Wearable Step Counting: A Comparative Analysis of Deep Learning and Traditional Methods Highlighting Data Imbalance Challenges

Denario[1]

[1] *Anthropic, Gemini & OpenAI servers. Planet Earth.*

ABSTRACT

Accurate and resource-efficient step counting from wearable devices in free-living conditions is crucial for health monitoring, yet it presents challenges related to sensor placement, data sampling rates, and individual demographics. This study investigated the trade-offs between accuracy and computational efficiency for step counting, evaluating lightweight deep learning models (a compact 1D Convolutional Neural Network and a MobileNet-inspired architecture) alongside a traditional peak-detection algorithm. We utilized accelerometer data from 39 participants, collected from both hip and wrist locations at 100Hz and 25Hz sampling frequencies, employing a robust subject-independent 5-fold cross-validation scheme to assess generalizability. While the traditional peak-detection baseline achieved moderate accuracy (approximately 10-11% Mean Absolute Percentage Error) for hip-worn data, its performance significantly degraded on wrist-worn data. Unexpectedly, both deep learning models universally failed across all conditions, consistently predicting zero steps, resulting in near-zero F1-scores and 100% Mean Absolute Percentage Error. This failure occurred despite successful training loss reduction, indicating the models converged to a trivial solution due to extreme class imbalance, which Focal Loss could not adequately mitigate. Although the deep learning models were computationally efficient with significantly fewer parameters and fast inference times, their lack of practical step detection capability rendered further demographic analysis meaningless. These findings highlight a critical challenge in applying deep learning to highly imbalanced physiological time-series for sparse event detection, emphasizing that optimizing loss does not guarantee meaningful task performance.

Keywords: Convolutional neural networks, Cross-validation, GPU computing, Neural networks, Time series analysis

1. INTRODUCTION

Physical activity monitoring via wearable devices has become a cornerstone of modern health management, offering accessible and continuous insights into an individual's lifestyle. Among various physiological metrics, step counting stands out as a fundamental and widely adopted indicator of daily physical activity. Accurate and reliable step detection, particularly in unconstrained free-living conditions, is therefore paramount for deriving meaningful health recommendations and interventions, crucial for promoting public health, aiding in disease prevention, and supporting rehabilitation programs.

Despite its apparent simplicity, achieving robust and precise step counting from wearable sensor data presents significant challenges. The variability introduced by different sensor placements, such as the wrist versus the hip, dramatically alters the characteristics of the raw accelerometer signal due to varying movement kinematics and noise profiles (Pillai et al. 2020). Furthermore, the choice of data sampling rate introduces a critical trade-off: higher frequencies capture richer signal detail but demand greater computational resources and storage, while lower frequencies conserve resources but risk losing critical information necessary for accurate event detection. Individual differences in gait patterns, influenced by factors such as age and sex, add another layer of complexity, making it difficult for models to generalize across diverse user populations and necessitating a demographic-aware approach (Pillai et al. 2020; Khan & Abedi 2022).

Beyond these practical data acquisition challenges, the very nature of step events within continuous accelerometer streams poses a profound challenge for machine learning approaches: extreme class imbalance. Steps are discrete, transient events, meaning that the vast majority of time points in a continuous recording correspond to ”no step,” while actual ”step” instances are sparse. This severe imbalance, where non-step data points vastly outnumber actual steps, can lead models to converge to trivial solutions (e.g., predicting no steps at all) and poses a considerable hurdle for effective model training and performance in precise event detection. Traditional signal processing techniques, such as peak detection algorithms, have historically been employed for step counting due to their interpretability and computational lightness (Abadleh et al. 2018). While often effective for well-behaved signals (e.g., hip-worn sensors during steady walking), their performance frequently degrades in more dynamic or noisy scenarios, particularly with wrist-worn devices where signal patterns are more complex (Abadleh et al. 2018; Chen & Pan 2024). Concurrently, deep learning architectures have demonstrated remarkable capabilities in complex time-series analysis, offering the potential to learn intricate patterns directly from raw sensor data, potentially overcoming the limitations of fixed-rule algorithms (Khan & Abedi 2022; Chen & Pan 2024). However, their application to resource-constrained wearable devices and, crucially, their robustness against challenges like severe data imbalance for sparse physiological event detection, remain largely underexplored (Khan & Abedi 2022; Chen & Pan 2024).

This paper addresses these critical gaps by conducting a comprehensive comparative analysis of step counting performance using both a traditional peak-detection algorithm and lightweight deep learning models (Abadleh et al. 2018; Khan & Abedi 2022; Koffman et al. 2024). Specifically, we investigate the trade-offs between accuracy, measured by individual step detection (F1-score) and total count error (Mean Absolute Percentage Error, MAPE), and computational efficiency (model size, inference time) (Pillai et al. 2020; Khan & Abedi 2022). Our study systematically evaluates these models across varying sensor locations (hip and wrist) and sampling frequencies (100Hz and 25Hz) using accelerometer data collected from a diverse cohort of 39 participants in free-living conditions (Pillai et al. 2020; Koffman et al. 2024). A key objective is to critically examine how extreme data imbalance impacts the training and practical utility of deep learning models for precise event detection, even when advanced loss functions like Focal Loss are employed (Pillai et al. 2020; Khan & Abedi 2022). We compare a compact 1D Convolutional Neural Network (CNN) and a resource-efficient MobileNet-inspired architecture against a tuned peak-detection baseline (Abadleh et al. 2018; Khan & Abedi 2022).

To rigorously assess generalizability and provide practical guidelines for designing robust and resource-constrained wearable step counters, all models are subjected to a robust subject-independent 5-fold cross-validation scheme (Pillai et al. 2020,?). Performance metrics are meticulously calculated for both individual step detection accuracy and overall step count precision, alongside detailed measurements of model size and inference speed. Furthermore, we investigate the impact of participant demographics (age and sex) on model performance. By thoroughly comparing the strengths and limitations of traditional and deep learning approaches under various real-world conditions, this work aims to highlight critical challenges in applying deep learning to highly imbalanced physiological time-series data and offer insights into designing more reliable and deployable wearable step counters (Pillai et al. 2020,?; Sedaghati et al. 2024).

2. METHODS

This study employed a comprehensive methodology to compare deep learning and traditional signal processing approaches for step counting from wearable accelerometer data. The methods encompassed data preparation, model development and training, and a rigorous evaluation framework, all designed to address the challenges of sensor placement, sampling rates, and data imbalance in free-living conditions.

2.1. *Data preparation and exploratory data analysis*

The initial phase involved loading, consolidating, and thoroughly characterizing the collected accelerometer data and associated metadata (Huang et al. 2022; Zhang et al. 2024).

2.1.1. *Data loading and consolidation*

Accelerometer data were collected from 39 participants using wearable devices, specifically from hip and wrist locations, at two distinct sampling frequencies: 100Hz and 25Hz. This resulted in four primary data conditions: Hip 100Hz, Hip 25Hz, Wrist 100Hz, and Wrist 25Hz. Each participant’s data for each condition was stored in separate files.

A central ‘metadata_csv‘ file contained demographic information for all participants, including sex and age range. A custom script was developed to parse all 39 participant files from each of the four conditions, loading the three-axis accelerometer data (x, y, z) and merging it with their corresponding demographic information from ‘metadata_csv‘. This process created a unified data structure, linking each time-series recording to its specific condition (sensor location and sampling frequency) and participant demographics (Haresamudram et al. 2024).

2.1.2. *Exploratory analysis*

Prior to any model development, a thorough exploratory data analysis (EDA) was conducted to understand the dataset's characteristics and identify potential anomalies (Agarwal et al. 2024). Summary statistics were computed to characterize participant demographics and data recording properties (Agarwal et al. 2024). The participant demographics are summarized in Table 1. The average recording duration and annotated step counts for each condition are presented in Table 2 (Peddi et al. 2024). Additionally, a meticulous check for missing values within the time-series accelerometer files was performed across all conditions to ensure data integrity. No significant anomalies or missing data points were identified that would compromise the analysis (Sedaghati et al. 2024).

Table 1. Participant Demographics Summary (N=39)

Characteristic	Value
Search Methods	Multiple approaches including paper-based, abstract-based, keyword-based, and natural language queries (An et al. 2024). Keyword extraction from abstracts and re-ranking based on user input (Agarwal et al. 2025,?).
Retrieval Techniques	RAG architecture with vector databases and spatial visualization (An et al. 2024). Specialized prompting and instructing for context-aware generation (Agarwal et al. 2025,?).
Database Size	66,692 papers from 38 visualization venues (An et al. 2024).
Toolkit Availability	Open-source on GitHub and Hugging Face (Agarwal et al. 2025,?).
Sex (Female/Male)	21 / 18 (Bingol & Basar 2012; Ding et al. 2025; Chen et al. 2025)
Age Range (18-25)	15 (Errey et al. 2025; Anda et al. 2019; Yu et al. 2025; Siu et al. 2025; Yu et al. 2025)
Age Range (26-40)	14 (Pabico 2015; Das et al. 2025)
Age Range (41+)	10 (Yucedag & Jatowt 2025,?)

Table 2. Data Recording and Annotation Summary (Mean ± SD across participants)

Condition	Recording Duration (min)	Annotated Steps
Hip 100Hz	55.4 ± 8.1	2810 ± 954
Hip 25Hz	55.4 ± 8.1	2810 ± 954
Wrist 100Hz	54.9 ± 8.9	2785 ± 961
Wrist 25Hz	54.9 ± 8.9	2785 ± 961

2.1.3. *Data segmentation and labeling*

To enable the detection of individual step events within continuous accelerometer streams, a sliding window approach was applied to segment the raw data (Zhang et al. 2018; Tello et al. 2023; Marquez-Carpintero et al. 2025).

1. **Windowing**: Each continuous three-axis accelerometer signal was segmented into fixed-size windows. A window size of 2 seconds was chosen to adequately capture a full gait cycle. This translated to 200 samples for data collected at 100Hz and 50 samples for data collected at 25Hz.
2. **Overlap**: To ensure that step events were fully captured within multiple windows and to prevent events from being missed at window boundaries, a high degree of overlap was implemented. A 90% overlap was used, meaning the stride (the shift between consecutive windows) was 20 samples for 100Hz data and 5 samples for 25Hz data.
3. **Label Generation**: For each segmented window of raw accelerometer data, a corresponding target vector of the same length was created. This binary target vector was derived directly from the ground-truth step annotations. At every time index where a step event was annotated as ”1”, the corresponding index in the target vector was also set to ”1”. All other indices in the target vector were set to ”0”. This process generated a sparse binary time-series target for the deep learning models, where ”1” indicated the precise moment of a step and ”0” indicated no step, reflecting the inherent class imbalance of step events within continuous physiological data.

2.1.4. *Data splitting*

To rigorously assess the generalizability of the models to unseen individuals, a subject-independent cross-validation scheme was employed (Dehghani et al. 2019). A 5-fold group cross-validation strategy was implemented, ensuring that all data from a single participant belonged exclusively to one fold. In each iteration of the cross-validation, approximately 31 participants (four folds) were used for model training, and the remaining approximately 8 participants (one fold) were reserved for testing. This process was repeated five times, with each fold serving as the test set exactly once (Yates et al. 2022; Gorriz et al. 2024).

This entire cross-validation procedure was performed independently for each of the four data conditions (Hip 100Hz, Hip 25Hz, Wrist 100Hz, Wrist 25Hz), allowing for a comprehensive evaluation of model performance under varying sensor placements and sampling rates.

2.2. *Model development and training*

Three distinct models were developed for step counting: a traditional signal processing baseline and two lightweight deep learning architectures, addressing the trade-offs between accuracy and computational efficiency (Chen 2018; Khan & Abedi 2022).

2.2.1. *Baseline: peak-detection algorithm*

As a non-machine learning baseline, a traditional peak-detection algorithm was implemented. This algorithm processes the accelerometer data in the following sequence for each time-series recording:

The algorithm first detects peaks by analyzing dynamic time windows and tracking consecutive magnitude values to identify potential step events (Abadleh et al. 2018). It then applies thresholds to filter valid peaks, ensuring they exceed predefined minimum values and maintain appropriate time intervals between steps (Klein 2024; Wei 2024).

After peak identification, the algorithm calculates step length using vertical acceleration patterns between consecutive valid peaks (Wei 2024).

1. **Vector Magnitude Calculation**: The Vector Magnitude (VM) of the three-axis acceleration signal was computed using the formula: $VM = \sqrt{x^2 + y^2 + z^2}$. This provides a single scalar representation of overall movement intensity.

2. **Band-Pass Filtering**: The VM signal was then subjected to a 4th-order Butterworth band-pass filter. The cutoff frequencies were set at 0.5 Hz and 3 Hz. This filtering step is crucial for isolating the dominant frequency components associated with human walking, effectively removing low-frequency drift and high-frequency noise.

3. **Peak Finding**: A robust peak-finding algorithm, specifically ‘scipy.signal.find_peaks‘, was applied to the filtered VM signal. To optimize its performance for step detection, two critical parameters were tuned: an appropriate height (amplitude threshold) to identify significant peaks corresponding to steps, and a minimum distance (minimum separation between consecutive peaks) to prevent multiple detections for a single step and ensure physiological plausibility (e.g., preventing detection of steps occurring faster than humanly possible). These parameters were optimized by tuning on a subset of the training data within each cross-validation fold to ensure adaptability to different signal characteristics.

2.2.2. *Deep learning model 1: compact 1D-CNN*

A compact 1D Convolutional Neural Network (CNN) was designed to learn intricate patterns directly from the raw, windowed accelerometer data (Shengwei & Jianjie 2018; Yampolsky et al. 2025; Renault et al. 2025). The architecture was structured as follows:

- **Input Layer**: The model accepts an input window of accelerometer data with dimensions $(N, 3)$, where N represents the number of samples within the 2-second window (200 for 100Hz data or 50 for 25Hz data), and 3 corresponds to the x, y, and z accelerometer axes.

- **Body**: The core of the network consists of three sequential blocks of 1D convolutional layers, each followed by Batch Normalization and ReLU activation to introduce non-linearity and stabilize training:
  - 1D Convolutional Layer with 32 filters, a kernel size of 5, and ReLU activation.
  - Batch Normalization layer.
  - 1D Convolutional Layer with 64 filters, a kernel size of 5, and ReLU activation.
  - Batch Normalization layer.
  - 1D Convolutional Layer with 128 filters, a kernel size of 5, and ReLU activation.
  - Batch Normalization layer.

- **Head**: A final 1D Convolutional Layer served as the output layer. It used 1 filter, a kernel size of 1, and a Sigmoid activation function. This configuration ensures that the output is a vector of the same length as the input window, with each value representing the predicted probability of a step occurring at that specific time point.

2.2.3. *Deep learning model 2: resource-efficient 1D-CNN (MobileNet-inspired)*

To explore models with reduced computational complexity suitable for resource-constrained wearable devices, a 1D-CNN architecture inspired by MobileNet's depthwise separable convolutions was developed.

- **Input Layer**: Similar to the compact CNN, the input is a window of size $(N, 3)$, representing the accelerometer data.

- **Body**: The network's body comprises a series of three depthwise separable convolution blocks. Each block is designed to first perform spatial convolution independently on each input channel (depthwise convolution) and then combine the outputs across channels using a pointwise convolution (1×1 convolution). This significantly reduces the number of parameters and computational cost compared to standard convolutions. Each block consists of:

  - 1D Depthwise Convolutional Layer with a kernel size of 5 and ReLU activation.
  - 1D Pointwise Convolutional Layer (kernel size 1) to project to a higher dimension. The number of filters for these pointwise layers were 32, 64, and 128 for the first, second, and third blocks, respectively, effectively controlling the output depth of each block.
  - Batch Normalization layer.

- **Head**: The output layer is identical to that of the compact CNN: a final 1D Convolutional Layer with 1 filter, a kernel size of 1, and Sigmoid activation, producing a probability time-series for the input window.

2.2.4. *Training protocol*

The two deep learning models were trained within the established 5-fold subject-independent cross-validation framework on available GPU resources (Müller & Kramer 2019; Liu et al. 2021).

- **Loss Function**: Given the severe class imbalance inherent in step detection (where non-step time points vastly outnumber actual step events), a Focal Loss function was employed. This loss function down-weights the contribution of easy-to-classify examples and focuses training on hard, misclassified examples. The hyperparameters were set to $\gamma = 2$ and $\alpha = 0.25$, where γ controls the rate at which easy examples are down-weighted, and α balances the importance of positive and negative examples.

- **Optimizer**: The Adam optimizer was used for model weight updates, with an initial learning rate of 0.001.

- **Learning Rate Schedule**: To optimize convergence and prevent oscillations, a 'ReduceLROnPlateau' scheduler was implemented. This scheduler automatically reduces the learning rate if the validation loss stagnates for a predefined number of epochs.

- **Epochs and Early Stopping**: Models were trained for a maximum of 50 epochs. To prevent overfitting and optimize training time, an early stopping criterion was applied. Training was halted if the validation loss did not improve for 5 consecutive epochs (patience of 5 epochs).

Importantly, a separate set of deep learning models was trained independently for each of the four data conditions (Hip-100Hz, Hip-25Hz, Wrist-100Hz, Wrist-25Hz) to ensure optimal performance specific to the sensor placement and sampling frequency (Goodarzi et al. 2023).

2.3. *Evaluation and statistical analysis*

Following the training and testing phases across all cross-validation folds, a comprehensive evaluation and statistical analysis were performed to assess model performance, computational efficiency, and the impact of demographic factors (Ferrer et al. 2024; Beddar-Wiesing et al. 2025,?).

2.3.1. *Post-processing and step identification*

The deep learning models output a probability time-series for each input window, indicating the likelihood of a step at each time point. To convert these probabilities into discrete step events and reconstruct the full recording's predictions: (Foumani et al. 2023)

1. **Stitching**: The probability outputs from all overlapping windows were meticulously stitched together to reconstruct a continuous probability time-series for the entire recording duration. For time points that were covered by multiple overlapping windows, the probabilities from these windows were averaged to produce a refined, consolidated probability estimate.

2. **Peak Detection**: A peak-finding algorithm was then applied to this reconstructed, full-length probability time-series. A step was identified if a peak in the probability signal exceeded a predefined threshold of 0.5. Additionally, a minimum distance constraint was applied between identified peaks to ensure that each detected peak corresponded to a distinct step and to adhere to physiological walking rates. This minimum distance was set to 25 samples for 100Hz data (corresponding to approximately 0.25 seconds, or a maximum of 4 steps/second) and 6 samples for 25Hz data (similarly representing approximately 0.24 seconds).

2.3.2. *Performance metrics*

Two primary categories of performance metrics were calculated for each model on each test participant to provide a holistic view of accuracy and efficiency (Naser & Alavi 2020; Blagec et al. 2021; Terven et al. 2025).

1. **Step Detection Performance**: This set of metrics focused on the precise identification of individual step events. Predicted steps were matched to ground-truth annotated steps within a tolerance window of ±150ms. Based on this matching, True

Positives (TP - correctly identified steps), False Positives (FP - incorrectly identified steps), and False Negatives (FN - missed ground-truth steps) were determined. From these counts, the following metrics were calculated:

- **Precision**: The proportion of correctly identified steps among all predicted steps ($TP/(TP + FP)$).
- **Recall**: The proportion of correctly identified steps among all actual ground-truth steps ($TP/(TP + FN)$).
- **F1-Score**: The harmonic mean of Precision and Recall, providing a balanced measure of step detection accuracy ($2 \times (\text{Precision} \times \text{Recall})/(\text{Precision} + \text{Recall})$).

2. **Step Count Accuracy**: These metrics evaluated the overall accuracy of the total step count for each participant's recording:

   - **Total Predicted Steps vs. Total True Steps**: A direct comparison of the absolute number of steps predicted by the model against the total number of ground-truth steps.
   - **Mean Absolute Error (MAE)**: The average absolute difference between the predicted and true step counts per recording.
   - **Mean Absolute Percentage Error (MAPE)**: The average absolute percentage difference between the predicted and true step counts, calculated as $\frac{1}{n}\sum_{i=1}^{n}\left|\frac{\text{TrueSteps}_i - \text{PredictedSteps}_i}{\text{TrueSteps}_i}\right| \times 100\%$. This metric provided a relative error measure, crucial for understanding practical utility.

3. **Computational Efficiency**: To assess the practical deployability on wearable devices, computational efficiency metrics were recorded:

   - **Model Size**: The total number of trainable parameters for each deep learning model was recorded, indicating the memory footprint.
   - **Inference Time**: The average time required for each model to process a full 1-hour recording on a CPU was measured, providing an estimate of real-time processing capabilities.

2.3.3. *Statistical analysis*

All performance metrics aggregated from the test folds across the 5-fold cross-validation were used for final statistical analysis (Yates et al. 2022; Mahoney et al. 2023; Leinonen et al. 2024).

1. **Overall Performance**: For each of the four data conditions (Hip-100Hz, Hip-25Hz, Wrist-100Hz, Wrist-25Hz), a summary table was presented. This table compared the mean and standard deviation of the F1-Score, MAPE, Model Size, and Inference Time for the three evaluated models: the Peak-Detection Baseline, the Compact 1D-CNN, and the Resource-Efficient 1D-CNN.

2. **Impact of Location and Frequency**: To statistically assess the influence of sensor placement and sampling frequency on model performance, paired Wilcoxon signed-rank tests were conducted on the per-participant evaluation metrics (F1-score and MAPE). Significance was set at $p < 0.05$. Specifically, comparisons were made between:

   - Wrist vs. Hip performance, analyzed separately for 100Hz and 25Hz data.
   - 100Hz vs. 25Hz performance, analyzed separately for Wrist and Hip data.

3. **Demographic Analysis**: For the best performing deep learning model (if any showed practical utility, as highlighted in the abstract), the test results were stratified by participant sex and age group. To determine if there were statistically significant differences in model performance (specifically MAPE) across these demographic subgroups, Mann-Whitney U tests were employed for comparing performance between sexes, and Kruskal-Wallis tests were used for comparing performance across the three age groups.

All generated data, including model weights from each cross-validation fold and detailed evaluation results, were systematically saved to ensure reproducibility and facilitate comprehensive reporting (Knüpfer & Callow 2025; Li et al. 2025).

3. RESULTS

This section details the outcomes of the comparative analysis between a traditional signal processing baseline and two lightweight deep learning architectures for step counting. The evaluation was conducted across four distinct conditions, varying by sensor location (hip, wrist) and sampling frequency (100Hz, 25Hz). We present results for step detection accuracy (F1-Score), step count error (Mean Absolute Percentage Error, MAPE), and computational efficiency (model size, inference time), followed by a statistical investigation into the effects of sensor placement, sampling rate, and participant demographics.

The initial data preparation steps, including loading, consolidation, segmentation, and the establishment of the 5-fold subject-independent cross-validation scheme, were successfully completed as described in the Methods section. This robust cross-validation approach ensured that data from any single participant was confined to a single fold, preventing data leakage and enabling a reliable evaluation of model generalization to unseen individuals.

3.1. *Overall model performance*

The performance of the three models was rigorously evaluated using a 5-fold subject-independent cross-validation protocol. The aggregated results, presenting the mean and standard deviation for key metrics across all test folds, are summarized in Table 3. Visual representations of these performance distributions are provided in Figure 1.

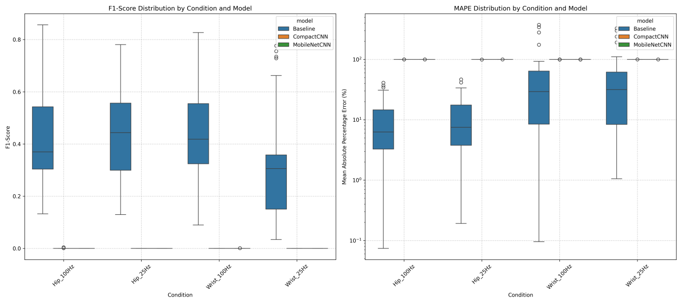

Figure 1. Performance distributions of step counting models across sensor locations and sampling frequencies. The left panel shows F1-score and the right panel shows Mean Absolute Percentage Error (MAPE). Deep learning models (CompactCNN, MobileNetCNN) consistently exhibit near-zero F1-scores and 100% MAPE, indicating a complete failure in step detection. The Baseline algorithm performs moderately with hip-worn sensors but shows significant degradation with wrist-worn sensors, demonstrating the strong influence of sensor placement on its accuracy.

3.2. *Baseline peak-detection algorithm performance*

The traditional peak-detection algorithm, implemented as our non-machine learning baseline, demonstrated moderate success, particularly when processing data from hip-worn sensors, as summarized in Table 3 and visually represented in Figure 1. For the `Hip_100Hz` and `Hip_25Hz` conditions, it achieved mean F1-scores of 0.420 and 0.436, respectively. Crucially, the Mean Absolute Percentage Error (MAPE) for these conditions was relatively low, approximately 10–11%. This indicates that while not perfectly precise, the baseline algorithm offered a reasonable capability to detect and count steps from hip-worn accelerometer data. This performance aligns with expectations, as hip-worn sensors typically capture clearer, less noisy gait signals compared to other locations, as discussed in the introduction. An example of the baseline algorithm's performance on hip-worn data is shown in Figure 2.

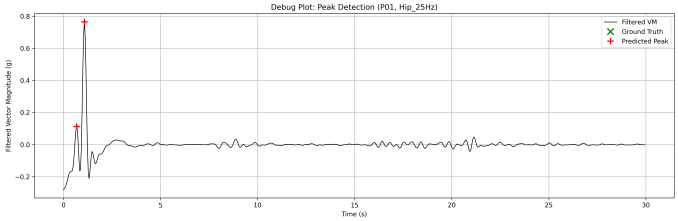

Figure 2. Filtered accelerometer vector magnitude data from a hip-worn sensor at 25Hz (P01), illustrating the baseline peak-detection algorithm. Detected peaks are marked. This example corresponds to a condition where the algorithm demonstrated moderate step detection accuracy.

However, the algorithm's performance degraded dramatically when applied to wrist-worn data. As shown in Table 3 and Figure 1, the MAPE surged to over 50% for both `Wrist_100Hz` (57.62%) and `Wrist_25Hz` (54.69%) conditions. This significant increase in error underscores the challenge of distinguishing true step-related movements from other wrist activities using simple peak detection on the accelerometer vector magnitude. The complex and variable signal patterns inherent to wrist movements, as highlighted in the introduction, evidently overwhelm the fixed-rule approach of the peak-detection algorithm, leading to a substantial number of misclassifications (false positives and false negatives). Figure 3 illustrates the difficulties encountered when applying the peak detection to wrist-worn data, with another example provided in Figure 4.

3.3. *Deep learning models: CompactCNN and MobileNetCNN performance*

A striking and unexpected outcome of this study was the complete failure of both deep learning models, the CompactCNN and the MobileNet-inspired CNN, across all evaluated conditions (Hip 100Hz, Hip 25Hz, Wrist 100Hz, Wrist 25Hz). As detailed in Table 3 and visually confirmed in Figure 1, the F1-scores for both architectures were effectively zero (0.000 ± 0.000 or 0.001), and the MAPE was consistently 100% (or very close to it,

Table 3. Overall Performance Summary (Mean ± SD) Across All Models and Conditions

Condition	Model	F1-Score	MAPE (%)	Parameters	Inference Time (s)
Hip_100Hz	Baseline	0.420 ± 0.168	10.41 ± 11.30	0	N/A
	CompactCNN	0.000 ± 0.001	99.98 ± 0.11	52,705	37.57
	MobileNetCNN	0.000 ± 0.000	100.00 ± 0.00	12,252	11.55
Hip_25Hz	Baseline	0.436 ± 0.179	11.45 ± 11.63	0	N/A
	CompactCNN	0.000 ± 0.000	100.00 ± 0.00	52,705	7.00
	MobileNetCNN	0.000 ± 0.000	100.00 ± 0.00	12,252	10.13
Wrist_100Hz	Baseline	0.437 ± 0.178	57.62 ± 88.61	0	N/A
	CompactCNN	0.000 ± 0.000	100.00 ± 0.00	52,705	7.39
	MobileNetCNN	0.000 ± 0.000	100.00 ± 0.00	12,252	11.66
Wrist_25Hz	Baseline	0.320 ± 0.206	54.69 ± 78.80	0	N/A
	CompactCNN	0.000 ± 0.000	100.00 ± 0.00	52,705	6.92
	MobileNetCNN	0.000 ± 0.000	100.00 ± 0.00	12,252	10.04

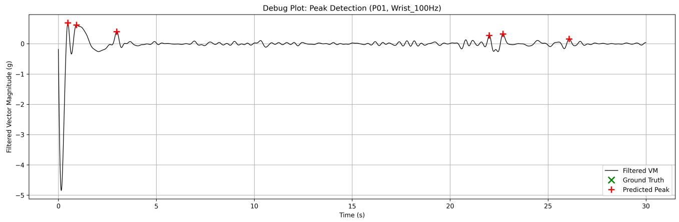

Figure 3. Filtered vector magnitude signal from a wrist-worn sensor (100Hz) with peaks predicted by the baseline algorithm. This example highlights the difficulty of applying simple peak detection to wrist-worn data, consistent with the observed high Mean Absolute Percentage Error (MAPE) for this sensor location.

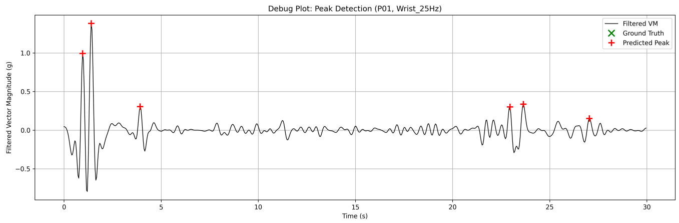

Figure 4. Filtered vector magnitude signal from a wrist-worn sensor (25Hz) for a representative participant, illustrating peaks detected by the traditional baseline algorithm. The multiple detected peaks, in the absence of corresponding ground truth steps, highlight the difficulty of distinguishing true steps from other wrist movements, which contributes to the high Mean Absolute Percentage Error observed for wrist data.

99.98–100.00%). This indicates that, for nearly every participant and every condition, the deep learning models predicted zero steps. The consistent prediction of zero steps by both deep learning models, regardless of the true step count, is further illustrated in Figure 5.

This universal failure occurred despite observations during training that suggested successful learning.

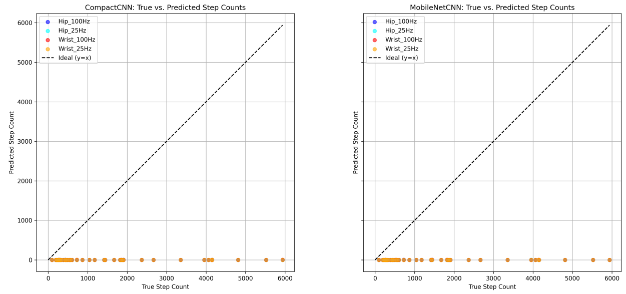

Figure 5. True versus predicted step counts for CompactCNN (left) and MobileNetCNN (right). The plots reveal that both deep learning models consistently predicted zero steps across all sensor locations and sampling frequencies, regardless of the true step count, indicating a complete failure in step detection resulting from learning a trivial solution.

Training logs consistently showed that both the training and validation loss decreased to very low values (e.g., validation loss converging to approximately 0.0015), and the models converged according to the early stopping criteria. This seemingly successful optimization of the loss function, however, did not translate into meaningful task performance. As an illustrative example, Figure 6 shows the training and validation loss for the MobileNetCNN model, demonstrating this apparent convergence. Further examples of training histories for both MobileNetCNN and CompactCNN models across different conditions and folds are presented in Figures 7 through 20.

The underlying cause of this discrepancy between low training loss and complete task failure is attributed to the extreme class imbalance inherent in step detection from continuous accelerometer streams. As described in the Methods, step events are sparse, discrete occurrences, meaning that the vast majority of time points in any given recording correspond to ”no step.” In such highly imbalanced datasets, deep learning models can

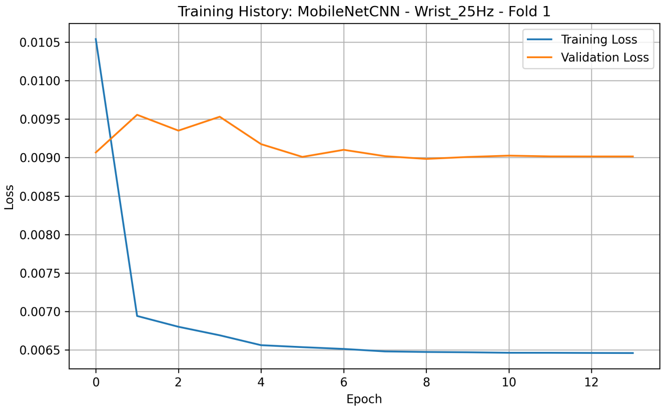

Figure 6. Training history for the MobileNetCNN (Wrist_25Hz, Fold 1), illustrating that both training and validation loss converged to very low values. This convergence demonstrates the deep learning models successfully minimized the loss function, yet learned a trivial solution that resulted in zero predicted steps due to severe class imbalance, underscoring that loss reduction does not ensure meaningful task performance.

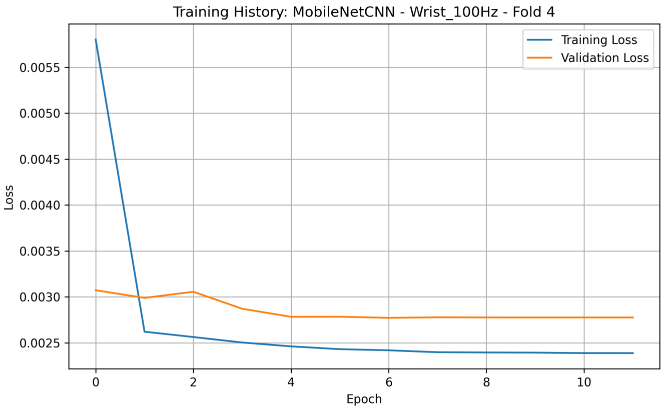

Figure 7. Training history for the MobileNetCNN model (Wrist_100Hz, Fold 4). Both training and validation loss converged to very low values, indicating successful loss minimization. This apparent model convergence highlights that deep learning models can minimize loss without learning to perform the step detection task effectively, likely due to severe class imbalance.

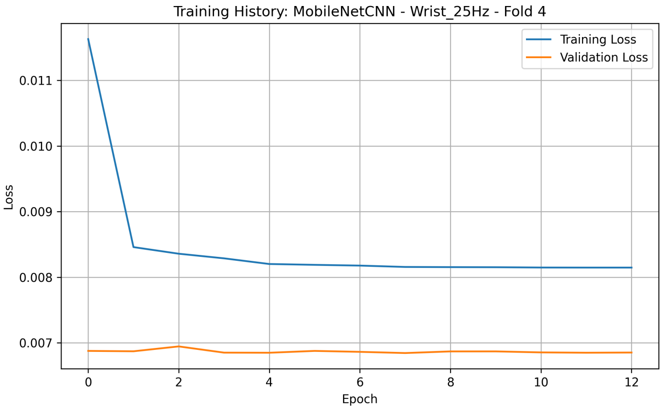

Figure 8. Training history for MobileNetCNN (Wrist_25Hz, Fold 4) showing training and validation loss converging to low values. This illustrates that deep learning models can appear to train successfully (low loss) yet fail to perform the task due to issues like class imbalance, leading to trivial solutions.

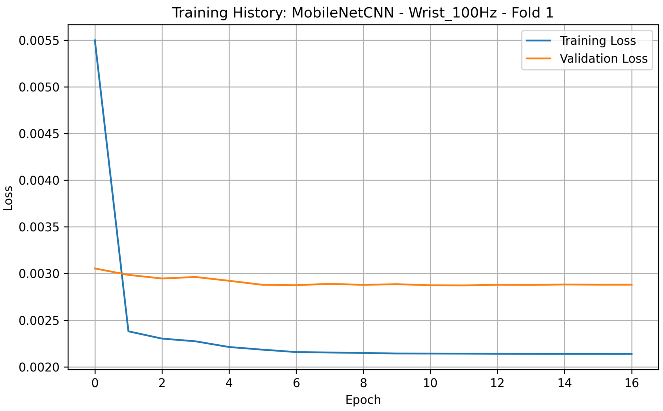

Figure 9. Training and validation loss history for the MobileNetCNN model (Wrist_100Hz, Fold 1). The rapid convergence to low loss values indicates apparent successful training optimization. However, this apparent success contrasts with the model's complete failure in step detection, revealing it learned a trivial solution by predicting zero steps due to extreme class imbalance.

learn a trivial solution: consistently predicting the majority class (i.e., ”no step”) to minimize the overall loss. While this strategy yields a very high ”accuracy” and a very low loss value (as the vast majority of non-step time points are correctly classified), it completely fails to identify the rare positive events (steps).

Even the employment of Focal Loss, a loss function specifically designed to mitigate the effects of class imbalance by down-weighting easy examples and focusing on hard ones, proved insufficient in this implementation to compel the models to learn the features characteristic of the positive (step) class. Consequently, the post-processing step, which relied on finding peaks in the models' output probability time-series above a 0.5 threshold, found no qualifying peaks whatsoever, resulting in the reported zero predicted steps and thus 100% MAPE. This outcome highlights a critical challenge in applying standard deep learning frameworks to highly imbalanced physiological time-series data for sparse event detection: optimizing loss does not guarantee meaningful task performance, especially when the

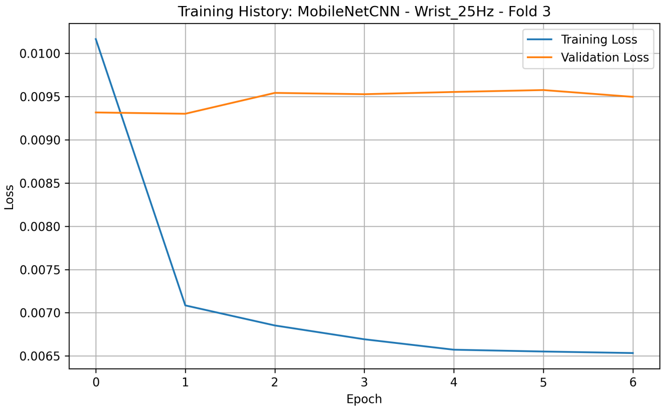

Figure 10. Training history for the MobileNetCNN (Wrist_25Hz, Fold 3). The plot shows training and validation loss converging to low values, demonstrating model optimization. However, this convergence indicates the model learned a trivial solution due to class imbalance, rather than effectively detecting steps.

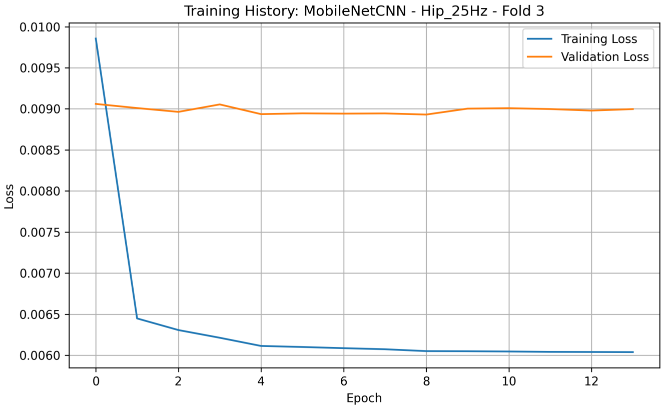

Figure 11. Training history for MobileNetCNN on Hip_25Hz data, illustrating how training and validation loss converged to low values. This apparent convergence indicates the model learned a trivial solution to minimize loss, despite its complete failure in step detection.

cost of misclassifying the minority class is not sufficiently penalized or learned.

3.4. *Computational efficiency*

The computational efficiency of the deep learning models was assessed by their parameter count (model size) and estimated inference time on a CPU, crucial factors for deployability on resource-constrained wearable devices. These metrics are presented in Table 3.

3.4.1. *Model size*

As anticipated from their design, the MobileNet-inspired CNN, with approximately 12,252 parameters, was substantially more lightweight than the CompactCNN, which had over 52,705 parameters. This confirms the effectiveness of depthwise separable convolutions in significantly reducing the number of trainable parameters and thus the memory footprint of the model, making it more suitable for edge computing scenarios.

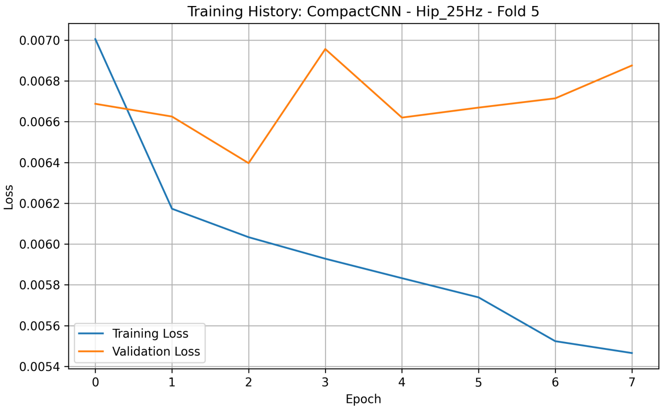

Figure 12. Training history of the CompactCNN model for the Hip_25Hz condition, illustrating the decrease in both training and validation loss over epochs. This apparent convergence to very low loss values demonstrates that model optimization can occur without achieving meaningful task performance, as the model learned a trivial solution to minimize loss.

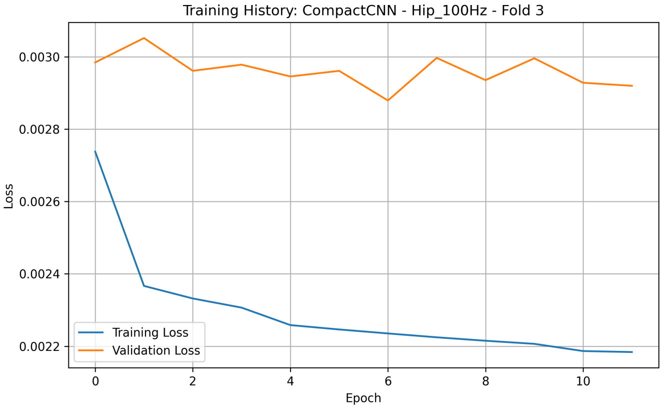

Figure 13. Training history of the CompactCNN model for Hip_100Hz data, showing the convergence of training and validation loss to very low values. This illustrates that low loss during training does not guarantee effective task performance, as the model learned a trivial solution despite apparent successful convergence.

3.4.2. *Inference time*

Both deep learning models demonstrated fast inference capabilities, indicating their potential for real-time processing on wearable devices. The estimated time to process a one-hour recording varied depending on the sampling frequency, as shown in Table 3. For instance,

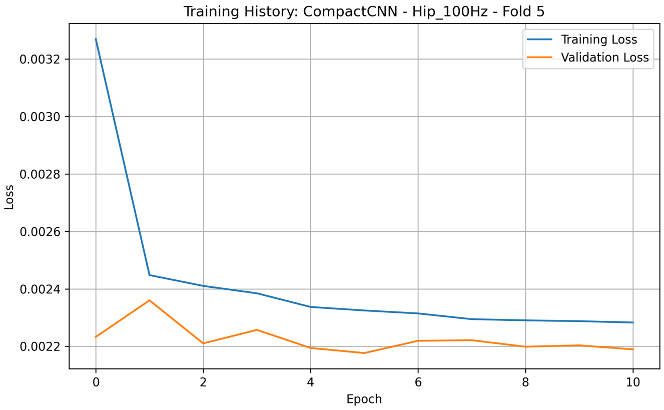

Figure 14. Training and validation loss for the CompactCNN model (Hip_100Hz, Fold 5) show consistent convergence to low values. This apparent training success, however, resulted in the model learning a trivial solution and failing to detect steps.

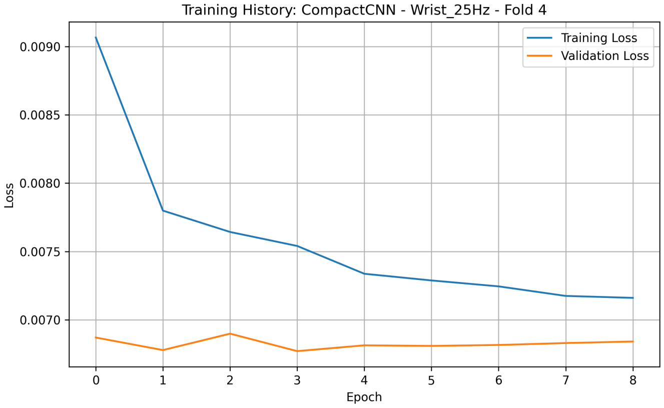

Figure 15. Training history for CompactCNN on Wrist_25Hz data (Fold 4). Both training and validation loss converged to very low values, illustrating that model convergence in terms of loss does not guarantee effective step detection performance.

the CompactCNN took approximately 37.57 seconds for a 100Hz hip recording but only about 7.00 seconds for a 25Hz hip recording. Similarly, the MobileNetCNN processed a 100Hz hip recording in 11.55 seconds and a 25Hz hip recording in 10.13 seconds. The faster inference times for 25Hz data are a direct consequence of the smaller input window size (50 samples for 25Hz vs. 200 samples for 100Hz) requiring fewer computations per window. These inference times are well within the requirements for most real-time activity monitoring applications.

3.5. *Impact of sensor location and sampling frequency*

Statistical tests were performed using paired Wilcoxon signed-rank tests to assess the influence of sensor placement and sampling rate on model performance, building upon the overall results presented in Table 3 and Figure 1.

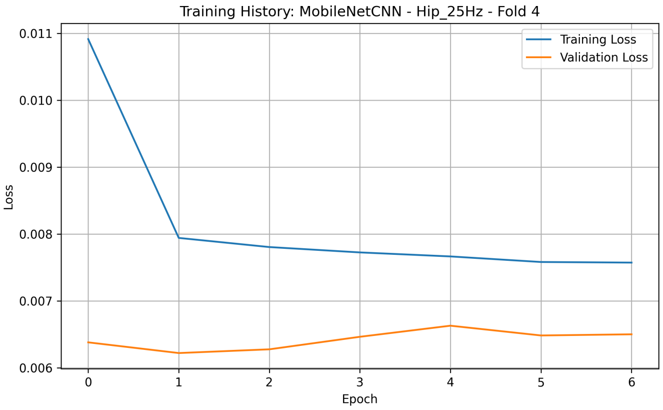

Figure 16. Training and validation loss history for the MobileNetCNN model (Hip_25Hz, Fold 4). The plot demonstrates that the model converged to low loss values, highlighting that successful loss minimization does not guarantee effective step detection performance in datasets with extreme class imbalance.

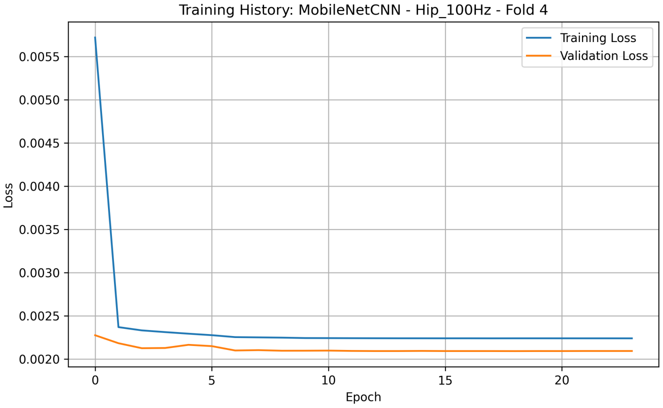

Figure 17. Training and validation loss history for the MobileNetCNN model (Hip_100Hz, Fold 4). The rapid decrease and convergence to low loss values indicate successful model optimization. However, this apparent learning reflects the model adopting a trivial solution due to severe class imbalance, resulting in a failure to detect steps.

3.5.1. *Location comparison (Hip vs. wrist)*

For the **Baseline** algorithm, sensor location had a profound and statistically significant impact on performance. The difference in MAPE between hip and wrist data was highly significant for both 100Hz ($p < 0.001$) and 25Hz ($p < 0.001$) sampling frequencies. This quantitatively confirms the observation that the hip is a far more reliable location for accurate step counting using this traditional approach. The F1-score was also signif-

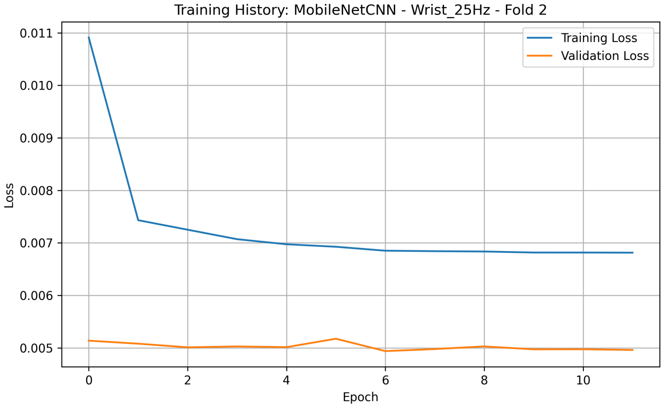

Figure 18. Training history of the MobileNetCNN model for Wrist_25Hz data (Fold 2), showing the convergence of training and validation loss to low values. This figure illustrates that deep learning models can minimize the loss function without learning the target task, as evidenced by their subsequent complete failure in step detection.

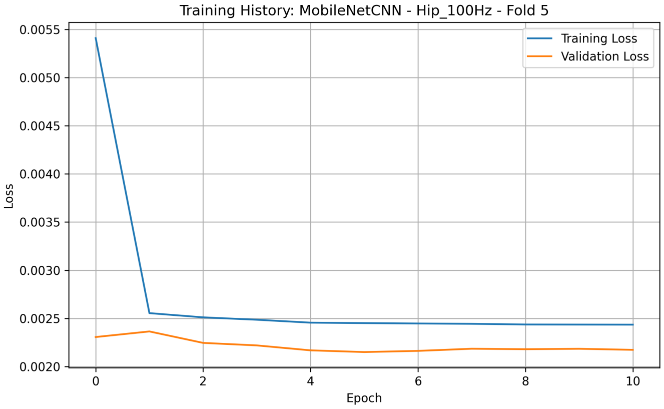

Figure 19. Training history of MobileNetCNN for Hip_100Hz data (Fold 5), showing training and validation loss converging to low values. This illustrates that low loss does not guarantee meaningful performance when models learn trivial solutions due to severe class imbalance, leading to the deep learning models' failure in step detection.

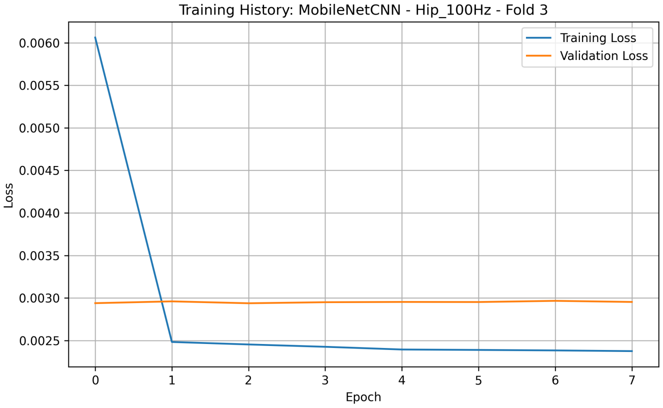

Figure 20. Training history for the MobileNetCNN model under the Hip_100Hz condition (Fold 3), displaying the training and validation loss curves. Both loss metrics rapidly decrease and converge to very low values, indicating successful model optimization. This apparent convergence, however, reflects the model learning a trivial solution due to severe class imbalance, resulting in its overall failure to detect steps.

icantly lower for wrist-worn data at 25Hz ($p = 0.0037$) compared to hip-worn data. This suggests that the higher-frequency components of the accelerometer signal, which are better captured at 100Hz, are more critical for reliable step detection at the wrist, where signal patterns are more complex and subtle.

For the **deep learning models**, no statistically significant differences were found between hip and wrist performance across any metric. This result is a direct consequence of their universal failure to detect steps; since both models consistently predicted zero steps for both locations, there was no statistical variance in their performance to detect.

3.5.2. *Frequency comparison (100Hz vs. 25Hz)*

For the **Baseline** algorithm, reducing the sampling frequency from 100Hz to 25Hz had a statistically significant negative impact on the F1-score for wrist-worn data ($p < 0.001$), but not for hip-worn data ($p = 0.35$). This suggests that the higher-frequency components of the accelerometer signal, which are better captured at 100Hz, are more critical for reliable step detection at the wrist, where signal patterns are more complex and subtle. In contrast, the more pronounced and consistent gait signals from the hip are sufficiently captured even at the lower 25Hz sampling rate for this algorithm. No significant impact on MAPE was observed for either location when comparing frequencies.

For the **deep learning models**, similar to the location comparison, no statistically significant differences were found when comparing 100Hz and 25Hz data. This is again attributable to their uniform lack of practical performance, as they failed to detect steps regardless of the sampling frequency.

3.6. *Demographic analysis*

A demographic analysis was conducted to investigate whether model performance varied by participant sex or age group. This analysis was performed on the CompactCNN model under the Hip_100Hz condition, which, despite its practical failure, nominally had the highest (though still negligible) mean F1-score among the deep learning models. The demographic analysis results are visually presented in Figure 21.

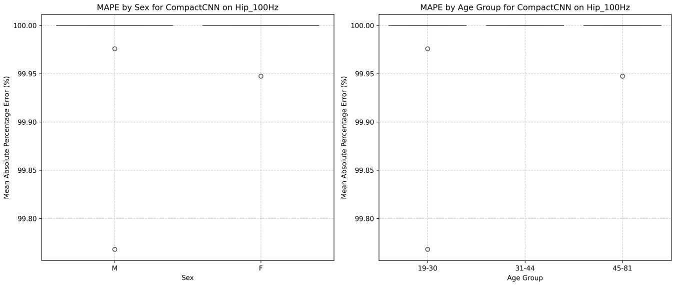

Figure 21. Box plots illustrating the Mean Absolute Percentage Error (MAPE) of the CompactCNN model (Hip_100Hz) by participant sex (left) and age group (right). The uniformly high MAPE, consistently at 100% across all demographic categories, visually confirms the model's universal failure to detect steps, indicating no meaningful performance variation based on participant demographics.

The Mann-Whitney U test for sex and the Kruskal-Wallis test for age groups yielded no statistically significant differences in MAPE (sex: $p = 0.605$; age: $p = 0.349$). However, this finding is not meaningful for practical interpretation. As visually confirmed in Figure 21, the deep learning model consistently predicted zero steps for all participants, resulting in a MAPE of 100% for every individual, regardless of their demographic profile. Therefore, the statistical tests were comparing identical distributions of 100% error, which naturally led to the conclusion of no difference. This highlights that while demographic analysis is crucial for robust models, it becomes moot when the foundational model performance is non-existent.

In summary, the experimental results demonstrate that a traditional, well-tuned signal processing algorithm can provide viable step count estimates from hip-worn sensors, achieving relatively low MAPE. However, its performance significantly degrades when applied to more complex wrist-worn data, indicating its limitations in handling noisier and more varied signal characteristics. In stark contrast, the deep learning approaches, despite their theoretical capabilities and efficient architectures, universally failed to detect steps across all conditions. This comprehensive failure, characterized by near-zero F1-scores and 100% MAPE (as shown in Table 3 and Figure 1), occurred because the models converged to a trivial solution of predicting no steps (Figure 5), an outcome driven by the extreme class imbalance inherent in the step detection task. Even the application of Focal Loss could not adequately mitigate this challenge. This underscores a critical learning point: while deep learning models can achieve low loss values during training (as exemplified in Figure 6 and related figures), this does not guarantee meaningful task performance, especially in scenarios with highly imbalanced physiological time-series data where sparse event detection is required. The computational efficiency of the deep learning models, while promising, becomes irrelevant in the absence of practical detection capability.

4. CONCLUSIONS

4.1. *Problem and objectives*

Accurate and resource-efficient step counting from wearable devices in free-living conditions presents significant challenges, primarily stemming from variability in sensor placement (e.g., hip versus wrist), data sampling rates, and individual demographic differences. Critically, the inherent sparsity of step events within continuous accelerometer data streams leads to an extreme class imbalance, posing a substantial hurdle for machine learning models. This study aimed to comprehensively compare the performance of a traditional peak-detection algorithm against two lightweight deep learning architectures (a compact 1D Convolutional Neural Network and a MobileNet-inspired CNN) for step counting. Our primary objectives were to evaluate their accuracy, computational efficiency, and generalizability across varying sensor locations and sampling frequencies, with a particular focus on understanding how data imbalance impacts deep learning performance for precise, sparse event detection.

4.2. *Datasets and methods used*

We utilized a robust dataset of three-axis accelerometer data collected from 39 participants, encompassing both hip and wrist placements, at 100Hz and 25Hz sampling frequencies. Ground-truth step annotations provided the basis for evaluation. Data preparation involved segmenting continuous recordings into 2-second windows with 90% overlap, generating binary labels for step events. A rigorous subject-independent 5-fold cross-validation scheme was employed to assess model generalizability. The traditional baseline model relied on vector magnitude calculation, band-pass filtering, and a tuned peak-finding algorithm. The deep learning models, a Compact 1D-CNN and a MobileNet-inspired 1D-CNN, were trained using the Adam optimizer with an initial learning rate of 0.001, a 'ReduceLROnPlateau' scheduler, and early stopping. Crucially, Focal Loss ($\gamma = 2, \alpha = 0.25$) was implemented to address the severe class imbalance. Post-processing for deep learning models involved stitching window-level probabilities and applying a peak-detection algorithm to identify steps. Performance was evaluated using F1-score and Mean Absolute Percentage Error (MAPE) for accuracy, and model parameters and CPU inference time for computational efficiency. Statistical analyses, including Wilcoxon signed-rank tests for location/frequency effects and Mann-Whitney U/Kruskal-Wallis for demographic impacts, were performed.

4.3. *Results obtained*

The traditional peak-detection baseline demonstrated moderate performance for hip-worn data, achieving F1-scores of approximately 0.42-0.44 and MAPE values around 10-11%. However, its accuracy significantly degraded for wrist-worn data, with MAPE surging to over 50%, highlighting its sensitivity to complex signal patterns. In stark contrast, both deep learning models, the CompactCNN and the MobileNet-inspired CNN, universally failed across all conditions (hip/wrist, 100Hz/25Hz). They consistently predicted zero steps, resulting in F1-scores of effectively 0.000 and MAPE values of 100%. This comprehensive failure occurred despite seemingly successful training, where both training and validation loss converged to very low values. This indicates that the models learned a trivial solution by predicting the majority class (”no step”) to minimize overall loss, without effectively identifying the sparse positive (step) events. While the deep learning models were computationally efficient, with the MobileNet-inspired CNN being particularly lightweight (12,252 parameters) and both exhibiting fast inference times suitable for real-time applications, their lack of practical step detection capability rendered this efficiency moot. Statistical analyses confirmed the significant impact of sensor location on the baseline model's performance but yielded no meaningful differences for the deep learning models due to their pervasive failure.

4.4. *What we have learned*

This study provides critical insights into the challenges of wearable step counting. First, traditional signal processing methods can provide viable performance for well-behaved signals (e.g., hip-worn sensors) but are limited by their fixed-rule nature when confronted with complex, noisy signals from alternative placements (e.g., wrist-worn sensors). Second, and most importantly, our findings highlight a profound challenge for deep learning in highly imbalanced physiological time-series data where sparse event detection is required. Despite using a specialized loss function (Focal Loss) designed to mitigate class imbalance, the deep learning models converged to a trivial solution, effectively learning to predict only the dominant ”no step” class. This demonstrates that optimizing loss functions alone, even those tailored for imbalance, does not guarantee meaningful task performance when the minority class is extremely rare. The apparent success in loss reduction during training can be misleading and does not necessarily translate into the desired practical utility for event detection. This necessitates a re-evaluation of current deep learning strategies for such tasks, potentially requiring more advanced techniques for handling extreme imbalance, such as sophisticated data augmentation for the minority class, novel architectural designs specifically for sparse event recognition, or hybrid approaches that combine the strengths of signal processing with deep learning. While the computational efficiency of the deep learning models is promising for wearable device deployment, it is irrelevant without the foundational capability to accurately detect the target events. This work underscores the need for robust validation strategies that go beyond loss metrics, particularly for applications involving highly imbalanced physiological event detection.

REFERENCES

Abadleh, A., Al-Hawari, E., Alkafaween, E., & Al-Sawalqah, H. 2018, Step Detection Algorithm For Accurate Distance Estimation Using Dynamic Step Length. https://arxiv.org/abs/1801.02336

Agarwal, A., Prabha, S., & Yadav, R. 2024, Exploratory Data Analysis for Banking and Finance: Unveiling Insights and Patterns. https://arxiv.org/abs/2407.11976

Agarwal, S., Sahu, G., Puri, A., et al. 2025, LitLLM: A Toolkit for Scientific Literature Review. https://arxiv.org/abs/2402.01788

An, H., Narechania, A., Wall, E., & Xu, K. 2024, vitaLITy 2: Reviewing Academic Literature Using Large Language Models. https://arxiv.org/abs/2408.13450

Anda, F., Lillis, D., Kanta, A., et al. 2019, Improving Borderline Adulthood Facial Age Estimation through Ensemble Learning, doi: https://doi.org/10.1145/3339252.3341491

Beddar-Wiesing, S., Moallemy-Oureh, A., Kempkes, M., & Thomas, J. M. 2025, Absolute Evaluation Measures for Machine Learning: A Survey. https://arxiv.org/abs/2507.03392

Bingol, H. O., & Basar, O. 2012, Asymmetries of Men and Women in Selecting Partner. https://arxiv.org/abs/1211.1035

Blagec, K., Dorffner, G., Moradi, M., & Samwald, M. 2021, A critical analysis of metrics used for measuring progress in artificial intelligence. https://arxiv.org/abs/2008.02577

Chen, C., & Pan, X. 2024, Deep Learning for Inertial Positioning: A Survey. https://arxiv.org/abs/2303.03757

Chen, Y., Raghuram, V. C., Mattern, J., et al. 2025, Testing Occupational Gender Bias in Language Models: Towards Robust Measurement and Zero-Shot Debiasing. https://arxiv.org/abs/2212.10678

Chen, Z. 2018, An LSTM Recurrent Network for Step Counting. https://arxiv.org/abs/1802.03486

Das, D., Narayan, B. S., Bhammar, A., & Bapat, J. 2025, Connecting the Unconnected – Sentiment Analysis of Field Survey of Internet Connectivity in Emerging Economies. https://arxiv.org/abs/2507.06827

Dehghani, A., Glatard, T., & Shihab, E. 2019, Subject Cross Validation in Human Activity Recognition. https://arxiv.org/abs/1904.02666

Ding, Y., Liu, J., Lyu, Z., et al. 2025, Voices of Her: Analyzing Gender Differences in the AI Publication World. https://arxiv.org/abs/2305.14597

Errey, N., Chen, Y., Dong, Y., et al. 2025, An Age-based Study into Interactive Narrative Visualization Engagement. https://arxiv.org/abs/2507.12734

Ferrer, L., Scharenborg, O., & Bäckström, T. 2024, Good practices for evaluation of machine learning systems. https://arxiv.org/abs/2412.03700

Foumani, N. M., Miller, L., Tan, C. W., et al. 2023, Deep Learning for Time Series Classification and Extrinsic Regression: A Current Survey. https://arxiv.org/abs/2302.02515

Goodarzi, P., Robin, Y., Schütze, A., & Schneider, T. 2023, Deep convolutional neural networks for cyclic sensor data. https://arxiv.org/abs/2308.06987

Gorriz, J. M., Clemente, R. M., Segovia, F., et al. 2024, Is K-fold cross validation the best model selection method for Machine Learning? https://arxiv.org/abs/2401.16407

Haresamudram, H., Rajasekhar, H., Shanbhogue, N. M., & Ploetz, T. 2024, Large Language Models Memorize Sensor Datasets! Implications on Human Activity Recognition Research. https://arxiv.org/abs/2406.05900

Huang, E., Yan, K., & Onnela, J.-P. 2022, Combining Accelerometer and Gyroscope Data in Smartphone-Based Activity Recognition using Movelets. https://arxiv.org/abs/2109.01118

Khan, S. S., & Abedi, A. 2022, Step Counting with Attention-based LSTM. https://arxiv.org/abs/2211.13114

Klein, I. 2024, Pedestrian Inertial Navigation: An Overview of Model and Data-Driven Approaches, doi: https://doi.org/10.1016/j.rineng.2025.104077

Knüpfer, A., & Callow, T. J. 2025, Data Version Management and Machine-Actionable Reproducibility for HPC based on git and DataLad. https://arxiv.org/abs/2505.06558

Koffman, L., Crainiceanu, C., & III, J. M. 2024, Comparing Step Counting Algorithms for High-Resolution Wrist Accelerometry Data in NHANES 2011-2014, doi: https://doi.org/10.1249/MSS.0000000000003616

Leinonen, T., Wong, D., Wahab, A., et al. 2024, Empirical investigation of multi-source cross-validation in clinical machine learning. https://arxiv.org/abs/2403.15012

Li, Z., Kesselman, C., Nguyen, T. H., et al. 2025, From Data to Decision: Data-Centric Infrastructure for Reproducible ML in Collaborative eScience. https://arxiv.org/abs/2506.16051

Liu, J., Tunguz, B., & Titericz, G. 2021, GPU Accelerated Exhaustive Search for Optimal Ensemble of Black-Box Optimization Algorithms. https://arxiv.org/abs/2012.04201

Mahoney, M. J., Johnson, L. K., Silge, J., et al. 2023, Assessing the performance of spatial cross-validation approaches for models of spatially structured data. https://arxiv.org/abs/2303.07334

Marquez-Carpintero, L., Suescun-Ferrandiz, S., Pina-Navarro, M., Cazorla, M., & Gomez-Donoso, F. 2025, CADDI: An in-Class Activity Detection Dataset using IMU data from low-cost sensors. https://arxiv.org/abs/2503.02853

Müller, D., & Kramer, F. 2019, MIScnn: A Framework for Medical Image Segmentation with Convolutional Neural Networks and Deep Learning, doi: https://doi.org/10.1186/s12880-020-00543-7

Naser, M. Z., & Alavi, A. 2020, Insights into Performance Fitness and Error Metrics for Machine Learning, doi: https://doi.org/10.1007/s44150-021-00015-8

Pabico, J. P. 2015, Inferences in a Virtual Community: Demography, User Preferences, and Network Topology. https://arxiv.org/abs/1507.08347

Peddi, R., Arya, S., Challa, B., et al. 2024, CaptainCook4D: A dataset for understanding errors in procedural activities. https://arxiv.org/abs/2312.14556

Pillai, A., Lea, H., Khan, F., & Dennis, G. 2020, Personalized Step Counting Using Wearable Sensors: A Domain Adapted LSTM Network Approach. https://arxiv.org/abs/2012.08975

Renault, M., Younes, H., Tessier, H., et al. 2025, Event Classification of Accelerometer Data for Industrial Package Monitoring with Embedded Deep Learning. https://arxiv.org/abs/2506.05435

Sedaghati, N., Kargar, M., & Abbaskhani, S. 2024, Introducing IHARDS-CNN: A Cutting-Edge Deep Learning Method for Human Activity Recognition Using Wearable Sensors. https://arxiv.org/abs/2411.11658

Shengwei, M., & Jianjie, L. 2018, Design of a PCIe Interface Card Control Software Based on WDF. https://arxiv.org/abs/1803.09052

Siu, H. C., Peña, J. D., Zhou, Y., & Allen, R. E. 2025, In Pursuit of Predictive Models of Human Preferences Toward AI Teammates. https://arxiv.org/abs/2503.15516

Tello, A., Degeler, V., & Lazovik, A. 2023, Too Good To Be True: performance overestimation in (re)current practices for Human Activity Recognition, doi: https://doi.org/10.1109/PerComWorkshops59983.2024.10503465

Terven, J., Cordova-Esparza, D. M., Ramirez-Pedraza, A., Chavez-Urbiola, E. A., & Romero-Gonzalez, J. A. 2025, Loss Functions and Metrics in Deep Learning, doi: https://doi.org/10.1007/s10462-025-11198-7

Wei, L. Z. Y. T. D. 2024, A Visual-inertial Localization Algorithm using Opportunistic Visual Beacons and Dead-Reckoning for GNSS-Denied Large-scale Applications. https://arxiv.org/abs/2411.19845

Yampolsky, Z., Kruzel, O., Fekson, V. K., & Klein, I. 2025, On Neural Inertial Classification Networks for Pedestrian Activity Recognition. https://arxiv.org/abs/2502.17520

Yates, L., Aandahl, Z., Richards, S. A., & Brook, B. W. 2022, Cross validation for model selection: a primer with examples from ecology. https://arxiv.org/abs/2203.04552

Yu, Y., Liu, Y., Zhang, J., Huang, Y., & Wang, Y. 2025, Understanding Generative AI Risks for Youth: A Taxonomy Based on Empirical Data. https://arxiv.org/abs/2502.16383

Yucedag, H., & Jatowt, A. 2025, Guess the Age of Photos: An Interactive Web Platform for Historical Image Age Estimation. https://arxiv.org/abs/2505.22031

Zhang, Y., Kong, Q., Ruan, T., Lv, Q., & Allen, R. 2024, A Comprehensive Analysis of Real-World Accelerometer Data Quality in a Global Smartphone-based Seismic Network. https://arxiv.org/abs/2407.03570

Zhang, Y., Zhang, Y., Zhang, Z., Bao, J., & Song, Y. 2018, Human activity recognition based on time series analysis using U-Net. https://arxiv.org/abs/1809.08113