Low-Cost Edge AI-Based Smart Structural Condition Monitoring System for Bridges Using Self-Calibration

Prof. Pranali Nitnaware[1], Sameer Sonwane[2], Yash Khobragade[3], Trupti Kursunge[4], Aboli Bodhale[5]
[1,2,3,4,5]*Industrial IoT, Priyadarshini College of Engineering, Nagpur, India*

***Abstract*—Bridge deterioration occurs gradually and is difficult to detect with standard visual inspections. Electronic monitoring systems struggle due to high costs and unreliable inexpensive sensors. These sensors frequently drift, which can be confused with actual damage. This paper presents a cost self-calibrating structural condition monitoring system for bridges. It integrates sensing, edge intelligence, calibration, data fusion, degradation modelling, and digital-twin visualization in a single design. An ESP32 microcontroller gathers vibration, tilt, strain, temperature, and humidity data from an MPU6050, an HX711-based strain gauge, and a DHT11 sensor. Each data stream is filtered using a Kalman filter. A 5-3-5 TinyML autoencoder operates on the microcontroller. Its reconstruction error is converted into a risk probability and a six-class event label. An anomaly-gated communication protocol transmits events and periodic heartbeats, minimizing unnecessary data transfers.**

On the host side a multi-hypothesis Bayesian calibrator determines whether a strain change comes from sensor drift or real loading. It uses vibration quiescence, tilt and compensated strain to make this decision. A two-state Kalman filter separates the structural signal from persistent drift bias.

The corrected data then go into a weighted fusion engine. This engine includes per sensor explanations, an autoencoder-based Structural Health Index, a model for timber decay and remaining useful life based on Miner's rule and a Euler–Bernoulli twin. The digital twin streams to a dashboard at 20 Hz and stores data in a cloud database.

Analysis shows that the transmission schedule cuts uploaded records by than 99% compared to continuous streaming. Testing was done with a physics-based simulator and bench hardware. Scenarios included operation, heavy vehicle loads, vibrations, failure conditions and sensor drift. The prototype hardware costs, about USD 12.60. The main contribution is integrating all these components into a working system not introducing an algorithm. Field validation will be done in work.

***Index Terms*—structural health monitoring; Edge AI; TinyML; autoencoder; Bayesian self-calibration; Kalman filter; sensor drift; digital twin; ESP32**

I. INTRODUCTION

1.1. Background and Motivation

Bridges are assets and I notice that their condition changes all the time. Vehicles repeatedly load Bridges, wind and traffic cause vibrations, and daily and seasonal temperature swings alter strain and shape. Over years, these forces, together with moisture and material ageing, lower the load-carrying capacity of Bridges. Timber bridges and composite girder bridges are especially at risk, from moisture-driven decay and cumulative fatigue. Periodic visual inspection, typically done every year or more, is only qualitative; it cannot see fibre degradation and it cannot record short overload events. Sensor-based health monitoring (SHM) offers a quantitative alternative provided that the data can be trusted [9] [31] [32].

1.2. Problem Statement

Three practical challenges hinder the use of electronic SHM on ordinary bridges. First is cost: professional sensing, data-acquisition hardware and manual calibration put full systems out of reach for small authorities, educational projects and many secondary structures. The second is sensor reliability: low-cost sensors suffer offset, slow drift, noise and temperature or humidity sensitivity, and without correction these faults give misleading readings and false alarms. A strain gauge that drifts by a few tens of micro strain cannot be distinguished, from the strain channel alone, from a bridge that has genuinely settled by the same amount. The third is fragmentation: acquisition, analysis, visualization and maintenance prediction are often handled by separate tools, so no unified real-time

picture exists. A single parameter also says little. An increase in vibration might be benign, whereas simultaneous changes in vibration, strain, tilt and temperature are far more informative.

1.3. Objectives

The goal is to build a low-cost Edge AI-based self-calibrating bridge health monitoring prototype using multisensor fusion and digital-twin technology for real-time condition monitoring and anomaly detection.

The specific objectives are:

- to acquire vibration, tilt, strain, temperature, and humidity with an ESP32-based sensing node; to identify and compensate sensor offset, gradual drift, baseline variation, and noise through Bayesian self-calibration; to estimate drift with a Kalman-based decomposition (raw measurement → filtering → drift estimation → corrected measurement);
- to fuse the sensors into a single condition assessment with an explanation of each sensor's contribution;
- to detect abnormal patterns on the device with a lightweight TinyML autoencoder;
- to provide a real-time dashboard and a 3D digital twin of the monitored structure;
- to provide a prototype degradation and remaining-useful-life (RUL) estimate;
- to persist relevant records in a cloud database while minimizing data volume;
- and to integrate all of the above on low-cost hardware suitable for academic experimentation.

1.4. Contributions and Organization

This work does not claim to have invented the ESP32 platform, the sensors, TinyML, Bayesian inference, Kalman filtering, digital twins, sensor fusion, or RUL prediction. Its contribution lies in the design and prototype implementation of a single, calibration-aware pipeline that integrates these elements on very inexpensive hardware, along with a transparent specification of each stage. Section 2 reviews related work, Section 3 outlines the research gap, and Section 4 details the architecture and hardware. Sections 5 to 10 discuss the edge intelligence, self-calibration, drift decomposition, fusion and health index, degradation models, and digital twin. Section 11 addresses host software, cloud storage, and the dashboard. Section 12 explains the validation method, Section 13 presents results and discussion, and Section 14 lists the limitations. Section 15 concludes.

II. RELATED WORK

Wireless and IoT-based bridge monitoring. Long-term wireless monitoring has been demonstrated on several full-scale bridges (References [1] through [5]), and an IoT-based low-power vibration system has been evaluated in the field (Reference [6]). Long-term wireless data have also been used to study how the modal properties of a cable-stayed bridge vary statistically (Reference [7]). These studies demonstrate that wireless 2 sensing of bridges is feasible. The primary focus of these studies is on network design, energy consumption, and data collection.

Calibration and Reliability of Sensors. Yang et al. (Reference [8]) review progress in calibrating bridge SHM sensing systems, and Rizzo and Enshaeian (Reference [9]) discuss the wider challenges of bridge health monitoring. Yu et al. (Reference [11]) describe the design and calibration of wireless sensors for civil structures, and Kostic and Gül (Reference [10]) show that temperature effects must be taken into account when detecting damage from vibration data using neural networks. In this body of work, calibration is generally treated as a procedure carried out around the measurement system, not as a continuous, automatic function running alongside the monitoring itself.

Machine learning for damage and anomaly detection. Damage identification using vibration data of bridges has been achieved through neural networks and other machine-learning-based techniques [12]– [14], [17]. A number of review articles and taxonomies have provided an overview of the area [15], [16], [19]. The studies reviewed here focus on the detection algorithm, and data quality is considered as a preprocessing step. More recent research has focused on shifting intelligence towards the sensor.

For example, SHM analysis has been performed on embedded platforms for the Z24 bridge [18]. Autoencoders [34] provide a good option for this due to their ability to learn normal behaviour without requiring labelled examples of anomalies. Additionally, TinyML techniques [35] enable autoencoders to run on microcontrollers.

Digital twins, fusion and edge-cloud architectures. Reference [20] presents a systematic review of digital twin-based anomaly detection for bridge management. Several digital-twin frameworks have been developed to support: load rating [21], physics-informed neural network integration [22], suspension bridges [23], traffic load monitoring using machine vision [24], IoT integration for railway bridges [25], [26], and model-data integration [29]. Recent developments combine edge-cloud architectures, multi-sensor fusion, and anomaly detection within a digital twin framework [27]. Alternatively, researchers may use unsupervised deep learning with feature fusion [28]. Reference [30] provides a broader review of AI, IoT and digital twins for vibration based SHM. The original concept of digital twins originated in aerospace [36].

III. RESEARCH GAP

The research literature shows that every component of a modern monitoring system has been studied: wireless and IoT sensing, learning-based detection, edge deployment, sensor fusion, and digital twins.
However, what is not as apparent in the research literature reviewed by us is an automated method to monitor and correct for the reliability of sensor data using low-cost hardware within the same pipeline used for both detection and visualization.

Gap 1: drift and damage are not separated in real time. A shift in a strain channel can come from adhesive creep, thermal effects, or genuine loading. Reviews [8], [9] and work on temperature effects [10] recognise the problem, but a continuous method that weighs competing explanations using other channels (vibration level, tilt, temperature) and then corrects the baseline gradually is still needed.

Gap 2: detection models assume trustworthy input. Learning-based detectors [12]-[19] are usually evaluated on the quality of their detection. An uncorrected drift in the input can be read as damage, so the reliability of the sensor data must be established before, or together with, the anomaly decision.

Gap 3: low-cost hardware is rarely the design target. Wireless and IoT studies [1]-[7] show long-term feasibility, but self-correcting operation on very inexpensive components, where noise and drift are worse, has received less attention. Edge-oriented studies [18], [27] point in the right direction without making low-cost self-calibration their central aim.

Gap 4: calibration state is not connected to the digital twin. Digital twins [20]-[29] mirror the structure using measured data, so their output is only as good as those data. Linking calibration confidence, sensor availability, and anomaly state to the twin and to a health index makes the visualization more trustworthy for an operator. these observations describe the reviewed literature. they are not a claim that no other work exists.

IV. SYSTEM ARCHITECTURE AND HARDWARE

4.1. Three-Tier Architecture

The design of this system has three layers; each layer has its own function. At the lowest level, there is the Edge layer. The Edge layer uses the ESP32 to sample sensor readings, filter those samples, execute the TinyML autoencoder, and then determine which data to send up to the next layer. The Host layer is a Python Fast API service.

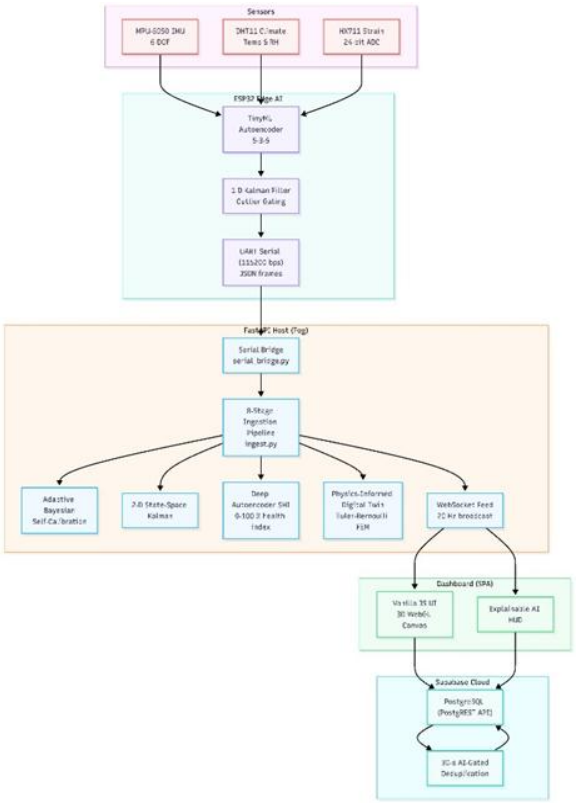


Fig. 1. Edge–host–cloud architecture of the proposed system.

The Host layer receives frames from the Edge layer via a USB Serial Bridge. Once the frames arrive at the Host layer, they enter into an eight-stage ingestion pipeline: Payload Validation, Hardware Sanitization, Bayesian Calibration, Kalman Decomposition, Health Index Estimation, Timber Decay Modelling, Remaining Useful Life (RUL) Estimation and finally Digital Twin Evaluation with Cloud Queuing.

Finally, the third layer contains both a Supabase PostgreSQL Database for storing data persistently as well as a browser dashboard which provides real-time updates regarding the entire system's current status through a WebSocket connection.

4.2. Edge Node and Sensors

The edge-node hardware includes a WROOM-32 (ESP32) module that has two 32-bit Xtensa LX6 processors running at 240 MHz, 520 KB of RAM (SRAM), and 4 MB of Flash Memory. One processor manages communications, serial data streams, and general maintenance tasks; the second processor is dedicated to providing deterministic 100-millisecond sensor readings, Kalman filter operations, and neural network inferences. The edge-node also includes a dual-watchdog timer that will reset the system after 3000 milliseconds when the I2C bus becomes unresponsive. The size of the firmware image is approximately 642 KB, and the power requirements of the board are approximately 110mA using 3.3V logic. Table 2 provides a list of the sensors employed.

Table 2. Sensors and their roles

Sensor	Specification as configured	Role in the system
MPU6050 (6-DOF IMU)	I²C address 0x68; accelerometer ±2 g (16,384 LSB/g); gyroscope ±250 °/s (131 LSB/(°/s))	Vibration RMS and peak, crest factor, angular motion rate, deck pitch and roll
DHT11 (climate)	Single-wire protocol; 0–50 °C (±2 °C); 20–90 %RH (±5 %RH)	Temperature for thermal strain compensation; humidity for moisture and decay modelling
HX711 + strain gauge	24-bit differential ADC; programmable gain 128; full-bridge Wheatstone configuration	Strain (micro strain) and load response

The MPU6050 is connected to the GPIO 19 (SDA) and GPIO 18 (SCL) via 4.7Kohm pull-ups and has its AD0 pin tied to ground. The DHT11 data line is connected to the GPIO 4 through a 10Kohm pull-up and a 100nF decoupling capacitor. The HX711 utilizes the GPIO 5 as a data input and the GPIO 23 as a clock input. An onboard LED on the GPIO 2 indicates the operating status of the device (1 Hz blink in normal operation; steady on in case of malfunction). All devices have a single common ground connection and operate at 3.3V power supply.

4.3. Hot-Swap Detection

In addition to being disconnected as a result of weather, vibration, or vandalism, Field Sensors can also be disconnected as a result of weather, vibration, or vandalism. Firmware monitors the I²C bus for 1500 milliseconds after each time that it is scanned. In the event that the MPU6050 does not acknowledge at its address, then a bit in an active-sensor bitmask will be cleared in an active-sensor bitmask and all of the MPU6050's channels will be removed from all subsequent computations. When the MPU6050 returns, the bit in the active-sensor bitmask will be set again. The same bitmask is utilized in the Autoencoder Loss (Section 5.4) and in the Host Health Index (Section 8.3).

Therefore, a missing sensor will reduce the amount of evidence that is provided to determine whether there is an issue with the system, however, it will not provide a false alarm.

V. EDGE SIGNAL PROCESSING AND TINYML ANOMALY DETECTION

5.1. Per-Channel Kalman Filtering

Each of the many channels are processed through their own one-dimensional Kalman filter [33] which has a stationary process model (F = 1, no control input). In each cycle we predict and update our estimate:

$$\hat{x}^{-}[k] = \hat{x}[k-1], P^{-}[k] = P[k-1] + Q \quad (1)$$

$$\begin{aligned} K[k] &= P^{-}[k] / (P^{-}[k] + R), \hat{x}[k] \\ &= \hat{x}^{-}[k] + K[k](z[k] \\ &- \hat{x}^{-}[k]), P[k] \\ &= (1 - K[k])P^{-}[k] \end{aligned} \quad (2)$$

Table 3 shows the settings used for tuning. By having a large R value as compared to Q value (the process noise) we can attenuate electrical jitter and continue to

track actual events. Since the wood moisture changes slowly (on the order of hours or days) we use this to damp the wood moisture.

Table 3. Edge Kalman filter tuning

Channel	Q	R	Initial P	Behaviour
Vibration RMS	0.05	2.00	1.00	Attenuates jitter, preserves shock peaks
Micro strain (HX711)	0.05	2.00	1.00	Filters ADC drift, follows axle crossings
Deck tilt pitch	0.05	2.00	1.00	Suppresses traffic vibration, isolates static tilt
Wood moisture	0.02	3.00	1.00	Heavily damped, slow response

5.2. Outlier Rejection

The three-sigma innovation gate will reject spikes in the measurements due to either electromagnetic interference or I2C bit errors. The innovation y is equal to the absolute difference between the current measurement (z[k]) and the predicted measurement ($\hat{x}^{-}$[k]). The threshold T is equal to three times the square root of the sum of the uncertainty in the prediction (P^{-}[k]) and the uncertainty in the measurement ®. If the innovation y exceeds the threshold T, then that particular measurement is discarded and the prediction is used instead. Thus, isolated outliers are rejected. However, if there are more than five consecutive outlier measurements, the filter assumes that this represents a true step change (such as a long-term sag), resets the value of P to 5.0, and initializes the estimate at the new value.

5.3. Feature Normalization

The autoencoder takes in five features which are all normalized to be within the range of [0, 2] through engineering reference values (Table 4) so that none of them dominates based upon what they measure.

Table 4. Input features of the edge autoencoder

Index	Feature	Normalization	Reference value
x_0	Vibration RMS	min (2, vibRms / 0.35)	0.35 g, warning threshold for deck vibration
x_1	Motion rate	min (2, motionRate / 45)	45 °/s, rotational excursion limit
x_2	Temperature deviation	min (2, |T – 22| / 15)	15 °C, diurnal gradient envelope
x_3	Tilt (sag) angle	min (2, |tilt| / 2.5)	2.5°, permissible beam rotation
x_4	Crest factor	min (2, peak/RMS / 5)	5.0; values below 3.0 are normal

5.4. Bottleneck Autoencoder

An autoencoder learns how to create the same output that it was given. It does so by creating an intermediate latent representation, which has fewer dimensions than the original input. This is called a "bottleneck" because the network cannot simply copy the input to the output. As such, the network will have to learn what makes up the "typical" behaviour of the system and therefore will be able to reconstruct those examples better than others. The encoder part of this autoencoder uses a Rectified Linear Unit (ReLU) activation function while the decoder has no non-linearities (it's completely linear), as illustrated in Figure 3:

$$z_j = \max\left(0, b_j^1 + \Sigma_i W_{ji}^1 x_i\right), \text{where } j = 0, 1, 2; \text{ and } i = 0 \quad (3)$$

$$\hat{x}_i = b^2{}_i + \Sigma_j W_{ij}^2 z_j \text{where } i = 0 \quad (4)$$

The three units within the latent space are associated with three physical phenomena: dynamic shock; rotational motion; and thermal or sag creep. All of the weight parameters for the autoencoder are stored as static constants in Flash memory. When performing the forward pass of the autoencoder, the code is written in standard C++ and does not use any matrix libraries or perform any dynamic memory allocation. Therefore, the execution time for this operation is deterministic.

In addition, the reported inference time for this operation is approximately 1.2 milliseconds (or 1.18 milliseconds) The error is averaged over those channels that have physical connections. The MSE of these channels (i.e., those in the "Active" set) is given as follows:

$$\text{MSE} = \frac{1}{N_{\text{active}}} \sum_{i \in \text{Active}} (x_i - \hat{x}_i)^2 \quad (5)$$

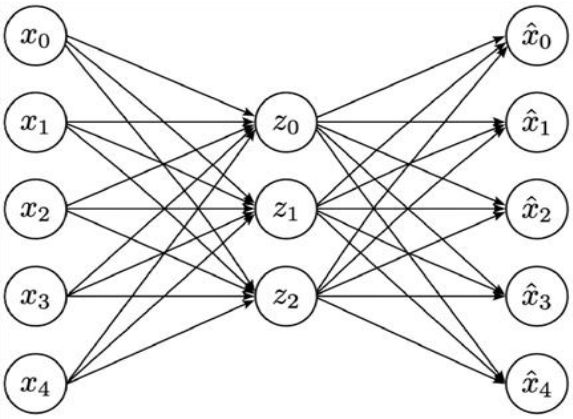


Fig. 3. Autoencoder (bottleneck) running on the ESP32.

5.5. Probabilistic Risk

We convert the error to a risk probability using a logistic function cantered at a structural threshold MSE = 0.20 and having steepness 12. This converts the error into a risk probability that is clamped to [0.01, 0.99]:

$$P(\text{Risk}) = \text{clamp}\left(\frac{1}{1+\exp\left(-12(\text{MSE}-0.20)\right)}, 0.01, 0.99\right) \quad (6)$$

Table 5. Behaviour of the risk transfer function

MSE	Logit	P(Risk)	Interpretation
0.05	−1.80	0.14	Nominal baseline
0.10	−1.20	0.23	Slight deviation
0.20	0.00	0.50	Transition boundary
0.30	+1.20	0.77	Elevated risk
0.35	+1.80	0.86	Severe distortion

Threshold for transmission risk at which the gate begins to transmit; i.e., threshold for P(Risk) ≥ 0.15 is reached when MSE ≥ 0.20 + ln (0.15/0.85)/12 ≈ 0.055. Threshold for structural warning, i.e., threshold for P(Risk) ≥ 0.20 is reached when MSE ≥ 0.084.

Therefore, the gate is intentionally sensitive, and the selectivity of the gate in practice depends on how well the autoencoder parameters are matched to the normal data of each structure. (Section 14).

5.6. On-Chip Drift Probability and Event Classification

A tilt sensor will lose calibration as it experiences cycles of heating and cooling, and also due to creep caused by long term loading.

Therefore, we can only estimate the likelihood of tilt loss during periods when there are no vehicles on the bridge. We do this based upon the difference between the current tilt baseline and the original tare tilt baseline, along with a "quiescence factor" that is calculated from the current vibration level. The quiescence factor is:

$$Q_s = 1 - \text{clamp}\left(\frac{\text{vibRMS}_{\text{run}} - 0.025}{0.055}, 0, 1\right) \quad (7)$$

And probability of tilt loss is:

$$P(\text{Drift}) = \sigma\left(0.35(\Delta\text{tilt}_{\text{baseline}} - 1.8)\right) \cdot Q_s \quad (8)$$

If we have a vehicle on the bridge causing it to vibrate, then Qs will go towards zero and thus P(Drift) will also go toward zero. This means we won't confuse a valid dynamic deflection with a drift in tilt. After determining which class each 100 ms sample belongs to, we use a priority order to determine which of the six classes each 100 ms sample is assigned to.

Table 6. Priority-ordered edge event classes

Priority	Class	Rule	Meaning
1	VIBRATION_SHOCK	vibRMS > 0.35 g or peak > 0.60 g	Impact, seismic or collision event
2	DYNAMIC_MOTION	motion rate > 35 °/s	Severe deck oscillation, wind resonance
3	TILT_EXCURSION	|Δtilt| > 2.0°	Settlement, foundation shift, bearing failure
4	SENSOR_DRIFT	P(Drift) ≥ 0.70	Electronic zero-point creep; tare needed
5	STRUCTURAL_WARNING	P(Risk) ≥ 0.20	Reconstruction error elevated; inspect
6	NOMINAL	none of the above	Normal operation

5.7. Anomaly-Gated Transmission

Frames are sent only if P(Risk) ≥ 0.15, or the event class is not NOMINAL, or 60 seconds have elapsed since the last heartbeat (see Fig. 4). All other nominal frames are suppressed at the source; thus, data reduction begins before using radios and cables.

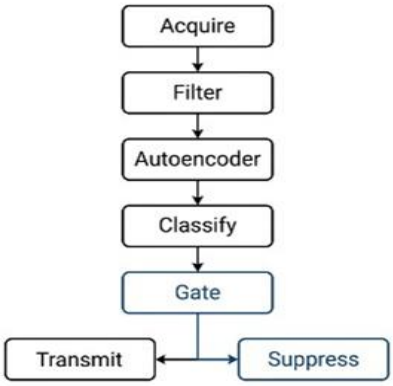


Fig. 4. Edge anomaly-gated transmission flow.

VI. ADAPTIVE BAYESIAN SELF-CALIBRATION

6.1 The Drift-Versus-Load Ambiguity

Suppose the strain gauge on a wooden bridge beam shows a steady change of +25 micro-strains. This could be due to permanent deformation of the bridge from an overload, or it could be due to drift of the sensor from either adhesive creep or moisture penetration into the sensor's housing. A simple threshold cannot distinguish between these two causes and will either trigger false alarms, or hide actual damage. The calibration process uses the fact that there are distinct differences in additional data streams for each cause. Vibration while under load and end rotation of a beam are typical indicators of real flexing of the beam, while sensor drift occurs when the bridge remains stable and level.

6.2 Hypotheses and Priors

Three hypotheses which are mutually exclusive and collectively exhaustive were developed based upon their relative frequencies of occurrence.

Table 7. Calibration hypotheses

Hypothesis	Prior	Phenomenon	Expected signature
H_drift	0.10	Sensor drift or adhesive creep	Persistent strain offset, low vibration, no tilt
H_physical	0.15	Real traffic load or settlement	Strain shift with elevated vibration and correlated tilt
H_normal	0.75	Ambient baseline variation	Small fluctuations inside the thermal envelope

6.3 Evidence and Likelihoods

The calibrator produces E = [Δε, $\bar{v}$, $\bar{\theta}$, T] over a 30 sample sliding window (minimum 10 samples for decision) where Δε represents the mean strain shift from the current offset, $\bar{v}$ represents the mean vibration RMS, $\bar{\theta}$ represents the mean tilt and T represents the temperature. The effects of thermal expansion are accounted for prior to producing E. For timber samples tested along the grain, a coefficient of 4.5 με/°C was used:

$$\varepsilon_c = \varepsilon_{\text{raw}} - 4.5(T - T_{\text{ref}}) \quad (9)$$

Independent physical factors produce the likelihood of E under each hypothesis. A large strain-shift contradicts the normal hypothesis; therefore, the strain-shift factor increases with increasing strain-shift:

$$f_s = \frac{1}{1 + \exp\left(-0.10(|\Delta\varepsilon| - 15)\right)} \quad (10)$$

The vibration and tilt factors are the discriminator. Quiet bridges have drift, bridges that vibrate and rotate support physical loading and bridges with low vibration support normal variation:

$$\begin{aligned} q_v &= \exp(-18\max(0, \overline{v} - 0.04)), \\ t_v &= 1 - \exp(-25\overline{v}), \\ n_v &= \exp(-15\overline{v}) \end{aligned} \quad (11)$$

$$\begin{aligned} q_\theta &= \exp\left(-30|\overline{\theta} - \theta_0|\right), \\ t_\theta &= 1 - \exp\left(-35|\overline{\theta} - \theta_0|\right) \end{aligned} \quad (12)$$

$$\begin{aligned} L_{\text{drift}} &= f_s q_v q_\theta, \\ L_{\text{phys}} &= f_s t_v t_\theta, \\ L_{\text{norm}} &= \exp(-0.15|\Delta\varepsilon|)\, n_v \end{aligned} \quad (13)$$

The likelihood for each hypothesis is clamped to [0.01, 0.99]. Factors for channels that are disabled or disconnected are set to 1, i.e., neutral evidence. This allows the calibrator to operate on subsets of available sensors.

These factors represent heuristically determined physical relationships and not density functions derived from observed data.

Therefore, the posterior distribution below should be viewed as a calibrated belief score and not necessarily as a true probability.

6.4 Posterior and Recalibration Rule

Bayes' Rule provides the Posterior Probability (P(H_drift|E)) of a sensor drifting.

$$P(H_{\text{drift}}|E) = \frac{L_{\text{drift}}P_{\text{drift}}}{L_{\text{drift}}P_{\text{drift}}+L_{\text{phys}}P_{\text{phys}}+L_{\text{norm}}P_{\text{norm}}} \quad (14)$$

When recalibrating, the method is intentionally cautious. It updates the Offset and Strain Calibration when P(H_drift|E) > 0.75 and the Net Drift exceeds 12 με. When this occurs, it adjusts the Offset by a small fraction of the drift using

Equation (15):

$$\text{Offset} \leftarrow \text{Offset} + 0.05\Delta\varepsilon, \quad \varepsilon_{\text{cal}} = \varepsilon_c - \text{Offset} \quad (15)$$

The 5% "Learning Rate" prevents one poor decision from significantly altering the Baseline. Also, if there is an actual drift in the system, it will be gradually eliminated through subsequent decisions. During each Loading Event, the Dominant Factor is P(H_physical). Therefore, the Offset remains Constant and the Strain Response remains Intact for both the Risk Model and the Fatigue Model. The process is summarized in Fig. 5.

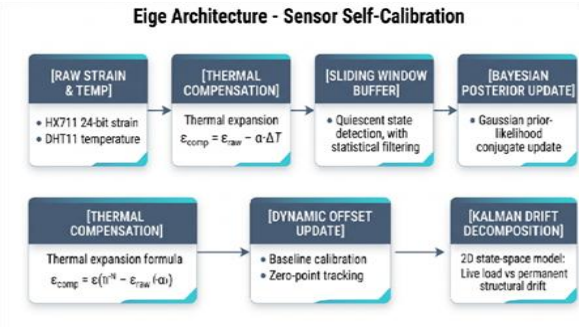


Fig. 5. Self-calibration and drift-estimation flow.

VII. STATE-SPACE KALMAN DRIFT DECOMPOSITION

While the Bayesian module makes discrete offset decisions, a two-state augmented Kalman filter provides a continuous separation of the elastic response from slowly varying bias. The state vector is x = [x_signal, x_drift] ^{T}, where x_signal is the transient elastic strain and x_drift the sensor zero-point bias.

The model is:

$$x[k] = Fx[k-1] + w, \quad z[k] = Hx[k] + v \quad (16)$$

$$F = \begin{bmatrix} 0.98 & 0 \\ 0 & 1.00 \end{bmatrix}, H = [1 \quad 1], \quad Q = \text{diag}(0.500, 0.001), R = 2.0 \quad (17)$$

The factor 0.98 models the elastic rebound of the girder after a vehicle leaves. The unit entry for the drift state models creep as a random walk that does not rebound, and the zero off-diagonal terms express that bending mechanics and semiconductor or adhesive creep are independent. The process noise for the signal is 500 times larger than for the drift, encoding that traffic strain changes within fractions of a second whereas creep evolves over hours or days. Each step performs the standard prediction and update, and returns the estimated physical signal and the drift bias.

A modeling comment is relevant here. Only the sum of the signal x_signal and drift x_drift is observable from the sensor, thus allowing separation of these states solely through their dynamic behaviour (e.g., decay rate and process noise).

Thus, the decomposition of the system into its components (signal and drift) will be no better than the accuracy of our assumptions regarding their dynamics. For this reason, the Bayesian model remains an additional check on the decomposition using independent observations.

VIII. MULTI-SENSOR FUSION AND STRUCTURAL HEALTH INDEX

8.1. Weighted fusion and interpretability

The fusion engine scales each active channel and its corresponding connection to a risk score ranging from zero to one hundred percent; and then combines the scores using adjustable weighting factors:

$$\text{Overall Risk} = \frac{\sum_i \text{Risk}_i \text{Weight}_i}{\sum_i \text{Weight}_i}, \quad \text{Health Score} = 100 - \text{Overall Risk} \quad (18)$$

To prevent "black-box" results, the contribution of each individual sensor to the potential danger (risk) is shown:

$$\text{Contribution}_i(\%) = \frac{\text{Risk}_i \text{Weight}_i}{\sum_j \text{Risk}_j \text{Weight}_j} \times 100 \quad (19)$$

If the health score drops from 95% to 68%, the dashboard will identify which channel was the primary cause for the reduction in health.

For example, if there is a vertical vibration channel that has caused 64% of the risk, it would be identified as the main driver of the reduction in health.
In addition, there would be a simple-to-read explanation of why the health score dropped. An engineer can use a run-time sensor manager to enable, disable, change settings or re-tare channels without having to recompile firmware or restart the server.

8.2 Host Autoencoder
The edge autoencoder was made to be shallow on purpose. The host system has a much larger network with eight calibrated inputs such as humidity, roll angle, and wood moisture; it uses these inputs to create a Structural Health Index (SHI). The topology of this network is 8-6-3-6-8.

Table 8. Host autoencoder topology

Layer	Size	Activation	Function
Input	8	Identity	Normalized calibrated telemetry vector
Encoder hidden	6	tanh	Non-linear multi-sensor correlations
Latent bottleneck	3	Linear	Compressed global bridge state
Decoder hidden	6	tanh	Expansion for reconstruction
Output	8	Linear	Reconstructed normal sensor values

8.3 Masked Loss and Health Mapping
The values of the loss function will be calculated based on the differences between the predicted va1lues (x̂ i) and actual (observed) values (x̃i) for each of the active channels; the unplugged sensors do not contribute to the loss calculation:

$$\text{Loss} = \frac{1}{|\text{Active}|}\sum_{i\in\text{Active}}(\tilde{x_i} - \hat{x_i})^2 \quad (20)$$

The value of the loss function is then transformed into a 0–100% index using a sigmoid function and limited to [5, 99.8]:

$$\ell = \max\left(0, \frac{\text{Loss}-0.015}{0.05}\right), \quad \text{SHI} = \frac{100}{1+\exp(2.5(\ell-1.2))} \quad (21)$$

Table 9. SHI grading tiers

SHI	Status	Suggested response
85–100 %	HEALTHY	Nominal operation; routine inspection schedule
70–84.9 %	MONITORING	Increase logging frequency; check drainage and wear
50–69.9 %	WARNING_INSPECTION	Physical inspection within 30 days; restrict heavy permit loads
< 50 %	CRITICAL_HAZARD	Immediate action: close or restrict the bridge and deploy a repair crew

IX. DEGRADATION MODELS FOR TIMBER MEMBERS

9.1 Moisture Content and Fungal Decay
Fungal decay of wood is one of the major causes of deterioration in timber bridges. For fungi to grow and cause decay, they require both free moisture within the cells (i.e., moisture content greater than approximately 20%) and an adequate supply of oxygen.

Additionally, fungi generally thrive under moderate temperature conditions. The decay model utilizes data from field measurements of moisture and temperature to calculate a "Scheffer Climate Index" [38] that represents the environmental conditions for fungal growth. As shown in Table 10, three different moisture zones exist around the fiber saturation point (FSP) of 28%. (Table 10)

Table 10. Moisture regimes and decay behaviour

Regime	Moisture content	Biological activity	Decay rate
NEGLIGIBLE_DRY	< 20 %	Dormant (0 %)	0 mm/year
ELEVATED_INCUB ATING	20–28 %	Scales from 0 to 65 %	0.10–0.40 mm/year
CRITICAL_ACTIVE_ROT	> 28 %	70–100 %	1.20–1.80 mm/year

The maximum growth occurs at an optimum temperature of 27°C and decreases linearly as the temperature deviates from this point:

$$f_T = \max\left(0.10, 1 - \frac{|T - 27|}{18}\right), \quad f_M = \frac{\mathrm{MC} - 20}{8} \tag{22}$$

$$\text{Activity}(\%) = 65 f_T f_M, \quad \text{Decay rate}(\mathrm{mm/year}) = 0.40 f_M f_T \tag{23}$$

When there is no wood probe available, the equilibrium moisture content is determined based on the temperature and relative humidity using the Hailwood-Horrobin sorption isotherm [39]:

$$W = 330 + 0.452T + 0.00415T^2, \quad K = 0.791 + 0.000463T - 0.00000844T^2 \tag{24}$$

$$K_1 = 6.34 + 0.0775T - 0.0000935T^2, \quad H = \frac{\mathrm{RH}}{100} \tag{25}$$

$$\mathrm{EMC} = \frac{1800}{W}[A + B], \quad A = \frac{KH}{1 - KH} \tag{26}$$

$$B = \frac{K_1 KH + 2K_1{}^2 K^2 H^2}{1 + K_1 KH + K_1{}^2 K^2 H^2} \tag{27}$$

9.2 Cumulative Fatigue and Remaining Useful Life

The cumulative fatigue is calculated using the Palmgren-Miner rule [37], which states that each load cycle uses up a portion of the life of the material and that failure will occur once the total usage equals or exceeds one:

$$D = \sum_i \frac{n_i}{N_i}, \quad \text{failure when } D \geq 1 \tag{28}$$

The number of cycles to failure N_i under a given bending stress sigma can be determined by an S-N relationship for Douglas Fir:

$$\log_{10} N = A - B\left(\frac{\sigma}{\mathrm{MOR}_{\mathrm{eff}}}\right), \quad A = 10.2, B = 8.6 \tag{29}$$

The effective modulus of rupture decreases with an increase in moisture content; at moisture levels greater than 20%, rot adds to the fatigue damage. The formulas used for calculating this include:

$$\mathrm{MOR}_{\mathrm{eff}} = 55[1 - 0.022 \min(\mathrm{MC} - 12, 16)], \quad (\mathrm{MC} > 12\%) \tag{30}$$

$$\mathrm{Rot}_{\mathrm{penalty}} = 1 + 0.08(\mathrm{MC} - 20), \quad (\mathrm{MC} > 20\%) \tag{31}$$

As a result of 120,000 vehicle passes per year being assumed, the yearly rate of fatigue is equal to (120,000/N) times the rot penalty.

In addition to the annual fatigue, total wear includes a base aging component which increases as the health index decreases.

Additionally, RUL represents the amount of available health space above a critical SHI value of 40%, divided by total wear:

$$\mathrm{Wear}(\%/\mathrm{yr}) = 100 \cdot \mathrm{Fatigue}_{\mathrm{rate}} + 1.2\left(\frac{100}{\mathrm{SHI}}\right) \tag{32}$$

$$\mathrm{RUL}(\mathrm{years}) = \frac{\max(1, \mathrm{SHI} - 40)}{\max(0.40, \mathrm{Wear})} \tag{33}$$

Table 11. RUL-based maintenance tiers

RUL	Tier	Recommended action
> 20 years	ROUTINE_SCHEDULE_5_YEARS	Maintain the standard five-year inspection cycle
8–20 years	PRIORITY_INSPECTION_12_MONTHS	Schedule non-destructive timber sounding within 12 months
2–7.9 years	URGENT_TIMBER_RESTORATION	Plan stringer sistering and deck replacement within six months
< 2 years	IMMEDIATE_CLOSURE_OR_DERATING	Close the bridge or apply a 50 % weight restriction

The constants used were based upon existing engineering practices related to designing using wood products. Prior to using these constants as an input for determining the remaining useful life (RUL), they must be calibrated against experimental data. The RUL value is a first attempt at creating a tool that can help predict the life of a bridge, however, it has not been officially validated by a third party.

X. PHYSICS-BASED DIGITAL TWIN

The digital twin utilizes the standard Euler-Bernoulli beam equation [40]: d^2/dx^2 [E I(x) d^2y/dx^2] = q(x), where E is Young's Modulus, I is the second moment

of area, and q represents the distributed line load (dead weight and live load).

For a simply supported span of length L, the span was discretized into 11 nodes. Both dead and live response were superposed: a uniform line load w represented the dead load and a concentrated point load P located at mid-span represented the live load.

$$y_{\text{dead}}(x) = \frac{wx}{24EI}(L^3 - 2Lx^2 + x^3) \quad (34)$$

$$y_{\text{live}}(x) = \frac{Px}{48EI}(3L^2 - 4x^2), \quad x \le \frac{L}{2} \quad (35)$$

$$M(x) = \frac{wx(L-x)}{2} + \frac{P\min(x, L-x)}{2} \quad (36)$$

$$\sigma(x) = \frac{M(x)y_{\text{fiber}}}{I}, \quad \text{Stress ratio} = \frac{\sigma(x)}{\sigma_{\text{allow}}} \quad (37)$$

Structural properties are representative of the selected bridge type (Table 12).

Each beam segment is colored from green through amber to red based on the stress ratio and cables are displayed for the cable-stayed type and triangulated web for the truss.

Table 12. Bridge archetype profiles

Archetype	Span	E	I (m⁴)	Allowable stress	Dead load w
Historic covered timber	36.5 m	11 GPa	0.1728	18 MPa	12,000 N/m
Concrete cable-stayed	120 m	35 GPa	1.4500	28 MPa	45,000 N/m
Warren steel truss	48 m	200 GPa	0.8500	165 MPa	22,000 N/m
Prestressed concrete girder	32 m	32 GPa	0.5500	22 MPa	28,000 N/m

The digital twin will be used to monitor and visualize; however, the digital twin models the beam as a simple supported member and can't replace the results of a professional finite-element analysis.

Visualization of the deflection and stress of the digital twin.

XI. HOST SOFTWARE, CLOUD STORAGE, AND DASHBOARD

11.1 Serial Bridge and Ingestion Pipeline

A serial bridge receives JSON frames via USB (COM4 at 115200 bps) from the ESP32. The serial bridge utilizes non-blocking polling, has buffer overflow protection and will automatically reconnect. The serial bridge checks the backend every two seconds for the current sensor settings. Prior to sending the data to the backend, the serial bridge sets all disabled channels to 0.

Additionally, the serial bridge verifies that all reported numeric data are valid before reporting that a channel is active. Once the serial bridge validates that a channel is active, the serial bridge sends the data through the eight process steps listed in Table 13.

Table 13. Host ingestion pipeline

Stage	Operation	Transformation
1	Payload validation	Schema check; bind edge metrics P(Risk), P(Drift), latent z_0–
2	Hardware sanitization	z_2 Update connection map; isolate disconnected channels
3	Bayesian calibration	30-sample window; thermal compensation; drift posterior
4	Kalman decomposition	Split strain into elastic signal and drift bias
5	Structural Health Index	Masked 8-6-3-6-8 autoencoder, 0–100 %
6	Timber decay	Biological activity, decay rate, equilibrium moisture
7	Fatigue RUL	Miner's damage and S–N life, RUL in years
8	Digital twin and cloud queue	11-node beam solution; cloud upload with rate limiting

The backend is written in Python 3.12, utilizing FastAPI and Uvicorn. It employs Pydantic v2 to ensure strict validation of incoming payloads and NumPy to perform the required matrix calculations.

Additionally, it maintains a local SQLite database using SQLAlchemy. The firmware is developed utilizing PlatformIO on the Espressif32 platform and

includes an ArduinoJson library for serialization purposes as well as a MPU6050 I²C driver.

11.2 Cloud Persistence and Data Reduction

Asynchronous transmissions of validated events and heartbeats to a Supabase PostgreSQL table via PostgREST are secured using TLS. Each record contains the bridge identifier, timestamp, probability of risk and drift, health index, event type, vibration, motion, temperature, tilt, latent variables, raw and cloud payload size, and indicators for whether or not the event was an anomaly or heartbeat.

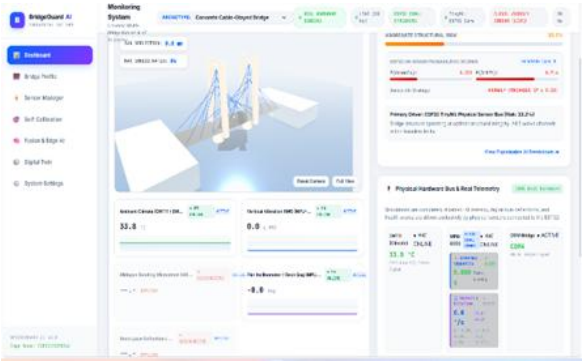
Filters restrict the amount of data transferred to the cloud. These include: (i) the edge gate discussed in section 5.7; (ii) a host composite filter requiring at least one of nine conditions (e.g., high vibration, tilt, SHI below 75%, rotational hazard, alert); and (iii) a 30 second deduplication window within the cloud module to regulate upload rates while the

Event class is not modified, however, sends immediately when the class changes (e.g., from nominal to vibration shock). Therefore, there is no delay in changing states.

11.3 Dashboard

The dashboard is a single page application in HTML, CSS, and Javascript which receives the full system state through a WebSocket at 20 Hz.

There are seven tabs on the dashboard: a live dashboard with key indicators, 3D bridge's view, Edge-AI bars and cloud status; a tab to select a bridge profile; a tab to manage sensors; a self-calibration tab showing raw vs. calibrated strain and drift posterior; a fusion & edge-ai tab showing contributions bars and latent values; a full screen digital twin; and a settings tab for Serial Port, Baud Rate, Broadcast Rate, and CSV export.



XII. VALIDATION METHODOLOGY

Overload, cracking, or seismic excitation are dangerous conditions that cannot be safely created on a real bridge. Therefore, the prototype was tested with a hardware bench using COM4 connecting the ESP32 and sensors to the host to check acquisition, on-device inference, transmission, and the dashboard.

In addition to this, a physics-based simulator was utilized to provide the same input pipeline to the host, but in a controlled manner. The simulator creates a diurnal sinusoidal temperature cycle, vehicle crossings, and moisture changes due to rainfall. The vehicle crossing follows a parabolic influence line with progress p in [0, 1]. ε_dyn is the dynamic strain in με:

$$\text{Influence}(p) = 4p(1-p),$$

$$\varepsilon_{\text{dyn}} = 140\left(\frac{P_{\text{veh}}}{P_{\text{design}}}\right)\text{Influence}(p^{(38)}$$

Therefore, at 85% of the design load, the peak dynamic strain is approximately 119 με, consistent with the 120 με applied to the heavy vehicle scenario. Five scenarios are defined (Table 14), and a sixth test introduces an artificial drift into the strain channel.

Table 14. Validation scenarios

Scenario	Stimulus	Expected response
normal	Solar temperature cycle; passenger cars of 1–3 t	NOMINAL; high health index; no alerts
heavy_vehicle	Semi-trailer at 85 % of design load; peak about 120 με; vibration about 0.18 g; roll 0.8°	DYNAMIC _MOTION or VIBRATION_ SHOCK
vibration_peak	Gust or seismic tremor; RMS 0.42 g; peak 0.75 g; crest factor 4.8; 4.2 Hz	VIBRATION_ SHOCK with immediate transmission
failure_anomaly	135 % overload with cracked stringer; strain above 400 με; permanent tilt 3.2°	STRUCTURAL_ WARNING or critical; SHI below 40 %
drift injection	Offset added to strain on a quiet, level bridge	Drift posterior rises; offset corrected gradually

XIII. RESULTS AND DISCUSSION

13.1 Edge Decision Behavior

The edge model operates according to the transfer function given in Table 5. Probability increases gradually from 0.14 at MSE=0.05 to 0.86 at MSE=0.35. A 0.15 transmission gate occurs when MSE is approximately 0.055. Measurement of the inference time (approximately 1.2 milliseconds) indicates that there will be sufficient time available within each 100-millisecond sampling interval to perform additional operations such as filtering, I²C probing, and serial output.

13.2 Data Reduction

If all 10 Hz samples were transmitted, then approximately 864,000 records would be generated daily. Assuming that each record has a file size of approximately 500 bytes, this equates to approximately 432 megabytes per bridge per day.

However, if anomaly detection is employed to gate transmissions, and no anomalies occur, then only a 60second heartbeat is sent by the quiet bridge. This results in 1,440 records per day and approximately 0.72 megabytes of data, representing a reduction of 99.83%. If burst transmission is limited due to rate limiting during a busy day, then approximately 2,880 records may be generated, requiring approximately 1.44 megabytes of data. This represents a reduction of 99.67%. Emergency conditions will require immediate uploads upon detection of any state change (see Table 15). The payload figure for each frame reported by the firmware is 2,840 bytes compared to 458,920 raw bytes, indicating a 99.4% reduction in data volume. This is distinct from the record count reduction and therefore cannot be compared directly. The record figures are derived from the upload schedule and were not measured during an extended period of field testing.

Table 15. Data volume under different operating modes

Mode	Records per day	Volume per day	Reduction
Continuous 10 Hz stream	864,000	about 432 MB	baseline
Quiet bridge (60 s heartbeat)	1,440	about 0.72 MB	99.83 %
Busy day (rate-limited bursts)	2,880	about 1.44 MB	99.67 %
Emergency event	upload on each state change	not limited	full event capture

13.3 Scenario Behaviour

The performance of the pipeline was consistent with what it was designed for during functional verification, across all scenarios of Table 14. Under normal operating conditions, the system operated normally with a high health index and no alarms. The heavy-vehicle and vibration tests resulted in classification of these tests into one of two categories: dynamic motion or shock event, and they were immediately sent. The failure scenario resulted in a decrease in the health index, and the production of the highest-severity state. An offset due to a sensor on a quiet, level bridge was identified in the drift test, and the offset was corrected in a stepwise manner. In contrast, loading events with vibration and tilt increased the risk indicators, and could not be corrected by calibrating them away. Removing a sensor from the system caused the channel of the removed sensor to be excluded from further calculation without generating an alarm. This shows that the integrated pipeline is functioning as intended. The above is not a statistical measurement of the performance of the pipeline. Statistical measurements of the performance of the pipeline can be found in Table 16 along with actual measured data.

Table 16. Quantitative metrics to be reported from measured runs

Metric	Definition	Measured value
Drift-correction error	Residual strain error after calibration (με)	[insert]
Time to drift detection	Time from drift onset to posterior > 0.75	[insert]
False-alarm rate	Alerts per day in the normal scenario	[insert]
Detection rate	Fraction of injected events correctly classified	[insert]
Loading preserved	Fraction of load events not “calibrated away”	[insert]

13.4 Cost

In Table 17 we have listed all of the hardware used in our prototype system. We estimate that the total cost will be $12.60 USD (Table 17) for a deployment of a single unit. This cost estimate supports the concept of low-cost SHM experimentation. The items listed in Table 17 include the node, sensor module(s), cables and breadboards. However, the strain gauge element, its enclosure and a field power source are not included as part of this estimate. They would be added to the total cost if they were to be included in a complete unit.

13.5 Discussion

Table 17. Prototype bill of materials

Component	Model	Qty	Unit cost (USD)
ESP32 microcontroller node	ESP32-WROOM-32D	1	4.50
6-DOF IMU	MPU-6050	1	1.80
Climate sensor	DHT11	1	1.20
Strain-gauge ADC	HX711	1	1.10
USB cable	Micro-USB, 1 m	1	1.50
Breadboard and jumper wires	400-point, 22 AWG	1	2.50
Total			12.60

As shown above, the main finding was architectural. When calibration is treated as a continuous process through the entire pipeline rather than as a maintenance task, then the same system can determine whether or not to rely upon a particular reading; and the reasoning behind this determination will be based upon physical evidence that can be examined by any engineer.

Additionally, when combined with edge screening, it is possible for all aspects of the sensor-level decision (i.e., the data-volume decision and the visualization) to be driven by the corrected data. The fact that the fusion output is explainable and the relationship between sensor status and the health index addresses Gap 4.

Therefore, an operator will not only know that a bridge appears to be unhealthy but also which channel caused it to appear so and if that channel is currently being relied upon.

XIV. CONCLUSION AND FUTURE WORK

In this paper, we introduced a low-cost, self-calibrating bridge condition monitoring prototype which includes: multi-sensor acquisition; an on-device TinyML autoencoder; anomaly-gated transmission; Bayesian drift-vs. load discrimination; Kalman signal and drift decomposition; explainable fusion; a Structural Health Index (SHI); timber decay and fatigue models; a physics-based digital twin; and selective cloud persistence.

One key aspect of the prototype is that we assume that the reliability of a given sensor should be determined based upon the performance of its neighbours.

Thus, we are able to correct for drift while avoiding suppression of real structural responses. As well, our design significantly reduces uploaded records (>99%) in the operating modes analyzed herein and operates on hardware costing ~USD 12.60.

Future work includes:

1) testing the prototype on a scaled physical bridge model under controlled loading conditions and later on a real structure using reference grade sensors;
2) training autoencoders on labeled normal-condition data and tuning thresholds per structure;
3) adding temperature compensated strain gages, a dedicated inclinometer, and a more accurate climate sensor;
4) quantifying calibration error, detection rate, and false alarm rate using metrics listed in Table 16;
5) validating twin and fatigue models against finite element software and material tests;
6) studying long term stability, unmodeled sensor faults, and solar-powered wireless deployment; and
7) hardening security for networked operation.

APPENDIX A. Telemetry Payload Fields

Each frame posted to the ingestion endpoint is a validated JSON object. Table 18 summarizes its main fields.

Table 18. Main fields of the telemetry payload

Field group	Content
Identification	bridge_id, timestamp
Measurements	vibration_rms, peak_frequency_hz, raw_strain, calibrated_strain, wood_moisture_pct, temperature_c, humidity_pct, tilt_pitch_deg, tilt_roll_deg, traffic_load_tons
Edge AI result	probabilistic_risk, p_anomaly, p_drift, drift_flag, anomaly_classification, latent_z0–z2, raw_bytes, cloud_bytes, compression_ratio_pct
Hardware status	status and connected flag for MPU6050, DHT11, and HX711
Flags	is_anomaly, is_heartbeat

REFERENCES

[1] N. A. Hoult, P. Fidler, P. G. Hill, C. Middleton, et al., "Long-Term Wireless Structural Health Monitoring of the Ferriby Road Bridge," *Journal of Bridge Engineering*, vol. 15, no. 2, pp. 153–159, 2010.

[2] J. Meyer, R. Bischoff, G. Feltrin, and M. Motavalli, "Wireless Sensor Networks for Long-Term Structural Health Monitoring," *Smart Structures and Systems*, vol. 6, no. 3, pp. 263–275, 2010.

[3] Y. Lei, W. Shen, Y. Song, and Y. Wang, "Intelligent Wireless Sensors with Application to the Identification of Structural Modal Parameters and Steel Cable Forces: From the Lab to the Field," *Advances in Civil Engineering*, vol. 2010, Art. no. 316023, 2010.

[4] H. R. Lin, C. S. Chen, P. Y. Chen, et al., "Design of Wireless Sensor Network and Its Application for Structural Health Monitoring of Cable-Stayed Bridge," *Smart Structures and Systems*, vol. 6, no. 8, pp. 939–951, 2010.

[5] A. K. M. A. Islam, F. Li, and P. K. Kolli, "Monitoring Structural Health of a Bridge Using Wireless Sensor Network," in *Proc. Structures Congress 2010*, ASCE, 2010, pp. 3634–3644.

[6] X. Tong, H. Yang, L. Wang, and Y. Miao, "The Development and Field Evaluation of an IoT System of Low-Power Vibration for Bridge Health Monitoring," *Sensors*, vol. 19, no. 5, Art. no. 1222, 2019.

[7] P. Asadollahi and J. Li, "Statistical Analysis of Modal Properties of a Cable-Stayed Bridge through Long-Term Wireless Structural Health Monitoring," *Journal of Bridge Engineering*, vol. 22, no. 9, 2017.

[8] Y. Yang, T. Chen, W. Lin, M. Jing, et al., "Research Progress on Calibration of Bridge Structural Health Monitoring Sensing System," *Advances in Bridge Engineering*, vol. 5, Art. no. 32, 2024.

[9] P. Rizzo and A. Enshaeian, "Challenges in Bridge Health Monitoring: A Review," *Sensors*, vol. 21, no. 13, Art. no. 4336, 2021.

[10] B. Kostić and M. Gül, "Vibration-Based Damage Detection of Bridges under Varying Temperature Effects Using Time-Series Analysis and Artificial Neural Networks," *Journal of Bridge Engineering*, vol. 22, no. 10, 2017.

[11] Y. Yu, J. Ou, and H. Li, "Design, Calibration and Application of Wireless Sensors for Structural Global and Local Monitoring of Civil Infrastructures," *Smart Structures and Systems*, vol. 6, nos. 5–6, pp. 641–659, 2010.

[12] J. C. Weinstein, M. Sanayei, and B. R. Brenner, "Bridge Damage Identification Using Artificial Neural Networks," Journal of Bridge Engineering, vol. 23, no. 11, 2018.

[13] "Vibration-Based Damage Detection in Bridges via Machine Learning," KSCE Journal of Civil Engineering, vol. 22, pp. 5123–5132, 2018. [Authors to be added].

[14] N. Jayasundara, D. P. Thambiratnam, T. H. T. Chan, and A. Nguyen, "Damage Detection and Quantification in Deck Type Arch Bridges Using Vibration Based Methods and Artificial Neural Networks," Engineering Failure Analysis, vol. 109, Art. no. 104265, 2020.

[15] "Artificial Intelligence and Structural Health Monitoring of Bridges: A Review of the State-of-the-Art," IEEE Access, vol. 10, pp. 88058–88078, 2022. [Authors to be added].

[16] "Algorithms and Techniques for the Structural Health Monitoring of Bridges: Systematic

Literature Review," Sensors, vol. 23, no. 9, Art. no. 4230, 2023. [Authors to be added].

[17] "Damage Detection in Railway Bridges Using Traffic-Induced Dynamic Responses," Engineering Structures, vol. 238, Art. no. 112189, 2021. [Authors to be added].

[18] "Bringing Intelligence to the Edge for Structural Health Monitoring: The Case Study of the Z24 Bridge," IEEE Open Journal of the Industrial Electronics Society, vol. 5, pp. 781–794, 2024. [Authors to be added].

[19] "Bridge Structural Health Monitoring: A Multi-Dimensional Taxonomy and Evaluation of Anomaly Detection Methods," Buildings, vol. 15, no. 19, Art. no. 3603, 2025. [Authors to be added].

[20] A. Jiménez Rios, V. Plevris, and M. Nogal, "Bridge Management through Digital Twin-Based Anomaly Detection Systems: A Systematic Review," Frontiers in Built Environment, vol. 9, Art. no. 1176621, 2023.

[21] A. Li, M. Bayat, G. Comert, and P. Ziehl, "An Autonomous Bridge Load Rating Framework Using Digital Twin," in Proc. Structural Health Monitoring 2021, DEStech/DPI Proceedings, 2021.

[22] S. Moradi, S. Eftekhari Azam, and M. Mofid, "A Physics Informed Neural Network Integrated Digital Twin for Monitoring of the Bridges," in Proc. Structural Health Monitoring 2021, DEStech/DPI Proceedings, 2021.

[23] S. Gunner, E. Voyagaki, G. Gavriel, et al., "Digital Twins for Civil Engineering: The Clifton Suspension Bridge (UK)," in Proc. SHMII 2021, 2021, pp. 1561–1566.

[24] D. Dan, Y. Ying, and L. Ge, "Digital Twin System of Bridges Group Based on Machine Vision Fusion Monitoring of Bridge Traffic Load," IEEE Transactions on Intelligent Transportation Systems, vol. 23, no. 11, pp. 22190–22205, 2021.

[25] "Integration of Railway Bridge Structural Health Monitoring into the Internet of Things with a Digital Twin: A Case Study," Sensors, vol. 24, no. 7, Art. no. 2115, 2024. [Authors to be added].

[26] "A Demonstration of a Digital Twin Framework for Structural Health Monitoring: Application to Bridge Infrastructures," Journal of Infrastructure Intelligence and Resilience, vol. 5, no. 1, Art. no. 100184, 2026. [Authors to be added].

[27] Y. Liu, "Edge-Cloud Collaborative Digital Twin Architecture for Bridge Structural Health Monitoring: Multi-Sensor Fusion and Anomaly Detection Algorithm Design," Discover Applied Sciences, vol. 8, Art. no. 938, 2026, doi: 10.1007/s42452-026-08991-6.

[28] V. Mousavi, M. Rashidi, S. Ghazimoghadam, and B. Samali, "A Data-Driven Digital Twin Model for Bridge Health Monitoring Using Feature Fusion and Unsupervised Deep Learning," Information Fusion, vol. 126, Art. no. 103534, 2026.

[29] "Hybrid Monitoring Methodology: A Model-Data Integrated Digital Twin Framework for Structural Health Monitoring and Full-Field Virtual Sensing," Advanced Engineering Informatics, 2024. [Authors to be added].

[30] R. S. Chaudhari, S. Nama, M. M. Narkhede, V. P. Bhangale, R. B. Gaikwad, et al., "Next Generation Vibration-Based Structural Health Monitoring for Bridges: A Systematic Review of AI, IoT, and Digital Twin Integrations," Journal of Building Pathology and Rehabilitation, vol. 11, Art. no. 136, 2026.

[31] C. R. Farrar and K. Worden, Structural Health Monitoring: A Machine Learning Perspective. Chichester, U.K.: Wiley, 2012.

[32] J. M. W. Brownjohn, "Structural Health Monitoring of Civil Infrastructure," Philosophical Transactions of the Royal Society A, vol. 365, no. 1851, pp. 589–622, 2007.

[33] R. E. Kalman, "A New Approach to Linear Filtering and Prediction Problems," Journal of Basic Engineering, vol. 82, no. 1, pp. 35–45, 1960.

[34] G. E. Hinton and R. R. Salakhutdinov, "Reducing the Dimensionality of Data with Neural Networks," Science, vol. 313, no. 5786, pp. 504–507, 2006.

[35] P. Warden and D. Situnayake, TinyML: Machine Learning with TensorFlow Lite on Arduino and Ultra-Low-Power Microcontrollers. Sebastopol, CA, USA: O'Reilly Media, 2019.

[36] E. Glaessgen and D. Stargel, "The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles," in Proc. 53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference, Honolulu, HI, USA, 2012.

[37] M. A. Miner, “Cumulative Damage in Fatigue,” Journal of Applied Mechanics, vol. 12, no. 3, pp. A159–A164, 1945.
[38] T. C. Scheffer, “A Climate Index for Estimating Potential for Decay in Wood Structures Above Ground,” Forest Products Journal, vol. 21, no. 10, pp. 25–31, 1971.
[39] A. J. Hailwood and S. Horrobin, “Absorption of Water by Polymers: Analysis in Terms of a Simple Model,” Transactions of the Faraday Society, vol. 42B, pp. 84–92, 1946.
[40] S. Timoshenko, Strength of Materials, Part I: *Elementary Theory and Problems*, 3rd ed. New York, NY, USA: D. Van Nostrand, 1955.