

Compact Topological Descriptors for Real-Time Activity Recognition on Wearable Microcontrollers

Shuhao Dong
University of Bristol
Bristol, UK
shuhao.dong@bristol.ac.uk

Jiayu Shang
University of Bristol
Bristol, UK
jiayu.shang@bristol.ac.uk

James Pope
University of Bristol
Bristol, UK
jp16127@bristol.ac.uk

Telmo de Menezes e Silva Filho
University of Bristol
Bristol, UK
telmo.silvafilho@bristol.ac.uk

Ian Craddock
University of Bristol
Bristol, UK
Ian.Craddock@bristol.ac.uk

George Oikonomou
University of Bristol
Bristol, UK
G.Oikonomou@bristol.ac.uk

Abstract

Continuous human activity recognition (HAR) on battery-powered wearables is tightly constrained in memory and power, yet the CNNs and LSTMs that reach high accuracy are costly for always-on deployment. Topological Data Analysis (TDA) yields descriptors that capture the shape of inertial signals and are robust to high-frequency noise and cross-subject variation, but prior TDA-based HAR predominantly relies on persistence images or landscapes that are too expensive for microcontrollers. Using 1D Persistent Homology, we instead extract a compact set of scalar persistence descriptors and pair them with a small INT8-quantised dense classifier, making TDA-based HAR deployable on a microcontroller in real time. Under Leave-One-Subject-Out cross-validation on UCI-HAR and WISDM, this TDA-Dense model reaches 92.78% and 81.69% accuracy, competitive with Conv1D and DeepConvLSTM at markedly lower cross-subject variance, while matching conventional feature engineering at much lower input dimensionality. On a Cortex-M33, it needs only 1.81 KB of RAM and 0.148 mJ per inference, 6.4× and 51× lower energy than the two deep baselines respectively, positioning compact topological descriptors as an energy-efficient alternative for continuous edge HAR.

CCS Concepts

• **Computing methodologies** → **Machine learning**; • **Computer systems organization** → **Embedded systems**.

Keywords

TinyML, Edge Computing, Internet of Things, Human Activity Recognition

1 Introduction

Deploying neural networks on edge devices for real-time Human Activity Recognition (HAR) is a critical enabler for continuous monitoring systems, providing movement context for downstream algorithms such as gait and physical activity level analysis. However, continuous on-device classification is strictly bottlenecked by available RAM, flash storage and battery-imposed energy budgets [10].

Deep learning architectures that ingest raw inertial streams and learn temporal dependencies report high accuracy [5], but continuously processing high-dimensional windows on a microcontroller

demands prolonged active CPU time that rapidly drains the power budget, and such models generalise poorly to unseen users under sensor noise and placement variation.

As an efficient alternative, we pair a lightweight dense classifier with explicit feature engineering. Extracting features upfront offloads computation from the network onto the Digital Signal Processing (DSP) extensions of modern microcontrollers, enabling fast inference while limiting input dimensionality and memory footprint.

We adopt descriptors from Topological Data Analysis (TDA) [4], which summarise the shape of an inertial window and have previously been explored for HAR. Our aim is therefore not to introduce TDA as a new HAR representation, but to characterise whether a compact set of scalar 1D persistence descriptors can provide an efficient feature family for microcontroller-scale deployment. Unlike high-dimensional persistence images or landscapes from time-delay embedding, these descriptors are fixed-length, physically interpretable, and computable using a lightweight prominence-based pass. Combined with post-training quantisation (PTQ), this yields a TDA-based HAR pipeline that can run end-to-end on a microcontroller in real time. The primary contributions of this work are:

- We characterise a scalar 1D persistent-homology feature set as an embedded HAR design point, evaluating its feasibility within a microcontroller budget and deploying the resulting INT8-quantised pipeline end-to-end on a Cortex-M33 in real time.
- A controlled ablation under LOSO cross-validation on UCI-HAR [11] and WISDM [7] that, under an identical classifier, finds the persistent-homology descriptors competitive with a strong conventional feature set at substantially lower input dimensionality.
- An on-device characterisation (latency, memory and flash footprint, and measured energy per inference) quantifying the practical cost of each model family on the Cortex-M33, demonstrating higher energy-efficiency of topological descriptors.

2 Related Work

Continuous HAR relies on extracting meaningful patterns from multi-dimensional time-series data. While early systems used manual statistical feature engineering, the field has largely shifted to

deep architectures, particularly Convolutional Neural Networks [14], Long Short-Term Memory networks [13] and Gated Recurrent Units [6], which learn temporal features directly from raw inertial data and achieve strong accuracy. However, their large multiply-accumulate counts and memory footprints make them unsuitable for always-on deployment within the energy budgets of battery-operated wearables.

To bridge this gap, Tiny Machine Learning (TinyML) compresses deep architectures for microcontroller deployment [1] using techniques such as INT8 quantisation [16], pruning [12] and efficient memory allocation [9], with recent work targeting efficient HAR via DeepConvLSTM and lightweight 1D CNNs [15]. Yet these compressed temporal models still demand significant SRAM for sliding-window buffers and recurrent state, and remain susceptible to generalisation drops from subject-specific sensor displacement noise.

A complementary direction supplies richer hand-crafted features to inherently lightweight models. TDA, specifically 1D Persistent Homology, captures the shape of a time-series by tracking the persistence of local extrema, yielding invariants such as persistence diagrams and entropy [4]. Prior HAR studies have explored topological representations, including compact 1D descriptors [2] as well as higher-dimensional encodings such as persistence images and landscapes [3, 8]. However, less attention has been paid to their role as part of a complete TinyML design space: comparison against conventional peak, frequency and shape descriptors under the same lightweight classifier, and measured latency, memory and energy cost on a microcontroller. This systems-level characterisation is the gap addressed in this work.

3 Methods

3.1 Feature Extraction

To reduce the computational burden of end-to-end temporal modelling on the embedded microcontroller, our pipeline extracts compact feature vectors from each sliding window before inference. For a window X of length N on a given sensor axis, we compute 8 persistence-based descriptors and 6 standard statistical features. Because raw inertial signals can contain high-frequency artefacts caused by loose attachment or small mechanical vibrations, the signal is first smoothed by a 3-sample moving average, yielding X' .

We characterise the shape of X' using 1D Persistent Homology via superlevel-set filtration [4]. For 1D signals, the persistence of each local maximum coincides with its topographic prominence, obtained in a single pass over X' . Let $P = \{p_1, p_2, \dots, p_m\}$ be the set of prominences of the local maxima of X' , where p_i is the persistence of the i th detected peak, and let $p_{(1)} \geq p_{(2)} \geq \dots \geq p_{(m)}$ denote the same values sorted in descending order.

Since full vectorisations of the persistence diagram, such as persistence images, are too costly to compute within a microcontroller budget, we instead summarise the diagram into a small set of scalar descriptors that each abstract a high-level property of its persistence structure. The following 8 descriptors are then computed:

Betti-0 count (m) is the total number of topological features detected in the window. **Total topological energy** (U) is the sum of all persistences. **Maximum persistence** is the largest detected persistence, $p_{(1)}$, and **second largest persistence** is $p_{(2)}$. **Mean persistence** (μ_p) is U/m . **Variance of persistence** (σ_p^2) is the

spread of the persistences. **Persistence entropy** (H) is computed from the normalised prominences $\hat{p}_i = p_i/U$ by Eq. (1):

$$H = - \sum_{i=1}^m \hat{p}_i \ln(\hat{p}_i + \epsilon), \quad (1)$$

where $\epsilon = 1 \times 10^{-12}$ avoids undefined logarithms. The **top-3 persistence ratio** quantifies the dominance of the three strongest components:

$$R_{top3} = \frac{p_{(1)} + p_{(2)} + p_{(3)}}{U} \quad (2)$$

We design this set to be complementary rather than redundant: the count (m) and total energy (U) capture how many structural features are present and their aggregate strength; the ordered persistences ($p_{(1)}, p_{(2)}$) and their mean isolate the dominant motion cycles; and the variance, entropy, and top-3 ratio describe how persistence is distributed, separating rhythmic motion concentrated in a few strong peaks from irregular motion spread across many weak ones. We retain 6 statistical features (empirical mean, minimum, maximum, standard deviation, Root Mean Square (RMS), and the mean of a 25-sample moving-average signal) alongside them because persistence is amplitude-invariant by construction, so absolute level, spread, and energy supply complementary magnitude information the topological descriptors discard.

Additionally, to form a strong conventional baseline for comparison, we compute 18 standard time- and frequency-domain features per axis that are commonly used in HAR: 7 frequency-domain features from the FFT magnitude spectrum (spectral energy, dominant frequency, centroid, entropy, flatness, low-to-high band ratio, and dominant-peak magnitude), 6 distributional shape features (skewness, kurtosis, mean-crossing rate, interquartile range, median absolute deviation, and signal magnitude area), and 5 peak features computed from the raw signal without persistence information (count, mean, standard deviation, and maximum of peak heights, and mean inter-peak interval). Features are extracted independently for each axis and concatenated to form the dense-model input.

3.2 Model Design and Evaluation

We evaluate six models across three architectural families. Four share an identical dense classifier, training schedule, and pruning procedure, differing only in input representation: **Stat-Dense** (statistical features alone), **TDA-Dense** (statistical plus topological descriptors), **Conv-Dense** (statistical plus conventional features), and **ConvTDA-Dense** (all three). This shared classifier lets accuracy differences be attributed to the feature sets rather than the model. The remaining two operate end-to-end on raw sliding windows (3 s window, 1 s step) as shown in Fig. 1.

We use two public datasets with complementary sensing conditions in terms of available channels and sensor placement. UCI-HAR contains waist-mounted accelerometer and gyroscope data from 30 subjects over six activities; WISDM provides pocket-carried accelerometer data from 36 subjects. For each dataset, magnitude axes are appended to the raw channels: the dense models operate on per-axis features extracted from this representation, while Conv1D and DeepConvLSTM take the raw windows directly.

For the dense models, feature selection is applied within each training fold using Random Forest Gini importance, retaining the

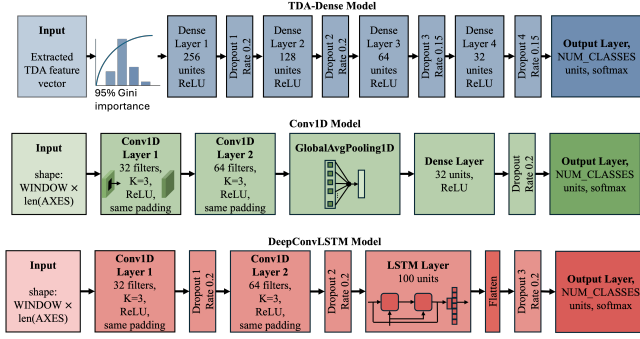


Figure 1: Implemented model structures.

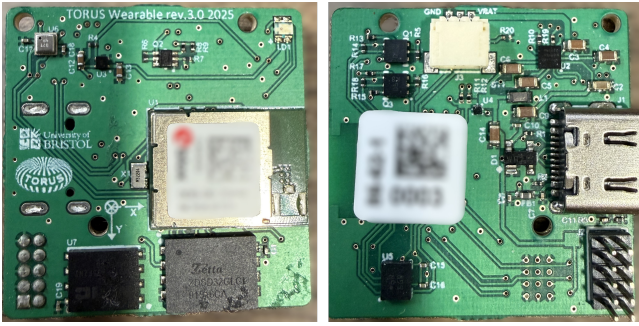


Figure 2: Custom wearable device for real-time on-device HAR.

top-ranked subset reaching 95% of cumulative importance to keep the classifier compact for deployment. All models use PTQ, favouring quantisation-friendly operators and bounded activations calibrated from a representative UCI-HAR dataset. Evaluation follows LOSO cross-validation, and we report accuracy, weighted F1, and cross-fold standard deviations under identical windowing, normalisation, and label definitions.

3.3 On-device Performance

We deploy the models on a custom wearable (Fig. 2) that acquires inertial data at 50 Hz from a BMI270 6-axis IMU. Computation runs on a Nordic nRF5340 SoC, whose ARM Cortex-M33 core provides 1 MB flash and 512 KB RAM under the Zephyr RTOS.

After PTQ, calibrated with a representative subset of UCI-HAR training data, the INT8 models were converted to C byte arrays and integrated through TensorFlow Lite for Microcontrollers (TFLM), compiled with CMSIS-NN kernels so that INT8 operations use optimised Arm Cortex-M DSP instructions. The same firmware implements the feature extraction used by the dense models.

We report four on-device metrics. Inference latency is measured with the hardware cycle counter (`k_cycle_get_32()`). Flash usage is the stored model array size; RAM usage is the allocated TFLM tensor arena, excluding RTOS and library overhead. Energy is measured with a Nordic Power Profiler Kit II (PPK2) in source-meter mode at 3.7 V and 10 kHz (accuracy $\pm 10\%$, offset $\pm 2\%$). For each model, a 3-minute current trace is recorded at a 1 Hz inference rate;

individual events are segmented using a 1.5 mA threshold with a minimum-duration constraint from cycle counts, and active energy per inference is computed by numerical integration over each event window.

4 Results and Discussion

4.1 INT8-quantised Model Performance

The LOSO results are presented in Table 1 and the subject-level distributions in Fig. 3. On UCI-HAR, TDA-Dense reaches 92.78% accuracy, matching or exceeding both deep baselines, while models with conventional features are marginally higher. Augmenting the statistics-only baseline with topological descriptors raises mean accuracy by 4.31% and lifts the minimum accuracy from 73.68% to 80.17%, the highest of any model; a Wilcoxon signed-rank test confirms this gain is significant on both datasets ($p < 0.001$), with maximum per-subject gains exceeding 18% and 30% on UCI-HAR and WISDM respectively. By contrast, Conv1D falls to 65.65% on its worst subject, indicating that despite competitive mean scores the temporal model can sometimes fail substantially on individual users.

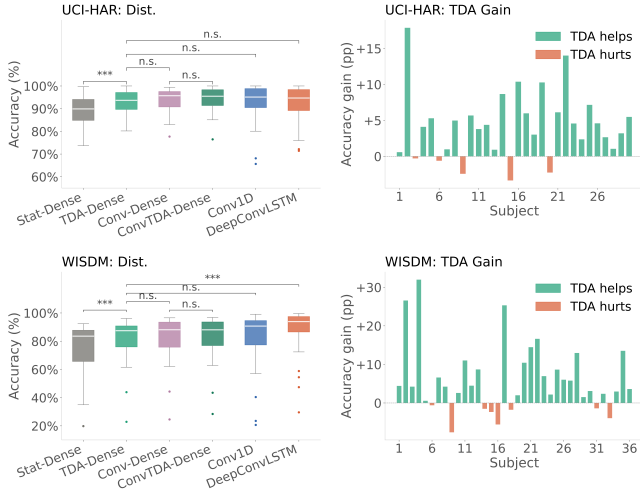
A central concern is whether these topological descriptors merely re-express conventional time- and frequency-domain features. Two comparisons address this directly. First, TDA-Dense is statistically indistinguishable from the strong conventional baseline Conv-Dense on both datasets ($p = 0.0767$ on UCI-HAR, $p = 0.4321$ on WISDM). Second, although adding the topological descriptors on top of the conventional set (ConvTDA-Dense) yields no significant accuracy gain over Conv-Dense ($p = 0.0615$ and $p = 0.3764$), its mean accuracy and weighted F1 are the highest among all feature-based models and beat temporal models in UCI-HAR. We therefore do not claim that persistent homology improves accuracy over conventional engineering; rather, it attains equivalent or higher accuracy from a more compact per-axis representation (14 versus 24 features) obtained in a single prominence pass, and, as Section 4.2 shows, at substantially lower on-device cost. The two descriptor families may be largely interchangeable in accuracy for this task, and the case for topology rests on efficiency.

Across both datasets the feature-based dense models show markedly lower cross-subject variance than the deep baselines. TDA-Dense attains the lowest weighted-F1 standard deviation of any model on both datasets, and its accuracy standard deviation is nearly 40% below the deep models on UCI-HAR and similarly tighter on WISDM. Operating on topological shape rather than raw amplitude yields more predictable performance across subjects, a practical advantage for deployment on unseen users.

On WISDM, DeepConvLSTM attains the highest mean accuracy, exceeding TDA-Dense by 6.19% ($p < 0.001$). This gap is expected: WISDM’s pocket-mounted acceleration carries richer temporal dynamics that recurrent layers exploit but a permutation-insensitive dense classifier cannot. Otherwise TDA-Dense is statistically comparable to Conv1D on both datasets and to DeepConvLSTM on UCI-HAR (Fig. 3). As Section 4.2 quantifies, the recurrent model’s accuracy advantage on WISDM is obtained at an order of magnitude higher energy per inference. A minority of subjects show slight degradation under topological features, suggesting that for highly

Table 1: Performance comparison among six INT8-quantised models.

Model	UCI-HAR			WISDM		
	Mean Acc	Min	Weighted F1	Mean Acc	Min	Weighted F1
Stat-Dense	88.48±7.31	73.68	87.87±7.90	75.61±17.42	19.78	76.44±15.71
TDA-Dense	92.78±5.36	80.17	92.54± 5.67	81.69±15.30	22.86	82.68±13.37
Conv-Dense	93.55±5.52	77.71	93.22±6.23	82.58±15.75	24.53	83.52±14.00
ConvTDA-Dense	94.19±5.32	76.47	93.84±6.14	83.23±14.89	28.40	84.09±13.42
Conv1D	92.74±8.75	65.65	92.40±9.39	82.86±19.74	20.65	83.28±17.94
DeepConvLSTM	91.79±8.40	71.43	91.35±9.15	87.88±16.07	29.59	88.34±14.52

**Figure 3: Subject-level accuracy distribution under LOSO with Wilcoxon test results, and the effect of TDA features on both datasets.**

idiosyncratic movement the dominant persistence descriptors can suppress discriminative micro-variations.

4.2 Edge Performance

Idle current during sensor acquisition and RTOS scheduling averaged 325.1 μ A, the power floor against which inference overhead is measured.

Table 2 summarises deployment cost. The difference between total latency and inference time is the preprocessing cost: topological feature extraction (6.14 ms) for TDA-Dense, conventional extraction (15.45 ms) for Conv-Dense, and input reshaping (0.61 ms) for the neural baselines. The TDA-versus-Conv-Dense pair is the most informative comparison, as both feed an identical classifier: the gap is purely feature-extraction cost. Topological extraction is 2.5 $\times$ cheaper in preprocessing time and yields 2.3 $\times$ lower energy per inference (0.148 vs. 0.338 mJ), because the conventional FFT and percentile computations keep the core active longer. Against the deep baselines, TDA-Dense completes end-to-end in 9.49 ms, over 7 $\times$ and 57 $\times$ faster than Conv1D and DeepConvLSTM, whose raw-window processing dominates latency.

The footprints reveal a complementary trade-off. Conv1D has the smallest flash (17.51 KB) through convolutional weight-sharing but demands 10 $\times$ more SRAM (18.85 KB) than TDA-Dense for intermediate feature maps; DeepConvLSTM is the most expensive on both axes (84.35 KB flash, 28.66 KB RAM) due to recurrent state. TDA-Dense uses 71.67 KB flash for its dense weights, but extracting features upfront shrinks the runtime tensor arena to just 1.81 KB, since it need only hold the small dense forward pass rather than a full sliding-window buffer. The reported RAM is the tensor arena; feature extraction uses a separate static buffer, a few hundred bytes for TDA-Dense against several kilobytes of FFT and sorting buffers for Conv-Dense, which would widen the RAM gap between the two feature-based models further.

Fig. 4 shows the median inference event for each model, selected from steady-state execution to exclude transients. Active current is similar across models (3.27 to 3.85 mA), confirming that Cortex-M33 core power is set by operating voltage and clock rather than the specific computation; consequently, energy per inference is driven by the duration of active computation. TDA-Dense consumes 0.148 mJ, 6.4 $\times$ less than Conv1D (0.953 mJ) and 51 $\times$ less than DeepConvLSTM (7.533 mJ). This is the central result: offloading temporal feature learning to deterministic TDA extraction and feeding a compact descriptor to a small dense classifier yields the lowest energy per inference while remaining competitive in accuracy compared to temporal models and conventional feature engineering.

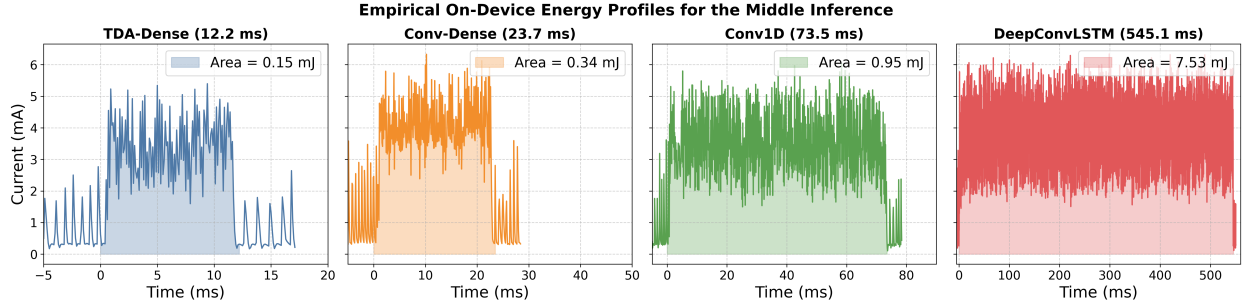
Energy is integrated over the measured active-current window, which exceeds the cycle-counted Total Latency by roughly 2.5 ms; it therefore reflects the full active pulse rather than Total Latency times mean current. We attribute this margin to threshold imprecision in segmenting active from idle states and to RTOS scheduling and peripheral startup outside the cycle-counted region.

5 Conclusion and Future Work

We presented a systems characterisation of a lightweight HAR pipeline that pairs deterministic topological descriptors with a small quantised dense classifier, and made TDA-based HAR deployable on a microcontroller in real time. Under LOSO validation on two widely-used public datasets, the compact persistence descriptors behaved as a competitive and complementary feature family: they match deep temporal networks in accuracy with markedly lower cross-subject variance, and they equal a strong conventional feature set from roughly half the per-axis dimensionality. On-device profiling on a Cortex-M33 microcontroller showed this at the smallest RAM footprint and lowest energy per inference of the deployed

Table 2: On-device model latency, energy consumption and footprint.

Model	Inference Count	Inference Time (ms)	Total Latency (ms)	Energy per Inference (E) (mJ)	Active Current (mA)	Flash footprint (KB)	RAM footprint (KB)
TDA-Dense	178	3.35 ± 0.01	9.49 ± 0.05	0.148 ± 0.004	3.27	71.67	1.81
Conv-Dense	179	4.98 ± 0.04	20.43 ± 0.11	0.338 ± 0.005	3.85	95.87	2.27
Conv1D	179	70.27 ± 0.09	70.88 ± 0.10	0.953 ± 0.009	3.50	17.51	18.85
DeepConvLSTM	179	542.09 ± 0.28	542.70 ± 0.28	7.533 ± 0.061	3.73	84.35	28.66

**Figure 4: Current profiles of representative median inference events for the four deployed models.**

models, including $2.3\times$ lower energy than the conventional-feature classifier that uses the same architecture. We therefore position topological descriptors not as a replacement for conventional feature engineering in our setting, but as a compact, noise-robust, and energy-efficient component for continuous edge HAR.

Future work. Our central finding, that scalar topological descriptors match conventional features in accuracy and earn their place through efficiency rather than discriminative power, rests on two datasets and a single dense classifier; confirming it requires broader validation across larger activity sets, additional sensor placements, and more complex sensing tasks where the two feature families may diverge. The minority of subjects that degrade under topological features motivates adaptive, per-subject weighting that blends topological and statistical descriptors rather than applying one fixed representation to every user. Finally, because 0-dimensional persistence on a 1D signal reduces to peak prominence, genuinely topological structure beyond conventional statistics may instead emerge from higher-dimensional persistence computed on delay-embedded or multivariate windows; whether such features are both discriminative for real-time HAR and cheap enough for a microcontroller is an open question that would extend this characterisation toward a stronger topological contribution.

Acknowledgments

This work was supported by the TORUS Project funded by the UK Engineering and Physical Sciences Research Council (EPSRC), [Grant EP/X036146/1]. For the purpose of open access, the author(s) has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission.

References

- [1] Youssef Abadade, Anas Temouden, Hatim Bamoumen, Nabil Benamar, Yousra Chtouki, and Abdelhakim Senhaji Hafid. 2023. A Comprehensive Survey on TinyML. *IEEE Access* 11 (2023), 96892–96922.
- [2] Redona Brahmetaj, Elena Botti, Ivan Bautmans, Eva Swinnen, and Bart Jansen. 2026. Topological Versus Spatiotemporal Gait Parameters for Fall Risk Detection with IMU Sensors. In *Sensor-Based Activity Recognition and Artificial Intelligence*, Özlem Durmaz Incel, Jingwen Qin, Gerald Bieber, and Arjan Kuijper (Eds.). Springer Nature Switzerland, Cham, 156–171.
- [3] Sunil Chaudhari, Sanjay Kumar Singh, and Milan Joshi. 2022. A review on topological data analysis in human activity recognition. *Applications of machine intelligence in engineering* (2022), 577–590.
- [4] Frédéric Chazal and Bertrand Michel. 2021. An introduction to topological data analysis: fundamental and practical aspects for data scientists. *Front. Artif. Intell.* 4 (2021), 667963.
- [5] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2021. Deep Learning for Sensor-based Human Activity Recognition: Overview, Challenges, and Opportunities. *ACM Comput. Surv.* 54, 4, Article 77 (May 2021).
- [6] Saurabh Gupta. 2021. Deep learning based human activity recognition (HAR) using wearable sensor data. *Int. J. Inf. Manag. Data Insights* 1, 2 (2021), 100046.
- [7] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. 2011. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.* 12, 2 (March 2011), 74–82.
- [8] Yushi Liu, Kamen Ivanov, Junxian Wang, Fuhai Xiong, Jiahong Wang, Min Wang, et al. 2024. Topological Data Analysis for Robust Gait Biometrics Based on Wearable Sensors. *IEEE Trans. Consum. Electron.* 70, 2 (2024), 4910–4921.
- [9] Yu-Yuan Liu, Hong-Sheng Zheng, Yu Fang Hu, Chen-Fong Hsu, and Tsung Tai Yeh. 2024. TinyTS: Memory-Efficient TinyML Model Compiler Framework on Microcontrollers. In *2024 IEEE Int. Symp. on High-Perf. Comp. Arch. (HPCA)*. 848–860.
- [10] Leandro Antonio Pazmiño Ortiz, Ivonne Fernanda Maldonado Soliz, and Vanessa Katherine Guevara Balarezo. 2025. Advancing TinyML in IoT: A Holistic System-Level Perspective for Resource-Constrained AI. *Future Internet* 17, 6 (2025).
- [11] Jorge Reyes-Ortiz, Davide Anguita, Alessandro Ghio, Luca Oneto, and Xavier Parra. 2013. Human Activity Recognition Using Smartphones. UCI Machine Learning Repository. DOI: <https://doi.org/10.24432/C54S4K>.
- [12] Muhammad Yasir Shabir, Gianluca Torta, and Ferruccio Damiani. 2025. TinyML model compression: A comparative study of pruning and quantization on selected standard and custom neural networks. *Telecommun. Syst.* 88, 4 (2025), 1–21.
- [13] Chamani Shiranthika, Nilantha Premakumara, Huei-Ling Chiu, Hooman Samani, Chathurangi Shyalika, and Chan-Yun Yang. 2020. Human Activity Recognition Using CNN & LSTM. In *2020 5th Int. Conf. on Inf. Technol. Res. (ICITR)*. 1–6.

- [14] Pratibha Tokas, Vijay Bhaskar Semwal, and Surabhi Verma. 2025. A Real-Time Deployable Attention-Driven CNN–LSTM Framework for Human Activity Recognition Using Wearable Sensor. *IEEE Sens. J.* 25, 21 (2025), 40461–40471.
- [15] Haotian Zhou, Xiujun Zhang, Yu Feng, Tongda Zhang, and Lijuan Xiong. 2025. Efficient human activity recognition on edge devices using DeepConv LSTM architectures. *Sci. Rep.* 15, 1 (2025), 13830.
- [16] Shaojie Zhuo, Hongyu Chen, Ramchalam Kinattinkara Ramakrishnan, Tommy Chen, Chen Feng, Yicheng Lin, et al. 2022. An Empirical Study of Low Precision Quantization for TinyML. arXiv:2203.05492 [cs.LG] <https://arxiv.org/abs/2203.05492>