# Chapter 3

# Methodology

## 3.1 Preface

This chapter presents the methodological framework adopted to address the research gaps identified in Chapter 2 and to achieve the objectives outlined in Chapter 1. The methodologies are organized across four key components: (1) multi-directional vibration analysis using VMD and FFT to characterize vehicle-induced responses over various hump profiles; (2) development of RVEL and PCVE frameworks to standardize and quantify vehicle-specific vibration emissions; (3) machine learning based vehicle classification using vibration-derived features; and (4) GPS-augmented traffic forecasting and route optimization incorporating. All components are unified through a robust signal processing pipeline, ensuring consistent input to subsequent analyses. The experimental design, instrumentation and data acquisition strategies are detailed separately in Chapter 4.

## 3.2 Signal Preprocessing Framework

This section presents the complete preprocessing pipeline applied to the raw tri-axial vibration signals collected using a wireless accelerometer. The signals represent individual vehicle pass-by events and are inherently noisy due to background vibrations, electronic DC components and varying signal durations. The preprocessing pipeline consists of four critical stages: (i) raw signal acquisition, (ii) spectral inspection using FFT, (iii) band-pass filtering to remove irrelevant frequency components and (iv) windowing to eliminate edge transients.

### 3.2.1 Raw Signal Acquisition and Frequency Inspection via FFT

Tri-axial ground vibration signals are recorded for each vehicle pass-by event using a wireless accelerometer system mounted at predefined offsets from the pavement surface. The accelerometer captures the dynamic response in three orthogonal directions: lateral (X), longitudinal (Y) and vertical (Z), expressed as $x(t) = \{x(t), y(t), z(t)\}$. The signals are sampled at a high acquisition rate $f_s$, providing sufficient temporal resolution to capture transient vehicle-induced vibrations.

The raw vibration signals are typically contaminated with several undesirable components that obscure the physical interpretation of vehicular dynamics. These include DC offset introduced by sensor bias, which leads to non-zero mean values; low-frequency drift arising from thermal effects, ground tilt, or sensor instability; high-frequency noise caused by electronic interference, structural resonance, or mechanical jitter; and non-stationarity in the signals due to variations in event duration and triggering patterns across different vehicle types.

A Fast Fourier Transform (FFT) is performed on a fixed-duration window of the raw signal to identify the dominant frequency bands associated with vehicle-induced dynamics. This step identifies dominant spectral content while minimizing noise and irrelevant components. The discrete Fourier transform is defined as:

$$X(f) = \sum_{n=0}^{N-1} x(n)\, e^{-j2\pi f n/N} \tag{3.1}$$

where $x(n)$ is the discrete-time signal, $X(f)$ is the spectral representation and $N$ is the number of samples in the chosen window.

The frequency resolution of the spectrum is given by:

$$\Delta f = \frac{f_s}{N} \tag{3.2}$$

This frequency-domain inspection enables identification of spectral regions that are physically meaningful and those dominated by noise. The empirical spectral characteristics are subsequently used to define cutoff thresholds for band-pass filtering in the next stage. These cutoff frequencies are selected to retain vibrational components arising from tire-road-structure interaction while excluding low-frequency drift and high-frequency disturbances. This step ensures that only the dynamic components characteristic of clean vehicle passage are retained for downstream tasks such as filtering, segmentation and signal decomposition.

### 3.2.2 Band-Pass Filtering and Noise Removal

After initial spectral inspection, a band-pass filtering stage is applied to suppress frequency components not associated with vehicle-induced dynamics. This step aims to retain only those signal frequencies that fall within the empirically identified band of interest, as observed from the FFT analysis, while attenuating components due to DC drift, environmental noise and sensor artifacts.

A digital band-pass filter is designed and implemented to achieve this objective. Among various filter types, a second-order Butterworth filter is selected for its desirable characteristics: a maximally flat response in the passband and a smooth transition that avoids abrupt signal distortions. The Butterworth filter's cutoff frequencies are selected based on empirically observed vibrational energy bands from FFT analysis. These cutoff frequencies are normalized with respect to the sampling rate $f_s$ before implementation.

The discrete-time band-pass filter follows the standard difference equation form:

$$y(n) = \sum_{k=0}^{M} b_k x(n-k) - \sum_{k=1}^{N} a_k y(n-k), \tag{3.3}$$

Where $x(n)$ is the input signal, $y(n)$ is the output signal and $b_k, a_k$ are the filter coefficients determined by the filter design parameters. The order of the filter (typically set to 2 for stability and efficiency) and the desired passband are specified during coefficient generation using standard digital signal processing libraries.

This filtering operation eliminates the non-informative spectral content outside the desired band, preserving only the frequency components relevant to the vehicle-road interaction. However, due to convolution and initial-state conditions, one side effect of digital filtering is the introduction of transient distortions at the signal edges, particularly at the beginning. These distortions must be removed before further processing and are handled explicitly in the subsequent windowing stage.

### 3.2.3 Windowing to Remove Transients of the Signal

Digital filtering introduces initial transients due to convolution artifacts, which are mitigated through hard windowing that discards unstable segments. A rectangular window function is applied to discard the initial portion of the signal where transient effects are dominant. The window is defined as follows:

$$w(n) = \begin{cases} 0, & 0 \leq n < N_{\text{cut}} \\ 1, & n \geq N_{\text{cut}} \end{cases} \tag{3.4}$$

Where $N_{\text{cut}}$ is the empirically chosen cutoff index beyond which the signal is assumed to be stable and physically meaningful. This index is selected based on visual inspection of several signals and prior knowledge of filter-induced distortions.

The resulting transient-free signal is computed as:

$$x_{\text{windowed}}(n) = x_{\text{filtered}}(n) \cdot w(n) \tag{3.5}$$

This operation preserves the signal portion corresponding to steady-state vehicle-induced vibrations while suppressing filter initialization effects. The windowed signal is then passed on to the segmentation and feature extraction stages, ensuring clean and consistent data for subsequent analysis.

### 3.2.4 Energy-Based Segmentation for Uniformity of the Signals

Although filtering and transient removal significantly improve signal quality, the resultant vibration records often remain non-uniform in length. This variation arises due to manual triggering during data acquisition and the differing durations of vehicle pass-by events. To ensure consistency across all samples, extracting a standardized segment that captures the most energetic portion of the vehicle-induced response is essential.

To achieve this, a sliding window-based segmentation strategy is adopted. A fixed-duration window, defined by a standardized number of samples $N_w$ based on the system's sampling frequency, is shifted along the length of each preprocessed signal. At each step, the vibrational energy of the segment is computed as:

$$E_k = \sum_{n=0}^{N_w-1} x_k(n)^2 \tag{3.6}$$

Here, $x_k(n)$ denotes the signal values within the $k^{\text{th}}$ window and $E_k$ is the total vibrational energy contained in that window. The window that yields the maximum energy is assumed to represent best the core vibrational response associated with the vehicle's passage.

The optimal window index is identified as:

$$k^* = \arg\max_k E_k \tag{3.7}$$

The corresponding segment is then defined as:

$$x_{k^*}(n) = x(n + k^*), \quad n = 0, 1, \ldots, N_w - 1 \tag{3.8}$$

Here, $x_{k^*}(n)$ represents the portion of the original preprocessed signal starting at index $k^*$, containing exactly $N_w$ samples and corresponding to the highest energy window. This segment is retained for all subsequent steps, including feature extraction, signal decomposition and classification.

This segmentation process ensures that every vibration sample used in the study is temporally aligned, energetically representative and standardized in length properties essential for robust comparative analysis across vehicle types and directions. Ultimately, this step transforms variable-length field signals into a uniform, high-quality dataset optimized for downstream machine learning and signal processing applications.

## 3.3 Variational Mode Decomposition for Signal Analysis

This study employs Variational Mode Decomposition (VMD), a robust, data-driven signal processing technique, to extract latent vibrational dynamics embedded within raw vehicle-induced signals. VMD decomposes complex, non-stationary signals into band-limited intrinsic mode functions (IMFs), each representing oscillatory components concentrated around distinct frequency bands. Compared to Empirical Mode Decomposition (EMD), VMD avoids mode mixing and offers greater stability due to its formulation within a constrained variational framework.

### 3.3.1 Rationale for Using VMD

Traffic-induced vibrations, especially those generated during vehicle-hump interactions, exhibit broadband, nonlinear and non-stationary characteristics. Classical spectral methods, such as the Fourier or wavelet transforms, often do not resolve such dynamic content with temporal precision. VMD addresses these limitations by adaptively decomposing a signal $x(t)$ into $K$ IMFs $u_k(t)$, each centered at a specific frequency $\omega_k$. The decomposition minimizes the sum of bandwidths of all modes, subject to their collective reconstruction of the original signal:

$$\min_{\{u_k\},\{\omega_k\}} \left\{ \sum_{k=1}^{K} \left\| \partial_t \left[ \left( \delta(t) + \frac{j}{\pi t} \right) * u_k(t) \cdot e^{-j\omega_k t} \right] \right\|_2^2 \right\} \tag{3.9}$$

$$\text{subject to: } \sum_{k=1}^{K} u_k(t) = x(t) \tag{3.10}$$

The problem is solved using the Augmented Lagrangian approach via the Alternating Direction Method of Multipliers (ADMM). This allows the extraction of IMFs that are non-overlapping in frequency and localized in time, suitable for isolating vibrational signatures corresponding to different structural and vehicular interactions.

### 3.3.2 Extraction of IMFs and Dominant Mode Identification

VMD is independently applied to each directional component (X, Y and Z) of the pre-processed and segmented vibration signal. The number of decomposition modes $K$ is empirically selected based on inspection of spectral energy spread and convergence behavior. Each resulting mode $u_k(t)$ encapsulates distinct oscillatory content linked to physical events such as axle loading or structural resonance.

To identify the most informative IMF for further analysis, the energy of each mode is quantified using the RMS-based energy expression:

$$E_k = \frac{1}{N}\sum_{n=1}^{N} u_k^2(n) \tag{3.11}$$

$$k^* = \arg\max_k E_k \tag{3.12}$$

This ensures the selection of the most energetic mode representing the primary vibrational event.

### 3.3.3 Feature Extraction from VMD Modes

From each IMF, particularly the dominant one, both time-domain and frequency-domain features are extracted to describe the signal's statistical, energetic and spectral properties. These features are critical for distinguishing between different vehicle classes and their responses across varying hump profiles.

**Time-Domain Features** The following statistical features are computed from each IMF $u_k(t)$:

- **Mean:** $\mu = \frac{1}{N}\sum_{n=1}^{N} u_k(n)$
- **Variance:** $\sigma^2 = \frac{1}{N}\sum_{n=1}^{N}(u_k(n) - \mu)^2$
- **Standard Deviation:** $\sigma = \sqrt{\sigma^2}$
- **Root Mean Square (RMS):** $RMS = \sqrt{\frac{1}{N}\sum_{n=1}^{N} u_k^2(n)}$
- **Skewness:** $\frac{1}{N}\sum_{n=1}^{N}\left(\frac{u_k(n)-\mu}{\sigma}\right)^3$
- **Kurtosis:** $\frac{1}{N}\sum_{n=1}^{N}\left(\frac{u_k(n)-\mu}{\sigma}\right)^4$
- **Crest Factor:** $\mathrm{CF} = \frac{u_{\max}}{RMS}$
- **Peak-to-Peak:** $u_{\max} - u_{\min}$

- **Energy:** $E = \sum_{n=1}^{N} u_k^2(n)$
- **Shannon Entropy:** $H = -\sum_{i=1}^{M} P(u_i) \log P(u_i)$

**Frequency-Domain Features** FFT is applied to each IMF to derive spectral characteristics:

- **Dominant Frequency (DF):** $DF = \arg\max_f |U_k(f)|^2$
- **Power Spectral Density (PSD):** $PSD(f) = \frac{|U_k(f)|^2}{\Delta f}$
- **Signal-to-Noise Ratio (SNR):** $SNR = 10 \log_{10} \left( \frac{P_{\text{signal}}}{P_{\text{noise}}} \right)$

These features, extracted per IMF and per direction, form the foundation for subsequent machine learning models. They effectively capture the multidimensional vibrational behavior associated with each vehicle class and their corresponding structural interactions.

## 3.4 Vibration Equivalence Metric Formulation

This section presents the formulation of vibration equivalence metrics, specifically the Reference Vibration Emission Level (RVEL) and Passenger Car Vibration Equivalence (PCVE), to systematically quantify and normalize vehicle-induced vibrations for infrastructure and traffic assessment.

### 3.4.1 Reference Vibration Emission Levels (RVEL)

The RVEL metric captures the baseline relationship between vehicle speed and vibration level across vehicle categories. It establishes a direction-wise linear model of vibration response as a function of speed using real-world pass-by data. The model is given by:

$$V_{D,C}(S) = \alpha_{D,C} S + \beta_{D,C}, \tag{3.13}$$

where $V_{D,C}(S)$ denotes the predicted vibration metric (Max, RMS, or Average Power) for direction $D$ and vehicle class $C$ at speed $S$. The coefficients $\alpha_{D,C}$ and $\beta_{D,C}$ are obtained via least-squares regression over binned speed intervals.

**Metric Selection:** Three vibration metrics are used: Maximum Amplitude (mm/s$^2$), RMS Acceleration (mm/s$^2$) and Average Power ($\mu$W). Each metric is modeled independently across the X, Y and Z directions for every vehicle class. This multidimensional modeling enables directional interpretation of vehicle-induced vibration behavior.

**Data Averaging:** Vibration measurements are aggregated over discrete speed intervals to reduce signal noise and improve the stability of regression estimates. Representative vibration metrics are computed within each interval by averaging multiple vehicle pass-by instances. This process enhances the robustness of the linear model formulation while preserving the underlying relationship between vehicle speed and vibration response.

**Model Validation:** The quality of the linear fit for the RVEL formulation is evaluated using standard statistical metrics, including the coefficient of determination ($R^2$), root-mean-square error (RMSE) and the significance levels (p-values) of the estimated parameters. These diagnostics ensure the appropriateness of the linear model in capturing speed-dependent vibration trends across different vehicle classes and directions.

### 3.4.2 Passenger Car Vibration Equivalence (PCVE)

PCVE is defined as the ratio of the predicted vibration metric of any vehicle class $C$ to that of a passenger car at a given speed $S$:

$$\mathrm{PCVE}_{D,C}(S) = \frac{V_{D,C}(S)}{V_{D,\mathrm{Car}}(S)}. \tag{3.14}$$

This framework transforms absolute vibration levels into a common baseline relative to a car, enabling cross-class comparison.

**Directional PCVE:** Separate PCVE values are computed for the X, Y and Z directions, denoted as $\text{PCVE}_X$, $\text{PCVE}_Y$ and $\text{PCVE}_Z$.

This directional transformation allows comparison of vehicle-induced vibrations relative to a standard reference across individual axes, aiding in infrastructure impact analysis and decision-making frameworks such as route selection. The normalized PCVE metric facilitates the integration of vibrational considerations into higher-level traffic and infrastructure management systems.

## 3.5 Vibration-Based Vehicle Classification Model

This section presents the methodology adopted for vibration-based vehicle classification using tri-axial accelerometer signals. The process involves a sequential pipeline: energy-based signal segmentation, multidimensional feature extraction, class imbalance mitigation using ADASYN and stacked ensemble classification. The entire framework is mathematically grounded and computationally implemented in Python.

### 3.5.1 Energy-Based Feature Extraction from Vibration Signals

Each vibration signal is processed to extract a fixed-length segment of high informational value to ensure uniformity across the time-series data acquired from varied vehicle pass-by events. The preprocessed max energy vibration segment obtained using the energy-based segmentation method in Section 3.2.4 is used here for consistent feature extraction. The high-resolution segment ensures uniform temporal length across vehicle classes. These fixed-length segments form the foundation for consistent and directionally comparable vehicle classification.

### 3.5.2 Class Imbalance Analysis and Mitigation

In multi-class classification problems, class imbalance presents a critical challenge, particularly when the distribution of vehicle categories varies significantly in real-world traffic data. Unequal representation among classes can lead to biased classifiers, disproportionately favoring majority classes and reducing the model's generalization ability on minority class instances.

The Adaptive Synthetic Sampling (ADASYN) technique was employed to mitigate this issue. ADASYN enhances minority class representation by generating synthetic instances in feature space, particularly focusing on samples that are harder to classify. The local data density governs this adaptive generation and ensures better decision boundary learning.

The effectiveness of the ADASYN balancing technique can be quantified using the Imbalance Ratio (IR), defined as:

$$IR = \frac{N_{\text{majority}}}{N_{\text{minority}}}, \tag{3.15}$$

where $N_{\text{majority}}$ and $N_{\text{minority}}$ denote the number of samples in the largest and smallest class, respectively. A high imbalance ratio indicates a skewed dataset. Post-balancing, an ideal IR approaches unity.

Additionally, the average class size before and after balancing is computed as:

$$\text{Average Class Size} = \frac{1}{C}\sum_{i=1}^{C} N_i, \tag{3.16}$$

where $C$ is the number of vehicle classes and $N_i$ represents the count of instances in the $i^{th}$ class.

ADASYN was chosen over conventional methods like SMOTE due to its emphasis on sample difficulty, generating more data points near class boundaries with high learning complexity. This enhances classifier performance under real-world traffic conditions where rare vehicle types (e.g., tractors or e-rickshaws) are typically underrepresented.

### 3.5.3 Ensemble Learning via Stacked Classifier

A stacking ensemble framework was employed, comprising two base classifiers and a logistic regression-based meta-classifier.

- **Base Classifiers:**
  - $h_1$: Random Forest - ensemble of decision trees with bagging.
  - $h_2$: XGBoost - gradient-boosted trees offering high performance on structured data.
- **Training Strategy:**

$$h_i : \mathbf{X}_{\text{train}} \to \hat{\mathbf{y}}_{i,\text{train}}, \quad \mathbf{Z}_{\text{train}} = [\hat{\mathbf{y}}_{1,\text{train}}, \hat{\mathbf{y}}_{2,\text{train}}] \tag{3.17}$$

- **Meta Classifier:**

$$H : \mathbf{Z}_{\text{train}} \to \hat{y}_{\text{meta-train}} \tag{3.18}$$

  Final predictions:

$$\hat{y}_{\text{final}} = H([\hat{y}_{1,\text{valid}}, \hat{y}_{2,\text{valid}}]) \tag{3.19}$$

The ensemble structure leverages the strengths of both variance-reducing bagging and bias-reducing boosting methods for robust classification.

### 3.5.4 Loss Function, Regularization and Optimization

Logistic regression, as the meta-classifier, used cross-entropy loss:

$$\mathscr{L} = -\frac{1}{n}\sum_{i=1}^{n}\left[y_i \log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)\right] \tag{3.20}$$

With L2 regularization:

$$\mathscr{L}_{\text{reg}} = \mathscr{L} + \lambda \sum_{j=1}^{d} \theta_j^2 \tag{3.21}$$

This minimized overfitting and enhanced generalization.

### 3.5.5 Model Evaluation and Statistical Validation

The final model was evaluated using accuracy, precision, recall and F1-score:

$$\text{Accuracy} = \frac{\text{TP}+\text{TN}}{\text{TP}+\text{TN}+\text{FP}+\text{FN}}, \tag{3.22}$$

$$\text{Precision} = \frac{\text{TP}}{\text{TP}+\text{FP}}, \tag{3.23}$$

$$\text{Recall} = \frac{\text{TP}}{\text{TP}+\text{FN}}, \tag{3.24}$$

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3.25}$$

Repeated training on random splits yielded test accuracies $\{x_1, x_2, ..., x_N\}$. The sample mean $\bar{x}$, standard deviation $s$ and 95% confidence interval:

$$\bar{x} \pm t_{\alpha/2, N-1} \cdot \frac{s}{\sqrt{N}} \tag{3.26}$$

Were computed, along with a one-sample $t$-test:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{N}} \tag{3.27}$$

to test against a baseline accuracy $\mu_0$.

A Kernel Density Estimation (KDE) plot visualized the accuracy distribution, overlaying the mean and confidence bounds to confirm the robustness and consistency of the classification framework.

## 3.6 Traffic Forecasting and Route Optimization Framework

This section presents a comprehensive methodology for traffic volume forecasting and route optimization through a machine learning based overlay architecture. The process integrates statistical time-series modeling, kernel-based regression and decision-tree learning, augmented with spatiotemporal GPS metadata for practical real-time traffic condition inference.

#### Preprocessing and Time-Series Construction

Let $\mathscr{T}_t^s$ represent the raw vehicle count at time $t$ for a specific site $s \in \mathbb{S}$, where $\mathbb{S}$ denotes the set of all sites. To prepare the data for forecasting, we define the following preprocessing operations:

- **Detrending and Differencing:** To ensure stationarity, the first-order differenced time series is given by:

$$\Delta\mathscr{T}_t^s = \mathscr{T}_t^s - \mathscr{T}_{t-1}^s = (1 - \mathbf{B})\mathscr{T}_t^s, \tag{3.28}$$

  where $\mathbf{B}$ is the back-shift operator, such that $\mathbf{B}\mathscr{T}_t^s = \mathscr{T}_{t-1}^s$.

- **Anomaly Suppression:** Let $\mu_t$ and $\sigma_t$ be the rolling mean and standard deviation in a window $w$. An observation $\mathscr{T}_t^s$ is flagged as an outlier if:

$$|\mathscr{T}_t^s - \mu_t| > \kappa\sigma_t, \quad \text{for some } \kappa \in \mathbb{R}^+. \tag{3.29}$$

- **Missing Value Imputation:** For $t \in \mathbb{T}_{\text{missing}}$, missing entries are replaced using:

$$\mathscr{T}_t^s = \frac{1}{w} \sum_{i=t-w/2}^{t+w/2} \mathscr{T}_i^s, \tag{3.30}$$

ensuring temporal continuity.

### Time-Series Forecasting: ARIMA and SVM Models

**ARIMA$(p,d,q)$ Model:** The ARIMA model for the stationary differenced series $\mathbb{T}_t^s = \Delta^d \mathscr{T}_t^s$ is expressed as:

$$\Phi(B)\mathbb{T}_t^s = \Theta(B)\mathscr{E}_t, \tag{3.31}$$

where $\Phi(B) = 1 - \sum_{i=1}^{p} \phi_i B^i$ is the autoregressive polynomial and $\Theta(B) = 1 + \sum_{j=1}^{q} \theta_j B^j$ is the moving average polynomial. $\mathscr{E}_t$ is the white noise term with zero mean and variance $\sigma^2$.

**SVM Regression Model:** Given the input sequence $\{\mathbf{x}_i, y_i\}_{i=1}^N$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector representing the lagged time series and GPS metadata, SVM solves the following optimization:

$$\min_{\omega,b,\xi,\xi^*} \quad \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) \tag{3.32}$$

$$\text{s.t.} \quad y_i - \omega^\top \psi(\mathbf{x}_i) - b \leq \varepsilon + \xi_i, \tag{3.33}$$

$$\omega^\top \psi(\mathbf{x}_i) + b - y_i \leq \varepsilon + \xi_i^*, \tag{3.34}$$

$$\xi_i, \xi_i^* \geq 0, \tag{3.35}$$

where $\psi(\cdot)$ is the RBF kernel mapping and $\varepsilon$ is the insensitive loss margin. The kernel function is defined as:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\gamma\|\mathbf{x}_i - \mathbf{x}_j\|^2\right), \quad \gamma > 0. \tag{3.36}$$

Both linear (ARIMA) and nonlinear (SVM) models are employed to capture the temporal dynamics of traffic volume. ARIMA handles seasonality and trend components effectively, while SVM accounts for nonlinear interactions influenced by external metadata such as GPS location. While ARIMA captures temporal correlations and trends, SVM introduces flexibility to model nonlinear dependencies influenced by auxiliary inputs such as GPS metadata.

**Model Comparison:** The performance of ARIMA and SVM models is evaluated using Mean Absolute Error (MAE), Root Mean Square Error (RMSE) and Mean Absolute Percentage Error (MAPE) on a held-out test set. The choice of the final forecasting model for each site is based on the minimum RMSE.

### GPS Data Augmentation

Each prediction $\hat{\mathscr{T}}_t^s$ from ARIMA or SVM is associated with a GPS coordinate $\mathscr{G}_s = (\text{lat}_s, \text{lon}_s)$, forming the augmented feature:

$$\mathscr{X}_t^s = \left[\hat{\mathscr{T}}_t^s, \text{lat}_s, \text{lon}_s\right]^\top . \tag{3.37}$$

The full augmented dataset becomes:

$$\mathscr{D}_t = \bigcup_{s\in\mathbb{S}, t\in\mathbb{T}} (\mathscr{X}_t^s, c_t^s), \tag{3.38}$$

where $c_t^s \in \{c_g, c_e, c_w\}$ is a traffic condition label: *good*, *excellent*, or *worse*.

### Traffic Classification via XGBoost

The classifier $\Pi$ is trained on the dataset $\mathscr{D}_t$ to learn the mapping:

$$\Pi : \mathscr{X}_t^s \to c_t^s, \tag{3.39}$$

Using gradient-boosted decision trees. Each weak learner minimizes the regularized loss:

$$\mathscr{L} = \sum_{i=1}^{N} \ell(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k), \tag{3.40}$$

where $\ell$ is the cross-entropy loss and $\Omega(f_k)$ is a regularization term controlling model complexity. The final prediction is given by:

$$\hat{y}_i = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathscr{F}, \tag{3.41}$$

with $\mathscr{F}$ denoting the space of regression trees.

### Route Optimization Logic

Let $\mathscr{R} = \{r_1, r_2, \ldots, r_n\}$ denote a set of alternate routes. For each route $r_i$, let $\overline{c}(r_i)$ be the average predicted traffic class severity (numerically encoded). The optimal route $\mathscr{R}^*$ is selected via:

$$\mathscr{R}^* = \arg\min_{r_i \in \mathscr{R}} \left( \overline{c}(r_i) + \lambda \cdot \text{Length}(r_i) \right), \tag{3.42}$$

where $\lambda$ is a penalty parameter balancing congestion severity with route length.

This mathematically grounded pipeline offers a scalable and predictive system for real-time traffic state assessment and intelligent routing in mixed traffic environments.

## 3.7 Conclusion

This chapter presented a comprehensive and logically structured methodological pipeline designed to address the research objectives introduced earlier. The approach integrates high-resolution tri-axial vibration data acquisition with robust signal preprocessing stages, including spectral inspection, band-pass filtering, transient removal and energy-based segmentation. These preprocessing steps ensure that the vibration signals are clean, temporally aligned and standardized across diverse vehicle classes.

Following this, Variational Mode Decomposition (VMD) was applied to extract intrinsic mode functions capable of isolating vehicle-specific vibrational patterns. A diverse set of time- and frequency-domain features was computed from these modes, forming the basis for accurate vehicle classification using a stacked ensemble learning framework. Further, vibration equivalence metrics (RVEL and PCVE) were developed to quantify vehicular impacts and a GPS-augmented forecasting and route optimization system was proposed for infrastructure-aware traffic management.

These components form a cohesive, data-driven methodology linking signal-level dynamics with higher-level traffic modeling. The subsequent chapter elaborates on the experimental design, site conditions, instrumentation layout and data acquisition protocols that operationalize this methodological framework in real-world scenarios.

Building on the methodological constructs defined in this chapter, the next chapter details the experimental setup, sensor deployment and site-specific data collection procedures used to operationalize the proposed framework.