engineering proceedings

MDPI

Proceeding Paper

Automatic Drinking Event Recognition and Fluid Intake Estimation in Sport Using an Embedded Wearable Sensor †

Timothy Malche

Department of Computer Applications, Manipal University Jaipur, Jaipur 303007, Rajasthan, India; timothy.malche@jaipur.manipal.edu

† Presented at the 13th International Electronic Conference on Sensors and Applications, 18–20 November 2026; Available online: https://sciforum.net/event/ECSA-13.

Abstract

Proper hydration is important for athletic performance and recovery, but many athletes do not accurately track how much water or energy fluid they drink during training. At present, many systems use bottles with hardware like weight sensors, liquid-level sensors, and flow sensors. This study proposes a wearable sensor that uses an embedded accelerometer sensor and TinyML running on a microcontroller to automatically detect drinking activity and estimate fluid intake. A three-axis accelerometer and a microcontroller were mounted on a sports water bottle. Data were collected from different participants performing three activities such as idle, pickup, and active drinking with an anomaly detection algorithm. When a player is not performing any of the predefined activities which are related to drinking detection, the anomaly detection algorithm classifies it as a different activity and is hence not included in the analysis. Features were extracted from the accelerometer signals and used to train a neural network. Detected drinking events and timestamps are transmitted through Bluetooth Low Energy. Fluid intake was estimated from the duration and frequency of drinking events. The TinyML model achieved 96.6% classification accuracy across the three states on held-out test data (macro F1 0.954, ROC AUC 0.98) with an inference latency of <5 ms. The system operated continuously while remaining within the available power budget. The system successfully recorded drinking events and generated session-level hydration logs for all participants. The results of the proposed system show that a TinyML model can accurately recognize drinking activities and support automatic hydration monitoring. The system offers session-level drinking logs and fluid intake estimates that can assist athletes to stay optimally hydrated. Future work will include adding data from additional participants, personalized hydration recommendations, and real-time alerts when fluid intake is less than the recommended levels.

Keywords: TinyML; embedded machine learning; wearable sensor; accelerometer; drinking detection; hydration monitoring; neural network; INT8 quantization; anomaly detection; Gaussian mixture model; Bluetooth Low Energy; SDG 3 (Good Health and Well-being)

Check for updates

Academic Editor: Jean-Marc Laheurte

Published: 3 November 2026

Copyright: © 2026 by the author. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

Fluid loss during training degrades endurance, thermoregulation and cognitive function once body-mass loss exceeds roughly 2% [1], and restoring fluid afterwards improves subsequent performance, with the largest gains in the heat [2]. Yet most athletes rely on thirst or on a rough recollection of how many bottles they emptied, so intake during a session is rarely known with any precision. A sensor that records when and for how long an athlete drinks, with no manual logging, would close this gap.

 https://doi.org/10.3390/engproc2026163071

Existing systems measure the liquid directly with load cells, capacitive or impedance level sensing, or in-line flow metres [3], which is accurate but requires a purpose-built container. An alternative is to sense the motion of the container: a small inertial node attached to any bottle observes the tilt and return trajectory of a drink, and the gravity vector rotating through the sensor frame gives a signature largely independent of the bottle itself [4,5]. This is the principle of wrist-worn activity recognition [6,7], where drinking and eating gestures are well studied [8,9], but a bottle-mounted node avoids confusion with the many other arm movements that occur in sport.

Running the classifier on the node itself, rather than streaming raw samples to a phone, is what makes a small battery-powered device practical. Tiny machine learning (TinyML) places a quantized network inside the microcontroller so that only compact event records leave the device, which cuts radio duty cycle, keeps raw motion data private and removes any dependence on a network [10–13].

This paper presents a bottle-mounted sensor node based on the Arduino Nicla Sense ME that classifies one-second accelerometer windows into *drinking*, *picked* (bottle handled but not consumed from) and *not_drinking* (idle), flags motion outside these states with a Gaussian mixture model (GMM) anomaly detector, and turns the event stream into a session-level hydration log sent over Bluetooth Low Energy (BLE), supporting low-cost preventive monitoring in line with United Nations Sustainable Development Goal 3, Good Health and Well-being.

2. Materials and Methods

2.1. System Architecture

Figure 1 shows the system. All signal processing and inference run on the sensor node: each one-second window is converted into a class label and an anomaly score, consecutive labels are merged into drinking events, and each event is sent as a BLE notification to a smartphone or gateway that keeps the session log. No raw acceleration data leave the bottle.

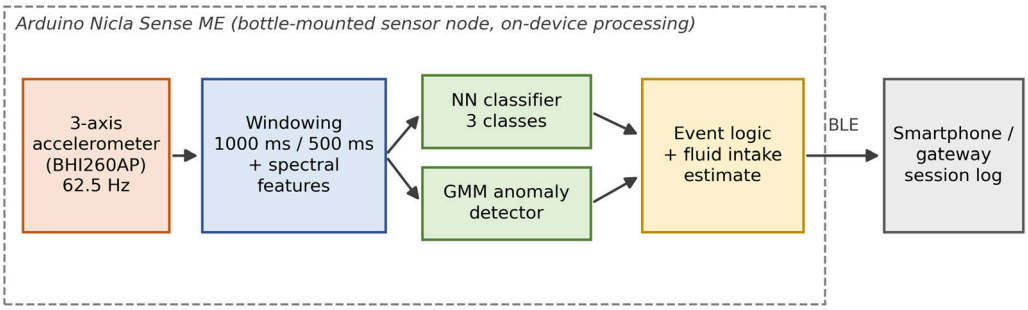


Figure 1. System overview. Sensing, feature extraction, classification, anomaly detection and event logic all run on the bottle-mounted Nicla Sense ME; only event records are transmitted over BLE.

2.2. Sensor Node and Choice of Platform

The sensor node is an Arduino Nicla Sense ME (22.86 × 22.86 mm, Arduino, Monza, Italy), which integrates a Bosch BHI260AP smart sensor (Bosch Sensortec GmbH, Reutlingen, Germany) hub with a built-in six-axis IMU and an ANNA-B112 module (Nordic nRF52832, Arm Cortex-M4F at 64 MHz, 64 kB RAM, 512 kB flash) that hosts the application and the BLE stack [14,15]. Only the three accelerometer axes are used in this work.

Figure 2 shows the assembled node. The board sits in a translucent snap-fit enclosure that adds only a few millimetres to each dimension, powered by a single-cell LiPo pouch cell (3.7 mAh); the USB-C connector is used for charging and flashing. The node is attached near the base of the bottle with double-sided adhesive, oriented so that the board *z*-axis runs

along the bottle axis. This position keeps it clear of the carry strap and the drinking end and far from the grip point, so a given tilt rate produces the largest tangential acceleration at the sensor.

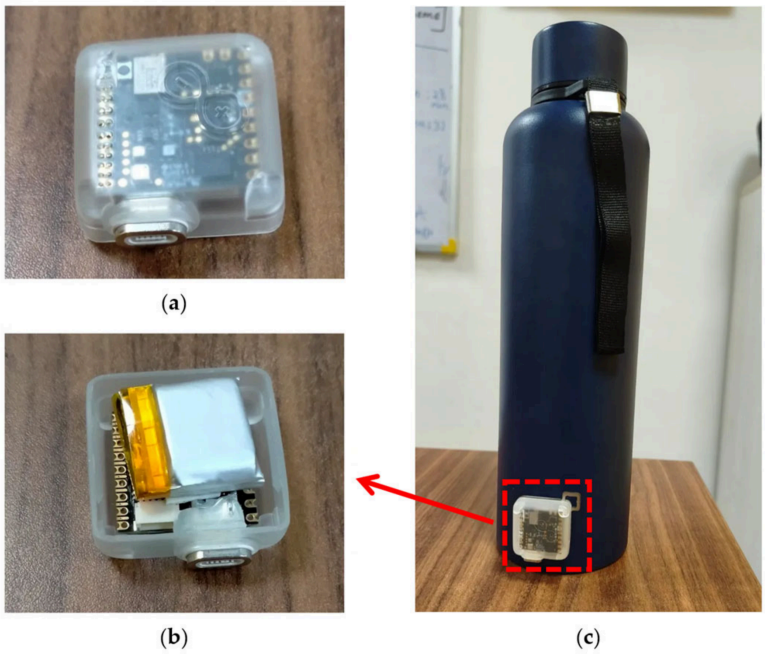


Figure 2. Sensor node: (**a**) Nicla Sense ME in its enclosure; (**b**) internal view showing the board and the single-cell LiPo battery; (**c**) node mounted near the base of the sports bottle used in the study.

The Nicla Sense ME was chosen over other TinyML-capable boards for four reasons. First, footprint: it is small enough to sit on the outside of a bottle without changing how the bottle is gripped or balanced. Second, the BHI260AP is a smart sensor hub rather than a bare IMU; it has its own core that samples the accelerometer and buffers it, so the application processor is not tied up with sampling. Third, the ANNA-B112 module combines a Cortex-M4F with floating-point unit and a BLE 5 radio in one package, so INT8 kernels run efficiently and no separate radio is needed. Fourth, the board carries an integrated LiPo charger and is well supported by the embedded machine learning toolchains used here. An Arduino Nano 33 BLE Sense offers a comparable radio and IMU in a 45 × 18 mm form factor, and an ESP32-class board computes more at a higher idle current.

2.3. Data Collection and Activity Classes

Acceleration was sampled at 62.5 Hz, which is sufficient for the low-frequency tilt and return motion of a drink (a full sip cycle lasts 2–5 s) while keeping the DSP cost small. Data was collected from five participants and streamed over a serial link to a host PC, with each recording labelled on capture. Three classes were defined (Figure 3): *not_drinking*, bottle at rest on a surface; *picked*, bottle lifted, carried or without being consumed from; and *drinking*, bottle raised, tilted to the mouth and returned. Motion outside these classes, such as the bottle being dropped, shaken or carried in a bag, is left to the anomaly detector rather than being given its own class.

Recordings were assigned to a training set and a held-out test set, and 20% of the training windows were then set aside as a validation set. The test set was kept fully separate from training, and all results in Section 3 are reported on the validation and test sets.

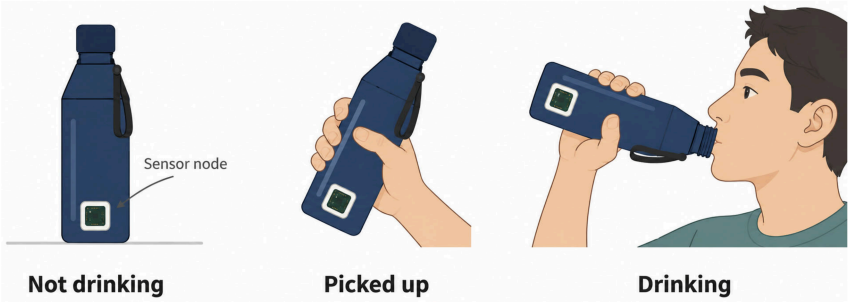


Figure 3. The three activity classes as seen by the bottle-mounted node: *not_drinking* (bottle at rest), *picked* (bottle handled but not consumed from) and *drinking* (bottle tilted to the mouth).

2.4. Windowing and Spectral Features

The processing pipeline takes the three raw axes into 1000 ms windows with a 500 ms stride, passes them through one spectral analysis block, and feeds the resulting features to a neural-network classifier and a GMM anomaly in parallel. In a raw *drinking* window, the dominant information is the slow rotation of the gravity vector, visible as the x and y channels moving in opposite directions during the tilt while z stays close to 9.8 m/s^2.

Each axis is passed through the spectral analysis block (axis scale 0.0746, decimation 1, sixth-order low-pass filter, 32-point FFT with log spectrum, no frame overlap). The parameters were initialized by an automated parameter search and kept because the resulting feature space separated the classes well (Section 3.1). The 30.6 Hz cut-off is only slightly below the 31.25 Hz Nyquist frequency, so the filter acts only as an anti-aliasing guard and the discriminative information lies in the shape of the log power spectrum. For each axis the block outputs RMS, skewness, kurtosis, spectral skewness, spectral kurtosis and 16 log-spectral-power bins of 1.95 Hz width (0.98–32.23 Hz), giving 21 features per axis and a 63-dimensional feature vector.

2.5. Classifier Architecture

Let $\mathbf{x} \in \mathbb{R}^{63}$ be the feature vector of one window. The classifier is a fully connected network with two hidden layers and a SoftMax output, given in Equations (1) and (2):

$$\mathrm{h}^{(1)} = \varphi(\mathrm{W}^{(1)}\mathrm{x} + \mathrm{b}^{(1)}),\ \mathrm{h}^{(2)} = \varphi(\mathrm{W}^{(2)}\mathrm{h}^{(1)} + \mathrm{b}^{(2)}), \quad (1)$$

$$\hat{y}_c = \exp(z_c)/\textstyle\sum_{j=1}^{3} \exp(z_j),\ \mathrm{z} = \mathrm{W}^{(3)}\mathrm{h}^{(2)} + \mathrm{b}^{(3)}, \quad (2)$$

where $\varphi(\cdot) = \max(0, \cdot)$ is the ReLU activation, $\mathrm{W}^{(1)} \in \mathbb{R}^{20\times63}$, $\mathrm{W}^{(2)} \in \mathbb{R}^{10\times20}$ and $\mathrm{W}^{(3)} \in \mathbb{R}^{3\times10}$, and $\hat{y}_c$ is the posterior probability of class c, giving 1523 trainable parameters in total. Training minimizes the categorical cross-entropy with an L1 penalty on the hidden activations, seen in Equation (3):

$$L = -(1/N)\textstyle\sum_{n=1}^{N}\sum_{c=1}^{3} y_{n,c}\log\hat{y}_{n,c} + \lambda\sum_{l=1}^{2}\|\mathrm{h}^{(l)}\|_1, \quad (3)$$

with $\lambda = 1 \times 10^{-5}$. The model was trained for 100 epochs using Adam with a learning rate of 1×10^{-3} and a batch size of 32. The weights with the lowest validation loss were retained. The model was then quantized to INT8 using per-channel scales for the weight tensors and per-tensor scales for the activations [16], so that inference runs entirely in integer arithmetic on the Cortex-M4F.

2.6. Anomaly Detection

Because a three-class SoftMax always returns one of its classes, an unseen motion would be forced into *drinking*, *picked* or *not_drinking*. A GMM anomaly-detection block [17]

therefore runs in parallel on the same features. Let $\mathrm{u} \in \mathbb{R}^4$ be the sub-vector of the four y-axis spectral-power features selected by a feature-importance ranking on the training set, all y-axis spectral power bins (16.6–18.55, 22.46–24.41, 28.32–30.27 and 30.27–32.23 Hz). Three Gaussian components are fitted to the training windows by expectation–maximization, seen in Equation (4):

$$p(\mathrm{u}) = \sum_{\mathrm{k}=1}^{3} \pi_{\mathrm{k}} \, \mathcal{N}(\mathrm{u}|\mu_{\mathrm{k}}, \Sigma_{\mathrm{k}}), \tag{4}$$

where π_{k} are the mixing weights, μ_{k} the means and Σ_{k} the covariances. The anomaly score decreases monotonically with $\log p(\mathrm{u})$, so a window scoring above a threshold set from the training distribution is excluded from the event logic. These four bins carry the high-frequency vibration content on the axis perpendicular to the bottle, which is where dropping, shaking and transport differ most from the smooth tilt of a drink; restricting the detector to four dimensions also keeps the covariance estimates well-conditioned for a training set of this size.

2.7. Event Detection and Fluid Intake Estimation

Window-level outputs are converted into events on the device. A drinking event is opened when consecutive windows are classified as *drinking* with an anomaly score below threshold and is closed when the label returns to *picked* or *not_drinking*; a short debounce suppresses single-window flicker at the boundaries. The event duration d_{i} is the number of contributing windows multiplied by the 0.5 s stride, and fluid intake per session is estimated by Equation (5):

$$V = r \sum_{\mathrm{i}} d_{\mathrm{i}}, \tag{5}$$

where r is a per-user sip rate in mL/s, obtained once per athlete by weighing the bottle before and after a short series of drinks while the node records the event durations and stores them on the gateway. Each closed event is sent as a BLE GATT notification carrying its start timestamp, duration and estimated volume; because the node transmits only on events, the radio is idle for most of a session.

3. Results

3.1. Feature Space

Figure 4 shows the training windows projected into two dimensions of the feature space. The classes form distinct clusters: *not_drinking* a compact region, *picked* an elongated one, and *drinking* the largest, because it contains the tilt, hold and return phases. The only overlap is between the tail of the *drinking* cluster and the start of the *picked* cluster, which corresponds to the lift phase common to both activities.

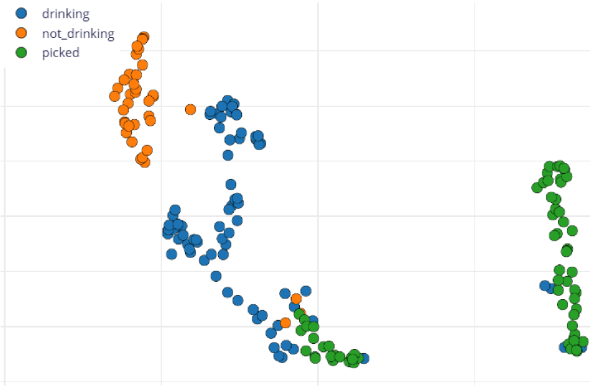
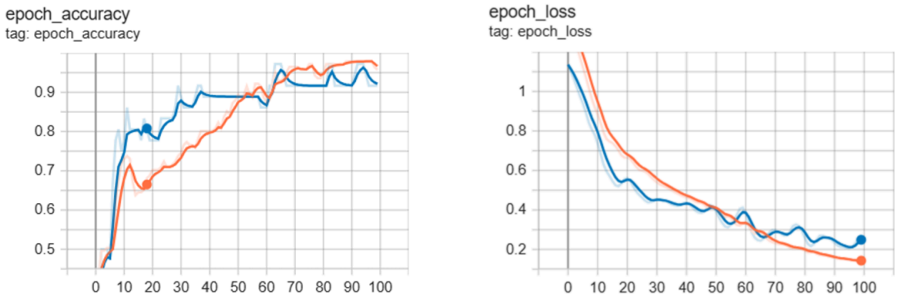


Figure 4. Two-dimensional projection of the training windows after spectral feature extraction.

3.2. Training

Figure 5 shows the training graphs. Validation accuracy rises steeply over the first 20 epochs and settles between 0.92 and 0.97 after epoch 60, while training accuracy climbs slowly to about 0.97. Training loss falls monotonically to 0.14, whereas validation loss flattens at roughly 0.25 from epoch 70 onwards. The small, stable gap between the curves, with no rising validation loss, indicates that 100 epochs was an adequate budget and that the model is not over-fitting.



(**a**) (**b**)

Figure 5. Training graphs over 100 epochs: (**a**) accuracy; (**b**) loss. Orange: training; blue: validation.

3.3. Classification Performance

Figure 6 gives the row-normalized confusion matrices of the INT8 model. On the validation set the model reaches 97.2% accuracy with a loss of 0.21, its only confusion being 6.3% of *drinking* windows labelled *picked*, and per-class F1 scores of 0.97, 1.00 and 0.97. On the held-out test set the accuracy is 96.6%, with 2.9% of *drinking* windows labelled *picked* and 8.3% of *picked* windows labelled *not_drinking*, giving per-class F1 scores of 0.99, 0.96 and 0.92. Both confusions are between physically adjacent states, lifting the bottle to drink and setting a picked bottle down, which is the overlap visible in Figure 4.



(**a**) (**b**)

Figure 6. Confusion matrices of the INT8 model, row-normalized to percentages of the true class: (**a**) validation set; (**b**) held-out test set.

Table 1 summarizes the aggregate metrics. Quantization did not degrade performance: on the test set, the INT8 and float32 models produce identical confusion matrices, and on the validation set the INT8 model is slightly better (97.2% against 94.4%). The area under the ROC curve is 0.98 on both sets.

https://doi.org/10.3390/engproc2026163071

Table 1. Aggregate metrics of the float32 and INT8 models.

Metric	Validation float32	Validation INT8	Test float32	Test INT8
Accuracy	94.4%	97.2%	96.6%	96.6%
Macro F1	0.958	0.979	0.954	0.954
ROC AUC	0.985	0.984	0.984	0.985
Loss	0.204	0.210	0.202	0.200

3.4. Anomaly Detection Results

Figure 7 shows the fitted GMM on two of the four selected axes. The training windows lie inside the shaded one-, two- and three-standard-deviation contours, with a dense cluster at high spectral power (handling motion) and a broader cluster at low power (rest and slow tilt). Windows from motions not present in the training set fall outside the outer contour, score high, and are dropped from the event logic.



Figure 7. Training windows and the fitted GMM components on the *y*-axis spectral power bins 16.6–18.55 Hz and 22.46–24.41 Hz.

3.5. On-Device Deployment

The trained pipeline was compiled into an Arduino library with the INT8 model and flashed to the Nicla Sense ME (Table 2). The complete pipeline executes in under 5 ms per window, two orders of magnitude below the 500 ms stride, so the processor sleeps for most of each period and the node runs continuously through a training session on the LiPo battery. The node logged every drinking event, and the gateway produced a per-session hydration log for each participant.

Table 2. Deployment resources on the Arduino Nicla Sense ME.

Quantity	Value
Sampling rate/window/stride	62.5 Hz/1000 ms/500 ms
Model	63–20–10–3 dense, 1523 parameters, INT8
Inference latency (DSP + NN + GMM)	<5 ms
Peak RAM	5 kB
Flash	100 kB
Radio	BLE GATT notification per event

4. Discussion

The results confirm that a single three-axis accelerometer on the bottle is enough to recognize drinking with high accuracy and that recognition survives INT8 quantization without loss. The closest prior bottle-mounted system, by Griffith et al., used hand-engineered inclination features and support vector regression on 1908 drinks from 84 people to estimate volume directly from motion [4]; our approach separates the problems, recognizing events on the device and leaving volume to a per-user rate calibration, which keeps the model small at the cost of assuming a roughly constant sip rate within a session.

Processing on the node also means that no motion trace of the athlete is stored or transmitted and that the module can be moved between bottles, which fits the preventive-health aim of SDG 3 [18]. The main limitation is the size of the dataset, i.e., recordings from five participants and a single bottle, enough to show that the pipeline works but not to bound performance across athletes, bottle shapes or fill levels, and fill level in particular changes the dynamics of a drink [4]. The anomaly detector was also evaluated only on training-distribution data, and the fluid-intake estimate has not yet been validated against weighed intake.

5. Conclusions

A bottle-mounted TinyML sensor node based on the Arduino Nicla Sense ME was built to recognize drinking events and estimate fluid intake in sport. Using 63 spectral features from a three-axis accelerometer at 62.5 Hz, a 63–20–10–3 INT8 neural network classifies *drinking*, *picked* and *not_drinking* windows with 96.6% accuracy (macro F1 0.954, ROC AUC 0.98) on a held-out test set, a parallel GMM detector rejects motions outside these classes, and the whole pipeline runs in under 5 ms per window on the device. Future work will extend the dataset to more participants, bottle types and fill levels, validate the volume estimate against weighed intake, and add personalized hydration targets with real-time alerts.

Funding: This research received no external funding.

Institutional Review Board Statement: Ethical review and approval were waived for this study because the recordings consisted of non-identifiable bottle motion data collected from consenting volunteers.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The labelled dataset, the processing and model configuration and the exported model are available from the corresponding author on reasonable request.

Conflicts of Interest: The author declares no conflicts of interest.

References

1. Francisco, R.; Armstrong, L.E.; Silva, A.M. Recommendations for Optimizing Research Regarding the Effects of Dehydration on Athletic Performance. *Sports Med*. **2026**, *56*, 23–34. [CrossRef] [PubMed]
2. McCartney, D.; Desbrow, B.; Irwin, C. The Effect of Fluid Intake Following Dehydration on Subsequent Athletic and Cognitive Performance: A Systematic Review and Meta-Analysis. *Sports Med. Open* **2017**, *3*, 13. [CrossRef] [PubMed]
3. Liu, M.; Bian, S.; Zhou, B.; Grünerbl, A.; Lukowicz, P. Smart Cup: An Impedance Sensing Based Fluid Intake Monitoring System for Beverages Classification and Freshness Detection. In Proceedings of the 2022 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp/ISWC '22 Adjunct), Cambridge, UK, 11–15 September 2022. [CrossRef]
4. Griffith, H.; Shi, Y.; Biswas, S. A Container-Attachable Inertial Sensor for Real-Time Hydration Tracking. *Sensors* **2019**, *19*, 4008. [CrossRef] [PubMed]
5. Gómez-Carmona, O.; Casado-Mansilla, D.; López-de-Ipiña, D.; García-Zubia, J. Human-in-the-Loop Machine Learning: Reconceptualizing the Role of the User in Interactive Approaches. *Internet Things* **2024**, *25*, 101048. [CrossRef]
6. Bulling, A.; Blanke, U.; Schiele, B. A Tutorial on Human Activity Recognition Using Body-Worn Inertial Sensors. *ACM Comput. Surv.* **2014**, *46*, 33. [CrossRef]

7. Zhang, S.; Li, Y.; Zhang, S.; Shahabi, F.; Xia, S.; Deng, Y.; Alshurafa, N. Deep Learning in Human Activity Recognition with Wearable Sensors: A Review on Advances. *Sensors* **2022**, *22*, 1476. [CrossRef] [PubMed]
8. Wang, C.; Kumar, T.S.; De Raedt, W.; Camps, G.; Hallez, H.; Vanrumste, B. Eating Speed Measurement Using Wrist-Worn IMU Sensors Towards Free-Living Environments. *IEEE J. Biomed. Health Inform.* **2024**, *28*, 5816–5828. [CrossRef] [PubMed]
9. Gohl, P.D.; Spellen, A.N.; Queiroz, L.I.; Souto, E.J. InGesture: An Eight-Class Inertial Sensor Dataset for Fluid Intake and Hand-Gesture Recognition. *Data Brief* **2026**, *66*, 112761. [CrossRef] [PubMed]
10. Heydari, S.; Mahmoud, Q.H. Tiny Machine Learning and On-Device Inference: A Survey of Applications, Challenges, and Future Directions. *Sensors* **2025**, *25*, 3191. [CrossRef] [PubMed]
11. Diab, M.S.; Rodriguez-Villegas, E. Embedded Machine Learning Using Microcontrollers in Wearable and Ambulatory Systems for Health and Care Applications: A Review. *IEEE Access* **2022**, *10*, 98450–98474. [CrossRef]
12. Capogrosso, L.; Cunico, F.; Cheng, D.S.; Fummi, F.; Cristani, M. A Machine Learning-Oriented Survey on Tiny Machine Learning. *IEEE Access* **2024**, *12*, 23406–23426. [CrossRef]
13. Huang, Z.; Yu, L.; Herbozo Contreras, L.F.; Dwivedi, G.; Kavehei, O. On-Device Machine Learning for Secure, Generalizable and Real-Time Medical Data Processing: On the Emergence of μ-Trainers. *R. Soc. Open Sci.* **2025**, *12*, 251213. [CrossRef]
14. Arduino. Nicla Sense ME Documentation. Available online: https://docs.arduino.cc/hardware/nicla-sense-me/ (accessed on 8 September 2026).
15. Bosch Sensortec. BHI260AP Ultra-Low-Power Smart Sensor with Integrated IMU: Datasheet. Available online: https://www.bosch-sensortec.com/media/boschsensortec/downloads/datasheets/bst-bhi260ap-ds000.pdf (accessed on 8 September 2026).
16. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 18–23 June 2018; pp. 2704–2713. [CrossRef]
17. Reynolds, D. Gaussian Mixture Models. In *Encyclopedia of Biometrics*; Li, S.Z., Jain, A., Eds.; Springer: Boston, MA, USA, 2009; pp. 659–663. [CrossRef]
18. United Nations. Sustainable Development Goal 3: Ensure Healthy Lives and Promote Well-Being for All at All Ages. Available online: https://sdgs.un.org/goals/goal3 (accessed on 8 September 2026).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.