles, thereby supporting more targeted and efficient repair interventions.
5. To demonstrate the adaptability and practical utility of the proposed AI-based vibration monitoring framework by successfully applying its core principles and methodologies to both laboratory-scale bridge beam experiments and real-world road network field studies, thereby showcasing its potential for integrated, network-wide health management.

The initial set of two objectives relates to bridge components, the subsequent objectives focus on road networks, and the final objective highlights the adaptability of the proposed framework across both domains.

1.4 Thesis organization

This thesis is organized into seven chapters, each building upon the preceding one to present a comprehensive account of the developed AI-based vibration monitoring framework for structural damage detection in transportation infrastructure as shown in Fig. 1.6.

A brief overview of the contents of each chapter is presented below:

1. Chapter 1: Introduction sets the stage by providing a background on the critical importance of transportation infrastructure, highlighting the limitations of conventional monitoring methods, and introducing the motivation and specific objectives of this research.

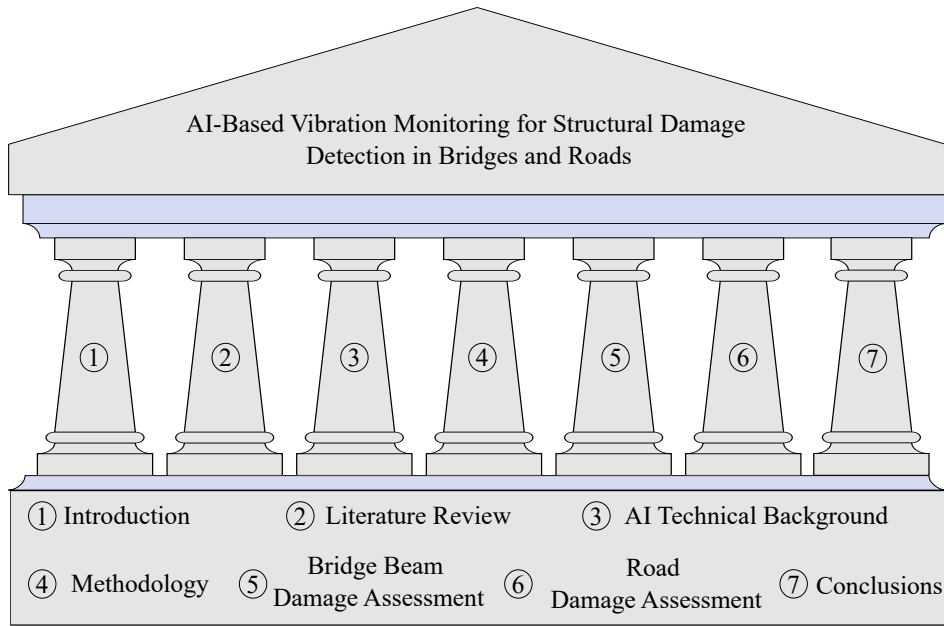


Figure 1.6: Thesis Organization

1. Chapter 2: Literature review offers a comprehensive review of existing SHM techniques, with a particular focus on vibration-based methods and the evolving role of AI in civil engineering applications. This chapter identifies key research gaps that this thesis aims to address.
2. Chapter 3: Technical background of AI models and evaluation metrics provides an in-depth overview of the AI models employed in this research, including their architectures, training procedures, and the mathematical formulations underlying their operation. This chapter also discusses the selection and calculation of key performance metrics used to evaluate model accuracy, robustness, and generalizability.
3. Chapter 4: Proposed methodology for AI-based vibration monitoring and damage assessment details the unified framework developed in this research. It systematically outlines the common principles and specific adaptations for data acquisition, feature engineering, and the application of various AI algorithms for damage assessment across both bridge beam structures and road networks.
4. Chapter 5: Bridge-beam damage assessment presents the empirical findings for beam-like structural components. It details the hybrid dataset, frequency-feature preparation, MIMO regression using machine learning (ML) and deep learning (DL) models, and quantitative evaluation for joint prediction of damage location, width,

and depth. The chapter reports metrics, error analyses, and case-wise validations on laboratory beam tests.

5. Chapter 6: Road damage assessment reports the application to roadway condition monitoring. It describes the acceleration-based data acquisition, sliding-window feature extraction, random forest classification, and comparative benchmarking. Results include confusion matrices, precision–recall analyses, and deployment-oriented discussion demonstrating the framework’s effectiveness for multi-class road damage detection.
6. Chapter 7: Conclusions and future work synthesizes the key findings from the entire research, discusses the significant contributions of this thesis to the field, highlights the broader implications for intelligent infrastructure management, and outlines promising avenues for future research.

Literature Review

2.1 Introduction

The growing success of AI, particularly ML and DL, has significantly influenced the field of SHM in recent years. These data-driven approaches have shown exceptional capability in modeling complex patterns, enabling real-time and accurate detection of structural anomalies. Motivated by these advancements, this research focuses on AI-based vibration monitoring techniques for damage detection and quantification in civil infrastructure—specifically bridges and roads. This chapter provides a comprehensive review of relevant literature, with a primary focus on VBM. It examines the development and application of ML and DL techniques for processing time-series vibration data to detect, localize, and quantify structural damage. Foundational concepts, model architectures, feature extraction strategies, and performance evaluation metrics are discussed in detail. In particular, the review highlights how hybrid approaches and advanced AI frameworks can address the limitations of traditional methods and enhance prediction accuracy and robustness.

The rest of this chapter is structured to provide a clear and logical progression of concepts and research developments. Section 2.2 introduces ML techniques, outlining the regression and classification models for structural components and roads, and their relevance in processing vibration data for structural damage detection. Section 2.3 explores DL architectures, emphasizing their role in regression based multi-output models. Section 2.4 identifies critical research gaps and limitations in the existing literature, including the lack of unified, multi-task models and real-time deployment strategies.

2.2 Machine learning

The application of ML in SHM has significantly evolved over the past decade, offering intelligent and data-driven alternatives to traditional rule-based or purely signal-processing frameworks. Conventional SHM approaches—such as threshold-based anomaly detection

and visual inspection—often suffer from key limitations [11]. These include heavy reliance on prior knowledge of the structure, sensitivity to environmental and operational variability, and difficulty in generalizing across different structural types or loading conditions. Moreover, many signal-processing-only methods require extensive manual feature engineering and may fail to detect subtle damage patterns embedded in noisy or high-dimensional vibration data [12]. In contrast, ML models are capable of learning complex patterns from vibration signals and subsequently mapping these features to damage states, thereby facilitating automated and accurate detection, localization, and quantification of structural damage in civil infrastructure [13]. In VBM, ML methods are primarily employed in two major tasks: (i) regression, to localize and quantify damage severity. These models rely on features extracted from time- or frequency-domain representations of vibration signals obtained from sensors mounted on structures, and (ii) classification, to detect the presence or type of damage. Common features include natural frequencies, frequency response functions, mode shapes and other statistical and derived features [14], [15].

2.2.1 Regression-based ML models for bridge components

Regression-based ML models have been extensively used in SHM for predicting quantitative parameters such as damage location, damage severity, frequency shifts, or stiffness degradation. These models are particularly relevant for applications involving structural components like beams, columns, plates and trusses [16, 17, 18, 19], where damage is characterized by measurable changes in modal parameters. Among these structural forms, beam elements are of particular interest in this research due to their ubiquitous application and well-defined dynamic behavior. Beams are fundamental load-carrying components in bridges, buildings, towers, and other civil infrastructure systems [20]. Furthermore, beams are often the first elements to exhibit early signs of fatigue or cracking under cyclic or excessive loading, thereby acting as early indicators of structural distress.

Numerous studies have explored the dynamic characteristics of beam structures for structural damage detection (SDD), focusing primarily on mode shapes and natural frequencies. These features are known to be highly sensitive to even minor changes in stiffness, making them effective tools for damage identification and localization. For instance, researchers have simulated real-world conditions by introducing noisy signals to evaluate the effectiveness of various damage indices such as strain energy and modal curvature.

Population-based optimization algorithms like the ant lion optimizer and frequency mode shape-based techniques have also been proposed to improve localization accuracy with fewer parameters [21, 22]. The literature further highlights that beam structures are commonly subjected to a wide range of damage scenarios. Altunicsik et al. [23] conducted operational modal analysis on hollow circular cross-sections and observed the influence of multiple crack configurations across six damage scenarios. Damage in beams commonly results from repeated loading, environmental exposure (e.g., corrosion), or sudden overloading events—all of which can lead to the initiation and propagation of cracks [24, 25]. If undetected, these cracks can coalesce and severely compromise structural integrity, especially in metallic beams used in bridges and towers. Studies have shown that the accumulation of multiple minor cracks can lead to disproportionately large structural failures, underscoring the importance of early, accurate, and quantitative damage assessment.

Several studies have proposed a wide range of methodologies for SDD in beam structures [26, 27, 28, 29, 30, 31, 32, 33, 34]. For example, Ahmad et al. [35] performed both analytical and experimental vibration analysis on cantilever beams and estimated the first four natural frequencies using the fast fourier transform (FFT) applied to measured vibration data. In another notable contribution, Rucevskis et al. [36, 37] developed a damage detection algorithm that does not require baseline (healthy-state) data, instead relying solely on the modal features of damaged structures—such as mode shapes and natural frequencies—for localization. Building on these foundational efforts, more advanced techniques have been developed using modal curvatures and damage indices to enhance sensitivity and precision. Garrido et al. proposed two relative methods to assess damage severity in simply supported beams. Their analytical approach employed changes in modal curvature as a key indicator for quantifying structural degradation. The corresponding experimental setup, which incorporated accelerometers to capture vibration data under various damage scenarios, is illustrated in Fig. 2.1. Similarly, Jassim et al. [38] presented a comprehensive review of vibration-based methods for damage detection in cantilever beams, highlighting various indices that are effective for both localizing and quantifying damage.

The integration of ML into SDD has marked a significant shift from purely physics-based or statistical approaches to intelligent, data-driven frameworks [39, 40, 41]. By leveraging vibration data, ML algorithms are capable of identifying complex patterns that may not be apparent through traditional modal analysis alone, thereby improving the ac-

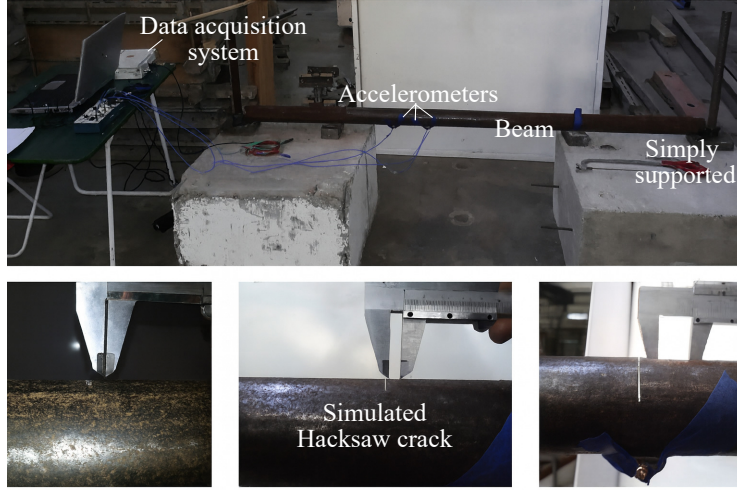


Figure 2.1: Beam Damage Detection Experimental Setup [1].

accuracy and reliability of damage quantification in structural components [42, 43]. Among the earliest regression-based models, linear regression (LR) has been applied to establish relationships between modal parameters — such as natural frequencies and damping ratios — and damage severity in simplified laboratory setups [44, 45]. These models offer interpretability and computational efficiency but struggle to capture the non-linearities introduced by complex geometries, heterogeneous materials, and environmental variability [46]. Probabilistic classifiers such as Naïve Bayes have also been applied in early SDD frameworks for classifying structural states based on statistical distributions of features [47, 48], offering fast training and robustness to small datasets, though often at the cost of reduced accuracy when feature independence assumptions are violated. Several studies have demonstrated the effectiveness of LR and Naïve Bayes in controlled environments for baseline damage detection in beams and truss structures [49, 50]. Tree-based ensemble regression models such as random forest (RF) regression and gradient boosting machine (GBM) —have emerged as powerful tools for predicting multiple damage parameters from vibration-based features. These models are non-parametric and can effectively handle high-dimensional inputs like modal frequencies, damping ratios, and curvature changes, which are common in structural health monitoring scenarios. The RF approach, introduced for SHM by [51] and later adapted for vibration-based damage detection by [52, 53], employs bootstrap-aggregated decision trees to reduce variance and enhance stability. Meanwhile, boosting methods, refined in SHM applications by [54, 55, 56], build models sequentially to correct prediction errors iteratively, achieving high accuracy even with limited or noisy datasets. The adaptability of these models to varied structural types—ranging from beams

to large-span bridges—has been demonstrated in works by [57, 58], highlighting their generalization capability across different operational conditions.

Table 2.1: Review of regression-based ML models for vibration-based damage detection

Method / Algorithm	Description	Input Features	Contribution / Application	Reference
LR	Regression between modal parameters and damage severity	Natural frequencies, damping ratios	Baseline prediction of damage severity in beams/trusses	[44, 45]
Naïve Bayes	Probabilistic classification of structural states	Statistical feature distributions	Fast detection in controlled lab setups	[47, 48]
RF	Bootstrap-aggregated decision trees	Modal frequencies, damping, curvature	Multi-output prediction of damage depth and location	[51, 52]
GBM	Sequential boosting to reduce prediction error	Modal frequencies, damping, curvature	High accuracy with noisy or limited datasets	[54, 56]
Hybrid: RF + PSO	Ensemble with meta-heuristic optimization	Vibration-based features	Improved hyperparameter tuning and generalization	[59]
Hybrid: Ensemble + ACO	Tree models combined with ant colony optimization	Vibration-based features	Balanced robustness and regression precision	[60, 61]

A key advantage of these ensemble methods is their ability to provide feature importance scores, enabling the identification of the most sensitive vibration features for structural degradation detection, as emphasized by [62, 63]. This interpretability aids in optimizing sensor placement and improving model transparency for engineering decision-making. Studies by [64, 65] have shown that tree-based methods, when integrated with VBM frameworks, outperform traditional statistical approaches, particularly under uncertainty or environmental variability. Furthermore, these models demonstrate robustness against overfitting and maintain strong predictive performance in real-world conditions, making them highly suitable for automated, near real-time structural monitoring in civil infrastructure.

Building upon the strengths of tree-based ensemble regression models, recent research has increasingly focused on hybrid-based approaches that integrate multiple learning paradigms to further enhance SHM performance. Such models aim to leverage the complementary capabilities of different algorithms—for example, combining the interpretability and feature selection efficiency of ensemble trees with the nonlinear mapping power of kernel-based methods or neural networks [66, 67, 68]. This integration allows hybrid frameworks to handle diverse structural responses, noisy environments, and multi-output prediction tasks more effectively than standalone models [69, 70]. For instance, [60, 61] fused ensemble models with ant colony optimization (ACO) to exploit the former’s robustness to high-dimensional inputs and the latter’s precision in regression. Similarly,

[59] demonstrated that coupling RF regression with particle swarm optimization (PSO) improved hyperparameter tuning and generalization capability. These hybrid approaches offer a promising pathway toward achieving high accuracy, adaptability, and scalability in real-time structural damage detection and quantification across varied civil infrastructure systems.

2.2.2 Classification-based ML models for road damage detection

Road health monitoring is a critical component of transportation asset management because deteriorating pavements increase vehicle operating costs, compromise safety, and reduce ride comfort [71, 72]. As summarized in Fig. 2.2, AI-based road condition assessment broadly follows two streams: *vision-based* and *vibration-based*. On the **vision-based** methods operate at three levels of analysis: image classification (assigning an overall label), object detection (localizing discrete defects such as potholes with bounding boxes), and semantic segmentation (pixel-wise masks for cracks/rutting). Representative works include Cao et al. [73], who used a convolutional neural network (CNN) pipeline for real-time road damage detection from dashcam imagery, and Ahmed et al [74], who reconstructed pothole geometry from stereo images. While these approaches provide rich visual documentation, their performance is sensitive to illumination, shadows, weather, and occlusions, and they often demand substantial labeled data and compute for reliable deployment in the wild [75]. Moreover, with monocular imagery the absolute geometry of defects—especially pothole depth—cannot be recovered reliably; depth typically requires calibrated LiDAR and controlled setups [76, 77]. These limitations motivate the **vibration-based**, where vehicle-mounted inertial sensing feeds either traditional ML or DL pipelines to infer surface condition from dynamic responses. In this paradigm, speed-normalized acceleration features combined with quarter-car models or supervised mappings can act as proxies for defect severity and even depth estimation, provided vehicle parameters are accounted for [78, 79].

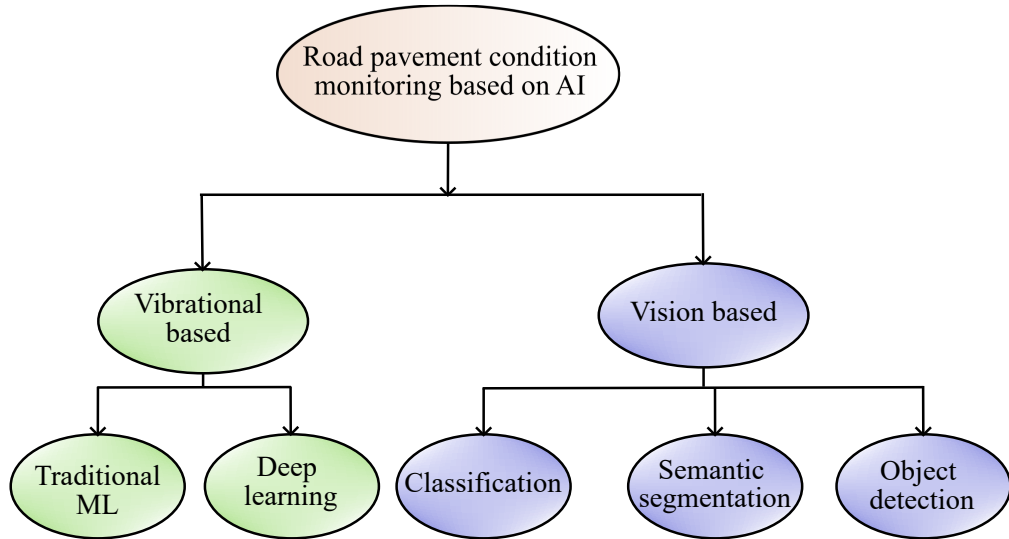


Figure 2.2: Road condition monitoring based on AI techniques [2]

Vibration-based methods use inertial sensing to capture vehicle–road interaction dynamics (Fig. 2.3). These systems exploit the fact that surface irregularities induce measurable changes in vehicle vibrations, which can be detected and analyzed to infer road surface conditions. Such approaches are generally more robust to environmental factors



Figure 2.3: Accelerometer placement on vehicle: (a) overview of sensor installations; (b) direction of sensor box installation; and (c) placement of sensor box [3]

and can be implemented using low-cost sensors such as accelerometers and gyroscopes. In vibration-based road health monitoring, vehicle-mounted sensing platforms play a central role in capturing surface condition data. These platforms often use dedicated inertial measurement units comprising tri-axial accelerometers and gyroscopes, or commodity smartphones, sometimes combined with GPS for speed and location tracking [80, 81]. Mounting location significantly affects the recorded signals: sensors placed on the cabin or chassis tend to produce cleaner data with reduced noise, while sensors mounted closer to the axle capture stronger transient responses but are more sensitive to vehicle dynamics and suspension effects [82]. Typical implementations operate at sampling rates between 50–1200 Hz, with careful attention to maintaining sensor orientation and synchronizing with vehicle speed to minimize velocity-induced distortions [83].

In vibration-based road health monitoring, raw accelerations are typically conditioned to improve signal-to-noise ratio by removing the gravity component (via static alignment or high-pass filtering), applying band-pass filters (e.g., 0.5–30 Hz) to suppress drift and irrelevant high-frequency noise, and normalizing to account for speed variability [84]. Additional cleaning often includes outlier suppression to mitigate transient spikes from non-road impacts [83, 85]. When GPS is available, speed can be incorporated as a covariate during model training or used to reparameterize signals in the distance domain, thereby reducing speed-induced variability [86]. From these conditioned signals, compact descriptors are derived for learning tasks: (i) time-domain statistics such as mean, variance, peaks, crest factor, skewness, and kurtosis to capture abrupt transients; (ii) frequency-domain measures from FFT (spectral centroid, dominant frequency, band energies) to characterize periodicity and roughness; and (iii) time–frequency representations via wavelets to detect localized, multi-scale events [87, 88, 89]. In network-level assessments, simplified vehicle–road interaction models (e.g., quarter-car) are also employed to map vibration responses to IRI, linking dynamic measurements to standardized ride-quality metrics [90, 91].

Given the non-stationary nature of road-induced vibrations, segmentation of the signal into fixed-length, overlapping windows is a standard practice prior to feature computation. For a sampling rate f_s , a window of length T seconds contains $N = \lfloor f_s T \rfloor$ samples. Window durations in the range of 0.25–1.0 s, with overlaps of 50–75%, are common; shorter windows enable finer localization of events, while longer windows provide more stable statistical measures across varying speeds [92, 93]. Label assignment strategies vary: in

event-present labeling, a window is marked positive if it overlaps any annotated defect, while centered-event labeling ensures the defect lies near the center of the segment. To avoid information leakage and ensure realistic evaluation, route-based or trip-wise data splits are often adopted during model validation [94, 95]. Recent works have also proposed combining localized defect detection windows with segment-level IRI estimation to support both network-wide pavement evaluation and targeted maintenance.

Bansal et al. [96] conducted classification using various ML models and proposed RF as the most suitable model for the binary classification of potholes. Bhatlawande et al. [97] developed a vision-based method for pothole classification by collecting images from existing literature and found that the RF classifier performed better for binary classification. Borlea et al. [98] conducted studies to enhance the quality of clusters in the supervised KNN algorithm without reducing processing time. Protic et al. [99] proposed an XOR-based detector to identify anomalies in computer network traffic, explaining that the only way to reduce the size of datasets is by converting them into features. For this purpose, they employed four binary classifiers: k-nearest neighbors (KNN), decision tree (DT), feedforward neural network (FNN), and support vector machine (SVM). Zhao et al. [100] designed an approach using particle clustering-deep Q-network (PC-DQN) for the localization of poisonous gas sources. They introduced the concept of transfer learning to reuse the trained Q-network for new scenarios. Ghadge et al. [101] used the RF classifier to detect road bumps, allowing drivers to prepare in advance.

Researchers have also conducted studies to detect and classify localized road damages such as potholes using different ML models such as SVM, RF, neural networks, etc. [10, 102]. [103] adopted an ML-based pothole detection approach that relies on features extracted from simulated vehicle sensor data. The study reported that it achieved a high level of accuracy (99.6% simulated, 88.9% empirical). Bhatt et al. [104] conducted a comparative study of SVM and GBM models using data collected from accelerometers and gyroscopes on an iPhone 6S, reporting accuracies of 92.9% and 92.02%, respectively, although precision (0.78) and recall (0.42) values were lower. Other studies have focused more on specific pothole detection methods. Koch et al. [105] developed an image-based automated pothole detection system using histogram shape-based thresholding, morphological thinning, and elliptic regression. However, the method struggles with accurately detecting certain lighting conditions, such as when the sun is overhead, and does not con-

sider maintenance assessments. Wu et al. [106] integrated mobile point cloud and images for pothole detection, but their approach does not assess the ride quality of the pavement. Chibani et al. [92] introduced a dynamic sliding window technique, though it lacks emphasis on broader road condition assessments. Gu et al. [107] proposed a spatial temporal kernel density estimation (STKDE) method to forecast pothole likelihood across space and time, but misclassifications could lead to incorrect pothole reporting, and the study does not account for road maintenance or ride quality.

To quantify overall road condition, researchers have adopted a range of pavement condition indicators. These indicators are generally categorized as profile-based or response-based. The profile-based indicators use direct statistical approaches that rely on vertical road elevation measurements and include the power spectral density (PSD) of the road elevation [108], statistics in specific wavebands [109], and various straightedge indexes [110]. Beyond these, broader pavement condition assessment frameworks also consider rut depth, cracking or aggregated surface distress indices, structural capacity, and surface friction, although the last two are less frequently used in routine evaluations. Many of these indices require specialized imaging, profiling, or ground-penetrating measurement systems and cannot be fully captured using vibration data alone.

The response-based approach relies on indirect statistics from vehicle-mounted sensors and encompasses several indices, including the International Roughness Index (IRI), profile index (PI), ride number (RN), mean roughness index (MRI), Half-car roughness index (HRI), and the Maysmeter index (MI) [111, 112]. Among these, the IRI has become the most widely used due to its stability over time and transferability across different regions worldwide [113, 7]. A general approach to quantifying road roughness involves collecting sensory data using an instrumented vehicle. For example, Cantisani et al. [114] presented a model applicable to real and standard vehicles; however, their study focuses primarily on lower speed limits and does not fully address the challenges posed by degraded roads. Tedeschi et al. [115] developed an automatic pavement distress recognition system based on the OpenCV library, although the framework does not cover all pavement distress categories. Similarly, Li et al. [116] introduced a particle swarm optimization (PSO) algorithm enhanced with a gated recurrent unit (GRU) neural network, but their approach centers on pavement performance prediction rather than direct maintenance and health monitoring.

2.3 Deep learning

2.3.1 Regression-based models for bridge components

DL employs multi-layer neural architectures that learn hierarchical features directly from data, offering clear advantages over conventional ML for SHM of bridge beams. DL enables end-to-end learning from raw accelerations, strains, displacements, or images, captures nonlinear and path-dependent beam behavior under moving loads and environmental variability, and models spatio-temporal structure through 1D CNN feature extractors coupled with sequence models. Crucially, DL supports multi-output regression (mapping input features $\mathbf{X} \in \mathbb{R}^{T \times d}$ to targets $\mathbf{y} \in \mathbb{R}^m$), multimodal fusion, and uncertainty quantification. This section reviews DL regression approaches for bridge components—spanning vibration, strain, and vision data, calling out where DL clearly improves accuracy, robustness, and real-world deployability.

Researchers such as [120, 121, 122, 123, 124, 125] have employed many deep learning techniques for damage detection and quantification in structures. They used models such as forest-based algorithms and neural networks. Among these models, many focused on RF algorithms, CNN, long short-term memory (LSTM)s, etc. Building on advancements in deep learning, [126] introduced an innovative approach to SDD in data-scarce scenarios, leveraging an adaptive simulated annealing PSO-CNN (ASAPSO-CNN) framework augmented by noise-based data enhancement techniques, demonstrating superior accuracy and robustness when validated on experimental beam models and real-world steel truss bridges. DL models, such as the integration of deep neural networks (DNNs) with optimization algorithms like the evolved artificial rabbit optimization (EVARO), have shown significant promise in accurately localizing and quantifying structural damage in truss bridges [127, 128]. [129] proposed a novel SDD method integrating surrogate-assisted evolutionary optimization algorithms, demonstrating the potential of combining advanced surrogate modeling techniques with robust optimization strategies like the improved termite life cycle optimizer (ITLCO) to enhance accuracy and computational efficiency in damage detection.

Roy et al. [130] proposed DenseSPH-YOLOv5, an advanced damage detection model, achieved a mean average precision of 85.25% and F1-score of 81.18% at a detection rate of 62.4 FPS, outperforming existing state-of-the-art models, demonstrating its robustness

and efficiency in real-time applications. [131] MGLNN framework, proposed for multi-graph semi-supervised learning, outperformed state-of-the-art methods with classification accuracy improvements of up to 3.8% on benchmark datasets, while demonstrating robust scalability across varying graph complexities. [132] proposed a systematic approach integrating GBM was proposed to optimize empirical coefficients for 2D VARANS-VOF models, achieving an exceptional regression coefficient ($R^2 \geq 0.99$). The BCMO-ANN algorithm investigated by [133] effectively optimized the vibration and buckling behaviors of functionally graded porous (FGP) microplates, utilizing artificial neural networks to predict stochastic responses with material uncertainties, while analyzing the influence of modal parameters such as natural frequencies. In conclusion, the utilization of DL models in SHM showcases promising advancements in damage detection and quantification.

2.4 Gaps in literature

Despite substantial progress, notable gaps remain across the two related problem domains considered in this work—damage detection and quantification in beam-like structural components and road health monitoring using vehicle-borne sensors. In both areas, much of the literature narrows the scope to isolated subproblems, depends on strong prior assumptions about the structure or operating environment, and reports performance that falls short of the accuracy required for practical deployment.

For bridge beams, most studies address either where damage localization or quantification, but seldom both within a single predictive framework. Approaches frequently rely on dense instrumentation, detailed finite-element surrogates, or extensive modal testing, which limits scalability across sites and operating conditions. Integration of experimental and numerical evidence also remains sparse, constraining generalization. In this work, the problem is tackled in two stages aligned with these gaps: first, a ML framework is developed for dual-output prediction, providing simultaneous estimates of two damage attributes; second, a DL framework based on a multi-output sequence model is introduced to concurrently infer three damage attributes—expanding beyond the ML stage while preserving a unified treatment of localization and severity. Both stages are designed for periodic (rather than continuous) monitoring using routinely collected responses, reducing infrastructure burden while moving toward a practical, joint assessment of beam damage.

Table 2.2: Limitations of existing literature for road damage detection

Method Category	Ref.	Proposed Approach	Limitations in Existing Methods
Vision-Based	[134]	Detailed repair cost estimation provided	Lack of cost estimation
	[135]	Cost-effective equipment enabling depth analysis of road damages	Expensive equipment, insensitive to depth
	[136]	Consideration of multiple road damage types	Sole focus on potholes
	[137]	Multi-class classification approach	Limited to binary classification
Vibration-Based	[138]	Utilization of multiple algorithms with comparative analysis	Sole use of SVM algorithm
	[139]	Implementation of multi-class classification	Binary classification only
	[92]	Integration of sliding window technique with optimized window size	Dependency on dynamic sliding window technique
	[10]	Implementation of multi-class classification	Restricted to binary classification

In road health monitoring, the literature remains largely focused on a single damage class—most commonly potholes—and on binary decisions, with many pipelines tied to costly vision hardware or to vibration-only schemes that fix sliding-window settings heuristically and evaluate a single classifier. Such choices limit robustness across vehicles, speeds, and surface types, and they offer limited support for maintenance workflows that require prioritization and cost estimation. Framing road assessment as a multi-class problem and employing a RF classifier on vibration features provides a more versatile alternative that better accommodates heterogeneous signals while remaining computationally efficient for edge or fleet deployment. A consolidated comparison in Table 2.2 situates these limitations alongside the present approach, outlining how broader damage coverage, tuned windowing, and algorithmic diversity translate into improved operational usefulness and decision support.

Background : Formulation of AI Models in SHM

3.1 Introduction

SHM has undergone a significant transformation—from manual visual inspections to sophisticated vibration-based diagnostic systems. Traditional rule-based SHM approaches, while foundational, are often constrained by their dependence on expert interpretation, susceptibility to environmental noise, aging effects, and limited scalability across large-scale infrastructure. The emergence of AI, particularly ML and DL, has brought a paradigm shift, enabling automated, data-driven, and scalable solutions for real-time monitoring. The evolution of AI technologies is illustrated in Fig. 3.1. These techniques utilize sensor signals—especially vibration data—to detect, localize, quantify, and classify structural damage with improved accuracy and robustness, addressing many of the inherent limitations in conventional methods.

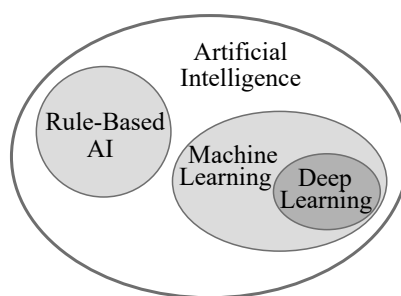


Figure 3.1: Categorization of AI technologies

AI broadly refers to computational methods that emulate human cognitive functions such as learning, reasoning, and decision-making. Early AI systems were primarily rule-based, designed with fixed instructions suitable for well-structured problems like chess, but ill-equipped for adaptive and noisy environments, such as those encountered in SHM. To overcome these challenges, modern AI embraces data-centric learning through ML, a subset of AI that allows systems to learn patterns from data without explicit programming. DL, a further subset of ML based on deep neural networks, enhances this capability by

autonomously learning features from raw, high-dimensional inputs. These advancements have made AI particularly effective in SHM applications, where models must adapt to variability, noise, and nonlinear system behavior.

This chapter is structured to provide the theoretical and methodological background on the formulation of AI models in SHM. Section 3.1 introduces the scope and relevance of AI-based approaches in SHM. Section 3.2 discusses machine learning models, beginning with the mathematical foundations of prediction functions (section 3.2.1) and followed by model evaluation metrics (section 3.2.2). Section 3.3 focuses on deep learning models, covering their architectures (Section 3.3.1), mathematical formulations (section 3.3.2), loss functions and optimization strategies (section 3.3.3), and evaluation and inference processes (section 3.3.4). Finally, section 3.4 summarizes the key insights from the chapter and sets the stage for the methodology presented in the subsequent chapter.

3.2 Machine learning models

ML encompasses a variety of approaches—including supervised, unsupervised, and semi-supervised learning—that are increasingly applied in SHM. These data-driven methods leverage sensor signals to automate the detection of structural anomalies, offering robust alternatives to manual or rule-based diagnostic systems. Among these approaches, supervised learning is the most widely used in SHM, particularly when labeled data (i.e., measurements associated with known damage states) are available. Supervised ML models learn patterns in vibration responses that correspond to damage presence, location, severity, or type, and subsequently generalize this knowledge to new cases. This makes them especially suitable for tasks such as damage detection, localization, quantification, and classification.

In contrast, unsupervised learning operates without labeled data and instead seeks to identify intrinsic structure or unexpected changes directly from sensor measurements. Typical unsupervised techniques in SHM include clustering, dimensionality reduction, and anomaly detection, which are valuable when damage labels are scarce or unavailable. Semi-supervised learning occupies a middle ground between supervised and unsupervised approaches by leveraging a small amount of labeled data together with a larger pool of unlabeled data. This makes it particularly useful in SHM settings where labeling damage states

is costly or infeasible, allowing models to improve classification or regression performance without requiring exhaustive annotation. While these methods contribute significantly to data-driven SHM, the present study focuses exclusively on supervised learning due to the availability of labeled vibration data and the need to predict specific damage-related outputs.

The overall development process for supervised ML in SHM is illustrated in Fig. 3.2. The learning process begins with vibration data acquisition, where accelerometers or similar sensors collect dynamic responses from structures. This raw data is then passed through feature engineering, which transforms complex signals into structured information. Common features include time-domain statistics, frequency-domain metrics, or wavelet-based energy coefficients. Once features are ready, a training dataset is created by pairing them with known outcomes—this could be damage levels, types of defects, or severity ratings. With this data, we move into model training, where the model begins to learn patterns and relationships that map input features to their respective outputs. This is where the core prediction function $\hat{y} = f(\mathbf{x}; \theta)$ is learned, with θ representing the internal parameters of the model.

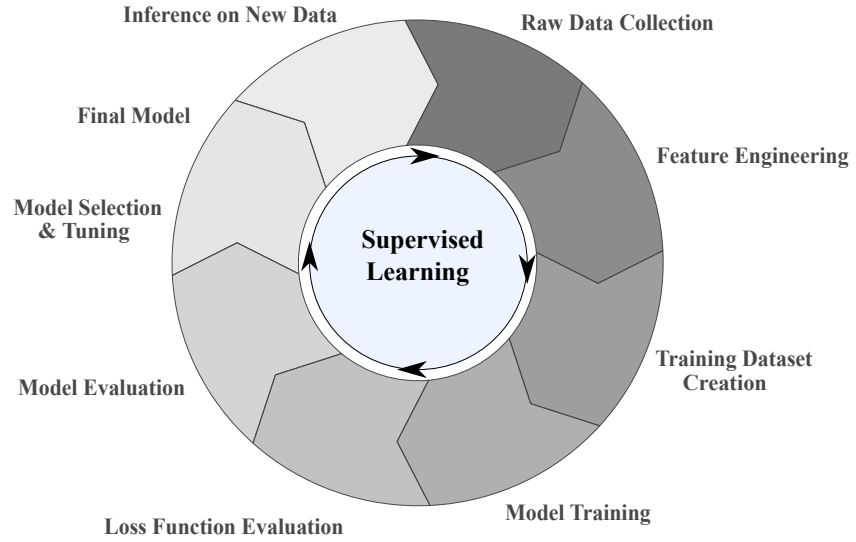


Figure 3.2: ML model development process

Later, during loss function evaluation, the performance of this learned function is assessed by checking how far off its predictions are from the actual labels. Model evaluation then verifies how well the model performs on new data using metrics like accuracy, F1-score, or RMSE. If needed, model selection and tuning is carried out to refine algorithms or adjust hyperparameters for better performance. Finally, in the inference stage, this pre-

diction function is used to estimate damage on new, unseen data—making it the critical bridge between training and practical SHM application.

3.2.1 Prediction function expressions

The learned prediction function $\hat{y} = f(\mathbf{x}; \theta)$, introduced above, varies depending on the machine learning algorithm applied. Each model—whether regression or classification—has its own mathematical form, objective function, and mechanism for learning parameters. This subsection summarizes the key formulations associated with the models used in this study. The expressions include prediction functions, loss objectives, and constraints, offering insights into how each model learns from data in SHM tasks.

3.2.1.1 Regression models

Regression-based machine learning models are widely used in SHM for predicting continuous outputs, such as damage severity, stiffness degradation, or vibration response metrics. These models learn from labeled input–output pairs and aim to generalize the relationship for unseen data.

(a) Linear regression

LR is one of the most widely used and fundamental models for predicting continuous outcomes. It assumes that the relationship between the independent variables and the dependent variable is linear and can be expressed as a weighted sum of the input features.

$$\hat{y} = \beta_0 + \sum_{i=1}^n \beta_i x_i \quad (3.1)$$

In this formulation, \hat{y} is the predicted output, x_i are the input features, and $\beta_0, \beta_1, \dots, \beta_n$ are the model coefficients. The parameters are typically estimated using the ordinary least squares (OLS) method, which minimizes the residual sum of squares between the observed values and the predicted outputs. LR is appreciated for its simplicity, speed, and ease of interpretation, making it a reliable baseline model for regression tasks. Each coefficient offers direct insight into the relationship between a feature and the output. However, the model’s performance can degrade when the true relationship is non-linear or when the data contains multicollinearity or influential outliers. In such cases, regularized or non-linear

models may offer improved accuracy.

(b) Random forest (RF)

RF is a robust ensemble learning method that operates by constructing a multitude of decision trees during training and averaging their outputs to enhance predictive performance and reduce overfitting (Fig. 3.3).

$$\hat{y} = \frac{1}{T} \sum_{t=1}^T h_t(\mathbf{x}) \quad (3.2)$$

where $h_t(\mathbf{x})$ denotes the prediction from the t^{th} decision tree, and T is the total number of trees in the ensemble.

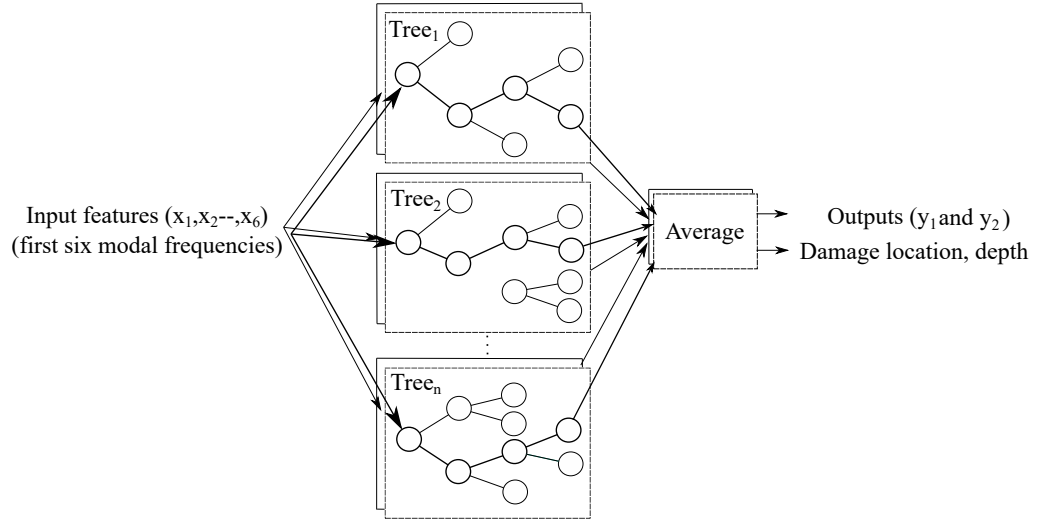


Figure 3.3: Working principle of random forest regressor

Each tree is trained using a bootstrapped sample of the data and a randomly selected subset of features, which introduces diversity and decreases variance without substantially increasing bias. RFs are well-suited for modeling non-linear relationships and capturing complex feature interactions. They are relatively immune to overfitting when compared to individual decision trees and are less sensitive to noise in the data. The model performs well even without feature scaling and handles both continuous and categorical variables effectively. Key hyperparameters—such as the number of trees, maximum depth, and minimum samples per split—are typically optimized using cross-validation or grid search. Despite its strong performance and robustness, the Random Forest model can be memory-intensive and lacks transparency, making it less interpretable than simpler models like linear regression.

(c) Gradient boosting

GBM regression is an iterative ensemble technique that builds models sequentially to minimize prediction errors. Each new learner is trained to correct the residuals of the previous ensemble, thereby improving overall performance over successive iterations.

$$\hat{y}_t = \hat{y}_{t-1} + \eta \cdot h_t(\mathbf{x}) \quad (3.3)$$

In this formulation, \hat{y}_t represents the updated prediction at iteration t , $h_t(\mathbf{x})$ is the newly trained weak learner (typically a decision tree), and η is the learning rate that controls the contribution of each learner to the final model. The process begins with an initial estimate—often the mean of the target variable—and incrementally refines it by adding learners that minimize the gradient of the chosen loss function (e.g., mean squared error (MSE)). Key hyperparameters include the number of boosting iterations, learning rate, tree depth, and subsampling ratio, which are tuned to avoid overfitting.

Gradient boosting is highly effective for capturing complex, non-linear relationships and frequently achieves high accuracy in regression tasks. However, its success heavily depends on careful parameter tuning, and it may become computationally expensive or prone to overfitting without proper regularization strategies.

(d) Support vector regression (SVR)

Support vector regression (SVR) extends the principles of SVMs to regression tasks. The goal is to find a function $f(\mathbf{x})$ that deviates from the actual target y_i by no more than a specified margin ε , while maintaining the model's flatness by minimizing the norm of the weights $\|\mathbf{w}\|$.

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b \quad (3.4)$$

$$\min_{\mathbf{w}, b} \quad \frac{1}{2} \|\mathbf{w}\|^2 + C \sum (\xi_i + \xi_i^*) \quad (3.5)$$

$$\text{subject to} \quad \begin{cases} y_i - f(\mathbf{x}_i) \leq \varepsilon + \xi_i \\ f(\mathbf{x}_i) - y_i \leq \varepsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0 \end{cases} \quad (3.6)$$

SVR allows for both linear and non-linear regression through the use of kernel functions such as the radial basis function (RBF), polynomial, or sigmoid kernels. The hyperparameters C , ϵ , and kernel parameters are usually determined through grid search or cross-validation. SVR is effective in capturing complex patterns and performs well on high-dimensional and small-to-medium-sized datasets. However, it can be computationally expensive and may require careful tuning of parameters to achieve optimal performance.

(e) **K-nearest neighbors (KNN)**

KNN Regression is a non-parametric method that predicts the target value of a new sample by averaging the outputs of its k nearest neighbors from the training dataset. The closeness between samples is typically evaluated using distance metrics such as Euclidean distance.

$$\hat{y} = \frac{1}{k} \sum_{j \in \mathcal{N}_k(\mathbf{x})} y_j \quad (3.7)$$

KNN is simple to implement and effective when local data patterns dominate, as it does not require an explicit training phase. This makes it computationally light during training but potentially expensive at inference time, especially for large datasets. Additionally, its performance is highly influenced by the choice of k , the presence of noise, and the curse of dimensionality in high-dimensional feature spaces. Proper feature scaling and dimensionality reduction techniques are often necessary to improve its effectiveness.

3.2.1.2 Classification models

Supervised classification models in ML are widely used for damage state identification in SHM, particularly for tasks like classifying bridge segments as damaged or undamaged, or distinguishing types of defects in road or beam structures. This subsection presents the mathematical formulations of the classification models employed in this study. The expressions define how each algorithm maps input features to predicted class labels, along with the associated loss functions or decision rules.

(a) Logistic regression

Logistic regression is a widely used classification algorithm that models the probability of a class label based on input features. It assumes a linear relationship between the input variables and the log-odds of the outcome.

For binary classification, the logistic function (sigmoid) is used:

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + e^{-\mathbf{x}^\top \boldsymbol{\beta}}} \quad (3.8)$$

For multi-class classification, the model is extended using the softmax function:

$$P(y = k \mid \mathbf{x}) = \frac{e^{\mathbf{w}_k^\top \mathbf{x}}}{\sum_{j=1}^K e^{\mathbf{w}_j^\top \mathbf{x}}} \quad (3.9)$$

Model parameters $\boldsymbol{\beta}$ or \mathbf{w} are estimated using maximum likelihood estimation (MLE), typically solved via gradient descent or quasi-Newton methods. Logistic regression is highly interpretable—each coefficient directly indicates the influence of a feature on the predicted probability. However, it is limited in capturing non-linear relationships and may underperform on datasets with complex feature interactions or high multicollinearity.

(b) Support vector machine (SVM)

SVM aims to identify the optimal hyperplane that maximally separates classes by maximizing the margin between the closest data points, known as support vectors. The model parameters are estimated by solving a constrained optimization problem, typically using quadratic programming.

$$f(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + b \quad (3.10)$$

SVM employs the Hinge loss function to penalize misclassified points:

$$\mathcal{L} = \sum_{i=1}^N \max(0, 1 - y_i f(\mathbf{x}_i)) \quad (3.11)$$

To handle non-linearly separable data, SVM utilizes kernel functions—such as the RBF, polynomial, or sigmoid—to project the input features into higher-dimensional spaces where a linear separation is feasible. SVMs are particularly effective in high-dimensional spaces and are known for their strong generalization on small to medium-sized datasets.

However, they can be computationally expensive for large-scale problems and lack interpretability due to the complexity of kernel transformations.

(c) K-nearest neighbors (KNN)

KNN is a non-parametric, instance-based learning algorithm that assigns a class to a sample based on the majority vote among its k closest training examples. The proximity between samples is typically calculated using distance metrics such as Euclidean or Manhattan distance. Since KNN stores the entire dataset, it does not involve a traditional training phase, making it simple to implement.

$$\hat{y} = \arg \max_c \sum_{i \in \mathcal{N}_k(\mathbf{x})} \mathbb{I}(y_i = c) \quad (3.12)$$

While KNN can effectively handle problems with complex decision boundaries, it is computationally expensive during inference due to the need to compute distances for each prediction. Additionally, its performance is sensitive to the choice of k , distance metric, and the scale of features, often requiring normalization or standardization.

(d) Naive Bayes

Naive Bayes classifiers apply Bayes' theorem with the strong assumption that input features are conditionally independent given the class. The posterior probability for a class C_k given an input feature vector \mathbf{x} is computed as:

$$P(C_k | \mathbf{x}) \propto P(C_k) \prod_{j=1}^n P(x_j | C_k) \quad (3.13)$$

The model estimates the class priors $P(C_k)$ and likelihoods $P(x_j | C_k)$ using frequency-based or Bayesian techniques. Despite its simplicity, Naive Bayes is highly efficient and often performs well in high-dimensional spaces, making it particularly useful for applications such as text classification. However, its performance can degrade when the assumption of feature independence is violated, as it fails to capture feature interactions.

(e) Decision tree (DT)

DTs are non-parametric models that classify data by recursively partitioning the feature space based on thresholds that maximize information gain or minimize impurity. Common

criteria used for selecting splits include the Gini index and entropy:

$$\text{Gini} = 1 - \sum p_k^2 \quad \text{or} \quad \text{Entropy} = - \sum p_k \log_2 p_k \quad (3.14)$$

The tree is typically built using greedy algorithms such as CART, which select the best split at each node. DTs are highly interpretable and allow for intuitive visualization of decision rules, making them useful for explaining model predictions. However, they are prone to overfitting, particularly when grown deep on noisy datasets. To enhance generalization, techniques such as pruning or integration into ensemble methods like RF or GBM are commonly employed.

(f) Random forest (RF)

RF is an ensemble-based classification method that aggregates the outputs of multiple DTs to enhance model accuracy and robustness. It builds each tree using bootstrapped samples of the training data and selects random subsets of features at each split to ensure diversity among trees. The final class prediction is determined by majority voting:

$$\hat{y} = \text{mode} \{h_1(\mathbf{x}), \dots, h_T(\mathbf{x})\} \quad (3.15)$$

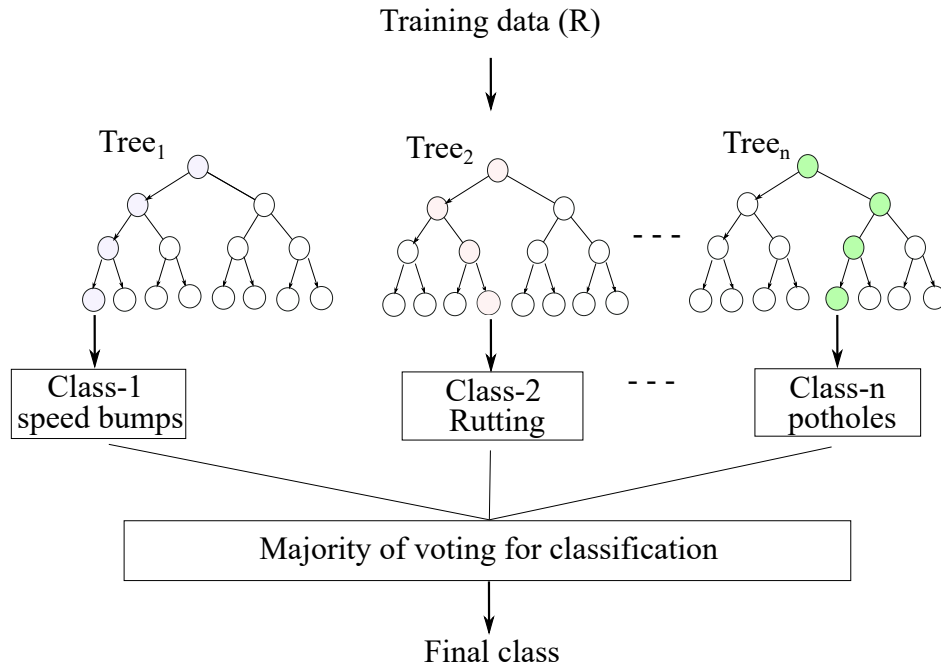


Figure 3.4: Working principle of random forest classifier.

Here, $h_t(\cdot)$ denotes the prediction from the t^{th} decision tree in an ensemble of T trees.

Parameters such as the number of trees, maximum tree depth, and minimum samples per split are typically optimized through cross-validation or random/grid search. RFs offer strong generalization performance, especially in high-dimensional and noisy datasets, and are less prone to overfitting than individual trees. However, they may require more memory and computation, and their ensemble nature can reduce interpretability compared to standalone decision trees.

(g) Gradient boosting machine (GBM)

GBM constructs trees sequentially, where each new tree corrects the residuals of the previous ones by fitting a weak learner to the gradient of the loss function. Parameters such as learning rate, tree depth, and number of estimators are fine-tuned using grid or random search. GBM models, are known for high accuracy and handling complex data patterns. However, they are sensitive to hyperparameters, require careful tuning, and may be slow to train.

$$P(y = k \mid \mathbf{x}) = \frac{e^{F_k(\mathbf{x})}}{\sum_{j=1}^K e^{F_j(\mathbf{x})}} \quad (3.16)$$

Where $F_k(\mathbf{x})$ is the accumulated prediction score for class k after boosting iterations.

Each of these machine learning models—whether regression-based or classification-based—has demonstrated varying levels of effectiveness in addressing structural health monitoring tasks. Their suitability often depends on the complexity of vibration data, the number of input features, and the trade-off between interpretability and predictive power. However, selecting an appropriate model also requires a careful evaluation of its performance on both training and unseen data.

3.2.2 Model evaluation metrics

To systematically evaluate the performance of both regression and classification models employed in this research, a range of standard evaluation metrics were utilized. These metrics serve to quantify model accuracy, robustness, and generalization ability. The selection of metrics was guided by the nature of the learning task—whether it involved predicting continuous values (regression) or class labels (classification).

(a) Regression metrics

The regression performance is reported using MSE, mean absolute error (MAE), median absolute error (MedAE), the coefficient of determination (R^2), explained variance (EV), mean squared logarithmic error (MSLE), and mean absolute percentage error (MAPE). MSE emphasizes large deviations by squaring the residuals, MAE summarizes the average absolute deviation, MedAE provides a robust central tendency of errors, R^2 quantifies the proportion of variance explained by the model, EV measures how much target variance is captured by predictions, MSLE focuses on relative errors on a logarithmic scale, and MAPE expresses errors as a percentage of the true values.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad \text{MAE} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|, \quad (3.17)$$

$$\text{MedAE} = \text{median}(|y_i - \hat{y}_i|), \quad R^2 = 1 - \frac{\sum_{i=1}^N (y_i - \hat{y}_i)^2}{\sum_{i=1}^N (y_i - \bar{y})^2}, \quad (3.18)$$

$$\text{EV} = 1 - \frac{\text{Var}(y - \hat{y})}{\text{Var}(y)}, \quad \text{MSLE} = \frac{1}{N} \sum_{i=1}^N [\log(1 + y_i) - \log(1 + \hat{y}_i)]^2, \quad (3.19)$$

$$\text{MAPE} = \frac{100}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right|. \quad (3.20)$$

Here \bar{y} is the sample mean of $\{y_i\}$, and $\text{Var}(\cdot)$ denotes variance.

(b) Classification metrics

For classification tasks (e.g., damaged vs. undamaged segments), performance is evaluated using metrics derived from the confusion matrix entries—true positive (TP), true negative (TN), false positive (FP), and false negative (FN)—as illustrated in Fig. 3.5. Accuracy measures overall correctness, precision quantifies the reliability of positive predictions, recall captures the ability to detect actual positives, and the F1 score balances precision and recall. For multi-class problems, these definitions are computed per class k in a one-vs-rest manner, with macro/micro averages reported as needed.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}, \quad (3.21)$$

$$\text{Precision}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FP}_k}, \quad \text{Recall}_k = \frac{\text{TP}_k}{\text{TP}_k + \text{FN}_k}, \quad (3.22)$$

$$\text{F1}_k = \frac{2 \text{Precision}_k \text{Recall}_k}{\text{Precision}_k + \text{Recall}_k}. \quad (3.23)$$

		Predicted classification		
Actual classification	Class-1	Class-2	Class-3	
	TP FN FP	TN FN TN	TN TN FN	
	FN TN TN	FP TP FP	TN TN FP	
	FP TN TN	TN FP TN	FN FN TP	

Figure 3.5: Confusion matrix in classification

(c) Model selection and inference

Following the evaluation, models were compared based on their metric scores, and the best-performing algorithms were selected through cross-validation. Hyperparameter tuning, including techniques such as grid search, was employed to refine the models further. Ultimately, the selected models were deployed in the final inference phase, where the trained prediction function

$$\hat{y} = f(\mathbf{x}; \theta)$$

was used to estimate the structural health condition from unseen vibration datasets. This systematic approach ensures reliability, robustness, and interpretability in real-world structural health monitoring applications.

3.3 Deep learning models

DL models are particularly effective in SHM tasks that involve high-dimensional, sequential, or complex raw sensor data—such as vibration signals from bridges and roadways. Unlike traditional ML, which depends heavily on hand-crafted features, DL models can

automatically extract and learn hierarchical representations from raw input, making them ideal for end-to-end damage detection or localization.

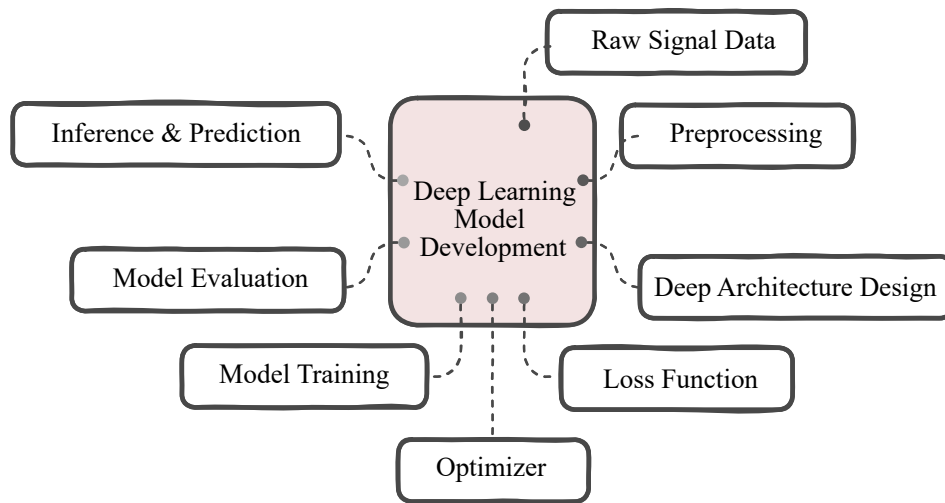


Figure 3.6: DL model development process

When sensor data exhibit temporal patterns (as in time-series vibration data) or spatial features (in frequency-transformed domains). DL models are particularly suitable here, as they minimize the need for extensive manual feature engineering and instead enable automatic learning of the most relevant features for damage detection directly from raw data. Figure 3.6 illustrates the overall deep learning model development pipeline used in this research. It encompasses all key stages—from raw signal data acquisition, preprocessing, and architecture design to training, evaluation, and inference. Each component is further explained in detail in the following subsections.

3.3.1 Architectures

In DL, an architecture refers to the structural design of a neural network—how layers are organized, connected, and configured to process input data. The architectures used in this thesis include a FNN, suitable for static inputs and regression; a multilayer perceptron (MLP), which introduces one or more hidden layers to model complex non-linear relationships for both regression and classification; and recurrent neural network (RNN)s designed for sequential/time-series data. Two gated RNN variants are employed: LSTM, which uses input, forget, and output gates to capture long-range dependencies and mitigate vanishing gradients, and the gated recurrent unit (GRU), a lighter alternative that often attains comparable performance with reduced computation. In addition, a hybrid 1D CNN–GRU–ResNet

model combines convolutional layers for local feature extraction, GRUs for temporal sequence modeling, and residual connections for stable optimization, making it well suited to learning from vibration signals in structural health monitoring.

3.3.2 Mathematical formulations

DL architectures are defined by how layers are organized, connected, and parameterized to transform inputs into targets. The mathematical expressions below make explicit the forward computations, while the accompanying text clarifies the role of each term and how parameters are estimated from data. Figures are placed exactly as in the previous code to preserve layout and cross-references.

(a) FNN

A feed-forward neural network maps an input feature vector $\mathbf{x} \in \mathbb{R}^d$ to a multi-output prediction $\hat{\mathbf{y}} \in \mathbb{R}^m$ via a hidden representation as shown in Fig. 3.7:

$$\hat{\mathbf{y}} = f(\mathbf{x}) = \mathbf{W}_2 \sigma(\mathbf{W}_1 \mathbf{x} + \mathbf{b}_1) + \mathbf{b}_2. \quad (3.24)$$

Here, $\sigma(\cdot)$ is a nonlinearity (e.g., rectified linear unit (ReLU) or tanh); $\mathbf{W}_1 \in \mathbb{R}^{p \times d}$, $\mathbf{b}_1 \in \mathbb{R}^p$ form the first affine map to a p -dimensional hidden layer, and $\mathbf{W}_2 \in \mathbb{R}^{m \times p}$, $\mathbf{b}_2 \in \mathbb{R}^m$ project the hidden activation to the m regression outputs (e.g., damage characteristics). The parameter set $\theta = \{\mathbf{W}_1, \mathbf{b}_1, \mathbf{W}_2, \mathbf{b}_2\}$ is learned by minimizing a supervised loss such as MSE with regularization (weight decay, dropout) and early stopping for generalization.

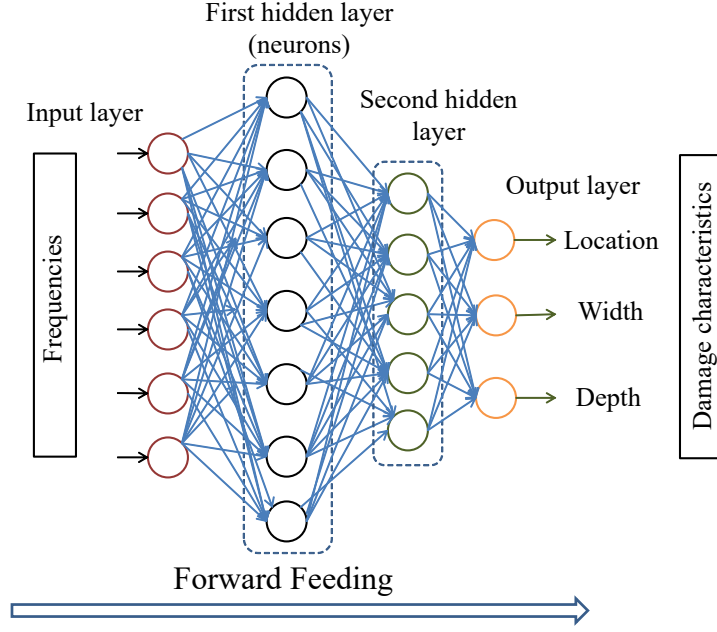


Figure 3.7: FNN structure for prediction of damage characteristics

(b) MLP

A MLP extends the FNN to multiple hidden layers, enabling deeper feature hierarchies.

Starting with $\mathbf{h}^{(0)} = \mathbf{x}$,

$$\mathbf{h}^{(l)} = \sigma(\mathbf{W}^{(l)}\mathbf{h}^{(l-1)} + \mathbf{b}^{(l)}), \quad l = 1, \dots, L-1, \quad \hat{\mathbf{y}} = \mathbf{W}^{(L)}\mathbf{h}^{(L-1)} + \mathbf{b}^{(L)}. \quad (3.25)$$

Depth allows the network to capture complex nonlinearity, while batch normalization and dropout (if used) stabilize and regularize learning. All weights and biases $\{\mathbf{W}^{(l)}, \mathbf{b}^{(l)}\}$ are estimated end-to-end by minimizing the chosen regression loss on the training data (commonly with Adam).

(c) RNN

For sequential inputs $\{\mathbf{x}_t\}_{t=1}^T$, a RNN maintains a hidden state that summarizes past information as illustrated in Fig. 3.8:

$$\mathbf{h}_t = \tanh(\mathbf{W}_x\mathbf{x}_t + \mathbf{W}_h\mathbf{h}_{t-1} + \mathbf{b}), \quad \hat{\mathbf{y}} = \mathbf{W}_y\mathbf{h}_T + \mathbf{b}_y. \quad (3.26)$$

The matrices \mathbf{W}_x , \mathbf{W}_h govern, respectively, how current inputs and prior memory influence the latent state \mathbf{h}_t ; \mathbf{W}_y maps the final summary to the regression targets. Training uses back-propagation through time (truncated to window length T), often with gradient clipping for

stability.

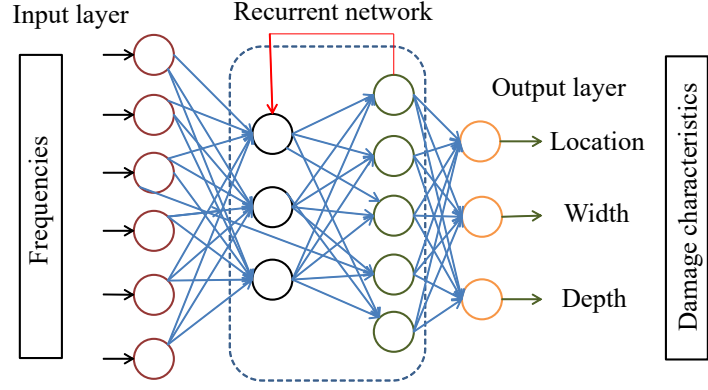


Figure 3.8: RNN structure for damage detection using frequencies

(d) LSTM

LSTM augments the basic recurrence with gated memory to capture long-range temporal effects and mitigate vanishing gradients (see Fig. 3.9). At time t , with $[\cdot, \cdot]$ denoting concatenation and \odot element-wise multiplication,

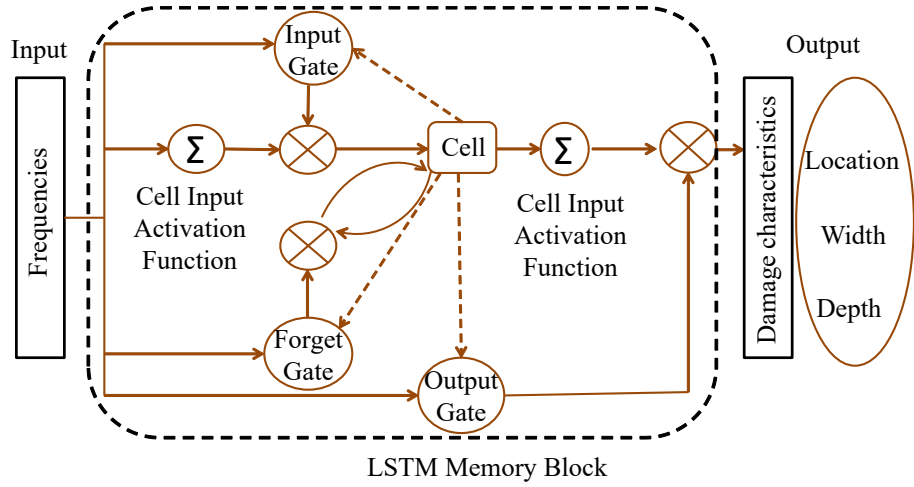


Figure 3.9: Damage characteristics detection using LSTM

$$\begin{aligned}
 \mathbf{f}_t &= \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_f) \quad (\text{forget gate}), \\
 \mathbf{i}_t &= \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_i) \quad (\text{input gate}), \\
 \tilde{\mathbf{C}}_t &= \tanh(\mathbf{W}_C[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_C) \quad (\text{candidate memory}), \\
 \mathbf{C}_t &= \mathbf{f}_t \odot \mathbf{C}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{C}}_t \quad (\text{cell update}), \\
 \mathbf{o}_t &= \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}, \mathbf{x}_t] + \mathbf{b}_o) \quad (\text{output gate}), \\
 \mathbf{h}_t &= \mathbf{o}_t \odot \tanh(\mathbf{C}_t) \quad (\text{hidden state}).
 \end{aligned} \tag{3.27}$$

The gates learn *when* to remember, overwrite, or expose information, which is crucial when damage signatures evolve over time. Parameters in all gate and projection matrices are learned with the same supervised objective as above.

Pseudocode: LSTM Cell Forward Pass

Given input sequence $x = \{x_1, \dots, x_T\}$ and initial states $h_0 = 0, C_0 = 0$:
for $t = 1$ **to** T : concatenate $[h_{t-1}, x_t]$; compute $f_t, i_t, \tilde{C}_t, C_t, o_t$ as above; set $h_t = o_t \odot \tanh(C_t)$.
return h_T (or a pooled sequence) as the sequence representation.

(e) 1D CNN-GRU-ResNet

The hybrid model combines complementary mechanisms. A 1D CNN first applies learned temporal kernels to extract translation-invariant local motifs from the signal; a GRU then models longer-range dynamics through gated recurrence; residual (skip) connections stabilize deeper stacks by adding block inputs to outputs and preserving low-level details useful for precise regression (Fig. 3.10). The final fully connected head maps the aggregated representation to the multi-dimensional target $\hat{\mathbf{y}}$. Formally, letting ϕ_{CNN} denote the convolutional encoder and ψ_{GRU} the recurrent block,

$$\mathbf{z} = \phi_{\text{CNN}}(\mathbf{x}), \quad \mathbf{h} = \psi_{\text{GRU}}(\mathbf{z}), \quad \mathbf{h}_{\text{res}} = \mathbf{h} + \mathcal{P}(\mathbf{z}), \quad \hat{\mathbf{y}} = \mathbf{W}\mathbf{h}_{\text{res}} + \mathbf{b}, \quad (3.28)$$

where $\mathcal{P}(\cdot)$ is an identity or 1×1 projection to match dimensions before forming the residual sum. All convolutional filters, GRU gates, residual projections, and output weights are learned jointly by minimizing a multi-output regression loss (e.g., a weighted sum of per-target MSEs) with standard regularization.

Pseudocode: 1D CNN-GRU-ResNet

Input: time series x .
 1) Feature extraction: $z \leftarrow \text{Conv1D}(x)$.
 2) Temporal modeling: $h \leftarrow \text{GRU}(z)$.
 3) Residual fusion: $h_{\text{res}} \leftarrow h + \mathcal{P}(z)$.
 4) Output head: $\hat{y} \leftarrow \text{Dense}(h_{\text{res}})$.
Return: \hat{y} .

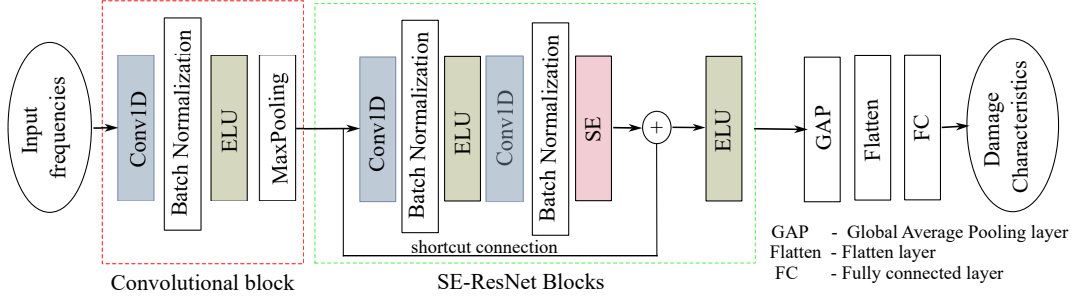


Figure 3.10: Damage characteristics detection using 1D CNN-GRU-ResNet

Estimation and interpretation (common to all models). Given training pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^N$ or sequences $\{(\mathbf{X}_i, \mathbf{y}_i)\}_{i=1}^N$, parameters θ are obtained by minimizing $\mathcal{L}(\theta) = \frac{1}{N} \sum_{i=1}^N \ell(\hat{\mathbf{y}}_i(\theta), \mathbf{y}_i)$, with ℓ chosen as MSE for regression (and combined or weighted if outputs differ in scale/importance). Inputs are standardized; predictions are de-standardized to physical units for reporting. Sensitivity tools can be used to interpret which inputs most strongly influence each output.

3.3.3 Loss functions and optimization

Building on the architectures described above, model parameters θ are estimated by minimizing a supervised loss that measures the discrepancy between predictions and ground truth. Because the targets in this work are continuous (e.g., damage location, depth, and width), regression losses are adopted and, for multi-output settings, combined as a weighted sum across outputs to balance scale and importance.

(a) Loss functions for regression

Two complementary criteria are used. The mean squared error (MSE) penalizes larger deviations more strongly and is sensitive to outliers,

$$\mathcal{L}_{\text{MSE}} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2, \quad (3.29)$$

while the mean absolute error (MAE) provides a robust alternative based on absolute deviations,

$$\mathcal{L}_{\text{MAE}} = \frac{1}{N} \sum_{i=1}^N |y_i - \hat{y}_i|. \quad (3.30)$$

For multi-output regression, the overall objective can be written as $\mathcal{L} = \sum_k w_k \mathcal{L}^{(k)}$, where w_k weights each target to reflect units or task priority.

(b) Optimization algorithms

Parameters are updated iteratively using first-order optimizers. Let $g_t = \nabla_{\theta} \mathcal{L}(\theta_t)$ denote the gradient at iteration t . Adam employs momentum and per-parameter adaptive scaling via first and second moment estimates,

$$\theta_{t+1} \leftarrow \theta_t - \eta \cdot \frac{\hat{m}_t}{\sqrt{\hat{v}_t + \varepsilon}}, \quad (3.31)$$

where \hat{m}_t and \hat{v}_t are bias-corrected running averages of g_t and g_t^2 , η is the learning rate, and ε stabilizes division. root mean square propagation (RMSProp) normalizes updates by an exponential moving average of squared gradients,

$$\theta_{t+1} \leftarrow \theta_t - \frac{\eta}{\sqrt{E[g^2]_t + \varepsilon}} \cdot g_t, \quad (3.32)$$

which is effective for nonstationary objectives and mini-batch noise. In addition to these conventional optimizers for weight updates, a nature-inspired electric eel foraging optimization (EEFO) strategy is used as an *outer-loop* hyperparameter search to select learning rate, batch size, layer widths, and regularization settings for the DL models. This separation—gradient-based inner optimization for θ and metaheuristic outer optimization for hyperparameters—provides stable training across architectures and datasets discussed in the preceding sections.

3.3.4 Evaluation and inference

After model training, performance is assessed using separate validation or test datasets—typically through holdout validation or k-fold cross-validation. Regression-based metrics such as MSE, MAE, and others are used to quantify the prediction accuracy on unseen data.

Once the best-performing model is selected, it is deployed in an inference setting to predict structural indicators such as damage severity, location, or stiffness degradation on new vibration signals collected from real-world or simulated structures.

While DL models often yield superior accuracy and generalization compared to traditional ML approaches, they are computationally more expensive, data-hungry, and relatively opaque in terms of interpretability. Hence, their deployment in SHM applications

must be balanced against task complexity, model transparency, and the availability of training data.

3.4 Summary

This chapter set the stage for using AI algorithms in structural damage assessment. It first outlined how traditional ML relies on hand-crafted features, then showed how DL learns useful representations directly from data. The chapter described the model families employed, clarified what their governing equations imply in practice, and summarized how predictions are evaluated for both regression and classification. It also explained the loss functions and optimizers that drive training, making clear not just what the models are, but how they learn. With this foundation in place, the next chapter moves from concepts to practice, presenting the proposed methodology, data construction, and model implementation details.

Proposed Methodology for Damage Assessment

4.1 Introduction

This chapter presents the methodology developed in this thesis to detect, localize, quantify and classify structural damage in both bridge beams and road networks using AI-based vibration monitoring. The novelty of this approach lies in two key advancements —(i) simultaneous localization and quantification of damage, (ii) binary and multi-class classification defects for maintenance prioritization. Despite the differences in data acquisition and characteristics between bridge and road scenarios, a generalized AI-based framework has been established. The methodology is scalable, adaptive, and robust, ensuring applicability to both laboratory and field settings.

This chapter develops the proposed methodology for AI-based structural damage assessment in a structured manner. Section 4.1 introduces the motivation and scope of the methodology, followed by an outline of the overall framework in section 4.2. Section 4.3 describes the data acquisition process, which forms the basis for both regression- and classification-based modeling. Section 4.4 presents the regression-based damage assessment approach, including feature extraction (section 4.4.1), AI model development (section 4.4.2), optimization and hyperparameter tuning (section 4.4.3), and inference and decision support (section 4.4.4). Similarly, section 4.5 details the classification-based damage assessment framework through the same sequence of steps: feature extraction, model development, optimization, and inference. Finally, Section 4.6 provides a synthesis of the methodological contributions presented in this chapter and outlines how they form the basis for Chapter 5, where the proposed framework is implemented and evaluated through two case studies.

4.2 Overall framework

The overall flow of the proposed AI-based damage assessment pipeline is structured into five key stages: (i) data acquisition, (ii) feature extraction, (iii) model development, (iv) optimization and hyperparameter tuning, and (v) inference and decision support, as illustrated in Fig. 4.1.

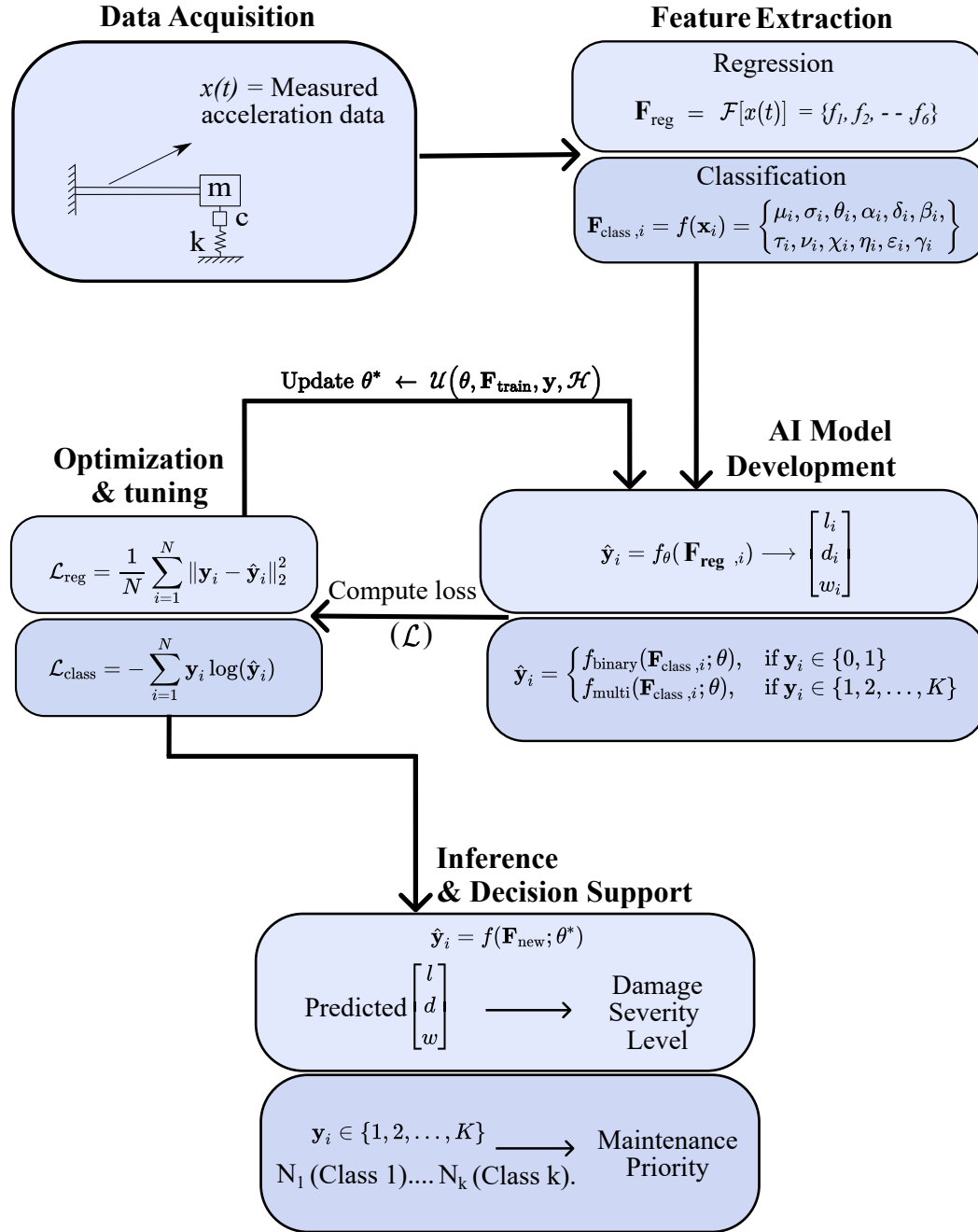


Figure 4.1: Proposed methodology for AI-based structural damage assessment

Vibration signals are collected from bridge beams or road networks using appropriate sensors or vehicle-mounted sensors. Time-domain and frequency-domain features relevant to damage detection are extracted from the acquired signals using statistical and signal processing techniques. AI models are trained to learn relationships between the extracted features and damage characteristics (e.g., damage location, severity, or class). A combination of traditional grid search and advanced optimization strategies—including gradient-based techniques (such as adaptive moment estimation (Adam) and RMSProp) and metaheuristic algorithms like EEFO—is employed to fine-tune model hyperparameters, thereby enhancing generalization capability and robustness across both structural scenarios.

Apply the best-trained models to new, unseen data in real time or in batches. Predicted damage parameters are mapped to agreed severity bands, and each unit of analysis is automatically assigned to a relevant defect class. Class counts and density maps then summarize how widespread each defect is and where hotspots occur. Combining these labels with spatial context and unit-rate cost curves yields prioritized and maintenance decisions.

4.3 Data acquisition

The methodology begins with the acquisition of acceleration signals from structural systems subjected to dynamic excitation. As illustrated in Figure 5.6, two distinct experimental configurations are considered for collecting response data.

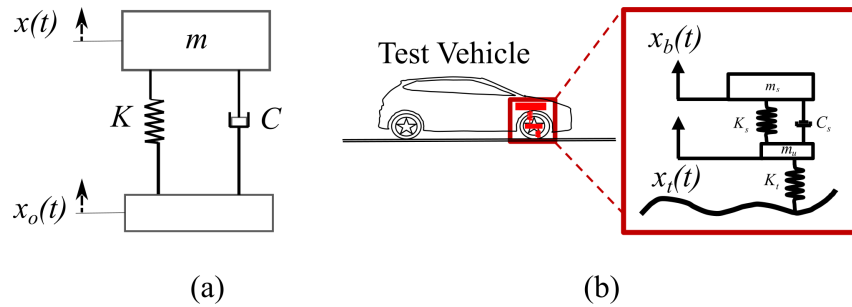


Figure 4.2: Generalized data acquisition from dynamically excited structural systems

In the first configuration [Figure 4.2(a)], a structural specimen is modeled as a single-degree-of-freedom (SDOF) system characterized by mass (m), stiffness (K), and damping coefficient (C). The system is displaced from its equilibrium position and released, initiating

ing a free vibration response. The resulting decaying acceleration signal $x(t)$ is recorded using strategically positioned piezoelectric sensors. These signals embed information about the dynamic properties of the system and are sensitive to damage or stiffness reductions.

In the second configuration [Figure 4.2(b)], acceleration data is captured from a system experiencing continuous external excitations, modeled using a quarter-car analogy. Here, the vertical displacement response $x_b(t)$ of the system is influenced by the input excitation profile $x_t(t)$, and modulated by suspension stiffness K and damping C . The recorded vibration responses reflect surface irregularities or dynamic perturbations encountered during motion. In both setups, the measured acceleration signals $x(t)$ encapsulate the dynamic response characteristics of the system, which serve as a foundation for downstream feature extraction and model development aimed at identifying, localizing, quantifying and classifying structural anomalies.

4.4 Regression-based damage assessment

4.4.1 Feature extraction

The collected time-domain signals are converted into the frequency domain using the Fast Fourier Transform (FFT) (see Fig. 4.3):

$$\mathbf{F}_{\text{reg}} = \mathcal{F}[x(t)] \quad (4.1)$$

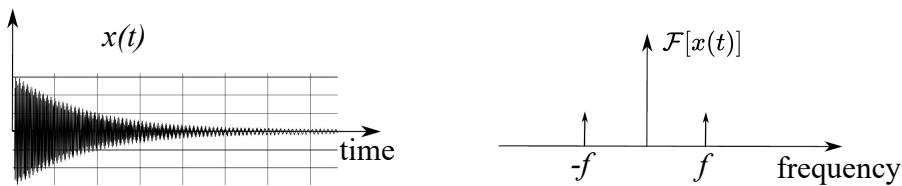


Figure 4.3: Feature extraction for regression task

The feature vector \mathbf{F}_{reg} consists of dominant modal frequencies and amplitude characteristics that reflect the dynamic properties of the system under damage. These features are compact, robust, and sensitive to variations in stiffness due to cracks or other defects. To derive these features, consider the discrete vibration signal $a[n]$, sampled at a rate f_s with a total of N samples. Its discrete Fourier transform is given by

$$A[k] = \sum_{n=0}^{N-1} a[n] \cdot e^{-j\frac{2\pi}{N}kn}, \quad (4.2)$$

where $A[k]$ represents the complex amplitude of the k -th frequency component [140]. The corresponding frequency resolution is expressed as

$$f_k = \frac{k \cdot f_s}{N}. \quad (4.3)$$

To obtain physically interpretable quantities, the magnitude spectrum is calculated as

$$|A[k]| = \sqrt{\text{Re}(A[k])^2 + \text{Im}(A[k])^2}. \quad (4.4)$$

The dominant peaks identified in this spectrum correspond to the first six modal frequencies of the structure. These modal frequencies serve as critical features for damage detection, as they are highly sensitive to stiffness reductions caused by cracks or other structural defects.

4.4.2 AI model development

The extracted features \mathbf{F}_{reg} serve as inputs for a supervised multi-output regression model is trained to simultaneously predict three critical damage parameters:

$$\hat{\mathbf{y}}_i = f_{\theta}(\mathbf{F}_{\text{reg}}) = \begin{bmatrix} l_d \\ d_d \\ w_d \end{bmatrix} \quad (4.5)$$

Here, l_d , d_d , and w_d represent damage location, depth, and width, respectively. $f_{\theta}(\cdot)$ denotes the trained regression function parameterized by θ , which represents the set of learnable model parameters (e.g., weights and biases in neural networks, split criteria in tree models). These parameters are optimized during the training phase to minimize the prediction error, as discussed in Chapter 3. The mathematical formulation aligns with the generalized supervised learning structure presented earlier, where the model learns a functional mapping from feature space to target space by minimizing a defined loss function over the training dataset.

4.4.3 Optimization and hyperparameter tuning

Model parameters θ are optimized by minimizing the discrepancy between the predicted outputs $\hat{\mathbf{y}}_i$ and the ground truth values \mathbf{y}_i using the MSE loss:

$$\mathcal{L}_{\text{reg}} = \frac{1}{N} \sum_{i=1}^N \|\mathbf{y}_i - \hat{\mathbf{y}}_i\|_2^2 \quad (4.6)$$

The generalized learning process can be expressed as:

$$\theta^* \leftarrow \mathcal{U}(\theta, \mathbf{F}_{\text{reg,train}}, \mathcal{H}) \quad (4.7)$$

where \mathcal{H} is the set of hyperparameters (e.g., learning rate, regularization strength, number of estimators, tree depth), and $\mathcal{U}(\cdot)$ denotes the update operator, determined by the algorithmic choice. For gradient-based models:

$$\theta^* = \theta - \eta \nabla_{\theta} \mathcal{L}(\theta; \mathbf{F}_{\text{reg,train}}) \quad (4.8)$$

For closed-form models (e.g., OLS regression):

$$\theta^* = (\mathbf{F}_{\text{reg,train}}^{\top} \mathbf{F}_{\text{reg,train}})^{-1} \mathbf{F}_{\text{reg,train}}^{\top} \mathbf{y} \quad (4.9)$$

For tree-based models (e.g., RF, GBM):

$$\theta^* = \text{GreedySplit}(\mathbf{F}_{\text{reg,train}}, \mathcal{H}) \quad (4.10)$$

Hyperparameters \mathcal{H} are tuned externally, while parameters θ are learned from data. These processes are often iterative: \mathcal{H} defines the learning configuration, and $\mathcal{U}(\cdot)$ optimizes θ within that configuration.

4.4.4 Inference and decision support

Once trained, the regression model infers damage parameters from new input data. These predictions are mapped to severity levels and visualized spatially for maintenance prioritization. Quantified outputs help decision-makers schedule targeted inspections, rehabilitation, and risk mitigation.

4.5 Classification-based damage assessment

4.5.1 Feature extraction

The collected acceleration data is segmented using a sliding window technique with 50% overlap to preserve temporal continuity. From each windowed segment \mathbf{x}_i , a corresponding statistical feature vector is extracted, capturing the local characteristics of the data:

$$\mathbf{F}_{\text{class},i} = f(\mathbf{x}_i) = \{\mu_i, \sigma_i, \theta_i, \alpha_i, \delta_i, \beta_i, \tau_i, \nu_i, \chi_i, \eta_i, \varepsilon_i, \gamma_i\} \quad (4.11)$$

Here, μ and σ denote the mean and standard deviation, while $\beta, \eta, \varepsilon, \gamma$ represent capability potential and higher-level derived metrics used for decision-making in surface condition classification.

4.5.2 AI model development

A supervised classification model is developed using the statistical features extracted from the segmented signals, as described in the previous section. Each feature vector $\mathbf{F}_{\text{class},i}$ represents localized statistical and non-statistical characteristics of the raw signal, and serves as the input for learning-based classification.

Depending on the nature of the application, the classification task is framed either as a binary or a multi-class problem:

$$\mathbf{y}_i = \begin{cases} f_{\text{binary}}(\mathbf{F}_{\text{class},i}; \boldsymbol{\theta}), & \mathbf{y}_i \in \{0, 1\} \\ f_{\text{multi}}(\mathbf{F}_{\text{class},i}; \boldsymbol{\theta}), & \mathbf{y}_i \in \{1, 2, \dots, K\} \end{cases} \quad (4.12)$$

Here, f_{binary} and f_{multi} denote the classification functions parameterized by $\boldsymbol{\theta}$, which encapsulates the learnable parameters of the model (e.g., weights in neural networks, support vectors in SVM, or node splits in decision trees). These parameters are optimized during training using labeled data to minimize a classification loss function and improve predictive accuracy on unseen instances.

The binary classification task is designed to flag whether a given segment exhibits structural irregularity or not ($\mathbf{y}_i \in \{0, 1\}$), while the multi-class formulation further categorizes the type or severity of the identified anomaly into one of K discrete classes

($\mathbf{y}_i \in \{1, 2, \dots, K\}$). This flexibility allows the framework to adapt to varying diagnostic requirements—from simple defect detection to detailed condition classification.

4.5.3 Optimization and hyperparameter tuning

The classification models introduced in the previous section are trained by minimizing the discrepancy between the predicted class probabilities $\hat{\mathbf{y}}_i$ and the ground truth labels \mathbf{y}_i . This is achieved using the categorical cross-entropy loss function:

$$\mathcal{L}_{\text{class}} = - \sum_{i=1}^N \mathbf{y}_i \log(\hat{\mathbf{y}}_i) \quad (4.13)$$

The learnable parameters θ , which define the internal structure of the classification model (e.g., weights in a neural network, kernel coefficients in support vector classifiers), The general learning process can be expressed as:

$$\theta^* \leftarrow \mathcal{U}(\theta, \mathbf{F}_{\text{class,train}}, \mathcal{H}) \quad (4.14)$$

where θ denotes the trainable parameters (e.g., weights in a neural network, coefficients in linear models, support vectors in SVMs, or split rules in tree models), $\mathbf{F}_{\text{class,train}}$ is the training feature matrix, \mathcal{H} represents the set of hyperparameters, and $\mathcal{U}(\cdot)$ is the update operator defined by the learning algorithm.

For gradient-based classifiers such as logistic regression, neural networks, and SVMs in primal form:

$$\theta^* = \theta - \eta \nabla_{\theta} \mathcal{L}(\theta; \mathbf{F}_{\text{class,train}}) \quad (4.15)$$

For closed-form classifiers, parameters are computed analytically without iterative updates. For tree-based models such as RF or GBM:

$$\theta^* = \text{GreedySplit}(\mathbf{F}_{\text{class,train}}, \mathcal{H}) \quad (4.16)$$

To further improve model generalization and robustness, a hybrid hyperparameter tuning strategy is employed—similar to the one used for regression models. This involves optimizing external configuration parameters such as learning rate, number of hidden layers, activation functions, and regularization strength etc. This dual-stage optimization—internal parameter learning and external configuration tuning via hybrid

search—ensures the classification models are both well-calibrated and capable of generalizing to unseen data.

4.5.4 Inference and decision support

The final model is used to classify each incoming data segment, and the predicted output class \hat{y}_i is aggregated over spatial zones to calculate the defect distribution $\{N_k\}_{k=1}^K$. This aggregated information is then passed to a maintenance decision engine, which ranks the road segments based on the severity and frequency of detected defects, generates repair plans by integrating spatial mapping of damage locations, and allocates resources efficiently according to the defect category.

4.6 Summary

This chapter established a generalized AI-based methodology for structural damage assessment that integrates vibration data acquisition, feature extraction, supervised learning models, hyperparameter optimization, and decision support mechanisms. Two complementary pathways were developed: a regression-based framework for simultaneous localization and quantification of damage, and a classification-based framework for binary and multi-class defect categorization. Together, these approaches provide a scalable and adaptive pipeline capable of addressing both laboratory-controlled bridge beam testing and field-scale road monitoring. The proposed methodology not only formalizes the technical foundation for damage detection but also links model outputs directly to maintenance decision-making. Building upon this framework, the next chapter applies the methodology to two case studies—bridge components and road networks—to evaluate its effectiveness and practical utility in real-world scenarios.

Bridge Beam Damage Assessment

5.1 Introduction

This chapter presents the practical application of the AI-based structural damage assessment regression methodology, introduced in Chapter 4, to laboratory-scale bridge beam specimens. Bridge beams are critical load-bearing elements within bridge superstructures, responsible for transferring vehicular and environmental loads across spans to the substructure components such as piers and abutments. Owing to their continuous exposure to dynamic traffic loads, environmental effects, and temperature variations, these elements are especially prone to localized damage, particularly near support zones and mid-span regions where stress concentrations are typically highest. Such progressive deterioration can severely impair the structural integrity and serviceability of the bridge system, making timely detection and assessment of damage in beams essential for ensuring long-term performance and safety. To address this challenge, the study first employed machine learning techniques to perform multi-output regression, enabling the simultaneous prediction of two fundamental damage parameters—location and depth—from frequency-domain features derived from the vibration response signals. Building upon this foundation, a deep learning framework was subsequently developed to enhance the predictive capability by incorporating an additional output parameter, namely damage width, thereby facilitating a more comprehensive and precise assessment of beam damage.

The following sections demonstrate the application of the proposed methodology to bridge beam damage assessment, highlighting both experimental and numerical perspectives. Section 5.1 introduces the objectives and scope of the case study. section 5.2 details the experimental setup and data collection procedures, supported by numerical simulation for modal parameter estimation and validation (section 5.2.1), followed by the construction of a hybrid dataset integrating experimental and simulated data (section 5.2.2). Section 5.3 focuses on the implementation of ML models, where the problem is formulated as a multi-input multi-output (MIMO) regression task (section 5.3.1). The subsequent subsections address model development and evaluation (section 5.3.2), hyperparameter tuning (sec-

tion 5.3.3), and inference and testing (section 5.3.3). Section 5.4 extends the analysis to deep learning models, employing an EEFO-based optimization strategy. This includes the formulation of extended output parameters (section 5.4.1), development of model architectures (section 5.4.2), and performance evaluation (section 5.4.3). Section 5.5 provides comparative summary interpretations between ML and DL approaches, highlighting their relative strengths and limitations. Finally, section 5.6 concludes the chapter by summarizing key findings and preparing the ground for the subsequent case study on road damage assessment.

5.2 Experimental setup and data collection

In this experimental investigation, steel cantilever beams were employed to simulate real-world degradation conditions. A total of nine beam specimens were prepared and tested. Each beam was mounted in a cantilever configuration, where one end was rigidly fixed using a bolted connection. The fixture was designed to prevent translation and rotation in all directions, thus replicating ideal fixed-end boundary conditions, while the other end was left free. The beam had a fixed length of 900 mm, a width of 36 mm, and a depth of 12 mm

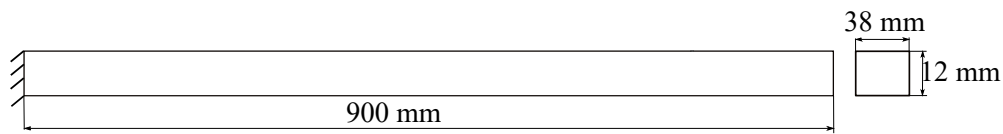


Figure 5.1: Cantilever beam

as illustrated in Fig. 5.1. The material properties of the steel used are presented in Table 5.1.

The damage locations and characteristics were carefully controlled and recorded.

Table 5.1: Material Properties of Steel

Young's Modulus (E)	210 GPa
Poisson's Ratio (ν)	0.27
Density (ρ)	7760 kg/m ³
Yield Strength (σ_{yield})	250 MPa
Ultimate Tensile Strength (σ_{ultimate})	380 MPa

Artificial damage was introduced in cantilever beam specimens using an angle grinder equipped with a 4-inch metal cutting blade to simulate notch-type damage at controlled locations. Across nine specimens, multiple damage scenarios were systematically created

by varying both the location and depth of the notches. For instance, the first beam was damaged at 90 mm from the fixed end with incremental depths ranging from 1.2 mm to 4.8 mm. The second beam featured damage at 180 mm, and the procedure was repeated along the span of each beam to simulate a wide range of structural damage conditions. The damage creation process is shown in Fig. 5.2.

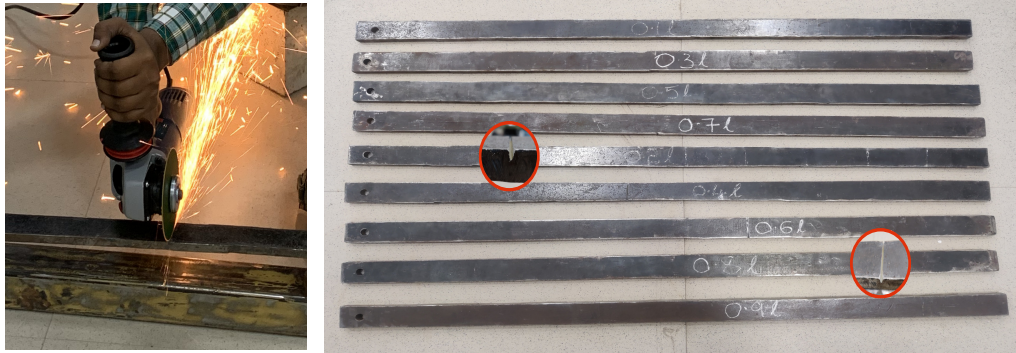


Figure 5.2: Inducing damage on the beam using an angle grinder with 4-inch metal cutting blade

Although the experimental procedure remained uniform across all tests, the target outputs derived from the collected vibration data were model-specific. For the ML regression models, only two damage descriptors—location and depth—were considered as output variables. However, in the case of DL models, the output space was extended to include an additional third parameter, damage width as illustrated in Fig. 5.3. The width of each notch was carefully controlled and measured using a digital Vernier caliper. Four discrete damage width levels were introduced across specimens, specifically: 0.2 mm, 0.4 mm, 0.6 mm, and 0.8 mm. These additional width measurements were incorporated exclusively for training and evaluating the DL models, enabling more comprehensive modeling of structural damage severity. This variation in output dimensionality allowed for a comparative study of classical and deep learning-based regression techniques, while also demonstrating the extended capabilities of DL architectures to learn complex damage characteristics from vibrational data.

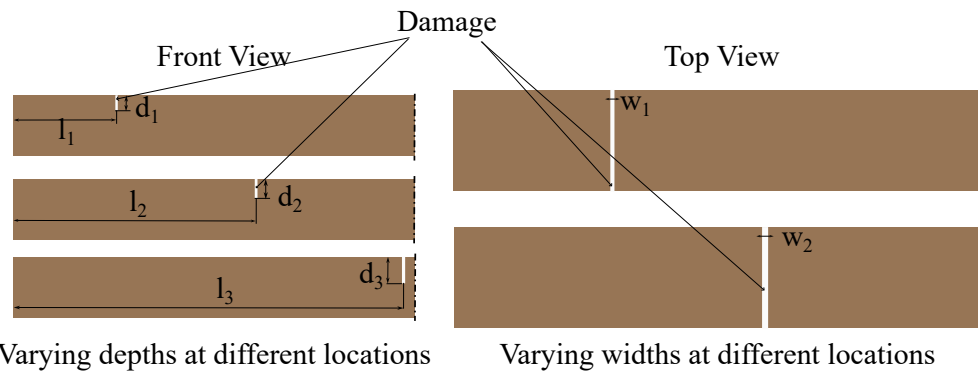


Figure 5.3: Damaged portions at different locations with varying widths and depths

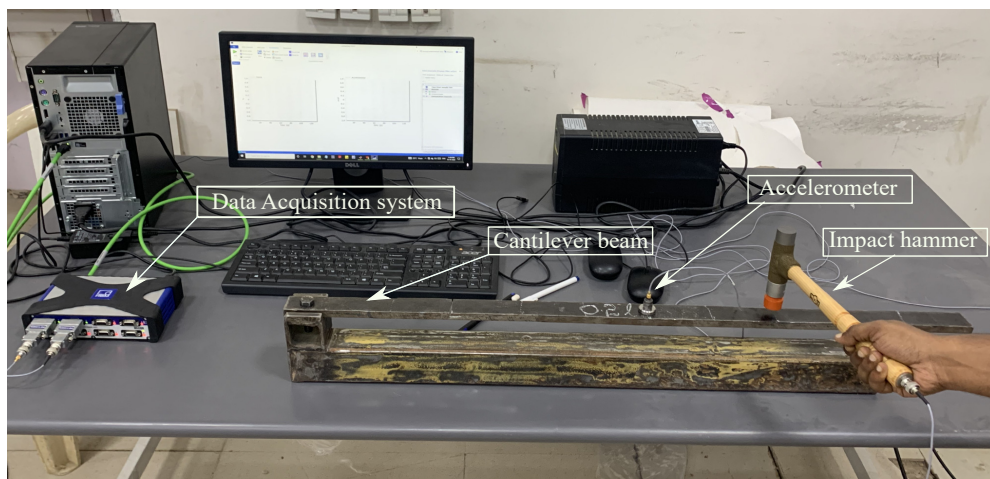


Figure 5.4: Experimental setup performing modal analysis on the damaged cantilever beam using accelerometer and a data acquisition system.

To excite the beam and measure its dynamic response, an instrumented impact hammer (Type 8207) with a soft tip was used. An accelerometer (Type 4534-B-002) was mounted near the damage to record the vibration signals resulting from each excitation. The data acquisition was performed using the QuantumX MX840B system, which supports eight universal input channels and multiple transducer technologies. Signal recording and visualization were carried out using the *CatmanEasy* software with a sampling frequency of 1200 Hz. The experimental setup is shown in Fig. 5.4. After each damage case was created, the vibration data were collected by striking the beam with the impact hammer and recording the resulting acceleration signals. These time-domain signals were subsequently transformed into the frequency domain to extract the first six natural frequencies of each beam condition.

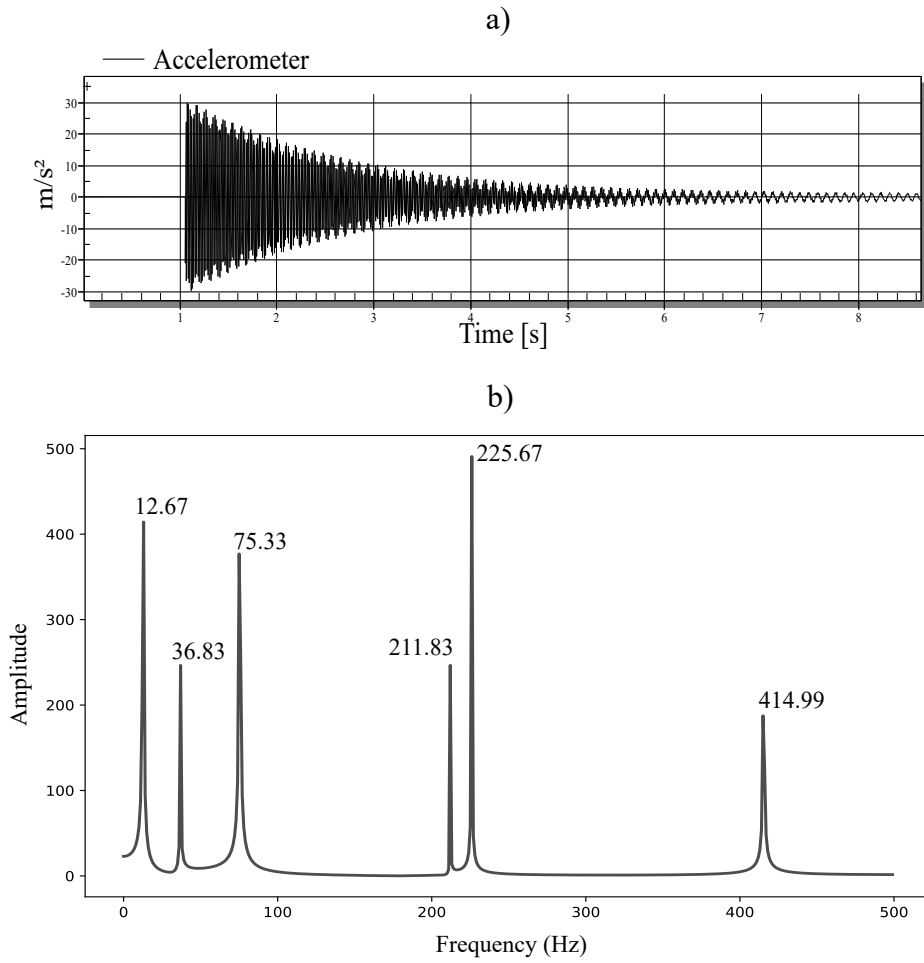


Figure 5.5: Vibration data of a damaged beam a) time domain b) frequency domain.

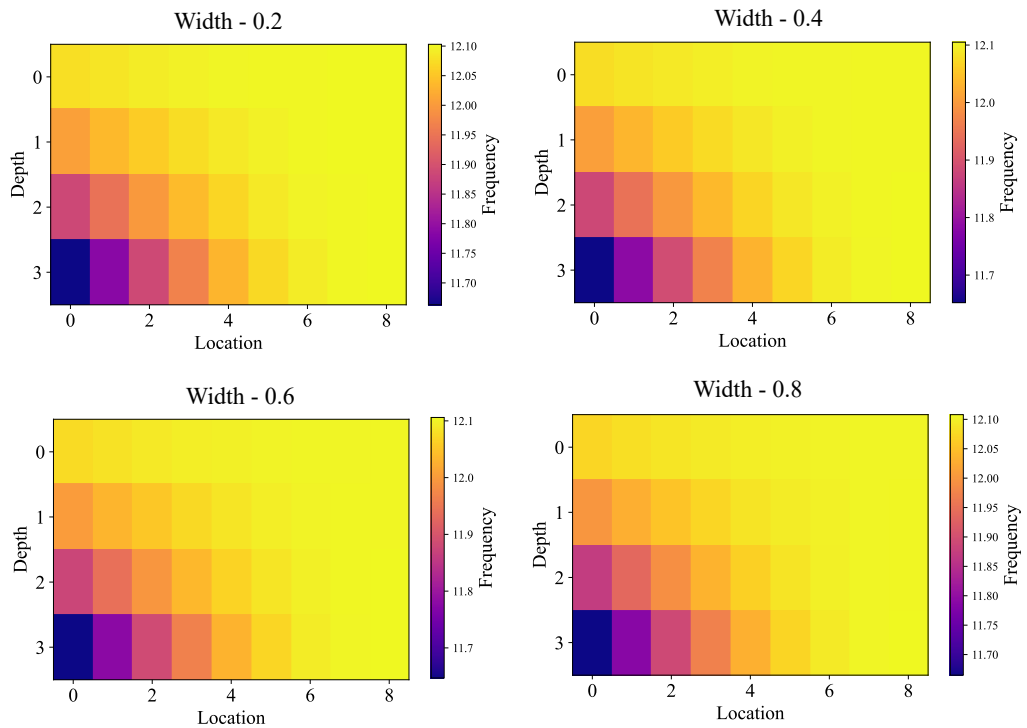


Figure 5.6: Change in frequencies with varying widths of damage.

A representative signal in both time and frequency domains is illustrated in Fig. 5.5. The variation in natural frequencies with respect to damage location, width, and depth is illustrated through heat maps (see Fig. 5.6). A clear horizontal gradient indicates that frequency values increase as the damage shifts from the fixed end toward the free end. Additionally, even slight changes in width and depth produce noticeable shifts in the color intensity, highlighting the sensitivity of frequency response to these damage parameters.

5.2.1 Numerical simulation of modal parameters and validation

To complement the experimental analysis and generate additional data for machine learning, a finite element (FE) model of the cantilever beam was developed using ANSYS. The purpose of this simulation was twofold: (i) to validate the experimentally obtained modal frequencies, and (ii) to produce synthetic data across a wider range of damage scenarios for robust model training. The developed finite element model of the beam is shown in Fig. 5.7. The beam geometry was recreated to match the experimental specimen, with dimensions of 900 mm length, 36 mm width, and 12 mm depth. One end of the beam was modeled as fully fixed, while the other was left free, simulating ideal cantilever boundary conditions. Damage was introduced in the model as a notch-like crack, and finer meshing was applied locally at the damaged regions to improve accuracy.

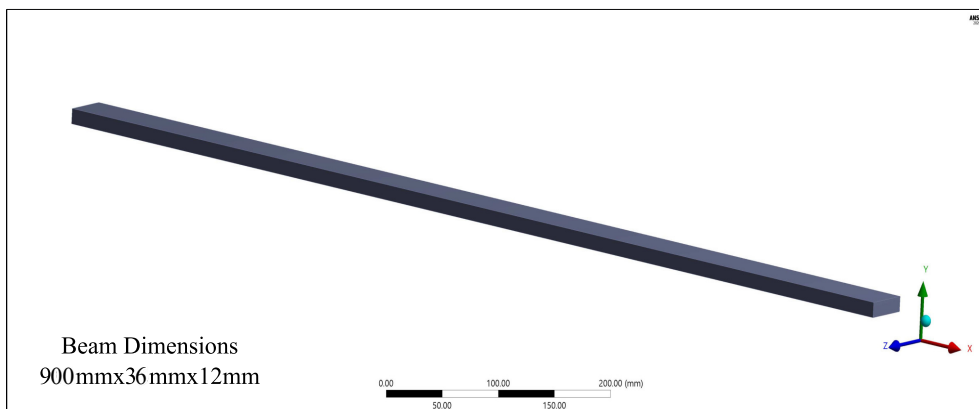


Figure 5.7: Numerically modelled cantilever beam

Table 5.2: Mesh convergence by RMS percentage error

Number of Elements	RMS Error (%)
92	0.290
183	0.199
308	0.349
672	0.099
1150	0.049
2142	0.0249
4392	0.0249
11528	0.0049
52100	0.0005

To ensure the reliability of the simulation results, a mesh convergence study was carried out. RMSE of the first six modal frequencies was used as the convergence criterion. Table 5.2 presents the results of the mesh convergence study. It was found that a mesh consisting of approximately 52,098 elements, with an average element size of 1.4 mm, resulted in an RMS error of only 0.0005%. The refined mesh near the damage region is shown in Fig. 5.8.

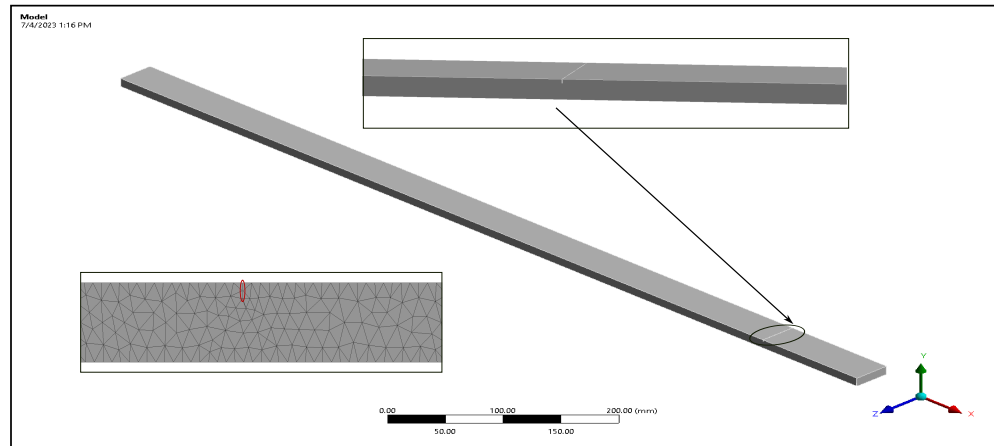


Figure 5.8: Cantilever beam with damage at a distance of 785 mm from fixed end.

Following convergence, modal analysis was performed for various damage scenarios by altering both the location and depth of the crack. A total of 720 simulations were performed, and the frequencies of the first six modes were calculated. Damage was created at intervals of 5 mm from the fixed end, with varying magnitudes of 1.2 mm, 2.4 mm, 3.6 mm, and 4.8 mm, respectively. For each configuration, the first six natural frequencies were extracted to serve as input features for subsequent predictive modeling. In addition to the dataset, an extended set of 145 simulations was carried out for the DL models by incorporating an additional output parameter—damage width. In this case, damage con-

figurations were varied across location, depth, and four distinct width values: 0.2 mm, 0.4 mm, 0.6 mm, and 0.8 mm. The frequencies corresponding to these configurations were used to train and evaluate the DL models for simultaneous prediction of damage location, depth, and width. Fig. 5.9 presents the mode shapes for the second and fifth vibration modes under one such damage condition.

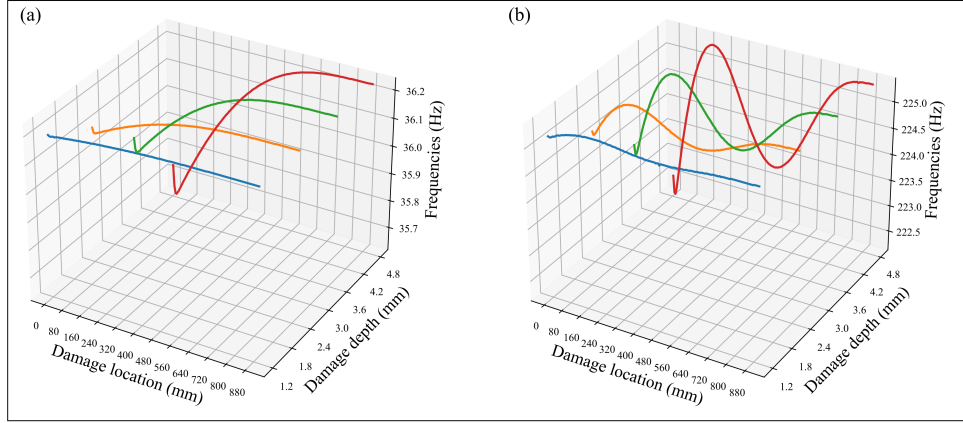


Figure 5.9: Frequency variation with changing damage location and depth, (a) for second mode and (b) fifth modes

Table 5.3: Comparison of experimental and numerical modal frequencies for different damages

Damage (l, w, d)	Numerical Frequencies (Hz)	Experimental Frequencies (Hz)	Difference (%)
(90, 0.2, 1.2)	13.131, 39.361, 82.331, 230.379, 244.971, 450.723	12.077, 36.200, 75.720, 211.880, 225.300, 414.530	8.73
(180, 0.4, 2.4)	13.001, 39.053, 81.869, 228.603, 243.492, 445.980	12.033, 36.145, 75.773, 211.580, 225.360, 412.770	8.05
(270, 0.6, 3.6)	13.066, 39.323, 82.350, 228.427, 245.269, 449.828	11.994, 36.097, 75.595, 209.690, 225.150, 412.930	8.94
(360, 0.8, 4.8)	12.670, 37.994, 76.753, 215.740, 228.219, 422.961	11.972, 36.078, 74.523, 209.660, 223.940, 412.490	5.70
(450, 0.2, 1.2)	13.095, 39.209, 81.913, 229.378, 243.818, 448.116	12.098, 36.223, 75.675, 211.910, 225.250, 413.990	8.25
(540, 0.4, 2.4)	13.164, 39.422, 82.066, 229.978, 244.813, 450.693	12.095, 36.220, 75.401, 211.300, 224.930, 414.090	8.85
(630, 0.6, 3.6)	13.099, 39.225, 81.474, 226.243, 243.340, 446.822	12.097, 36.225, 75.243, 208.940, 224.730, 412.650	8.30
(720, 0.8, 4.8)	12.885, 38.673, 79.264, 220.502, 238.208, 428.780	12.104, 36.238, 75.426, 208.050, 224.960, 403.510	6.10
(810, 0.8, 4.8)	12.850, 38.749, 79.835, 223.025, 239.158, 436.820	12.108, 36.246, 75.758, 211.350, 225.370, 411.360	6.20

Table 5.3 shows a representative comparison of modal frequencies across different damage locations, depths and widths. The numerical results showed strong agreement with experimental data, supporting the validity of the simulation framework and justifying its use in generating extended datasets for model training. The table also displays the percentage error, with the maximum error among all modes being 8.94%, indicating that the numerical

and experimental results are in close agreement and validate the accuracy of the simulation model. The six modes of the beam for one of the case has been shown in Fig. 5.10

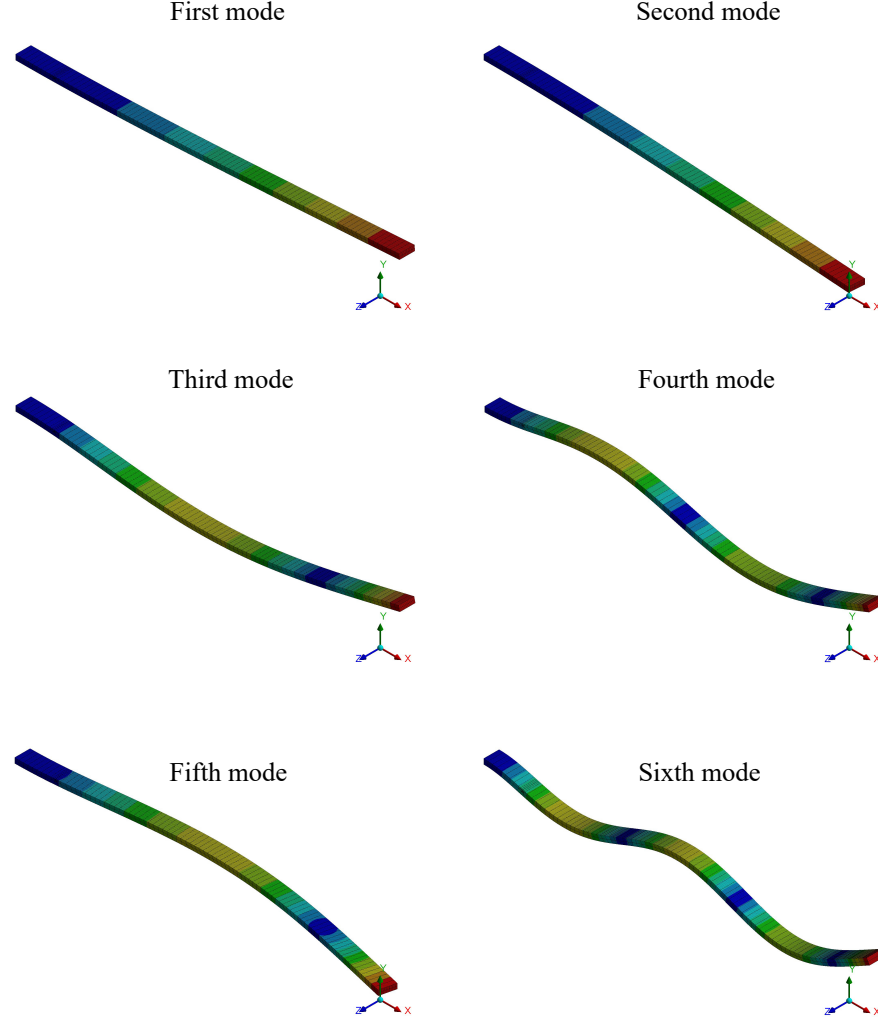


Figure 5.10: Six modes of the beam

5.2.2 Hybrid dataset construction

Integration of experimental and numerical data has been essential to ensure a comprehensive and scalable approach to damage detection in beam-like structures. While experimental methods offer high-fidelity insights into the modal behavior of physically damaged beams, conducting a wide range of tests is often impractical due to time, material, and cost constraints. To address this, validated finite element simulations were used to generate complementary data covering a broader range of damage scenarios. Each data instance in the constructed dataset consists of six input features—the first six natural frequencies

of the beam—mapped to two output parameters: damage location and depth. To further enhance prediction capability, the DL models were extended to include a third output parameter—damage width—alongside location and depth. The combination of physical and simulated data allowed for a robust and diverse training set that improves model generalizability across unseen structural damage configurations. This hybrid dataset was used uniformly in both the ML and DL phases to enhance model robustness and reduce overfitting.

5.3 Implementation using ML: MIMO regression model

The end-to-end workflow built on the hybrid dataset is shown in Fig. 5.11. The process begins with selecting the target beam and estimating its first six natural frequencies from dynamic tests; in parallel, a validated FE model is exercised over a wider set of damage scenarios to generate complementary frequency samples.

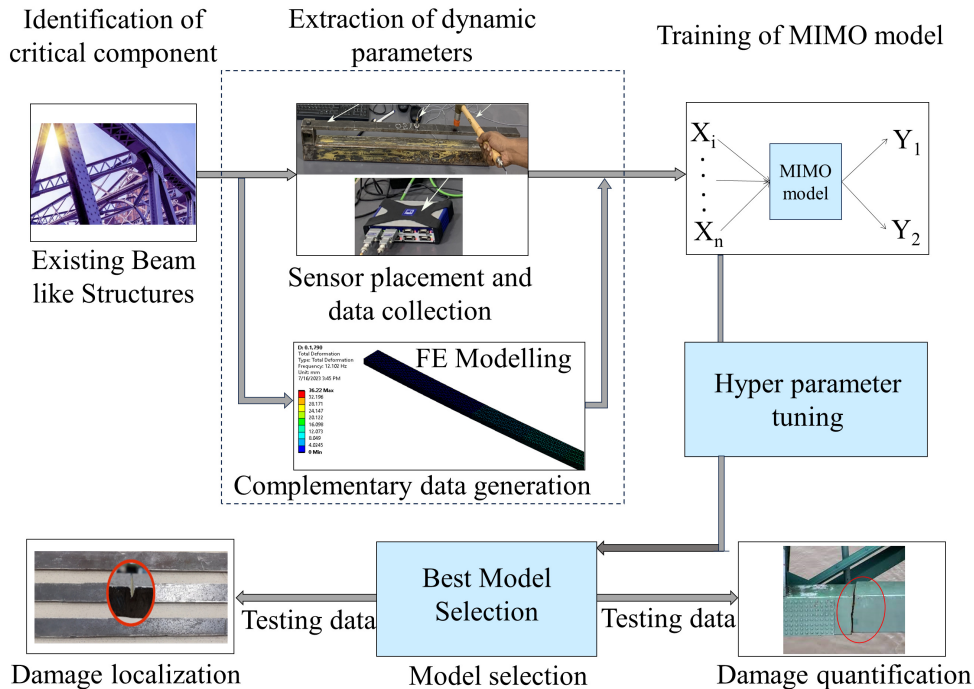


Figure 5.11: Flowchart for the implementation of proposed methodology using ML

These experimental and simulated records are merged into an input matrix with six features and targets describing the damage state. In the machine-learning stage, a MIMO re-

gressor maps the features to two outputs—location and depth—whereas the deep-learning stage reuses the same pipeline but extends the target to three outputs by adding width. Standardization, stratified train/validation/test splits, and cross-validated hyperparameter tuning precede model selection and evaluation on held-out data. The trained predictor is then used for inference to localize and quantify damage; the data pathway and quality controls remain identical for the DL implementation, which differs only in the prediction architecture.

5.3.1 Problem formulation as MIMO regression

The objective of this study is to develop a data-driven model capable of simultaneously predicting multiple damage characteristics—namely, the location and depth of damage in cantilever beam structures—based on their vibrational response. This task is formulated as a supervised learning problem using MIMO regression.

Each data instance in the dataset is represented by an input vector $\mathbf{X} \in \mathbb{R}^6$, consisting of the first six natural frequencies of the beam under a given damage condition:

$$\mathbf{X} = [f_1, f_2, f_3, f_4, f_5, f_6]$$

The corresponding output vector $\mathbf{Y} \in \mathbb{R}^2$ includes the damage parameters to be predicted:

$$\mathbf{Y} = [\text{Location}, \text{Depth}]$$

The learning objective is to construct a function f_θ that maps the input modal features to their corresponding damage parameters:

$$f_\theta : \mathbb{R}^6 \rightarrow \mathbb{R}^2, \quad f_\theta(\mathbf{X}) \approx \mathbf{Y}$$

This formulation supports the development of a single regression model that can predict both outputs simultaneously, leveraging correlations between input frequencies and multiple structural response characteristics. Such a MIMO setup is more efficient and interpretable than training separate single-output models, and allows the model to exploit joint information embedded in the output variables.

5.3.2 Model development and evaluation

The hybrid dataset comprising experimental and numerically simulated modal frequencies (Section 5.2.2) was used to train four machine learning models for MIMO regression: SVR, LR, GBM, and RF. Each model used six input features (first six modal frequencies) to predict two output variables: damage location and damage depth.

Table 5.4: Training parameters of MIMO regression models.

Model	Parameter	Value
SVR	Kernel	RBF
	Regularization (C)	1.0
	Epsilon (ϵ)	0.1
LR	Loss Function	MSE
	Coefficients	See Table 5.5
GBM	Estimators	100
	Learning Rate	1.0
	Maximum Depth	3
	Loss Function	Least Squares
RF	Estimators	40
	Maximum Depth	10
	Min Samples Split	2
	Min Samples Leaf	1

Table 5.5: Learned coefficients of LR model

	w_1	w_2	w_3	w_4	w_5	w_6	Bias (b)
Damage Location	-17868.65	16601.14	3759.54	-21.64	-3172.59	-13.37	55558.18
Damage Depth	106.38	-91.52	-11.17	-0.30	8.96	-0.20	1000.90

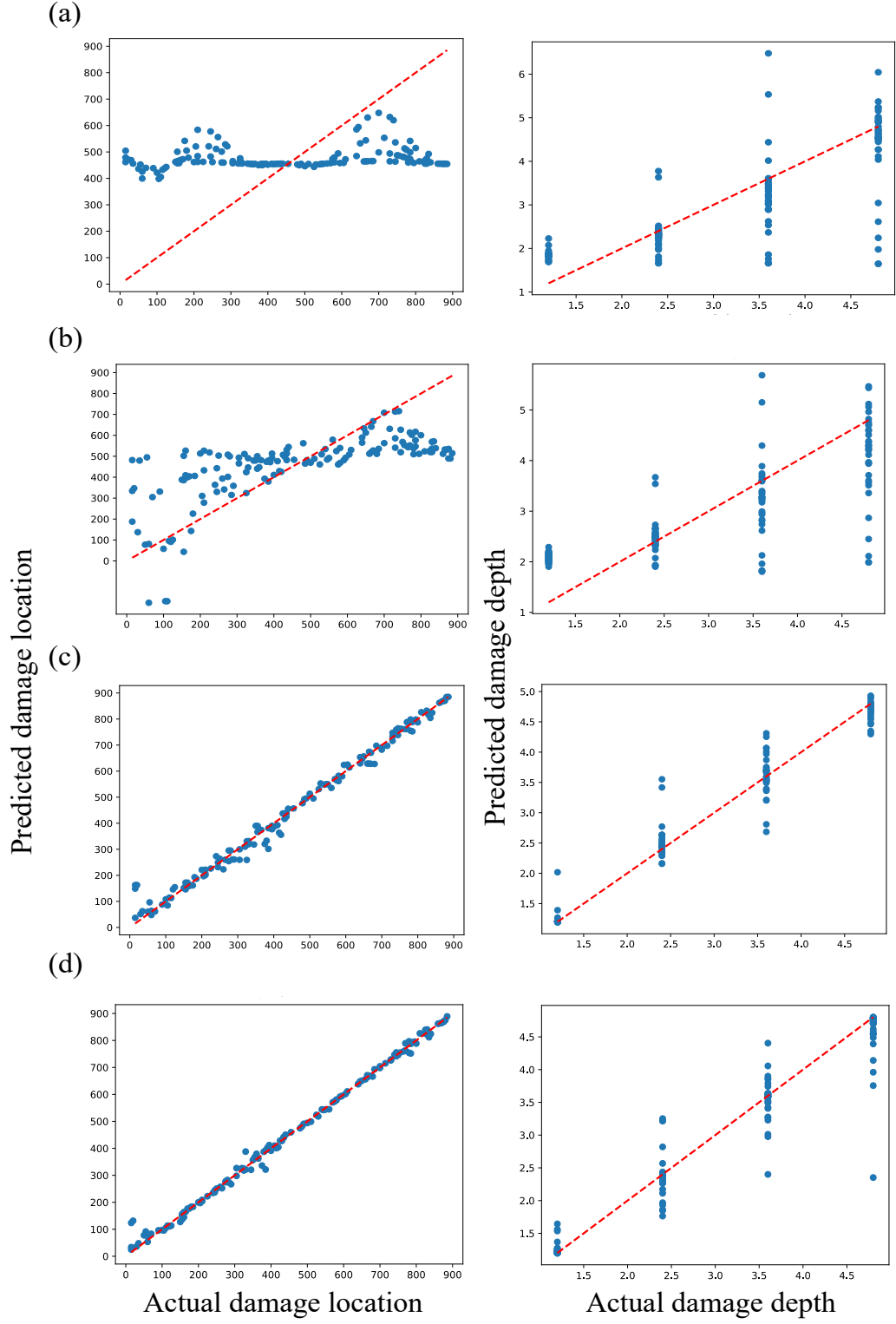


Figure 5.12: Regression performance plots of all MIMO models for damage location and depth, (a) SVR (b) LR (c) GBM (d) RF

The feature importance scores computed from the RF model, as shown in Table 5.6, provide valuable insights into the contribution of each modal frequency towards predicting the output variables. Notably, the first and second mode frequencies exhibit the highest

importance values of 0.3736 and 0.4353, respectively, indicating that these two modes play a dominant role in capturing structural response variations due to damage.

Table 5.6: Feature importance of modal frequencies (RF model).

Modal Frequency	Importance Score
First Mode	0.3736
Second Mode	0.4353
Third Mode	0.0671
Fourth Mode	0.0506
Fifth Mode	0.0612
Sixth Mode	0.0122

On the other hand, the sixth modal frequency shows a minimal contribution (0.0122), suggesting limited influence on the prediction task. This ranking of feature relevance not only validates the physical intuition that lower modes are more sensitive to structural degradation but also supports efficient dimensionality reduction and model simplification in practical applications.

Table 5.7: Various test statistics for the trained models.

Model	Output	MSE	MAE	MedAE	R ² score	EV score	MSLE	MAPE
RF	Damage location	0.33	9.816	5.69	0.998	0.998	0.03	0.139
	Damage depth	0.118	0.164	0.036	0.993	0.991	0.001	0.056
GBM	Damage location	0.84	7.64	11.867	0.987	0.987	0.117	0.248
	Damage depth	0.061	0.136	0.061	0.966	0.967	0.003	0.046
LR	Damage location	36.90	154.55	128.13	0.451	0.452	0.0037	1.1833
	Damage depth	0.766	0.663	0.636	0.585	0.585	0.061	0.319
SVR	Damage location	64.36	218.23	233.55	0.043	0.0478	0.868	1.810
	Damage depth	0.753	0.6004	0.530	0.592	0.593	0.0552	0.2662

All models were trained using 576 samples and evaluated using R² as the primary metric. Fig. 5.12 provides a visual comparison of the prediction performance of all four models for R². It is evident from subfigures (c) and (d) that GBM and RF models closely match the ground truth data, while subfigures (a) and (b) for SVR and LR exhibit notable deviations, particularly in location estimation. These plots reinforce the tabulated statistics by showcasing the robustness and predictive capability of ensemble models in capturing non-linear mappings between modal frequencies and structural damage parameters. The comparative results from Table 5.7 clearly indicate that the RF and GBM models significantly outperform SVR and LR in both localization and depth quantification tasks. Specifically, the RF model achieved the highest R² scores of 0.998 for damage location and 0.993 for damage depth, alongside the lowest MSE values (0.33 and 0.118, respectively). GBM also

delivered commendable performance with R^2 scores of 0.987 and 0.966 for location and depth, respectively.

5.3.3 Hyperparameter tuning

The grid search technique was utilized for hyperparameter tuning, where specific values for hyperparameters were systematically defined in a grid. This process involved creating a grid of potential parameter values for each MIMO model. The selected hyperparameters denote the optimal starting point for refining the model's performance. This approach to hyperparameter tuning is advantageous because it doesn't exhaustively explore every possible combination. Instead, it employs a randomized selection from the predefined grid of hyperparameter values to train and assess the model. The number of search iterations was limited to 100, considering the available resources and time constraints. The initial grid of hyperparameter values is outlined in Table 5.8.

Table 5.8: Grid of Parameters for Hypertuning in MIMO Regression Models

Model	Parameter	Grid Values	Tuned Value
SVR	Kernel type	{'RBF', 'linear', 'poly'}	'poly'
	Reg. Param (C)	1, 10, 25	10
GBM	Estimators	50 - 200	152
	Learning Rate	0.05, 0.1, 0.5	0.1
	Max Depth	3, 5, 7	5
RF	Estimators	10 - 100	45
	Max Depth	None, 10, 20	15
	Min Samples Split	2, 5, 10	5

The method systematically evaluated the models' performance for each combination of hyperparameters within the grid. During this process, the models were trained and cross-validated multiple times using different hyperparameter combinations. Performance metrics such as MSE, MAE, and R^2 score were meticulously analyzed for each combination. Table 5.9 showcases substantial improvements in the performance metrics of MIMO regression models after hyperparameter tuning. LR was excluded from the tuning process due to its simplicity and lack of intricate interdependencies requiring fine-tuning. The RF model demonstrated a 12% reduction in MSE and a 14% decrease in MAE for damage location prediction.

Table 5.9: Improved performance metric after hyperparameter tuning.

Model	Output	MSE	MAE	MedAE	R ²	EV Score	MSLE	MAPE
RF	Damage location	0.29	8.46	5.69	0.997	0.998	0.04	0.134
	Damage depth	0.095	0.138	0.036	0.994	0.991	0.0062	0.0512
GBM	Damage location	0.70	6.56	11.867	0.983	0.983	0.1136	0.2328
	Damage depth	0.0552	0.1216	0.061	0.972	0.973	0.0026	0.0394
SVR	Damage location	51.45	183.52	233.55	0.067	0.0704	0.8208	1.6392
	Damage depth	0.5866	0.4704	0.4176	0.6144	0.6152	0.0488	0.2368

Similarly, the GBM model exhibited significant enhancements, with a 17% reduction in MSE and a 15% decrease in MAE. The SVR model showed remarkable progress, achieving a substantial 20% decrease in MSE and a 16% reduction in MAE for damage location. Regarding damage depth prediction, notable improvements were observed as well. RF experienced nearly a 19% reduction in MSE and a 16% decrease in MAE. GBM demonstrated around a 10% decrease in both MSE and MAE, indicating enhanced depth prediction accuracy. SVR exhibited significant enhancements with approximately a 22% reduction in MSE and a 22% decrease in MAE for damage depth. These enhancements highlight the effectiveness of the hyperparameter tuning process, making the MIMO models more precise and reliable in capturing the complexities of real-world structural damage.

5.3.4 Inference and testing

To evaluate the generalization capability of the trained model, a new dataset comprising six previously unseen damage cases was introduced in a cantilever beam. These damages were deliberately positioned at three distinct regions: near the fixed support, at the mid-span, and near the free end. The corresponding modal frequencies were extracted through modal analysis and fed as input into the best-performing model—RF—as identified in previous sections. The model was tasked with simultaneously predicting both the location and depth of the damages.

The actual and predicted results for each damage scenario are presented in Table 5.10. The RF-based hybrid MIMO model exhibits a high level of accuracy in predicting both damage location and depth across various segments of the beam. For damage cases near the fixed support, the deviations between actual

Table 5.10: Actual and predicted damage location and depth of new testing dataset.

Category	Location (mm)		Depth (mm)	
	Actual	Predicted	Actual	Predicted
Near Fixed Support	83	91.2	0.8	1.01
	208	196.1	1.9	1.656
Mid-Section	402	414	2.7	2.556
	597	593.5	5.1	4.8
Free End	773	787.3	3.3	3.816
	887	889.31	4.1	3.74

and predicted values are minimal, suggesting the model's strength in detecting early structural impairments. In the mid-span and free-end regions as well, the model performs consistently, capturing both location and depth with commendable precision. These findings emphasize the effectiveness of modal frequencies as input features and validate the RF model's robustness in real-world damage scenarios.

5.4 Implementation using DL: EEFO - optimized model

As a direct extension of the ML framework, the DL implementation (Fig. 5.13) retains the same hybrid dataset and preprocessing but replaces the MIMO regressor

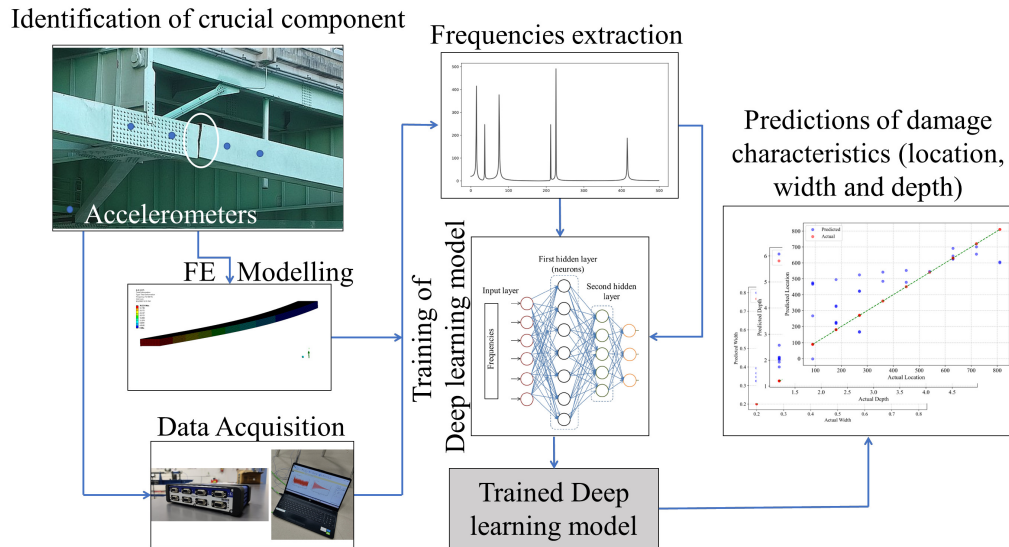


Figure 5.13: Flowchart of the proposed frequency-based DL model

with a neural network that learns the nonlinear mapping from the first six natural frequen-

cies to the damage state. The target vector is expanded to three outputs—location, width, and depth—so the network uses a shared feature extractor and a multi-output regression head to predict all quantities simultaneously. Inputs are standardized; the dataset is partitioned into stratified train/validation/test splits; and training minimizes a weighted sum of mean-squared errors with early stopping and regularization to prevent overfitting. After model selection on the validation set, the trained DL model is used for inference on experimental frequency measurements to localize and quantify damage, while the data pathway and evaluation protocol remain identical to those used for the ML stage.

5.4.1 Extended output parameter

Building upon the previous MIMO regression framework that predicted two damage parameters—location and depth—this deep learning-based approach extends the output space to include damage width as an additional critical descriptor. The inclusion of width provides a more holistic representation of damage severity, thereby enhancing the diagnostic capability of the model.

In this study, each data instance is represented by an input vector $\mathbf{X} \in \mathbb{R}^6$, comprising the first six natural frequencies of the cantilever beam:

$$\mathbf{X} = [f_1, f_2, f_3, f_4, f_5, f_6] \quad (5.1)$$

The corresponding output vector $\mathbf{Y} \in \mathbb{R}^3$ now includes:

$$\mathbf{Y} = [\text{Location}, \text{Width}, \text{Depth}] \quad (5.2)$$

Unlike classical regression models, the DLL models explored in this study are capable of capturing complex, non-linear relationships between frequency-domain inputs and multiple structural damage parameters. This extended output formulation enables simultaneous prediction of all three attributes in a single inference pass, making it well-suited for real-time structural health monitoring and decision support applications.

5.4.2 Model architectures development

The following DL models were tested: The hybrid dataset collected from the damaged beams, along with their damage location, width, and depth, are used to train the selected

DL models. This process includes data preprocessing, model architecture selection, compilation configuration, and training setup. Rows with missing values were removed, and features were scaled between 0 and 1 using the MinMaxScaler. The dataset was then split using an 80-20 ratio for training and testing.

5.4.2.1 Hyperparameter optimization

Hyperparameters were optimized using the EEFO algorithm. EEFO leverages electrolocation-based global search and foraging-based local

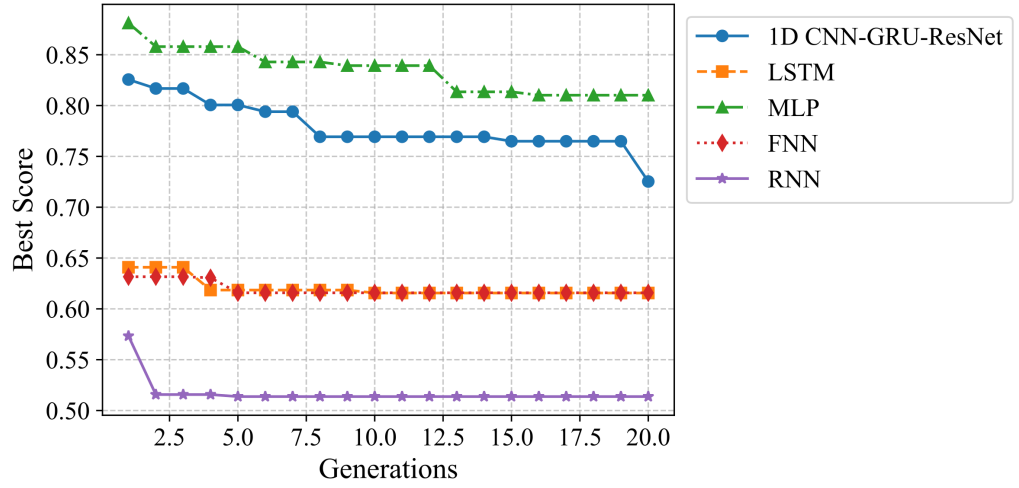


Figure 5.14: Evolution of best scores across generations for DL models

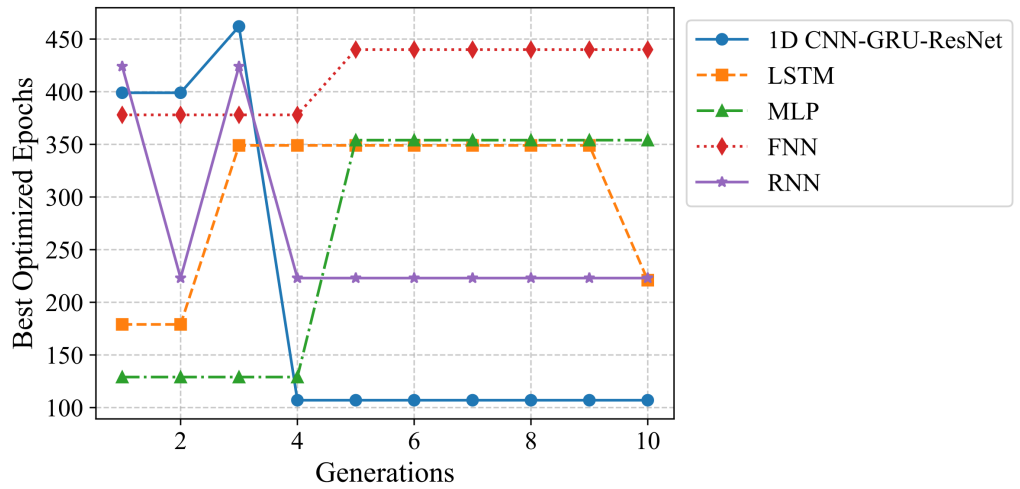


Figure 5.15: Variation of best optimized epochs across generations for DL models

refinement to explore the hyperparameter space, tuning variables such as learning rate, neuron count, dropout rate, batch size, and number of epochs. The convergence behavior and optimized epochs are illustrated in Figures 5.14 and 5.15, respectively. Final hyperparameters selected through EEFO were used in training and validation. All models were compiled using the Adam optimizer with MSE loss and MAE monitoring. Table 5.11 presents the EEFO-optimized parameters.

5.4.2.2 Learning behavior and efficiency

Fig. 5.16 shows the training and validation loss (MSE) curves. The 1D CNN-GRU-ResNet achieved the fastest convergence with the lowest final loss, followed by stable behavior in FNN and MLP. Fluctuations in RNN and LSTM were due to the complexity of learning temporal dependencies in frequency data.

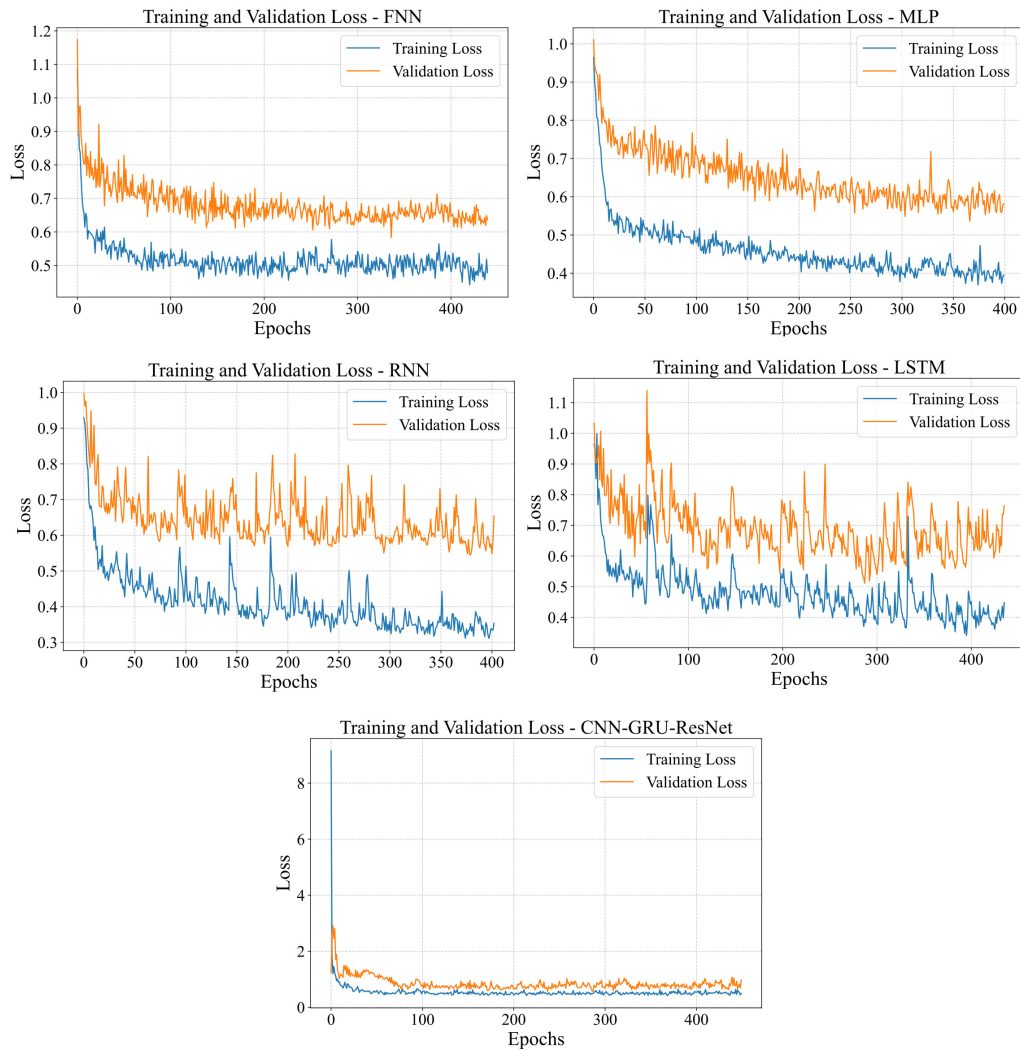


Figure 5.16: Model learning curves

Table 5.11: EEFO-optimized hyperparameters for the deep learning models

Model	Units	Learning Rate	Dropout Rate	Batch Size
FNN	224	0.00804	0.304	21
MLP	233	0.00304	0.462	61
RNN	147	0.00176	0.180	15
LSTM	323	0.0083	0.209	17
1D CNN-GRU-ResNet	435 (CNN), 224 (GRU)	0.00792	0.198	8

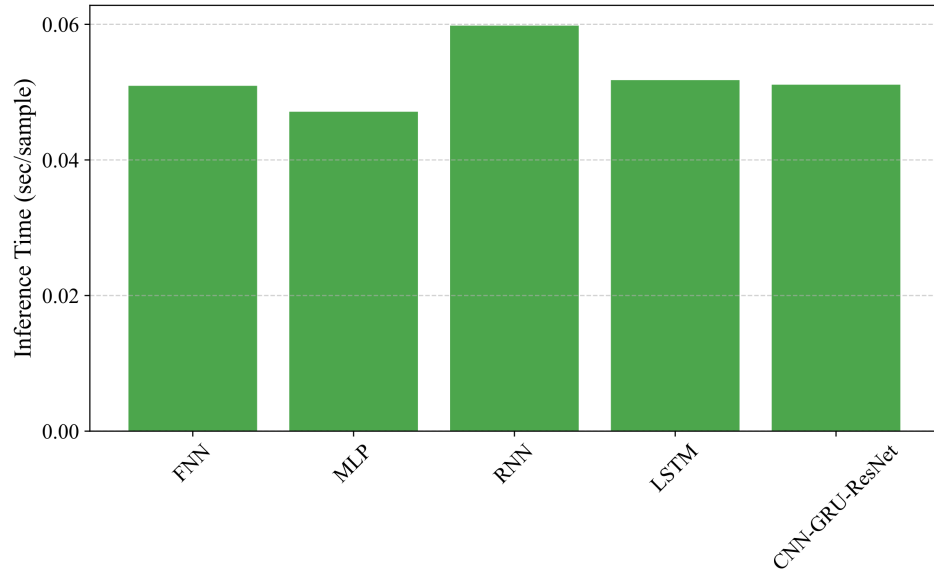


Figure 5.17: Inference time comparison across DL models

Fig. 5.17 illustrates inference time per sample. MLP showed the fastest inference due to its simple architecture, while RNN had the highest due to sequential processing. The 1D CNN-GRU-ResNet balanced inference speed and predictive performance effectively. Overall, this DL framework, enriched by the EEFO-optimized settings and extended output space, enhances the reliability and resolution of structural damage detection—making it highly suitable for practical structural health monitoring and maintenance planning.

5.4.3 Performance analysis

The performance of each model was assessed using key metrics such as MSE and MAE, as shown in Table 5.12, for both training and testing datasets. These metrics provide a comprehensive evaluation of the models' predictive accuracy and generalization capabilities. Among the models, the LSTM model exhibits the lowest MSE for both training and testing, indicating its superior ability to capture temporal dependencies and generalize

effectively across datasets. The MLP model also performs well, showing a balanced trade-off between training and testing errors. While its MSE and MAE values are slightly higher than those of LSTM, they remain competitive, highlighting its efficiency in learning patterns from the data. The RNN model demonstrates strong performance in training, with a relatively low MSE, but its testing error is higher compared to LSTM and MLP, suggesting potential overfitting to training data.

Table 5.12: Performance Metrics of Different Models

Model	MSE (Train)	MSE (Test)	MAE (Train)	MAE (Test)
FNN	0.508	0.582	0.534	0.589
MLP	0.572	0.585	0.586	0.569
RNN	0.443	0.626	0.451	0.678
LSTM	0.378	0.446	0.494	0.572
CNN-GRU-ResNet	0.504	0.528	0.518	0.531

The FNN model records higher errors across both MSE and MAE, indicating that it may not be as effective in capturing complex relationships in the dataset. The CNN-GRU-ResNet model, while leveraging convolutional and recurrent layers for feature extraction and sequential learning, does not show a significant advantage over other models in terms of overall performance. This suggests that while the model can effectively learn spatial and temporal patterns, its complexity may not always translate to superior predictive accuracy in this particular dataset. Overall, the LSTM model emerges as the best performer, followed closely by 1D CNN-GRU-ResNet. The RNN model shows promising results but exhibits a higher test error, while FNN demonstrates comparatively larger deviations.

From the testing dataset, it has been observed that the absolute errors for width predictions across all models are relatively small, despite appearing significant when expressed as percentages due to the low actual values (e.g., 0.4 and 0.6). For example, the CNN-GRU-ResNet model shows an error of only 0.07 for an actual width of 0.4 and 0.00 for an actual width of 0.6, demonstrating high precision. Similarly, the LSTM model achieves near-perfect predictions for width, with an error of just 0.02 for an actual width of 0.4. For location predictions, CNN-GRU-ResNet and LSTM provide the most accurate estimates.

(a) FNN

	Actual	Predicted	Error
l	90	105.04	15.04
	270	369.43	99.43
	450	457.64	7.64
	720	685.69	34.31
w	0.4	0.41	0.01
	0.6	0.59	0.01
	0.8	0.79	0.01
d	1.2	1.38	0.18
	2.4	2.35	0.05
	3.6	3.42	0.18
	4.8	4.79	0.01

(b) MLP

	Actual	Predicted	Error
l	90	94.05	4.05
	270	260.88	9.12
	450	457.80	7.80
	720	700.41	19.59
w	0.4	0.51	0.11
	0.6	0.60	0.00
	0.8	0.80	0.00
d	1.2	1.57	0.37
	2.4	2.39	0.01
	3.6	3.50	0.10
	4.8	4.86	0.06

(c) RNN

	Actual	Predicted	Error
l	90	63.87	26.13
	270	289.14	19.14
	450	380.02	69.98
	720	680.51	39.49
w	0.4	0.50	0.10
	0.6	0.65	0.05
	0.8	0.89	0.09
d	1.2	1.49	0.29
	2.4	2.32	0.08
	3.6	3.37	0.23
	4.8	4.81	0.01

(d) LSTM

	Actual	Predicted	Error
l	90	94.88	4.88
	270	271.40	1.40
	450	454.25	4.25
	720	713.8	6.2
w	0.4	0.47	0.07
	0.6	0.65	0.05
	0.8	0.81	0.01
d	1.2	1.59	0.39
	2.4	2.57	0.17
	3.6	3.79	0.19
	4.8	4.68	0.12

(e) CNN-GRU-ResNet

	Actual	Predicted	Error
l	90	85.69	4.31
	270	268.5	1.50
	450	439.02	10.98
	720	708.18	11.82
w	0.4	0.42	0.02
	0.6	0.6	0.00
	0.8	0.81	0.01
d	1.2	1.33	0.13
	2.4	2.40	0.00
	3.6	3.48	0.12
	4.8	4.63	0.17

The CNN-GRU-ResNet model predicts a location of 708.18 mm for an actual value of 720 mm, with an error of only 11.82 mm, while LSTM predicts 713.8 mm with an error of 6.2 mm, making it the best-performing model for location estimation. In contrast, models such as RNN and FNN show larger deviations, with errors reaching up to 39.49 mm and 34.31 mm, respectively.

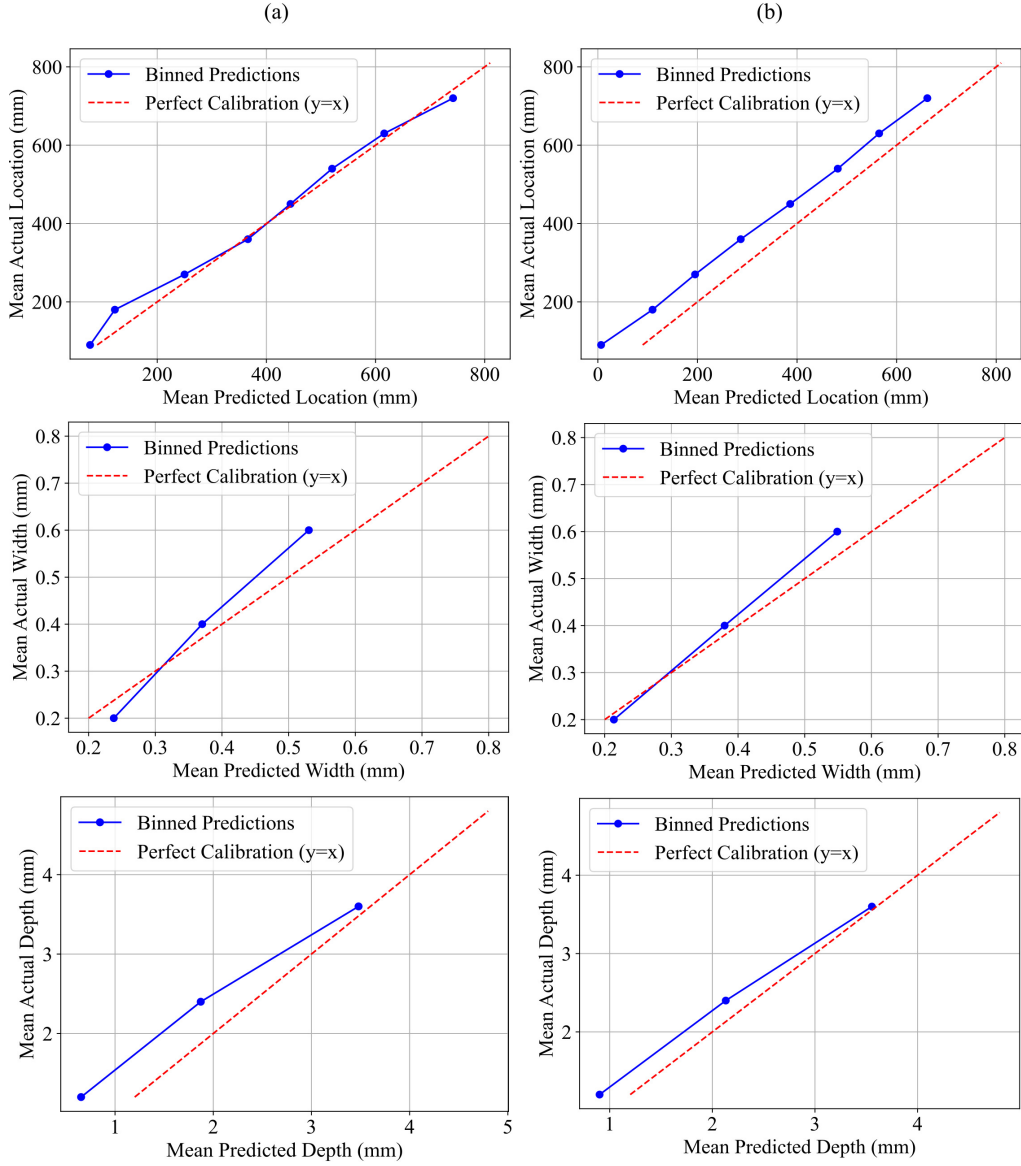


Figure 5.18: Calibration plots (a) LSTM, (b) 1D CNN-GRU-ResNet

Regarding depth predictions, CNN-GRU-ResNet and LSTM again demonstrate superior accuracy. For an actual depth of 4.8, CNN-GRU-ResNet predicts 4.63 with an error of 0.17, while LSTM achieves an even lower error of 0.12. Other models, such as FNN and RNN, exhibit higher depth prediction errors, particularly for smaller values like 1.2 mm, where RNN deviates by 0.29 mm compared to CNN-GRU-ResNet's smaller deviation of 0.09 mm. Overall, LSTM and CNN-GRU-ResNet emerge as the best-performing models, particularly for location and depth predictions, with minimal absolute errors across different test cases. The strong performance of these models highlights their ability to accurately capture structural damage characteristics. However, while actual vs. predicted values demonstrate accuracy, it is also necessary to verify whether the model is well-calibrated,

meaning that its confidence in predictions aligns with actual outcomes.

A well-calibrated model ensures that its predicted values accurately reflect true outcomes across different intervals, making calibration an essential factor in assessing model reliability. Fig. 5.18 presents the calibration plots for LSTM and 1D CNN-GRU-ResNet across the three damage parameters—location, width, and depth—where Fig. 5.18 (a) corresponds to LSTM, and Fig. 5.18 (b) corresponds to 1D CNN-GRU-ResNet. The red dashed line represents perfect calibration ($y = x$), serving as a reference for ideal prediction confidence, while the blue markers indicate binned mean predictions, providing insight into how well each model's predictions align with actual values.

For location calibration, LSTM exhibits stronger alignment with the perfect calibration line, demonstrating minimal deviations across the entire range. The binned predictions closely follow the diagonal, suggesting that LSTM maintains well-balanced confidence in its location predictions. In contrast, 1D CNN-GRU-ResNet shows slight underestimation at lower values and minor deviations in higher values, indicating a small systematic bias. This suggests that LSTM provides more consistent calibration for location, ensuring that predicted values correspond well to actual outcomes. However, in the case of width calibration, 1D CNN-GRU-ResNet demonstrates superior calibration stability, with its binned predictions closely following the perfect calibration line across all intervals. LSTM, on the other hand, exhibits slight underestimation, particularly in lower width values, suggesting a mild miscalibration issue. This indicates that 1D CNN-GRU-ResNet provides more reliable and stable width predictions, whereas LSTM predictions may be slightly overconfident in certain ranges.

For depth calibration, 1D CNN-GRU-ResNet once again outperforms LSTM, as its binned predictions are more consistently aligned with the ideal calibration line. LSTM, in contrast, exhibits larger deviations, particularly in the lower depth values, where its predictions appear underestimated. This suggests that LSTM struggles with uncertainty estimation for depth, making 1D CNN-GRU-ResNet the more reliable model for this parameter. Overall, while LSTM demonstrates better calibration for location, the 1D CNN-GRU-ResNet model provides superior calibration performance for width and depth. The more stable and consistent calibration observed in 1D CNN-GRU-ResNet for these parameters suggests that it maintains a better balance between predicted values and actual outcomes, making it a more reliable model for practical applications where confidence es-

timination is crucial.

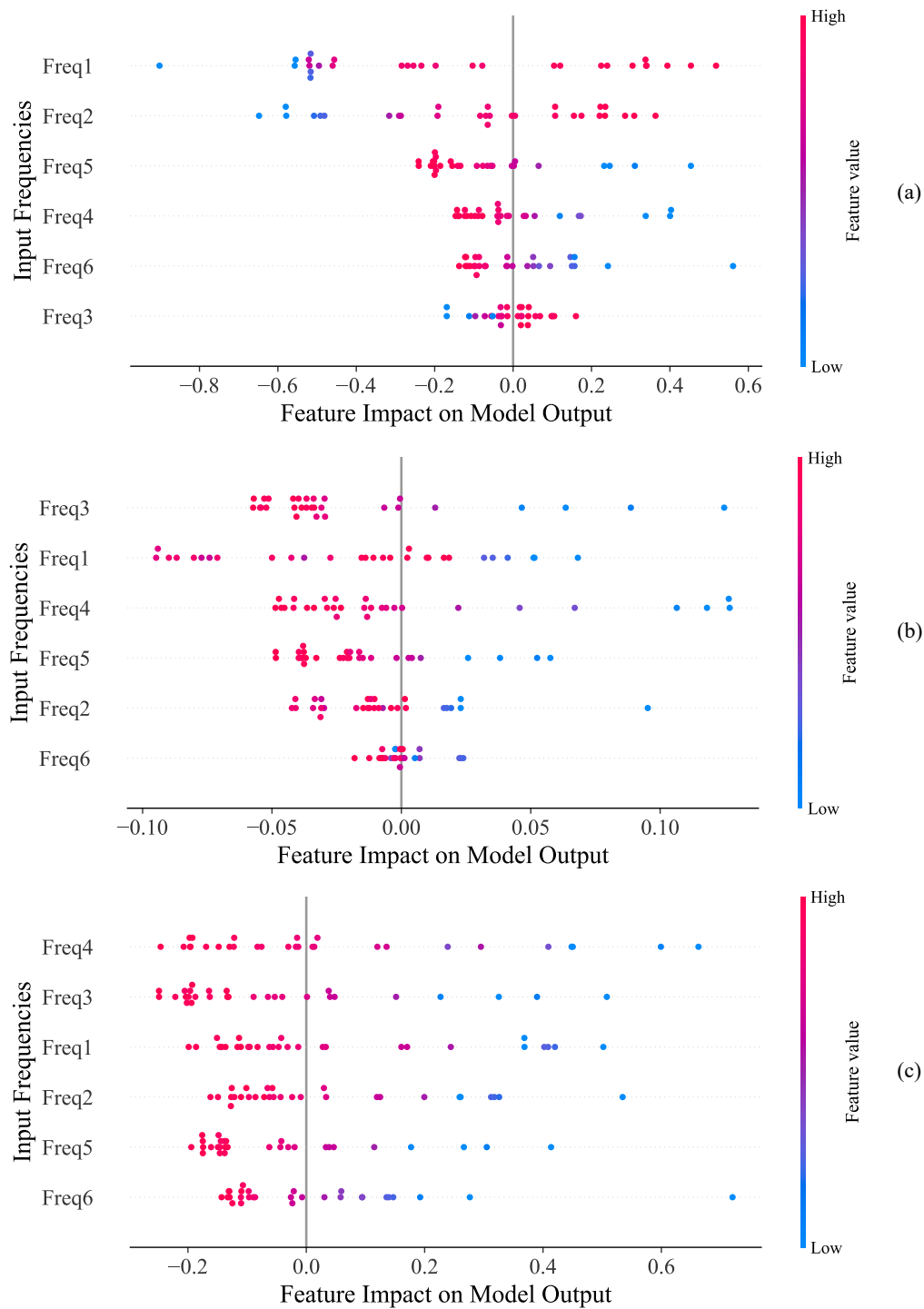


Figure 5.19: Best model: SHapley additive explanations (a) location, (b) width, (c) depth

The SHAP analysis presented in Fig. 5.19 provides a comprehensive understanding of how different modal frequencies contribute to the prediction of damage characteristics—location (a) using the LSTM model, and width (b) and depth (c) using the 1D CNN-

GRU-ResNet model. Each subplot represents the feature importance of the six input frequencies (Freq1–Freq6) for the respective damage output, with the color gradient indicating feature values (blue for low values and red for high values). The SHAP analysis presented in Fig. 5.19 provides a comprehensive The horizontal spread of points signifies the impact of each frequency on the model’s predictions, with values farther from zero having a stronger influence. It has been observed that the ranking of influential frequencies differs for each output parameter. Since damage location, width, and depth affect the structural response differently, the modal frequencies exhibit distinct sensitivities to each damage characteristic.

Freq1 and Freq2 exhibit dominant influence, which aligns with structural dynamics principles, as lower-order frequencies are typically more sensitive to global changes in stiffness distribution, making them crucial for identifying damage location. Freq3 and Freq1 have the highest contribution, indicating that specific frequency modes are more responsive to lateral expansion of damage, likely due to localized stiffness variations. Freq4 and Freq3 emerge as the most critical contributors, suggesting that depth variations induce greater shifts in higher-order frequency modes, which are known to capture localized reductions in stiffness and mass loss effects.

5.5 Comparative summary implementations

The key differences between the ML and DL approaches used in this thesis are outlined in Table 5.13. Both approaches used modal frequencies (f_1 to f_6) as input features, but their capabilities diverged in terms of output and overall flexibility. The ML-based MIMO model, particularly with Random Forest, showed excellent accuracy in predicting damage location and depth, while being relatively fast and easier to interpret. In contrast, the deep learning models—optimized using the EEFO algorithm—were able to handle more complex tasks, including the prediction of crack width alongside location and depth. Among them, LSTM performed best for locating damage, whereas CNN-GRU-ResNet proved more effective for estimating width and depth. DL also enabled deeper insight through SHAP-based interpretability, making it a valuable choice for multi-parameter damage assessment, despite its higher computational demand. Overall, the choice between ML and DL depends on whether the focus is on speed and simplicity or on extracting richer, more

detailed damage information.

Table 5.13: Comparison of ML and DL-based approaches for damage assessment

Aspect	ML (MIMO Model)	DL (EEFO-Optimized)
Input Features	Modal Frequencies (f_1 to f_6)	Modal Frequencies (f_1 to f_6)
Outputs	Location, Depth	Location, Width, Depth
Models Used	RF, SVR, GBM, LR	LSTM, CNN-GRU-ResNet, RNN, MLP
Optimization	Grid Search	EEFO
Best Performer	RF ($R^2 = 0.997$)	LSTM (location), CNN-GRU (width/depth)
Interpretability	MSE, MAE, R^2	SHAP, MSE, MAE
Strengths	Simpler, fast, high accuracy	Multi-parameter prediction

5.6 Conclusion

The chapter demonstrated the development and validation of advanced AI-based models for simultaneous localization and quantification of structural damage in beam-like elements. A MIMO regression framework was employed, where among the tested machine learning models (SVR, LR, RF, and GB), the Random Forest model achieved the highest accuracy with R^2 values of 0.995 for damage location and 0.936 for depth, and error margins ranging from 0.1–9.5% for location and 8–25% for depth. These results confirmed that the first six modal natural frequencies are highly sensitive indicators of stiffness reduction, enabling the detection of even small-scale damages. The experimental dataset was further complemented with finite element simulations, with modal frequency deviations maintained below 9%, thereby ensuring the reliability of the training data.

In parallel, deep learning models optimized using the EEFO algorithm significantly extended the predictive capability by incorporating damage width alongside location and depth. Among the tested architectures, the LSTM model achieved the best overall performance, with Mean Squared Error values of 0.378 (train) and 0.446 (test), and Mean Absolute Error values of 0.494 (train) and 0.572 (test). The hybrid CNN-GRU-ResNet

model provided comparably strong results, effectively capturing complex structural behaviors through a combination of convolutional, recurrent, and residual layers. Both models showed excellent sensitivity to small-scale defects, with LSTM predicting location within 1.5% error and width and depth within 8%, while CNN-GRU-ResNet achieved location predictions within 2.6% and width and depth within 5%. Robust generalization across varying scenarios was also observed, with calibration analysis highlighting LSTM as more reliable for location and CNN-GRU-ResNet as superior for width and depth estimation.

Finally, model interpretability using SHAP analysis revealed that lower-order modal frequencies (Freq1 and Freq2) contributed most strongly to location prediction, whereas higher-order modes (Freq3–Freq6) played a dominant role in predicting damage width and depth. Collectively, these findings establish a robust and scalable AI-based framework for structural health monitoring of beam-like elements, combining experimental and numerical datasets with interpretable ML and DL models to achieve accurate, reliable, and practically relevant damage detection.

Road Damage Assessment

6.1 Introduction

This chapter presents the practical application of the AI-based road damage classification methodology (outlined in Chapter 4) to real-world pavement segments under actual vehicular loading. A nation's development hinges on its transportation system, with many citizens relying on roads as their primary mode of transportation. Consequently, roads play a pivotal role in driving a country's economic growth and social progress. Beyond facilitating the movement of people and goods, roads serve as essential connectors to fundamental services such as employment, healthcare, and education. As a result, they form a vital component of civil infrastructure and require timely and efficient maintenance to ensure continuous serviceability. Regular inspection of road networks is, therefore, imperative to uphold performance standards and to prevent the escalation of localized defects into more serious structural failures.

Among the various forms of surface distress encountered in road networks are cracks, potholes, rutting, and other forms of deformation or depression. These damages may arise due to a multitude of factors, including improper mix design, substandard construction materials, or poor execution practices during the construction phase. In addition, base or subgrade settlement during operation, combined with natural aging, abnormal axle loads, and insufficient maintenance, further accelerates pavement deterioration. While it is impractical to expect a damage-free road over its service life, early detection and timely intervention remain critical to preventing accelerated degradation and extending pavement life. This underscores the growing need for advanced, automated damage detection systems capable of replacing traditional visual inspection methods, which are labor-intensive, slow, and disruptive to traffic flow. In this context, the integration of AI techniques with vehicle-mounted vibration sensing offers a scalable, cost-effective, and automated alternative for surface condition monitoring.

The present case study explores this integration through a two-stage classification framework. The first phase focuses on binary classification of road health—distinguishing

between healthy and damaged segments—and investigates the correlation between model predictions and the IRI. The second phase extends the analysis to a multi-class classification problem, aiming to identify specific surface-level damage types. Both stages utilize the same experimental setup and feature extraction methodology, while differing in label strategy, classification objective, and model evaluation metrics. Together, these efforts contribute to the development of a robust, real-time AI-enabled road assessment system that aligns with the broader goals of intelligent transportation infrastructure and smart city deployment.

This chapter applies the proposed AI-based methodology to the case study of road damage assessment. Section 6.1 introduces the objectives and scope of the study, followed by a description of the field study setup and data collection process in section 6.2, including the data acquisition system (section 6.2.1). Section 6.3 presents the first model, which focuses on the binary classification of road damage and the estimation of the IRI. This includes the determination of IRI (section 6.3.1), model development (section 6.3.2), and maintenance cost assessment (section 6.3.3). Section 6.4 introduces the second model, aimed at multi-class classification of road damages, covering data processing and feature extraction (section 6.4.1), model development (section 6.4.2), and a comparative analysis of the proposed classifier against alternative approaches (section 6.4.3). Section 6.5 extends the analysis by estimating rutting-related maintenance costs, thereby linking model outputs to practical decision-making. Finally, Section 6.6 concludes the chapter by summarizing the key findings

6.2 Field study setup and data collection

The Indian road network is broadly categorized into National Highways (NH), State Highways (SH), District Roads, and Rural Roads. Among these, State Highways (SH) serve as a crucial link, connecting district and rural roads to the national highway system. Due to increasing vehicular traffic, SHs frequently encounter severe pavement damage such as potholes, rutting, and surface cracks, which not only deteriorate driving conditions but also contribute significantly to road accidents. Timely detection and

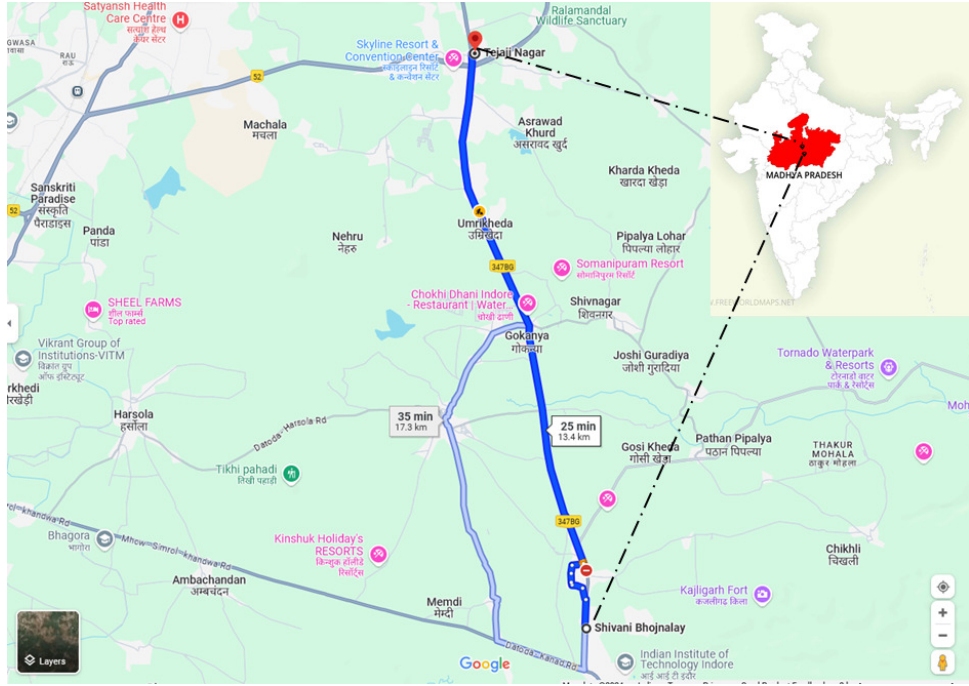


Figure 6.1: Route map of pilot study

rectification of these damages are therefore essential for ensuring public safety and extending the lifespan of SH infrastructure.

To validate the proposed AI based classification methodology, a field study was conducted on a 13-kilometer segment of SH 27, located between Tejaji Nagar and the university campus. This semi-urban stretch features various road defects including potholes, rutting, cracks, and depressions. The typical damages are captured from the selected road segment and shown in Fig. 6.2.

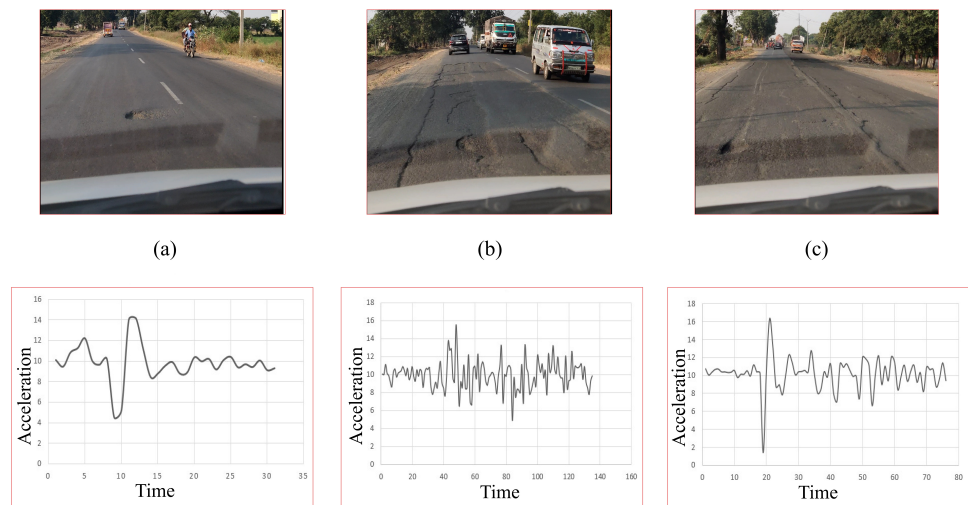


Figure 6.2: Different road damages and their respective vibration signals.

6.2.1 Data acquisition system

A vibration-based sensing system was deployed on a test vehicle for real-time data acquisition of damaged roads as illustrated in Fig. 6.3. For data collection, four uniaxial accelerometers (model 4534-B-002 from Hottinger Brüel & Kjær) were deployed at four locations: on the dashboard, the trunk, and the floor above each of the rear wheels as shown in Fig. 6.4. Each accelerometer had a sensitivity of 50 mV/ms^2 and was interfaced with the Hottinger Brüel & Kjær MX840B DAQ system. This eight-channel universal input DAQ system was operated via CatmanEasy software, which also enabled real-time data monitoring and logging. The technical details of accelerometer and the test vehicle are provided in Table 6.1. Data were recorded in .mat format for post-processing. Although multiple sensors were installed, only the data from the dashboard-mounted accelerometer were used for further analysis in this study.

Table 6.1: Details of car and accelerometer

Item	Component	Specification
Car	Engine Type	K15C Smart Hybrid
	Length (mm)	4395
	Width (mm)	1735
	Wheel Base (mm)	2740
	Dashboard	Sculpted with metallic Teak-Wooden finish
	Front Suspension	MacPherson strut and coil spring
	Rear Suspension	Torsion beam and coil spring
Accelerometer	Company	Hottinger Brüel & Kjær
	Frequency Range	0.2–12800 Hz
	Sensitivity	50 mV/ms^{-2}
	Weight	8.6 grams

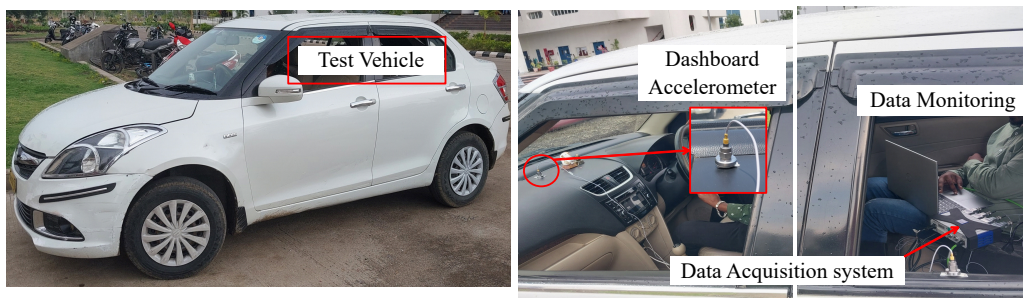


Figure 6.3: Installation of accelerometer and data monitoring.

This is because it was the first point in the vehicle to receive vibration inputs when encountering road surface anomalies, thereby providing the most reliable and least attenuated data. The other sensors, especially those farther from the point of impact (like those on

the trunk or floor), showed secondary effects such as damping and structural attenuation, making their data less suitable for direct interpretation of road conditions.

To ensure accurate ground truth, GPS and a smartphone-based video recording system were also used (see Fig. 6.5). The camera, mounted on the dashboard, continuously recorded the road surface conditions during the drive, synchronized with the accelerometer data using timestamp alignment, as illustrated in Fig. 6.6.

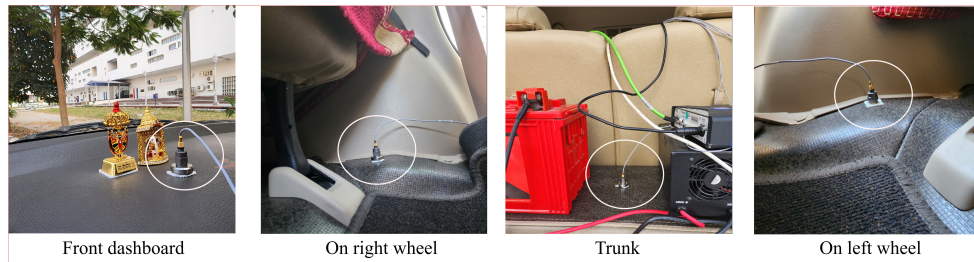


Figure 6.4: Sensors placement on the test vehicle.

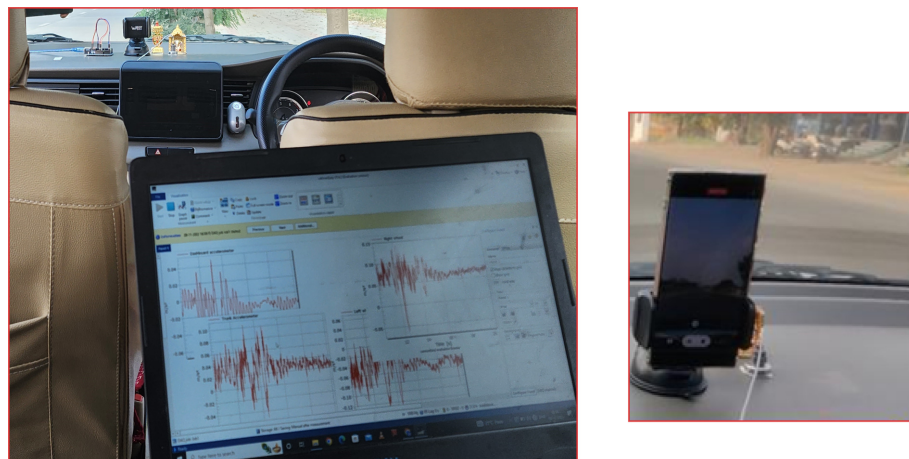


Figure 6.5: Monitoring of vibration response during field study

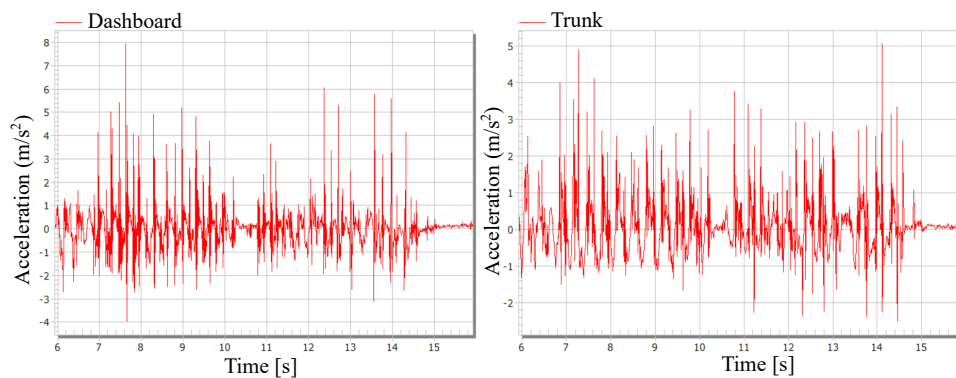


Figure 6.6: Acceleration data from the field monitoring

These measurements served as validation data for the accelerometer signal interpretation. The data collection was performed during daytime under optimal weather conditions to avoid disturbances like water logging or traffic congestion. The vehicle speed was maintained at a constant 28-30 km/h, ensuring consistent sampling of road-induced vibrations. The accelerometer axes were aligned with the vertical (Z-axis) direction to capture maximum vibrations perpendicular to the road surface.

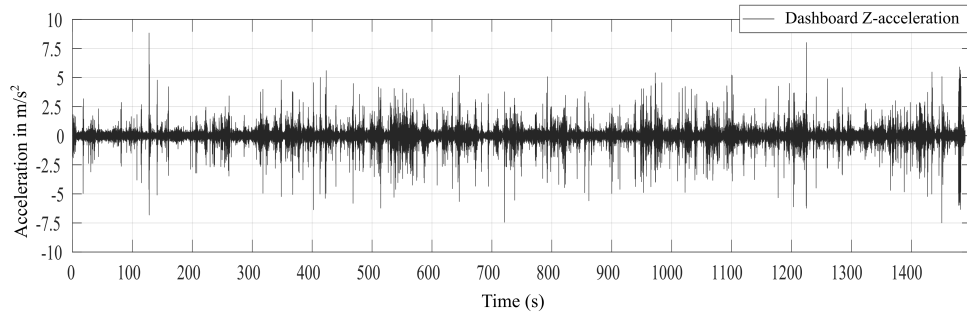


Figure 6.7: A typical acceleration data collected from the dashboard accelerometer.

The recorded acceleration data, illustrated in Fig. 6.7, specifically represents the Z-axis acceleration of the dashboard. In this visualization, the x-axis represents time in seconds, and the y-axis signifies acceleration in m/s^2 , excluding the acceleration due to gravity. The video has been recorded continuously, along with accelerometer. The smartphone camera, crucial for recording the video, was securely positioned on the dashboard using a mobile holder, as illustrated in Fig. 6.5. Additionally, the setup included a laptop displaying real-time readings of the accelerometer data, facilitating the monitoring of acceleration patterns throughout the field study. This comprehensive setup ensured precise alignment between video footage and corresponding accelerometer data, contributing to the robustness of the model training process. The time stamps of video are used to label anomalies in acceleration data, which were subsequently used to train the classification models. Data were sampled at a frequency of 1000 Hz. Over the entire road segment, more than 1.1 million data points were collected, capturing a wide range of road surface conditions.

Building upon this comprehensive dataset, the subsequent sections explore two key models. The first focuses on binary classification, which distinguishes between damaged and undamaged pavement conditions, further integrated with IRI estimations to support maintenance prioritization. The second model involves multi-class classification of road surface defects—such as potholes, cracks, and rutting—aiming to enhance the granularity

of damage detection and enable targeted maintenance strategies. Together, these methodologies offer a unified framework for intelligent road health monitoring and decision-making for optimal maintenance planning.

6.3 Model 1: Binary classification of road damage and estimation of IRI

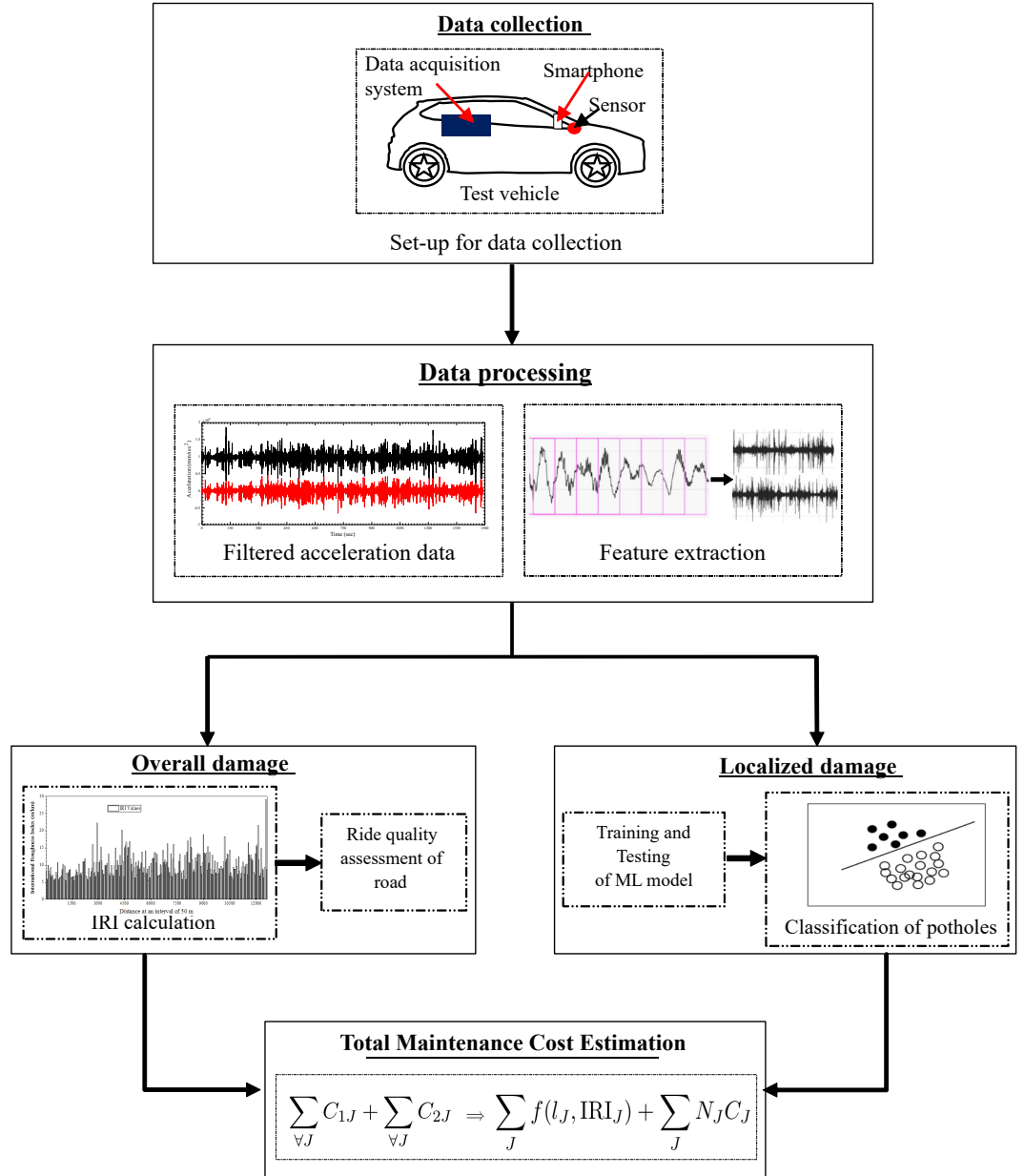


Figure 6.8: A schematic diagram illustrating the binary classification integration with IRI

The primary objective is to evaluate road ride quality while enabling real-time detection and classification of potholes. The ride quality assessment through IRI estimation is performed with the idea of a multi-degree freedom system, while the potholes are detected using classification based damage assessment models as explained in chapter 4. The results obtained for IRI and the number of potholes are then integrated and used to determine the overall maintenance cost. The proposed methodology is based on acceleration data collected from an existing road and analysis of the collected data for damage detection and classification. A flowchart showing the key steps of the proposed methodology is presented in Fig. 6.8. The following subsections provide a detailed explanation of each step.

This acceleration has needed to be filtered because of the noise presented by the test car. The sixth-order Butterworth filter [141] was used for filtering, where the cut-off frequency is taken as 4 Hz. There is neither a standard procedure nor a clear-cut to estimate a cut-off frequency [142]. Several researchers adopt different cut-off frequencies in the range of 0.5 Hz-13 Hz also to filter the horizontal and vertical accelerations [143]. Following the precedent cut-off frequency by [144], this research also adopts a 4 Hz cut-off frequency. The filtered acceleration data (see Fig. 6.9) was used to detect, classify, and localize the damage. For this purpose, the entire acceleration time series was segmented into several windows, and features were extracted from each window. A sliding window technique with fifty percent overlap was used for feature extraction, which improves the feature quality. To explain this process, a 240 m road segment is selected, as shown in the top plot of Fig. 6.10, where the Z-acceleration data is plotted against distance.

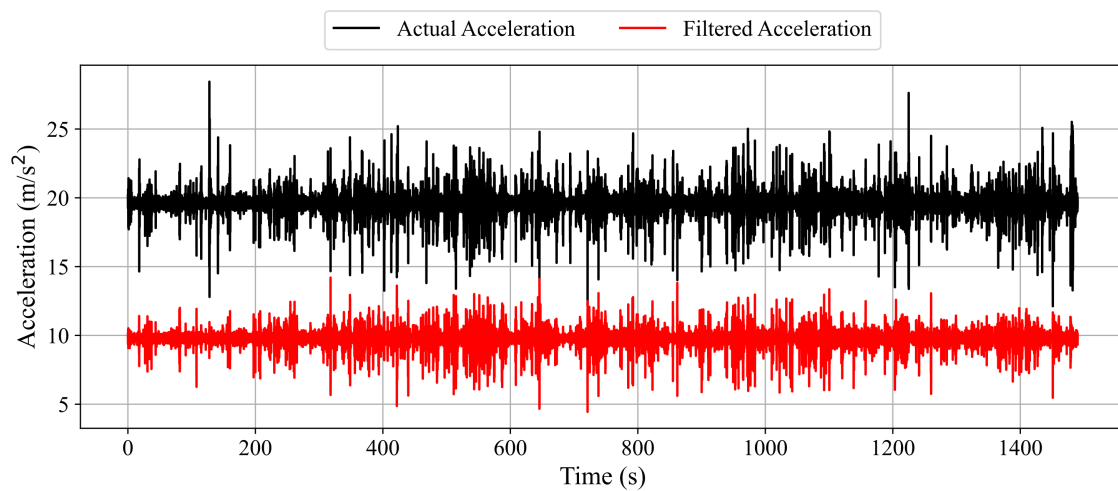


Figure 6.9: Typical raw acceleration data and filtered signal

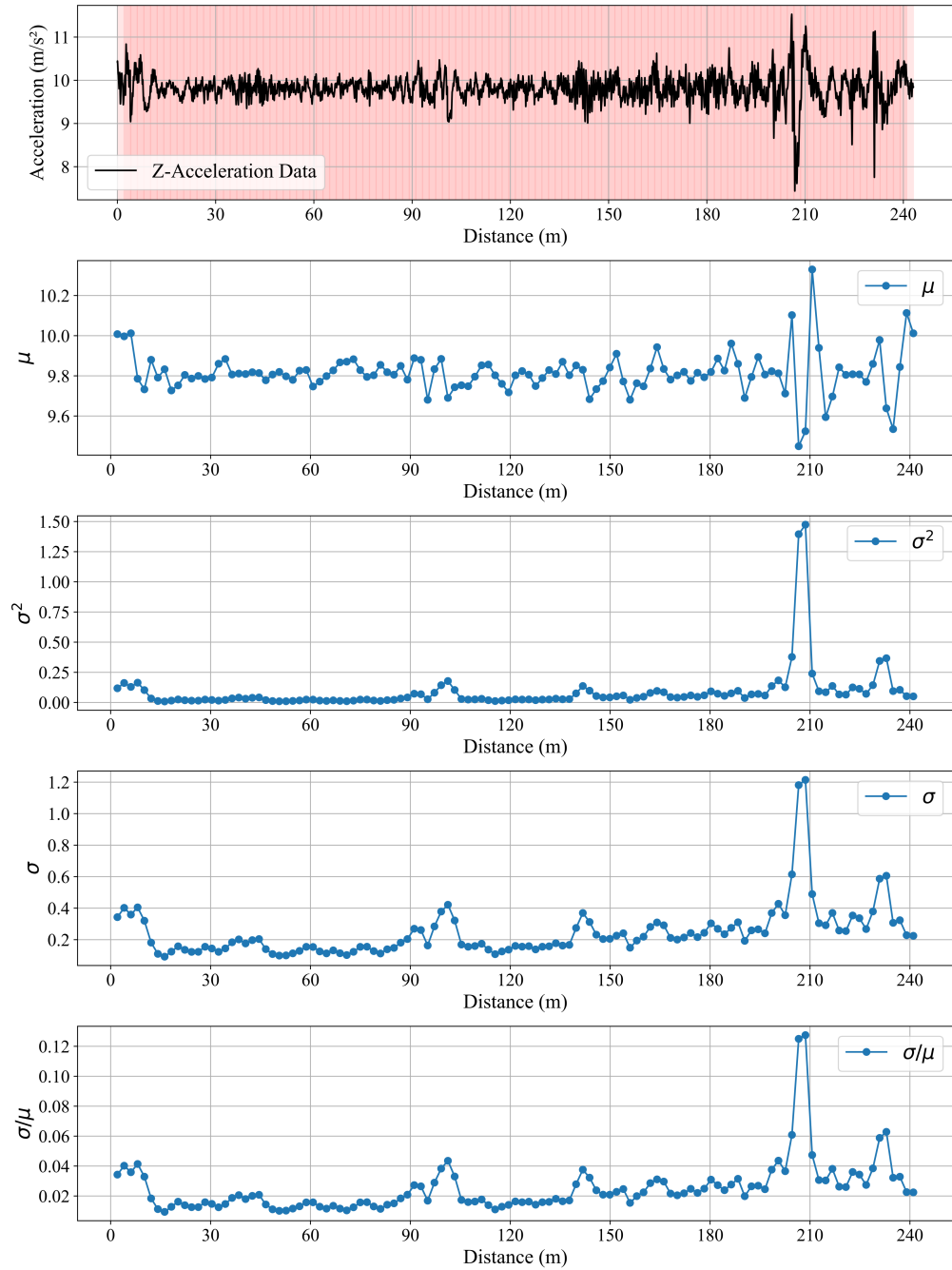


Figure 6.10: Sliding window technique

The red-colored boxes represent the individual windows used for segmenting the data, with each window serving as a small portion of the road segment over which statistical features have been calculated. The remaining plots have shown the corresponding features—mean (μ), variance (σ^2), standard deviation (σ), and coefficient of variation ($\frac{\sigma}{\mu}$)—for each window, helping to visualize how these features fluctuate over the 240-meter segment. A distinct observation has occurred between 200 m and 220 m, where all the feature values have changed suddenly, suggesting a potential road anomaly, most likely

a pothole. Similar to the features shown, a total of 12 have been used to train ML models.

6.3.1 Determination of IRI

The filtered signal obtained from the previous step is utilized to determine the IRI values using Eq. 6.5. Firstly, the acceleration signal is converted into velocity and then displacement. Fig. 6.11 shows the transformed velocity and displacement profile in the vertical direction.

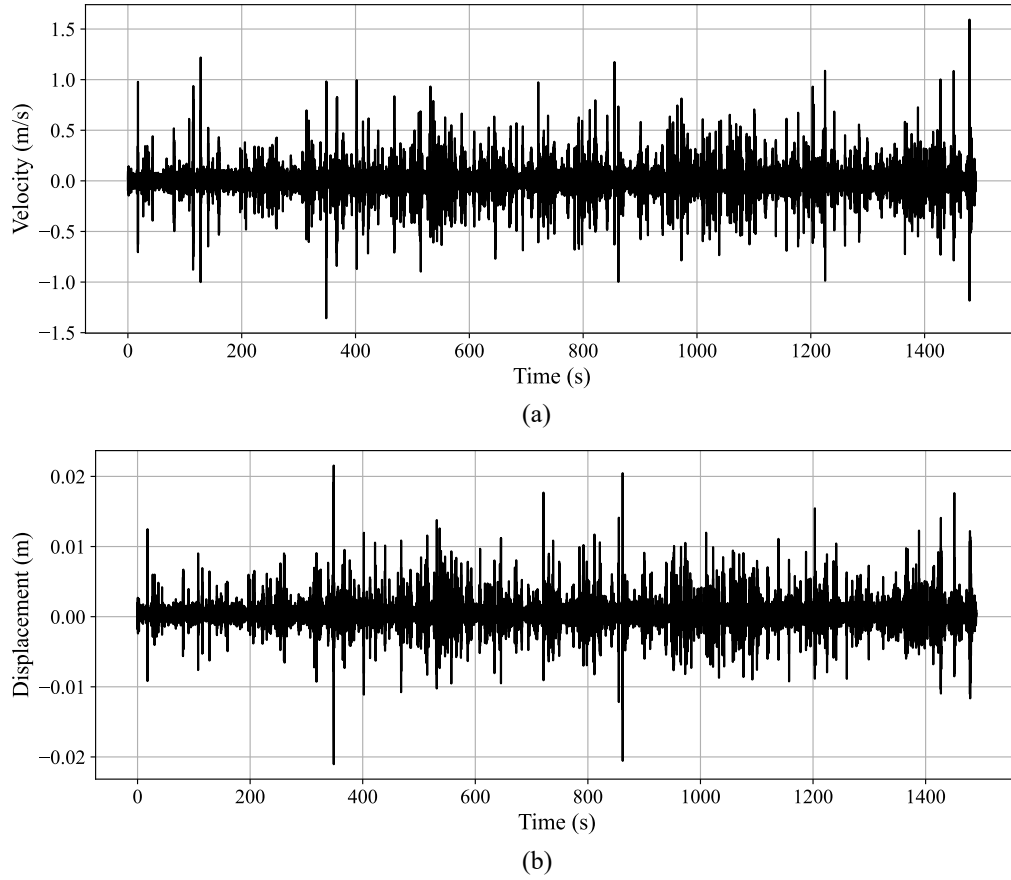


Figure 6.11: (a) The velocity in the Z direction is derived from the filtered acceleration and (b) The displacement in the Z direction is derived from the velocity

Let \ddot{Z} , \dot{Z} , Z and Y denote the *acceleration*, *velocity*, *position coordinate* and the road input displacement in the Z-direction. According to Newton's second law, the equation of motion for the quarter car can be written as [8]:

$$m_s \ddot{Z}_s + C_s (\dot{Z}_s - \dot{Z}_u) + K_s (Z_s - Z_u) = 0 \quad (6.1)$$

$$m_u \ddot{Z}_u + C_s (\dot{Z}_u - \dot{Z}_s) + K_s (Z_u - Z_s) + K_t (Z_u - Y) = 0 \quad (6.2)$$

which can be solved for \ddot{Z}_s and \ddot{Z}_u as follows:

$$\ddot{Z}_s = \frac{1}{m_s} [C_s (\dot{Z}_u - \dot{Z}_s) + K_s (Z_u - Z_s)] \quad (6.3)$$

$$\ddot{Z}_u = \frac{1}{m_u} [C_s (\dot{Z}_s - \dot{Z}_u) + K_s (Z_s - Z_u) + K_t (Y - Z_u)] \quad (6.4)$$

The values of Z_s and Z_u (representing the displacements of the sprung and unsprung masses, respectively) can be obtained from their respective accelerations (\ddot{Z}_s and \ddot{Z}_u) through *numerical integration*. This process involves: (i) starting with initial conditions, such as initial displacement ($Z_s(0)$, $Z_u(0)$) and velocity ($\dot{Z}_s(0)$, $\dot{Z}_u(0)$) at time $t = 0$; (ii) updating the velocities \dot{Z}_s and \dot{Z}_u by integrating the accelerations \ddot{Z}_s and \ddot{Z}_u over time; and (iii) updating the displacements Z_s and Z_u by integrating the velocities over time. This numerical integration allows tracking of the vertical motion of the vehicle over time, which is essential for analyzing the suspension system's response to road damages such as potholes.

Once Z_u and Z_s are determined, these are used to estimate the IRI value. The IRI is defined as the ratio of the accumulated absolute vertical displacement between the sprung and unsprung masses of a quarter-car model to the distance traveled [11], and is mathematically given as:

$$\text{IRI} = \frac{1}{L} \int_0^L |Z_s(x) - Z_u(x)| dx \quad (6.5)$$

where L is the length of the road segment, $Z_s(x)$ is the vertical displacement of the sprung mass, and $Z_u(x)$ is the vertical displacement of the unsprung mass at position x . The unit of IRI is m/km. The integration in Eq. 6.5 is evaluated numerically by using longitudinal profile elevation data of the pavement, which is derived by adding or subtracting the suspension deflection from the sensor readings. In short, the IRI determination for a given road segment involves data collection using a sensor such as an accelerometer, as well as the determination of velocity and displacement profile through numerical integration. Once the IRI values are known, they are used to classify the road condition. Although the vibration data were collected using a real test vehicle, the IRI values in this study were not computed directly from raw accelerations. Instead, IRI was obtained from a standardized quarter-car simulation model, where the sprung and unsprung displacements were derived through numerical integration. In the past, researchers have made attempts to classify road roughness (or travel quality) based on IRI values. Table 6.2 presents the FHWA IRI thresh-

olds for highways, classifying ride quality into categories such as Very good, Good, Fair, Mediocre, and Poor for both interstate and other highways. The table provides corresponding IRI values in both metric (m/km) and imperial (in./mil) units, offering a comprehensive scheme for assessing travel quality for different roads.

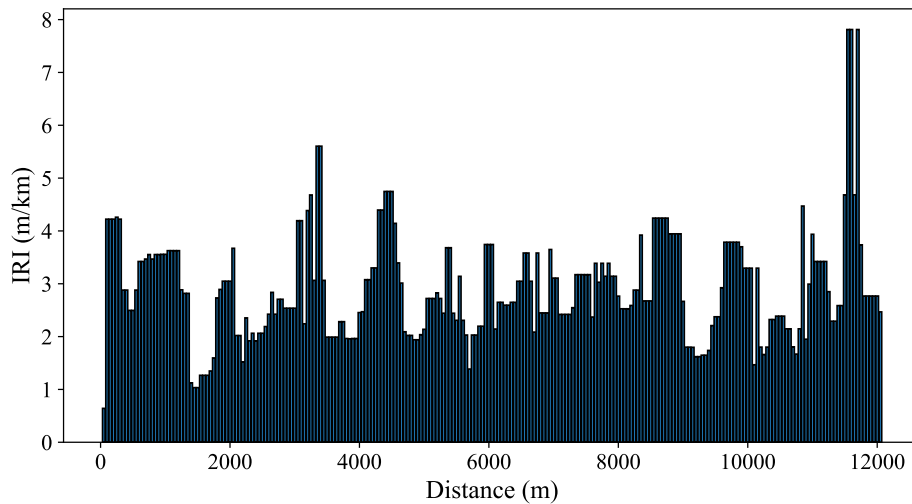


Figure 6.12: Segment-wise IRI values for every 50 m interval along the test road

Next, the IRI values have been calculated using the vertical displacement profile. For this purpose, the total road length was subdivided into 252 segments, each having a 50 m length. The IRI value was determined using Eq. 6.5, and the same has been shown in Fig. 6.12, where the distance traveled is plotted on the x-axis. Furthermore, the histogram of IRIs with superimposed Gaussian distribution is plotted in Fig. 6.13.

The IRI values vary between 1 m/km and 8 m/km, with an average value of approximately 2.9 m/km and a standard deviation of 1.0 m/km, indicating moderate variability in surface roughness along the road. According to the ride quality classification criteria, most segments fall within the “Good” to “Fair” category, while a few localized portions with IRI above 5 m/km exhibit “Poor” ride quality, suggesting the presence of surface irregularities and minor deterioration. The frequency distribution in Fig. 6.13 shows that most IRI values lie between 2 and 4 m/km, with a mean of 2.91 m/km and a standard deviation of 1.03 m/km. The approximately Gaussian pattern indicates that the road surface is predominantly smooth, with only a few segments exhibiting higher roughness. The classification of roads based on IRI is one aspect of road damage, and another is the local pothole that often makes ride quality even worse and needs to be identified and localized.

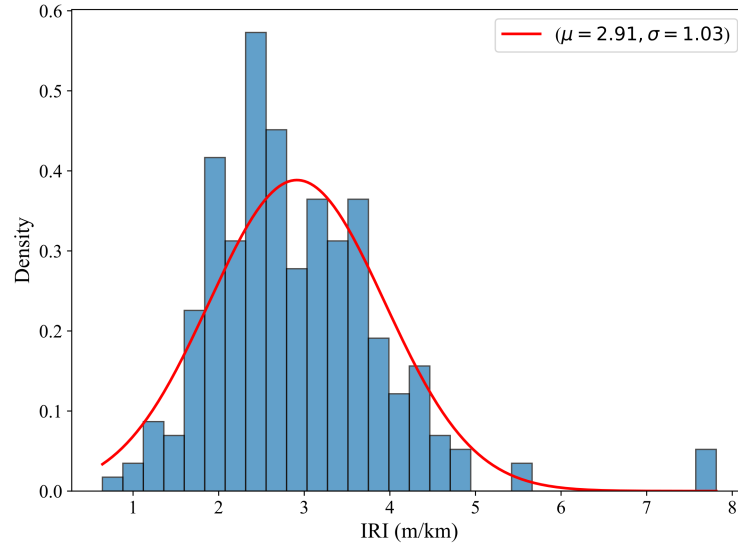


Figure 6.13: Distribution of IRI

Table 6.2: FHWA IRI thresholds for Highways [4].

Ride Quality	Interstate highways		Other highways	
	IRI (m/km)	IRI (in./mil)	IRI (m/km)	IRI (in./mil)
Very good	<0.95	<60	<0.95	<60
Good	0.95–1.50	60–94	0.95–1.50	60–94
Fair	1.50–1.89	95–119	1.50–2.69	95–170
Mediocre	1.90–2.70	120–170	2.70–3.48	171–220
Poor	>2.79	>170	>3.48	>220

6.3.2 Model development

The potholes are detected using ML models such as SVM, RF, logistic regression, and Naive Bayes models. For the selected stretch of road, a total of 73 potholes were identified manually. Out of these, data from 60 potholes was used for training, while the remaining pothole data was used to test the trained model. The features are extracted using the sliding window techniques as described earlier. A total of 2,234 windows were created by each consisting of 1250 data points. Furthermore, these windows have been split into two sets: first, a training set, and second, a test set. The training set consisted of 1787 windows (i.e., 80 percent) and the remaining 447 windows (i.e., 20 percent), which were used to test the trained model. The features used for training the models include statistical metrics, which capture the central tendency and variability within each sliding window.

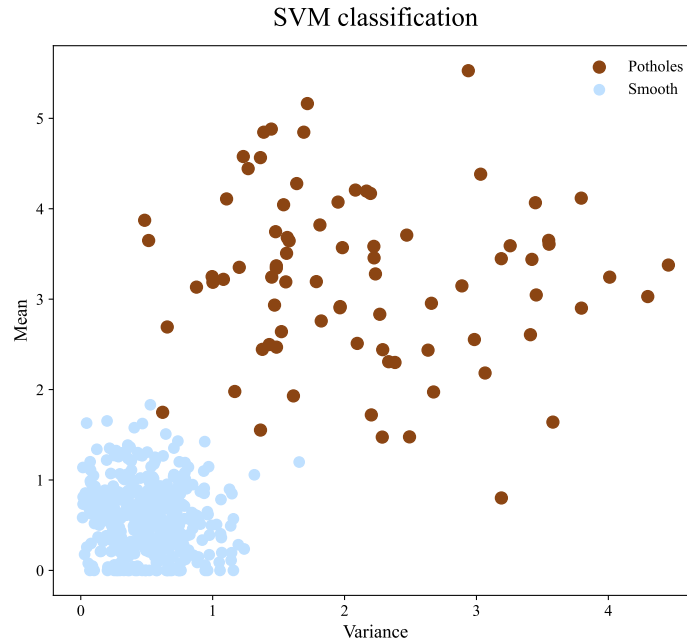


Figure 6.14: Pothole detection using SVM

These features include mean, variance, standard deviation, and coefficient of variation. Additionally, the difference between the first and last points in each window highlights trends, while threshold-based features flag significant deviations by comparing variability and other metrics against predefined thresholds. Based on these features, model parameters for various ML models are optimized to capture underlying patterns and anomalies in the data. This approach allows the models to effectively learn from the statistical properties and threshold-based deviations within each window. The optimized model parameters are provided in Table 6.3.

Table 6.3: Model parameters estimates of various ML models

Model	Parameter	Value
SVM	Kernel	RBF
	Gamma	0.1
	C	1
RF	N_estimators	50
	min_samples_split	10
	min_samples_leaf	2
	Bootstrap	True
Logistic Regression	Solver	Liblinear
	Penalty	l2
	C	10
Naive Bayes	Distribution Smoothing (α)	Gaussian 1.0

For the SVM model, the RBF kernel was chosen, along with a regularization param-

eter of $C = 1$. A smaller value of C often results in a wider margin, producing a less complex model with higher bias but lower variance. Conversely, a larger C would result in a smaller margin, allowing fewer misclassifications but yielding a more complex model with potentially lower bias and higher variance. The value of C was carefully selected based on several trials to optimize performance. Additionally, the gamma value was set to 0.1, which controls the influence of individual data points, with smaller values promoting a more generalized model. Fig. 6.14 shows the classification result using SVM, where the model performs reasonably well even with only two features, mean and variance. More features were utilized in this study, though these are challenging to visualize in three dimensions. For the RF model, 50 estimators (decision trees) were selected to strike a balance between accuracy and computational efficiency. The parameters `min_samples_split` and `min_samples_leaf` were carefully tuned to help prevent overfitting by controlling when nodes are split and ensuring a minimum number of samples in the leaf nodes, promoting model stability. For logistic regression, the liblinear solver was chosen for its efficiency on smaller datasets. A penalty of 12 and a regularization strength of $C = 10$ were selected to control model complexity and mitigate overfitting, ensuring the right balance between bias and variance. Finally, for Naive Bayes, the Gaussian distribution with a smoothing parameter $\alpha = 1.0$ was employed to ensure stable probability estimates, particularly for unseen or rare data points.

Table 6.4: Model Evaluation Metrics for Different Classes and Averaging Methods

Model	Class	Precision	Recall	F1-Score	Accuracy
SVM	1	0.87	0.54	0.67	0.81
	2	0.67	0.92	0.77	
RF	1	0.75	0.462	0.571	0.65
	2	0.61	0.85	0.71	
Logistic Regression	1	0.62	0.38	0.48	0.58
	2	0.56	0.77	0.65	
Naive Bayes	1	0.70	0.54	0.61	0.62
	2	0.63	0.77	0.69	

Once the models are trained, it has been applied to the test data to evaluate their performance. Table 6.4 provides the performance metrics for these models, where class-1 refers to a smooth surface while class-2 refers to potholes. Also, confusion matrices for the four models and their performance in classifying the two categories on the testing dataset are illustrated in Fig. 6.15. The metrics considered include precision, recall, F1-score, and

accuracy. It can be seen from Table 6.4 and Fig. 6.15 that the SVM model resulted in the highest accuracy of 81 percent. It has performed well on class 1 with a precision of 87 percent and a relatively lower recall of 57 percent, which indicates that the model does not capture all instances of class 1. Moreover, the result for class 2 shows a good balance with a precision of 0.67 and a high recall of 0.92.

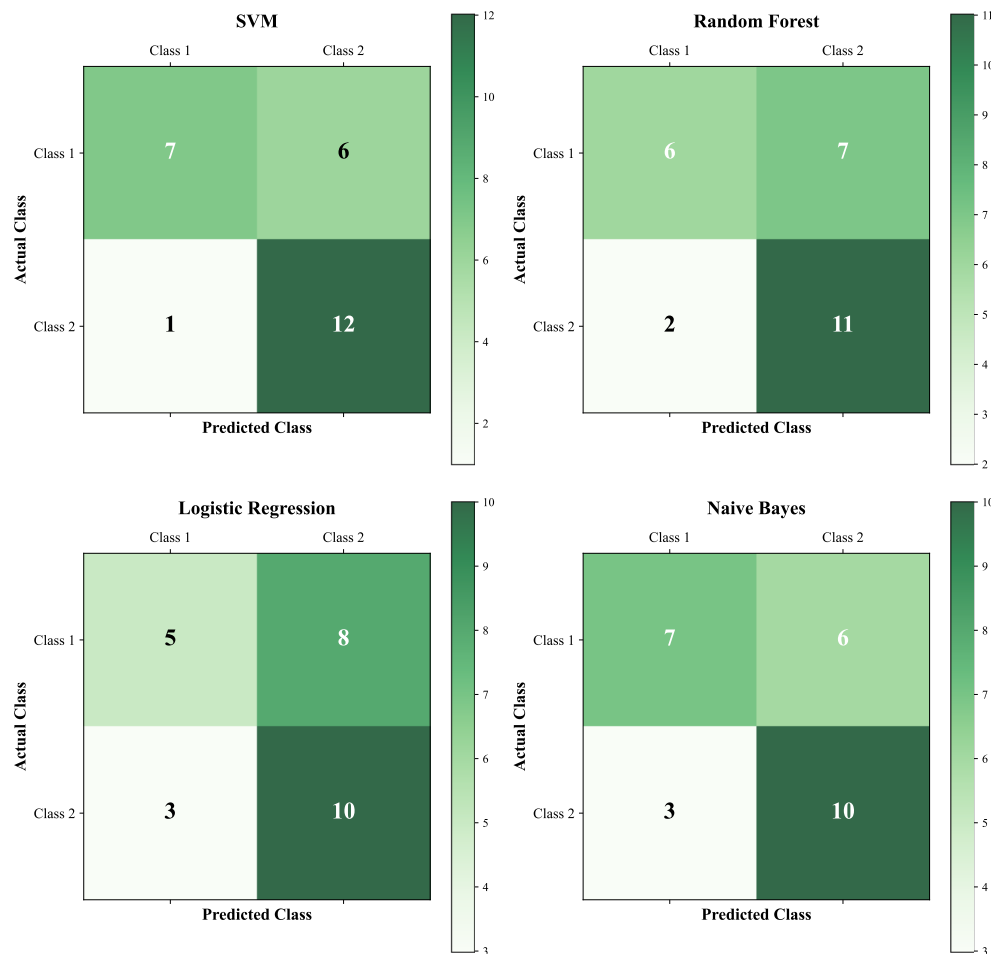


Figure 6.15: Testing dataset confusion matrices for the four models

Similarly, the RF and Naive Bayes models showed moderate performance, with a better ability to identify potholes (class 2) than smooth roads (class 1), but overall lower accuracy compared to SVM. Both models struggled with class 1 predictions, indicating a disparity between the precision and recall for different classes. The logistic regression model, while balanced in performance across both classes, had slightly lower precision and recall, making it less effective overall compared to SVM and RF.

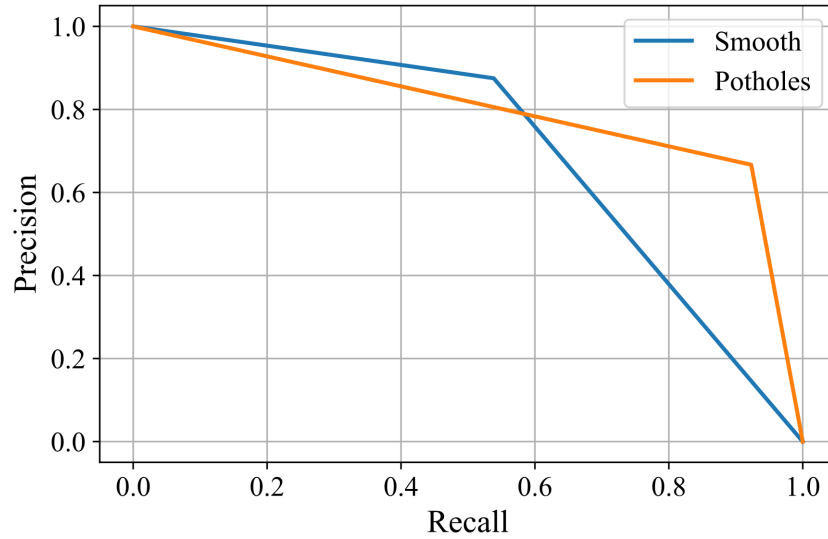


Figure 6.16: Precision-recall curve for SVM classification

The SVM model showed a good balance between precision and recall, performing better than the other models and achieving higher accuracy. This means the SVM was better at correctly classifying both classes, though there were still some errors. However, accuracy by itself doesn't always tell the full story, especially when there's a balance to be found between precision and recall. This is where the precision-recall curve becomes important. It helps us understand how well the SVM handles this balance, showing its strengths and where it might still need improvement. The precision-recall curve for the SVM model is illustrated in Fig. 6.16. By visualizing the relationship between precision and recall at different decision thresholds, one can gain deeper insights into the model's strengths and weaknesses.

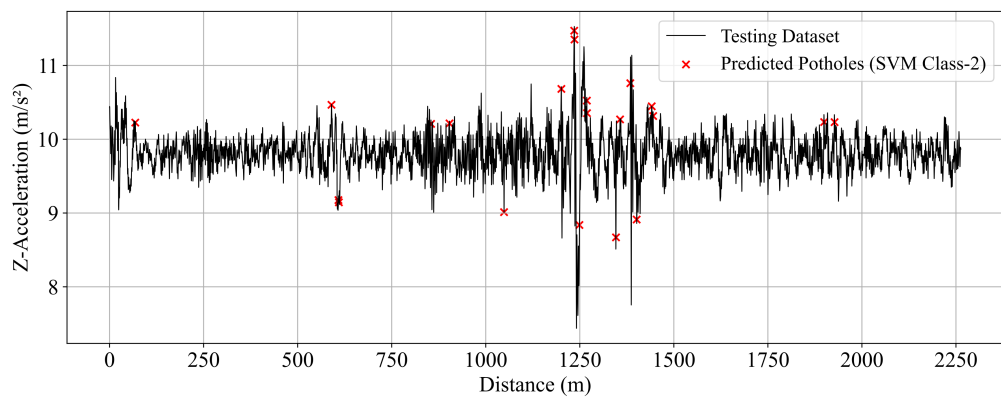


Figure 6.17: Localization using SVM

Since the SVM model achieved the best performance, its predictions on the testing dataset have been plotted against the acceleration data with respect to distance in Fig. 6.17. The red-colored crosses indicate the locations where the model predicted potholes (class-2). This plot is crucial for visualizing the exact positions of potholes along the test route, aiding in the localization and identification of road anomalies for maintenance purposes.

6.3.3 Maintenance cost assessment

The results obtained from previous steps, namely the IRI values and the number of potholes, were used to estimate the approximate maintenance cost for the selected road stretch. Unit costs for various items such as labor wages, material costs, and equipment hiring charges were sourced from the scheduled rates provided by the Central Public Works Department (CPWD), Government of India (see Table 6.8, Appendix). CPWD is the central authority for regulating and updating unit costs for infrastructure works in India. The Indian Roads Congress (IRC) provides additional guidelines and standards for road construction and maintenance. Using these authoritative sources, the maintenance cost per unit length of the road was estimated. The analysis in this study considers a one-way road configuration with a width of 7 m, and the resulting cost estimates are intended as guidance for long-term budgeting after extended periods without repair.

When budgets are limited, prioritizing which roads or segments to improve is essential to maximize the effectiveness of maintenance funds and improve overall service standards. However, determining the most effective allocation of limited maintenance resources is challenging [4]. Factors influencing prioritization include public safety risk, current road condition, usage priority, pavement distress, traffic volume and composition, utility value, local climate, and road type. The selection of predominant factors should be tailored to the socio-economic context and functional performance requirements. Notably, targeting segments with higher IRI values can provide a more comfortable ride quality and reduce pothole-related risks. The methodology for estimating segment-wise IRI-based maintenance cost is described in Table 6.5 of the Appendix. The segment-wise results, including both IRI-based and pothole-based costs, are presented in Table 6.9. The maintenance cost in any J^{th} stretch of road segment mainly depends upon two factors: (i) Quality of road, i.e., IRI value, and (ii) Number and size of potholes. Let C_{1j} and C_{2j} denote the maintenance cost associated with these two factors. Then, the total maintenance cost for all segments

can be given as:

$$\text{Total maintenance cost} = \sum_{\forall J} C_{1J} + \sum_{\forall J} C_{2J} \quad (6.6)$$

The first term C_{1J} can be given as a function of the IRI value and length of the J^{th} road segment, i.e., (IRI_J, l_J) . The second term, i.e., C_{2J} , can be found by multiplying the number of potholes with the average cost to rectify the potholes in the J^{th} segment. Mathematically, it can be given as:

$$\text{Total maintenance cost} = \sum_J f(l_J, \text{IRI}_J) + \sum_J N_J C_J \quad (6.7)$$

Where IRI_J is the roughness index of the J^{th} road segment, and $f(\cdot)$ is the cost function for the J^{th} segment. N_J denotes the number of potholes in segment J , and C_J denotes the average cost to rectify a pothole. The function $f(\cdot)$ can be found by fitting a curve between (l_J, IRI_J) and C_{1J} , obtained from historical data on similar roads. However, in this paper, we assume a linear relation between C_{1J} and l_J , i.e.,

$$C_{1J} = k_J l_J$$

Where k_J is a cost coefficient that reflects the influence of pavement roughness on maintenance costs: however, in practice, the cost function $f(\cdot)$ can be empirically determined using historical maintenance and condition data for greater accuracy. While this model focuses on direct costs associated with surface quality and potholes, future extensions could incorporate indirect costs such as user delays or safety impacts. Specifically, k_J is given as

$$K_J = \alpha \times \text{IRI}_J$$

α is a proportionality constant (with units of INR/meter per IRI unit). The value of α is determined based on local unit cost rates, such as those provided by the Central Public Works Department (CPWD), and encapsulates the average cost of maintenance activities per unit increase in IRI per meter of road segment. With this, equation 6.7 can be re-written as

$$\text{Total maintenance cost} = \sum_J k_J l_J + \sum_J N_J C_J \quad (6.8)$$

The subsequent pilot study section applies these formulations to real-world data, highlighting their utility in maintenance.

Table 6.5: Estimation of segment-wise IRI-based maintenance Cost

Step 1: Calibrating the cost coefficient α

- The cost coefficient α (INR per meter per IRI unit) links the IRI of a road segment to its estimated maintenance cost.
- In this study, α is derived from historical maintenance projects and expert judgment, considering the prevailing unit rates for labor, machinery, and materials (see Table 6.8).
- For practical application, α is taken from Table 6.6.

Step 2: Segment-wise cost calculation

- For each segment J , the segment-specific cost coefficient is:

$$k_J = \alpha \times \text{IRI}_J$$

where IRI_J is the average IRI value for segment J .

- The IRI-based maintenance cost for the segment J is then:

$$C_{1J} = k_J \times L_J$$

Example calculation for segment 1:

- Average IRI for Segment 1: $\text{IRI}_1 = 2.9$ m/km
- Segment length (L_J): 1,000 m
- Based on Table 6.6, this segment falls in the “Fair” category ($2.5 < \text{IRI} \leq 4.0$), with $\alpha = 600$ INR/m/IRI.
- The cost coefficient for the segment is:

$$k_1 = \alpha \times \text{IRI}_1 = 600 \times 2.9 = 1,740 \text{ INR/m}$$

- The IRI-based maintenance cost for this 1 km segment is then:

$$C_{1,1} = k_1 \times L_J = 1,740 \times 1,000 = 1.74 \text{ million INR}$$

Table 6.6: IRI range based maintenance cost coefficient (α) (30 km/hr)

IRI Range (m/km)	Typical Condition	Maintenance Type	α (INR/m/IRI)	Rationale
≤ 8	Good/Fair	Routine/minor	800	Low-cost, light work
$8 < \text{IRI} \leq 12$	Average/Poor	Overlay/patching	1,063	Medium, more materials
> 12	Very Poor	Major re-hab/reconstruction	1,500	Heavy, costly repairs

Table 6.7: Estimation of pothole repair cost

Step 1: Pothole classification and volume estimation

- Potholes are classified into three types based on observed dimensions:
 - Largest: $0.65 \text{ m} \times 0.65 \text{ m} \times 0.06 \text{ m} = 0.0254 \text{ m}^3$
 - Medium: $0.45 \text{ m} \times 0.45 \text{ m} \times 0.025 \text{ m} = 0.0051 \text{ m}^3$
 - Smallest: $0.2 \text{ m} \times 0.2 \text{ m} \times 0.01 \text{ m} = 0.0004 \text{ m}^3$

Step 2: Assign unit rates (from Table 6.8)

- Bitumen: 57,350.44 INR/tonne
- Aggregates: Approx. 1,000 INR/m³
- Labour: Unskilled 310 INR/day, Skilled 442 INR/day
- Machinery: Tipper 779.65 INR/hr, Roller 561.95 INR/hr, HMP 21,092.92 INR/hr

Step 3: Calculate and scale cost per pothole

- For each pothole type, sum the material, labor, and machinery costs according to the estimated volume and required resources.
- Apply a 50% scale factor to account for present-day rates and contingencies:

$$\text{Final Cost} = \text{Subtotal} \times 1.5$$

- For this analysis, the estimated repair costs per pothole are:
 - Largest: 2,250 INR
 - Medium: 1,500 INR
 - Smallest: 750 INR

For localized maintenance estimation, the results from the SVM model were utilized to identify and classify road sections as either potholes or smooth events based on accelera-

tion data. The SVM model performed binary classification of smooth and pothole sections. Subsequently, the potholes detected by the model were manually grouped into three representative types—largest (L): 0.65 m × 0.06 m, medium (M): 0.45 m × 0.025 m, and smallest (S): 0.2 m × 0.01 m—using ground survey data. Since most potholes were irregular, a square-shaped area was considered for repair estimation. The approach for pothole repair cost calculation is summarized in Table 6.7.

Table 6.8: Cost details of the material used, equipment and labour

Sl. No.	Description	Unit	Unit Price (INR)
1	Labour		
	Mate	Day	354.00
	Unskilled Labour	Day	310.00
	Skilled Labour	Day	442.00
2	Machinery		
	Air compressor 250 cfm	Hour	575.22
	HMP 100-110 TPH Capacity	Hour	21092.92
	Tipper 10 tonnes capacity	Hour	779.65
	Smooth wheeled roller 8-10 tonnes	Hour	561.95
3	Material		
	Bitumen	Tonne	57350.44
	Bitumen emulsion for tack coat (rate of spray 0.20 Kg/sq. m)	Tonne	47120.35
4	Aggregates [Grading I - 19mm (Nominal size)]		
	20-10mm 35 per cent	m ³	1840.00
	10-5 mm 23 per cent	m ³	74.78
	5mm and below 40 per cent	m ³	863.00
5	Safety signs		4533.85

Table 6.9: Kilometre-wise IRI-based and pothole-based maintenance cost estimation (in million INR)

Per km	Avg. IRI (m/km)	k_J (INR/m)	C_{1J}	L	M	S	C_{2J}	Total Cost
1	3.9	2,340	2.340	0	1	1	0.00225	2.342
2	3.6	2,160	2.160	1	1	2	0.00450	2.165
3	2.8	1,680	1.680	0	0	1	0.00075	1.681
4	4.4	2,640	2.640	1	1	2	0.00450	2.645
5	3.1	1,860	1.860	2	0	2	0.00600	1.866
6	2.7	1,620	1.620	1	0	1	0.00225	1.622
7	3.4	2,040	2.040	2	1	3	0.00750	2.048
8	2.9	1,740	1.740	2	0	2	0.00600	1.746
9	3.0	1,800	1.800	3	1	3	0.01050	1.811
10	3.5	2,100	2.100	2	1	2	0.00750	2.108
11	4.8	2,880	2.880	3	2	3	0.01200	2.892
12	5.2	3,120	3.120	3	2	2	0.01125	3.131
13	6.9	4,140	4.140	4	3	4	0.01575	4.156
Total	–	–	30.960	–	–	–	0.09075	31.051

Pothole repair costs are much lower than IRI-based segment maintenance costs because pothole repairs are highly localized, addressing only specific, small defects with limited material and labor, while IRI-based maintenance involves large-scale

interventions such as overlays or rehabilitation across the entire road segment, resulting in significantly higher costs. However, even though the cost of pothole repair is relatively minor, it is essential to estimate and include these costs because IRI-based maintenance alone cannot fully address all maintenance needs. Potholes often have considerable depth and can extend into underlying pavement layers, requiring specialized repair methods that go beyond surface treatments. If left undressed, potholes can lead to rapid deterioration, safety hazards, and higher future expenses.

6.4 Model 2: Multi-class classification of road damages

Binary classification effectively distinguishes between damaged and undamaged pavement sections but lacks the ability to identify specific types of damage. However, for practical maintenance planning and resource allocation, identifying the nature of the damage—whether it is a pothole, rutting, or crack—is equally crucial.

This limitation of binary classification motivates the adoption of a multi-class classification framework. As detailed in the section 6.2, time-series vibration signals were acquired using a dashboard-mounted uniaxial accelerometer during field surveys. These vibration patterns inherently vary across different damage types, offering a rich dataset for multi-class classification. Fig. 6.18 illustrates the proposed multi-class classification pipeline, which builds upon the binary detection approach by enabling the identification of multiple types of road surface anomalies. It has been observed that the occurrence of road damages is non-uniform, and their lengths vary significantly. Consequently, relying on a single reading of acceleration data may be insufficient to capture the full characteristics of these anomalies. Therefore, a windowed approach is adopted, where features are extracted from segments of acceleration data over a defined time frame to provide contextual information for accurate classification.

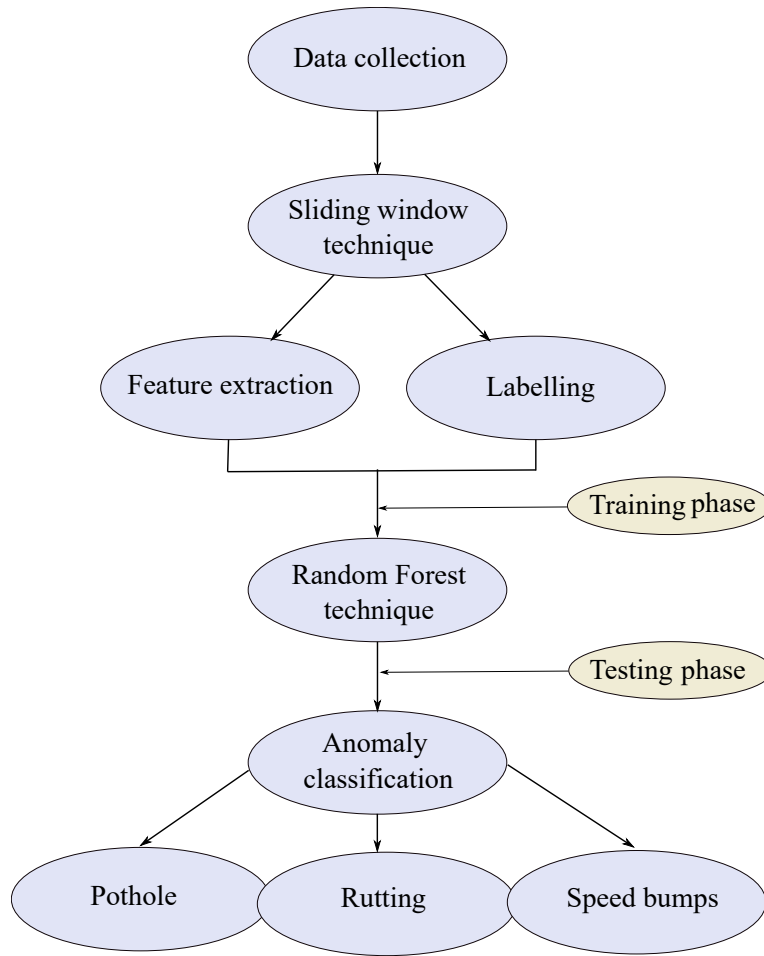


Figure 6.18: A flowchart detailing the sequential steps for the process of multiple road anomaly classification.

6.4.1 Data processing and feature extraction

The occurrence of road anomalies is often non-uniform, with varying lengths and spatial extents. Consequently, a single instance of vibration data may be insufficient for accurately detecting and classifying such anomalies, necessitating the use of time-series acceleration data to capture the dynamic behavior of the vehicle-road interaction over a continuous segment. To address this issue, a sliding window technique has been implemented, as illustrated in Fig. 6.19. For this technique, selecting an optimal window is crucial for the effective extraction of features. The process of determining the optimal window size is detailed below. The working flow of the sliding window technique is as follows: Firstly, it starts with the initial set of observations from the collected data and extracts all the proposed features. Then, it moves to the next set of observations with 50% overlap and extract the

features. Likewise, it converts acceleration observations into windows along with their corresponding labels. During this process, the total observations may not be exactly divide into same window size i.e., some of the observations will not fill the whole window size. In this case, these observations are concatenated with smooth condition acceleration signals ($g = 9.81\text{m/s}^2$) and then features have been extracted. Observations in the window are represented by 'r.'

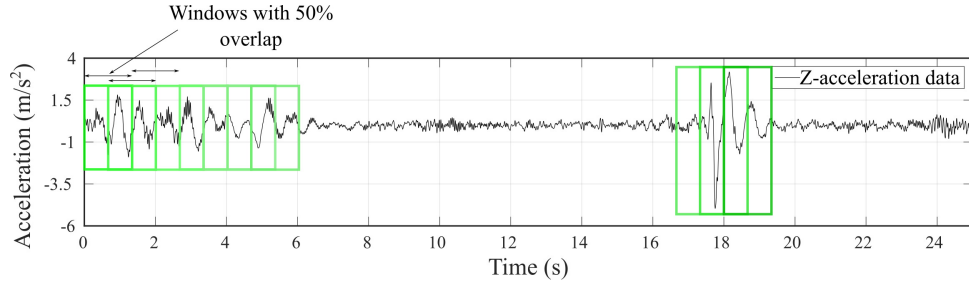


Figure 6.19: Feature extraction for classification task

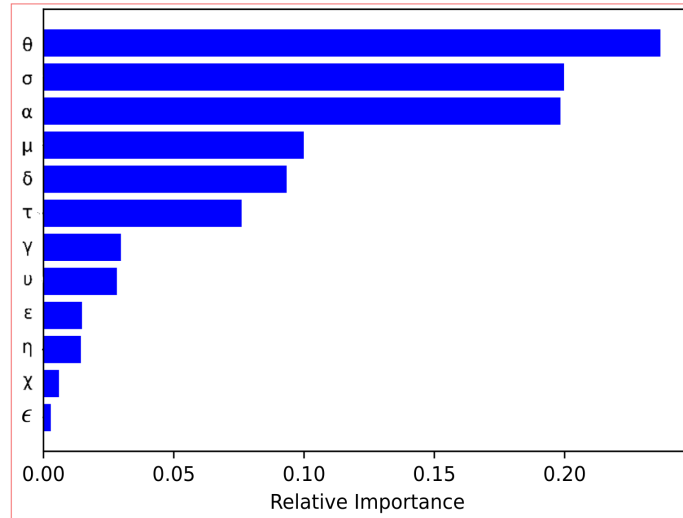


Figure 6.20: Importance or weight of various features.

It has been necessary to assign features for the acceleration data after segmentation in windows for training the RF model. A total of 12 features were used in this study and the same has been presented in Table 6.10. Five statistical features and seven non-statistical features were calculated. For non-statistical features the threshold limits was set by conducting experiments on roads. The weights/importance of different features of the trained model for predicting road

damages are shown in Fig. 6.20. Other features like median, peak value, skewness (for the randomness of the distribution), percentiles, quartiles, and outliers are available. The problem is that these features only consider some of the observations of the window, and most only use one observation, which is ineffective for detecting damages. These features also affect the performance of the model by increasing its dimensionality.

Table 6.10: Features used in field study.

Type	Feature	Calculation
Statistical features	Mean (μ)	$\frac{\sum_{n=1}^{30} r_n}{n}$
	Standard deviation (σ)	$\frac{\sum_{n=1}^{30} (r_i - \mu)^2}{n - 1}$
	Variance (θ)	$(\sigma)^2$
	Coefficient of variation (α)	$\frac{\sigma}{\mu}$
	Difference (δ)	Difference of maximum and minimum values in each window.
Non-Statistical features	Capability potential (β)	Threshold value = $g \pm 0.3g$, if observed acceleration is $>$ threshold $\beta = 0.6$ else 0.3.
	Threshold for standard deviation (τ)	If $\sigma > 0.15g$, then the value is 0.9 else 0.2.
	Threshold for variance (ν)	If $\theta > 0.15g$, then the value is 0.8 else 0.2.
	Derived feature1 (η)	ν , α , τ , and β . If any three of these exceed their limits, then the value of η is 0.8 else 0.2.
	Threshold for coefficient of variation (χ)	If $\alpha > 0.015$, then the value of χ is 0.8 else 0.2.
	Derived feature2 (ε)	$\nu + \tau + \chi + \beta + \eta$
	Derived feature3 (γ)	If $\varepsilon \geq 2$, then $\gamma = 0.8$ else 0.2.

Fig. 6.21 presents a typical plot of features such as variance, standard deviation, coefficient of variance, and mean, illustrating the change in feature values. Training data has been labelled in the initial stage using the recorded video data. Now, the extracted features and the corresponding labels were used to train the RF classifier.

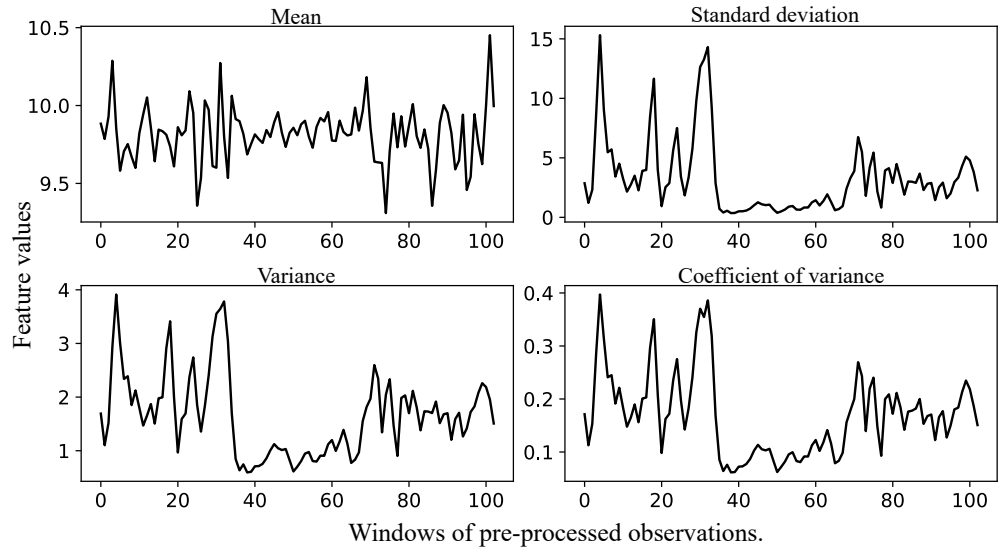


Figure 6.21: Important features used in the field study

6.4.1.1 Optimal window size selection

The choice of an optimal window size for classification relies on the type and length of road damages. To determine the most effective window size, a sensitivity analysis was conducted. A set of windows, ranging from 10 to 1500 observations with intervals of 10, was considered. The analysis started with a window size of 10 observations, each with a 50% overlap, from which the proposed features were extracted. Subsequently, the process continued with window sizes of 20, 30, 40, and so on, up to 1500 observations. The RF model was trained using features from each window size, and the resulting classification outcomes were plotted and visualized in Fig. 6.22. This approach systematically explores various window sizes to identify the one that optimally captures the characteristics of road damages for improved classification accuracy.

Fig. 6.22 has four parts; one is accuracy, others are precision, recall, and f1-score. These three have been calculated separately for each anomaly prediction. Fig. 6.22(a) illustrated about how accuracy has been changing with the window size. At 1250, it has a high accuracy of 83.34% and decreases continuously afterwards. Fig. 6.22(b) and (c) presented how precision and recall has been changing for each anomaly with the window size and Fig. 6.22(d) showed the variation of f1-score for each anomaly. For speed bumps, the precision is '1' at 1250 window size, while for rutting, the precision is '1' within the 920 to 1250 window size range.

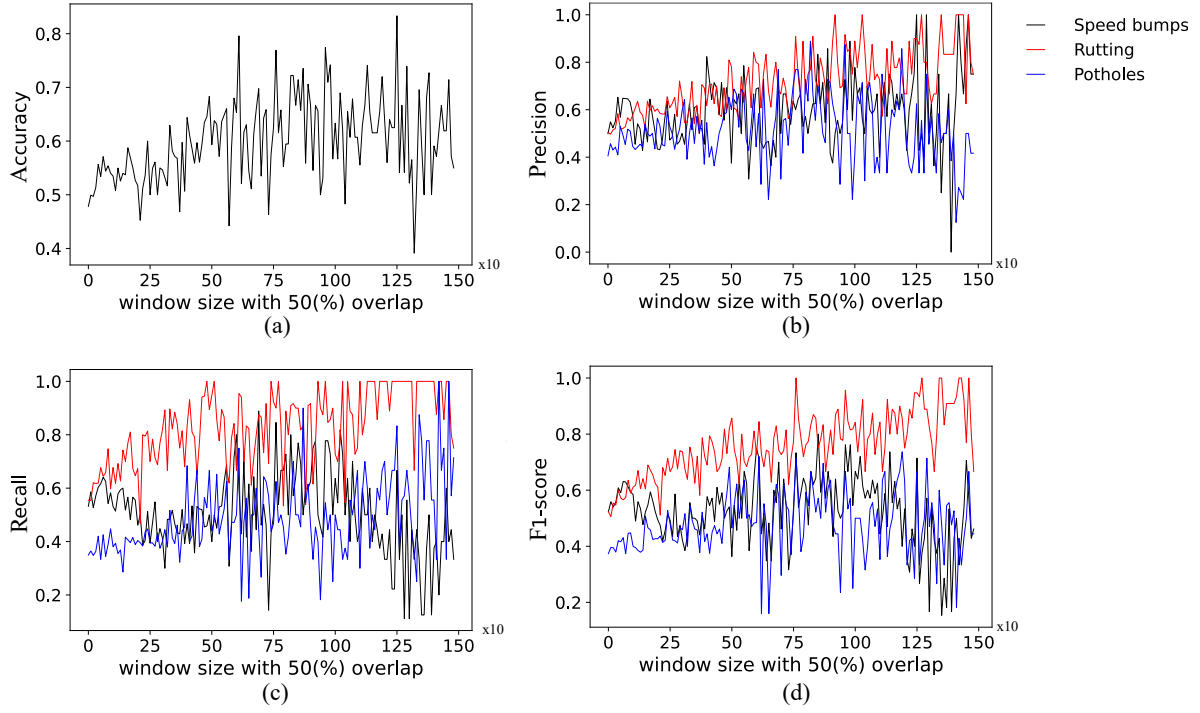


Figure 6.22: Appropriate selection of window size.

However, potholes have a precision of 0.88 at 820 size of the window. Similarly recall values also changing for each class, with respect to window size and showed a value of 1 for class-2 in the range of 1000-1500 window size. Recall value for class-3 showing maximum at 1250-1500 window size whereas for class-1 it was in the range of 50-75 window size. Moreover the maximum f1-score for class 1 & 3, was in the range of 1000-1500 window size. But for class-2, the first peak was at 750 and later has been shown in the range of 1250-1500 window size. After analyzing these plots, the optimal window size for the recorded road damages has been identified as 1250. This specific window size was chosen for the extraction of all proposed features, as detailed in section 6.4.1. Subsequently, utilizing these features and their corresponding labels, the RF classifier was trained.

6.4.2 Model development

The model has subsequently been trained utilizing the extracted features, and testing has been carried out to assess the performance of the model. The relationship between the number of estimators and key performance metrics, including accuracy, precision, F1 score, and recall, is depicted in Fig. 6.23. The figure illustrates the impact of varying the number of estimators on crucial performance metrics,

such as accuracy, precision, F1 score, and recall, with the x-axis representing the number of estimators and the y-axis indicating the corresponding metric scores.

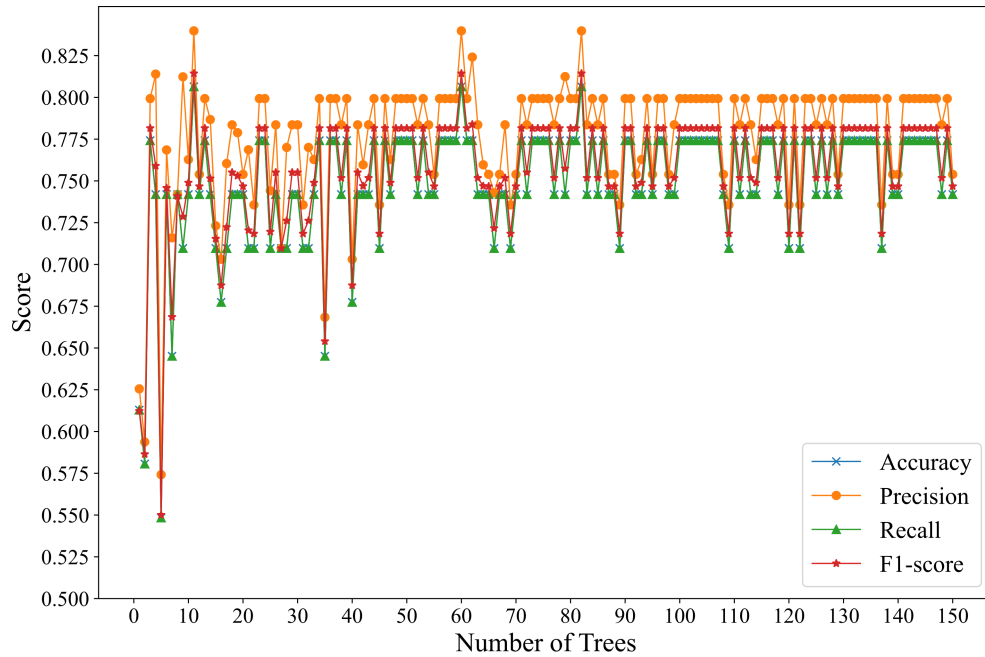


Figure 6.23: Convergence study for classifier metrics

Final model parameters used to train the RF classifier has been presented in Table 6.11. It enumerates the key parameters employed in the model, providing insights into how they influenced the behavior of the RF classifier. The 'Bootstrap' has been set to 'True,' allowing the training data to be sampled with replacement during tree construction. The 'Maximum depth' parameter was set to 80 to determine the depth of individual decision trees. The model relies on 12 features for each split, denoted by the 'Features' parameter. The 'Minimum impurity split' parameter is established at 10, signifying the impurity threshold for node splitting.

Table 6.11: Model parameters

Bootstrap	True
Maximum depth	80
Features	12
'n' estimators	80
'n' jobs'	1
Random state	0
Minimum impurity split	10

After the data collected from the selected road, it was processed, divided into

windows, features have been extracted and labeled. At this point, the total data was converted into feature arrays and the corresponding labels. This data has been split into training data and testing data with a ratio of 0.6 and 0.4. Now the splitted 60% data was used to train the RF classifier using the final model parameters. The RF model has been trained using the features and the corresponding labels and the 40% test dataset has been given to this trained model for the prediction. This test dataset contains a total number of 20 speed-bumps, rutting of 54 and potholes of 24 in numbers originally. Now the classifier predicted the TP's as 12, 54 and 20 for speed-bumps, rutting and potholes respectively, the classification report for each anomaly has been presented in Table 6.12.

Table 6.12: Model performance for classification of various damages.

Class	Precision	Recall	F1-score	Support
1	1	0.68	0.75	20
2	0.86	1	0.93	54
3	0.83	0.82	0.83	24
Accuracy			0.85	98
Macro avg.	0.89	0.87	0.84	98
Weighted avg.	0.88	0.89	0.86	98

Class 1, representing speed bumps, demonstrates exceptional precision (1.00), yet a recall of 0.68 suggests some instances were not accurately identified. The F1-score, a nuanced measure balancing precision and recall, stands at 0.75. Class 2, corresponding to rutting, showcases high precision (0.86) and perfect recall (1.00), resulting in an impressive F1-score of 0.93. Class 3, denoting potholes, displays commendable precision (0.83) and recall (0.82), yielding an F1-score of 0.83.

The performance of the RF classifier experienced a 12% decrease during the transition from a simulated study to a real-world field study. In our simulation, we endeavored to replicate road damages with consistent acceleration patterns, assuming damages of the same length and pattern throughout the road. However, the real-world scenario brought variations because the lengths of road damages differed, impacting how long it took for vehicles to pass each type. Several factors contributed to the differences between simulation and reality. Mechanical factors, like variations in sensor sensitivity despite using better accelerometers, played a

role. Vehicle vibrations also added complexity to real-world conditions. Environmental factors, such as sunny daytime conditions during data collection, might have influenced the accuracy of the classifier.

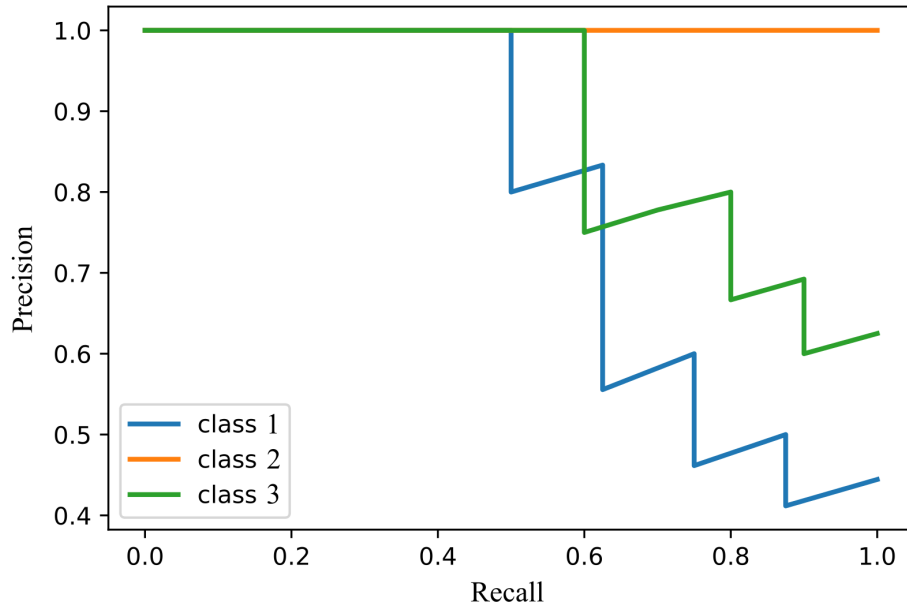


Figure 6.24: Precision-recall curve for the classification.

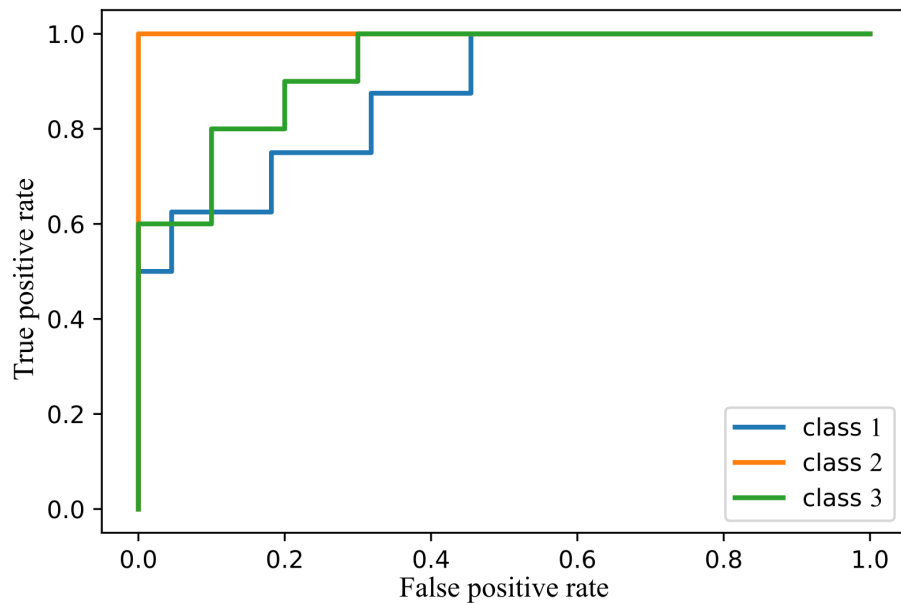


Figure 6.25: ROC curve for the classification.

For a better understanding of the classifier, the precision-recall curve on the test data has been depicted and is shown in Fig. 6.24. The curve illustrates the

relationship between precision and recall of each class at every possible threshold. The cut-off values determine the fraction of true positive or true negative prediction of the model for each class. A model with perfect performance would be able to discriminate between road damage and no damage with 100% recall and 100% precision. Hence, the graph line will pass through the co-ordinates (0, 1) and (1, 1). In the present study, the best performance has been achieved for the class 2 followed by class 3 and class 1.

receiver operating characteristic (ROC) curves for the RF model on the test data is shown in Fig. 6.25 to study the output of a classifier. In this study, the output has been binarized to extend ROC curve for multi-class classification. ROC curves shows the separability of the classes by all possible thresholds i.e., how the model is classifying each class. The perfect performance of the model shows the graph line with co-ordinates (0,0), (0,1) and (1,1).

6.4.3 Comparison of the proposed classifier with other classifiers

The comparison of the proposed RF classifier's performance is detailed in Table 6.13, where its classification metrics are comprehensively compared with other well-established classifiers, such as SVM, KNN, Naive Bayes, and DT. The selection of these classifiers for comparison was based on the complexity of the task, ensuring a meaningful and relevant assessment of performance across various classes. In the evaluation of classification metrics, the proposed RF classifier stands out as a remarkable performer, showcasing its effectiveness in addressing the complex task of classifying various road damages. The classification results reveal the RF classifier's exceptional accuracy, precision, recall, and F1 score across multiple classes, emphasizing its robust predictive capabilities.

When specifically examining the predictions for road damages, where class-1 corresponds to speed bumps, class-2 to rutting, and class-3 to potholes, the RF classifier consistently excels. With an accuracy of 0.87 for both class-1 and class-2, and an accuracy of 0.81 for class-3, the RF model demonstrates its proficiency in distinguishing between different types of road damages. Comparatively, other classifiers like SVM, KNN, Naive Bayes, and DT exhibit varied performance across different

metrics and classes.

Table 6.13: Classification Metrics for Different Classifiers

Classifier	Class	Accuracy	Precision	Recall	F1 Score
RF	1	0.87	1.00	0.68	0.75
	2	0.87	0.86	1.00	0.93
	3	0.81	0.83	0.82	0.83
SVM	1	0.77	1.00	0.36	0.53
	2	0.81	0.72	0.93	0.81
	3	0.71	0.33	0.50	0.40
KNN	1	0.87	0.82	0.82	0.82
	2	0.81	0.79	0.79	0.79
	3	0.81	0.50	0.50	0.50
Naive Bayes	1	0.81	1.00	0.45	0.63
	2	0.81	0.72	0.93	0.81
	3	0.74	0.38	0.50	0.43
DT	1	0.74	0.64	0.64	0.64
	2	0.74	0.75	0.64	0.69
	3	0.74	0.38	0.50	0.43

SVM, for example, achieves high precision but comparatively lower recall for rutting (class-2), indicating potential challenges in capturing all instances of this specific road damage. KNN, while competitive, may encounter issues with precision for class-3, suggesting a tendency for false positives in this category. The Naive Bayes classifier shows competitive results, but its performance is slightly lower compared to RF, particularly in precision and recall for class 1. The DT classifier performs reasonably well, aligning closely with RF in terms of accuracy and precision. However, it shows lower recall and F1 Score for class 2 and class 3. In summary, RF emerges as a robust classifier, showcasing superior performance in multi-class classification for road damages when compared to SVM, KNN, Naive Bayes, and DT.

6.5 Cost estimation for rutting

In this study, rutting, encompassing issues such as depressions, pavement surface uplift, and undulations, has been examined alongside potholes and speed bumps. To address these types of damages, it is crucial to select a specific area around the damage to level the road. Based on acceleration data, rutting damage is more likely to occur over an average length of 1 meter on the chosen road segment. The repair

procedure involves marking the area to be repaired, cleaning it using a mechanical broom, spreading 18 mm aggregates, and compacting them with a vibrating plate compactor. Subsequently, VG 30 is sprayed at a rate of one kilogram per square meter. Following this, 12 mm aggregates are spread, compacted, and resealing binder is sprayed at a rate of 0.4 kilograms per square meter. It is imperative to ensure that the repaired surface is 10 mm higher than the original road level.

Table 6.14: Various unit cost (in INR) used in the cost estimation.

Type	Details	unit cost (/hr or /kg)
Material	Aggregate - 18,12mm	1350 + 103.77 per m^3
	VG 30	39570 per tonne
Labour	4 men	617/-
	2 safety officers at both end of work site	1000/-
	Operator of vibrating plate compactor	738
Equipment	Safety signs	4533.85 per each sign
	Mechanical broom (hydraulic)	360/- per hr
	Vibrating plate compactor	400 per day
	Spraying lances.	300
	Cones, vests, shovels, pickaxes	700/-

Table 6.14 has presented a detailed overview of various unit costs essential for the ongoing cost estimation associated with repairing rutting damage. It systematically categorizes costs into material, labor, and equipment, specifying unit costs in Indian Rupees per hour or per kilogram. The table includes pertinent details such as the cost of 18.12mm aggregate per cubic meter, VG 30 bitumen per tonne, labor wages for 4 men, and the costs of equipment like safety signs, hydraulic mechanical broom per hour, vibrating plate compactor per day, and spraying lances. A detailed breakdown of the estimated costs associated with repairing typical rutting damage has been conducted using these materials, labor, and equipment, and the results are presented in Table 6.15.

Table 6.15: The estimated cost (in INR) for repairing of typical rutting damage.

classification	Items	Details of each item	Cost
Material	A	Material cost	13,912.56/-
Labor	B	Number of men \times wage	5206/-
Equipment	C	Cost of preparation and compaction equipment	10,327.7/-
User delay	D	User delay costs	0
Productivity	E	Average daily productivity	50m ² perday
	F	Estimated days for patching operation	2
Total cost	G	[A + (B+C+D) \times F]	45,179.96/-

The classification includes material costs (A), which encompass expenses related to procuring necessary materials, totaling 13,912.56/-. Labor costs (B) are calculated based on the number of men and their wages, resulting in 5206/-. Equipment costs (C) cover expenses related to the preparation and compaction equipment, amounting to 10,327.7/-. User delay costs (D) are specified as 0, indicating no additional expenses in this category. Productivity factors (E) are considered with an average daily productivity of 50m² per day. In this analysis, the anticipated duration for the patching operation (F) has been predetermined at 2 days. The total cost (G) is meticulously calculated using the formula $[A + (B + C + D) \times F]$, culminating in a conclusive cost estimate of 45,179.96/-.

6.6 Conclusion

This chapter presented a unified framework for comprehensive road condition assessment and maintenance cost estimation by integrating IRI analysis with vibration-based machine learning models. The approach was validated on a 13 km pilot road segment, where IRI values calculated at 50-meter intervals highlighted significant variability in ride quality. Notably, 33.7% of the segments were classified as “poor” (IRI > 10.8 m/km), emphasizing the urgent requirement for targeted maintenance. At the binary classific model achieving an accuracy of 81%, precision of 0.67, and recall of 0.92. These results confirm the potential of sensor-based vibration monitoring as a scalable tool for automated identification of surface irregularities.

Building upon binary defect detection, the framework was extended to multi-class classification of road damages such as potholes, speed bumps, and rutting. A RF-based model, supported by a sliding window segmentation strategy with 50% overlap, was employed for this task. The RF model demonstrated superior performance, achieving high classification accuracy across both simulation (97%) and field validation (85%) scenarios. Comparative evaluation against alternative models—including SVM, KNN, Naive Bayes, and Decision Tree—further confirmed the robustness of the RF approach, which consistently outperformed its counterparts in terms of accuracy, recall, precision, and F1-score. The integration of vibration data with image-based labeling enhanced the reliability of feature extraction and ground truth assignment, strengthening the generalization capacity of the model across different defect types.

The maintenance cost estimation was carried out through a dual-factor strategy that incorporated both segment-level surface roughness and localized damage classification. IRI values were mapped to cost coefficients derived from standard unit rates, while repair costs for potholes and rutting were estimated based on measured dimensions and conventional repair practices. Overall, the chapter demonstrates that transitioning from binary to multi-class classification not only enhances the diagnostic granularity of road condition monitoring but also strengthens the linkage between automated defect detection and actionable maintenance decisions.

Conclusions and Recommendations

The thesis presents a data driven framework for effective damage localization and quantification in transportation infrastructure using vibration-based condition information collected from both bridge components and pavements. Damage detection builds from carefully conditioned vibration signals, distilled into compact, physics-meaningful features, which tuned models convert into actionable outputs—damage presence, location, width and depth. A hybrid MIMO formulation for simultaneous localization and quantification. EEFO-optimized networks extend this framework to include an additional damage characteristic—width, using the same conditioned vibration features and validated through experiment. Extending the same sensing-and-modeling paradigm to pavements, IRI-based ride-quality maps are paired with automated multi-class defect detection, translating measurements into ranked interventions and transparent budgets. In this way, indirect vibrations become reliable indicators of where damage is, how severe it is, and what it will cost to address—enabling maintenance planning even when degradation is not directly observable or operating conditions vary. Most importantly, feasibility has been demonstrated from laboratory beams to real roads. This chapter highlights the significant contributions, key conclusions and recommendations for future work.

7.1 Significant contributions

The significant contributions from this thesis are as follows:

- A vibration-based method converts sensory data into useful dynamic parameters—modal frequencies for beam components and short-window descriptors for roads—and, with AI appropriate models, delivers consistent damage detection, localization, and quantification across bridge components and pavements.
- Developed a hybrid MIMO model that estimates damage location and depth

simultaneously. Building on this, EEFO-optimized DL networks were introduced to add a third attribute—damage width—while retaining the same vibration-derived dynamic parameters.

- An IRI-driven ride-quality assessment combined with a binary defect classifier was established to inform network-level resurfacing and targeted repairs. Subsequently, multi-class damage recognition was developed, and resulting outputs were integrated into maintenance workflows for prioritization and cost estimation.
- Applied the proposed AI-based vibration monitoring core methods to both laboratory-scale bridge beam experiments and real road-network field studies, supporting its potential for integrated, network-wide health management.

7.2 Conclusions

Building on the contributions outlined above, the core conclusions are summarized as follows:

- This thesis demonstrates that a vibration-based workflow—turning sensory data into physics-grounded features and pairing them with task-appropriate AI models—reliably detects, localizes, and quantifies damage across bridge components and roads.
- The first six modal natural frequencies for beam elements, 50%-overlap window features for pavements captured the essential damage signatures, enabling accurate inference with modest sensing and preprocessing.
- For simultaneous localization and quantification of damage, a hybrid MIMO formulation with RF outperformed SVR, LR, and GBM, achieving performance values of 0.995 (location) and 0.936 (depth). On unseen tests after hyperparameter tuning, errors remained within 0.1–9.5% (location) and 8–25% (depth), supporting decision-grade use.
- EEFO-optimized DL networks extended the framework to an additional damage attribute (width). Among five alternatives (FNN, MLP, RNN, LSTM,

1D CNN–GRU–ResNet), LSTM achieved the strongest aggregate fit (MSE 0.378/0.446 train/test; MAE 0.494/0.572), while the 1D CNN–GRU–ResNet matched or exceeded performance for geometric attributes.

- LSTM predicted location with $\leq 1.5\%$ error and width/depth within 8% even for small flaws (0.4 mm width, 1.2 mm depth); the 1D CNN–GRU–ResNet achieved $\leq 2.6\%$ (location) and $\leq 5\%$ (width/depth).
- SHAP analyses indicated that lower-order frequencies (Freq1–Freq2) predominantly inform damage location, whereas higher-order frequencies (Freq3–Freq6) carry more weight for width and depth.
- At network scale, IRI mapped at 50 m resolution exposed heterogeneous ride quality, with 33.7% of segments classified as “poor” —a clear indicator for maintenance prioritization.
- An SVM classifier achieved 81% accuracy (precision 0.67, recall 0.92) on field data for binary pothole detection.
- A RF model distinguished potholes, speed humps, and rutting, validating at 97% (simulation) and 85% (field), and outperforming SVM, KNN, Naïve Bayes, and DT baselines.
- Coupling segment-level IRI–based interventions with localized defect repairs produced a transparent, defensible corridor cost estimate, demonstrating a practical path from sensing to line-item planning for maintenance.

7.3 Recommendations for future work

There are several promising extensions of this research for effective condition-based maintenance of bridge elements and pavements using VBM and AI; key recommendations for future work are outlined below.

- The present work models single-site damage in bridge components; the framework should be extended to handle multiple, interacting damages, with MIMO targets updated accordingly.

- This dissertation shows lower accuracy for damage depth than for damage location, largely due to fewer depth-labelled cases; a broader set of tiny defects and auxiliary ground truth should be added to strengthen severity estimates.
- For pavements, labeling practice relied on manual video stamping, which limits scale; adopting weak/unsupervised pre-labels, active learning, and self-supervised pretraining can cut annotation effort and speed dataset growth.
- Across the experiments of both beams and pavements, acquisition conditions are relatively controlled; a factorial field campaign varying these factors, routes, seasons, and loads would improve robustness.

References

- [1] Herman Garrido, Martin Domizio, Oscar Curadelli, and Daniel Ambrosini. Numerical, statistical and experimental investigation on damage quantification in beams from modal curvature. *Journal of Sound and Vibration*, 485:115591, 2020.
- [2] Lorenzo Manoni, Simone Orcioni, and Massimo Conti. Recent advancements in deep learning techniques for road condition monitoring: A comprehensive review. *IEEE Access*, 12:154271–154293, 2024.
- [3] Chun-Hsing Ho, Matthew Snyder, and Dada Zhang. Application of vehicle-based sensing technology in monitoring vibration response of pavement conditions. *Journal of Transportation Engineering, Part B: Pavements*, 146(3):04020053, 2020.
- [4] Jianxiong Yu, Eddie YJ Chou, and Jyh-Tyng Yau. Development of speed-related ride quality thresholds using international roughness index. *Transportation research record*, 1974(1):47–53, 2006.
- [5] A Alvandi and C Cremona. Assessment of vibration-based damage identification techniques. *Journal of sound and vibration*, 292(1-2):179–202, 2006.
- [6] Siva Sankara Babu Chinka, Srinivasa Rao Putti, and Bala Krishna Adavi. Modal testing and evaluation of cracks on cantilever beam using mode shape curvatures and natural frequencies. In *Structures*, volume 32, pages 1386–1397. Elsevier, 2021.
- [7] Jie Li, Zhenwei Zhang, and Wenzhu Wang. International roughness index and a new solution for its calculation. *Journal of Transportation Engineering, Part B: Pavements*, 144(2):06018002, 2018.
- [8] Yuchuan Du, Chenglong Liu, Difei Wu, and Shengchuan Jiang. Measurement of international roughness index by using z-axis accelerometers and gps. *Mathematical Problems in Engineering*, 2014(1):928980, 2014.

- [9] Oche Alexander Egaji, Gareth Evans, Mark Graham Griffiths, and Gregory Islas. Real-time machine learning-based approach for pothole detection. *Expert Systems with Applications*, 184:115562, 2021.
- [10] Anup Kumar Pandey, Rahat Iqbal, Tomasz Maniak, Charalampos Karyotis, Stephen Akuma, and Vasile Palade. Convolution neural networks for pothole detection of critical road infrastructure. *Computers and Electrical Engineering*, 99:107725, 2022.
- [11] Adrian Komadina, Mislav Martinić, Stjepan Groš, and Željka Mihajlović. Comparing threshold selection methods for network anomaly detection. *IEEE Access*, 12:124943–124973, 2024.
- [12] Joseph Ndong and Kavé Salamatian. Signal processing-based anomaly detection techniques: a comparative analysis. In *Proc. 2011 3rd International Conference on Evolving Internet*, pages 32–39. IARIA, 2011.
- [13] Onur Avci, Osama Abdeljaber, Serkan Kiranyaz, Mohammed Hussein, Moncef Gabbouj, and Daniel 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*, 147:107077, 2021.
- [14] Sanghyun Choi, Sooyong Park, Nam-Hoi Park, and Norris Stubbs. Improved fault quantification for a plate structure. *Journal of sound and vibration*, 297(3-5):865–879, 2006.
- [15] Ricardo Perera, Antonio Ruiz, and Carlos Manzano. An evolutionary multiobjective framework for structural damage localization and quantification. *Engineering Structures*, 29(10):2540–2550, 2007.
- [16] F Parisi, AM Mangini, MP Fanti, and Jose M Adam. Automated location of steel truss bridge damage using machine learning and raw strain sensor data. *Automation in Construction*, 138:104249, 2022.
- [17] Malihe Avarzamani, Majid Ghazali, Morteza Karamooz Mahdiabadi, and Amin Farrokhabadi. Multiple damage detection in sandwich composite

structures with lattice core using regression-based machine learning techniques. *Mechanics Based Design of Structures and Machines*, 53(4):3105–3129, 2025.

- [18] Vikas Khalkar, Arul Marcel Moshi Antony Joseph Decruz, Logesh Kamaraj, Hariharasakthisudhan Ponnarengan, and Renjin J Bright. Damage identification in a cantilever beam using regression and machine learning models. *Iranian Journal of Science and Technology, Transactions of Civil Engineering*, 48(6):4473–4488, 2024.
- [19] Amanda Aryda Silva Rodrigues de Sousa, Jefferson da Silva Coelho, Marcela Rodrigues Machado, and Maciej Dutkiewicz. Multiclass supervised machine learning algorithms applied to damage and assessment using beam dynamic response. *Journal of Vibration Engineering & Technologies*, 11(6):2709–2731, 2023.
- [20] Marek Barski, Piotr Kedziora, Aleksander Muc, and Paweł Romanowicz. Structural health monitoring (shm) methods in machine design and operation. *Archive of Mechanical Engineering*, 61(4), 2014.
- [21] Mayank Mishra, Swarup Kumar Barman, Damodar Maity, and Dipak Kumar Maiti. Ant lion optimisation algorithm for structural damage detection using vibration data. *Journal of Civil Structural Health Monitoring*, 9(1):117–136, 2019.
- [22] K. Sivasubramanian and P. K. Umesha. A graphical method of damage detection and quantification for structures. *Multidiscipline Modeling in Materials and Structures*, 4(2):189–206, 2008.
- [23] Ahmet Can Altunışık, Fatih Yesevi Okur, and Volkan Kahya. Modal parameter identification and vibration based damage detection of a multiple cracked cantilever beam. *Engineering Failure Analysis*, 79:154–170, 2017.
- [24] Jin Zeng, Hui Ma, Wensheng Zhang, and Bangchun Wen. Dynamic characteristic analysis of cracked cantilever beams under different crack types. *Engineering Failure Analysis*, 74:80–94, 2017.

- [25] EI Shifrin and IM Lebedev. Identification of multiple cracks in a beam by natural frequencies. *European Journal of Mechanics-A/Solids*, 84:104076, 2020.
- [26] Fergyanto E Gunawan, Tran Huu Nhan, Muhammad Asrol, Yasuhiro Kanto, Insannul Kamil, et al. A new damage index for structural health monitoring: A comparison of time and frequency domains. *Procedia Computer Science*, 179:930–935, 2021.
- [27] Mustapha Dahak, Nouredine Touat, and Nouredine Benseddiq. On the classification of normalized natural frequencies for damage detection in cantilever beam. *Journal of Sound and Vibration*, 402:70–84, 2017.
- [28] Ganggang Sha, Maciej Radzieński, Maosen Cao, and Wiesław Ostachowicz. A novel method for single and multiple damage detection in beams using relative natural frequency changes. *Mechanical Systems and Signal Processing*, 132:335–352, 2019.
- [29] Seyed Majid Hosseini Pooya and Ali Massumi. A novel damage detection method in beam-like structures based on the relation between modal kinetic energy and modal strain energy and using only damaged structure data. *Journal of Sound and Vibration*, 530:116943, 2022.
- [30] Kaushar H Barad, DS Sharma, and Vishal Vyas. Crack detection in cantilever beam by frequency based method. *procedia engineering*, 51:770–775, 2013.
- [31] AA Lonkar and RK Srivastava. Modeling curvature damage surface for damage detection in cantilever beam. *International Journal of Recent Trends in Engineering*, 1(5):1, 2009.
- [32] K Ravi Prakash Babu, B Raghu Kumar, KL Narayana, and K Mallikarjuna Rao. Multiple crack detection in beams from the differences in curvature mode shapes. *ARPJ Journal of Engineering and Applied Sciences*, 10(4), 2015.

- [33] Gilbert-Rainer Gillich, Nuno MM Maia, Ion-Cornel Mituletu, Zeno-Iosif Praisach, Marius Tufoi, and Ionica Negru. Early structural damage assessment by using an improved frequency evaluation algorithm. *Latin American Journal of Solids and Structures*, 12:2311–2329, 2015.
- [34] F Homaei, S Shojaei, and G Ghodrati Amiri. A direct damage detection method using multiple damage localization index based on mode shapes criterion. *Structural Engineering and Mechanics*, 49(2):183–202, 2014.
- [35] Iftikhar Ahmad, M. Rizwan Siddiqui, Masroor Khan, Noman Iqbal, Sarvat Ahmad, and Mussaver Pervez. Vibration analysis of a multi degree of freedom cantilever beam using theoretical and computational tools. In *Proceedings of the International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, Apr 2016.
- [36] Sandris Rucevskis, Rims Janeliukstis, Pavel Akishin, and Andris Chate. Mode shape-based damage detection in plate structure without baseline data. *Structural Control and Health Monitoring*, 23(9):1180–1193, sep 2016.
- [37] Sandris Ručevskis, Mirosław Wesolowski, and Andris Chate. Vibration-based damage detection in a beam structure with multiple damage locations. *Aviation*, 13(3):61–71, 2009.
- [38] ZA Jassim, NN Ali, F Mustapha, and NA Abdul Jalil. A review on the vibration analysis for a damage occurrence of a cantilever beam. *Engineering Failure Analysis*, 31:442–461, 2013.
- [39] Seyed Omid Sajedi and Xiao Liang. A data-driven framework for near real-time and robust damage diagnosis of building structures. *Structural Control and Health Monitoring*, 27(3):e2488, 2020.
- [40] Rouzbeh Davoudi, Gregory R Miller, and J Nathan Kutz. Data-driven vision-based inspection for reinforced concrete beams and slabs: Quantitative damage and load estimation. *Automation in Construction*, 96:292–309, 2018.

- [41] Arman Malekloo, Ekin Ozer, Mohammad AlHamaydeh, and Mark Girolami. Machine learning and structural health monitoring overview with emerging technology and high-dimensional data source highlights. *Structural Health Monitoring*, 21(4):1906–1955, 2022.
- [42] Ali Zar, Zahoor Hussain, Muhammad Akbar, Timon Rabczuk, Zhibin Lin, Shuang Li, and Bilal Ahmed. Towards vibration-based damage detection of civil engineering structures: overview, challenges, and future prospects. *International Journal of Mechanics and Materials in Design*, 20(3):591–662, 2024.
- [43] Yonghui An, Eleni Chatzi, Sung-Han Sim, Simon Laflamme, Bartłomiej Blachowski, and Jinping Ou. Recent progress and future trends on damage identification methods for bridge structures. *Structural Control and Health Monitoring*, 26(10):e2416, 2019.
- [44] Qinghua Han, Qian Ma, Jie Xu, and Ming Liu. Structural health monitoring research under varying temperature condition: A review. *Journal of Civil Structural Health Monitoring*, 11(1):149–173, 2021.
- [45] Igor Ribeiro, Andreia Meixedo, Diogo Ribeiro, and Túlio Nogueira Bittencourt. Linear and nonlinear time-series methodologies for bridge condition assessment: A literature review. *Advances in Structural Engineering*, 27(13):2204–2227, 2024.
- [46] Zhihang Deng, Minshui Huang, Neng Wan, and Jianwei Zhang. The current development of structural health monitoring for bridges: A review. *Buildings*, 13(6):1360, 2023.
- [47] Indika Wickramasinghe and Harsha Kalutarage. Naive bayes: applications, variations and vulnerabilities: a review of literature with code snippets for implementation. *Soft Computing*, 25(3):2277–2293, 2021.
- [48] Jimmy Rosales Huamani, Jose Ogosi Auqui, Manuel Ego Aguirre Madrid, Jose Castillo Sequera, and Jose Gomez Pulido. A systematic review about the use of machine learning related to earthquake studies. *Advances in Civil Engineering*, 2025(1):4432234, 2025.

- [49] Ayoub Keshmiry, Sahar Hassani, Mohsen Mousavi, and Ulrike Dackermann. Effects of environmental and operational conditions on structural health monitoring and non-destructive testing: A systematic review. *Buildings*, 13(4):918, 2023.
- [50] Sotiris B Kotsiantis, Ioannis D Zaharakis, and Panayiotis E Pintelas. Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, 26(3):159–190, 2006.
- [51] Chenchao, Jun Li, Hong Hao, Ruhua Wang, and Ling Li. Development and application of random forest technique for element level structural damage quantification. *Structural Control and Health Monitoring*, 28(3):e2678, 2021.
- [52] Chuan Li, René-Vinicio Sanchez, Grover Zurita, Mariela Cerrada, Diego Cabrera, and Rafael E Vásquez. Gearbox fault diagnosis based on deep random forest fusion of acoustic and vibratory signals. *Mechanical systems and signal processing*, 76:283–293, 2016.
- [53] Tarek Abedin, SP Koh, Chong Tak Yaw, Chen Chai Phing, Sieh Kiong Tiong, Jian Ding Tan, Kharudin Ali, K Kadirgama, and F Benedict. Vibration signal for bearing fault detection using random forest. In *Journal of Physics: Conference Series*, volume 2467, page 012017. IOP Publishing, 2023.
- [54] Stavroula Tsiapoki, Omid Bahrani, Moritz W Häckell, Jerome P Lynch, and Raimund Rolfes. Combination of damage feature decisions with adaptive boosting for improving the detection performance of a structural health monitoring framework: Validation on an operating wind turbine. *Structural Health Monitoring*, 20(2):637–660, 2021.
- [55] Yujie Ying, James H Garrett Jr, Irving J Oppenheim, Lucio Soibelman, Joel B Harley, Jun Shi, and Yuanwei Jin. Toward data-driven structural health monitoring: Application of machine learning and signal processing to damage detection. *Journal of Computing in Civil Engineering*, 27(6):667–680, 2013.

- [56] Furui Wang and Gangbing Song. Bolt-looseness detection by a new percussion-based method using multifractal analysis and gradient boosting decision tree. *Structural Health Monitoring*, 19(6):2023–2032, 2020.
- [57] Yifu Lan, Youqi Zhang, and Weiwei Lin. Diagnosis algorithms for indirect bridge health monitoring via an optimized adaboost-linear svm. *Engineering Structures*, 275:115239, 2023.
- [58] Duan Yuanfeng, Duan Zhengteng, Zhang Hongmei, et al. Bridge damage identification based on convolutional autoencoders and extreme gradient boosting trees. *Journal of Southeast University (English Edition)*, 40(3), 2024.
- [59] Shengdong Cheng, Juncheng Gao, and Hongning Qi. Determination of the pile drivability using random forest optimized by particle swarm optimization and bayesian optimizer. *CMES-Computer Modeling in Engineering & Sciences*, 141(1), 2024.
- [60] Reshma Ahmed Swarna, Muhammad Minoar Hossain, Mst Rokeya Khatun, Mohammad Motiur Rahman, and Arslan Munir. Concrete crack detection and segregation: a feature fusion, crack isolation, and explainable ai-based approach. *Journal of Imaging*, 10(9):215, 2024.
- [61] Pinakshi Panda, Sukant Kishoro Bisoy, Amrutanshu Panigrahi, and Abhishash Pati. Ant lion and ant colony optimization integrated ensemble machine learning model for effective cancer diagnosis. *International Journal of Electrical & Computer Engineering (2088-8708)*, 15(1), 2025.
- [62] Amir Alhams, Ahmed Abdelhadi, Yousif Badri, Sadok Sassi, and Jamil Renno. Enhanced bearing fault diagnosis through trees ensemble method and feature importance analysis. *Journal of Vibration Engineering & Technologies*, 12(Suppl 1):109–125, 2024.
- [63] Javad Amanabadi, Touraj Taghikhany, and Mohammad Mahdi Alinia. Enhancing structural damage detection: A comprehensive study on feature engineering and hyperparameters for pattern recognition algorithms. *Journal of Vibration and Control*, page 10775463251315863, 2025.

- [64] Maximillian Weil, Carlos Sastre Jurado, Wout Weijtjens, and Christof Devriendt. Machine learning and uncertainty quantification to track and monitor natural frequencies in vibration-based shm applied to offshore wind turbines. *Data-Centric Engineering*, 6:e7, 2025.
- [65] Limin Sun, Zhiqiang Shang, Ye Xia, Sutanu Bhowmick, and Satish Nagarajaiah. Review of bridge structural health monitoring aided by big data and artificial intelligence: From condition assessment to damage detection. *Journal of Structural Engineering*, 146(5):04020073, 2020.
- [66] Sadiq Gbagba, Lorenzo Maccioni, and Franco Concli. Advances in machine learning techniques used in fatigue life prediction of welded structures. *Applied Sciences*, 14(1):398, 2023.
- [67] Partha Sengupta and Subrata Chakraborty. A state-of-the-art review on model reduction and substructuring techniques in finite element model updating for structural health monitoring applications. *Archives of Computational Methods in Engineering*, pages 1–32, 2025.
- [68] Mustafa Qahatan Alsudani, Salah H Abbdal Reflish, Kohbalan Moorthy, and Myasar Mundher Adnan. A new hybrid teaching learning based optimization-extreme learning machine model based intrusion-detection system. *Materials Today: Proceedings*, 80:2701–2705, 2023.
- [69] Donna Xu, Yaxin Shi, Ivor W Tsang, Yew-Soon Ong, Chen Gong, and Xiaobo Shen. Survey on multi-output learning. *IEEE transactions on neural networks and learning systems*, 31(7):2409–2429, 2019.
- [70] Gaston A Fermandois and Billie F Spencer Jr. Model-based framework for multi-axial real-time hybrid simulation testing. *Earthquake Engineering and Engineering Vibration*, 16(4):671–691, 2017.
- [71] Matheus Gomes Correia and Adelino Ferreira. Road asset management and the vehicles of the future: an overview, opportunities, and challenges. *International journal of intelligent transportation systems research*, 21(3):376–393, 2023.

- [72] E Schnebele, Burak F Tanyu, Guido Cervone, and NJETRR Waters. Review of remote sensing methodologies for pavement management and assessment. *European Transport Research Review*, 7(2):7, 2015.
- [73] Minh-Tu Cao, Quoc-Viet Tran, Ngoc-Mai Nguyen, and Kuan-Tsung Chang. Survey on performance of deep learning models for detecting road damages using multiple dashcam image resources. *Advanced Engineering Informatics*, 46:101182, 2020.
- [74] Adeel Ahmed, Moez Ashfaq, Muhammad Uzair Ulhaq, Senthana Mathavan, Khuram Kamal, and Mujib Rahman. Pothole 3d reconstruction with a novel imaging system and structure from motion techniques. *IEEE Transactions on Intelligent Transportation Systems*, 23(5):4685–4694, 2021.
- [75] Qaisar Abbas, Mostafa EA Ibrahim, and M Arfan Jaffar. A comprehensive review of recent advances on deep vision systems. *Artificial Intelligence Review*, 52(1):39–76, 2019.
- [76] Abhiram Karukayil, Christopher Quail, and Fernando Auat Cheein. Deep learning enhanced feature extraction of potholes using vision and lidar data for road maintenance. *IEEE Access*, 12:184541–184549, 2024.
- [77] Ali Faisal and Suliman Gargoum. Cost-effective lidar for pothole detection and quantification using a low-point-density approach. *Automation in Construction*, 172:106006, 2025.
- [78] Bill X Yu and Xinbao Yu. Vibration-based system for pavement condition evaluation. In *Applications of advanced technology in transportation*, pages 183–189. 2006.
- [79] Alessandro Di Graziano, Vincenzo Marchetta, and Salvatore Cafiso. Structural health monitoring of asphalt pavements using smart sensor networks: A comprehensive review. *Journal of Traffic and Transportation Engineering (English Edition)*, 7(5):639–651, 2020.
- [80] PM Harikrishnan and Varun P Gopi. Vehicle vibration signal processing for road surface monitoring. *IEEE Sensors Journal*, 17(16):5192–5197, 2017.

- [81] Mohamed AbdelRaheem, Muhammad Hassan, Mahmoud A Alyousify, Ali Hussein, and Amr A Nassr. Design and implementation of a vibration-based real-time internet of things framework for road condition monitoring. *IEEE Open Journal of Vehicular Technology*, 4:867–876, 2023.
- [82] Kristian Micko, Peter Papcun, and Iveta Zolotova. Review of iot sensor systems used for monitoring the road infrastructure. *Sensors*, 23(9):4469, 2023.
- [83] Amir Shtayat, Sara Moridpour, Berthold Best, and Hussein Daoud. Application of noise-cancelling and smoothing techniques in road pavement vibration monitoring data. *International Journal of Transportation Science and Technology*, 14:110–119, 2024.
- [84] Omar Raed Abuodeh. *Improved Vehicle-Bridge Interaction Modeling and Automation of Bridge System Identification Techniques*. PhD thesis, Clemson University, 2023.
- [85] Stephen J Elliott. A review of active noise and vibration control in road vehicles. 2008.
- [86] Wout Van Hauwermeiren, Joachim David, Luc Dekoninck, Toon De Pessemier, Wout Joseph, Dick Botteldooren, Luc Martens, Karlo Filipan, and Bert De Coensel. Assessing road pavement quality based on opportunistic in-car sound and vibration monitoring. In *Proceedings of the 26th International Congress on Sound and Vibration (ICSV 2019)*, page 8. Canadian Acoustical Association Ottawa, ON, Canada, 2019.
- [87] Zhiguo Zhang. Spectral and time-frequency analysis. In *EEG Signal Processing and feature extraction*, pages 89–116. Springer, 2019.
- [88] Dong Chen, Nengcheng Chen, Xiang Zhang, and Yuhang Guan. Real-time road pothole mapping based on vibration analysis in smart city. *IEEE Journal of selected topics in applied Earth observations and remote sensing*, 15:6972–6984, 2022.

- [89] Vyas Toral, Tandel Krushangi, and R Varia Harishkumar. Automated potholes detection using vibration and vision-based techniques. *World Journal of Advanced Engineering Technology and Sciences*, 10(1):157–176, 2023.
- [90] Shirin Bahaaeldin Hassan. *Calibration of IRI-based PSI Equation using updated quarter-Car parameters*. West Virginia University, 2021.
- [91] Mohammad Saleh Entezari and Amir Golroo. A review on the impacts of connected vehicles on pavement management systems. *International Journal of Pavement Engineering*, 24(1):2246093, 2023.
- [92] Nouredine Chibani, Faouzi Sebbak, Walid Cherifi, and Khadidja Belmesous. Road anomaly detection using a dynamic sliding window technique. *Neural Computing and Applications*, 34(21):19015–19033, 2022.
- [93] R Kothai, N Prabakaran, YV Srinivasa Murthy, Linga Reddy Cenkeramaddi, and Vijay Kakani. Pavement distress detection, classification and analysis using machine learning algorithms: a survey. *IEEE Access*, 2024.
- [94] Chao Wu, Zhen Wang, Simon Hu, Julien Lepine, Xiaoxiang Na, Daniel Ainalis, and Marc Stettler. An automated machine-learning approach for road pothole detection using smartphone sensor data. *Sensors*, 20(19):5564, 2020.
- [95] Khaled R Ahmed. Smart pothole detection using deep learning based on dilated convolution. *Sensors*, 21(24):8406, 2021.
- [96] Kashish Bansal, Kashish Mittal, Gautam Ahuja, Ashima Singh, and Sukhpal Singh Gill. Deepbus: Machine learning based real time pothole detection system for smart transportation using iot. *Internet Technology Letters*, 3(3):e156, 2020.
- [97] Shripad Bhatlawande, Atharva Deshpande, Shreyas Deshpande, and Swati Shilaskar. Proactive detection of pothole and walkable path for safe mobility of visually challenged. In *2022 3rd International Conference for Emerging Technology (INCET)*, pages 1–5. IEEE, 2022.

- [98] Ioan-Daniel Borlea, Radu-Emil Precup, and Alexandra-Bianca Borlea. Improvement of k-means cluster quality by post processing resulted clusters. *Procedia Computer Science*, 199:63–70, 2022.
- [99] Danijela PROTIC and Miomir STANKOVIC. Xor-based detector of different decisions on anomalies in the computer network traffic. *SCIENCE AND TECHNOLOGY*, 26(3-4):323–338, 2023.
- [100] Yong Zhao, Bin Chen, XiangHan Wang, Zhengqiu Zhu, Yiduo Wang, Guangquan Cheng, Rui Wang, Rongxiao Wang, Ming He, and Yu Liu. A deep reinforcement learning based searching method for source localization. *Information Sciences*, 588:67–81, 2022.
- [101] Manjusha Ghadge, Dheeraj Pandey, and Dhananjay Kalbande. Machine learning approach for predicting bumps on road. In *2015 International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 481–485. IEEE, 2015.
- [102] Muhammad Mubarak and Hossam Sallam. The most effective index for pavement management of urban major roads at a network level. *Arabian Journal for Science and Engineering*, 46:4615–4626, 2021.
- [103] Andrew Fox, BVK Vijaya Kumar, Jinzhu Chen, and Fan Bai. Crowdsourcing undersampled vehicular sensor data for pothole detection. In *2015 12th annual IEEE international conference on sensing, communication, and networking (SECON)*, pages 515–523. IEEE, 2015.
- [104] Umang Bhatt, Shouvik Mani, Edgar Xi, and J Zico Kolter. Intelligent pothole detection and road condition assessment. *arXiv preprint arXiv:1710.02595*, 2017.
- [105] Christian Koch and Ioannis Brilakis. Pothole detection in asphalt pavement images. *Advanced engineering informatics*, 25(3):507–515, 2011.
- [106] Hangbin Wu, Lianbi Yao, Zeran Xu, Yayun Li, Xinran Ao, Qichao Chen, Zhengning Li, and Bin Meng. Road pothole extraction and safety evalua-

- tion by integration of point cloud and images derived from mobile mapping sensors. *Advanced Engineering Informatics*, 42:100936, 2019.
- [107] Yangsong Gu, Yuandong Liu, Diyi Liu, Lee D Han, and Xiaoyang Jia. Spatiotemporal kernel density clustering for wide area near real-time pothole detection. *Advanced Engineering Informatics*, 60:102351, 2024.
- [108] Peter Andren. Power spectral density approximations of longitudinal road profiles. *International Journal of Vehicle Design*, 40(1-3):2–14, 2006.
- [109] Yves Delanne and Paulo AA Pereira. Advantages and limits of different road roughness profile signal-processing procedures applied in europe. *Transportation Research Record*, 1764(1):254–259, 2001.
- [110] Injun Song and Gordon F Hayhoe. Airport pavement roughness index relationships using the federal aviation administration (faa) profiling system. In *Airfield and Highway Pavement: Meeting Today's Challenges with Emerging Technologies*, pages 741–752, 2006.
- [111] Michael W. Sayers, Thomas D. Gillespie, and W. D. O. Paterson. Guidelines for conducting and calibrating road roughness measurements. Technical Report UMTRI-85-14, University of Michigan, Ann Arbor, Transportation Research Institute; International Bank for Reconstruction and Development, Washington, D.C., 1986. Report No. 46, Order No. PB86-165263/GAR.
- [112] Michael W Sayers. On the calculation of international roughness index from longitudinal road profile. *Transportation Research Record*, (1501), 1995.
- [113] Nueraihemaitijiang Abulizi, Akira Kawamura, Kazuya Tomiyama, and Shun Fujita. Measuring and evaluating of road roughness conditions with a compact road profiler and arcgis. *Journal of Traffic and Transportation Engineering (English Edition)*, 3(5):398–411, 2016.
- [114] Giuseppe Cantisani and Giuseppe Loprencipe. Road roughness and whole body vibration: Evaluation tools and comfort limits. *Journal of Transportation Engineering*, 136(9):818–826, 2010.

- [115] Antonio Tedeschi and Francesco Benedetto. A real-time automatic pavement crack and pothole recognition system for mobile android-based devices. *Advanced Engineering Informatics*, 32:11–25, 2017.
- [116] Jiale Li, Zhishuai Zhang, Xuefei Wang, and Weixi Yan. Intelligent decision-making model in preventive maintenance of asphalt pavement based on pso-gru neural network. *Advanced Engineering Informatics*, 51:101525, 2022.
- [117] Heriberto Pérez-Acebo, Miren Isasa, Itziar Gurrutxaga, Harkaitz García, and Aimar Insausti. International roughness index (iri) prediction models for freeways. *Transportation research procedia*, 71:292–299, 2023.
- [118] Xin Sun, Honglei Wang, and Shilong Mei. Explainable highway performance degradation prediction model based on lstm. *Advanced Engineering Informatics*, 61:102539, 2024.
- [119] Revanth Dugalam and Guru Prakash. Frequency-based deep learning model for beam damage detection. *European Journal of Environmental and Civil Engineering*, 0(0):1–27, 2025.
- [120] Vahab Toufigh and Iman Ranjbar. Unsupervised deep learning framework for ultrasonic-based distributed damage detection in concrete: integration of a deep auto-encoder and isolation forest for anomaly detection. *Structural Health Monitoring*, 23(3):1313–1333, 2024.
- [121] Kang Yang, Youliang Ding, Huachen Jiang, Yun Zhang, and Zhengbo Zou. Deep learning-based bridge damage identification approach inspired by internal force redistribution effects. *Structural Health Monitoring*, 23(2):714–732, 2024.
- [122] Thang Le-Xuan, Thanh Bui-Tien, and Hoa Tran-Ngoc. A novel approach model design for signal data using 1dcnn combining with lstm and resnet for damaged detection problem. In *Structures*, volume 59, page 105784. Elsevier, 2024.
- [123] Dianelys Vega Ruiz, Cássio Scarpelli Cabral de Bragança, Bernardo Lopes Poncetti, Túlio Nogueira Bittencourt, and Marcos Massao Futai. Vibration-

- based structural damage detection strategy using frfs and machine learning classifiers. In *Structures*, volume 59, page 105753. Elsevier, 2024.
- [124] Neng Wan, Minshui Huang, and Yongzhi Lei. High-efficiency finite element model updating of bridge structure using a novel physics-guided neural network. *International Journal of Structural Stability and Dynamics*, page 2650006, 2024.
- [125] Minshui Huang, Jianwei Zhang, Junliang Hu, Zhongtao Ye, Zhihang Deng, and Neng Wan. Nonlinear modeling of temperature-induced bearing displacement of long-span single-pier rigid frame bridge based on dcnn-lstm. *Case Studies in Thermal Engineering*, 53:103897, 2024.
- [126] Jianwei Zhang, Minshui Huang, Neng Wan, Zhihang Deng, Zhongao He, and Jin Luo. Missing measurement data recovery methods in structural health monitoring: The state, challenges and case study. *Measurement*, page 114528, 2024.
- [127] Lan Nguyen-Ngoc, Quyet Nguyen-Huu, Guido De Roeck, Thanh Bui-Tien, and Magd Abdel-Wahab. Deep neural network and evolved optimization algorithm for damage assessment in a truss bridge. *Mathematics*, 12(15):2300, 2024.
- [128] Minshui Huang, Jianwei Zhang, Jun Li, Zhihang Deng, and Jin Luo. Damage identification of steel bridge based on data augmentation and adaptive optimization neural network. *Structural Health Monitoring*, page 14759217241255042, 2024.
- [129] YiFei Li, Hoang-Le Minh, MaoSen Cao, Xiangdong Qian, and Magd Abdel Wahab. An integrated surrogate model-driven and improved termite life cycle optimizer for damage identification in dams. *Mechanical Systems and Signal Processing*, 208:110986, 2024.
- [130] Arunabha M Roy and Jayabrata Bhaduri. Densesph-yolov5: An automated damage detection model based on densenet and swin-transformer prediction head-enabled yolov5 with attention mechanism. *Advanced Engineering Informatics*, 56:102007, 2023.

- [131] Bo Jiang, Si Chen, Beibei Wang, and Bin Luo. Mglmn: Semi-supervised learning via multiple graph cooperative learning neural networks. *Neural Networks*, 153:204–214, 2022.
- [132] Bao-Loi Dang, Hung Nguyen-Xuan, and Magd Abdel Wahab. An effective approach for varans-vof modelling interactions of wave and perforated breakwater using gradient boosting decision tree algorithm. *Ocean Engineering*, 268:113398, 2023.
- [133] Van-Thien Tran, Trung-Kien Nguyen, H Nguyen-Xuan, and Magd Abdel Wahab. Vibration and buckling optimization of functionally graded porous microplates using bcmo-ann algorithm. *Thin-Walled Structures*, 182:110267, 2023.
- [134] Fuzeng Zhang and Askar Hamdulla. Research on pothole detection method for intelligent driving vehicle. In *2022 3rd International Conference on Pattern Recognition and Machine Learning (PRML)*, pages 124–130. IEEE, 2022.
- [135] Shuai Li, Chenxi Yuan, Donghai Liu, and Hubo Cai. Integrated processing of image and gpr data for automated pothole detection. *Journal of computing in civil engineering*, 30(6):04016015, 2016.
- [136] Yashon O Ouma and M Hahn. Pothole detection on asphalt pavements from 2d-colour pothole images using fuzzy c-means clustering and morphological reconstruction. *Automation in Construction*, 83:196–211, 2017.
- [137] Seung-Ki Ryu, Taehyeong Kim, and Young-Ro Kim. Image-based pothole detection system for its service and road management system. *Mathematical Problems in Engineering*, 2015, 2015.
- [138] Manuel Ricardo Carlos, Mario Ezra Aragón, Luis C González, Hugo Jair Escalante, and Fernando Martínez. Evaluation of detection approaches for road anomalies based on accelerometer readings—addressing who’s who. *IEEE Transactions on Intelligent Transportation Systems*, 19(10):3334–3343, 2018.

- [139] Andrew Fox, BVK Vijaya Kumar, Jinzhu Chen, and Fan Bai. Multi-lane pot-hole detection from crowdsourced undersampled vehicle sensor data. *IEEE Transactions on Mobile Computing*, 16(12):3417–3430, 2017.
- [140] R.G. Lyons. *Understanding Digital Signal Processing*. Pearson Education International, 2011.
- [141] Tom Irvine. An introduction to shock and vibration response spectra. In *Revision E*, pages 157–160. 2023. Available via enDAQ.com.
- [142] Qiqin Yu, Yihai Fang, and Richard Wix. Pavement roughness index estimation and anomaly detection using smartphones. *Automation in construction*, 141:104409, 2022.
- [143] Matthew Byrne, Tony Parry, Riccardo Isola, and Andrew Dawson. Identifying road defect information from smartphones. *Road & Transport Research: A Journal of Australian and New Zealand Research and Practice*, 22(1):39–50, 2013.
- [144] Rajiv Kumar, Abhijit Mukherjee, and VP Singh. Community sensor network for monitoring road roughness using smartphones. *Journal of Computing in Civil Engineering*, 31(3):04016059, 2017.

Appendix : Algorithms

This appendix provides detailed pseudocode of the algorithms developed and used in this research for signal processing, feature extraction, machine learning, and optimization. Each algorithm is presented in a consistent format with defined inputs, outputs, and comments for clarity.

Algorithm A.1: Sliding window method

The sliding window algorithm is used to segment continuous acceleration signals into overlapping windows of fixed size, which serve as the basis for feature extraction. The detailed steps involved in the algorithm are shown in Algorithm 1.

Algorithm 1 Sliding Window Segmentation

```
1: function SLIDING_WINDOW(iterable, size=30, step=1, fillvalue=9.81)
2:   if size < 0 or step < 1 then
3:     raise ValueError
4:   end if
5:   it ← ITER(iterable)
6:   q ← deque(ISLICE(it, size), maxlen=size)
7:   if not q then
8:     return                                ▷ empty iterable or size == 0
9:   end if
10:  q.extend(fillvalue for _ in range(size - len(q)))    ▷ pad to size
11:  while True do
12:    try:
13:      yield ITER(q)                                ▷ iter() to avoid accidental outside
modifications
14:      q.append(NEXT(it))
15:      q.extend(NEXT(it, fillvalue) for _ in range(step - 1))
16:    except StopIteration:
17:      return                                ▷ stop iteration if no more elements in the iterable
18:    end try
19:  end while
20: end function
```

Algorithm A.2: Feature extraction

This algorithm computes statistical and heuristic features from each window of acceleration data. These features are later used as input to machine learning models. The pseudocode describing the step-by-step feature extraction process is given below (Algorithm 2).

Algorithm 2 Extract Features from Windows

```

1: function EXTRACT_FEATURES(zlist, window_size, step_size)
2:   features  $\leftarrow$  []
3:   positions  $\leftarrow$  []
4:   for window in SLIDING_WINDOW(zlist, size=window_size,
   step=step_size) do
5:     mean ( $\mu$ )  $\leftarrow$  CALCULATE_MEAN(window)  $\triangleright \frac{\sum_{n=1}^{30} r_n}{n}$ 
6:     var ( $\theta$ )  $\leftarrow$  CALCULATE_VARIANCE(window)  $\triangleright (\sigma)^2$ 
7:     stdev ( $\sigma$ )  $\leftarrow$  CALCULATE_STANDARD_DEVIATION(window)  $\triangleright \frac{\sum_{n=1}^{30} (r_i - \mu)^2}{n - 1}$ , where r = number of observations in a window
8:     cv ( $\alpha$ )  $\leftarrow$  CALCULATE_COEFFICIENT_VARIATION(mean, stdev)  $\triangleright \frac{\sigma}{\mu}$ 
9:     cp ( $\beta$ )  $\leftarrow$  EVALUATE_CP(window)  $\triangleright$  Threshold value =  $g \pm 0.3g$ , if
       observed acceleration is  $>$  threshold  $\beta = 0.6$  else 0.3.
10:    cvc ( $\tau$ )  $\leftarrow$  EVALUATE_CVC(window)  $\triangleright$  If  $\sigma > 0.15g$ , then the value
       is 0.9 else 0.2.
11:    diff ( $\delta$ )  $\leftarrow$  CALCULATE_DIFFERENCE(window)  $\triangleright$  Difference of
       maximum and minimum values in each window.
12:    threshold ( $v$ )  $\leftarrow$  EVALUATE_THRESHOLD(window)  $\triangleright$  If  $\theta > 0.15g$ ,
       then the value is 0.8 else 0.2.
13:    varc ( $\eta$ )  $\leftarrow$  EVALUATE_VARC(window)  $\triangleright v, \alpha, \tau$ , and  $\beta$ . If any three
       of these exceed their limits, then the value of  $\eta$  is 0.8 else 0.2.
14:    cont ( $\chi$ )  $\leftarrow$  EVALUATE_CONT(window)  $\triangleright$  If  $\alpha > 0.015$ , then the
       value of  $\chi$  is 0.8 else 0.2.
15:    sc ( $\epsilon$ )  $\leftarrow$  CALCULATE_SCORE(varc, threshold, cvc, cp, cont)  $\triangleright$ 
        $v + \tau + \chi + \beta + \eta$ 
16:    cs ( $\gamma$ )  $\leftarrow$  EVALUATE_CS(sc)  $\triangleright$  If  $\epsilon \geq 2$ , then  $\gamma = 0.8$  else 0.2.
17:    features.append([mean, var, stdev, cv, cp, cvc, diff, threshold, varc,
       cont, sc, cs])
18:     $\triangleright$  Positions
19:    positions.append(LEN(window) / 2)
20:   end for
21:   return features, positions
22: end function

```

Algorithm A.3: Random forest classifier

The RF classifier trains and evaluates models for damage detection using extracted features. The workflow of the classifier, including data preparation, training, and evaluation, is shown below in Algorithm 3.

Algorithm 3 Random Forest Classification Workflow

```
1: Initialize empty lists: accuracy, precision, recall, f1score
2: for  $i$  in range(10, 1500, 10) do
3:    $x, Y \leftarrow \text{extract\_features}(\text{acceleration data}, i, \frac{i}{2})$ 
4:    $Yy \leftarrow []$ 
5:   for  $i$  in  $Y$  do
6:      $Yy.append(\text{labels}[\text{int}(i)])$ 
7:   end for
8:    $y \leftarrow \text{np.array}(Yy)$ 
9:    $x_{\text{train}}, x_{\text{test}}, y_{\text{train}}, y_{\text{test}} \leftarrow \text{train\_test\_split}(x, y, \text{random\_state} = 0, \text{test\_size} = 0.2)$ 
10:  Function INITIALIZE_RF_CLASSIFIER():
11:     $\triangleright \dots$  Other initialization steps
12:
13:  function CALCULATE_ENTROPY(probabilities)
14:    return  $-\sum_{i=1}^3 p_i \log_2 p_i$ 
15:  end function
16:
17:   $\text{entropy} \leftarrow \text{CALCULATE\_ENTROPY}([p1, p2, p3])$ 
18:
19:  function CALCULATE_GINI_INDEX(probabilities)
20:    return  $1 - \sum_{i=1}^3 (p_i)^2$ 
21:  end function
22:
23:   $\text{gini\_index} \leftarrow \text{CALCULATE\_GINI\_INDEX}([p1, p2, p3])$ 
24:   $\text{model.fit}(x_{\text{train}}, y_{\text{train}})$ 
25:   $y_{\text{pred}} \leftarrow \text{model.predict}(x_{\text{test}})$ 
26:  Append  $\text{accuracy\_score}(y_{\text{test}}, y_{\text{pred}})$  to accuracy
27:  Append  $\text{precision\_score}(y_{\text{test}}, y_{\text{pred}}, \text{average=None})$  to precision
28:  Append  $\text{recall\_score}(y_{\text{test}}, y_{\text{pred}}, \text{average=None})$  to recall
29:  Append  $\text{f1\_score}(y_{\text{test}}, y_{\text{pred}}, \text{average=None})$  to f1score
30: end for
```

Algorithm A.4: EEFO hyperparameter optimization

The Electric Eel Foraging Optimization (EEFO) algorithm is used for hyperparameter optimization of machine learning and deep learning models. It simulates the foraging behavior of electric eels to balance exploration and exploitation. The sequential steps of the EEFO algorithm are presented below (Algorithm 4).

Algorithm 4 EEFO for Hyperparameter Optimization

```
1: function EEFO(population_size, num_generations, search_space, model_function)
2:   Step 1: Initialize Population
3:   Generate an initial population of hyperparameter sets:
4:   population  $\leftarrow$  RandomUniform(search_spacelb, search_spaceub, (population_size, dimension))
5:   Step 2: Set Initial Best Solution
6:   best_solution  $\leftarrow$  None
7:   best_score  $\leftarrow$   $\infty$ 
8:   for generation = 1 to num_generations do
9:     for i = 1 to population_size do
10:      Step 3: Evaluate Fitness
11:      score  $\leftarrow$  model_function(population[i])
12:      if score < best_score then
13:        Step 4: Update Best Solution
14:        best_score  $\leftarrow$  score
15:        best_solution  $\leftarrow$  population[i]
16:      end if
17:    end for
18:    Store the best score for analysis
19:    best_scores[generation]  $\leftarrow$  best_score
20:    Print ("Generation", generation, "Best Score:", best_score)
21:  end for
22:  Step 5: Return Optimized Hyperparameters
23:  return best_solution, best_score
24: end function
```

Algorithm A.5: Deep learning model

A deep neural network is designed and trained for damage characterization based on frequency response data. The algorithm is summarized in the pseudocode below (Algorithm 5).

Algorithm 5 Deep Learning Model for Damage Characterization

```
1: function DEEPLearningModel(input_data)
2:   Step 1: Data Preparation
3:   Preprocess the input frequency data to ensure it is suitable for training:
4:    $X \leftarrow \text{Preprocess}(\text{input\_data})$ 
5:   Step 2: Model Architecture Design
6:   Design the neural network architecture with layers:
7:    $\text{model} \leftarrow \text{Sequential}()$ 
8:   Add input layer:  $\text{model.add}(\text{InputLayer}(\text{shape} = \text{input\_shape}))$ 
9:   Add hidden layers:
10:  for each  $l_i$  in layers do
11:     $\text{model.add}(\text{Dense}(\text{units} = l_i.\text{units}, \text{activation} = l_i.\text{activation}))$ 
12:  end for
13:  Add output layer:  $\text{model.add}(\text{Dense}(\text{units} = \text{out\_put\_units}, \text{activation} =$ 
     $\text{out\_put\_activation}))$ 
14:  Step 3: Compile the Model
15:  Choose the optimizer, loss function, and metrics:
16:   $\text{model.compile}(\text{optimizer} = 'adam', \text{loss} = 'mse', \text{metrics} = ['mae'])$ 
17:  Step 4: Train the Model
18:  Split the data into training and validation sets:
19:   $X_{\text{train}}, X_{\text{val}}, y_{\text{train}}, y_{\text{val}} \leftarrow \text{train\_test\_split}(X, y, \text{test\_size} = 0.2)$ 
20:  Train the model using the training data:
21:   $\text{history} \leftarrow \text{model.fit}(X_{\text{train}}, y_{\text{train}}, \text{validation\_data} = (X_{\text{val}}, y_{\text{val}}), \text{epochs} =$ 
     $\text{epochs}, \text{batch\_size} = \text{batch\_size})$ 
22:  Step 5: Evaluate the Model
23:  Evaluate the model's performance on the validation set:
24:   $\text{metrics} \leftarrow \text{model.evaluate}(X_{\text{val}}, y_{\text{val}})$ 
25:  Step 6: Make Predictions
26:  Use the trained model to make predictions on new data:
27:   $\text{predictions} \leftarrow \text{model.predict}(\text{new\_data})$ 
28:  Step 7: Return Predicted Damage Characteristics
29:  Output the predicted damage characteristics [location (l), width(w), and
    depth(d)]:
30:  return predictions
31: end function
```
