TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

THESIS NO.: M-102-MSMDE-2024-2026

Vibration Analysis of Euro 6 Diesel Engine: A Multidomain and Machine Learning Approach on Sensor Fault

by

Dipak Sharma

A THESIS

SUBMITTED TO THE DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER IN MECHANICAL SYSTEMS DESIGN AND ENGINEERING

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

LALITPUR, NEPAL

MAY, 2026

COPYRIGHT

TRIBHUVAN UNIVERSITY

INSTITUTE OF ENGINEERING

PULCHOWK CAMPUS

DEPARTMENT OF MECHANICAL AND AEROSPACE ENGINEERING

The undersigned certify that they have read, and recommended to the Institute of Engineering for acceptance, a thesis entitled **"Vibration Analysis of Euro 6 Diesel Engine: A Multidomain and Machine Learning Approach on Sensor Fault"** submitted by Dipak Sharma in partial fulfillment of the requirements for the degree of Master in Mechanical System Design and Engineering.

Supervisor: Laxman Paudel, PhD

Professor

Department of Mechanical and Aerospace Engineering, Pulchowk Campus

---

Supervisor: Surya Prasad Adhikari, PhD

Professor

Department of Mechanical and Aerospace Engineering, Pulchowk Campus

---

External Examiner: Hari Prasad Neupane, PhD

Professor

Department of Mechanical Engineering, Kathmandu University

Committee Chairperson: Sudip Bhattrai, PhD

Head of Department

Department of Mechanical and Aerospace Engineering, Pulchowk Campus

Date: 24 April 2026

Tribhuvan University
Institute of Engineering
Department of Mechanical and Aerospace Engineering
Pulchowk Campus

ABSTRACT

Diesel engines are the primary power sources in the mechanical system domain. Yet, fault diagnosis of these engines has relied on symptom-based inspection, especially in countries like Nepal, which is a practice that is more than often unreliable and reactive and lack predictive maintenance capability. The current vibration-based fault diagnosis literature focuses on mechanical and combustion faults; the vibration response of modern EURO 6 engines to sensor and actuator failures of the Engine Management System has received limited systematic study. This study proposes and validates a vibration-based fault detection and classification technique for eight sensor and actuator-based faults in the ISUZU 4KH1 EURO 6 Common-Rail Direct Ignition (CRDi) engine, which would serve as a foundation for an automated predictive maintenance strategy and Remaining Useful Life (RUL) estimation.

A BeanDevice® triaxial MEMS accelerometer is mounted on the head of the engine to acquire vibration responses wirelessly at 1,000 Hz across nine operating classes: one healthy idle baseline (800-850 RPM) and eight simulated sensor and actuator-based faults namely Accelerator Pedal Position (APP) sensor, Camshaft Position (CMP) sensor, Manifold Absolute Pressure (MAP) sensor, Cylinder number one Injector Circuit, Waste Gate Turbocharger (WGT) Control Valve, Exhaust Gas Regulation (EGR) Valve, Engine Coolant Temperature (ECT) sensor, and Mass Air Flow (MAF) sensor introduced via an intelligent fault-insertion module non-intrusively. A total of 71 sessions were recorded, and vibration data was analyzed on the basis of eighteen features (10 time-domain, 8 frequency-domain) along the 7,202 one-second non-overlapping windows, and a Support Vector Machine (SVM)

along with an RBF kernel was trained and evaluated under three split strategies of increasing robustness.

The vibration signature of the healthy idle condition revealed a dominant engine firing frequency of 27.34 Hz with RMS, kurtosis and crest factor of 1.34 g, 2.11 and 2.21 respectively. The APP sensor fault produced the most distinct signature (RMS = 1.63 g, kurtosis = 6.01, crest factor = 4.66) due to activation of limp-home mode after fault insertion. The SVM achieved 87.32% accuracy under Leave-One-File-Out Cross-Validation (LOFO-CV), with perfect classification (F1=1.00) of the APP, Injector 1, and WGT faults. A fair gap between random-split accuracy (92.51%) and file-aware split accuracy (83.49%) strongly provides empirical evidence of data leakage. Among the split strategy studied, the classification results obtained from LOFO-CV split strategy was found most trustworthy which could be the foundation for the automated vibration-based fault classification and predictive maintenance strategy. The findings of this research established the fact that non-intrusive MEMS-based vibration signal acquisition and multi-domain features extraction along with SVM-based machine learning classification can dependably classify EURO 6 sensor and actuator faults, often missed manual inspection. These findings pave a practical pathway for vibration-aware predictive maintenance of diesel-powered infrastructure.

ACKNOWLEDGMENTS

I am indebted to Prof. Laxman Paudel, PhD, and Prof. Surya Prasad Adhikari, PhD for their invaluable guidance, constructive suggestions throughout my research journey. Additionally, I would like to thank Prof. Hari Prasad Neupane, PhD, for his insightful feedback which helped to refine my report.

I would like to extend by appreciation to Asst. Prof. Sudip Bhattarai, PhD, Head of Department, and entire faculty of Department of Mechanical and Aerospace Engineering, Pulchowk Campus, Lalitpur for providing consistent support, necessary resources and academic environment to carry out this research.

Further, I am grateful for Er. Arun Kumar Katuwal, Chief, and the personnel of the Mechanical Training Center, Department of Road for providing access to the resources for the experimental setup and facilities to carry out this research work.

On a more personal note, I would like to acknowledge my friends and colleagues for their consistent support during this research journey. Finally, I am indebted to my family for their patience and constant belief in me that helped me sustain effortlessly throughout this endeavor.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS

ADC	Analog-to-Digital Converter
AI	Artificial Intelligence
ANN	Artificial Neural Network
APP	Accelerator Pedal Position
BARO	Barometric Pressure
CAN	Controller Area Network
CBM	Condition-Based Maintenance
CKP	Crankshaft Position
CMP	Camshaft Position
CNN	Convolutional Neural Network
CO	Carbon Monoxide
CRDi	Common-Rail Direct Injection
CWD	Choi-Williams Distribution
CWT	Continuous Wavelet Transform
DAQ	Data Acquisition
DEF	Diesel Exhaust Fluid
DOC	Diesel Oxidation Catalyst
DPF	Diesel Particulate Filter
ECM	Engine Control Module
ECT	Engine Coolant Temperature
ECU	Engine Control Unit
EGR	Exhaust Gas Recirculation
EMS	Engine Management System
FBG	Fiber Bragg Grating
FEM	Finite Element Method
FFT	Fast Fourier Transform
FRF	Frequency Response Function

FRP	Fuel Rail Pressure
FT	Fuel Temperature
HC	Hydrocarbons
HOS	Higher-Order Spectra / Higher-Order Statistics
IAT	Intake Air Temperature
ICE	Internal Combustion Engine
IEPE	Integrated Electronics Piezo-Electric
IMF	Intrinsic Mode Function
IoT	Internet of Things
KNN	K-Nearest Neighbor
LOFO-CV	Leave-One-File-Out Cross-Validation
LSTM	Long Short-Term Memory
MAE	Mean Absolute Error
MAF	Mass Air Flow
MAP	Manifold Absolute Pressure
MEMS	Micro-Electro-Mechanical System
MFCC	Mel-Frequency Cepstral Coefficients
MIL	Malfunction Indicator Lamp
ML	Machine Learning
NLAR	Non-Linear Auto-Regressive
NOx	Nitrogen Oxides
OBD	On-Board Diagnostic
OHV	Overhead Valve
PCA	Principal Component Analysis
PSD	Power Spectral Density
RBF	Radial Basis Function
RFR	Random Forest Regression
RMS	Root Mean Square

RMSE	Root Mean Square Error
RPM	Revolutions Per Minute
RUL	Remaining Useful Life
SCR	Selective Catalytic Reduction
STFT	Short-Time Fourier Transform
SVM	Support Vector Machine
SVR	Support Vector Regression
SVS	Service Vehicle Soon
TDC	Top Dead Center
TFR	Time-Frequency Representation
VGT	Variable Geometry Turbocharger
VMD	Variational Mode Decomposition
VQ	Vector Quantization
VSS	Vehicle Speed Sensor
WDCNN	Wide Deep Convolutional Neural Network
WGT	Wastegate Turbocharger
WT	Wavelet Transform
WVD	Wigner-Ville Distribution

CHAPTER 1 INTRODUCTION

1.1 Background

Diesel engines, which convert chemical energy of the fuel to mechanical energy through combustion process are the primary sources of energy for power generation, transportation, and industrial machinery. The ubiquitous application of diesel engines in modern mechanical systems requires its reliable and efficient operation for economic productivity, supply chain consistency and overall safety. However, their complexity involving various reciprocating and rotating components along with severe operating conditions like variable loads, high pressure operation etc. makes them susceptible to frequent failures. The failures of such engines range from bearing wear, fuel injection issues, sensor issues and failures etc. which results in unplanned downtime, high maintenance cost and increased safety risks. The maintenance costs accounts between 15-60% of the manufacturing cost of the final product in typical manufacturing facility, and in heavy industries, these costs are reported to reach almost as high as 50% of total production costs (Romanssini et al., 2023).

When faults occur in the engine, they often leave vibrational and acoustic footprint as a results of pressure variation in cylinders during combustion cycles, motion deterioration of pistons and connecting rods, fluctuations in valve train clearances, crank-connecting rod mechanism, and axial travel gaps between components (Böğrek & Sümbül, 2022). Characteristic vibrations signatures are produced by these mechanical interactions that reflect both normal operating behavior and potential faults. Common diesel engine faults include but not limited to faulty fuel injectors, blown head gaskets, improper air-fuel mixing, valve timing issues and issues with exhaust gas recirculation (Sharma et al., 2014). These faults tend to reduce engine power output, increased fuel consumption, accelerate wear and tear on the engine components and intensify engine emissions and pollutants.

With the advancement of technology, diesel engines have undergone significant advancements over the past decades, integrating electronics, electrical and mechanical subsystems into highly complex machines. As a consequence, fault diagnosis has also become increasingly challenging, requiring high technical skills, multi domain knowledge along with requirements of specialized tools and resources. Automotive maintenance practices in Nepal as well as many developing countries have not evolved

at the same pace as the vehicle technology and often relying heavily on manual inspection, auditory judgement(s), and the experience of individual mechanics which poses a significant challenge in vehicle fault diagnosis and effective repair.

Whilst conventional approaches can be utilized effectively by highly skilled and experienced technicians, the approach as a whole lacks consistency and repeatability. The conventional approach often fails to catch subtle faults and early warning signals like sensor glitches, injector circuit problems and valve malfunctions. This delayed intervention causes small issues to propagate into big failures, causing more damage and repair costs. Also, the conventional approach not backed up by any engineering data and diagnostic tools led to mis interpretations of the faults amongst various mechanics leading to variation in diagnosis and repair decisions. Furthermore, this approach does not support systematic knowledge accumulation, as most diagnostic decisions are not formally recorded or analyzed.

Vibration signals, rich in dynamic information about engine health, offer a non-intrusive method for complex diesel engine fault diagnosis (Nouby M. Ghazaly, 2022). Among the various techniques available for condition monitoring, including acoustic emission, temperature monitoring, oil and debris analysis, and corrosion monitoring, vibration analysis has emerged as the most robust and widely adopted tool for rotating and reciprocating machinery diagnostics (Chu et al., 2024).

Traditional maintenance strategies are classified into breakdown, preventive and predictive strategies. Breakdown maintenance approached often termed as reactive maintenance addresses failure after the breakdown often resulting unplanned downtime and damages to other cascading components and rise on emergency repair costs. Scheduled maintenance performs maintenance interventions at fixed time intervals regardless of in situ equipment condition, leading to unnecessary replacement of serviceable parts and does not cover failures occurring between scheduled inspections. Condition-based maintenance (CBM) and predictive maintenance strategies represents shift from time based to condition-based strategies which help to monitor actual condition of the equipment and detect the degradation that occur before failure. Predictive maintenance has proven to be the most efficient technique as it is based on data collected through continuous monitoring employing modern method like artificial intelligence (AI), machine learning (ML), and the Internet of Things (IoT) (MR & Mangu, 2025).

This research proposes a vibration-based fault diagnosis framework for diesel engine systems by combining advanced MEMS-based vibrational sensing and measurement and multi-domain processing techniques combined with Machine Learning (ML) classification, the framework aims to detect early faults during continuous operating conditions and provide actionable maintenance recommendations. A tri-axial MEMS accelerometer was incorporated to capture vibration signals under baseline (healthy) and various fault conditions from a 4-cylinder, 4- stroke engine. This experimental approach establishes baseline vibration signatures of idle operating conditions, providing reference thresholds against the fault conditions to be compared through statistical characterization. The model would provide foundation for fault severity assessment with Remaining Useful Life (RUL) prediction and maintenance scheduling. For experimental validation, ISUZU 4KH1 Diesel Engine Training Console would be used simulating idle conditions, realistic fault scenarios and measuring vibration signatures for comparison between normal and faulty conditions.

1.2 Statement of Problem

The current fault diagnosis method for diesel engines which are symptom based have significant limitations as they often rely on the inefficient method which largely depends on manual inspection, visible observations such as shaky vibration and odd sounds. Early symptoms are often missed, leading to delayed repair and maintenance and unexpected breakdowns and increased downtime for repairs are also common.

The limitations of the primary diagnostic methods are their inability to assess the changing conditions of the vibration signals. Engines operate at different speeds and loads making vibration signals transient. Due to the transient nature of vibration signals, symptom-based checks struggle to extract clear fault cues from these unstable signals, often confusing normal vibrations with real issues. The reactive approach of the symptom-based checks focuses on the reactive approach and therefore mostly fails to predict the problem beforehand. Additionally, it does not provide clear link of condition monitoring to the predictive maintenance approach required for the desirable optimum operating efficiencies of the engine.

The extent of the symptom-based analysis is often inadequate in early detection capabilities for critical faults like accelerator pedal issues, crankshaft/camshaft sensor

failure modes, manifold pressure sensor faults, cooling temperature sensor faults, injector circuit issues, WGT exhaust control valve faults, EGR valve issues, air flow sensor faults or oil indicator lamp faults. This often leads to escalation of minor issues into major problems resulting in potential significant issues and thereby increasing system downtime and maintenance costs.

With the approach to address these limitations of the traditional symptom-based fault diagnosis strategies, this study proposes a vibration-based fault diagnosis framework with maintenance focus for the ISUZU 4KH1 Diesel Engine as the experimental setup. This framework approach to detect the early warning faults detection through advanced signal processing, classification of the signals based on the fault types and recommend timely predictive maintenance action.

1.3 Objectives of Research

1.3.1 Main objective

- To perform an experimental vibration analysis on diesel engine system at idle and various sensor fault conditions and develop vibration-based fault diagnosis model.

1.3.2 Specific objectives

- To measure the vibrational characteristic of the diesel engine system at idle condition.
- To simulate different fault conditions on the diesel engine system and measure the vibration signature of the diesel engine system.
- To compare the vibration signature of the diesel engine between normal operating condition and simulated sensor-based faulty conditions.
- To develop vibration-based sensor-based fault diagnosis model applicable for predictive maintenance of the diesel engine system.

1.4 Scope of Work

This research focus on developing a vibration-based early fault diagnosis approach for diesel engines, using the ISUZU 4KH1 training console as the test bench emphasizing laboratory-based validation, MEMS- based non-intrusive vibration monitoring, and

integration of vibration signal processing model for fault detection. The outlined was the scope this research work covered

- **Experimental Setup:** The experimental setup consists an ISUZU 4KH1 EURO 6 compliant CRDi diesel engine mounted on a purpose-built educational training bench. This is a 4-cylinder, 4-stroke, inline, turbocharged, direct-injection diesel engine with a displacement of 2.999 liters, producing a maximum output power of 72 kW coupled with triaxial MEMS accelerometers at top of the head block. The training bench is equipped with an intelligent fault-setting module capable of introducing up to ten electronically controlled fault conditions through the engine's ECU, allowing repeatable and non-destructive fault simulation without physical modification to engine components.
- Data Collection: Acquire vibration data under normal baseline (healthy) and faulty conditions, simulated with various sensor failure using Intelligent Fault Insertion Module. All vibration data were acquired with the engine operating at its unloaded idle condition, corresponding to a nominal crankshaft rotational speed of approximately 800-850 RPM
- Signal Processing: Preprocess signals (adaptive thresholding, noise reduction); extract features; compare signatures via time-frequency analysis
- Feature Extraction: Simulate and classify 8 faults in term of vibration signals
- Model Development: Create a fault classification framework early fault diagnosis framework based on vibration input.

1.5 Significance

This research introduces an advanced vibration-based fault diagnosis strategy for sensor and actuator-based faults that will classify and detect faults using SVM method. This would be the foundation for predictive maintenance strategy and optimum maintenance scheduling to reduce unplanned outages and extend engine life. It promotes sustainable operation of the engine system by optimizing resource use.

CHAPTER 2 LITERATURE REVIEW

2.1 Anatomy and working of EURO 6 Diesel Engine System

EURO 6 Diesel Engine system is the result of significant manufacturing and control advancement in automobile industry. Integration of autonomous control of actuator based on different electronic sensor feedback helps on minimizing the exhaust emission keeping the balance on engine efficiency and performance output.

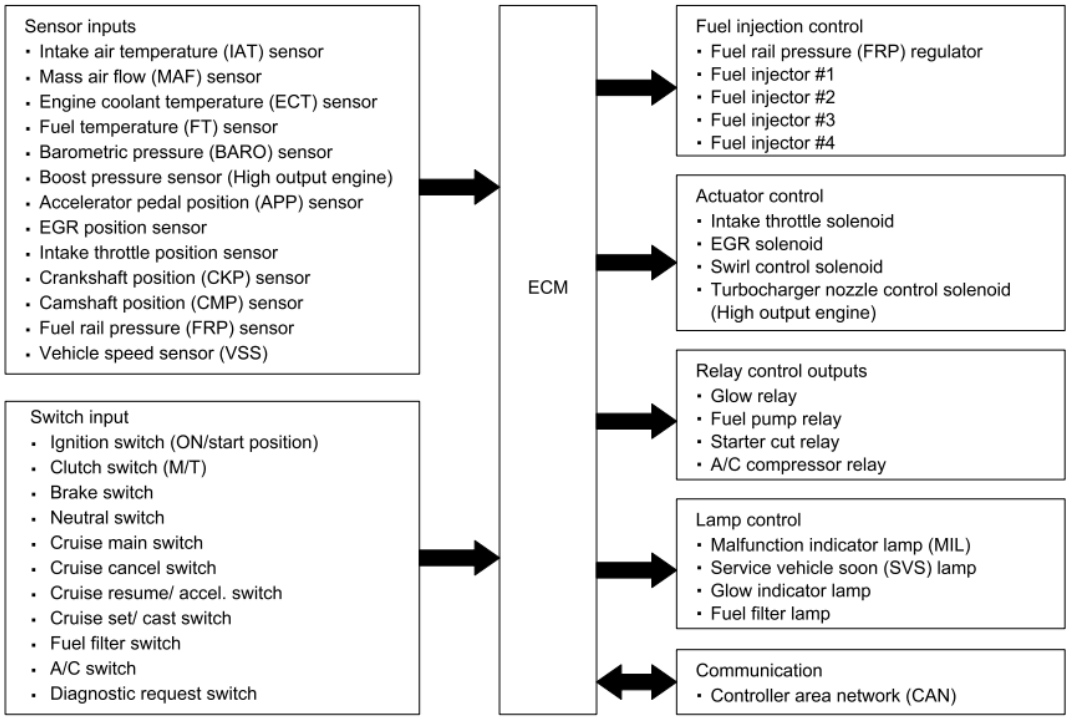

Figure 1 Schematic of Working of Complex Diesel Engine System(*Workshop Manual ISUZU Dmax Engine*, n.d.)

Figure 1 portrays the systematic working of modern EURO 6 compliant diesel engines with input from various switches and sensors are processed by the Engine Control Module (ECM) only to govern control and actuating action of the engine for optimum performance. The ECM performs the management and coordination function of the engine with the help of the inputs from sensors and process the inputs with the prescribed logic only to send the correct control signal to the actuators, relays and other output devices.

Anatomy of the engine is divided into seven sub sections: Sensor Inputs, ECM, Fuel Injection Control, Actuator Control, Relay Control Outputs, Lamp Control, and

Communication interaction of which manifests as a comprehensive engine management system.

2.2 Correlation of Sensor Faults and Vibration

Excessive vibrations in a EURO 6 diesel engine are symbols of loss of engines operation control. In most of the cases this is due to faulty electronic sensors and actuators interrupting the control on sensing and actuation of the ECU. These sensors when in normal working condition provide real-time measurement data to the Engine Control Module (ECM), and their malfunction can lead to the safe mode or misinformed operation of ECU that leads to improper fuel delivery, poor ignition timing, inappropriate air-fuel mixture, inapt exhaust management, all of which can manifest as elevation of vibrations signal on the engine. Key sensor faults whose onset consequences an excessive vibration on engine, along with explanations of how their failures contribute to the issue are summarized on Table 1.(*Workshop Manual ISUZU Dmax Engine*, n.d.)

Table 1 Effect of Sensor Failures to Vibration on the Diesel Engine System

S. N.	Sensor	Role	Fault Impact	Symptoms
1	Crankshaft Position (CKP) Sensor	Monitors crankshaft position for optimum fuel injection and ignition timing.	Mistimed combustion, uneven power delivery, and misfires.	Engine misfires, stalling, starting difficulty, noticeable shaking.
2	Camshaft Position (CMP) Sensor	Monitors camshaft position for optimum valve timing.	Incomplete combustion or cylinder imbalance.	Rough idle, reduced power, vibration worsening at specific RPMs.
3	Mass Air Flow (MAF) Sensor	Measures incoming air to control fuel injection rate accordingly.	Incorrect air-fuel mixture leading to uneven combustion.	Hesitation during acceleration, black smoke, vibration during throttle changes.

S. N.	Sensor	Role	Fault Impact	Symptoms
4	Boost Pressure Sensor (Turbocharged Engines)	Monitors turbocharger pressure to regulate intake air boost.	Over- or under-boosting results in detonation or incomplete combustion.	Whistling noises, power loss, vibration intensifying with engine speed.
5	Engine Coolant Temperature (ECT) Sensor	Monitors engine temperature to adjust fuel and timing.	Inappropriate mixture or timing, causing knocking or incomplete combustion.	Poor cold-start performance, black smoke, vibration decreasing as engine warms.

2.3 Vibration Characteristics of Internal Combustion Engines

The vibration characteristics of an internal combustion engine is intrinsically complex as the engine is excited simultaneously by various internal forces. As recognized by Ahirrao et al., (2018), the primary vibration sources in IC engines are unbalanced reciprocating and rotating masses, periodic combustion pressure variation, reactionary forces from piston-cylinder interaction, and the structural rigidity offered from mounting system. These forces manifest as two principal categories of vibration: torsional vibrations and longitudinal vibrations.

Torsional vibrations are because of the power stroke that drives the crankshaft acceleration while the compression stroke suppress it, creating angular velocity variations in the crankshaft. Tharanga et al., (2020) singularized this mechanism, noting that the piston pressure generates a tangential force at combustion stroke that crankshaft rotational speed, while the compression stroke decreases the engine's angular velocity, causing torsional excitation at the crankshaft. Longitudinal vibrations, by contrast, originate from the unbalanced forces acting on reciprocating and rotating components that propagate outward in three orthogonal directions, with the unbalanced forces at the engine block recorded as structural vibrations along each axis.

The impulsive natures of the internal forces makes it complicated for straightforward application of diagnostic tools on IC engines. Rodríguez-Matienzo, (1998) argued that unlike rotating machines, where diagnostic frequencies are methodically tied to shaft speed harmonics, IC engines generate forces with varying magnitude, making rotating-machinery techniques requiring some modification. The author established through experimental study on a large diesel engine that peaks observed in the spectra range of 2 to 5 kHz with no clear relationship to shaft-speed-based frequencies, and were instead attributable by impulsive combustion and mechanical impact forces. This insight gives the foundational methodological statement that vibroacoustic diagnosis of IC engines should be based on natural frequencies of the engine rather than crankshaft dynamics.

The role of mounting stiffness in vibration transmission to surrounding structures was further explored by Ahirrao et al., (2018), through FFT measurements on a single-cylinder diesel engine operating at 1500 rpm that loosening of foundation bolts raised its velocity RMS vibration levels from 0.73 mm/s to 1.18 mm/s in the radial X-direction, and that bolt tightening expectedly restored vibration to within ISO 2372 acceptable limits while meanwhile reducing sound pressure levels from 100 dB to 82 dB. This finding underscores the diagnostic importance of low-level structural changes that may be indiscernible to subjective assessment but are readily captured by vibration signal measurement via accelerometer.

2.3.1 Common IC engine Faults and Their Vibration Signature

Faults on IC engine are broadly classified into combustion-related and mechanical categories, each leaving exclusive fingerprints in acoustic and vibration spectra due to the engine's physical structure and operational changes (Akbalık et al., 2024).

Combustion-Related Faults

Misfire: This is a common fault caused by ignition failure, lack of fuel amount, or cylinder compression leakage (Chen, 2013). Misfire introduces abrupt decrease in engine speed and fluctuations in crankshaft angular velocity (Karabacak, 2024). On the spectrum note, misfire is associated with amplitude rise of the first few harmonics, particularly the 1st harmonic, which becomes dominant due to the resulting torque imbalance (Chen, 2013).

Engine Knock: Knock occurs when the air-fuel mixture ignites prematurely or unevenly, producing pressure oscillations that can damage internal components

(Karhinen et al., 2026). These vibrations typically excite high-frequency structural resonances, often in bands above 4 kHz and extending up to 12 kHz (Chen, 2013).

Airflow and Ignition Irregularities: Air flow intake or spark plug operation anomalies on the engine create unique spectral signatures. The spark plug issues are most pertinent in the 5.2-5.3 kHz range, while airflow issues are prominently seen in the 3.0-3.1 kHz frequency bin (Akbalık et al., 2024).

Fuel Injection Faults: Faults such as needle opening pressure decline or injector clogs account for over 27% of diesel engine problems which excite cylinder head vibration in a subtle manner than combustion impulses (Tharanga et al., 2020).

Mechanical and Structural Faults

Piston Slap: This fault results from oversized clearances between the piston and cylinder wall. It produces intense impulsive impacts near Top Dead Center (TDC) during the firing stroke (Tharanga et al., 2020). These signals are second-order cyclo-stationary and necessitate envelop analysis and spectral kurtosis to be segregated from lower-frequency combustion broadband noise (Chen, 2013).

Valve Train Faults: Improper valve clearances are detrimental to valve timing and combustion quality which repositions vibration energy to high-frequency regions (>8 kHz). For example, exhaust valve fault significantly increases vibration signal energy in the 6.5-12.5 kHz bands (Jafarian et al., 2018).

Bearing Knock: Big-end or main bearings clearances setting other than manufacturer's recommendation cause high-frequency vibrational resonant responses due to mechanical impacts. These are also second-order cyclo-stationary and are typically most detectable at higher engine speeds (e.g., 3000 RPM) via squared envelope technique (Chen, 2013).

Gearbox Faults: Defects on gear tooth or on gearbox manifest as gear mesh frequency fluctuation and the crankshaft rotational frequency. Faults are identified by increased amplitudes at the dominant gear mesh frequency harmonics and specific peaks in cepstrum analysis (Kn et al., 2021).

Structural Looseness and Imbalance: Unbalance is characterized by a dominant radial peak at 1x engine speed. Misalignment typically produces high axial vibrations at 1x, 2x, and 3x harmonics. Mechanical Looseness (such as loose foundation bolts)

manifests as sub-harmonics (0.5x, 1.5x) and a broad range of run-speed harmonics in the spectrum (Chu et al., 2024)

2.4 Vibration Measurement and Sensor Technologies

Vibration measurement is often used as a non-intrusive diagnostic tool to capture mechanical dynamics and then use it to assess structural integrity and component performance (Romanssini et al., 2023). Measurement parameters are selected as governed by frequency range of interest i.e. displacement used for low-frequency motion (<10 Hz); velocity (10-1000 Hz) effectively tracks energy distribution in rotating masses; and acceleration is preferred for high-frequency phenomena such as bearing impacts.

Primarily vibration signal sensing domain deploy three types of sensor: displacement probes (eddy current based), velocity pickups (electromagnetic induction), and accelerometers (Chu et al., 2024). Among them the most versatile for engine diagnostics are accelerometers. Piezoelectric accelerometers are the standard for high-frequency dependability, offering acceptable linearity and wide bandwidth (up to 30 kHz) (Romanssini et al., 2023). Alternatively, MEMS (Micro-electromechanical systems) sensors provide a cost-effective, compact solution ideal for IoT integration (Varanis et al., 2018). Capacitive MEMS sensors measure acceleration by detecting changes in differential capacitance between fixed structures and a mobile test mass (Romanssini et al., 2023).

Data reliability relies on sampling rates with higher rates associated with better reliability and capturing high-frequency transients. Moreover, perfect and accurate results require sensor placement near vibration sources (e.g., bearing housings) and strong mounting using stud bolts, adhesives, or magnetic bases (Shah et al., 2025).

The quality and significance of acquired vibration signals essentially determines the maximum performance of any subsequent diagnostic model. Several sensor placement strategies and signal conditioning approaches have been documented (Karhinen et al., 2026; Shah et al., 2025; Srinivaas et al., 2025).

Bearing housing or engine block surface in two perpendicular radial directions are the most widely adopted placement for accelerometers. Ahirrao et al., (2018) positioned piezoelectric accelerometers at the bearing housing of a diesel engine in X and Y radial directions, using an FFT analyzer with a measurement range of 200 mm/s RMS velocity

and a frequency range up to 1 kHz for velocity. Karabacak, (2024) tested vibration signals from the engine block via an accelerometer placed near the sixth cylinder of a 6-cylinder diesel engine at a sampling frequency of 24,000 Hz, providing sufficient bandwidth to capture the torsional vibration signatures related with both healthy and faulty operation.

As the number of cylinders increase in the engine, the complexity of the instrumentation also magnifies in large and complex engines. A recent study by Karhinen et al., (2026) demonstrated that conventional monitoring approach would require significant cost due to increased sensor density, wiring complexity and failure risk while studying 20-cylinder gas engine which required up to 40 sensors to detect knock and misfire events. To overcome that complexity, their work proposed use of single flywheel-mounted magnetic encoder as measurement device and recorded angular position of the crankshaft at 50 kHz to extract vibration information for the cylinder-specific fault classification.

An approach of merging experimental Frequency Response Function (FRF) measurements using an impact hammer and a two-channel analyzer with Finite Element Method (FEM) predictions of natural frequencies of engine block by Rodríguez-Matienzo, (1998). The diagnostic frequencies could be detected independently of the crankshaft speed by that study while comparison between FRF peaks and measured vibration spectra highlighting the effectiveness of the methodology valuable for larger engines where direct experimental characterization of all failure modes is impractical.

2.5 Signal Processing Techniques for Vibration Diagnosis

The fundamental analytical challenge in the vibration-based diagnosis is the conversion of raw vibration time series data into diagnostic features. The literature discussed below documents the progression of signal-based processing techniques.

Fast Fourier Transform (FFT) technique is the by far the common and employed on most previous researches. Siano & Panza, (2018) applied FFT on accelerometer signals attained from a lubrication system to identify cavitation-induced spectral components in the range of 7 to 14 kHz, successfully discretizing cavitation related fault from normal operating conditions. However, the authors noted the inherent limitation of the poor frequency resolution linked with short signal segments may produce spectra inadequate to resolve the desired frequencies, motivating the implementation of time-

domain counterparts. Nouby M. Ghazaly, (2022) similarly noted that while classical FFT approaches can recognize established faults, their capability for detecting incipient faults is constrained by this frequency resolution limitation.

Short-Time Fourier Transform (STFT) addresses the non-stationarity of engine vibration signals by implementing Fourier analysis to short windows, producing time-frequency information about signal evolution. Karabacak, (2024) used STFT to create spectrograms from vibration signals of healthy and misfire-affected faulty engines, which were used as two-dimensional input to train Convolutional Neural Networks (CNN). The limitation of STFT is that it provides a fixed time-frequency resolution determined by the chosen window length, which cannot simultaneously achieve high resolution in both domains.

Wavelet Transform (WT) overcomes this limitation by adapting variable-resolution analysis short windows for high-frequency components and long windows for low-frequency components. Nouby M. Ghazaly, (2022) revised the application of Morlet-wavelet-based feature extraction by Jiang et al., (2011), discovering Continuous Wavelet Transform (CWT) approach which adapts the time-accuracy and frequency resolution by controlling the wavelet scale parameter denoising method in removing noise from engine vibration signals with combined spectrum, cepstrum, CWT, and J48 decision tree methods in identifying gear faults under actual combustion operating conditions.

Higher-Order Statistics (HOS) offered sensitivity to non-Gaussian signal components characteristic of fault events and were proposed as complementary tools to FFT for fault detection on engine (Iglesias Martínez et al., 2024). The Wigner-Ville Distribution (WVD) provides high time-frequency resolution but generate cross-term interference in multi-component signals, making its practical applicability to engine diagnostics without prior signal denoising limited (Zhou et al., 2025). Zouani & Hanim, (2016) studied these techniques together with adaptive order-tracking, which compensates for speed variations by resampling the vibration signal in the angular domain rather than the time domain.

Crank-angle-domain analysis has appeared as effective for IC engines because it matches vibration signal processing with the mechanical events of the engine cycle. Tharanga et al., (2020) demonstrated that transforming time-domain vibration signals

to the crank-angle domain and then relating impulse positions against the engine's valve timing and fuel timing enabling direct detection of valve clearance faults and injection irregularities. This approach removes the speed-induced variation that complicates time-domain investigation and aligns the diagnostic reference frame with the physical effect contributing the fault signature.

2.6 Periodic Maintenance and Condition Monitoring

The progression toward preventive maintenance strategies and condition-based predictive maintenance from reactive maintenance signifies the broader engineering context motivating this research. Traditional preventive maintenance scheduled at fixed intervals irrespective of actual component condition often results in either over-servicing or under-servicing. Post-failure driven reactive maintenance often leads to unplanned downtime, collateral damage and associated safety risk.

Aftab Ahmed Soomro, (2025) proposed a comprehensive taxonomy of AI-based predictive maintenance systems to traditional approaches, with real-time data-driven AI systems against the time-based or failure-based triggering of traditional systems providing high accuracy of intelligent pattern recognition against the lower accuracy of manual check or threshold-based methods. The study confirmed that AI regression models deployed to multivariate diesel engine data can forecast fault with sufficient accuracy to complement proactive maintenance scheduling and hence extending component life.

Karabacak, (2024) cited vibration analysis as the promising diagnostic approach in the ICE fault diagnosis literature, suggesting that studies deploying modern ML techniques remain comparatively limited relative to the use of classical approaches, identifying this as an opportunity for research contribution. The research uses publicly available dataset from Randall's vibration-based condition monitoring work provided a reliable experimental foundation for comparing ML methods, an approach dependable with the standardized corroboration methods recommended in the condition monitoring literature landscape.

The incorporation of simulation-based data with real measurement data, as established by Karhinen et al., (2026), reports the fundamental data scarcity problem in engine fault diagnosis. The shortage of characterized real-world fault data arising from the scarcity of fault occurrences, safety restraints on premeditated fault introduction, subsequent

higher cost of meticulous engine testing has been a foremost barrier to the disposition of deep learning models in maintenance forecasting models. The proposed Sim-to-Real transmission learning framework propose a scalable solution, making possible the training of robust fault classification models using ample simulated fault data while using limited real data for domain alignment.

2.7 Machine Learning Approaches

The limitations of manual spectral analysis and threshold-based fault-finding strategy have propelled extensive research into ML methods which are robust and capable of automated fault classification.

Conventional ML algorithms such as Support Vector Machines (SVM), Artificial Neural Networks (ANN), decision trees, and k-nearest neighbor classifiers have been extensively investigated. Karabacak, (2024) mined six statistical features from engine vibration signals; mean, RMS, standard deviation, variance, skewness, and kurtosis as inputs to both ANN and SVM classifiers to differentiate healthy engines from misfire conditions in single to dual cylinder. The ANN realized an overall accuracy of 97.9%, outclassing SVM kernel variants (linear, quadratic, cubic, and Gaussian), each achieved 95.8% accuracy.

Aftab Ahmed Soomro, (2025) employed scientific comparison of Support Vector Regression (SVR), Random Forest Regression (RFR), and ANN models for predictive maintenance of engines, trained on sensor datasets on multiple variables like engine speed, intake manifold pressure, fuel injection rates, exhaust gas temperature, coolant temperature, oil pressure, and vibration signatures. The ANN model was best with RMSE of 2.96, MAE of 2.11 and R^2 of 0.96, while for the considerable accuracy within quicker prediction time, the RFR model predicted at 9.8 ms vs 18.2 ms taken by ANN, making it more suitable for real-time monitoring applications. Feature importance analysis on the RFR model acknowledged crankshaft speed variability, exhaust gas temperature, pressure drop, and vibration amplitude as the most significant predictors of fault onset, boosting its applicability for condition monitoring systems.

Deep learning methods such as Convolutional Neural Networks (CNNs) operating on time-frequency features, have demonstrated enhanced performance in fault classification when patterns are spatially arranged in the time-frequency plane. Karabacak, (2024) pre-trained GoogleNet and ResNet-50 CNN architectures on STFT

spectrograms of healthy and faulty engine vibration signals, achieving 100% validation accuracy analyzing 22 training samples per class for both architectures. This dependably achieved superior test accuracy but required lengthier duration on training, illustrating the tradeoff on accuracy-computational cost front while selecting the model concluding that increasing model complication positively affects accuracy at the expense of training time, and that CNN-based methods outperform classical ANN and SVM methods as training sample is increased.

Karhinen et al., (2026) progressively developed simulation-augmented deep learning framework for fault diagnosis of a 20-cylinder industrial gas engine analyzing torsional vibration data. The framework used a revised Wide Deep Convolutional Neural Network (WDCNN) architecture with a 64-sample first-layer kernel to suppress high-frequency noise while conserving fault-relevant low-frequency torsional signatures. The major methodological novelty was the use of simulated fault data to train the model in the absence of real-world fault data, which are limited for large industrial engines due to operational safety restrictions. Domain randomization, min-max normalization, and semi-supervised regularization with real measurement data were employed to bridge the simulation-to-reality gap. The resulting framework achieved 100% fault detection accuracy and 95.7% fault classification accuracy when assessed on real engine fault data, demonstrating that high-precision fault monitoring is realizable with reduced instrumentation.

2.7.1 Review of Previous Works

The foundational research of Lee & White (1997) introduced higher-order time-frequency representations including Wigner bi- and tri-spectra demonstrating their superiority over second-order methods like WVD for sensing impulsive signals of rotating machinery with performing an experiment on car engine and industrial gearbox. . This established the theoretical support for time-frequency analysis in engine diagnostics. Finally, the researchers suggested extending the approach to more complex systems, non-Gaussian noise environments, and real-time applications.

Geng et al. (2003) afterwards developed an analytic model with software synchronization for non-stationary engine vibration analysis that also include wavelet packet filtering, and pseudo-Wigner-Ville distribution. Vibration analysis was done on reciprocating engine for fault diagnosis, examining impact excitations and time-varying

properties to prepare the model that captures time-varying properties, using vibration signature which successfully diagnosed faults in the 6190 ZLC diesel engine. Finally, the researchers proposed the application of the model across diverse engine types integrating with real-time monitoring systems.

Wu & Chuang (2005) later performed fault diagnosis of internal combustion engines exploring acoustic and vibration signals, with an aid of digital image processing for matching pattern on the signal testing on different component of IC engines such as cooling fans, and drive axles. The research validated the effectiveness even at low rpm where traditional time/frequency methods fail, achieving sufficient accuracy in faults detection under various conditions recommending automated pattern matching with AI integrated ML for classification, and testing on more fault types.

Chao Jin et al. (2014) proposed diesel engine fault diagnosis technique using integrated vibration, cylinder pressure, and rpm signals, along with with wavelet decomposition, discriminant analysis, and classification strategy simulating combustion faults and valve leakage faults in diesel engines. It is found that the integrated diagnosis using combined signals and reduced features is effective, validated on a small-scale test bed. Finally, the researchers proposed real-world health monitoring applications, expanding to more fault types, and integrating with prognostics.

Patil & Raut (2016) performed analysis of IC engines vibration signature using Fast Fourier Transform (FFT) for variation analysis on firing frequency on IC engine faults such as misalignment, unbalance, gear wear, and bearing faults. The research verified the effectiveness of FFT on fault detection through frequency variations and hence suggesting on performance improvement by alleviating excess vibrations. Finally, the researchers recommended using advanced methods such as wavelet and cyclo-stationary analysis laying the foundation for automated real-time fault prediction.

Jafarian & Mobin (2016) used Principal Component Analysis (PCA) to investigate vibration analysis for detecting automobile engines faults via reducing dimensionality and identifying patterns over the signal acquired by four accelerometers by simulating fault via poppet valve clearance altering and misfiring in Over Head Valve (OHV) engines. The research was found superior in efficiency in classifying and detecting fault with better fault categorization by deploying PCA. Finally, the researchers

recommended it to extending to additional faults and potential for automated real-time monitoring supplementing advanced statistical analysis methods.

Ahirrao et al. (2018) measured vibration in engines near resonance by adjusting the stiffness of the engine mounting via tightening of the mounting bolts thereby altering engine supports and looseness effects on structural vibrations. The research found that stiffness factor on bolts reduces vibrations and noise thereby reducing fatigue failure whereas looseness increases vibration severity. Finally, the researchers suggested including damping models, testing on diverse engine types, and integrating with predictive maintenance systems.

Siano & Panza (2018) experimented vibration recording on different pump fault and developed a diagnostic model for detection employing FFT, Non-Linear Auto-Regressive (NLAR) along ANN. In this research of pump cavitation detection via vibration, analogous to engine faults, the detectability of cavitation induced faults with the help of real-time vibration data employing NLAR-ANN was accurately established and the researcher recommended applying the model to engine components like injectors, for multi-fault detection, enhancing the methodology by using multiple sensor.

Zhao et al. (2017) optimized vibration Mel frequency under different operation conditions to develop a novice fault diagnosis method for diesel engines using adaptive correlation threshold, VMD-MFCC features, Vector Quantization (VQ), and K-Nearest Neighbor (KNN). They investigated valve clearance induced engine faults under multiple operating conditions and found over than 90% accuracy for valve faults recommending expanding the model to more faults, real-time employment with deep learning integrations.

Zhang et al. (2019) used Fiber Bragg Grating (FBG) sensors to investigate vibration of an internal combustion engine along with support vector machines algorithm, wavelet preprocessing, vibration energy features concentrating on valve and fuel injection faults detection. The research found high accuracy and superior performance of the method in fault identification recommending incorporation of more sensors, broadening fault types to automate the condition monitoring in harsh operating environments.

Taghizadeh-Alisaraei & Mahdavian (2019) deployed Welch test, STFT, WVD, and CWD for time-frequency representation (TFR) of vibration signal to detect the fuel

injector-based fault propagation via knocking effect on IC. The research found that faulty injectors increasing vibration RMS by 12.9% and kurtosis by 20.6% and recommended automating the detection procedure before deploying for maintenance of systems operating under fluctuating loads.

Tharanga et al. (2020) monitored vibration signal along with the crank angle references to investigated diesel engine fault diagnosis on combustion-related to identify variation on impulsive behavior and later relate it to determine engine condition. Finally, the researchers proposed more studies based on advanced feature extraction and expand the range for complex faults.

Nouby M. Ghazaly (2022) provided a review on fault diagnosis using vibration data analysis, covering methods like STFT, HOS, WVD, WT, and adaptive order-tracking with the focus on faults such as misfire, knock, piston slap. The research effectively demonstrated that these techniques are effective for feature extraction and diagnosis, capable of detecting early defect and providing ground for making decisions to preventing damage recommending on advanced denoising, multiple techniques integration and deployment to modern hybrid engines.

2.8 Research Gaps

Most of the previous works deals on the mechanical related faults such as bearing failures, combustion related faults such as mistimed firing with inadequate scientific investigation of vibration signal characteristic across multiple sensor and actuator based fault types induced progressively which is tried to be addressed by this research by measuring the vibrational response via triaxial MEMS accelerometer for the normal and idle working condition and several sensor and actuator based fault conditions simulated on an EURO 6 diesel engine, analyzing these signal on time domain and frequency domain features and characterizing the features.

While ML methods have established reasonable classification accuracy, their practical employment in real world maintenance environments involves validation of the methods against comprehensive experimented vibration databases on ranges of characterized fault conditions and healthy baseline data. This thesis inventories an experimental dataset acquired from a vibration recording of diesel engine test rig that can serve as a validation resource for the developed fault diagnosis model.

The integration of fault detection model developed into an actionable automated maintenance decision model would provide the maintenance-aware dimensions as suggested by the thesis title to the maintenance literature and is largely under explored in the most of the previous vibration signal-based fault diagnosis studies. This research work bridges this existing gap between fault detection accuracy and subsequent maintenance decision making at early stages of fault onset as informed by vibration-based condition indicator trend.

CHAPTER 3 RESEARCH METHODOLOGY

In this chapter the methodology adopted for vibration-based fault analysis of ISUZU 4KH1 EURO 6 compliant diesel engines is described. The methodology comprises of the experimental setup and instrumentation, the fault simulation setup, the data acquisition technique, signal preprocessing strategy, feature extraction pipeline, and the machine learning classification model. The methodological workflow is organized sequentially as Experimental Setup and Engine Configuration, Data Acquisition Protocol, Signal Preprocessing and Feature Extraction, Machine Learning Classification using Support Vector Machine (SVM) and Evaluation Strategies and Performance Assessment.

3.1 Methodological Framework

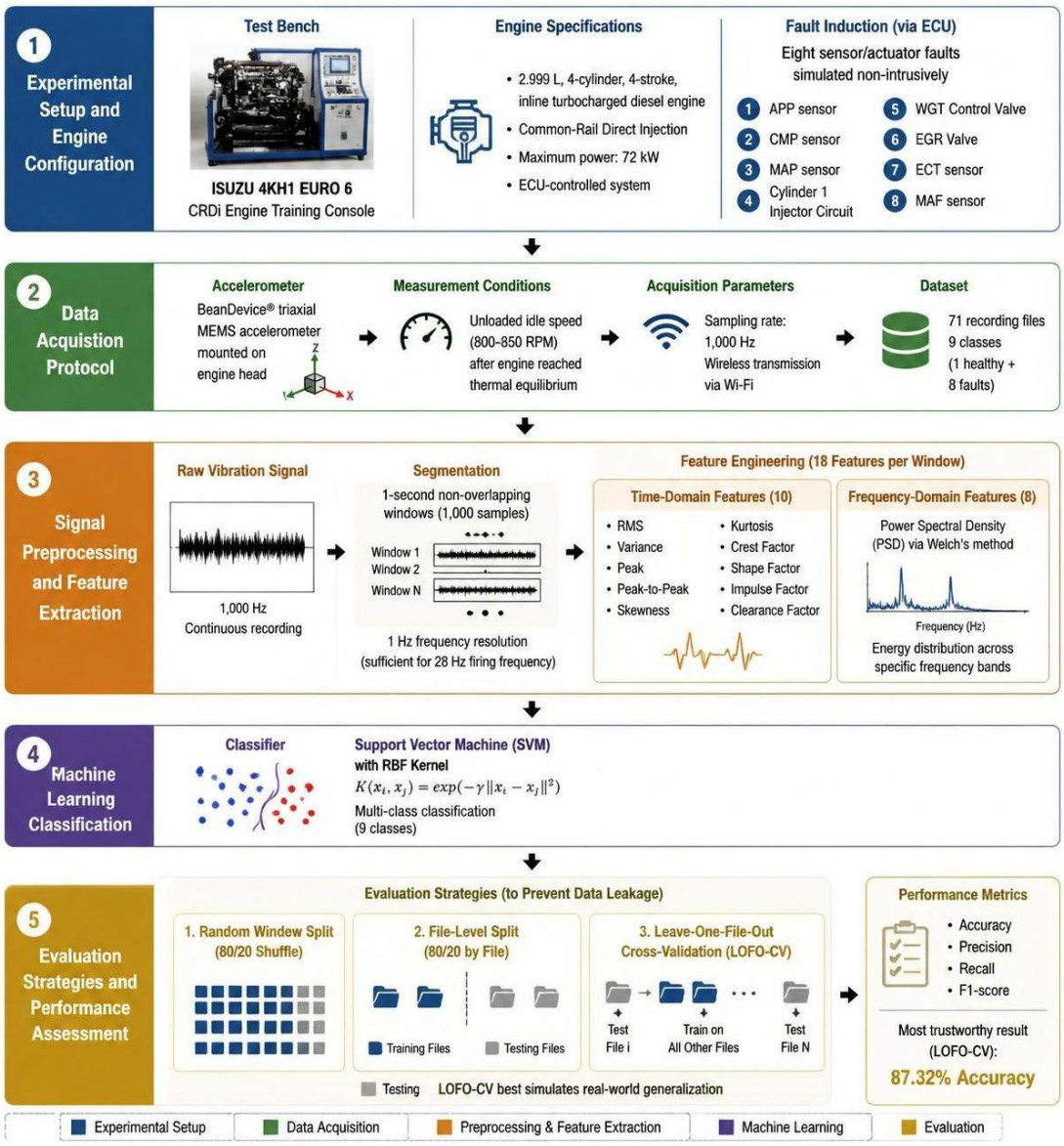

Figure 2 Methodological Framework of the Research

The methodology adopted for the research as depicted in Figure 2 outlines a systematic, step-by-step strategy to achieve the research objective. The process was segregated into five phases: Experimental Setup and Engine Configuration, Data Acquisition Protocol, Signal Preprocessing and Feature Extraction, Machine Learning Classification using Support Vector Machine (SVM) and Evaluation Strategies and Performance Assessment.

The experiments were conducted at the ISUZU 4KH1 training console at the Mechanical Training Centre, Department of Road, Lalitpur, Nepal. All the

experimental steps adhered to the industry standard safety protocol, ethical data handling practices, and established principles for engineering research and ensuring reproducibility of the research.

3.2 Preparation phase

The preparation phase includes rigorous review of previous works and become the critical phase yielding the solid experimental base and assimilate the knowledge required to complete the research gap analysis.

3.2.1 Literature Review and Gap Analysis

A systematic literature review of the vibration signal-based fault diagnosis researches was performed to find the research gaps, focusing on limited integration of the vibration-based signal analysis specific to sensor and actuator-based faults on diesel engines. Key methodologies incorporating time-domain analysis, Fast Fourier Transform (FFT), and feature extraction methods were synthesized to gain insight related to experimental framework. The review further explored to capture recent developments and experiments and researches.

3.3 Experimental Setup

3.3.1 Test Bench Setup

The vibration measurements and fault simulations were conducted on ISUZU 4KH1 Diesel Engine Training Console at the Mechanical Training Center, Department of Roads, Nepal. The engine is Euro 6 compliant, an inline 4- cylinder with Common Rail Direct Injection (CRDi) technology with a displacement of 2,999 cc and maximum power output of 72 kW. The Engine consists of Engine Management System (EMS) consisting of Engine Control Module (ECM), multiple sensors, ECM-controlled actuators and closed-loop feedback algorithms for combustion optimization, boost pressure regulation, fuel control and Exhaust Gas Recirculation (EGR) management. The experimental setup is depicted in Figure 3.

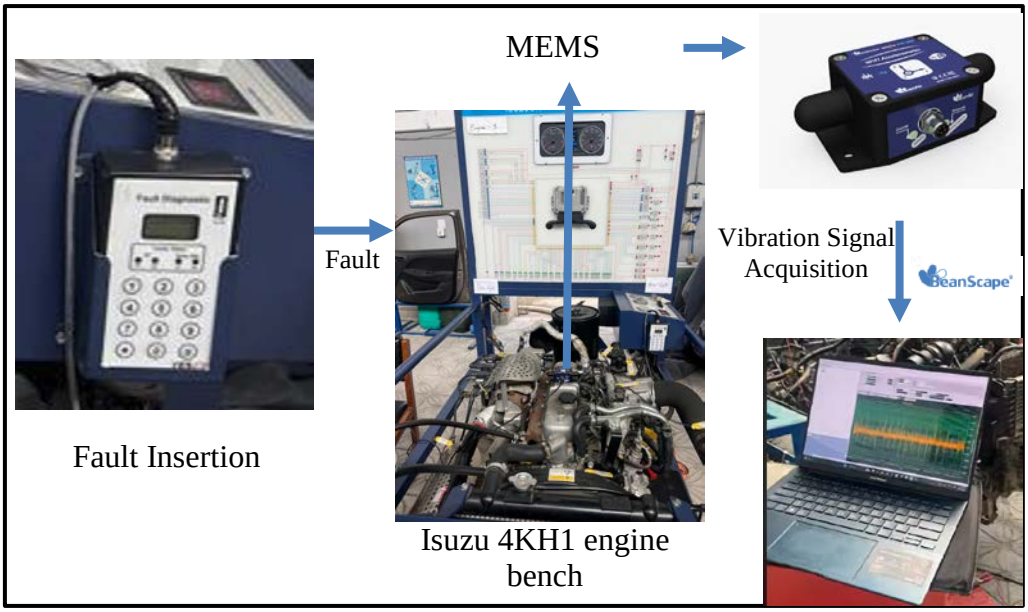

Figure 3 Experimental Setup

The diesel engine training console is coupled with intelligent-fault insertion module enabling non-intrusive simulation of electronic sensor and actuator-based fault conditions. Additionally, this module can insert of sensor and actuator faults in controlled manner without physical modification of the engine components which was handy to ensure repeatable and reversible fault conditions simulation and subsequent measurement and analysis. The console provided a visual interface for real-time monitoring engine parameters such as engine RPM, Malfunction Indication Lamp (MIL) during experimentation. The specification of the Engine Test Bench is presented in Table 2.

Table 2 Specification of Engine Test Bench

Parameter	**Specification**
Engine Model	ISUZU 4KH1
Configuration	Inline 4-cylinder
Displacement	2,999 cc
Maximum Power	72 kW
Injection System	CRDi
Emission Standard	EURO 6
Engine Management	ECM with sensor-actuator feedback loop
Fault Simulation	Intelligent fault-insertion module
Operating Condition Tested	Idle (800-850 RPM)

A BeanDevice® triaxial MEMS accelerometer manufactured by BeanAir GmbH was utilized for vibration data collection. This sensor measures acceleration along three axes simultaneously and transmits the measured vibration data to the computer via Wi-Fi protocol, making external conditioning hardware and cabled connection required on most setup redundant.

The sensor was installed on the cylinder head of the engine bench firmly using a nano-tape adhesive of approximately 1 mm thickness. The cylinder head was closest to the combustion chamber, being the major vibration source of an engine, carefully selected to avoid possible signal transmission loss of signal for combustion dynamics, valve-train subtleties, variation on injector firing actions, and resonance responses. The sensor deployed poses a valid calibration certificates and all the operation and testing procedure were in compliance with applicable safety protocols.

Data were documented at a sampling rate of 1,000 Hz through the BeanGateway® wireless acquisition gateway complemented by the BeanScape® software platform for post processing before acquired. The sampling rate of 1,000 Hz provides a Nyquist frequency of 500 Hz, which is adequate for capturing the fundamental firing frequency and its first several harmonics for a four-cylinder diesel engine operating at idle speed (Weiergräber et al., 2016).

Table 3 Sensor Configuration

Parameter	**Detail**
Sensor	BeanDevice® triaxial MEMS accelerometer
Manufacturer	BeanAir GmbH
Measurement Axes	Tri-axial (Z, X, Y)
Sampling Rate	1,000 Hz
Data Transmission	Wi-Fi wireless protocol
Signal Conditioning	Not required (integrated)
Mounting Location	Cylinder head
Mounting Method	Nano-tape adhesive (~1 mm thickness)
Data Gateway	BeanGateway®
Post-processing Software	BeanScape®

3.4 Baseline Operation and Fault Condition Simulated

The engine was operated under normal and healthy idle conditions for a stabilization period of at least three minutes preceding each recording session to allow thermal and mechanical steady-state to be reached (Keel-Blackmon et al., 2016). Healthy baseline vibration signatures were established and measured across steady-state idle operation of the engine. Vibration signals from triaxial accelerometer were documented along with engine parameters including RPM. Repeatability was established across 5 different tests in different sessions with data excluded where signal quality was not met.

A total of eight sensor and actuator-based fault conditions were inserted via the intelligent fault-simulation module of the diesel engine training console, in addition to the healthy idle baseline condition. Each fault was introduced in a non-intrusive manner through the module's interface, which interrupts the signal path between the selected sensor/actuator and the ECM. For each healthy and faulty trial, the engine was started either in the healthy idle condition or with the specified fault condition already active. This is to make sure that the recorded vibration data captures the steady-state engine response under each fault, rather than the transient characteristics of fault insertion.

The nine classes of operating condition investigated are listed in Table 4. The numbering of the fault was solely assigned by fault insertion module and was representative of a training bench. It is also to be noted that Fault 2 inserted from the module was excluded from this investigation as the insertion failed to start the engine and hence did not provide the accelerometer any usable recordings during experiment.

Table 4 Fault Class Description

Status	Component Affected	Expected Engine Response
IDLE	None (healthy baseline)	Normal idle (800-850 RPM)
Fault 1	Accelerator Pedal Position (APP) Sensor	Limp mode (~1,250 RPM)
Fault 3	Camshaft Position (CMP) Sensor	Degraded timing control
Fault 4	Manifold Intake Pressure (MAP) Sensor	Erroneous fueling
Fault 5	Cylinder 1 Injector Circuit	Missing power stroke
Fault 6	Wastegate Turbocharger (WGT) Valve	Boost pressure anomaly
Fault 7	Exhaust Gas Recirculation (EGR) Valve	Altered EGR flow
Fault 8	Engine Coolant Temperature (ECT) Sensor	Incorrect thermal feedback
Fault 9	Mass Air Flow (MAF) Sensor	Erroneous air mass data

As depicted in the Table 4, each studied fault expected vibration impact is unique. Fault 1(APP) forces the ECM into the limp-mode with fixed fueling response thereby elevating the engine speed to approx. 1250-1300 RPM and essentially altering the combustion dynamics and vibration response. The ECM's ability to harmonize injection timing as per the crankshaft position feedback was demonstrated by Fault 3(CMP sensor) thereby departing from the optimum combustion operation of engine. Fault 4 (MAP) yielded incorrect manifold pressure data to the ECU fueling algorithm resulting in an erroneous air-fuel mixture calculation and resulting in periodic irregularity in combustion. The Fault 5(Injector 1) cased loss of single power stroke among four stroker per two crankshaft revolutions and generating subsequent combustion disharmony when one of the fuel injectors was disabled. Fault 6(WGT) disrupted the boost pressure input to the engine, while Fault 7(EGR) modified the real-time optimization of recirculated exhaust gas fraction. Fault 8(ECT) provided improper feedback of the engine coolant temperature and Fault 9 (MAF) presents erroneous air mass flow rate data, both of which affect ECM's predefined combustion optimization approaches.

3.5 Data Acquisition Protocol

For each simulated condition, multiple sessions were conducted to record and ensure statistical representativeness. The data acquisition protocol followed these steps:

- Secure mounting of the accelerometer on the engine head was confirmed along with the wireless connectivity of the sensor with the PC used in the data acquisition prior to each test session.
- For healthy idle baseline recordings, the engine was started in normal idle mode and allowed to reach steady state thermal equilibrium before data acquisition.
- For vibration response recording of fault condition, the specific fault was stimulated through the fault-insertion module before engine startup.
- Vibration data were recorded continuously for durations ranging from approximately 120 seconds to over 300 seconds per session.
- Each recorded file was stored by the Beanscape platform as a semicolon-delimited text file containing timestamp and tri-axial acceleration data (Ch_Z, Ch_X, Ch_Y) in units of gravitational acceleration (g).

- Altogether of 71 files across all nine conditions were recorded as 6 and 13 recording sessions were measured per fault class.

3.6 Signal Preprocessing

Experimental vibration data under both normal and faulty conditions were measured and recorded to enable systematic analysis of vibration signature. For this the signal received must be tuned and pre-processed in following manner

3.6.1 Vibration Magnitude Computation

For each vibration signal, the tri-axial vibration data were combined into a single magnitude signal using the Euclidean norm (Van Hees et al., 2013)

$$magnitude(t) = \sqrt{Z(t)^2 + X(t)^2 + Y(t)^2}$$

This magnitude representation provides the resultant vibration energy at each instant regardless of directional orientation, providing a measure suitable for feature extraction without rotational bias. All subsequent preprocessing, visualization, and feature computation were performed on this magnitude signal.

3.6.2 Stable Region Extraction

Raw vibration recordings in case of faulty conditions include transient regions corresponding to engine startup and shutdown that are not representative of the steady-state operating condition and if included on the analysis would mis-appropriately pollute the vibration data. An automated algorithm was designed to isolate the stable engine operation region from each recording by removing these transient data from the raw vibration data. The algorithm operates as follows:

- The resultant magnitude signal is divided into 0.5-second non-overlapping windows.
- For each window the RMS energy is computed after removing the DC offset (mean subtraction) to focus on pure fault driven vibratory content rather than static acceleration due to gravity.
- An energy threshold is set at 30% above the minimum observed window energy, defining the boundary between inactive (engine off) and active (engine running) regions. The initial and final windows exceeding this threshold define the active region boundaries.

- A 10-second buffer data is removed from the beginning and a 5-second buffer from the end of the previously detected active region to exclude startup and shutdown transients' dynamics.
- A minimum region length of 10 seconds is enforced; if the resulting region is shorter, the full signal is considered as a fallback and of inappropriate quality.

For signals where the energy was less than 5% of the maximum energy (indicating the engine running throughout the entire recording sessions), only the start and end buffers were removed without threshold-based detection.

3.6.3 Signal Alignment Investigation

Phase alignment based on cross-correlation was investigated as a potential preprocessing strategy. The objective of this was to align all windows with vibration signals to a common reference (the first IDLE recording) by investigating the time lag which investigate the cross-correlation between signal's magnitude and the reference magnitude. While this approach produced sensible results for within-class alignment (e.g., between IDLE recordings), it was deemed unsuitable for cross-fault class alignment. The computed alterations between vibration signals of fault and the IDLE reference were obviously large, often exceeding the signal duration itself, due to the fundamental dissimilarity of vibrational features between healthy and faulty vibration signals.

This result presented the decision to eliminate potential signal alignment as phase alignment provides no clear benefit to vibrational feature comparison between idle and fault condition and it would subsequently introduce needless data loss.

3.6.4 Signal Segmentation

The preprocessed vibration signals were segmented into fixed-length non-overlapping windows of 1 second (1,000 samples at the 1,000 Hz sampling rate) as prescribed by (Randall, 2011) . The choice of window length was guided by two considerations:

- Frequency resolution: As prescribed by the Rayleigh resolution criterion, a time record length of $T = 1$ s assures a frequency resolution of $\Delta f = 1/T = 1$ Hz, sufficient to resolve the fundamental frequency of (~27 Hz) and its harmonics.
- Statistical stability: At the diesel engine idle speed range of 800-850 RPM, one complete engine cycle completes on approximately 141-150 milliseconds,

producing firing event approximately every 30-40 ms. Each 1 second window therefore contains roughly about 30 firing event which is sufficient for statistical characterization of the vibration signal as prescribed by (Prażnowski et al., 2021)

Non-overlapping windows were used for analysis as the total data volume (7,202 windows from 71 files) without overlap-based integration was deemed sufficient for machine learning. Each window reserved its parent file identity and data label for subsequent file-aware evaluation strategies.

3.7 Feature Extraction

Altogether of 18 features were extracted from each 1-second window of vibration data for each recording, among which 10 statistical descriptors are time-domain based and other 8 descriptors are frequency-domain based. These features were selected based on their recognized diagnostic value in literature landscape of vibration-based condition monitoring and their interpretability in the horizon of diesel engine fault diagnosis.

3.7.1 Time-Domain Feature

Ten statistical features were computed directly from the time-domain magnitude signal of each window. These features quantify the signal’s energy level, dispersion, shape, and impulsive content

Table 5 Time Domain Features Extracted

Feature	Formula	Physical Interpretation
RMS	$\sqrt{(\text{mean}(\text{mag}^2))}$	Overall vibration energy level
Variance	var(mag)	Spread of signal values
Std Deviation	std(mag)	Dispersion around mean
Peak	max(|mag|)	Maximum absolute amplitude
Peak-to-Peak	max(mag) - min(mag)	Full amplitude range
Skewness	$E[(\text{mag} - \mu)^3]/\sigma^3$	Asymmetry; sensitive to impulses
Kurtosis	$E[(\text{mag} - \mu)^4]/\sigma^4 - 3$	Peak sharpness; fault sensitivity
Crest Factor	Peak / RMS	Impulsiveness indicator
Shape Factor	RMS / mean(|mag|)	Waveform shape descriptor
Impulse Factor	Peak / mean(|mag|)	Combined impulse severity

3.7.2 Frequency-Domain Features

For the frequency domain features, Welch's method was preferred over the raw periodogram for its superior variance reduction through segment averaging. Eight frequency-domain features were computed to analyze the vibration data from the Welch's method-based Power Spectral Density (PSD) calculated of each window, with a segment length of 256 samples.

Table 6 Frequency Domain Features Extracted

Feature	Formula / Description	Physical Interpretation
Dominant Freq.	f at max(PSD)	Frequency with highest energy
Spectral Centroid	$\Sigma(f \times PSD) / \Sigma(PSD)$	Weighted mean frequency
Spectral Variance	$\Sigma((f\text{-centroid})^2 \times PSD) / \Sigma(PSD)$	Spectral energy spread
Band 0-20 Hz	Σ PSD(0-20 Hz)	Very low frequency content
Band 20-50 Hz	Σ PSD(20-50 Hz)	Fundamental firing range
Band 50-100 Hz	Σ PSD(50-100 Hz)	Harmonic and combustion energy
Band 100-200 Hz	Σ PSD(100-200 Hz)	High-frequency structural response
Total Spectral Energy	Σ PSD (all)	Broadband energy content

Four different frequency bands as tabulated in Table 6 were selected to to stratify the spectral energy of the vibration signal: the sub-fundamental band, the fundamental firing frequency band, the first and second harmonic band, and the higher-order harmonic band. This band stratification helps on the detection of fault class-wise spectral energy distribution.

3.8 Machine Learning Classification

3.8.1 Classifier Selection: Support Vector Machine

A Support Vector Machine (SVM) along with a Radial Basis Function (RBF) kernel was carefully chosen as the classifier for this study. SVM was chosen for several reasons pertinent to vibration-based fault diagnosis: its effectiveness in high-dimensional feature spaces as stated by (Wang et al., 2022). In this research the number of features (18) is moderate relative to the number of samples (7,202) supports its robustness through margin maximization is suitably fit and the RBF kernel's ability to handle non-linear decision boundaries, which is essential given that the feature analysis-based class boundaries are not linearly separable. Hyperparameter

optimization was achieved using GridSearchCV with 5-fold cross-validation within the training data set.

3.8.2 Feature Scaling

All features were made homogenous to zero mean and unit variance by means of scikit-learn's Standard Scaler before classification. Critically, the scaler was fitted solely on the training data and using the training statistics was then applied to the test data to prevent data leakage from the test data set into the scaling parameters, which is a common methodological error in vibration classification studies (Wheat et al., 2024).

3.8.3 Evaluation Strategies

A unique aspect of this study is to perform a thoughtful comparison of three progressively demanding evaluation strategies to enumerate the impact of splitting methodology on ML classification accuracy. This comparison is motivated by a structural characteristics of vibration datasets: windows extracted from the same recording file share file-specific characteristics (sensor coupling, environmental noise, recording baseline) that can inflate apparent accuracy if windows from the same file appear in both training and test sets.

Strategy 1 - Random Window Split (80/20): In this strategy windows generated from all files are stratified and divided randomly into 80% training data sets and 20% test data sets using stratified sampling to reserve class proportions. This approach is adopted solely to validate the data leakage problem on this split as windows processed from the same recorded source file can appears in both sets, allowing the classifier to exploit file-specific features rather than learning general fault signatures.

Strategy 2 - File-Level Split: In this split strategy one complete data file per fault class is taken as test set and all remaining data files used for training sets. This eliminates intra-file leakage but on the negative side it provides only a single point of performance that depends on that specific file selected for data testing.

Strategy 3 - Leave-One-File-Out Cross-Validation (LOFO-CV): This strategy is the most rigorous approach, in which each and every file is hold out as the test set. Every prediction is performed on data from the recording session assuming it as entirely unseen while training, and every file takes parts on both training and testing providing the most dependable estimate of actual generalization performance.

3.8.4 Performance Metrics

Performance of fault classification was evaluated by means of overall accuracy, per-class recall, precision, and F1-score. Confusion matrices were obtained for each evaluation strategy to envisage the pattern of misclassifications. For the rigorous LOFO-CV approach, accuracy statistics (mean and standard deviation) on fold-level were also conveyed to characterize the robustness of performance across each recording files.

3.9 Software and Implementation

The signal processing and ML pipeline was executed in Python 3 within the Google Colab environment. The principal libraries used were: NumPy for numerical computation, Pandas for data administration, Matplotlib for visualization, SciPy (scipy.signal.welch and scipy.stats) for spectral approximation and feature computation, and scikit-learn for Support Vector Machine classification, feature scaling, hyperparameter tuning, and performance evaluation.

3.10 Findings and Reporting

The findings of the research study were documented systematically and submitted to the department as the thesis report as per the requirement of the Department of Mechanical Engineering.

CHAPTER 4 RESULTS AND DISCUSSIONS

This section describes the result obtained from experimental investigation of the vibration signal of a diesel engine. This study includes vibration data acquisition under one healthy idle condition and eight sensor and actuator-based faulty idle condition, signal preprocessing focusing on steady operation of engine, separation into short analysis windows, time-domain and frequency-domain feature extraction and classification of operating condition via SVM training and testing under three different split strategy.

4.1 Dataset Overview and Signal Characterization

4.1.1 Summary of Data Acquisition

Vibration data was collected using a tri-axial sensor mounted on the head of diesel engine, sampling at 1000 Hz. The vibration signal data along the three axes were captured simultaneously and stored on the text file via the Beanscape® platform. As presented in Table 7, 71 different recording sessions were carried across investigated nine operating class. It is noted that Fault 2 was not investigated as the insertion of this particular fault prohibit the starting of the engine and it did not yield any usable recordings during the experiment. Recording periods across 71 sessions were varied from 40 seconds to over 300 seconds enforcing some variation on the experimental procedure.

Table 7 Dataset Composition

Class	Files	Windows (1 s)
IDLE	13	2,135
Fault 1	8	413
Fault 3	10	940
Fault 4	8	813
Fault 5	8	869
Fault 6	6	751
Fault 7	6	383
Fault 8	6	450
Fault 9	6	448
Total	71	7,202

4.1.2 Raw Signal Visualization

Visualization of raw vibration recordings with the acceleration value of vibration signal along the time domain discovered several standout features. The raw signals displayed transient sections specially at the start and end of each recording session corresponding to the engine startup and shutdown. During the processed steady phase, the vibration signals across all operating class showed periodic behavior corresponding and consistent with the firing cycle of an engine.

Operating classes comparison on the basis of time domain data revealed that onset of Fault 1 generated vibration amplitudes up to 8 g which is substantially higher than all other conditions which remained below 3.5 g.

4.2 Windowed Signal Segmentation

The segmentation of total preprocessed 71 sessions produced 7,202 one-second windows altogether across all nine classes investigated. The class distribution as shown in Figure 4 reveals that IDLE class dominates with 2,135 windows owing to the robust baseline establishment requirement for comparison and longer recording durations of healthy-state files, while the smallest classes (Fault 7, Fault 8, and Fault 9) contain 383 to 450 windows.

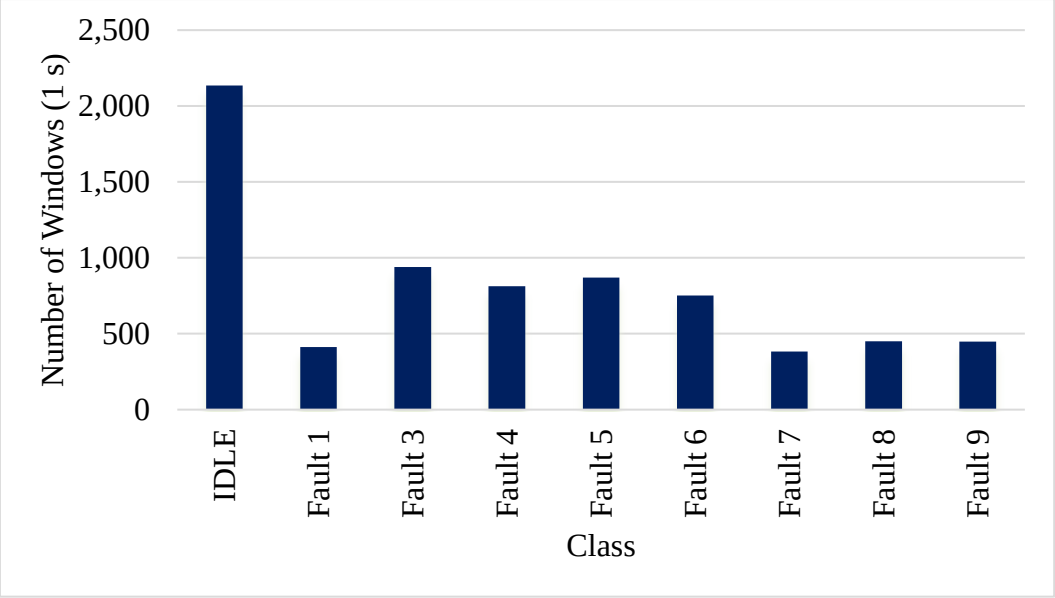

Figure 4 Distribution of analysis windows across fault classes

4.3 Time-Domain and Frequency-Domain Analysis of Vibration Windows

Demonstrative 1-second windows from each operation class were analyzed in both time domain and frequency domain spectrum to characterize the discrete vibration signatures allied with each operating condition. Figure 5 presents paired time-series and Power Spectral Density (PSD) plots for one randomly selected window per class, estimated using Welch's method with a segment length of 256 samples.

4.3.1 Time-Domain Observations

The time-domain observation of the healthy idle condition displayed moderate-amplitude vibration (typically 0.5-3 g) and a visibly noisy broadband presence. This periodic structure is characteristic of internal combustion engines, where each combustion event generates an impulsive force that propagates through the engine structure (Ramachandran et al., 2012). Fault 1 produced distinctly higher amplitudes at peak (up to 8g) with suggesting impulsive vibration behavior which is a classic footprint of a malfunction of a diesel engine that produces transient energy eruptions (Delvecchio et al., 2018). Among Fault 3 through 9 the vibration amplitudes of the engine is within the average 0.5-3.5 g range with notable variation on periodicity and modulation, making them visually difficult to differentiate among by time-domain examination alone. This similarity in vibration features stresses the challenge of detecting these fault classes using time-domain analysis alone and necessitates feature extraction and

frequency-domain analysis, as recommended in the vibration monitoring literature (Romanssini et al., 2023).

4.3.2 Frequency-Domain Observations

The frequency-domain representation of the vibration signal across the different operating class investigated was shown by PSD analysis of the representative one-second window in Figure 5. This PSD data along the fault class shows that the dominant energy is concentrated on low frequency band at approximately below 30 Hz except for fault 1 which is concentrated on approximately just above 40 Hz. These spectral peaks are characteristic features of of normal idle engine operation, vibrations signal induced by engine combustion phenomena dominates the frequency content often below 100 Hz band (Geng et al., 2003).

Fault 1 formed higher PSD peak at approximate level of 0.07 g^2/Hz compared to other fault classes which is around 0.010-0.015 g^2/Hz. This rise in PSD is in sync with the time-domain observation of elevated vibration energy level within this fault. The differences in harmonic structure, although in subtle amount, were visible between other investigated fault classes. Fault 3 and Fault 5 showed peaks near the fundamental peak and secondary harmonics which is nearly two-fold of fundamental peak i.e just below 60 Hz band. Fault 8 and Fault 9 displayed somewhat broader spread in the second harmonics frequency zone i.e. 75-100 Hz. These spectral variances among the operation classes inspired the implementation of band-specific energy structures in the feature extraction framework. The spectral band energy has been used as an important diagnostic feature, establishing the fact that different fault types generates specific spectral energy on different frequency bands forming the basis of reliable vibration-based fault detection (Zhao et al., 2017).

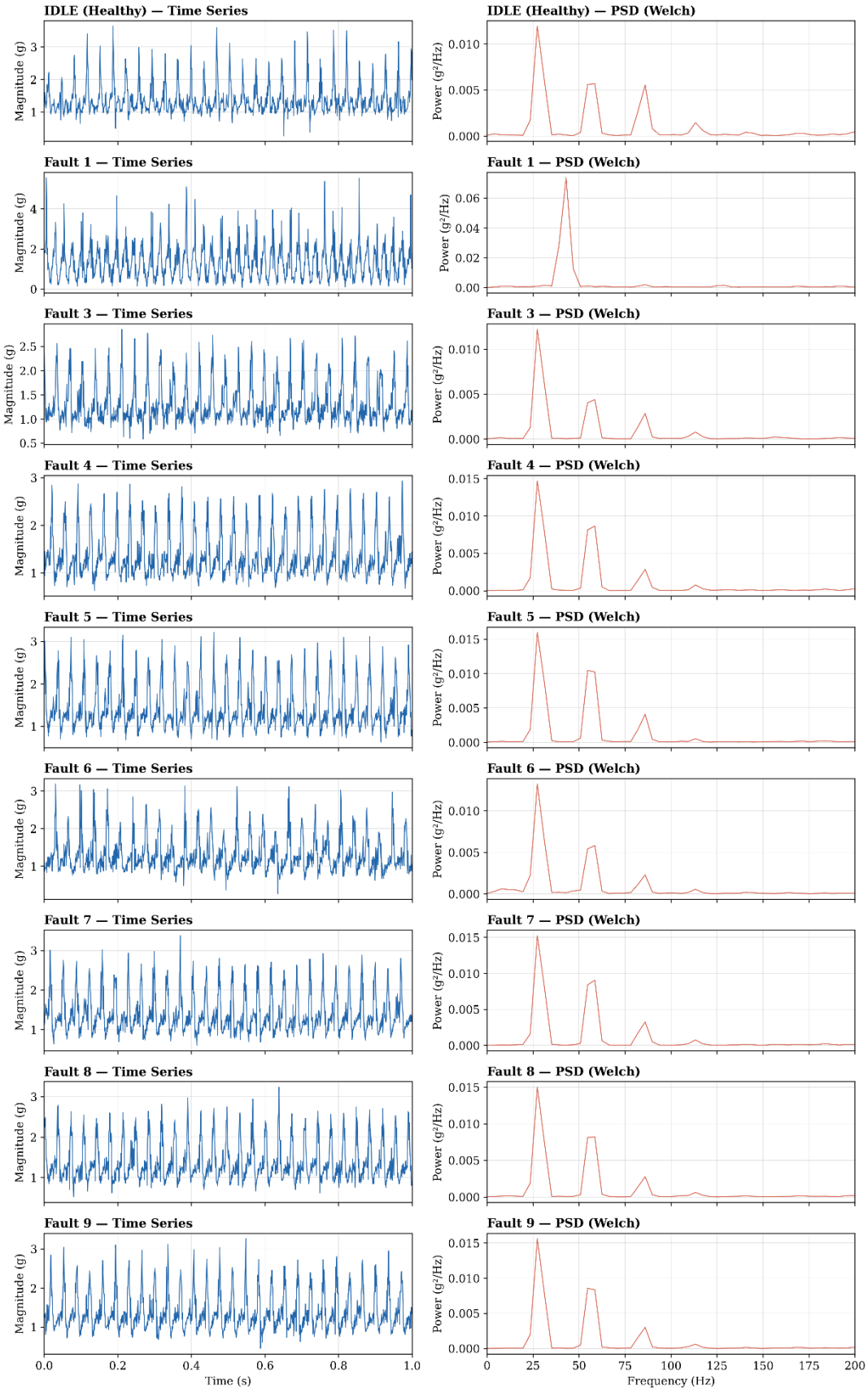

Figure 5 Representative Time-Domain Magnitude and Welch PSD

4.4 Feature Extraction and Analysis

4.4.1 Feature Set Description

18 features were extracted from the preprocessed 1-second window, combining 10 time-domain statistical descriptors and 8 frequency-domain descriptors. The time-domain features comprised of Root Mean Square (RMS), standard deviation, variance, peak amplitude, peak-to-peak amplitude, kurtosis, skewness, crest factor, impulse factor, and shape factor. The frequency-domain features, computed from the Welch PSD estimate, included dominant frequency, spectral centroid, spectral variance, energy in four frequency bands (0-20 Hz, 20-50 Hz, 50-100 Hz, and 100-200 Hz), and total spectral energy. Some of the selected features are listed on the Table 8 and analyzed subsequently.

Table 8 Selected feature set across operating conditions

Class	**RMS (g)**	**Kurtosis (g)**	**Crest Factor**	**Skewness**	**Band 50-100 Hz (g^2)**	**Dominating Freq (Hz)**	**Total Energy (g^2)**
IDLE	1.34	2.11	2.21	1.63	0.01	27.34	0.04
Fault 1	1.63	6.01	4.66	1.33	0.01	42.77	0.19
Fault 3	1.38	1.47	2.27	1.29	0.02	27.34	0.05
Fault 4	1.44	1.16	2.22	1.29	0.02	27.34	0.06
Fault 5	1.52	1.2	2.26	1.4	0.03	27.34	0.07
Fault 6	1.4	1.94	2.38	1.3	0.02	27.34	0.05
Fault 7	1.47	1.13	2.2	1.28	0.02	27.34	0.06
Fault 8	1.45	1.1	2.18	1.25	0.02	27.34	0.06
Fault 9	1.46	1.02	2.18	1.43	0.02	27.34	0.06

4.4.2 Feature Distribution Analysis

The box-plot distributions of major six diagnostic features on the basis of uniqueness and distinguishability across all tested nine classes are portrayed in Figure 6 . Several key observations appeared from this analysis that will directly be the basis of anticipated classification performance.

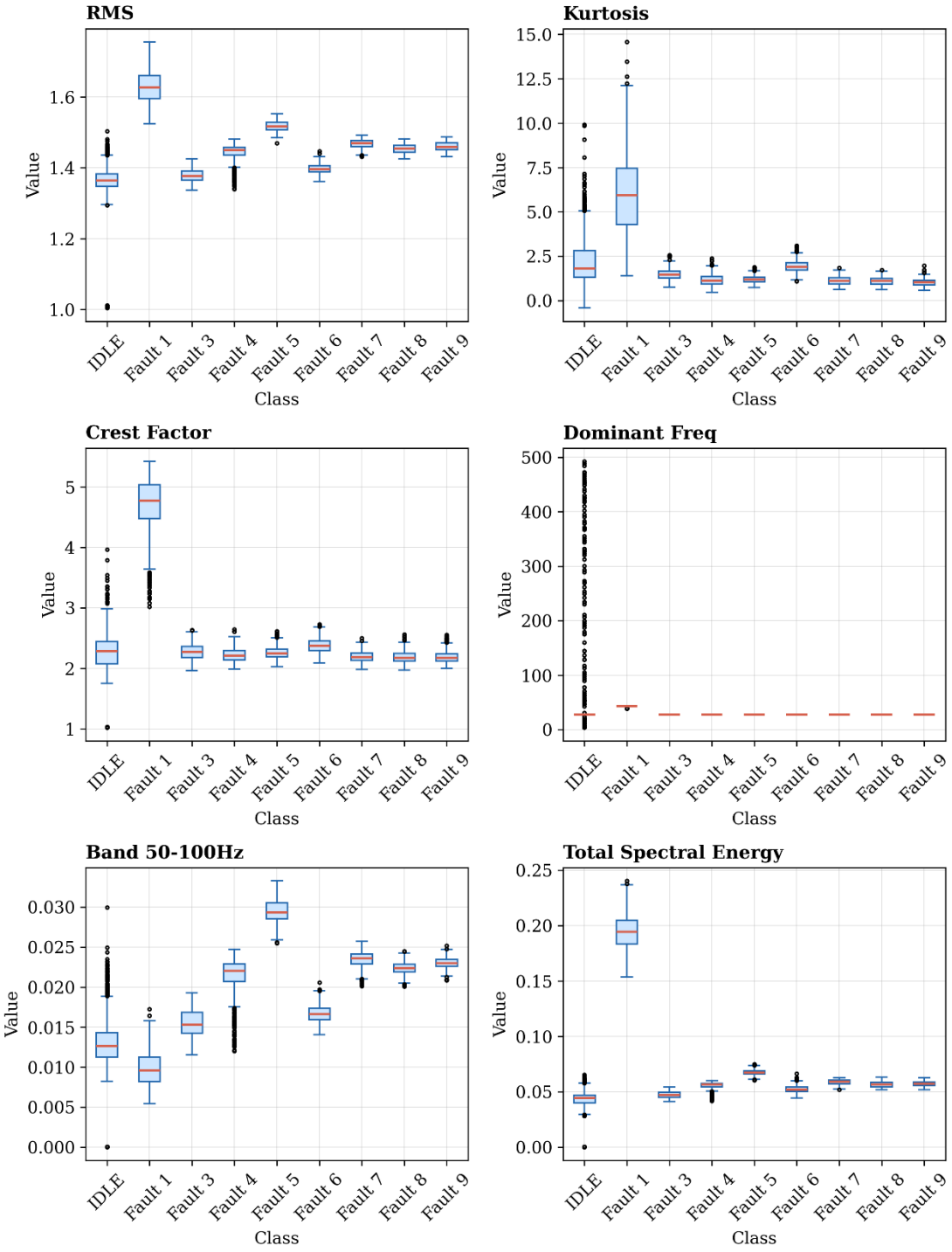

Figure 6 Feature Distribution across Fault Class

RMS: When Fault 1 is simulated on the engine bench the vibration signal was found to have the maximum median RMS (≈1.63 g), showing the elevated vibration signal during limp-mode operation imposed by the fault. The idle baseline operation generates the vibration signal with the lowest median RMS (≈1.35 g) as expected. RMS when observed along Faults 3-9 the signal is in the intermediate magnitude i.e. 1.37-1.52 g.

Among them the distinct fault set are Fault 5 (Injector 1) and Fault 6 (WGT) with medians on the higher side of the range. The RMS feature on its own reflect distinct feature base for Fault 1 but shows confusing overlaps among Faults 7-9, necessitating complementary sets of features for multi-class discrimination.

Kurtosis: If analyzed for the Kurtosis scenario during the different operating condition tested, widest spread (approximately 1.5-5, with maximum outliers up to ≈14) appears in the idle condition, suggesting some sporadic transient events which may be from valve-train noise or brief combustion anomalies. Fault 1(APP) in congruence with other features shows substantially elevated median kurtosis of up to 6 coupled with a wide distribution, indicating possible impulsive combustion nature during imposed limp-mode operation. When the engine is run on simulating Faults 3-9 the kurtosis extracted from vibration signal is low and closely clustered in the range of 1.4-2.0, echoing almost periodic vibration without dominant impulse.

Crest Factor: When the vibration response of the engine is extracted for crest factor, Fault 1 as usual have the highest median crest factor up to 4.7, corroborating the elevated peak-to-RMS ratio during operation. The idle healthy baseline shows a crest factor of moderate median up to 2.2 but wide scatter. Crest factor like other features cluster in the 2.1-2.4 range when Faults 3-9 are activated.

Dominant Frequency: When the engine is run on idle condition substantial variation in dominant frequency (mean 27 Hz with outliers up to 500 Hz) is visible, presenting the dilemma of pinpointing a unique dominant frequency as multiple harmonics are present at similar magnitudes. Fundamental frequency peak when Fault 1 is activated concentrates loosely near 40 Hz, confirming the RPM increment and subsequent fluctuation during limp mode. When insinuated Faults 3-9 on the engine bench the dominant peak frequency cluster tightly near 27 Hz, confirming that engine is operating in 800-850 rpm range under these sensor fault conditions.

Band Energy (50-100 Hz): Band energy features along 50-100 Hz signify irregularity impact above the fundamental in terms of harmonic and combustion energy. Fault 5 (Injector 1) when induced is associated with the band energy (0.030 g^2), highest among counterparts, combined with strong second harmonic and sub-harmonic content echoing the fact that there is a loss of a power stroke per cycle. Fault 4 (MAP sensor) and Fault 7 (EGR valve) when simulated also show considerably raised band energy,

driven by possible combustion imbalance instated by incorrect ECU sensor inputs and actuator outputs. The idle healthy baseline and Fault 1 asynchronous to other features record the lowest band energy in this band, as their spectral impact is visibly concentrated below 50 Hz range.

Total Spectral Energy: When the vibrational response of the engine is extracted for total spectral energy, Fault 1, once again, is prominent player, with a median magnitude of almost 0.195 g^2 which is nearly 3.5 times higher than the healthy baseline magnitude of 0.048–0.072 g^2. This feature solely provides strong base for high-confidence detection of the APP sensor-induced fault scenario of the engine. Total Spectral energy are grouped closely when Faults 3-9 are inserted demanding application of multi-feature combinations for reliable discrimination.

Overall, these most significant six feature analysis shows that not any single features separate all nine classes exclusively, thus confirming the need of a multi-domain and multi-feature bases ML approach for the fault classification of investigated sensor and actuator fault. On the basis of these six features, Fault 1 is is distinctly visible. Accurate classification and discretization of Faults 3 through 9 require the combined discretization analysis of all 18 features for accurate and effective classification for which SVM classifiers are suited.

4.5 SVM Classification Results

4.5.1 Evaluation Strategy and Rationale

The SVM classifier was trained and evaluated using three splitting strategies to demonstrate the critical importance of evaluation methodology in vibration-based fault diagnosis. A Radial Basis Function (RBF) kernel was selected, and hyperparameters were optimized through grid search 5-fold cross-validation within the training set. All features were standardized using Standard Scaler fitted exclusively on training data to prevent information leakage through the scaler.

Table 9 Comparison of SVM classification accuracy across three evaluation strategies

Split Strategy	Accuracy (%)	Leakage Risk	Reliability	Description
Random Window	92.51	High	Low	Windows from same file in train and test
File-Level	83.49	None	Medium	Single held-out file per class
LOFO-CV	87.32	None	High	Every file tested exactly once (71 folds)

4.5.2 Split 1: Random Window Split (80/20)

The first strategy of random 80/20 split achieved the highest test accuracy of 92.51 %. The per-class F1-scores ranged from 0.69 (Fault 9) to 1.00 (Fault 1, Fault 5, Fault 6). However, this result is exaggeratedly inflated due to expected data leakage as because windows from the same source file appear in both training and test sets, the classifier can exploit file-specific details for its accuracy rather than learning generalizable fault signs. This approach solely to validate the danger of data splitting in vibration analysis.

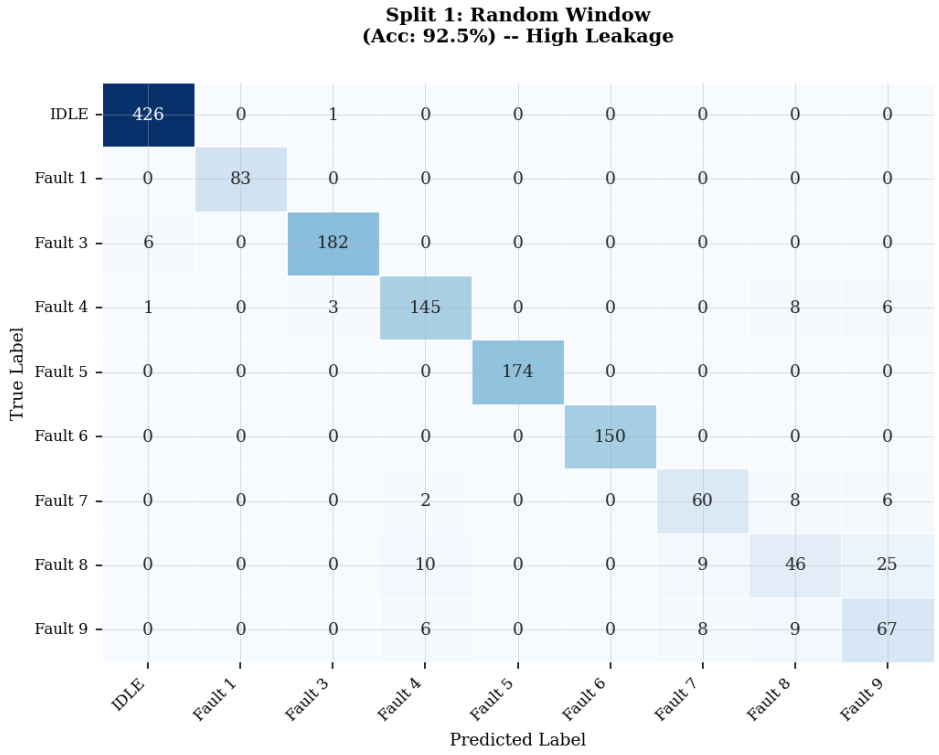

Figure 7 Confusion Matrix for Random Split Strategy

The highest accuracy on confusion matrix as shown on Figure 7 is normally not treated as a success rather than the red flag as the data training flaws as model inflate the test accuracy by essentially memorizing the signal signature rather than learning from the trained fault patterns. Most operation classes depict perfect diagonal robustness (Fault 5: 174/174, IDLE: 426/427, Fault 3: 182/188). The only weakness appears in Faults 7-9, which confuse with each other as for Fault 8, 25 stances were misclassified as Fault 9 getting 46 stances on the correct side.

4.5.3 Split 2: File-Level Split

The file-level split, where one file per class was used for testing, yielded genuine accuracy of 83.49 %. This drop as compared to the random split approves the existence of data leakage in the first split approach. Notable performance differences in F1 score were observed: Fault 1, 5 and 6 maintained flawless classification while Fault 3 and IDLE achieved good classification score of 0.95. However, F1-score collapsed to 0.08 (only 4% recall) for Fault 7, and Fault 4 the classification precision reduced at 0.70. These results obtained established that a single file may fail to represent its class in perfection suggesting the more dependable and comprehensive LOFO-CV split. The importance of temporal splitting to avoid data leakage has been emphasized in cross-validation literature for vibration-based diagnostics (Zhao et al., 2017).

Figure 8 shows the confusion matrix for file level spilt strategy with idle shows new confusion with fault 3 on 11 stances. Fault 7 is confused as fault 4 on 33 sample stances and as fault 8 on 26 stances while Fault 8 with 29 stances misclassified as fault 4 loses which is nearly half of the correctly classified sample.

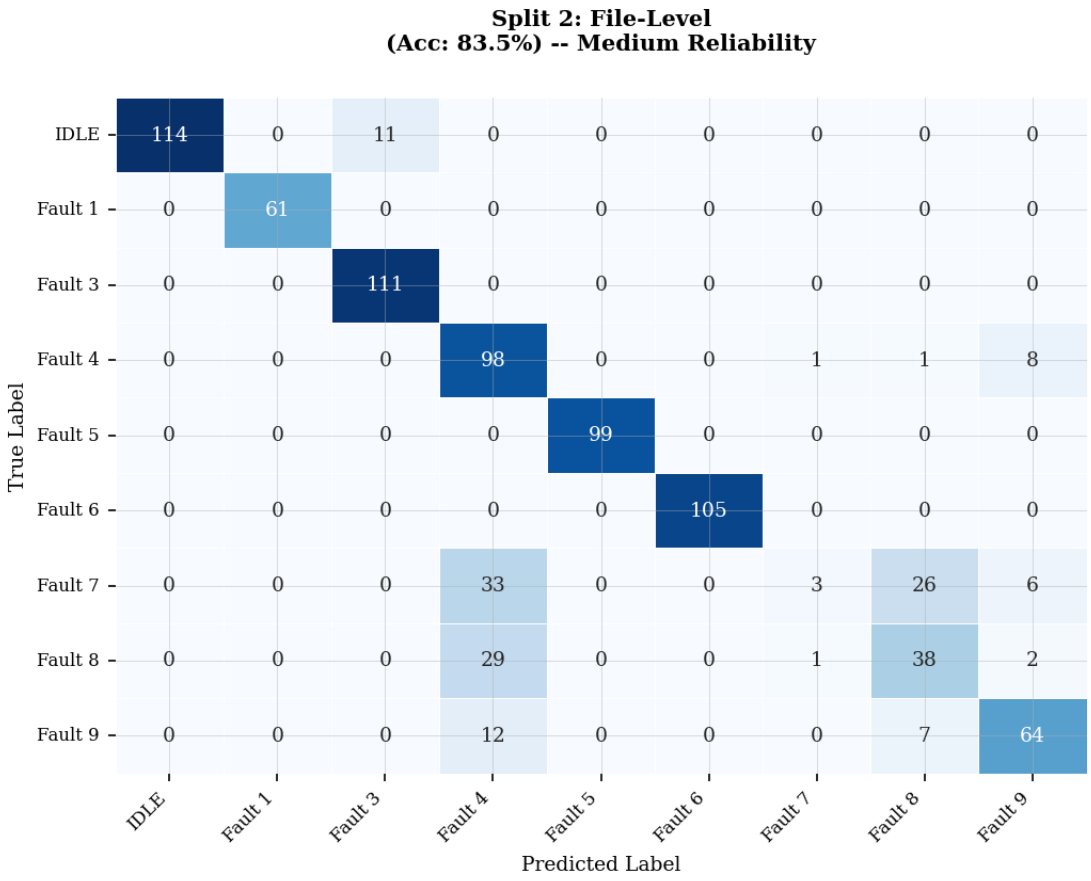

Figure 8 Confusion Matrix for file level split

4.5.4 Split 3: Leave-One-File-Out Cross-Validation (LOFO-CV)

The LOFO-CV evaluation split considered as the most reliable approach, tested every file precisely once across 71 folds. The overall accuracy was 87.32%, with a mean fold accuracy of 84.43% and standard deviation of 25.73% indicating that the evaluation performance of the model is excellent in some files and catastrophically on some with Fault 7, 8 and 9 being the one. The evaluated accuracy of 83.0% for nine-class classification evaluation relates favorably with several results published in fault diagnosis landscape of engine. (Wang et al., 2022) achieved 90% accuracy for three-class classification using Hilbert-Huang transform and SVM on audio signals from a gasoline engine. (Sharma et al., 2014) reported 84.8% overall classification accuracy for misfire detection in an IC engine using vibration signals with a linear model tree, noting that even marginally lower accuracy does not affect practical performance because the system could still distinguish between misfire and normal conditions with perfect accuracy. (Zhao et al., 2017) achieved 98.54% accuracy for valve clearance

fault diagnosis, though with a smaller number of fault classes (seven conditions) and using a more complex signal processing.

The confusion matrix as shown in Figure 9 for this splits clearly show two distinct blocks in a upper left and lower right featuring clean comparison and heavy cross-contamination respectively. Faults 5 (869/869), Fault 1 (413/413) and Fault 6 (751/751) are perfectly classified while Fault 3 (902/940) is pseudo-perfectly classified.

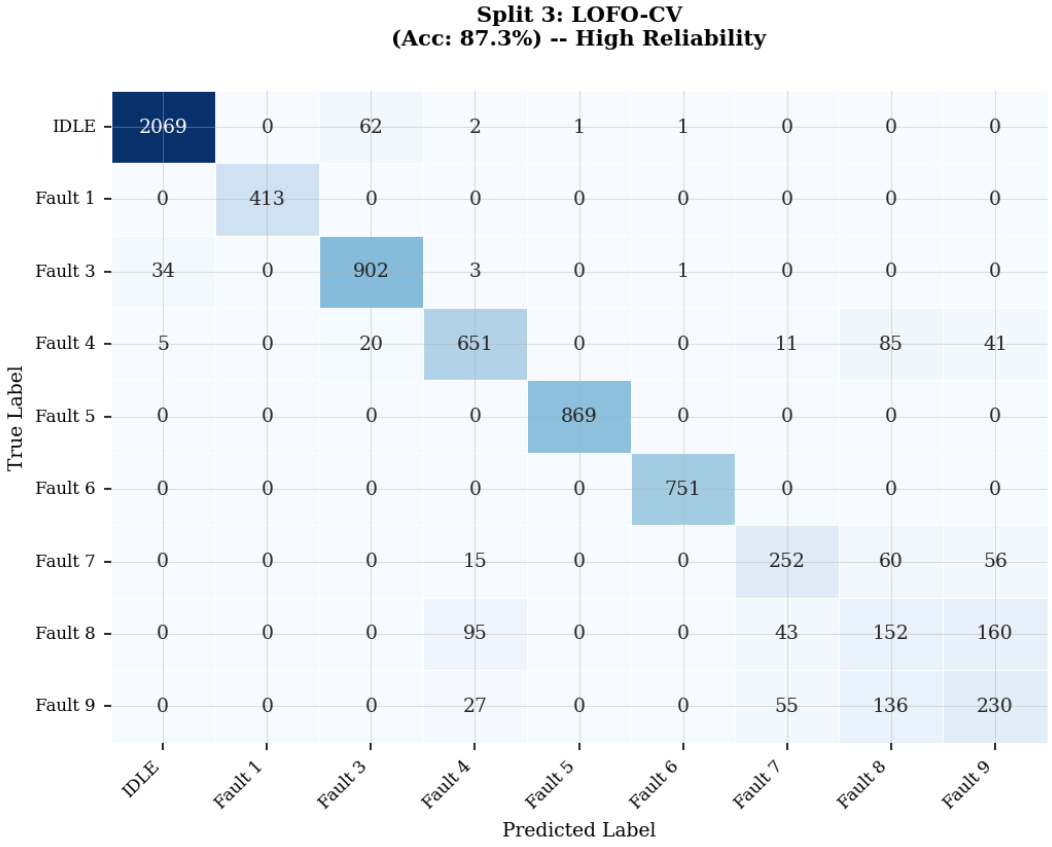

Figure 9 Confusion Matrix for LOFO-CV

The classification analysis of the LOFO-CV results discloses the following clear hierarchy.

Table 10 Classification performance under LOFO Cross-Validation

Class	Precision	Recall	F1-Score	Support
IDLE	0.98	0.97	0.98	2135
Fault 1	1.00	1.00	1.00	413
Fault 3	0.92	0.96	0.94	940
Fault 4	0.82	0.8	0.81	813
Fault 5	1.00	1.00	1.00	869
Fault 6	1.00	1.00	1.00	751
Fault 7	0.7	0.66	0.68	383
Fault 8	0.35	0.34	0.34	450
Fault 9	0.47	0.51	0.49	448

Tier 1 - Perfectly classified (F1 = 1.00): Idle class, Fault 1, 5 and 6 were classified with flawless precision and recall under LOFO-CV. Fault 1's perfect classification is consistent with its clear and obvious separation in RMS, kurtosis, and crest factor observed in the time domain feature analysis. Fault 5 and Fault 6 evidently possess distinctive spectral signatures in the band energy features to enable reliable file-level generalization.

Tier 2 - Well classified (F1= 0.81-0.98): Fault 3 (F1= 0.94) showed consistently strong performance. Fault 4 (F1=0.81) showed good but imperfect classification, with 20 windows misclassified as Fault 5 in the confusion matrix, consistent with the feature overlap observed between these two classes in the band energy distributions.

Tier 3 - Challenging classes (F1=0.34-0.68): Fault 7 (F1=0.68), Fault 8 (F1=0.34), and Fault 9 (F1=0.49) exhibited substantial misclassification. As revealed substantially by the confusion matrix, Fault 7 windows were frequently confused as Fault 8 on 60 stances and Fault 9 on 55 stances. Similarly, Fault 8 was misclassified with Fault 7 on 43 stances and Fault 9 on 160 stances. In the detection case of Fault 9 the split confused it with Fault 7 on 136 stances and with Fault 8 on 136 stances. This suggest that Faults 7, 8, and 9 share similar vibration signature in the 18-dimensional feature space studied.

The fold accuracy also shows the higher standard deviation of 25.73% stresses the unpredictability within these classes at challenging tier. The individual classification

fold accuracies vary from 2.94% (Fault 7, sample 6) in challenging class to 100% for many files in the well-classified classes. This variation on fold accuracy suggests that some fault classes contain operational conditions or recording artifacts that differ meaningfully from other recording files within the same class.

This finding propels the literature landscape on automated fault detection of the diesel engine with significant implications for the real-world deployment of vibration-based diagnostic system Basic SVM model reporting 83-87% accuracy on genuinely new recordings, which pave the foundation for accurate sensor and actuator based fault classification for predictive maintenance strategy.

4.5.5 Feature Discriminability and Class Confusion

The feature analysis and classification results present the unique fault diagnosis challenges as it successfully captured the dominant physical differences between engine conditions

- Fault 1 is distinct, impulsive with high-energy signature and elevated RMS, kurtosis, and crest factor.
- Other fault groups the significant distinction were captured by band energy features; and the statistical shape such as skewness and shape factor of vibration distributions.

The confusion and misclassification while detecting Faults 7, 8, and 9 suggests that these fault conditions propagates with generation of somewhat similar vibration signature in the measured frequency range (0-200 Hz). Fault 7 (EGR), Fault 8(ECT), Fault 9 (MAF) when onset produces similar engine enforcement producing combustion on either rich fuel mix or lean fuel mix.

4.5.6 Implications for Maintenance Awareness

Regarding a maintenance-awareness perspective of this investigative thesis study, the findings strongly demonstrate that MEMS-based vibration monitoring can reliably and accurately classify five of the eight type sensor and actuator-based fault condition. For the three classes with some confusion on classification, the SVM classification model could still flag fault condition and narrow the diagnosis. The adoption of the fault classification model could meaningfully reduce diagnostic effort and improve the accuracy of the diagnosis and facilitate the transition of fault detection methodology in

countries like Nepal from judgement and inspection- based by the operator to automated and vibration based.

CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS

5.1 Conclusions

The investigation and analysis of the result of the vibration signal produced by MEMS accelerometer during the operation of one healthy and eight different sensor and actuator based-faulty condition presented on this thesis leads to the following conclusion.

5.1.1 Vibration Characteristics under Healthy Idle Condition

The operation of the diesel engine on healthy idle condition produced vibration amplitudes from 0.5 to 3.0 g. The dominant spectral energy as investigated by welch PSD was found to be concentrated in the lower 0-50 Hz frequency band especially near the 30 Hz, its low-order harmonics. The idle condition showed high variance in dominant frequency, spanning the full 0-500 Hz measurement range.

5.1.2 Vibration Signatures under Simulated Fault Conditions

The APP sensor fault (Fault 1) produced the most dramatic vibration signal difference reflecting the elevating engine speed to approximately 1,250 RPM due to ECM limp-mode activation, with peak amplitudes reaching up to 8 g median kurtosis of 5.5 and crest factor of 4.6. The cylinder 1 injector circuit failure (Fault 5) and wastegate turbocharger control valve failure (Fault 6) produced distinctive spectral features that enabled flawless classification under the most demanding file-level evaluation.

5.1.3 Fault Diagnosis Model Performance

The SVM classifier achieved a global accuracy of 87.32% under the LOFO CV, which is the most consistent estimate of real-world performance among the random window split (92.51%) and file-level split (83.49%). The LOFO-CV classification put forwards three performance tiers: perfectly classified faults (Fault 1, Fault 5, and Fault 6 with F1-scores of 1.00), well-classified fault classes (IDLE with F1 of 0.98, Fault 3 with 0.94, and Fault 4 with 0.81), and challenging classes (Fault 7 with F1 of 0.68, Fault 9 with 0.49, and Fault 8 with 0.34).

5.1.4 Predictive Maintenance Framework Applicability

The research validates that vibration monitoring deploying low-cost MEMS accelerometers can be viable non-intrusive diagnostic approach for modern electronically-controlled diesel engines with latest emission standard compliances. The model can reliably diagnose sensor and actuator-based anomaly with accurately identifying six of the eight tested fault classes, reducing diagnostic effort significantly compared to instinctive methods based on judgement. For the three fault classes with higher confusion degree, the system can narrow the diagnosis providing actionable information to maintenance decision-makers.

5.2 Recommendations for Future Work

The following constructive recommendations are given for future study based on the findings, insights gained and limitations for practical implementation of vibration signal on fault analysis and maintenance of diesel engine.

5.2.1 Exploration of Advanced Classification Methods

Although the SVM model performed well on fault classification, deep learning models such as Convolutional Neural Networks (CNN) based on raw time-series vibration data data which may discretize the fault with more accuracy. Methods with combining multiple classifiers and multi-sensor fusion approaches including other parameters like acoustic emission together with vibration features are also recommended for investigation. Transfer learning approaches, as suggested by Karhinen et al., (2026) could address the scarcity problem related to data for rare fault conditions.

5.2.2 Integration to Maintenance-Aware Framework

This research establishes the foundation for the fault detection and classification integrative with maintenance-aware framework. Future work on the basis of foundation should include Remaining Useful Life (RUL) estimation using the vibration data. Integration of the vibration-based fault diagnostic model with preventive and predictive maintenance scheduling programs would complement this research with the impact on overall cost reduction and efficiency on maintenance strategy. Development of a real-time monitoring console that will analyze the vibration data from the accelerometer employed and displays fault likelihoods and recommend needed maintenance action to engine operators would be beneficial in maintenance management of engine systems.

5.2.3 Real Load Engine Validation and Acoustic Integration

This experimental research was conducted on a training bench which was operated without load and under controlled laboratory conditions with simulated faults. This test bench experimental approach on the one hand ensured repeatability and safety but on the other hand it may not entirely represent the accurate response of faults onset and propagation in diesel engines deployed with real-time load variation. Validation studies incorporating the acoustic signal response data during fault onset of an operational diesel engines in actual operating conditions such as road vehicles, industrial machinery and generators is required critically to ensure the robustness of the vibration based diagnostic model.

REFERENCES

Aftab Ahmed Soomro, A. N. (2025). Data-Driven Predictive Maintenance Of Diesel Engines Using Advanced Machine Learning And Ai-Based Regression Algorithms For Accurate Fault Detection And Real-Time Condition Monitoring. Https://Doi.Org/10.5281/Zenodo.15852571

Ahirrao, N. S., Bhosle, S. P., & Nehete, D. V. (2018). Dynamics and Vibration Measurements in Engines. Procedia Manufacturing, 20, 434-439. https://doi.org/10.1016/j.promfg.2018.02.063

Akbalık, F., Yıldız, A., Ertuğrul, Ö. F., & Zan, H. (2024). Engine Fault Detection by Sound Analysis and Machine Learning. Applied Sciences, 14(15), 6532. https://doi.org/10.3390/app14156532

Böğrek, A., & Sümbül, H. (2022). A Novel Engine Vibration Measurement System based on the MEMS Sensor. International Journal of Automotive Science and Technology, 6(4), 357-363. https://doi.org/10.30939/ijastech..1168298

Chao Jin, Wenyu Zhao, Zongchang Liu, Lee, J., & Xiao He. (2014). A vibration-based approach for diesel engine fault diagnosis. 2014 International Conference on Prognostics and Health Management, 1-9. https://doi.org/10.1109/ICPHM.2014.7036371

Chen, J. (2013). Internal combustion engine diagnostics using vibration simulation [UNSW Sydney]. https://doi.org/10.26190/UNSWORKS/16365

Chu, T., Nguyen, T., Yoo, H., & Wang, J. (2024). A review of vibration analysis and its applications. Heliyon, 10(5), e26282. https://doi.org/10.1016/j.heliyon.2024.e26282

Delvecchio, S., Britte, L., & Bianciardi, F. (2018). Turbocharger Noise Identification on a V8 Engine. MTZ Worldwide, 79(5), 48-53. https://doi.org/10.1007/s38313-018-0018-4

Geng, Z., Chen, J., & Barry Hull, J. (2003a). Analysis of engine vibration and design of an applicable diagnosing approach. International Journal of Mechanical Sciences, 45(8), 1391-1410. https://doi.org/10.1016/j.ijmecsci.2003.09.012

Geng, Z., Chen, J., & Barry Hull, J. (2003b). Analysis of engine vibration and design of an applicable diagnosing approach. International Journal of Mechanical Sciences, 45(8), 1391-1410. https://doi.org/10.1016/j.ijmecsci.2003.09.012

Iglesias Martínez, M. E., Antonino-Daviu, J. A., Dunai, L., Conejero, J. A., & Fernández De Córdoba, P. (2024). Higher-Order Spectral Analysis and Artificial Intelligence for Diagnosing Faults in Electrical Machines: An Overview. Mathematics, 12(24), 4032. https://doi.org/10.3390/math12244032

Jafarian, K., & Mobin, M. (2016). Misfire Fault Detection in the Internal Combustion Engine using the Artificial Neural Networks (ANNs).

Jafarian, K., Mobin, M., Jafari-Marandi, R., & Rabiei, E. (2018). Misfire and valve clearance faults detection in the combustion engines based on a multi-sensor vibration signal monitoring. Measurement, 128, 527-536. https://doi.org/10.1016/j.measurement.2018.04.062

Jiang, Y., Tang, B., Qin, Y., & Liu, W. (2011). Feature extraction method of wind turbine based on adaptive Morlet wavelet and SVD. Renewable Energy, 36(8), 2146-2153. https://doi.org/10.1016/j.renene.2011.01.009

Karabacak, Y. E. (2024). Condition monitoring of internal combustion engines with vibration signals and fault detection by using machine learning techniques.

International Journal of Automotive Engineering and Technologies, 13(4), 191-200. https://doi.org/10.18245/ijaet.1251886

Karhinen, A., Hämäläinen, A., Palestini, C., Kyrki, V., & Viitala, R. (2026). Data-driven torsional vibration-based fault diagnosis of large internal combustion engines without real fault data. Engineering Applications of Artificial Intelligence, 164, 113342. https://doi.org/10.1016/j.engappai.2025.113342

Keel-Blackmon, K., Curran, S., & Lapsa, M. V. (2016). Summary of OEM Idling Recommendations from Vehicle Owner's Manuals (ORNL/TM--2016/50, 1257900; p. ORNL/TM--2016/50, 1257900). https://doi.org/10.2172/1257900

Kn, R., Kumar, H., Gn, K., & Kv, G. (2021). Fault diagnosis of internal combustion engine gearbox using vibration signals based on signal processing techniques. Journal of Quality in Maintenance Engineering, 27(2), 385-412. https://doi.org/10.1108/JQME-11-2019-0109

Lee, S. K., & White, P. R. (1997). Higher-Order Time-Frequency Analysis And Its Application To Fault Detection In Rotating Machinery. Mechanical Systems and Signal Processing, 11(4), 637-650. https://doi.org/10.1006/mssp.1997.0098

MR, S., & Mangu, S. (2025). Predictive Maintenance: A Bibliometric Analysis.

Nouby M. Ghazaly, M. M. A. (2022). A Review on Engine Fault Diagnosis through Vibration Analysis. International Journal on Recent Technologies in Mechanical and Electrical Engineering, 9(2), 01-06. https://doi.org/10.17762/ijrmee.v9i2.364

Patil, N. A., & Raut, L. P. (2016). VIBRATION ANALYSIS OF CI ENGINE USING FFT ANALYZER.

Prażnowski, K., Bieniek, A., Mamala, J., & Deptuła, A. (2021). The Use of Multicriteria Inference Method to Identify and Classify Selected Combustion Engine Malfunctions Based on Vehicle Structure Vibrations. Sensors, 21(7), 2470. https://doi.org/10.3390/s21072470

Ramachandran, T., Padmanaban, K. P., & Nesamani, P. (2012). Modeling and Analysis of IC Engine Rubber Mount Using Finite Element Method and RSM. Procedia Engineering, 38, 1683-1692. https://doi.org/10.1016/j.proeng.2012.06.205

Randall, R. B. (2011). Vibration-based condition monitoring: Industrial, aerospace and automotive applications. Wiley. https://doi.org/10.1002/9780470977668

Rodríguez-Matienzo, J. (1998). Diagnosis of IC Engines using vibrations.

Romanssini, M., De Aguirre, P. C. C., Compassi-Severo, L., & Girardi, A. G. (2023). A Review on Vibration Monitoring Techniques for Predictive Maintenance of Rotating Machinery. Eng, 4(3), 1797-1817. https://doi.org/10.3390/eng4030102

Shah, R., Mittal, V., & Lotwin, M. (2025). Recent Advances in Vibration Analysis for Predictive Maintenance of Modern Automotive Powertrains. Vibration, 8(4), 68. https://doi.org/10.3390/vibration8040068

Sharma, A., Sugumaran, V., & Babu Devasenapati, S. (2014). Misfire detection in an IC engine using vibration signal and decision tree algorithms. Measurement, 50, 370-380. https://doi.org/10.1016/j.measurement.2014.01.018

Siano, D., & Panza, M. A. (2018). Diagnostic method by using vibration analysis for pump fault detection. Energy Procedia, 148, 10-17. https://doi.org/10.1016/j.egypro.2018.08.013

Srinivaas, A., Sakthivel, N. R., & Nair, B. B. (2025). Machine Learning Approaches for Fault Detection in Internal Combustion Engines: A Review and Experimental Investigation. Informatics, 12(1), 25. https://doi.org/10.3390/informatics12010025

Taghizadeh-Alisaraei, A., & Mahdavian, A. (2019). Fault detection of injectors in diesel engines using vibration time-frequency analysis. Applied Acoustics, 143, 48-58. https://doi.org/10.1016/j.apacoust.2018.09.002

Tharanga, K. L. P., Liu, S., Zhang, S., & Wang, Y. (2020). Diesel Engine Fault Diagnosis with Vibration Signal. Journal of Applied Mathematics and Physics, 08(09), 2031-2042. https://doi.org/10.4236/jamp.2020.89151

Van Hees, V. T., Gorzelniak, L., Dean León, E. C., Eder, M., Pias, M., Taherian, S., Ekelund, U., Renström, F., Franks, P. W., Horsch, A., & Brage, S. (2013). Separating Movement and Gravity Components in an Acceleration Signal and Implications for the Assessment of Human Daily Physical Activity. PLoS ONE, 8(4), e61691. https://doi.org/10.1371/journal.pone.0061691

Varanis, M., Silva, A., Mereles, A., & Pederiva, R. (2018). MEMS accelerometers for mechanical vibrations analysis: A comprehensive review with applications. Journal of the Brazilian Society of Mechanical Sciences and Engineering, 40(11), 527. https://doi.org/10.1007/s40430-018-1445-5

Wang, L., Liu, H., Liang, J., Zhang, L., Ji, Q., & Wang, J. (2022). Research on the Rotor Fault Diagnosis Method Based on QPSO-VMD-PCA-SVM. Frontiers in Energy Research, 10, 944961. https://doi.org/10.3389/fenrg.2022.944961

Weiergräber, M., Papazoglou, A., Broich, K., & Müller, R. (2016). Sampling rate, signal bandwidth and related pitfalls in EEG analysis. Journal of Neuroscience Methods, 268, 53-55. https://doi.org/10.1016/j.jneumeth.2016.05.010

Wheat, L., Mohrenschildt, M. V., Habibi, S., & Al-Ani, D. (2024). Impact of Data Leakage in Vibration Signals Used for Bearing Fault Diagnosis. IEEE Access, 12, 169879-169895. https://doi.org/10.1109/ACCESS.2024.3497716

Workshop manual ISUZU Dmax Engine. (n.d.).

Wu, J.-D., & Chuang, C.-Q. (2005). Fault diagnosis of internal combustion engines using visual dot patterns of acoustic and vibration signals. NDT & E International, 38(8), 605-614. https://doi.org/10.1016/j.ndteint.2005.02.007

Zhang, F., Jiang, M., Zhang, L., Ji, S., Sui, Q., Su, C., & Lv, S. (2019). Internal Combustion Engine Fault Identification Based on FBG Vibration Sensor and Support Vector Machines Algorithm. Mathematical Problems in Engineering, 2019(1), 8469868. https://doi.org/10.1155/2019/8469868

Zhao, X., Cheng, Y., Wang, L., & Ji, S. (2017). Real time identification of the internal combustion engine combustion parameters based on the vibration velocity signal. Journal of Sound and Vibration, 390, 205-217. https://doi.org/10.1016/j.jsv.2016.11.013

Zhou, Y., Ma, Z., & Fu, L. (2025). A Review of Key Signal Processing Techniques for Structural Health Monitoring: Highlighting Non-Parametric Time-Frequency Analysis, Adaptive Decomposition, and Deconvolution. Engineering. https://doi.org/10.20944/preprints202502.1299.v1

Zouani, A., & Hanim, S. (2016). Overview of noise and vibration in automotive engines. International Journal of Vehicle Noise and Vibration, 12(2), 162. https://doi.org/10.1504/IJVNV.2016.079054

APPENDIX 1: RAW DATA SAMPLES

Appendix 1: Raw Data Sample

BeanDevice : AX 3D

Range for accelerometer: -10g / +10g

Mac Id : DCF31C2731070000

Network Id : 0106

Pan Id : FFFE

Measure mode : Streaming

Streaming Options : Continuous Monitoring

Unit for accelerometer : g

DATE_FORMAT : M/d/yyyy h:mm:ss tt.fff

Date : 3/17/2026 7:58:02 AM.010

Sampling rate : 1000

Sensor Ids:0|1|2

Sensor Labels:Ch_Z|Ch_X|Ch_Y

TimeStamp;Measure Ch_Z(g);Ch_X(g);Ch_Y(g)

0;0.767;-0.287;0.781

1;0.706;-0.197;0.949

2;0.694;-0.363;0.706

3;0.72;-0.292;0.79

4;0.701;-0.2;0.81

5;0.694;-0.199;0.49

6;0.689;-0.234;0.984

7;0.977;-0.29;-0.049

8;0.887;-0.296;0.591

9;0.898;-0.213;-0.034

10;1.022;-0.287;0.223

11;1.099;-0.236;-0.05

12;1.128;-0.149;-0.048

13;1.345;-0.088;-0.176

14;1.043;0.139;0.097

15;0.928;-0.031;-0.067

16;1.44;0.095;0.057

17;1.351;0.134;-0.222

18;1.256;-0.006;-0.315

19;1.223;0.507;-0.24

20;1.271;-1.111;-1.047

21;1;0.458;-0.947

22;0.216;1.254;-1.524

23;1.16;-0.491;-1.538

24;0.674;0.128;-1.712

25;0.731;0.178;-1.327

26;1.167;0.173;-1.641

27;1.358;0.107;-0.647

28;0.879;0.372;-0.583

29;0.643;-0.244;-0.453

30;0.746;0.198;0.416

31;0.791;-0.218;0.214

32;0.764;-0.22;0.681

33;0.601;0.094;0.708

34;0.795;-0.337;0.705

35;0.751;-0.24;0.948

36;0.637;-0.272;0.649

37;0.701;-0.188;0.857

38;0.764;-0.203;0.965

39;0.693;-0.397;0.847

40;0.904;-0.059;0.375

41;0.813;-0.411;0.881

42;0.931;-0.332;0.182

43;0.81;-0.111;0.548

44;0.962;-0.282;-0.004

45;0.96;-0.282;0.232

46;0.96;-0.223;-0.186

47;1.274;0.006;-0.067

48;1.043;-0.029;0.376

49;1.19;1.027;-0.607

50;0.683;-0.782;0.556

51;0.861;-0.619;0.012

52;1.151;0.149;-0.149

53;1.149;0.137;-0.202

54;1.254;0.145;-0.453

55;1.273;0.463;-0.608

56;1.155;-0.186;-0.956

57;0.449;0.009;-1.182

58;1.567;0.09;-1.592

59;1.776;1.14;-2.159

60;2.144;1.662;-1.751

61;0.953;-0.162;-1.35

62;0.644;-0.708;-0.509

63;1.126;0.344;-0.876

64;1.153;-0.211;-0.154

65;1.152;1.063;-0.189

66;0.69;-0.426;0.336

67;0.792;-0.186;0.131

68;0.734;-0.289;0.967

69;0.8;-0.135;0.6

70;0.758;-0.058;1.06

71;0.678;-0.317;0.814

72;0.676;-0.216;0.895

73;0.705;-0.513;0.87

74;0.699;-0.219;0.651

75;0.921;-0.109;0.933

76;0.811;-0.28;0.694

77;0.914;-0.344;0.229

APPENDIX 2: PYTHON CODE FOR SIGNAL PROCESSING

```
def load_txt_file(file_path):
    with open(file_path, 'r')
as f:
        lines = f.readlines()

    # Find where actual data
starts
    for i, line in
enumerate(lines):
        if "TimeStamp" in
line:
            start_idx = i + 1
            break

    data = []
    for line in
lines[start_idx:]:
        parts =
line.strip().split(';')
        if len(parts) < 4:
            continue
        try:
            z =
float(parts[1])
            x =
float(parts[2])
            y =
float(parts[3])
            data.append([z, x,
y])
        except:
            continue

    return np.array(data)  #
shape (N, 3)

#
==============================
==
# STEP 6: DATASET SUMMARY
#
==============================
==

import pandas as pd
import matplotlib.pyplot as
plt

# ---------------------------
----
# 1. Create Summary DataFrame
# ---------------------------
----
summary_df = pd.DataFrame({
    "file_name": file_names,
    "label": labels,
    "length": [d.shape[0] for
d in data]
})

print("🔹 First few
entries:")
display(summary_df)

# ---------------------------
----
# 2. Count Samples Per Class
# ---------------------------
----
class_counts =
summary_df["label"].value_coun
ts().sort_index()

class_counts_df =
class_counts.reset_index()
class_counts_df.columns =
["Label", "Number of Files"]

# Add readable names
def label_name(x):
    if x == 0:
        return "IDLE"
    else:
        return f"Fault {x}"

class_counts_df["Class Name"]
=
class_counts_df["Label"].apply
(label_name)

class_counts_df =
class_counts_df[["Label",
"Class Name", "Number of
Files"]]

print("\n🔹 Samples per
class:")
display(class_counts_df)



# ---------------------------
----
# 4. Visualization (Bar Plot)
# ---------------------------
----
plt.figure(figsize=(8,4))
plt.bar(class_counts_df["Class
Name"],
class_counts_df["Number of
Files"])
plt.xticks(rotation=45)
plt.title("Number of Samples
per Class")
plt.ylabel("Count")
```

```
plt.grid()
plt.show()

# ----------------------------
----
# 5. Additional Insights
# ----------------------------
----
print("\n🔹 Total samples:",
len(data))
print("🔹 Unique labels:",
sorted(set(labels)))

def plot_xyz(signal,
title="XYZ Signal"):
    plt.figure(figsize=(12,5))

    plt.plot(signal[:,0],
label='Z')
    plt.plot(signal[:,1],
label='X')
    plt.plot(signal[:,2],
label='Y')

    plt.legend()
    plt.title(title)
    plt.xlabel("Time Index")
    plt.ylabel("Amplitude
(g)")
    plt.grid()
    plt.show()

#we can choose which data we
want to visualize here
plot_xyz(data[2],
title=file_names[2])

def
compare_faults(labels_to_compa
re, num_samples=2):
    plt.figure(figsize=(12,5))

    for lbl in
labels_to_compare:
        count = 0
        for i, l in
enumerate(labels):
            if l == lbl:
                plt.plot(data[
i][:,0], label=f"Fault {lbl}"
if count==0 else "",
alpha=0.6)
                count += 1
                if count >=
num_samples:
                    break
    plt.legend()
    plt.title("Comparison
Across Faults")
    plt.xlabel("Time Index")
    plt.ylabel("Amplitude")
    plt.grid()
    plt.show()

# Example
compare_faults([0,1, 3, 5])

def
plot_all_stacked_magnitude(dat
a, file_names, labels):
    n = len(data)

    plt.figure(figsize=(12,
2*n))

    for i in range(n):
        signal = data[i]
        mag =
np.sqrt(signal[:,0]**2 +
signal[:,1]**2 +
signal[:,2]**2)

        plt.subplot(n, 1, i+1)
        plt.plot(mag)  # ☑
magnitude
        plt.title(f"{file_name
s[i]} | Label: {labels[i]}",
fontsize=8)
        plt.xticks([])
        plt.yticks([])

    plt.tight_layout()
    plt.show()

# Run
plot_all_stacked_magnitude(dat
a, file_names, labels)

def
compare_samples_across_faults(
sample_idx=0):
    plt.figure(figsize=(12,5))

    unique_labels =
sorted(set(labels))  # all
faults + IDLE

    for lbl in unique_labels:
        # get all indices of
this label
        indices = [i for i, l
in enumerate(labels) if l ==
lbl]
```

```
        if len(indices) >
sample_idx:
            i =
indices[sample_idx]  # pick
same sample position

            signal = data[i]
            mag =
np.sqrt(signal[:,0]**2 +
signal[:,1]**2 +
signal[:,2]**2)

            name = "IDLE" if
lbl == 0 else f"Fault {lbl}"

            plt.plot(mag,
label=name, alpha=0.7)

    plt.legend()
    plt.title(f"Comparison
Across Faults (Sample
#{sample_idx})")
    plt.xlabel("Time Index")
    plt.ylabel("Magnitude
(g)")
    plt.xlim(10000,10200)
    plt.grid()
    plt.show()

compare_samples_across_faults(
sample_idx=0)
# Run this diagnostic cell to
understand your energy
distribution
import numpy as np

def compute_magnitude(signal):
    return
np.sqrt(signal[:,0]**2 +
signal[:,1]**2 +
signal[:,2]**2)

# Check the first few files
for i in range(min(3,
len(data))):
    signal = data[i]
    mag =
compute_magnitude(signal)

    win = int(0.5 * 1000)
    n_windows = len(mag) //
win
    energy = np.array([
        np.sqrt(np.mean(mag[i*
win:(i+1)*win]**2))
        for i in
range(n_windows)
    ])
    print(f"File:
{file_names[i]}")
    print(f"  Energy
min:    {energy.min():.4f}")
    print(f"  Energy
max:    {energy.max():.4f}")
    print(f"  Energy
mean:   {energy.mean():.4f}")
    print(f"  5th
pct:       {np.percentile(ener
gy, 5):.4f}")
    print(f"  50th
pct:      {np.percentile(energ
y, 50):.4f}")
    print(f"  95th
pct:      {np.percentile(energ
y, 95):.4f}")
    print(f"  3x noise floor:
{3 * np.percentile(energy,
5):.4f}")
    print()
import numpy as np

SAMPLING_RATE = 1000  # Hz

def compute_magnitude(signal):
    return
np.sqrt(signal[:,0]**2 +
signal[:,1]**2 +
signal[:,2]**2)

def
detect_stable_region(signal,
                             wind
ow_sec=0.5,
                             star
t_buffer_sec=10,
                             end_
buffer_sec=5,
                             min_
region_sec=10):

    mag =
compute_magnitude(signal)  #
original magnitude, untouched

    # Remove DC offset ONLY
for detection purposes — not
stored, not returned
    mag_ac = mag -
np.mean(mag)
    mag_ac = np.abs(mag_ac)

    win = int(window_sec *
SAMPLING_RATE)
    n_windows = len(mag_ac) //
win
```

```
    energy = np.array([
        np.sqrt(np.mean(mag_ac
[i*win:(i+1)*win]**2))
        for i in
range(n_windows)
    ])

    e_min = energy.min()
    e_max = energy.max()
    e_range = e_max - e_min

    if e_range < 0.05 * e_max:
        print("  → Engine
running throughout, using full
signal with buffers only.")
        start_sample =
int(start_buffer_sec *
SAMPLING_RATE)
        end_sample   =
len(signal) -
int(end_buffer_sec *
SAMPLING_RATE)
        return signal[max(0,
start_sample) :
min(len(signal), end_sample)]

    threshold = e_min + 0.30 *
e_range

    active = energy >
threshold

    if not np.any(active):
        print("  WARNING: No
active region found. Using
full signal.")
        return signal

    first_active =
np.argmax(active)
    last_active  = len(active)
- 1 - np.argmax(active[::-1])

    start_sample =
first_active * win +
int(start_buffer_sec *
SAMPLING_RATE)
    end_sample   =
(last_active + 1) * win -
int(end_buffer_sec *
SAMPLING_RATE)

    start_sample = max(0,
start_sample)
    end_sample   =
min(len(signal), end_sample)
    min_samples =
int(min_region_sec *
SAMPLING_RATE)
    if end_sample -
start_sample < min_samples:
        print(f"  WARNING:
Region too short
({(end_sample-
start_sample)/SAMPLING_RATE:.1
f}s). Returning full signal.")
        return signal

    duration = (end_sample -
start_sample) / SAMPLING_RATE
    print(f"  → Stable region:
{duration:.1f}s")

    # Return the ORIGINAL
signal slice — no modification
to values
    return
signal[start_sample:end_sample
]

# --- Apply to all data ---
processed_data = []

for i, signal in
enumerate(data):
    print(f"Processing:
{file_names[i]}")
    stable =
detect_stable_region(signal)
    duration = len(stable) /
SAMPLING_RATE
    print(f"  → Stable region:
{duration:.1f}s\n")
    processed_data.append(stab
le)

def
plot_all_processed_stacked(pro
cessed_data, labels,
file_names, fs=1000):
    n = len(processed_data)
    plt.figure(figsize=(12,
2*n))

    for i in range(n):
        signal =
processed_data[i]
        mag =
np.sqrt(signal[:,0]**2 +
signal[:,1]**2 +
signal[:,2]**2)

        duration = len(mag) /
fs  # in seconds
```

```
        # t =
np.arange(len(mag)) / fs

        plt.subplot(n, 1, i+1)
        plt.plot(range(len(mag
)), mag)
        plt.title(f"{file_name
s[i]} | Label: {labels[i]} |
Duration: {duration:.1f}s",
fontsize=8)
        plt.yticks([])
        plt.xlabel("Time
Index", fontsize=7)

    plt.tight_layout()
    plt.show()

plot_all_processed_stacked(pro
cessed_data, labels,
file_names)

def plot_before_after(idx):
    original = data[idx]
    processed =
processed_data[idx]

    mag1 =
compute_magnitude(original)
    mag2 =
compute_magnitude(processed)

    plt.figure(figsize=(12,5))

    plt.plot(mag1,
label="Original", alpha=0.4)
    plt.plot(range(len(mag2)),
mag2, label="Stable Region",
alpha=0.9)

    plt.legend()
    plt.title(f"Before vs
After | {file_names[idx]}")
    plt.grid()
    plt.show()

plot_before_after(21)
import matplotlib.pyplot as
plt

def plot_before_after(idx):
    original = data[idx]
    processed =
processed_data[idx]

    mag1 =
compute_magnitude(original)
    mag2 =
compute_magnitude(processed)

    plt.figure(figsize=(12,5))

    plt.plot(mag1,
label="Original", alpha=0.4)
    plt.plot(range(len(mag2)),
mag2, label="Processed",
alpha=0.9)

    plt.legend()
    plt.title(f"Before vs
After | {file_names[idx]}")
    plt.grid()
    plt.show()

# test
plot_before_after(0)
plot_before_after(11)


#
==============================
==============================
# Signal Segmentation: Non-
Overlapping 1s Windows
#
==============================
==============================

fs = 1000
window_size = 1 * fs  # 1
second = 1000 samples

dataset = []       # list of
windows, each shape (1000, 3)
dataset_labels = []  #
corresponding label for each
window
dataset_files  = []  # which
file each window came from
(for split later)

for i, signal in
enumerate(processed_data):

    n_windows = len(signal) //
window_size

    for w in range(n_windows):
        start = w *
window_size
        end   = start +
window_size
        window =
signal[start:end]
        dataset.append(window)
        dataset_labels.append(
labels[i])
```

```
        dataset_files.append(f
ile_names[i])

print(f"Total windows   :
{len(dataset)}")
print(f"Window shape    :
{dataset[0].shape}")
print(f"Unique labels   :
{sorted(set(dataset_labels))}"
)

#
==============================
==============================
# Dataset Summary: Windows Per
Class
#
==============================
==============================

import pandas as pd
from collections import
Counter

count =
Counter(dataset_labels)

summary = pd.DataFrame({
    "Label"      :
list(count.keys()),
    "Class Name" : ["IDLE" if
l == 0 else f"Fault {l}" for l
in count.keys()],
    "Windows"    :
list(count.values())
}).sort_values("Label").reset_
index(drop=True)

print(summary.to_string(index=
False))
print(f"\nTotal windows:
{sum(count.values())}")

# Bar plot
import matplotlib.pyplot as
plt

plt.figure(figsize=(10, 4))
plt.bar(summary["Class Name"],
summary["Windows"])
plt.title("Number of Windows
Per Class")
plt.xlabel("Class")
plt.ylabel("Window Count")
plt.xticks(rotation=45)
plt.grid(axis="y")
plt.tight_layout()
plt.show()

#
==============================
==============================
# Windows Per File (to check
consistency within each class)
#
==============================
==============================

file_count =
Counter(dataset_files)

file_summary = pd.DataFrame({
    "File"    :
list(file_count.keys()),
    "Label"   :
[labels[file_names.index(f)]
for f in file_count.keys()],
    "Windows" :
list(file_count.values())
}).sort_values(["Label",
"File"]).reset_index(drop=True
)

print(file_summary.to_string(i
ndex=False))
import random
import numpy as np
import matplotlib.pyplot as
plt
from scipy.signal import welch

fs            = 1000
unique_labels =
sorted(set(dataset_labels))
n_classes     =
len(unique_labels)

fig, axes = plt.subplots(
    n_classes, 2,
    figsize=(12, 2.2 *
n_classes),
    dpi=150
)

plt.rcParams.update({
    "font.family"  : "serif",
    "font.size"    : 9,
    "axes.linewidth": 0.8,
    "axes.grid"    : True,
    "grid.linewidth": 0.4,
    "grid.alpha"   : 0.5,
    "lines.linewidth": 0.9,
})

for row, lbl in
enumerate(unique_labels):
```

```
    indices    = [i for i, l
in enumerate(dataset_labels)
if l == lbl]
    idx        =
random.choice(indices)
    window     = dataset[idx]

    # --- same as your old
code, just magnitude computed
first ---
    mag =
np.sqrt(window[:,0]**2 +
window[:,1]**2 +
window[:,2]**2)
    t   = np.arange(len(mag))
/ fs          # seconds (same
as your old code)

    f, pxx = welch(mag, fs=fs,
nperseg=256)  # no noverlap,
same as your old code

    class_name = "IDLE
(Healthy)" if lbl == 0 else
f"Fault {lbl}"

    # ---- Time series ----
    ax_t = axes[row, 0]
    ax_t.plot(t, mag,
color="#2166ac",
linewidth=0.8)
    ax_t.set_xlim(0, 1)
    ax_t.set_ylabel("Magnitude
(g)", fontsize=8)
    ax_t.set_title(f"{class_na
me} — Time Series",
fontsize=9, fontweight="bold",
loc="left")
    ax_t.tick_params(labelsize
=7)

    if row == n_classes - 1:
        ax_t.set_xlabel("Time
(s)", fontsize=8)
    else:
        ax_t.set_xticklabels([
])

    # ---- PSD ----
    ax_f = axes[row, 1]
    ax_f.plot(f, pxx,
color="#d6604d",
linewidth=0.8)  # linear
scale, same as your old code
    ax_f.set_xlim(0, 200)
    ax_f.set_ylabel("Power
(g²/Hz)", fontsize=8)
    ax_f.set_title(f"{class_na
me} — PSD (Welch)",
fontsize=9, fontweight="bold",
loc="left")
    ax_f.tick_params(labelsize
=7)

    if row == n_classes - 1:
        ax_f.set_xlabel("Frequ
ency (Hz)", fontsize=8)
    else:
        ax_f.set_xticklabels([
])

fig.suptitle(
    "Representative Vibration
Windows — Time Series and
Power Spectral Density\n"
    "One randomly selected 1-
second window per fault
class",
    fontsize=10,
    fontweight="bold",
    y=1.001
)

plt.tight_layout()

plt.savefig("window_visualizat
ion.pdf", dpi=300,
bbox_inches="tight",
format="pdf")
plt.savefig("window_visualizat
ion.png", dpi=300,
bbox_inches="tight")

plt.show()
print("Figure saved as
window_visualization.pdf and
window_visualization.png")

#
==============================
==============================
# Feature Extraction
#
==============================
==============================

import numpy as np
from scipy.signal import welch
from scipy.stats import skew,
kurtosis

fs = 1000

def extract_features(window):
    """
```

```
    Extract 18 time and
frequency domain features
    from a 1-second vibration
window.

    Parameters
    ----------
    window : np.array, shape
(1000, 3)
        Raw 3-axis vibration
window (Z, X, Y)

    Returns
    -------
    features : list of 18
floats
    """

    # --- Magnitude ---
    mag =
np.sqrt(window[:,0]**2 +
window[:,1]**2 +
window[:,2]**2)

    features = []

    # ------------------------
------------------------
    # TIME DOMAIN
    # ------------------------
------------------------

    rms        =
np.sqrt(np.mean(mag**2))
    var        = np.var(mag)
    std        = np.std(mag)
    peak       =
np.max(np.abs(mag))
    ptp        =
np.ptp(mag)
      # peak-to-peak
    sk         = skew(mag)
    kurt       = kurtosis(mag)
    crest      = peak / rms if
rms != 0 else 0
    mean_abs   =
np.mean(np.abs(mag))
    shape      = rms /
mean_abs if mean_abs != 0 else
0
    impulse    = peak /
mean_abs if mean_abs != 0 else
0

    features.extend([rms, var,
std, peak, ptp, sk, kurt,
crest, shape, impulse])

    # ------------------------
------------------------
    # FREQUENCY DOMAIN
    # ------------------------
------------------------

    f, pxx = welch(mag, fs=fs,
nperseg=256)

    dom_freq          =
f[np.argmax(pxx)]
    spectral_centroid =
np.sum(f * pxx) / np.sum(pxx)
if np.sum(pxx) != 0 else 0
    spectral_variance =
np.sum(((f -
spectral_centroid)**2) * pxx)
/ np.sum(pxx) if np.sum(pxx)
!= 0 else 0

    band1 = np.sum(pxx[(f >=
0)   & (f < 20)])
    band2 = np.sum(pxx[(f >=
20)  & (f < 50)])
    band3 = np.sum(pxx[(f >=
50)  & (f < 100)])
    band4 = np.sum(pxx[(f >=
100) & (f < 200)])
    total_energy = np.sum(pxx)

    features.extend([
        dom_freq,
        spectral_centroid,
        spectral_variance,
        band1,
        band2,
        band3,
        band4,
        total_energy
    ])

    return features

# ---- Feature names (for
reference and plotting later)
----
feature_names = [
    "RMS", "Variance", "Std
Dev", "Peak", "Peak-to-Peak",
    "Skewness", "Kurtosis",
"Crest Factor", "Shape
Factor", "Impulse Factor",
    "Dominant Freq", "Spectral
Centroid", "Spectral
Variance",
    "Band 0-20Hz", "Band 20-
50Hz", "Band 50-100Hz", "Band
100-200Hz",
```

```
    "Total Spectral Energy"
]
# ---- Apply to all windows ----
feature_list = []

for i, window in
enumerate(dataset):
    feats =
extract_features(window)
    feature_list.append(feats)

X = np.array(feature_list)
y = np.array(dataset_labels)
files =
np.array(dataset_files)

print(f"Feature matrix shape :
{X.shape}")
print(f"Label vector shape   :
{y.shape}")
print(f"Number of features   :
{len(feature_names)}")

import matplotlib.pyplot as
plt
import pandas as pd

plt.rcParams.update({
    "font.family"   : "serif",
    "font.size"     : 9,
    "axes.linewidth": 0.8,
    "axes.grid"     : True,
    "grid.linewidth": 0.4,
    "grid.alpha"    : 0.5,
})

# --- Define df_features
before use ---
df_features = pd.DataFrame(X,
columns=feature_names)
df_features["Label"] = y
df_features["Class"] = ["IDLE"
if l == 0 else f"Fault {l}"
for l in y]
# ----------------------------
----------

# Plot first 6 most diagnostic
features
plot_features = ["RMS",
"Kurtosis", "Crest Factor",
                 "Dominant
Freq", "Band 50-100Hz", "Total
Spectral Energy"]

fig, axes = plt.subplots(2, 3,
figsize=(12, 6), dpi=150)
axes = axes.flatten()

unique_labels = sorted(set(y))
class_names   = ["IDLE" if l
== 0 else f"Fault {l}" for l
in unique_labels]

for i, feat in
enumerate(plot_features):
    ax = axes[i]
    data_per_class =
[df_features[df_features["Labe
l"] == l][feat].values
                       for l in
unique_labels]
    ax.boxplot(data_per_class,
labels=class_names,
patch_artist=True,
                boxprops=dict(f
acecolor="#cce5ff",
color="#2166ac"),
                medianprops=dic
t(color="#d6604d",
linewidth=1.5),
                whiskerprops=di
ct(color="#2166ac"),
                capprops=dict(c
olor="#2166ac"),
                flierprops=dict
(marker="o", markersize=2,
color="#888888"))
    ax.set_title(feat,
fontsize=9, fontweight="bold",
loc="left")
    ax.set_xticklabels(class_n
ames, rotation=45, fontsize=7)
    ax.tick_params(labelsize=7
)

fig.suptitle(
    "Feature Distributions
Across Fault Classes",
    fontsize=10,
    fontweight="bold",
    y=1.01
)

plt.tight_layout()
plt.savefig("feature_distribut
ions.pdf", dpi=300,
bbox_inches="tight",
format="pdf")
plt.savefig("feature_distribut
ions.png", dpi=300,
bbox_inches="tight")
plt.show()
```

```
print("Figure saved as
feature_distributions.pdf and
feature_distributions.png")

# ---- Sanity check: feature
summary per class ----
import pandas as pd

df_features = pd.DataFrame(X,
columns=feature_names)
df_features["Label"] = y
df_features["Class"] = ["IDLE"
if l == 0 else f"Fault {l}"
for l in y]

# Get the summary DataFrame
feature_summary_df =
df_features.groupby("Class")[f
eature_names].mean().round(4)

print(feature_summary_df.to_st
ring())

# Save to Excel
excel_filename =
"feature_summary_per_class.xls
x"
feature_summary_df.to_excel(ex
cel_filename)
print(f"\nFeature summary
saved to {excel_filename}")
```

APPENDIX 3: PYTHON CODE FOR ML CLASSIFICATION

```
#
==============================
==============================
# Imports
#
==============================
==============================

import numpy as np
import pandas as pd
import matplotlib.pyplot as
plt
import seaborn as sns

from sklearn.svm import SVC
from sklearn.preprocessing
import StandardScaler
from sklearn.model_selection
import GridSearchCV,
LeaveOneGroupOut
from sklearn.metrics import
(classification_report,
confusion_matrix,
                            a
ccuracy_score)

# SVM hyperparameter grid
param_grid = {
    "C"     : [0.1, 1, 10,
100],
    "gamma" : ["scale",
"auto"],
    "kernel": ["rbf"]
}

# Class names for plots
class_names = {0: "IDLE"}
class_names.update({i: f"Fault
{i}" for i in range(1, 10)})
label_names = [class_names[l]
for l in sorted(set(y))]

#
==============================
==============================
# SPLIT 1: Random Window Split
(80/20)
# Included to demonstrate data
leakage
#
==============================
==============================

from sklearn.model_selection
import train_test_split

print("=" * 60)
print("SPLIT 1: Random Window
Split (80/20)")
print("=" * 60)

X_train_r, X_test_r,
y_train_r, y_test_r =
train_test_split(
    X, y,
    test_size=0.2,
    random_state=42,
    stratify=y
)

# Scale
scaler_r  = StandardScaler()
X_train_r =
scaler_r.fit_transform(X_train
_r)
X_test_r  =
scaler_r.transform(X_test_r)

# GridSearch
grid_r = GridSearchCV(SVC(),
param_grid, cv=5,
scoring="accuracy", n_jobs=-1)
grid_r.fit(X_train_r,
y_train_r)

y_pred_r =
grid_r.predict(X_test_r)
acc_r    =
accuracy_score(y_test_r,
y_pred_r)

print(f"\nBest parameters :
{grid_r.best_params_}")
print(f"Test accuracy   :
{acc_r*100:.2f}%")
print("\nClassification
Report:")
print(classification_report(y_
test_r, y_pred_r,
target_names=label_names))

#
==============================
==============================
# SPLIT 2: File-Level Split
(80/20 by file)
# One file per class held out
for testing
#
==============================
==============================

print("=" * 60)
print("SPLIT 2: File-Level
Split")
```

```
print("=" * 60)

unique_files  =
np.unique(files)
unique_labels_list =
sorted(set(y))

train_idx = []
test_idx  = []

# For each class, hold out the
last file alphabetically as
test
for lbl in unique_labels_list:
    class_files =
sorted(set(files[y == lbl]))
    test_file   =
class_files[-1]   # hold out
last file

    for i in range(len(y)):
        if y[i] == lbl:
            if files[i] ==
test_file:
                test_idx.appen
d(i)
            else:
                train_idx.appe
nd(i)

X_train_f = X[train_idx]
X_test_f  = X[test_idx]
y_train_f = y[train_idx]
y_test_f  = y[test_idx]

# Scale
scaler_f  = StandardScaler()
X_train_f =
scaler_f.fit_transform(X_train
_f)
X_test_f  =
scaler_f.transform(X_test_f)

# GridSearch
grid_f = GridSearchCV(SVC(),
param_grid, cv=5,
scoring="accuracy", n_jobs=-1)
grid_f.fit(X_train_f,
y_train_f)

y_pred_f =
grid_f.predict(X_test_f)
acc_f    =
accuracy_score(y_test_f,
y_pred_f)

print(f"\nBest parameters :
{grid_f.best_params_}")
print(f"Test accuracy   :
{acc_f*100:.2f}%")
print("\nClassification
Report:")
print(classification_report(y_
test_f, y_pred_f,
target_names=label_names))

#
==============================
==============================
# SPLIT 3: Leave-One-File-Out
Cross Validation (LOFO-CV)
# Most rigorous evaluation for
file-based datasets
#
==============================
==============================

print("=" * 60)
print("SPLIT 3: Leave-One-
File-Out Cross Validation")
print("=" * 60)

# Encode file names as group
integers for LeaveOneGroupOut
file_list    =
np.unique(files)
file_to_int  = {f: i for i, f
in enumerate(file_list)}
groups       =
np.array([file_to_int[f] for f
in files])

logo         =
LeaveOneGroupOut()
all_y_true   = []
all_y_pred   = []
fold_accs    = []

n_folds = logo.get_n_splits(X,
y, groups)
print(f"Total folds:
{n_folds}\n")
for fold, (train_idx,
test_idx) in
enumerate(logo.split(X, y,
groups)):

    X_tr, X_te = X[train_idx],
X[test_idx]
    y_tr, y_te = y[train_idx],
y[test_idx]

    # Scale
    scaler_l =
StandardScaler()
```

```
    X_tr     =
scaler_l.fit_transform(X_tr)
    X_te     =
scaler_l.transform(X_te)

    # GridSearch inside each
fold
    grid_l =
GridSearchCV(SVC(),
param_grid, cv=5,
scoring="accuracy", n_jobs=-1)
    grid_l.fit(X_tr, y_tr)

    y_pred_l =
grid_l.predict(X_te)
    fold_acc =
accuracy_score(y_te, y_pred_l)
    fold_accs.append(fold_acc)

    all_y_true.extend(y_te)
    all_y_pred.extend(y_pred_l
)

    test_file =
file_list[fold]
    print(f"Fold {fold+1:3d} |
File: {test_file:60s} | Acc:
{fold_acc*100:.2f}%")

all_y_true =
np.array(all_y_true)
all_y_pred =
np.array(all_y_pred)

acc_lofo =
accuracy_score(all_y_true,
all_y_pred)

print(f"\nMean fold
accuracy  :
{np.mean(fold_accs)*100:.2f}%"
)
print(f"Std fold accuracy   :
{np.std(fold_accs)*100:.2f}%")
print(f"Overall accuracy    :
{acc_lofo*100:.2f}%")
print("\nClassification
Report:")
print(classification_report(al
l_y_true, all_y_pred,
target_names=label_names))

#
==============================
==============================
# Confusion Matrices — All
Three Splits Separately
#
==============================
==============================

plt.rcParams.update({
    "font.family"  : "serif",
    "font.size"    : 9,
    "axes.linewidth": 0.8,
})

split_results = [
    (y_test_r,  y_pred_r,  f"S
plit 1: Random Window\n(Acc:
{acc_r*100:.1f}%) -- High
Leakage"),
    (y_test_f,  y_pred_f,  f"S
plit 2: File-Level\n(Acc:
{acc_f*100:.1f}%) -- Medium
Reliability"),
    (all_y_true, all_y_pred,
f"Split 3: LOFO-CV\n(Acc:
{acc_lofo*100:.1f}%) -- High
Reliability"),
]

for i, (y_t, y_p, title) in
enumerate(split_results):
    fig, ax =
plt.subplots(figsize=(7, 6),
dpi=150)
    cm = confusion_matrix(y_t,
y_p, labels=sorted(set(y)))
    sns.heatmap(
        cm,
        annot=True,
        fmt="d",
        cmap="Blues",
        xticklabels=label_name
s,
        yticklabels=label_name
s,
        linewidths=0.4,
        linecolor="white",
        ax=ax,
        cbar=False
    )
    ax.set_title(title,
fontsize=10,
fontweight="bold", y=1.05)
    ax.set_xlabel("Predicted
Label", fontsize=9)
    ax.set_ylabel("True
Label", fontsize=9)
    ax.tick_params(labelsize=8
)
    plt.setp(ax.get_xticklabel
s(), rotation=45, ha="right")
```

```
    plt.setp(ax.get_yticklabel
s(), rotation=0)

    fig_title = f"Confusion
Matrix -
{title.split('(')[0].strip().r
eplace('\n', ' ')}"
    plt.tight_layout(rect=[0,
0, 1, 0.95])

    # Save each figure
separately
    filename_prefix =
title.split(':')[0].lower().re
place(' ',
'_').replace('\n','')
    plt.savefig(f"{filename_pr
efix}_confusion_matrix.pdf",
dpi=300, bbox_inches="tight",
format="pdf")
    plt.savefig(f"{filename_pr
efix}_confusion_matrix.png",
dpi=300, bbox_inches="tight")
    plt.show()
    print(f"Figure saved as
{filename_prefix}_confusion_ma
trix.pdf and
{filename_prefix}_confusion_ma
trix.png")


#
==============================
==============================
# Summary Comparison Table
#
==============================
==============================

summary = pd.DataFrame({
    "Split Strategy"    : [
        "Random Window Split",
        "File-Level Split",
        "LOFO Cross-
Validation"
    ],
    "Accuracy (%)"      : [
        round(acc_r * 100, 2),
        round(acc_f * 100, 2),
        round(acc_lofo * 100,
2)
    ],
    "Leakage Risk"      :
["High", "None", "None"],
    "Reliability"       :
["Low", "Medium", "High"],
    "Notes"             : [
        "Windows from same
file in train and test",
        "Single held-out file
per class",
        "Every file tested
exactly once"
    ]
})

print(summary.to_string(index=
False))
```

APPENDIX 4: VIBRATION FEATURES

Appendix 4: Vibration Features Data Set

Class	Unit	Fault 1	Fault 3	Fault 4	Fault 5	Fault 6	Fault 7	Fault 8	Fault 9	IDLE
RMS	g	1.63	1.38	1.44	1.52	1.40	1.47	1.45	1.46	1.34
Variance	g^2	0.76	0.18	0.22	0.26	0.20	0.23	0.22	0.22	0.16
Std Dev	g	0.87	0.43	0.47	0.51	0.45	0.48	0.47	0.47	0.39
Peak	g	7.59	3.13	3.20	3.43	3.33	3.23	3.18	3.19	3.01
Peak-to-Peak	g	7.51	2.63	2.61	2.88	2.84	2.63	2.58	2.59	2.43
Skewness	Dimensionless	1.63	1.33	1.29	1.29	1.40	1.30	1.28	1.25	1.43
Kurtosis	Dimensionless	6.01	1.47	1.16	1.20	1.94	1.13	1.10	1.02	2.11
Crest Factor	Dimensionless	4.66	2.27	2.22	2.26	2.38	2.20	2.18	2.18	2.21
Shape Factor	Dimensionless	1.18	1.05	1.06	1.06	1.06	1.06	1.06	1.06	1.05
Impulse Factor	Dimensionless	5.51	2.39	2.34	2.40	2.52	2.33	2.31	2.31	2.32
Dominant Freq	Hz	42.77	27.34	27.34	27.34	27.34	27.34	27.34	27.34	27.34
Spectral Centroid	Hz	131.61	92.34	88.26	82.00	98.42	92.53	89.75	87.38	112.57
Spectral Variance	Hz^2	18447	12324	11903	9210	14969	13139	12345	11586	15433

Appendix 4: Vibration Features Data Set

Class	Unit	Fault 1	Fault 3	Fault 4	Fault 5	Fault 6	Fault 7	Fault 8	Fault 9	IDLE
Band 0-20 Hz	g^2	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Band 20-50 Hz	g^2	0.11	0.02	0.02	0.03	0.02	0.02	0.02	0.02	0.02
Band 50-100 Hz	g^2	0.01	0.02	0.02	0.03	0.02	0.02	0.02	0.02	0.01
Band 100-200 Hz	g^2	0.02	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Total Spectral Energy	g^2	0.19	0.05	0.06	0.07	0.05	0.06	0.06	0.06	0.04

Dipak Sharma

Final Report to aiplagcheck.pdf

Tribhuvan University

Document Details

Submission ID
trn:oid:::3117:587791146

Submission Date
May 8, 2026, 5:39 AM GMT+5:45

Download Date
May 8, 2026, 5:51 AM GMT+5:45

File Name
Final Report to aiplagcheck.pdf

File Size
1.7 MB

66 Pages

15,275 Words

94,744 Characters

6% Overall Similarity

The combined total of all matches, including overlapping sources, for each database.

Filtered from the Report

- Bibliography
- Quoted Text
- Cited Text
- Small Matches (less than 8 words)

Custom Section Exclusions

{titlesCount} Section Titles, {keywordsCount} Keywords

Section title	No. of Section Starters	Section Starters			
"Acknowledgements"	4	Acknowledgements	Acknowledgement	Acknowledgment	Acknowledgments

Match Groups

80 Not Cited or Quoted 6%
Matches with neither in-text citation nor quotation marks

0 Missing Quotations 0%
Matches that are still very similar to source material

0 Missing Citation 0%
Matches that have quotation marks, but no in-text citation

0 Cited and Quoted 0%
Matches with in-text citation present, but no quotation marks

Top Sources

5% Internet sources
2% Publications
0% Submitted works (Student Papers)

Integrity Flags

0 Integrity Flags for Review

No suspicious text manipulations found.

Our system's algorithms look deeply at a document for any inconsistencies that would set it apart from a normal submission. If we notice something strange, we flag it for you to review.

A Flag is not necessarily an indicator of a problem. However, we'd recommend you focus your attention there for further review.

Match Groups

- **80** Not Cited or Quoted 6%
  Matches with neither in-text citation nor quotation marks
- **0** Missing Quotations 0%
  Matches that are still very similar to source material
- **0** Missing Citation 0%
  Matches that have quotation marks, but no in-text citation
- **0** Cited and Quoted 0%
  Matches with in-text citation present, but no quotation marks

Top Sources

5% Internet sources
2% Publications
0% Submitted works (Student Papers)

Top Sources

The sources with the highest number of matches within the submission. Overlapping sources will not be displayed.

1	Internet	etd.aau.edu.et <1%
2	Internet	thesesjournal.com <1%
3	Internet	www.mdpi.com <1%
4	Internet	elibrary.tucl.edu.np <1%
5	Internet	core.ac.uk <1%
6	Internet	dl.dropboxusercontent.com <1%
7	Internet	link.springer.com <1%
8	Internet	mdpi-res.com <1%
9	Internet	www.irjmets.com <1%
10	Publication	Englesbe, Alexander C.. "Charge Dynamics in Femtosecond Laser Filaments", Univ... <1%

Vibration-Based Analysis of Sensor and Actuator Faults on an EURO 6 Diesel Engine

Dipak Sharma [1], Laxman Poudel [1] Surya Prasad Adhikari [1] Amrit Tiwari [2]

Abstract:

Diesel engines are indispensable in energy generation domains forming a clutch to energy and production. Their complex anatomy and modern development involving sophisticated mechanical and electronic control make their operation logic intricate and increase the fault probability which are often out of reach to traditional symptoms and judgment based maintenance technique. This paper presents a vibration-based fault analysis framework for a EURO 6-compliant ISUZU 4KH1 diesel engine. Triaxial MEMS sensor is used to acquire vibration data under healthy idle condition and eight different sensor/actuator based faults simulated by intelligent fault insertion module. Fast Fourier Transform (FFT), Power Spectral Density (PSD) and other six features are extracted from the data and studies across all scenario. It is found that each faulty scenario have distinct and repeatable vibration features as compared to the healthy idle baseline operation. Dominant frequency up-shift, RMS amplitude growth, kurtosis rise, band wise spectral energy variation, crest factor change, total spectral energy variations are the notable discerning features across different faults.Fault induced by Accelerator Pedal Position (APP) Sensor failure is the most vibrant faults with distinct changes in many of the features while injector fault is associated with sub-harmonic emergence and PSD band redistribution distinguish EGR valve faults among others. The findings of this research forms a robust base for vibration-based predictive maintenance approach where current studies on electronic sensor and actuator induced fault propagation in modern EURO 6 diesel engine systems are limited.

Keywords:

Diesel engine fault diagnosis, FFT, PSD, kurtosis, EURO 6, predictive maintenance, MEMS accelerometer

[1] *Department of Mechanical and Aerospace Engineering, Pulchowk Campus, Institute of Engineering, Tribhuvan University, Lalitpur, Nepal* [2] *Department of Mechanical and Aerospace Engineering, University of Houston*

[1] 080msmde009.dipak@pcampus.edu.np