

---

*Article*

# A Distributed Wearable Computing Framework for Human Activity Classification

Jhonathan L. Rivas-Caicedo <sup>1,\*</sup>, Kevin Niño-Tejada <sup>1</sup>, Laura Saldaña-Aristizabal <sup>1</sup> and Juan F. Patarroyo-Montenegro <sup>2,\*</sup>

<sup>1</sup> Department of Electrical and Computer Engineering, University of Puerto Rico, Mayaguez, PR 00680, USA; kevin.nino@upr.edu (K.N.-T.); laura.saldana@upr.edu (L.S.-A.)

<sup>2</sup> Department of Computer Science and Engineering, University of Puerto Rico, Mayaguez, PR 00680, USA

\* Correspondence: jhonathanleonardo.rivas@upr.edu (J.L.R.-C.); juan.patarroyo@upr.edu (J.F.P.-M.)

## Abstract

Human Activity Recognition (HAR) using wearable sensors plays a critical role in applications such as healthcare, sports monitoring, and rehabilitation. Traditional approaches typically rely on centralized models that aggregate and process data from multiple sensors simultaneously. However, such architecture often suffers from high latency, increased communication overhead, limited scalability, and reduced robustness, particularly in dynamic environments where wearable systems operate under resource constraints. This paper proposes a distributed neural network framework for HAR, where each wearable sensor independently processes its data using a lightweight neural model and transmits high-level features or predictions to a central neural network for final classification. This strategy alleviates the computational load on the central node, reduces data transmission across the network, and enhances user privacy. We evaluated the proposed distributed framework using our publicly available multi-sensor HAR dataset and compared its performance against a centralized neural network trained on the same data. The results demonstrate that the distributed approach achieves comparable or superior classification accuracy while significantly lowering inference latency and energy consumption. These findings underscore the promise of distributed intelligence in wearable systems for real-time and energy-efficient human activity monitoring.



Academic Editor: Lei Jing

Received: 30 June 2025

Revised: 28 July 2025

Accepted: 10 August 2025

Published: 12 August 2025

**Citation:** Rivas-Caicedo, J.L.; Niño-Tejada, K.; Saldaña-Aristizabal, L.; Patarroyo-Montenegro, J.F. A

Distributed Wearable Computing Framework for Human Activity Classification. *Electronics* **2025**, *14*, 3203. <https://doi.org/10.3390/electronics14163203>

**Copyright:** © 2025 by the authors. 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 (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Human Activity Recognition (HAR) plays a pivotal role in a wide range of applications, including healthcare, assisted living, sports performance analysis, and rehabilitation monitoring. Fundamentally, HAR entails classifying physical activities, such as walking, running, sitting, or more complex movements, based on sensor data collected from the human body. The growing availability and miniaturization of wearable sensors, including accelerometers, gyroscopes, magnetometers, and physiological monitors, have significantly improved the feasibility and granularity of HAR systems. These technological advancements have made it possible to perform continuous, real-time monitoring in naturalistic environments, thereby enabling personalized and proactive interventions [1].

Despite significant advancements in sensing technologies, achieving accurate and robust HAR under real-world conditions remains a considerable challenge. Traditional HAR

systems often employ a centralized architecture, where raw or pre-processed data from multiple wearable devices is transmitted to a central processor, typically a smartphone, cloud service, or edge hub, for classification. However, this centralized design introduces several technical limitations, particularly in dynamic, resource-constrained environments. These limitations include increased communication overhead, elevated energy consumption, data synchronization complexities, vulnerability to node disconnections, and concerns related to user data privacy [2]. Furthermore, centralized inference requires the central node to possess substantial processing power to handle multi-sensor data streams, which may be impractical for low-power, real-time applications such as continuous patient monitoring or remote fitness tracking.

### 1.1. Wearable Systems and Their Challenges

The adoption of wearable technologies in healthcare and activity monitoring has skyrocketed, particularly with the emergence of smartwatches, fitness trackers, and wearable medical devices. However, these systems are often limited by their hardware capabilities, especially in terms of computation, storage, and power [3]. This makes it impractical to run complex machine learning models locally on each device, pushing most processing tasks to the central node. Furthermore, wearable systems often operate under variable conditions: sensors may lose contact, network availability may fluctuate, and the physical placement of the device may introduce variability in signal quality. These challenges make robustness and fault tolerance essential design goals in modern HAR systems.

From a system architecture perspective, current wearable HAR solutions primarily focus on data capture and transmission, with minimal onboard processing [4,5]. The captured data is typically streamed to a mobile phone or cloud service, where inference is conducted using centralized neural network models. These models, while powerful, depend heavily on uninterrupted data flow and fail gracefully only in rare circumstances. A disconnection or failure in any part of the sensor network can significantly degrade performance or halt monitoring altogether [6].

### 1.2. Neural Networks in HAR and Edge Constraints

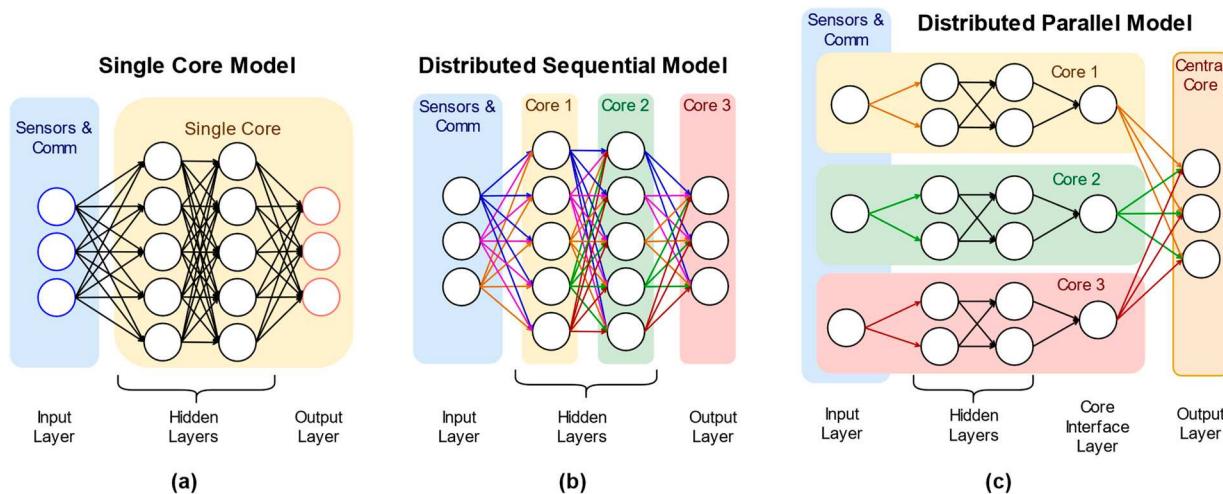
Recent advances in deep learning have dramatically improved HAR accuracy. Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs), and Transformer-based models have been applied successfully to multi-modal sensor data [4,5]. These models are capable of learning complex temporal and spatial features directly from raw signals, reducing the need for handcrafted features. However, their deployment remains challenging in wearable contexts due to computational and memory limitations.

Most deep learning-based HAR frameworks are still implemented in centralized architectures where data from all sensors is required simultaneously [1]. Edge deployment, which runs models directly on wearable devices, offers potential benefits in terms of latency reduction and data privacy, but is constrained by the limited processing capabilities of individual devices. Moreover, scaling such systems to accommodate multiple sensors or users exacerbates these limitations. Edge–cloud hybrid solutions have been proposed, but they still rely heavily on constant connectivity and central control, which may not be sustainable in all use cases.

### 1.3. Problem Statement: The Limits of Centralized and Distributed HAR

As shown in Figure 1a, a centralized Machine Learning (ML) model computes all measurements and extracted features together to obtain a series of outputs, which may be classifications or regressions. In a distributed sequential ML approach (Figure 1b), the model is partitioned by layers such that all measurements and features are received by the first core, which contains the first layer. The subsequent cores contain one layer at

a time, operating in a pipeline to reduce computation time. However, a single core will receive all measurements, which requires a direct connection to all sensors in the network. Finally, suppose each sensor contains a portion of the original network, unaware of the other signals. In that case, it will lose observability because some measurements will not be accounted for, resulting in an inaccurate inference (Figure 1c).



**Figure 1.** ML model distribution using sequential and parallel approaches: (a) example of a regular ML model running in a single mathematical core; (b) distributed model using a sequential approach. The arrows with the same color between layers indicate that they represent the same signal communicated between layers, and they are separated for illustrative purposes; (c) distributed parallel model. Each sensor signal is computed by its core, which is isolated from the cores of other sensors.

The fundamental problem with centralized HAR models lies in their structural dependency on full-sensor data integration at a single location. This dependency introduces significant bottlenecks in terms of latency, scalability, and fault tolerance. In scenarios where multiple sensors are deployed across the body, maintaining synchronization and uninterrupted data streams becomes increasingly difficult. Centralized systems are also more vulnerable to performance degradation due to packet loss, sensor dropout, or delayed data transmission.

In addition to these technical limitations, centralized systems pose privacy risks. Raw sensor data often contains sensitive information about users' behaviors and routines. Transmitting this data across networks increases the risk of potential security breaches. Similarly, energy consumption becomes a critical concern when high volumes of data must be wirelessly transmitted, particularly in battery-constrained devices like smartwatches or wearable patches [7].

#### 1.4. Proposal: A Distributed Framework for HAR

To address these challenges, we propose a *distributed neural network architecture* for human activity classification using wearable sensors. In this architecture, each wearable device processes its local data stream using a lightweight neural model trained to produce high-level feature representations or preliminary predictions. These outputs, rather than the raw data, are transmitted to a central classifier, which integrates information from all devices to perform final activity recognition. This distributed framework offers several compelling advantages:

- **Reduced Communication Overhead:** only concise representations are transmitted, minimizing bandwidth use and energy consumption.

- **Improved Fault Tolerance:** the system remains functional even if one or more devices disconnect or fail temporarily.
- **Enhanced Scalability:** new sensors can be added with minimal impact on the central node's computational load.
- **Greater Privacy:** since raw data is not transmitted, the system inherently protects sensitive user information.

This architectural shift allows us to reimagine HAR systems not as monolithic, centrally controlled pipelines but as collaborative, adaptive, and modular networks of intelligent agents. Each sensor node contributes to the overall classification task without requiring complete access to all data, thereby enabling deployment in real-time, low-power, and privacy-sensitive scenarios.

### 1.5. Related Work

Recent research in HAR has increasingly shifted from traditional centralized processing toward edge-based and distributed approaches that better accommodate real-world deployment constraints. Centralized models such as those employing deep convolutional and recurrent networks have demonstrated strong performance across various benchmark datasets [8,9]. These models often aggregate raw multi-sensor data to a cloud or local server for classification, achieving high accuracy but suffering from high communication costs, latency, and vulnerability to sensor failures or dropouts [10]. To address these issues, edge intelligence techniques have been developed, allowing devices to perform localized inference. For example, Haque et al. introduced LightHART, a transformer-based HAR model optimized for execution on mobile processors without sacrificing accuracy [11]. Similarly, the work of Agarwal and Alam proposed an ultra-lightweight deep learning model tailored to wearable hardware constraints [7].

Federated learning (FL) has emerged as a promising solution to reduce data transmission and enhance privacy. Frameworks such as FedHAR [12], FedHealth [13], and FedOpenHAR [14] enable multiple wearable devices to collaboratively train models while retaining data locally. These approaches mitigate privacy risks but still depend on stable connectivity and frequent model synchronization. Spiking neural networks (SNNs), like those explored in [15], offer biologically inspired, event-driven computation with low energy consumption, making them suitable for on-body deployment, though training remains challenging. Meanwhile, hybrid distributed architectures such as adARC [16] and Sannara et al.'s distributed sensor fusion scheme [17] combine local lightweight inference with central aggregation of intermediate representations or decisions, offering robustness to disconnections and improved scalability.

In addition to these frameworks, recent studies have explored enhancements to traditional FL through communication-efficient and heterogeneous model approaches. Sozinov et al. [18] evaluated FL for HAR using both softmax regression and deep neural networks, showing that although FL achieves slightly lower accuracy than centralized models, it significantly reduces privacy risks and data transmission costs. Gad et al. [19] proposed FedAKD, an FL strategy based on augmented knowledge distillation, which supports heterogeneous client models and improves communication efficiency by exchanging soft labels rather than gradients. Their results show superior performance under non-IID conditions, addressing one of the key limitations of traditional FL algorithms.

Beyond FL, split learning (SL) and federated split learning (FSL) have been introduced to reduce the computational load on wearable devices further while preserving privacy. Ndeko et al. [20] presented an FSL framework with differential privacy that partitions the model between the client and the server. Their method achieved improved accuracy,

reduced latency, and better privacy preservation compared to conventional FL, particularly in edge computing scenarios.

Building on these advancements in FL and SL, recent studies have emphasized the importance of energy awareness and heterogeneous device capabilities in real-world HAR deployments. Nguyen et al. proposed an energy-aware FL framework that dynamically selects participating clients based on energy status and data quality, achieving a better trade-off between accuracy, fairness, and device longevity [21]. Similarly, Thakur et al. introduced FedMeta, a meta-learning-based approach that enhances personalization and adaptation to non-iid data distributions commonly encountered in wearable HAR [22]. Saylam and Incel [23] provided a comprehensive review of FL techniques for edge devices, highlighting the challenges posed by device heterogeneity, communication bottlenecks, and the need for lightweight aggregation methods tailored to wearable systems.

In parallel, research has also focused on distributed training and inference paradigms that relax the server-centric assumptions of traditional FL. Khan et al. developed a multi-frequency federated learning (MFFL) algorithm for HAR using head-worn sensors, enabling asynchronous updates from devices operating at different sampling and computation rates while maintaining both accuracy and energy efficiency [24]. Furthermore, decentralized training approaches, such as ring-based and peer-to-peer topologies, have been explored to eliminate single points of failure, allowing HAR systems to maintain progress even under intermittent connectivity. These emerging strategies emphasize adaptability, privacy preservation, and scalability, making them particularly relevant for next-generation distributed HAR frameworks.

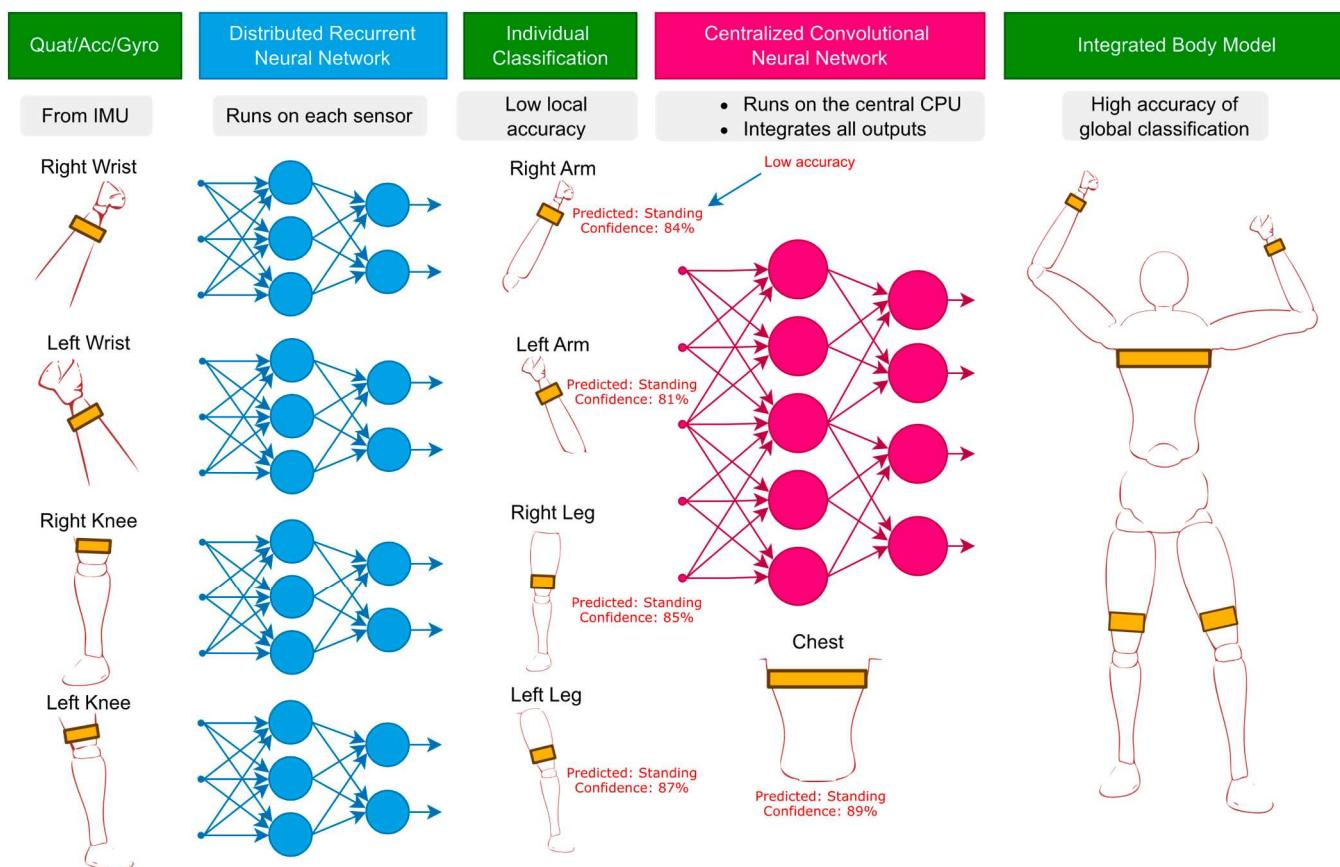
Complementary to learning strategies, energy efficiency remains a central concern for wearable HAR systems. Ding et al. [25] proposed dynamic inference schemes and sensor sampling strategies to reduce power consumption without sacrificing performance. Similarly, Rezaie et al. [26] introduced an adaptive activity-aware algorithm that adjusts sensing and computation rates based on user context, achieving substantial energy savings. These techniques focus on prolonging battery life, though they often assume single-device processing without inter-sensor coordination.

Despite these advances, many existing solutions either compromise accuracy for efficiency or require architectural assumptions (e.g., synchronized updates or homogeneous sensors) that are difficult to maintain in real-world deployments. In contrast, the framework proposed in this work adopts a modular, distributed architecture in which each sensor node runs an independent neural model to generate compact, informative outputs for fusion. This reduces communication overhead, enhances fault tolerance, and supports asynchronous operation, making it well-suited for scalable, privacy-conscious, and resource-constrained HAR deployments.

### 1.6. Contributions and Structure of the Paper

In this work, we attempt the following:

- Propose a **novel distributed neural network framework** for multi-sensor HAR using wearable devices, as shown in Figure 2.
- Implement and optimize lightweight local models for individual sensor nodes and a central fusion model for final classification.
- Validate our approach on a publicly available multi-sensor HAR dataset that we created and compare its performance against a centralized model.
- Demonstrate that our distributed model achieves **comparable or even better accuracy** while significantly reducing communication overhead and improving robustness to sensor disconnections.



**Figure 2.** General scheme of the proposed distributed computing wearable sensor network to be used as an application case. As shown, the Individual Classification stage is characterized by low accuracy, as each limb processes limited data. However, this accuracy is increased in the central core when all the outputs from each limb are integrated into a single classification verdict.

The remainder of the paper is organized as follows: Section 2 presents the dataset, experimental setup, and implementation details. Section 3 outlines the proposed distributed framework, detailing the design of both local and central neural network components. Section 4 reports the results, including accuracy comparisons, robust analysis, and energy consumption metrics. Finally, Section 5 discusses implications, limitations, and directions for future research.

## 2. Materials and Methods

### 2.1. Participants

A total of 67 participants (40 males and 27 females), aged between 20 and 60 years, were recruited for this study from the local university population and the surrounding community. All individuals voluntarily agreed to participate and provided informed consent before data collection. Participants self-reported being in good general health and physical condition at the time of the study.

The primary inclusion criterion was the absence of any known cardiovascular conditions or medical history that could compromise the participant's ability to perform the physical activities safely as outlined in the experimental protocol. Individuals with mobility impairments, recent injuries, or conditions affecting balance or coordination were excluded to ensure consistency and safety during data acquisition.

This study was approved by the University of Puerto Rico's Institutional Review Board (IRB) in compliance with the ethical standards established by the Collaborative Institutional Training Initiative (CITI Program) for research involving human subjects. All procedures

were reviewed and authorized before data collection. Before participating, all individuals received a detailed explanation of the study's objectives and methods and provided written informed consent.

## 2.2. Dataset

We collected a dataset using wearable sensors strategically placed on the human body. The dataset comprises multiple sessions of physical activity performed by 67 participants, with each session consisting of several repetitions of predefined activities. These activities were recorded using three types of signals: quaternion, linear acceleration, and angular velocity. These signals provide a rich representation of body movement and orientation, which is essential for robust classification.

Each session was recorded at a fixed sampling rate of 50 Hz, meaning that one sample corresponds to one timestamp of recorded data. On average, each session lasted approximately 37 min, resulting in thousands of individual data samples per session. The "Duration per session" column in Table 1 indicates the length of each activity per participant.

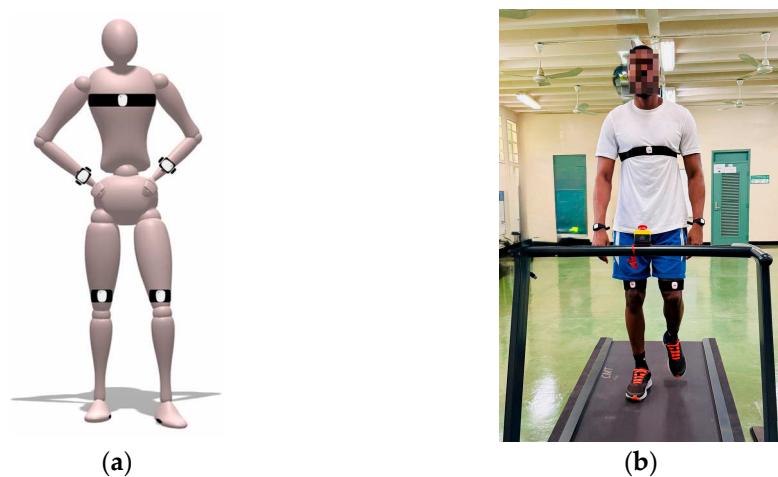
**Table 1.** List of classified activities with their corresponding label, description, and duration per session used in the training and evaluation of the model.

Activities	Label	Description	Duration per Session (s)
Sitting	0	Participant remains seated and mostly still	600
Folding Clothes	1	Participant simulates folding garments by hand	300
Sweeping	2	Participant mimics a sweeping motion with arms	300
Walking	3	Participant walks at a regular pace	300
Moving Boxes	4	Participant lifts and moves small to medium boxes	300
Riding a Bike	5	Participant performs a cycling motion while seated	300

Table 1 summarizes the list of activities selected for classification, their corresponding labels, and the average duration per subject for each activity. This provides a detailed overview of how the dataset was constructed for training and evaluating the classification models.

Many studies in the literature rely primarily on acceleration and gyroscope signals for activity recognition, as these modalities capture translational and rotational motion effectively [27]. However, in this work, we also incorporated quaternions because they represent the sensor's orientation in 3D space without suffering from the singularities and ambiguities [28]. By incorporating quaternions, the system achieves a more stable and comprehensive understanding of body posture and joint movement, which is particularly valuable when classifying activities that involve complex or continuous changes in orientation and speed.

To capture the full-body dynamics during the selected activities, wearable sensors were strategically placed on key locations of the body. In this study, five sensors were used and positioned as follows: on the chest to provide a reference for torso orientation and posture; on the right and left wrists to monitor upper limb movements; and on the left and right knees to capture lower limb dynamics. This sensor configuration enables the model to learn both global and local motion patterns across different body segments, improving its ability to distinguish between similar activities. A visual representation of the sensor placement is shown in Figure 3. The dataset collected is available in the resource [29].



**Figure 3.** Body sensor placement configuration: (a) wearable sensing prototype; (b) human subject walking while instrumented with the proposed sensor setup.

Anatomical considerations guided the placement strategy and aligned with established practices in the human activity recognition literature. Sensors located on the chest and limbs offer a suitable trade-off between capturing relevant motion features and ensuring wearability. While this configuration proved effective for the activities included in this study, it is essential to note that no experiments were conducted to assess alternative sensor placements. Classification performance may vary with different sensor configurations, particularly when motion sensitivity or signal discriminability is reduced due to location-specific factors.

### 2.3. Experimental Design

To evaluate the proposed distributed framework for human activity recognition, a controlled experiment was conducted using five MetaMotionRL wearable sensors (version r0.5) from Mbientlab Inc. (San Jose, CA, USA). Each sensor was configured to stream specific inertial data types at a frequency of 50 Hz, utilizing Bluetooth Low Energy (BLE) for wireless communication with a MetaBase app running on an iOS device.

The chest sensor captured quaternions, 3-axis linear acceleration, and 3-axis angular velocity. The sensors placed on the right hand and left knee streamed only acceleration and gyroscope data, while the sensors on the left hand and right knee transmitted quaternion data exclusively. All data streams were timestamped at the source and subsequently synchronized offline using a standard reference clock. Synchronization was achieved through linear extrapolation based on timestamps to ensure temporal alignment across all sensor inputs.

Each of the 67 participants underwent a structured experimental session with a total duration of approximately 37 min. The session consisted of six predefined activities listed in Table 1: folding clothes, sweeping, walking, moving boxes, riding a bike, and sitting. The first five activities were performed continuously for 5 min each, while the remaining time was allocated to the sitting activity.

All sessions were conducted under supervised conditions in a controlled indoor environment, following a strict and consistent timeline across participants to ensure standardization and reproducibility of the dataset. Each session began immediately after sensor initialization and synchronization, and data acquisition was conducted in real-time over approximately 37 min. per participant. However, model training and classification were performed offline after data collection. The sensor signals remained stable throughout each session, with no noticeable drift or degradation in data quality. This ensured consistent labeling and reliable input for offline model development and evaluation. Future work

will investigate longer-term deployments to assess the impact of continuous use, including possible effects of sensor drift or synchronization loss on classification performance.

Transitions between activities were excluded from the final dataset to avoid introducing ambiguity in the activity labels. The sitting activity occurred at the beginning of the session and after every two subsequent activities, resulting in a total of four sitting periods. The short intervals during which the participant transitioned from one activity to another were automatically identified by software using the predefined time intervals for each task and systematically discarded, ensuring that each data segment corresponded strictly to a single, well-defined activity. While this design decision promotes label consistency and simplifies the learning process, it does not account for the transitional or overlapping motions that often occur in real-world deployments. Therefore, in future work, we aim to extend the framework toward real-time implementation, where detecting and classifying transitions will be critical for enabling continuous and adaptive activity recognition in dynamic environments.

#### 2.4. Data Processing

To ensure temporal consistency, all recordings were synchronized post hoc using the timestamp information provided by the sensors. A linear extrapolation method was applied to align the signals across devices with a uniform sampling grid. Each recording session consisted of multiple predefined activities, performed in a controlled environment. Activity labels were assigned based on the sequence and timing of each task, which were recorded during acquisition.

To ensure the validity of the labeling process, all data acquisition sessions were supervised in real time by a member of the research team, who followed a predefined script indicating the sequence and duration of each activity. The timing of each task was logged concurrently with the sensor data to provide accurate alignment between activity execution and label assignment. After acquisition, the recorded sensor signals were visually inspected using custom plotting tools to verify that signal transitions (e.g., changes in acceleration or orientation) were consistent with the expected activity changes. Sessions where inconsistencies or ambiguities in timing or execution were detected were either repeated or excluded from the dataset. Although inter-observer agreement was not employed, this manual supervision and verification process served to ensure the reliability of the assigned labels and mitigate the effects of possible temporal drift or misalignment.

During preprocessing, any data entries with invalid or missing labels were excluded. In addition, samples containing missing values in any of the sensor features were removed to ensure the integrity of the dataset. Feature values were then normalized using z-score normalization (zero mean and unit variance), computed based on the training set statistics. This normalization was applied using the utilities from the scikit-learn library and saved for consistent scaling of future data.

The resulting normalized time series were segmented into overlapping windows of 50 samples (equivalent to 1 s) using a sliding window approach with a stride of 25 samples (50% overlap). Each window was assigned the most representative label from the segment to generate the final dataset used for model training and evaluation.

### 3. Proposed Distributed Framework

To address the limitations of centralized architectures in wearable human activity recognition systems, we propose a distributed neural network framework designed for deployment across multiple edge devices. Unlike conventional models that rely on aggregating all sensor data at a central processing unit, our approach distributes the inference workload by assigning individual neural networks to each sensor node. These local models

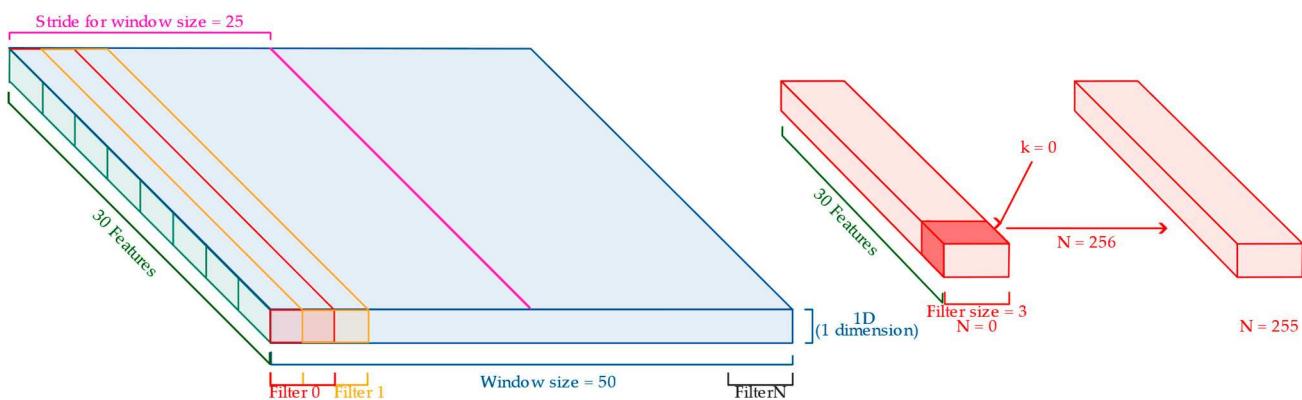
extract relevant motion features independently and transmit only compact, high-level representations to a central model, which is responsible for final decision-making.

This distributed architecture is designed to minimize communication overhead and energy consumption while enhancing scalability and robustness against common issues, such as sensor dropout or data transmission delays. In this initial stage, the framework was evaluated offline, simulating the behavior of distributed inference by independently processing each sensor's data through local models and aggregating their outputs in a central model. This approach allows for the validation of the distributed logic and its effectiveness in multi-sensor fusion before deploying the system on actual edge devices. The real-time deployment of this framework is outside the scope of this paper.

### 3.1. Centralized Model

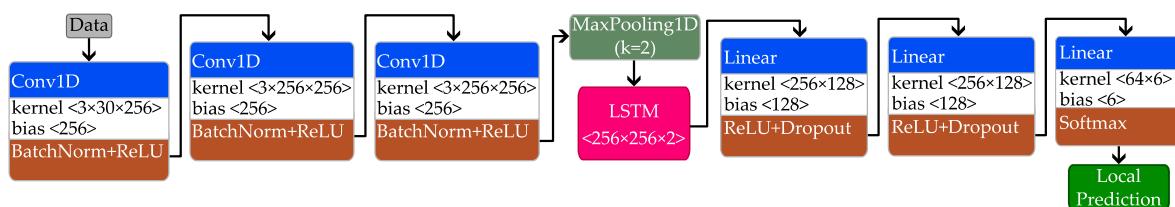
As an initial approach to human activity recognition, a centralized model was developed and trained using the complete set of synchronized sensor data. This model served as a baseline to evaluate the classification performance when a single neural network processes all information. A centralized architecture receives as input a concatenated vector that integrates the signals from all five sensors, combining features such as quaternions, linear acceleration, and angular velocity. This design assumes full availability of the data streams during inference, representing an ideal scenario without communication delays or disconnections. The purpose of this centralized model is twofold: to establish a performance benchmark under optimal conditions and to provide a direct comparison point for the subsequent implementation of a distributed architecture.

Figure 4 illustrates the structure of the input data and the operation of a single 1D convolutional layer applied to the raw sensor window. Each input sample consists of a sliding window of 50 timesteps and 30 features, where the 30 features correspond to the concatenation of sensor measurements (acceleration, angular velocity, and the four orientation components) collected from five body-worn devices. The vertical axis represents the feature dimension, while the horizontal axis spans the temporal sequence. The figure also shows how 1D convolutional filters with a kernel size of 3 operate along the time axis, integrating information across all 30 features at each step. This setup enables the network to extract local spatial-temporal patterns by sliding the filters across the sequence and generating feature maps that encode short-term dependencies. Although the figure depicts the first convolutional layer, the subsequent convolutional layers process the output of the preceding ones, similarly building progressively deeper hierarchical representations of the input signal.



**Figure 4.** Illustration of the input tensor structure and the operation of a 1D convolutional layer in the centralized model.

Figure 5 presents the architecture of the centralized model. The first part of the network consists of three consecutive 1D convolutional layers with a kernel size of 3 and appropriate padding to preserve the temporal resolution. Each convolutional layer is followed by batch normalization and a ReLU activation function. A max pooling layer with a kernel size of 2 is applied afterward to reduce the sequence length and emphasize the most relevant features. This is followed by a two-layer LSTM (Long-Short Term Memory) with a hidden size of 256, which encodes the temporal evolution of the signal across the reduced time dimension. The LSTM output is then passed through a series of three fully connected layers. The first two linear layers include ReLU activation and dropout for regularization, while the final layer applies a SoftMax function to generate the activity class prediction. This architecture was designed to strike a balance between expressiveness and generalization in a centralized setting, serving as a strong baseline for comparison against the distributed framework.



**Figure 5.** Implemented architecture for the Centralized Model.

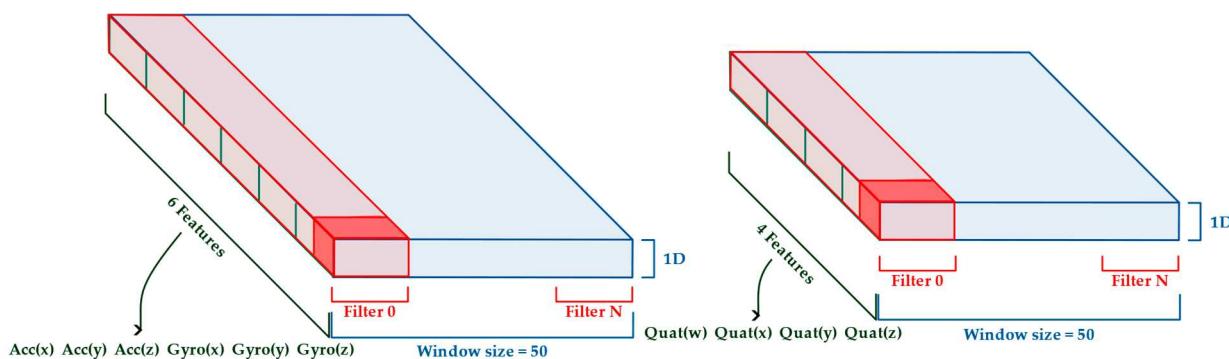
To ensure a robust architecture for the centralized model, several configurations of the CNN-LSTM (Convolutional Neural Network and Long-Short Term Memory) structure were explored, varying the number of convolutional layers and LSTM units. These configurations were evaluated on a held-out test set to assess generalization. A detailed comparison of these configurations and their corresponding performance is presented in Section 4. The best-performing configuration identified was selected for a more in-depth evaluation. To better understand its behavior, a confusion matrix was computed using validation data, along with precision, recall, and F1-score for each activity class. These metrics provide a detailed view of the model's performance on activities observed during training, helping to identify both well-classified categories and classes that may be confused due to similar motion patterns.

### 3.2. Distributed Model

To address the challenges of scalability, latency, and robustness in real-world deployments, a distributed neural network framework was implemented. Unlike the centralized model, which processes all sensor data jointly, the distributed approach decouples computation across multiple local models, each associated with a specific sensor and a central model that integrates their outputs. This design enables independent inference at the sensor level while allowing fusion at a higher level, reducing the computational burden on a single node and enhancing fault tolerance in case of sensor disconnections or transmission delays, an aspect that was addressed in a previous study focused on mitigating sensor disconnections [6].

#### 3.2.1. Local Model

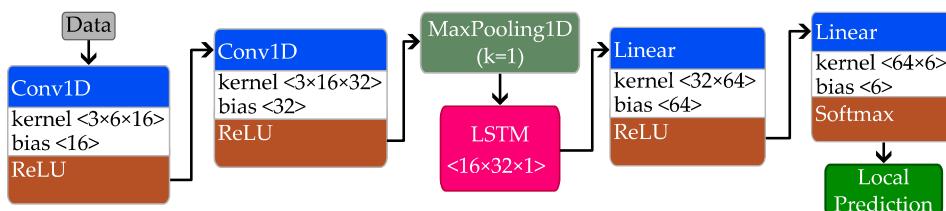
Each local model is designed to operate independently on the data collected from a single wearable sensor. These models process sensor-specific signals (such as quaternions, accelerometer, and gyroscope data) and output a probability distribution over the predefined set of activity classes. Figure 6 illustrates the structure of the input data and the operation of a single 1D convolutional layer applied to the raw sensor window.



**Figure 6.** Input tensor structure and the operation of a 1D convolutional layer in the local model.

Each model receives input windows of 50 timesteps and either six features (for sensors with accelerometer and gyroscope data) or four features (for sensors with quaternion data), resulting in tensors of shape  $50 \times 6$  or  $50 \times 4$ . The first stage consists of two 1D convolutional layers with a kernel size of 3 and increasing output channels (from 16 to 32), both of which are followed by ReLU activations. A max pooling layer with a kernel size of 1 is applied to preserve the temporal resolution. This is followed by a single-layer LSTM with a hidden size of 32, which captures temporal dynamics in the sensor data. The final representation is passed through two fully connected layers: the first with ReLU activation, and the second with a SoftMax output to produce class probabilities.

The architecture follows a lightweight CNN–LSTM structure optimized for execution on embedded devices, as illustrated in Figure 7.

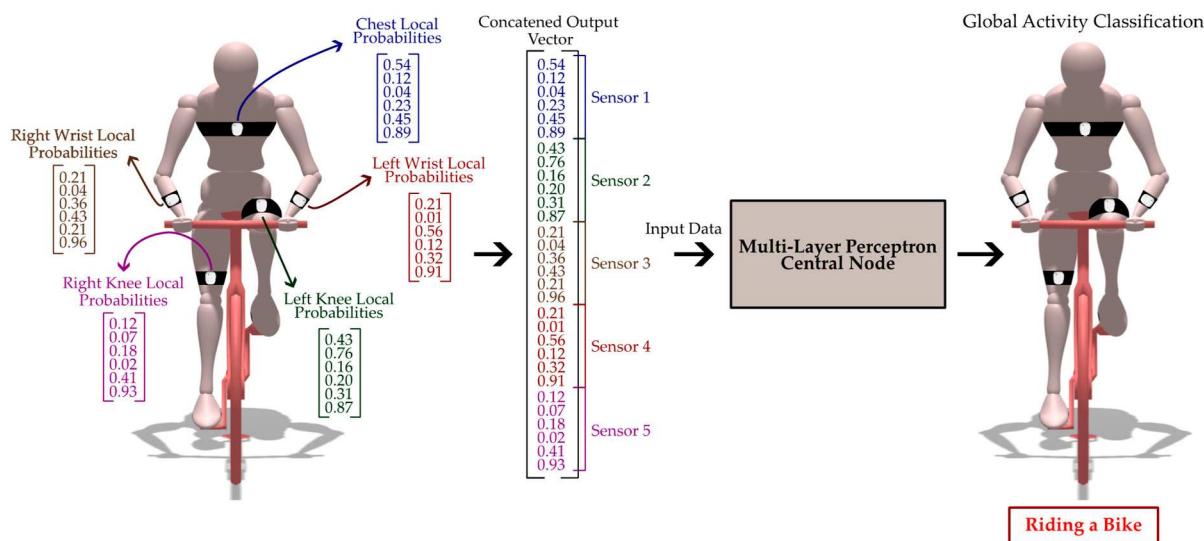


**Figure 7.** Implemented architecture for the local core on each sensor for the Distributed Model.

The lightweight design of each local model was intentionally chosen to facilitate real-time deployment on embedded platforms with limited computational resources. By minimizing the number of parameters and reducing architectural complexity, the models achieve low latency and energy efficiency while maintaining competitive classification performance. This design choice enables distributed inference directly at the sensor level, eliminating the need for constant communication with a centralized processor.

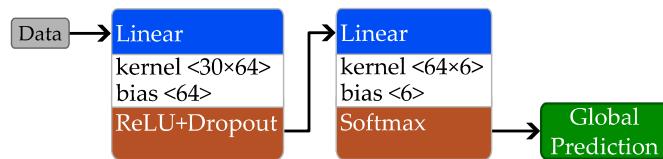
### 3.2.2. Central Model of the Distributed Network

The central model receives as input a single concatenated output vector formed by stacking the class probability distributions generated by each local model. Each local model, corresponding to a different body-worn sensor placed explicitly on the chest, left and right wrists, and left and right knees, outputs a probability vector of length  $C$ , where  $C$  is the number of activity classes. These vectors are concatenated vertically to form a unified input vector of dimension  $N \cdot C$ , where  $N$  is the number of sensors. Figure 8 illustrates how this representation encapsulates the local decisions of all sensors and enables the central multi-layer perceptron (MLP) to integrate them into a global activity classification.



**Figure 8.** Overview of the centralized decision fusion process in the proposed distributed framework. Each local model processes sensor-specific data and outputs a probability vector over the set of activity classes.

The architecture consists of an MLP composed of two fully connected layers, as shown in Figure 9. The first layer maps the 30-dimensional input to a 64-dimensional hidden representation, followed by a ReLU activation and dropout to promote generalization. The second layer applies a SoftMax activation over the final 6-dimensional output, producing the global activity prediction. This architecture enables the integration of individual local inferences while maintaining a compact and efficient structure, making it suitable for real-time decision fusion.



**Figure 9.** Implemented architecture for the central core on the Distributed Model.

This central model plays a key role in the distributed framework by acting as a decision-level fusion layer. While each local model is capable of performing standalone inference based on its respective sensor data, the central model enhances overall system performance by leveraging complementary information across sensors. This hierarchical design offers both robustness, allowing for partial predictions in the event of sensor failures, and scalability, as additional local models can be integrated seamlessly without requiring retraining of the entire system.

To assess the computational suitability of the proposed architecture for embedded and resource-constrained environments, we analyzed the complexity of both the local CNN-LSTM models and the central MLP. Table 2 presents the number of trainable parameters, floating point operations (FLOPs), and estimated memory usage for each model. These results demonstrate that the local models remain lightweight enough for on-device inference. At the same time, the central node incurs minimal computational overhead, making the entire framework suitable for deployment on wearable systems and low-power edge devices.

**Table 2.** Number of parameters, computational complexity (in FLOPs), and memory usage of the local CNN–LSTM models and the central MLP, highlighting the feasibility of real-time execution in embedded systems.

Model	Parameters		FLOPs	Memory Usage (MB)
	Layer	Trainable Parameters		
Local Node	Input Layer	0	229.83 KFLOPs	0.06
	Conv1D (1 × 3)	304		
	Activation (ReLU)	0		
	Conv1D (1 × 3)	784		
	Activation (ReLU)	0		
	MaxPooling (1 × 2)	0		
	LSTM (1 layer, hidden = 32)	6400		
	Dense (64)	2112		
	Activation (ReLU)	0		
Central Node	Predictions (Dense)	390	2.5 KFLOPs	0.01
	Total Trainable Parameters	9990		
	Input Layer	0		
	Dense (64)	1984		
Central Node	Activation (ReLU)	0	2.5 KFLOPs	0.01
	Dropout	0		
	Predictions (Dense)	390		
	Total Trainable Parameters	2374		

### 3.3. Training Configuration and Reproducibility

The local models, each based on a CNN–LSTM architecture, were trained independently for each sensor. Each model received as input only the data from its corresponding sensor and was trained as a standalone classifier. The central model, a lightweight MLP, was trained separately using the SoftMax outputs from the already trained local models as input features.

All models were trained using the categorical cross-entropy loss function, standard for multi-class classification problems. The Adam optimizer was selected due to its adaptive learning rate capabilities and efficient convergence, which have proven effective in human activity recognition (HAR) tasks [30]. A fixed learning rate of 0.001 was used, a value commonly adopted for stable convergence without excessive oscillations [31]. The batch size was set to 64, offering a good balance between generalization and training speed while remaining compatible with the memory constraints of embedded devices [32].

Training was limited to a maximum of 50 epochs. Early stopping was applied by monitoring validation loss, halting training if no improvement was observed for 10 consecutive epochs (patience = 10). While no additional regularization techniques such as dropout or L2 penalties were applied, the CNN–LSTM architecture incorporates max-pooling and ReLU activations, which naturally contribute to overfitting mitigation and feature sparsity [33].

The best-performing model on the validation set was saved and used for final evaluation. All training experiments were implemented in PyTorch v2.5.1 with CUDA 12.1 acceleration. Random seeds were fixed across all training and evaluation steps to ensure full reproducibility.

## 4. Results

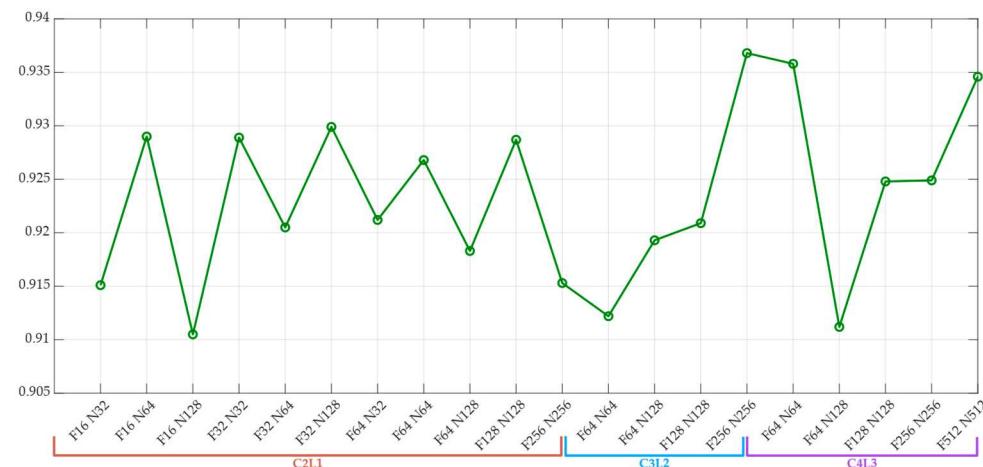
This section presents a comprehensive evaluation of both the centralized and distributed models using a test set composed of unseen data. The results include accuracy,

confusion matrices, and class-wise performance metrics, enabling a comparative analysis of the two approaches.

#### 4.1. Centralized Model Performance

To optimize the classification performance and assess the robustness of the centralized model, multiple architectural configurations were tested. Each configuration varied the number of convolutional and LSTM units, while maintaining the overall CNN–LSTM structure. The evaluation was conducted using a test set composed of data that was not used during training or validation, ensuring an unbiased assessment of generalization capability.

Figure 10 presents the accuracy obtained for each configuration. The models are grouped into three categories: C2L1, C3L2, and C4L3, where  $C$  denotes the number of convolutional layers and  $L$  the number of LSTM layers. Each configuration also varies in terms of convolutional filters ( $F$ ) and the number of hidden units in the LSTM layers ( $N$ ). Among all tested variants, the configuration with 256 convolutional filters and a two-layer LSTM with 256 hidden units ( $F = 256$ ,  $N = 256$ ,  $C = 3$ ,  $L = 2$ ) achieved the highest accuracy (93.68%), demonstrating the benefit of deeper representations when processing complex multi-sensor data.



**Figure 10.** Accuracy obtained during training under various architectural configurations of the centralized model. Each configuration explores a different combination of the number of filters ( $F$ ), fully connected neurons ( $N$ ), convolutional layers ( $C$ ), and LSTM layers ( $L$ ), aiming to identify the most effective setup for maximizing classification performance.

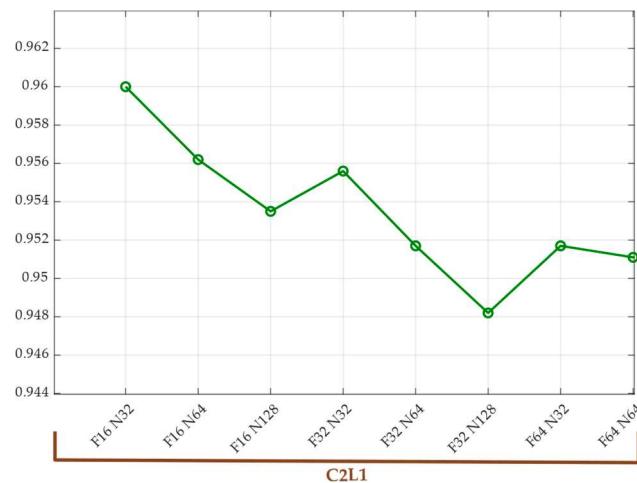
These results establish a robust and well-performing centralized baseline, which serves as a strong reference point for evaluating the distributed framework. The effectiveness of this configuration not only validates the design choices of the centralized approach but also provides a meaningful benchmark for subsequent comparisons.

#### 4.2. Distributed Model Performance

Building upon the centralized baseline, the distributed framework was evaluated under the same test conditions to enable a direct comparison. In this approach, each wearable sensor is associated with an independent local model responsible for processing its own data stream. These local models were trained separately using their respective sensor modalities and produce intermediate predictions in the form of SoftMax probability vectors.

To integrate the outputs from the local models, a lightweight central model based on an MLP was employed. This central node aggregates the class scores output by each sensor to generate the final activity classification.

Figure 11 presents the accuracy obtained for each configuration of the distributed framework. The models are grouped under the category C2L1, representing a lightweight and straightforward architecture tailored for embedded deployment. Among all tested variants, the configuration with 16 convolutional filters and a single-layer LSTM with 32 hidden units ( $F = 16$ ,  $N = 32$ ,  $C = 2$ ,  $L = 1$ ) achieved the highest accuracy (95.99%).



**Figure 11.** Accuracy achieved during training for various architectural configurations of the local models within the distributed framework. Each configuration represents a different combination of convolutional layers (C) and LSTM layers (L), aiming to identify the most effective structure for maximizing classification performance while maintaining a lightweight design.

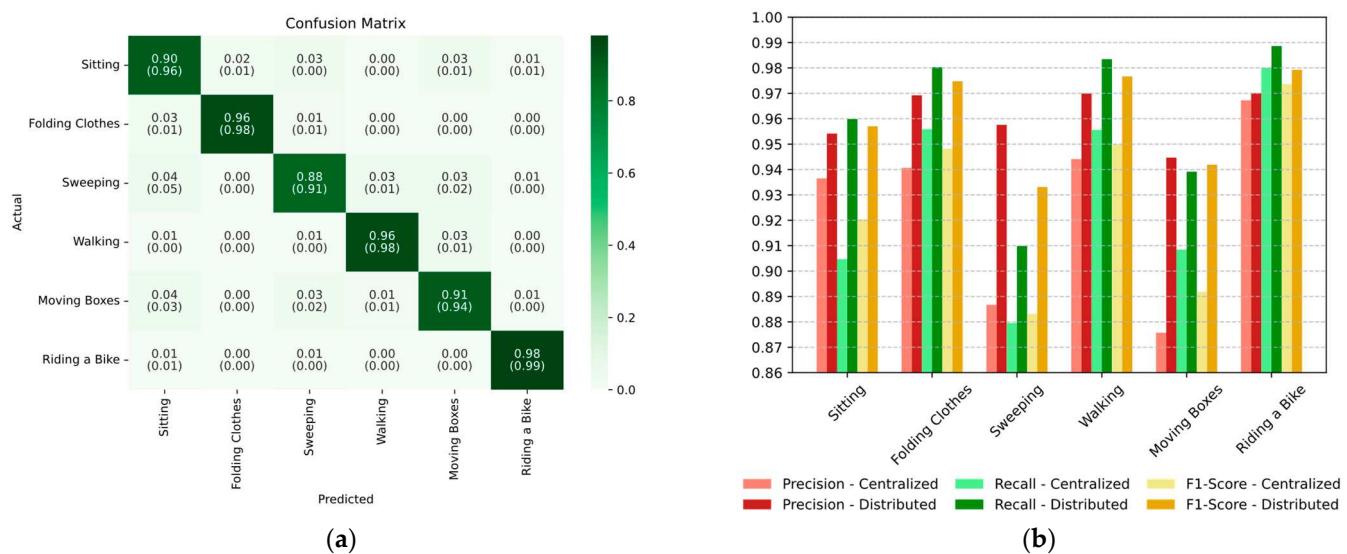
These results confirm that the distributed framework, despite relying on simpler and independently trained local models, can achieve competitive and better classification performance. The lightweight configuration with minimal computational complexity proved sufficient for capturing discriminative features when combined across sensors. This balance between efficiency and accuracy highlights the potential of distributed processing in wearable systems. A comparative analysis between the centralized and distributed approaches is presented in the following subsection.

#### 4.3. Comparison

To enable a direct evaluation of both approaches, the centralized and distributed models were tested under identical conditions using the same dataset. Figure 12 provides a detailed comparison of the class-wise performance for both approaches. It includes confusion matrices and the per-class precision, recall, and F1-scores. It can be appreciated that the distributed model achieved higher precision than the centralized approach, indicating a reduced rate of false positives in its predictions. In terms of overall accuracy and recall, both models performed comparably, with slight variations depending on the activity class. These results highlight the effectiveness of the distributed strategy, particularly considering its modularity and scalability, and set the stage for a more in-depth discussion on the trade-offs between the two approaches.

The comparative evaluation between the centralized and distributed models reveals clear and consistent performance differences in favor of the distributed approach. As shown in Figure 12, which presents the normalized confusion matrix, each value from the centralized model is accompanied by its distributed counterpart in parentheses. This format enables a direct, class-by-class comparison. Notably, the distributed model demonstrates improved classification accuracy in several key activities. For instance, the class Sitting shows an increase in true positive rate from 0.90 to 0.96, while Walking improves from 0.96 to 0.98. In more complex activities such as Sweeping and Moving Boxes, the distributed

model also shows notable gains, increasing from 0.88 to 0.91 and from 0.91 to 0.94, respectively, while simultaneously reducing off-diagonal misclassifications. These improvements suggest that sensor-level processing enables better preservation of temporal patterns and motion nuances that may be lost during early fusion in centralized architectures.



**Figure 12.** Comparison of classification performance between the centralized and distributed models on the test set: (a) normalized confusion matrix of the centralized model, where each centralized value is accompanied by its corresponding value in parentheses representing the distributed model. This enables direct comparison of prediction accuracy per class across both approaches; (b) class-wise bar plots of precision, recall, and F1-score for both models. Lighter bars represent the centralized model, while darker bars represent the distributed model, grouped by metric for each activity class.

Figure 12 further reinforces these findings by displaying per-class precision, recall, and F1-score for both models. The distributed model, represented by darker bars, consistently achieves higher recall across all classes. This trend is particularly evident in Sweeping, Walking, and Moving Boxes, where the recall gains are substantial. Since recall is critical in activity recognition systems to avoid missing actual events, these results highlight the distributed model's strength in capturing true positives. In terms of F1-score, which balances both precision and recall, the distributed model outperforms the centralized one in nearly every class, confirming its superior classification reliability.

While the centralized model performs slightly better in precision for a few activities, such as Sitting and Folding Clothes, the overall advantage shifts toward the distributed model due to its balanced improvement in both recall and F1-score. This trade-off is acceptable, especially in real-time systems where false negatives (missed detections) are more detrimental than occasional false positives.

In summary, the distributed model not only matches but often surpasses the centralized model in key evaluation metrics. Its superior performance in complex, dynamic activities demonstrates that localized processing with late fusion enhances robustness and generalization. These results support the design of distributed architectures in wearable activity recognition systems, where sensor-specific information and reduced latency are crucial for reliable real-time performance.

#### 4.4. Simulation of BLE Packet Loss and Sensor Disconnection

To evaluate the resilience of the proposed distributed framework under BLE communication failures, we conducted a series of experiments simulating sensor disconnections and packet loss during inference. These tests were designed to reflect real-world challenges

in wearable systems, including dropped connections, sensor malfunctions, and partial data loss.

The distributed system includes five unique sensor sources: the Chest, Left Knee, Right Hand, Left Hand, and Right Knee. The local models associated with each sensor were independently trained and fixed during evaluation. A centralized MLP model receives a 30-dimensional concatenated input vector formed by the class probability distributions of all five sensors. To simulate sensor failures or transmission loss, we manipulated the input vectors at inference time without retraining the central model, reflecting deployment conditions where real-time adaptation is not feasible.

We evaluated the following fault scenarios:

- Baseline (No loss): all sensors are active and transmitting valid data. This serves as the reference for maximum expected performance.
- BLE packet loss (10%): randomly zeroes out 10% of the predicted outputs for each sensor to simulate moderate transmission dropouts. The central model continues to receive data from all sensors.
- BLE packet loss (30%): a more severe version, where 30% of each sensor's outputs are zeroed out, simulating high BLE packet loss or interference.
- Single-sensor disconnection: Each sensor is individually removed by replacing its full output vector with zeros. This represents a total communication failure or power loss in one node.
- Two-sensor disconnection: All 10 possible combinations of two sensors being disconnected simultaneously were tested. This simulates more critical failures or simultaneous signal dropout in two channels, allowing for the analysis of sensor pair importance in global inference.

Each scenario was evaluated using identical test subjects and consistent inference parameters to ensure reproducibility and comparability across conditions. Table 3 summarizes the impact of BLE packet loss and specific disconnection scenarios on classification performance, measured as accuracy and F1-score averaged across 10 held-out test subjects.

**Table 3.** Performance of the distributed model under BLE packet loss and critical sensor disconnections. (Sensor disconnection was modeled by zeroing out the corresponding probability vectors without retraining the central model.)

Scenario	Accuracy Mean (%)	F1-Score Mean (%)
Baseline (No loss)	95.924	95.83
BLE Packet Loss 10%	95.023	94.92
BLE Packet Loss 30%	91.913	91.78
1 Sensor Disconnected (Chest)	92.907	92.78
2 Sensors Disconnected (Chest and Left Hand)	84.772	84.43

While all disconnection scenarios negatively impacted performance, results show that the distributed model remained robust under moderate packet loss (10%) and some single-sensor disconnections. However, removing the Chest sensor or combinations involving the Chest and Left Hand caused significant degradation, suggesting these sensors contribute highly informative features.

To further analyze this, we conducted a complete evaluation of all one-sensor and two-sensor disconnection combinations, shown in Table 4. The values represent average metrics across the 10 test subjects. Results show a clear trend where accuracy and F1-score vary depending on which specific sensors are dropped, confirming their relative importance to the central model.

**Table 4.** Classification performance of the distributed model under all combinations of one or two sensor disconnections. Highlighted rows indicate the worst-performing scenarios per group.

Scenario	Sensor	Accuracy Mean (%)	F1-Score Mean (%)
One-Sensor Disconnection	Chest	92.907	92.78
	Left Knee	94.386	94.11
	Right Hand	94.835	94.68
	Left Hand	92.554	92.36
	Right Knee	95.908	95.87
Two-Sensor Disconnection	Chest and Left Knee	88.339	88.02
	Chest and Right Hand	90.751	90.35
	Chest and Left Hand	84.772	84.43
	Chest and Right Knee	92.487	92.37
	Right Hand and Left Knee	93.327	92.97
	Right Hand and Left Hand	89.764	89.48
	Right Hand and Right Knee	95.165	95.12
	Left Hand and Left Knee	90.188	89.86
	Left Hand and Right Knee	93.148	93.08
	Right Knee and Left Knee	92.654	92.51

As can be observed, the Chest and Left-Hand combination yields the lowest overall performance, underscoring the collective influence of these two elements. In contrast, disconnections involving only peripheral joints (e.g., Right Knee or Left Knee) have a minimal effect on the central model's prediction, demonstrating fault tolerance in those cases.

These experiments were designed to be fully reproducible and can be replicated. The robust results emphasize the advantage of the distributed architecture in maintaining reliable inference even in the presence of communication failures.

#### 4.5. Latency and Energy Estimation

Beyond accuracy, latency and energy consumption are critical factors when evaluating the feasibility of activity recognition systems in real-time wearable scenarios. In this subsection, we report and compare the inference latency, energy consumption, power usage, and estimated battery life of both the centralized and distributed approaches.

All measurements were performed on an NVIDIA Jetson AGX Xavier (NVIDIA Corporation, Santa Clara, CA, USA) using TensorRT for model optimization and deployment. The Jetson platform was selected as a high-performance embedded system capable of profiling inference workloads, allowing relative comparison between centralized and distributed architectures. However, these results serve as a proxy and do not directly represent the on-device performance of ultra-low-power sensor nodes such as the MetaMotionRL (MbientLab Inc., San Jose, CA, USA).

Inference latency was recorded as the average time required to process a single input window and generate a prediction. For the centralized model, latency includes the entire processing pipeline of the fused multi-sensor input. In contrast, the distributed model performs independent local inferences at each node, followed by a lightweight aggregation step at the central unit. This modular and parallel structure led to significantly lower latency in the distributed setup.

Energy consumption per inference was estimated using TensorRT profiling tools, which account for both computational operations (FLOPs) and memory access patterns. The centralized model, due to its more complex architecture and multi-sensor integration, exhibited substantially higher energy consumption. The distributed framework, composed

of lightweight CNN–LSTM local models and a compact MLP fusion module, consumed considerably less energy per inference, a key advantage for deployment on low-power embedded platforms.

To further quantify performance, we computed the power consumption (W) of each model using the following expression:

$$\text{Power} = \text{Energy per inference (J)} \times \text{Inference rate (Hz)} \quad (1)$$

Assuming an inference rate of 50 predictions per second, consistent with the 50 Hz sampling rate commonly used in human activity recognition. This assumption enables extrapolation of average power usage. It is essential to note that the energy consumption of BLE communication was not included in this estimation, as accurate BLE transmission profiling requires direct measurement at the hardware level, which was beyond the scope of this study.

To estimate battery life, we used the specifications of the MetaMotionRL sensor (Mbi-enLab Inc., San Jose, CA, USA), which includes a 190 mAh battery operating at 3.7 V, yielding a total energy capacity of:

$$E_{\text{battery}} = 190 \text{ mAh} \times 3.7 \text{ V} = 703 \text{ mWh} = 2530.8 \text{ J} \quad (2)$$

Using this energy capacity and the estimated power consumption for each model, the expected battery life was computed as:

$$\text{Battery Life(s)} = \frac{E_{\text{battery}}}{\text{Power consumption}} \quad (3)$$

For the distributed model, the total energy per inference was computed as the sum of the energy usage from all five local models (processing quaternion, accelerometer, and gyroscope data), plus the energy usage of the central model. Total latency was calculated assuming parallel local inference, using:

$$\text{Latency\_total} = \max(\text{Latency}_{\text{localmodels}}) + \text{Latency}_{\text{centralmodel}} \quad (4)$$

Table 5 provides a comprehensive summary of these performance efficiency metrics. The distributed model achieved a total inference time of 0.643 ms and energy usage of 3.91 mJ, compared to 3.184 ms and 9.219 mJ for the centralized model. These improvements are primarily attributed to the distributed execution of lightweight local models and the minimal overhead of the central fusion module.

**Table 5.** Comparison of latency, energy consumption, power usage, and estimated battery life between the centralized and distributed models.

Attribute	Centralized	Distributed	
		Local Node	Central Node
Total Latency [ms]	3.184	0.6065	0.0366
Energy Per Inference [mJ]	9.219	0.6472	0.0279
Power Consumption [W]	0.46095		0.19555 (Total)
Battery Life [approx.]	1.52 h		3.59 h

This efficiency translates directly into extended operational autonomy. While the centralized model supports approximately 1.52 h of continuous operation per full battery charge, the distributed framework enables an estimated runtime of roughly 3.6 h per sensor. These values represent theoretical estimates based on Jetson-level profiling and

battery specifications and should be interpreted as relative indicators rather than absolute performance guarantees.

These results highlight the practical advantages of the distributed architecture in scenarios where energy efficiency, low-latency inference, and prolonged operation are critical. By distributing the computational load across local nodes and reducing the complexity at each sensor, the system enables extended usage while maintaining competitive classification performance.

The experimental evaluation confirms that the distributed approach consistently outperformed the centralized model across all evaluated metrics, including accuracy, precision, recall, and F1-score. While the centralized model benefits from fused multi-sensor input, the distributed framework demonstrated superior performance, particularly in activities involving subtle or overlapping motion patterns. This suggests that local models, when coordinated adequately through an efficient fusion strategy, can effectively capture discriminative features while also offering additional advantages in modularity and scalability.

Beyond predictive performance, the efficiency gains achieved by the distributed model are substantial. As shown in Table 5, total inference latency decreased from 3.18 ms in the centralized model to just 0.64 ms in the distributed setup. Similarly, energy per inference was reduced from 9.22 mJ to 0.68 mJ when combining local and central components. These reductions led to a significantly lower power consumption (0.195 W vs. 0.461 W), which in turn resulted in a more than twofold increase in estimated battery life, from approximately 1.5 h in the centralized model to nearly 3.6 h in the distributed case.

These findings confirm that the distributed architecture is better suited for real-time, embedded deployments where energy autonomy and responsiveness are critical constraints. The balance between performance and efficiency offered by this framework lays a solid foundation for future work on scalable, robust, and context-aware activity recognition systems in real-world wearable applications.

Finally, we acknowledge that the current energy and latency estimates are derived from a high-performance embedded proxy (Jetson AGX Xavier), and do not include the energy overhead of BLE communication. Future work will focus on empirical energy profiling directly on ultra-low-power sensor nodes, incorporating BLE transmission cost and inference latency at the microcontroller level for full-system evaluation.

#### 4.6. Implementation Challenges on Embedded Platforms

While this study focused on evaluating the proposed framework through offline simulations, transitioning the system to real-time embedded platforms introduces several practical challenges. First, BLE communication latency and packet loss must be carefully considered, as real-world transmission delays and asynchronous sensor updates can lead to temporary misalignments between sensor inputs. In our framework, such effects were simulated using zero-vector substitution during the inference process. Still, real-time handling may require buffering strategies, synchronization protocols, or lightweight extrapolation methods to maintain consistency across sensor nodes.

Second, the computational capabilities of embedded processors, such as microcontroller units (MCUs) or edge AI devices (e.g., Jetson Nano), can limit the complexity of local neural networks. While our CNN-LSTM models are designed to be lightweight, optimizing them further via model quantization, pruning, or hardware-specific inference engines (e.g., TensorRT, ONNX Runtime) would be necessary to meet real-time constraints.

Additionally, power consumption trade-offs must be managed carefully. Frequent wireless communication and local processing increase energy usage, which is critical in wearable applications. Efficient scheduling of inference and transmission tasks, possibly

driven by event-based triggering or adaptive sampling, will be essential for long-term operation.

Finally, synchronization among sensors becomes a non-trivial issue in multi-node systems. Without centralized clocking, drift can occur between nodes, resulting in temporal alignment issues. Implementing timestamp correction or leveraging protocols such as BLE time synchronization would be required.

Although real-time deployment was outside the scope of this work, future research will address these aspects through implementation on embedded prototypes with actual BLE communication links.

## 5. Discussion

The results presented in the previous section highlight the strengths and trade-offs between the centralized and distributed approaches for activity recognition using wearable sensors. This discussion builds on those findings by examining their practical implications, acknowledging limitations, and outlining directions for future research.

We acknowledge that no external motion capture system was used to validate the absolute accuracy of the Inertial Measurement Unit (IMU) orientation estimates. Instead, we relied on the onboard sensor fusion algorithms provided by the MetaMotionRL sensors, which integrate accelerometer, gyroscope, and magnetometer readings to produce stable orientation outputs in short recording windows. Since the activities were performed under controlled conditions and were of limited duration (~5 min per activity), the accumulated drift was negligible for activity classification. Furthermore, the model's generalization capability was evaluated across a diverse group of 67 participants, yielding consistent performance results. While validation with an external motion capture system would provide higher-fidelity ground truth for biomechanical analysis, it was not required for the goals of this study, which focus on robust and efficient activity recognition in practical wearable settings.

The centralized model, which processes fused data from all sensors simultaneously, achieved strong classification results. This aligns with previous findings, which show that centralized architectures benefit from access to multi-sensor input, enabling the capture of inter-sensor dependencies. However, the distributed model exhibited consistently higher precision across all activity classes, reflecting a more conservative prediction strategy with fewer false positives. This behavior is particularly advantageous in contexts where misclassifications could lead to undesired actions or safety risks.

Beyond classification performance, the distributed framework offers significant advantages in computational efficiency. By assigning local models to individual sensors and transmitting only their SoftMax outputs, the architecture reduces per-node processing requirements and enables parallel inference with lower latency. When deployed on the NVIDIA Jetson AGX Xavier using TensorRT, the distributed model achieved an order-of-magnitude improvement in both inference time and energy consumption compared to the centralized configuration. These gains translate into extended operational autonomy, with an estimated runtime increase from just over 1.52 h in the centralized case to approximately 3.6 h per sensor in the distributed configuration. This improvement makes the system more suitable for continuous applications such as physical therapy monitoring, occupational safety, and long-term activity tracking.

Although battery life was estimated and compared between configurations, we did not conduct experiments to assess how sensor performance or data quality may degrade as battery levels drop. Recording sessions in this study lasted approximately 37 min, well within the sensors' full runtime, and no perceptible degradation in signal quality, stability, or transmission was observed during that period. However, we acknowledge that

low battery conditions in extended deployments could potentially impact sampling rates, sensor accuracy, or communication reliability. Future work will investigate these aspects in the context of long-term, real-time operation.

It is essential to note that the distributed inference framework was evaluated using offline simulations, where data were collected and synchronized in advance under controlled conditions. This decision allowed us to focus on validating the inference architecture itself, specifically, the ability to classify activities based on partial, independently processed sensor inputs, without introducing variability from wireless communication or real-time processing constraints.

Nonetheless, we recognize that real-time implementation aspects such as BLE transmission delays, packet loss, and power consumption are critical for practical deployment and were not evaluated in this study. To address this, we have outlined a concrete plan for real-time implementation. Specifically, we intend to integrate BLE communication between each wearable sensor (MCU) and a central embedded hub (e.g., NVIDIA Jetson Nano or Xavier), using concurrent Bluetooth interfaces to emulate realistic operating conditions. During this phase, we will measure key transmission parameters, including:

- Total system delay, from sensing to central decision output (end-to-end latency);
- Packet loss rate, both random and due to temporary disconnections;
- Transmission power and energy consumption, measured per sensor over extended sessions.

Additionally, we aim to characterize the temporal behavior of the system as battery levels decrease, to analyze possible degradation in sampling rate, latency, or BLE stability over time. This implementation phase will also include the integration of error-handling mechanisms, predictive smoothing, and dropout-tolerant inference to mitigate the effects of intermittent data streams.

We also aim to characterize system behavior as battery levels decline, and implement error-handling mechanisms, predictive smoothing, and dropout-tolerant inference to mitigate instability in the data stream. This phase will involve a new round of data collection with an expanded participant cohort under dynamic and unconstrained conditions. Together, these efforts will bridge the gap between offline validation and real-time deployment, enabling a more comprehensive assessment of the system's viability in embedded and mobile contexts.

The distributed design additionally provides flexibility and robustness. Each local model can operate independently, allowing for partial inference even when one or more sensors become disconnected. This capability improves fault tolerance and supports modular scalability, critical features in real-world environments with unpredictable conditions. These aspects, along with reduced communication overhead, further reinforce the suitability of distributed learning systems for wearable applications with resource constraints.

Despite these advantages, the distributed framework presents certain limitations. Local models may lack the global context available in centralized architectures, which can impact classification performance in activities that involve inter-limb coordination. Furthermore, managing asynchronous data streams and synchronizing predictions across multiple nodes introduces architectural complexity, particularly in lossy or bandwidth-limited communication environments.

Another important consideration is the absence of explicit spatial calibration between sensors. In this study, each local model leveraged only its motion data, which was sufficient for the selected activities. However, tasks that require precise relative positioning, such as gait symmetry analysis, sign language interpretation, or specific rehabilitation protocols, may benefit from sensor alignment in a standard spatial frame. As the focus here was to

demonstrate a minimally coordinated distributed system, alternative spatial configurations and calibration strategies were not explored.

Lastly, it is essential to distinguish the proposed HAR framework from full-body biomechanical systems such as Xsens. These systems aim to reconstruct body pose using anatomically anchored sensor placements and high-precision inertial data, typically requiring extensive spatial calibration and personalized models. In contrast, our system focuses on activity classification using independently captured local motion features, without reconstructing global posture. This design supports computational efficiency, modular deployment, and ease of use in practical applications where full pose estimation is unnecessary or impractical.

In summary, while the centralized model offers high overall accuracy and more straightforward integration, the distributed approach provides superior precision, energy efficiency, and robustness. These qualities make it a strong candidate for real-time embedded activity recognition. Table 6 summarizes the key trade-offs discussed in this section and supports informed system design decisions based on specific deployment constraints.

**Table 6.** Summary of trade-offs between centralized and distributed.

Criterion	Centralized Model	Distributed Model
Overall Accuracy	High	Higher
Precision	Lower (more false positives)	Higher across all classes
Recall	High	Higher in some classes
Latency (Inference)	High (sequential processing)	Low (parallel local inference)
Energy Consumption	High	Low
Estimated Battery Life	1.52 h	3.6 h
Computational Load	Centralized, resource-intensive	Distributed across nodes, lightweight
Scalability	Limited (single point of processing)	High (modular, sensor-level models)
Fault Tolerance	Low (dependent on all sensor streams)	High (can tolerate partial sensor loss)
Implementation	Simpler to manage globally	More complex synchronization, but modular

Future work will focus on exploring architectures that adaptively combine centralized and distributed inference based on contextual factors and energy constraints. Promising research directions include developing methods to dynamically weight sensor contributions, incorporating attention mechanisms at the fusion stage, and enabling online learning directly at the edge. While the current evaluation was conducted in a controlled environment with a predefined activity protocol to ensure synchronization and label consistency, one of the primary motivations behind the proposed distributed framework is its potential scalability to real-world, dynamic scenarios. Accordingly, future efforts will include validating the model under unscripted, noisy conditions and free-form activity sessions to assess its robustness and generalizability. Moreover, extending evaluation to multi-user scenarios, outdoor environments, and intermittent connectivity will be crucial for advancing the deployment readiness of these systems. Finally, implementing and validating the proposed framework in real-world, real-time conditions beyond offline simulation remains an essential step toward practical application.

**Author Contributions:** Software, J.L.R.-C., K.N.-T., L.S.-A. and J.F.P.-M.; Investigation, J.L.R.-C., K.N.-T., L.S.-A. and J.F.P.-M.; Writing—review & editing, J.L.R.-C., K.N.-T., L.S.-A. and J.F.P.-M. All authors have read and agreed to the published version of the manuscript.

**Funding:** This research received financial support from the NSF EPSCoR Center for the Advancement of Wearable Technologies (CAWT) under Grant No. OIA-1849243.

**Institutional Review Board Statement:** The study was conducted under the Declaration of Helsinki and approved by the University of Puerto Rico's Institutional Review Board (IRB) under protocol number 2024120022 (approved on 27 January 2025).

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

**Data Availability Statement:** The dataset used in this study is openly available in Zenodo at <https://doi.org/10.5281/zenodo.15830858>.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Abbreviations

The following abbreviations are used in this manuscript:

CNN	Convolutional Neural Network
RNN	Recurrent Neural Network
ML	Machine Learning
FL	Federated Learning
FSL	Federated Split Learning
SL	Split Learning
MFFL	Multi-Frequency Federated Learning
SNN	Spiking Neural Network
BLE	Bluetooth Low Energy
LSTM	Long-Short Term Memory
MLP	Multi-Layer Perceptron
FLOPs	Floating Point Operations
MCUs	Microcontroller Units
IMU	Inertial Measurement Unit

## References

1. Baghezza, R.; Bouchard, K.; Bouzouane, A.; Gouin-Vallerand, C. From Offline to Real-Time Distributed Activity Recognition in Wireless Sensor Networks for Healthcare: A Review. *Sensors* **2021**, *21*, 2786. [[CrossRef](#)]
2. El-Adawi, E.; Essa, E.; Handosa, M.; Elmoogy, S. Wireless Body Area Sensor Networks Based Human Activity Recognition Using Deep Learning. *Sci. Rep.* **2024**, *14*, 2702. [[CrossRef](#)] [[PubMed](#)]
3. Agarwal, P.; Alam, M. A Lightweight Deep Learning Model for Human Activity Recognition on Edge Devices. *Procedia Comput. Sci.* **2020**, *167*, 2364–2373. [[CrossRef](#)]
4. 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](#)]
5. Saldana-Aristizabal, L.; Nino-Tejada, K.; Rivas-Caicedo, J.L.; Patarroyo-Montenegro, J.F. Evaluating Quaternion-Based Representations for Human Activity Recognition Using Motion Capture. In Proceedings of the International Symposium on Intelligent Computing and Networking (ISICN 2025), San Juan, PR, USA, 17–19 March 2025; Springer: Berlin/Heidelberg, Germany, 2025.
6. Rivas-Caicedo, J.L.; Saldana-Aristizabal, L.; Nino-Tejada, K.; Patarroyo-Montenegro, J.F. Mitigating Communication Failures in Multi-Sensor Wearable Systems: Extrapolation Methods for LSTM-Based Posture Classification. In Proceedings of the International Symposium on Intelligent Computing and Networking (ISICN 2025), San Juan, PR, USA, 17–19 March 2025; Springer: Berlin/Heidelberg, Germany, 2025.
7. Agarwal, S.; Alam, M. A Lightweight Deep Learning Model for Human Activity Recognition on Edge Devices. *arXiv* **2019**, arXiv:1909.12917. [[CrossRef](#)]
8. Chen, Z.; Zhu, Q.; Soh, Y.C.; Zhang, L. Robust Human Activity Recognition Using Smartphone Sensors via CT-PCA and Online SVM. *IEEE Trans. Ind. Inform.* **2017**, *13*, 3070–3080. [[CrossRef](#)]
9. Ord'óñez, F.J.; Roggen, D. Deep Convolutional and LSTM Recurrent Neural Networks for Multimodal Wearable Activity Recognition. *Sensors* **2016**, *16*, 115. [[CrossRef](#)] [[PubMed](#)]

10. Zeng, M.; Nguyen, L.T.; Yu, B.; Mengshoel, O.J.; Zhu, J.; Wu, P.; Zhang, J. Convolutional Neural Networks for Human Activity Recognition Using Mobile Sensors. In Proceedings of the 6th International Conference on Mobile Computing, Applications and Services (MobiCASE), Austin, TX, USA, 6–7 November 2014; pp. 197–205.
11. Haque, S.T.; Ni, J.; Li, J.; Yan, Y.; Ngu, A.H.H. LightHART: Lightweight Human Activity Recognition Transformer. In *Lecture Notes in Computer Science, Pattern Recognition*; Springer Nature: Cham, Switzerland, 2024; pp. 425–441. [CrossRef]
12. Yu, H.; Chen, Z.; Zhang, X.; Chen, X.; Zhuang, F.; Hiong, H.; Cheng, X. FedHAR: Semi-Supervised Online Learning for Personalized Federated Human Activity Recognition. *IEEE Trans. Mob. Comput.* **2021**, *22*, 3318–3332. [CrossRef]
13. Chaddad, A.; Wu, Y.; Desrosiers, C. Federated Learning for Healthcare Applications. *IEEE Internet Things J.* **2023**, *11*, 7339–7358. [CrossRef]
14. Dey, A.; Chakraborty, S.; Ray, R. FedOpenHAR: A Federated Learning Benchmark for Wearable Human Activity Recognition. *IEEE Internet Things J.* **2023**, *10*, 2163–2174.
15. Neftci, E.O.; Mostafa, H.; Zenke, F. Surrogate Gradient Learning in Spiking Neural Networks. *IEEE Signal Process. Mag.* **2019**, *36*, 61–63. [CrossRef]
16. Kalantarian, H.; Alshurafa, N.; Sarrafzadeh, M. Adaptive Activity Recognition with Crowdsourcing and Online Learning (AdARC). *IEEE J. Biomed. Health Inf.* **2017**, *21*, 1193–1201.
17. Sannara, P.; Maji, S.; Amintoosi, H. Edge-Distributed Sensor Fusion for Real-Time Human Activity Recognition. In Proceedings of the 21st IEEE Conference on Sensors, Dallas, TX, USA, 30 October–2 November 2022; IEEE: New York, NY, USA, 2022; pp. 1212–1217.
18. Sozinov, K.; Vlassov, V.; Girdzijauskas, S. Human Activity Recognition Using Federated Learning. In Proceedings of the 2018 IEEE International Conference on Parallel & Distributed Processing with Applications, Ubiquitous Computing & Communications, Big Data & Cloud Computing, Social Computing & Networking, Sustainable Computing & Communications (ISPA/IUCC/BDCloud/SocialCom/SustainCom), Melbourne, Australia, 11–13 December 2018; IEEE: New York, NY, USA, 2018; pp. 1103–1111.
19. Gad, G.; Fadlullah, Z.M.; Rabie, K.; Fouada, M.M. Communication-Efficient Privacy-Preserving Federated Learning via Knowledge Distillation for Human Activity Recognition Systems. In Proceedings of the IEEE International Conference on Communications, Rome, Italy, 28 May–1 June 2023; Institute of Electrical and Electronics Engineers Inc.: New York, NY, USA, 2023; pp. 1572–1578. [CrossRef]
20. Ndeko, J.; Shaon, S.; Beal, A.; Sahoo, A.; Nguyen, D.C. Federated Split Learning for Human Activity Recognition with Differential Privacy. In Proceedings of the IEEE Consumer Communications and Networking Conference, CCNC, Las Vegas, NV, USA, 10–13 January 2025; Institute of Electrical and Electronics Engineers Inc.: New Jersey, NJ, USA, 2025.
21. Fenoglio, D.; Li, M.; Casnici, D.; Laporte, M.; Gashi, S.; Santini, S.; Gjoreski, M.; Langheinrich, M. Multi-Frequency Federated Learning for Human Activity Recognition Using Head-Worn Sensors. In Proceedings of the 2024 International Conference on Intelligent Environments (IE), Ljubljana, Slovenia, 17–20 June 2024; pp. 17–24. [CrossRef]
22. Thakur, D.; Guzzo, A.; Fortino, G. Energy Aware Federated Learning with Application of Activity Recognition. In Proceedings of the 2023 IEEE International Conference on Dependable, Autonomic and Secure Computing, International Conference on Pervasive Intelligence and Computing, International Conference on Cloud and Big Data Computing, International Conference on Cyber Science and Technology Congress (DASC/PiCom/CBDCom/CyberSciTech), Abu Dhabi, United Arab Emirates, 14–17 November 2023; IEEE: New York, NY, USA, 2023; pp. 0636–0642.
23. Saylam, B.; Durmaz, O. Highlights Federated Learning on Edge Sensing Devices: A Review. *arXiv* **2023**, arXiv:2311.01201.
24. Khan, Q.W.; Khan, A.N.; Rizwan, A.; Ahmad, R.; Khan, S.; Kim, D.H. Decentralized Machine Learning Training: A Survey on Synchronization, Consolidation, and Topologies. *IEEE Access* **2023**, *11*, 68031–68050. [CrossRef]
25. Ding, G.; Tian, J.; Wu, J.; Zhao, Q.; Xie, L. Energy efficient human activity recognition using wearable sensors. In Proceedings of the 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW), Barcelona, Spain, 15–18 April 2018; IEEE: New York, NY, USA, 2018; ISBN 9781538611548.
26. Rezaie, H.; Ghassemian, M. An Adaptive Algorithm to Improve Energy Efficiency in Wearable Activity Recognition Systems. *IEEE Sens. J.* **2017**, *17*, 5315–5323. [CrossRef]
27. Yang, L.; Amin, O.; Shihada, B. Intelligent Wearable Systems: Opportunities and Challenges in Health and Sports. *ACM Comput. Surv.* **2024**, *56*, 1–42. [CrossRef]
28. Zmitri, M.; Fourati, H.; Vuillerme, N. Human Activities and Postures Recognition: From Inertial Measurements to Quaternion-Based Approaches. *Sensors* **2019**, *19*, 4058. [CrossRef] [PubMed]
29. Rivas, J.; Niño, K.; Saldaña, L.; Patarroyo, J. Wearable IMU Sensor Dataset for Human Activity Recognition (HAR). 2025. Available online: <https://zenodo.org/records/15769102> (accessed on 29 July 2025).
30. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980 2014.
31. Bengio, Y. Practical Recommendations for Gradient-Based Training of Deep Architectures. In *Neural Networks: Tricks of the Trade*; Montavon, G., Orr, G.B., Müller, K.-R., Eds.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 437–478.

32. Masters, D.; Luschi, C. Revisiting Small Batch Training for Deep Neural Networks. *arXiv* **2018**, arXiv:1804.07612.
33. Prechelt, L. Early Stopping—But When? In *Neural Networks: Tricks of the Trade*; Orr, G.B., Müller, K.-R., Eds.; Springer: Berlin/Heidelberg, Germany, 1998; pp. 55–69.

**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.