Journal of Engineering and Multidisciplinary Applied Sciences
Vol. 01, Issue 01, July–December 2026, pp. 23-32

Sixty Parameters Are Enough: Model Capacity, Quantisation and the Real Constraints on an On-Device Vibration Condition Monitor

Altaf Mulani

Professor and Head of Department, Electronics & Telecommunication Engineering, SKN Sinhgad College of Engineering, Pandharpur, Solapur, Maharashtra, India.

Email: draomulani.vlsi@gmail.com

ABSTRACT

Wireless condition monitoring nodes are increasingly expected to classify machine faults on the device rather than stream raw vibration to a server. The design question is usually framed as which model to deploy and how aggressively to compress it, on the assumption that model capacity and numerical precision are the constraints. Whether they are has rarely been tested against the alternatives on a common footing. This study measures how classification accuracy for bearing fault states depends on model capacity, on weight precision and on signal-to-noise ratio, and places those results alongside the device-level energy cost of streaming, feature reporting and on-device classification. Synthetic single-axis vibration signals were generated at 12 kHz in 2,048-sample windows for four states — healthy, inner race, outer race and ball fault — with characteristic defect frequencies, a modulated resonance and additive noise. Fourteen time, spectral and envelope features were extracted, and four classifiers were trained on 1,680 windows and tested on 720. Weights of the largest model were uniformly quantised from 32 to 2 bits. An energy model for a 2,400 mAh node reporting every ten minutes compared four reporting strategies. Logistic regression with 60 parameters achieved 97.36 % accuracy, matching an eight-unit network at 97.36 % and exceeding a two-layer network with 1,076 parameters at 96.94 % and a fifty-tree random forest with 28,452 parameters at 94.58 %. Quantisation to 8 bits was lossless and to 4 bits cost 1.11 percentage points, while 2 bits collapsed accuracy to 75.97 %. Accuracy fell from 99.88 % at 8 dB signal-to-noise ratio to 72.50 % at −4 dB. At device level, streaming raw windows gave 1.54 years of battery life against 7.11 years for on-device classification, and the two classifiers were indistinguishable in energy because inference occupied 2.6 µs and 48.0 µs against 2.11 ms for the fast Fourier transform that preceded both. Neither model capacity nor weight precision constrains this task: a 60-byte classifier is sufficient and 8-bit weights are free. The constraints are the signal-to-noise ratio, which no model choice repairs, and the radio, which dominates the energy budget at 76–95 % under every strategy. Effort directed at model architecture is effort spent on the one part of the system that is not limiting.

Keywords: condition monitoring; edge machine learning; quantisation; vibration analysis; industrial IoT; embedded classification.

1. INTRODUCTION

A wireless vibration node bolted to a pump or gearbox has a simple job and a hard constraint. The job is to notice that a bearing is degrading before it fails; the constraint is that it must do so for years on a primary cell, in a location where changing the battery costs more than the node.

The traditional architecture streams data to a server, which is straightforward and expensive in radio energy [13,15]. The current alternative is to classify on the device and report a state rather than a signal, which has become practical as microcontrollers have acquired the compute to run inference. That shift has brought with it a body of work on deploying neural networks to microcontrollers, on pruning, quantisation and architecture search, framed around the question of how much model can be fitted into how little memory [5-8,12,14].

The framing deserves scrutiny. Bearing fault detection is a task with a long history of physically motivated feature engineering [1,4]: the characteristic defect frequencies are calculable from the bearing geometry and shaft speed, and envelope analysis has extracted them reliably for decades [2,3]. If those features are computed on the device anyway — and something must transform the raw window before any classifier sees it — then the classifier operating on them may be a very small object, and the effort spent compressing a larger one may be misdirected.

This paper tests that proposition end to end. Three questions are addressed: how accuracy varies with model capacity on a physically motivated feature set; how far the weights can be quantised before accuracy suffers; and how those choices compare, in device energy, with the decision about what to transmit.

2. METHOD

2.1 Signal Generation

Vibration signals were synthesised at 12 kHz in windows of 2,048 samples, corresponding to 171 ms of data. Each signal contains shaft-rate and second-harmonic components, and for the three fault classes an impulse train at the relevant characteristic defect frequency convolved with a decaying 3.2 kHz structural resonance, with a small random jitter on impulse timing to represent slip [1,3]. Inner-race faults additionally carry shaft-rate amplitude modulation, which is the physical signature that distinguishes them from outer-race faults [1,2]. Gaussian noise was added at a signal-to-noise ratio drawn about a nominal value. The signal parameters are listed in Table 1.

Table 1: Signal parameters.

Parameter	Value
Sampling rate	12 kHz
Window length	2,048 samples (171 ms)
Shaft rate	30 Hz
Inner race defect frequency	5.41 × shaft rate = 162.3 Hz
Outer race defect frequency	3.58 × shaft rate = 107.4 Hz
Ball defect frequency	2.36 × shaft rate = 70.8 Hz
Structural resonance	3.2 kHz, exponentially damped
Nominal signal-to-noise ratio	2 dB, standard deviation 3 dB; swept −4 to 8 dB
Classes	Healthy, inner race, outer race, ball
Windows	2,400 total; 1,680 training, 720 test

Note: Signals are synthetic, generated from a physically motivated model rather than recorded from a test rig. This permits a controlled sweep of signal-to-noise ratio and guarantees balanced classes, and it is the study's principal limitation; see Section 4.3.

2.2 Features and Classifiers

Fourteen features were extracted from each window: root mean square, peak, crest factor and kurtosis in the time domain; five spectral band energies; and envelope spectrum peaks at the three characteristic defect frequencies and at the first two harmonics of the inner and outer race frequencies. The envelope features are the physically informed ones and carry most of the discriminative information [1,4].

Table 2 lists the four classifiers compared, which span the model families most commonly applied to fault diagnosis and available in standard machine learning libraries [5,17].

2.3 Quantisation and Device Model

Weights and biases of the largest network were uniformly quantised to a symmetric grid at precisions from 16 down to 2 bits, with the scale set by the maximum absolute value in each layer, and inference was recomputed in the quantised domain [9-11].

The device model represents a node with a 2,400 mAh primary cell taking one window every ten minutes [15]. Sensing draws 0.9 mA for the window duration, the microcontroller draws 6.0 mA at 64 MHz while computing, and the sub-GHz radio draws 45 mA with a 0.35 s session overhead and 0.42 s per kilobyte. Fast Fourier transform cost was estimated at six operations per butterfly [16] and inference at three cycles per multiply–accumulate.

Table 2: Classifiers compared.

Model	Parameters	Multiply–accumulates	Size at int8
Logistic regression	60	56	60 bytes
MLP, one hidden layer of 8	156	144	156 bytes
MLP, hidden layers of 32 and 16	1,076	1,024	1,076 bytes
Random forest, 50 trees	28,452	600	28,452 bytes

3. RESULTS

3.1 Model Capacity Buys Nothing

Table 3 and Figure 1 compare the classification performance and cost of the four models.

Table 3: Classification performance and model cost.

Model	Accuracy	Macro F1	Parameters	Size at int8	Multiply–accumulates
Logistic regression	97.36 %	97.37 %	60	60 B	56
MLP 14-8-4	97.36 %	97.36 %	156	156 B	144
MLP 14-32-16-4	96.94 %	96.95 %	1,076	1,076 B	1,024
Random forest, 50 trees	94.58 %	94.63 %	28,452	27.8 kB	600

The smallest model is the best. Logistic regression with 60 parameters achieves 97.36 %, exactly matching an eight-unit network and exceeding both the 1,076-parameter network and the random forest, which is worst at 94.58 % while occupying 474 times the memory of the logistic model.

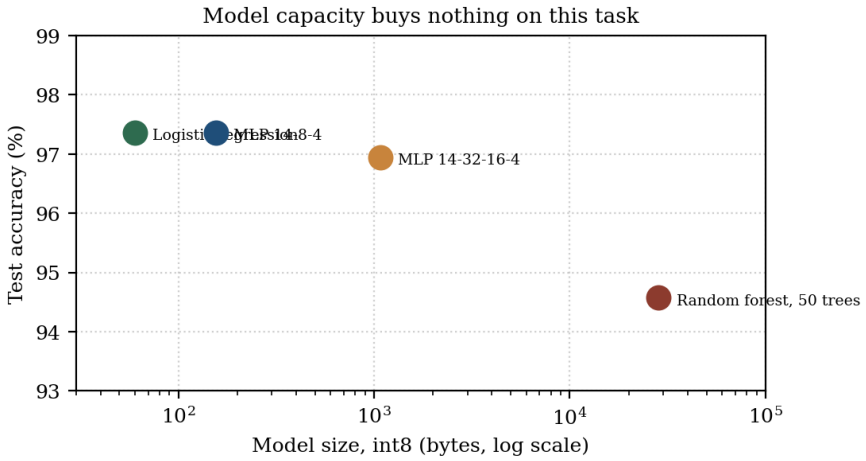


Fig.1: Test accuracy against model size, on a logarithmic size axis.

The explanation is that the discriminative work has already been done by the feature extraction. Envelope peaks at the three characteristic defect frequencies are close to a sufficient statistic for this problem, and the classifier's remaining task is to draw a nearly linear boundary between four well-separated clusters. Additional capacity has nothing to fit except noise, which is why the larger network is marginally worse rather than marginally better.

This is not a claim that neural networks are unsuitable for condition monitoring [5,6]. It is a claim about where the difficulty in this task lies, and it suggests that a deployment which computes envelope features on the device — as it must, since something has to transform the window — has already solved the hard part before the classifier is chosen.

3.2 Quantisation

Table 4 and Figure 2 give the accuracy of the largest network as its weight precision is reduced.

Table 4: Accuracy against weight precision, MLP 14-32-16-4.

Precision	Accuracy	Model size	Loss against 32-bit
32-bit float	96.94 %	4,304 B	—
16-bit	96.94 %	2,152 B	0.00 points
8-bit	96.94 %	1,076 B	0.00 points
6-bit	96.39 %	807 B	0.55 points
4-bit	95.83 %	538 B	1.11 points
3-bit	95.56 %	404 B	1.38 points
2-bit	75.97 %	269 B	20.97 points

Quantisation to 8 bits is exactly lossless on this task, and 4 bits costs 1.11 percentage points for an eightfold reduction in model size. Between 4 and 3 bits the curve remains nearly flat, and between 3 and 2 bits it collapses by 21 points.

The shape is characteristic and it has a practical reading. There is a wide plateau over which precision does not matter and a cliff beyond it, and the engineering task is to establish where the cliff is rather than to optimise along the plateau [10,11]. Choosing 8-bit weights, which every embedded inference library supports and most microcontrollers accelerate [8,9], sits comfortably inside the plateau and requires no further investigation.

It should be noted that the model being quantised is already unnecessary: the 60-byte logistic classifier outperforms it at any precision. Quantising a 4,304-byte model down to 538 bytes to fit a memory budget that a 60-byte model meets by a factor of nine is a well-executed solution to a problem that a different model selection would not have created.

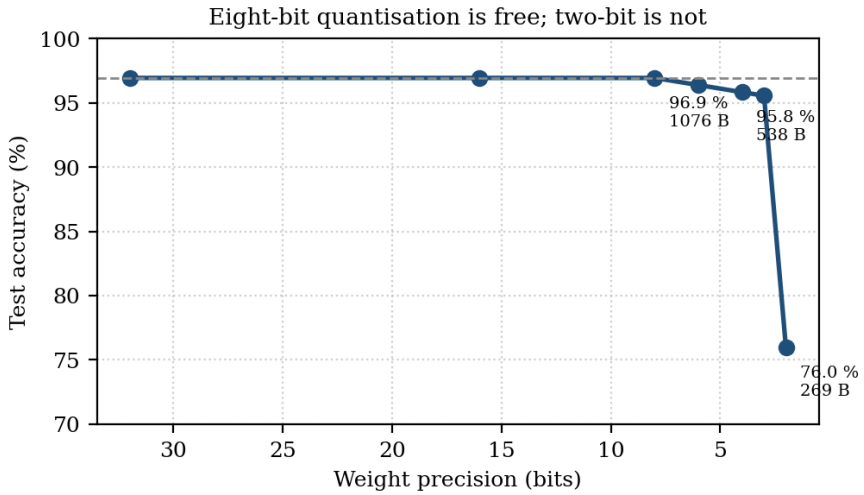


Fig.2: Accuracy against weight precision. The horizontal axis is reversed so that compression increases to the right.

3.3 What Actually Limits Accuracy

Table 5 and Figure 3 give the accuracy of the fixed trained model across a range of signal-to-noise ratios.

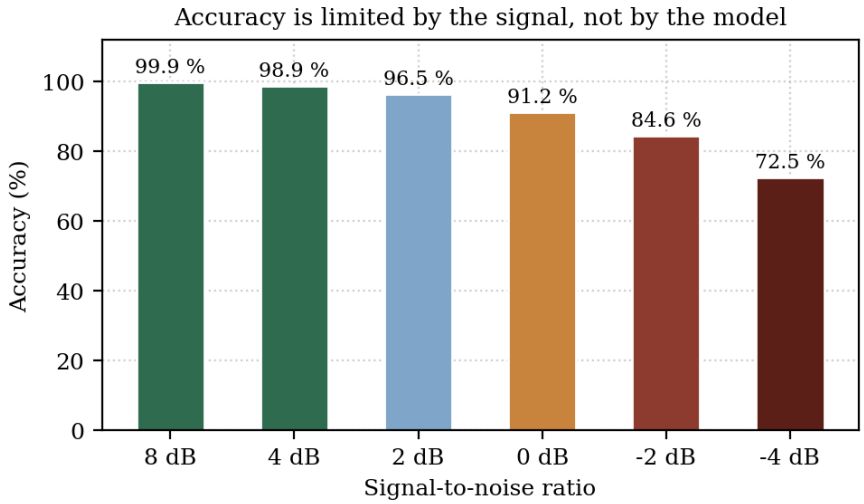


Fig.3: Accuracy against signal-to-noise ratio for the fixed trained model.

Accuracy falls from 99.88 % to 72.50 % across a 12 dB range of signal quality. Against that, the entire spread across four model architectures was 2.78 percentage points and the cost of 4-bit quantisation was 1.11.

The comparison is the point of this section. Signal quality moves accuracy by an order of magnitude more than any model or precision decision examined here, and signal quality is determined by the accelerometer, its mounting, its bandwidth and its placement relative to the

load zone — none of which is a software question [1]. A node mounted on a flexible bracket rather than a machined boss loses more accuracy than could be recovered by any amount of architecture work.

Table 5: Accuracy against signal-to-noise ratio.

Signal-to-noise ratio	8 dB	4 dB	2 dB	0 dB	−2 dB	−4 dB
Accuracy	99.88 %	98.88 %	96.50 %	91.25 %	84.62 %	72.50 %

For a system designer this reorders the development effort. Establishing the achievable signal-to-noise ratio at the intended mounting point, before selecting a classifier, bounds what any classifier can deliver and determines whether the deployment is viable at all.

3.4 Device-Level Energy

Table 6 and Figure 4 give the energy consumption and battery life under each reporting strategy.

Table 6: Energy and battery life by reporting strategy.

Strategy	Uploaded per window	Average current	Radio share	Battery life
A: stream raw window	4,096 B	0.1605 mA	94.9 %	1.54 years
B: send 14 features	56 B	0.0362 mA	77.2 %	6.81 years
C: classify on device, logistic	4 B	0.0347 mA	76.1 %	7.11 years
D: classify on device, MLP	4 B	0.0347 mA	76.1 %	7.11 years

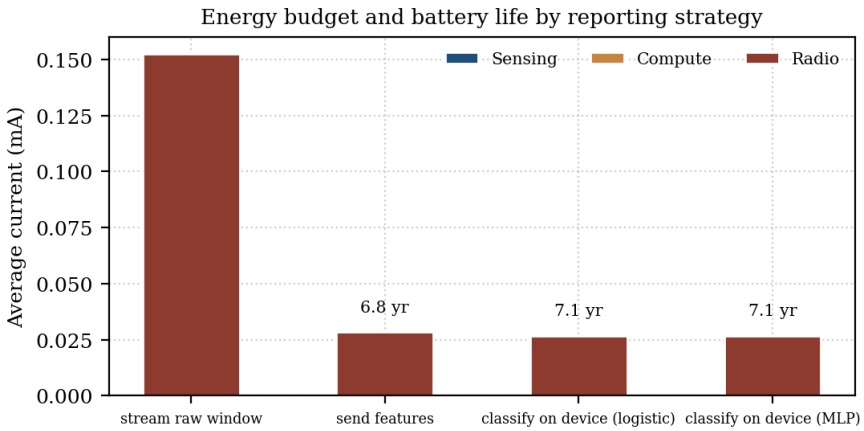


Fig.4: Average current decomposed into sensing, compute and radio, with battery life annotated.

Streaming raw windows gives 1.54 years; sending features gives 6.81 and classifying on the device 7.11. The step from raw to features is worth 4.4 times the battery life and the further step from features to a class label is worth 4 %.

The two classifiers are indistinguishable at device level, to four decimal places of average current. Inference occupies 2.6 µs for the logistic model and 48.0 µs for the network, against 2.11 ms for the fast Fourier transform that precedes both [16], so the classifier accounts for between 0.1 % and 2 % of the compute and the compute accounts for 0.06 % of the energy.

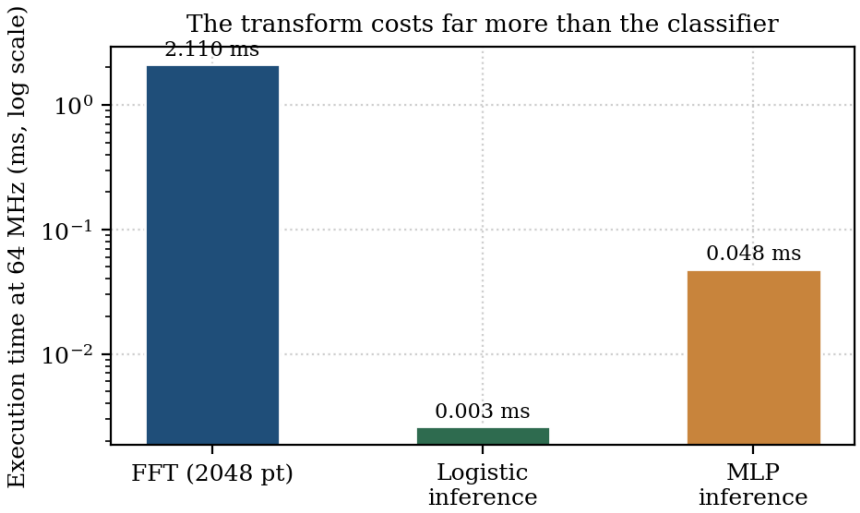


Fig.5: Execution time of each processing stage at 64 MHz, on a logarithmic scale.

The radio remains the dominant consumer under every strategy, at 76.1 % to 94.9 % of the budget, even when the payload has fallen to four bytes. The reason is the fixed session overhead: at 0.35 s of attach time per transmission, a four-byte message costs 94 % of what a fifty-six-byte message costs. Once payload has been reduced to a few bytes, further reduction achieves nothing, and the remaining lever is the reporting interval rather than the message size [15].

4. DISCUSSION

4.1 Principal Findings

Four results follow. A 60-parameter logistic classifier matched or exceeded models up to 474 times larger, so capacity is not the constraint. Weight precision is not the constraint either: 8 bits is lossless and the cliff lies between 3 and 2 bits. Signal-to-noise ratio moves accuracy by 27 percentage points across a plausible range, an order of magnitude more than any model decision. And at device level the radio consumes 76–95 % of the energy under every strategy, while inference consumes under 0.1 %.

4.2 Implications for Design

Table 7 summarises each finding, its implication and the corresponding design rule.

The first and fourth rows together describe a misallocation that this study was designed to test and found. A development effort that begins by selecting a network architecture, then prunes and quantises it to fit the microcontroller, will produce a working system whose accuracy is set by the accelerometer mounting and whose battery life is set by the radio duty cycle — neither of which the effort touched.

Table 7: Implications.

Finding	Implication	Design rule
60 parameters matched 28,452	Feature engineering has done the work	Try a linear model on physical features before any network
8-bit quantisation was lossless	Precision is on a plateau, not a gradient	Adopt int8 and stop; locate the cliff, do not optimise the plateau
Quantising an unnecessary model	Compression can solve a self-inflicted problem	Select the model before compressing it

Finding	Implication	Design rule
SNR moved accuracy by 27 points	The transducer and mounting dominate	Characterise the mounting point before selecting a classifier
Inference is 0.06 % of energy	Compute is not the budget	Do not trade accuracy for inference cost on this class of node
FFT costs 40× the inference	The transform is the compute cost	Optimise the transform, or avoid it with time-domain features
Radio is 76–95 % under every strategy	Payload reduction saturates	Once the payload is a few bytes, change the interval, not the message

The sixth row is a concrete redirection. If the transform is forty times the cost of the classifier, then the compute optimisation with any leverage is the transform, and the more interesting option is to avoid it: a classifier operating on time-domain and envelope statistics computed incrementally, without a full spectral transform, would reduce the compute term substantially. That term is only 0.06 % of the energy budget here, so the gain matters for latency and for duty-cycle headroom rather than for battery life.

4.3 Limitations

The signals are synthetic. They are generated from a physically motivated model that reproduces the characteristic defect frequencies, the resonance excitation and the shaft-rate modulation that distinguish inner from outer race faults, and they omit a great deal: variable load and speed, multiple simultaneous defects, progressive degradation rather than discrete states, gear mesh and electrical noise, mounting resonances, and the distributed and often ambiguous signatures of early-stage damage [4]. Real data would be harder in ways that would reduce all four models' accuracy, and it is possible though not demonstrated that a larger model would separate on real data where it does not here. That is the central caveat and it is the reason this study is offered as a test of a design assumption rather than as a benchmark.

The classification task is a balanced four-class problem with discrete, well-separated states. A realistic deployment faces a severely imbalanced problem in which the healthy class dominates and the operating point must be set for a very low false alarm rate, where model calibration matters more than accuracy and where the ranking between models could differ [5,6].

The device model is an energy budget rather than a measurement, and its currents are representative values for the component classes named. Transform cost is estimated from an operation count rather than profiled, and a hardware accelerator or an optimised library would change it; the radio model omits retransmission, acknowledgement, joining and time synchronisation, all of which consume sessions and make the reported battery lives optimistic.

Finally, quantisation was applied post-training and uniformly, with no quantisation-aware training, no per-channel scaling and no activation quantisation. Each of these is standard practice [9-11] and each would move the cliff in Table 4 toward lower precision, so the reported 2-bit collapse is a property of the simplest quantisation scheme rather than a limit of the approach.

5. CONCLUSION

On a bearing fault classification task built from physically motivated features, a 60-parameter logistic regression achieved 97.36 % accuracy — matching a network nearly three times its size and exceeding a random forest occupying 474 times the memory. Quantising the largest model to 8 bits was exactly lossless and to 4 bits cost 1.11 percentage points, with the cliff between 3 and 2 bits.

Neither model capacity nor weight precision therefore constrains the design. Two other things do. Signal-to-noise ratio moved accuracy from 99.88 % to 72.50 % across a 12 dB range, an order of magnitude more than every model and precision decision combined, and it is fixed by the transducer and its mounting rather than by software. And the radio consumed between 76 % and 95 % of the energy under every reporting strategy, while inference consumed less than a tenth of one per cent.

The practical conclusion is an ordering. Characterise the achievable signal quality at the intended mounting point, which bounds what any classifier can deliver; choose the reporting interval, which sets the battery life; and then select the smallest classifier that works, which on this evidence is likely to be very small indeed. Architecture search and aggressive compression address the part of the system that was never limiting.

DECLARATIONS

Ethical approval. Not applicable. This is a computational study using synthetic signals; no human participants, personal data, physical machinery or field measurements were involved.
Funding. This research received no specific grant from any funding agency in the public, commercial or not-for-profit sectors.
Conflicts of interest. The author declares no conflicts of interest, financial or non-financial, in relation to the subject matter of this manuscript.
Data disclaimer. All vibration signals are synthetic and generated from the model described in Section 2.1. No measured dataset, machine, bearing or commercial monitoring product is evaluated, and the accuracies reported are not benchmark results for any real condition-monitoring task.
Parameter disclaimer. Device currents, timings and battery capacity are representative assumptions for the component classes named. Absolute battery lives apply only to the case modelled; the comparisons follow from the structure of the budget, as set out in Section 4.3.
Data and code availability. This is a computational study; no measured or external datasets were used. All signals, models and device-level results were generated by the analysis code described in Section 2, which uses a fixed random seed so that every reported value can be reproduced exactly. The code is available from the corresponding author upon reasonable request.
Author's contribution. Altaf Mulani is the sole author of this work and was responsible for the conception and design of the study, signal modelling, feature extraction, classifier development, quantisation and energy analysis, interpretation of the results, preparation of the figures and tables, and drafting, revision and final approval of the manuscript. The author is accountable for all aspects of the work.

REFERENCES

1. Randall RB, Antoni J. Rolling element bearing diagnostics — a tutorial. Mech Syst Signal Process. 2011;25(2):485–520.
2. Antoni J. Cyclostationarity by examples. Mech Syst Signal Process. 2009;23(4):987–1036.
3. McFadden PD, Smith JD. Vibration monitoring of rolling element bearings by the high-frequency resonance technique — a review. Tribol Int. 1984;17(1):3–10.
4. Smith WA, Randall RB. Rolling element bearing diagnostics using the Case Western Reserve University data: a benchmark study. Mech Syst Signal Process. 2015;64–65:100–131.

5. Lei Y, Yang B, Jiang X, Jia F, Li N, Nandi AK. Applications of machine learning to machine fault diagnosis: a review and roadmap. Mech Syst Signal Process. 2020;138:106587.
6. Zhang S, Zhang S, Wang B, Habetler TG. Deep learning algorithms for bearing fault diagnostics — a comprehensive review. IEEE Access. 2020;8:29857–29881.
7. Banbury C, Reddi VJ, Lam M, et al. Benchmarking TinyML systems: challenges and direction. arXiv:2003.04821; 2020.
8. David R, Duke J, Jain A, et al. TensorFlow Lite Micro: embedded machine learning for TinyML systems. Proc Mach Learn Syst. 2021;3:800–811.
9. Jacob B, Kligys S, Chen B, et al. Quantization and training of neural networks for efficient integer-arithmetic-only inference. Proc IEEE Conf Comput Vis Pattern Recognit. 2018:2704–2713.
10. Krishnamoorthi R. Quantizing deep convolutional networks for efficient inference: a whitepaper. arXiv:1806.08342; 2018.
11. Gholami A, Kim S, Dong Z, Yao Z, Mahoney MW, Keutzer K. A survey of quantization methods for efficient neural network inference. arXiv:2103.13630; 2021.
12. Han S, Mao H, Dally WJ. Deep compression: compressing deep neural networks with pruning, trained quantization and Huffman coding. Proc Int Conf Learn Represent; 2016.
13. Shi W, Cao J, Zhang Q, Li Y, Xu L. Edge computing: vision and challenges. IEEE Internet Things J. 2016;3(5):637–646.
14. Rajapakse V, Karunanayake I, Ahmed N. Intelligence at the extreme edge: a survey on reformable TinyML. ACM Comput Surv. 2023;55(13s):1–30.
15. Martinez B, Monton M, Vilajosana I, Prades JD. The power of models: modeling power consumption for IoT devices. IEEE Sens J. 2015;15(10):5777–5789.
16. Cooley JW, Tukey JW. An algorithm for the machine calculation of complex Fourier series. Math Comput. 1965;19(90):297–301.
17. Pedregosa F, Varoquaux G, Gramfort A, et al. Scikit-learn: machine learning in Python. J Mach Learn Res. 2011;12:2825–2830.