# MLP-HAR: Boosting Performance and Efficiency of HAR Models on Edge Devices with Purely Fully Connected Layers

Anonymous Author(s)

## ABSTRACT

Neural network models have demonstrated exceptional performance in wearable human activity recognition (HAR) tasks. However, the increasing size or complexity of HAR models significantly impacts their deployment on wearable devices with limited computational power. In this study, we introduce a novel HAR model architecture named Multi-Layer Perceptron-HAR (MLP-HAR), which contains solely fully connected layers. This model is specifically designed to address the unique characteristics of HAR tasks, such as multi-modality interaction and global temporal information. The MLP-HAR model employs fully connected layers that alternately operate along the modality and temporal dimensions, enabling multiple fusions of information across these dimensions. Our proposed model demonstrates comparable performance with other state-of-the-art HAR models on six open-source datasets, while utilizing significantly fewer learnable parameters and exhibiting lower model complexity. Specifically, the complexity of our model is at least ten times smaller than that of the TinyHAR model and several hundred times smaller than the benchmark model DeepConvLSTM. Additionally, due to its purely fully connected layer-based architecture, MLP-HAR offers the advantage of ease of deployment. To substantiate these claims, we report the inference time performance of MLP-HAR on the Samsung Galaxy Watch 5 PRO and the Arduino Portenta H7 LITE, comparing it against other state-of-the-art HAR models.

## CCS CONCEPTS

• **Human-centered computing → HCI theory, concepts and models**.

## KEYWORDS

wearable human activity recognition; deep learning; lightweight neural networks

**ACM Reference Format:**

Anonymous Author(s). 2018. MLP-HAR: Boosting Performance and Efficiency of HAR Models on Edge Devices with Purely Fully Connected Layers. In *Woodstock '18: ACM Symposium on Neural Gaze Detection, June 03–05, 2018, Woodstock, NY*. ACM, New York, NY, USA, 7 pages. https://doi.org/XXXXXXX.XXXXXXX



*Conference acronym 'XX, June 03–05, 2018, Woodstock, NY*



ACM ISBN 978-1-4503-XXXX-X/18/06
https://doi.org/XXXXXXX.XXXXXXX

## 1 INTRODUCTION

Wearable human activity recognition (HAR) has gained increasing attention due to its effectiveness in health monitoring, fitness tracking, and human-computer interaction [8]. Advances in model architectures have significantly improved performance on various HAR tasks [1]. However, many state-of-the-art (SOTA) models neglect the need for deployment on wearable devices with limited computational and energy resources, such as smartwatches [39]. Given the multi-modal nature of HAR data and the need to capture global temporal information, SOTA HAR models [1, 16, 19, 22, 25, 35, 38] often employ complex hybrid architectures, including convolutional layers (CNNs)[34], self-attention mechanisms[31], and recurrent neural networks (RNNs)[25]. Figure 2 displays the inference times of several SOTA models when deployed on a Samsung Galaxy Watch 5 PRO, showing that the inference speeds of some HAR models remain sub-optimal.

The reasons for their long inference times are threefold: (1). Many models employ CNNs as feature extractors. However, due to the limited receptive field of CNN layers, multiple layers are stacked to capture global information, significantly increasing computational complexity [39]. (2). Variants of RNNs, such as Gated Recurrent Units (GRU) [29], Long Short-Term Memory (LSTM) [25], and bidirectional-LSTM [37], are commonly used. Their recurrent paradigm inherently hinders parallel computing, introducing slower latency. (3) Many studies incorporate self-attention mechanisms [31], which involve multi-branch designs and require storing attention matrices. These structures compromise memory utilization efficiency as they necessitate storing the outputs of each branch until combined through addition or concatenation [12], substantially increasing memory access costs and reducing inference speed.

Lightweight models are characterized by fewer trainable parameters, reduced computational demands, and a lower memory footprint. Consequently, lightweight models run faster and provide real-time feedback on edge devices. In this work, we introduce a lightweight HAR model with a plain topology, termed Multi-Layer Perceptrons for HAR (MLP-HAR). It relies solely on fully connected (FC) layers, without other operators or complex multi-branch structures. FC layers exhibit the highest computational efficiency because they leverage highly optimized matrix multiplication operations. The proposed model is designed with the characteristics of HAR tasks in mind. It employs FC layers to alternately operate along the modality and temporal dimensions, enabling effective multi-dimensional information fusion. In summary, the primary contributions of this work are:

- We propose the MLP-HAR model utilizing purely FC layers with a plain topology, making it deployment-friendly and memory-efficient.
- Extensive evaluation on six datasets shows that MLP-HAR's performance is comparable to SOTA models, but has significantly fewer learnable parameters and less model complexity.

- The MLP-HAR model's efficiency is further validated through performance assessments on the Samsung Galaxy Watch 5 PRO and the more resource-constrained Arduino Portenta H7 LITE. Results indicate that MLP-HAR offers improved inference times compared to other SOTA models.

## 2 RELATED WORK

CNNs [21, 34] have been foundational in HAR tasks due to their ability to extract local context information from time series data. However, the limited receptive fields of CNNs restrict their ability to effectively process long temporal information. This limitation has led to the adoption of RNN-based models [10, 29, 37], which excel in capturing global temporal dependencies. To capitalize on the strengths of both CNNs and RNNs, hybrid architectures [23, 25, 38] have been developed. For example, DeepConvLSTM [25] combines four convolutional layers with two LSTM layers, the former is used for local feature extraction and multi-modal fusion, while the latter extracts of global temporal dependencies. To further improve information extraction across sensor channels and time steps, self-attention mechanisms [31] have been integrated into HAR models. SOTA models like DeepConvLSTM-Attn [22], Attend [1], and ALAE-TAE [2] augment CNN-RNN frameworks with various attention designs to refine feature extraction. Additionally, some HAR models transform data inputs into frequency representations to enhance feature extraction. For instance, DeepSense [35] and GlobalFusion [16] use spectrogram inputs obtained through FFT Transform. Both models employ a CNN-RNN-attention-based hybrid architectures. However, these focus on maximizing accuracy often overlooks the essential aspects of deployment efficiency and real-time processing capabilities.

There are efforts have been made so far to encourage more lightweight HAR models. For instance, TinyHAR [40] introduces design principles tailored to HAR tasks, selecting appropriate modules for robust feature extraction while keeping the model size small and complexity low. Nonetheless, it still incorporates a combination of CNN, RNN, and self-attention modules, which are memory consuming and lack parallelization capabilities. Coelho *et al.*[11] focused on optimizing architectural parameters in purely CNN-based models. Ma *et al.*[18] trained multiple weak CNN-based models, with a model selector flexibly choosing the best classifier based on the input sample during deployment. However, the inherent limitations of purely CNN-based architectures have been demonstrated to hinder their performance on more complex tasks.

From a hardware perspective, Multiply-Accumulate (MAC)-based operations such as FC layers and CNN layers can be highly accelerated and parallelized through, e.g., crossbar-based AI acceleration architecture. Motivated by this insight, we aim to develop a model with a plain topology consisted entirely of FC layers. This model is intended to deliver SOTA performance while simultaneously reducing model size and complexity, and achieving faster inference times. Purely FC network architectures have proven effective in other domains, such as vision [30] and time series forecasting [9]. However, they typically neither address the specific characteristics of HAR tasks nor focus on minimizing model complexity and optimizing inference time for deployment. While MLP [24], a model composed purely of FC layers, had already been proposed to tackle HAR tasks, it naively treats the input time series as an image and completely ignores the multi-modality characteristic of HAR data and the differences between the temporal and sensor channel dimensions, which are significantly different from the relationship between "height" and "width" of images.

## 3 METHODOLOGY

In this section, we elucidate the proposed model architecture, illustrated in Figure 1, which comprises three main modules: the Data Embedding Module, the Information Mixing Module, and the Prediction Module. The Data Embedding Module extracts local temporal features from raw data. The Information Mixing Module employs a repeated alternating structure to facilitate information exchange and integration across temporal and sensor channel dimensions. Finally, the Prediction Module condenses the extracted features to make the final prediction. All modules consist solely of FC layers.

### 3.1 Data Embedding Module

In this module, the raw input segment data $\mathbf{X} \in \mathbb{R}^{L \times C}$ (where $L$ is the sliding window size and $C$ is the number of sensor channels), is split into intervals of length $\tau$, resulting in $T$ intervals, represented as $\mathbf{X_t} \in \mathbb{R}^{\tau \times T \times C}$. Each interval undergoes an FFT transformation to extract frequency domain representations, producing $\mathbf{X}_f \in \mathbb{R}^{2f \times T \times C}$, with $f$ representing the frequency magnitudes and phase pairs. The benefits of this process are twofold: (1) Interval segmentation reduces the raw data's temporal length, lowering computational demands in subsequent steps. (2) Frequency features are crucial for differentiating human activities, as signals acquired through wearable sensors often exhibit multi-frequency characteristics [17]. Extracting frequency information directly via FFT simplifies the model's task of frequency feature extraction [33]. Both the time and frequency representations are then processed through separate FC layers, each followed by layer normalization [3] and a ReLU activation. The output size of the FC layers is denoted as $d$. The feature maps extracted from both representations are concatenated to form a combined feature set with dimensions $2d \times T \times C$, which is then further fused by an additional FC layer to maintain consistent dimensions.

### 3.2 Mixer Module

After embedding the local information from each sensor channel, the Mixing Module fuses this information across both temporal and sensor channel dimensions. As shown in Figure 1(c), this module consists of two components: the temporal mixing block and the modality mixing block.

### 3.3 Temporal Mixing Block

The temporal mixing block fuses information within each sensor channel along the temporal dimension. It begins by flattening the feature map from $2d \times T \times C$ to $2dT \times C$, followed by layer normalization to standardize features within each channel. The feature map then passes through two FC layers. To reduce the parameter count, the first FC layer decreases the feature size using a shrink ratio $\sigma$, while the second FC layer restores it to $2d$. A ReLU function is applied after the first FC layer as activation. These two FC layers

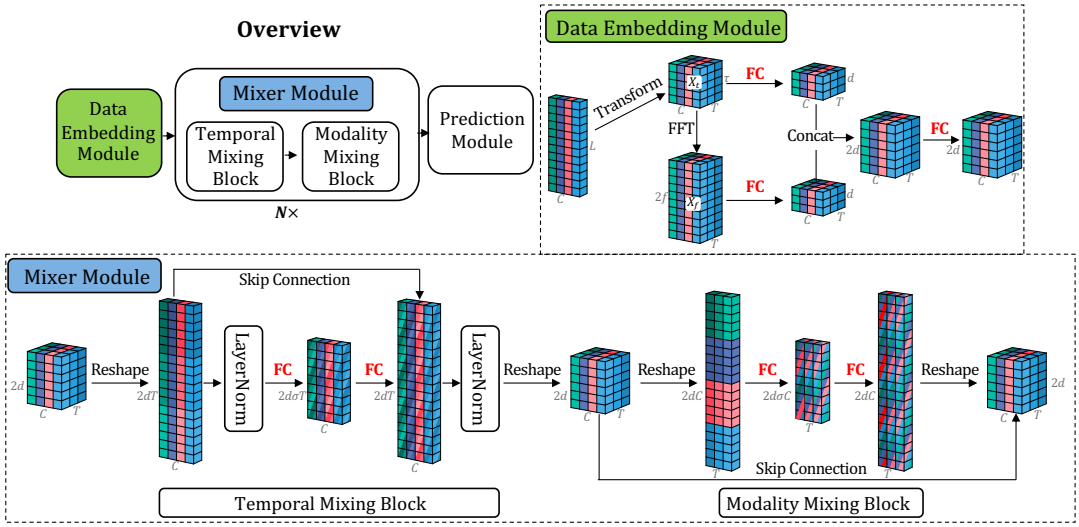


**Figure 1: The upper left part of this figure (a) shows an overview of the proposed structure. The upper right part of the figure (b) shows the specific structure of the data embedding module. The lower part of the figure (c) shows the specific structure of the mixer module. Different colors in the figure represent readings of different sensor channels, and different shades of the same color represent different intervals. Mixed color shows fused information from FC layer.**

enable efficient feature exchange across different temporal intervals. A skip connection reintegrates the fused features back into the original feature space, allowing each interval to incorporate information from others. The process concludes with another layer normalization before reshaping the data back to $2d \times T \times C$.

*3.3.1 Modality Mixing Block.* The Modality Mixing Block enables the exchange and fusion of information between intervals at the same time step but in different modalities. It omits layer normalization since it was applied in the temporal mixing block. Initially, the features are flattened based on the modality dimensions, transforming the feature map from $2d \times T \times C$ to $2dC \times T$. Subsequently, two FC layers facilitate interactions among features of different modalities. Finally, the fused features across modalities are reintegrated into the original feature set via a skip connection.

*3.3.2 Discussion.* This design enables rapid information propagation across different time steps and modalities. Unlike traditional HAR models, which follow a fixed sequence of feature extraction—first channel interaction, then temporal information—our model adopts a more flexible approach. By stacking this mixer model $N$ times, the iterative process allows multiple integrations of information across both dimensions for each interval, enhancing the model's ability to extract powerful features.

## 3.4 Prediction Module

This module includes two FC layers. The first FC layer fuses the features from each sensor channel into a single vector. The second FC layer then uses these fused vectors from all sensor channels to predict the final activity.

| Dataset | *#Class* | *Sensor* | *C* | *#SW* | *# T* | *# τ* | *# f* |
|---|---|---|---|---|---|---|---|
| PAMAP2 [27] | 18 | Acc,Gyro | 18 | 1.28 s | 8 | 16 | 16 |
| HAPT [28] | 12 | Acc,Gyro | 6 | 2.56 s | 8 | 16 | 16 |
| DSADS [6] | 19 | Acc,Gyro | 30 | 5 s | 5 | 25 | 25 |
| Daphnet [4] | 2 | Acc | 9 | 1 s | 4 | 16 | 16 |
| MotionSense [20] | 6 | Acc,Gyro | 12 | 2.56 s | 8 | 16 | 16 |
| Mhealth [5] | 6 | Acc,Gyro | 12 | 2.56 s | 8 | 16 | 16 |

**Table 1: Within the table, the column labeled *#Class* denotes the number of activity types, while *Sensor* specifies the sensor type used, with *Acc* for accelerometer and *Gyro* for gyroscope. The column *C* indicates the number of the sensor channels. The column *#SW* denotes the size of the sliding window.**

# 4 EXPERIMENTS AND DISCUSSIONS

## 4.1 Experiment Setup

*4.1.1 Datasets.* To validate our model's performance, we conducted experiments using six open-source datasets. A detailed summary of each dataset, along with data preparation and transformation parameters, is provided in Table 1. Sensor signals are first z-normalized and then segmented by sliding window. For some models, including ours, the segmented raw data undergoes an FFT transformation to generate the spectrogram. The parameters for this transformation, such as the number of intervals ($T$), interval length ($\tau$), and the quantity of amplitude and phase spectral pairs ($f$), are also listed in Table 1. The FFT transformation is implemented directly within the model using the torch.fft function, and the model only accepts raw time series data. In subsequent evaluations, the time and complexity of the FFT transformation are included in the measurement of inference time and model complexity.

*4.1.2 Compared Models.* In this experiment, we evaluated our approach against ten comparative models. Below is a brief introduction to these models:

**DCNN** [34]: A purely CNN-based HAR model. **DeepConvLSTM** [25] (DCL): A benchmark model with CNN and LSTM layers. We implemented the version from [7] with larger CNN kernel sizes (5 to 11) and reduced LSTM layers (from two to one). **DeepConvLSTM-Attention** [22] (DCL-Attn): Enhances the DCL model by adding a self-attention module after the LSTM layer. **Attend-Discriminate** (Attend) [1]: Advances the CNN-GRU architecture by using a self-attention mechanism to learn interactions between channels at each time step. **IF-ConvTransformer** [36] (IF-ConvT): Extends the Attend model by introducing a CNN block at the beginning to serve as a complementary filter and replacing the GRU layer with a self-attention module. **ALAE-TAE** [2]: Evolves from the DCL model by incorporating an attention encoder between the CNN and LSTM layers, using the squeeze and excitation technique [14] to enhance feature interrelationships. **DeepSense** [35]: Applies FFT transformation to input data, then processes it with a hybrid multi-branch CNN-GRU architecture. **GlobalFusion** [16]: Builds on DeepSense by adding two global self-attention modules to efficiently fuse features from various locations and sensor modalities. **TinyHAR** [40]: A lightweight HAR model. **MLP** [24]: Treats input time series as an image and applies the MLP architecture for vision [30] directly to the HAR task. It is important to note that, except for the MLP model [24], the configurations of all the aforementioned models strictly adhere to their descriptions and source code as presented in the referenced literature. The MLP model's configuration in the original work [24] is much larger than the other comparison models and cannot run on the watch due to exceeding the watch's memory. The specifications used for the MLP model in this experiment are: 5 layers, a patch-embedding size of 256, a patch dimension of 64, and a channel dimension of 256. For our proposed MLP-HAR model, we fix the number of mixer modules $N$ at 2 and the filter number $d$ at 6.

*4.1.3 Training & Evaluation Protocol.* To train the model weights, we employ the Adam optimizer [15] with default settings, starting with an initial learning rate of $10^{-4}$. The learning rate decays by a factor of 0.9 after every 7 epochs without improvement. Training is capped at a maximum of 200 epochs, with early stopping if the validation loss does not improve for 15 consecutive epochs. The batch size is fixed at 256. All models are implemented using the PyTorch framework [26] and trained on an NVIDIA A100 GPU.

During the evaluation phase, we assess the performance of all models using the Leave-One-Subject-Out (LOSO) Cross-Validation (CV) strategy. In each CV iteration, data from one subject serve as the test set, while data from other subjects form the training and validation sets, maintaining a training-to-validation ratio of 9:1. The classification performance is quantified using the macro F1-score. This LOSO-CV process is repeated five times with random seeds from 1 to 5, and we report the mean and variance of the F1-scores obtained. Additionally, we report the model size in terms of the number of trainable parameters (in thousands) and the computational complexity in Million MACs (MMACs). The models are also deployed on a Samsung Galaxy Watch 5 PRO[1]. We report the average inference time for each model to process 10,000 input samples on the watch.

[1]For deployment, we used post-training int8 quantization from TensorFlow and the inference framework is TensorFlow Lite Micro.

| IF-ConvT | ALAE-TAE | Attend | MLP-HAR | TinyHAR | GlobalFusion |
|---|---|---|---|---|---|
| 1.83 | 3.17 | 3.33 | 3.33 | 5.67 | 6.00 |
| MLP | DeepSense | DCL-Attn | DCL | DCNN | |
| 6.17 | 7.33 | 8.83 | 9.33 | 11.00 | |

**Table 2: Average ranking of all models across six datasets. The smaller the rank, the better the performance.**

## 4.2 Comparison to State-of-the-art

Figure 2 presents the performance of all models across six datasets, visualizing the averaged macro F1 score, model complexity, model size, and inference time for each dataset in column blocks. Since our model and TinyHAR exhibit significantly lower computational complexity and fewer learnable parameters compared to the other models, the y-axes for model complexity and size are logarithmically scaled with a base of 10.

Our proposed MLP-HAR model achieves superior performance on the DSADS dataset and comparable performance on the PAMAP2, Mhealth, and Motionsense datasets. On the HAPT dataset, its performance is 2.28% lower than the best model, IF-ConvT. Notably, to achieve such performance, our model requires only significantly fewer learnable parameters and lower complexity. For a comprehensive comparison, we conducted a pairwise average rank comparison using the Wilcoxon signed-rank test with Holm's alpha correction at 5% [13, 32]. The average ranking across all datasets, listed in Table 2, shows the IF-ConvT model achieving the best performance, followed by the Attend, ALAE-TAE, and proposed MLP-HAR models, which form a closely ranked cluster with no significant performance differences.

Compared to the best-performing model IF-ConvT, our MLP-HAR can provide 10× speed up in inference time while lose only 0.74% lower F1 score on average across six datasets. This slight accuracy drop can be further mitigated by the post-process in practical application, see Section 4.3.

Compared to the lightweight model TinyHAR, our MLP-HAR, despite having a similar number of learnable parameters, demonstrates significantly (10×) lower computational complexity. This reduced complexity also reflects in the inference time on the device, with our model running about 6× faster on average compared to TinyHAR.

Compared to the benchmark model DCL, MLP-HAR significantly outperforms it. The DCL model has much greater complexity and substantially slower inference times. For instance, when processing the DSADS dataset on the Samsung Galaxy Watch 5 PRO, DCL requires 298.61 ms, whereas our model operates in just 10.61 ms. Enhancements to the DCL framework, such as DCL-Attn, Attend, and ALAE-TAE models, have shown effectiveness in our experiments. However, these models are even slower than DCL. Interestingly, on most datasets, these three models exhibit generally lower complexity than DCL. Theoretically, the DCL model, which incorporates only one LSTM layer, should exhibit lower complexity than these three models, which use two LSTM layers. However, as detailed in section 4.1.2, the large kernel size of 11 in DCL's convolutional layers significantly increases its complexity. This observation highlights two key points: convolutional layers processing raw data contribute substantially to the model's overall complexity, and inference time is influenced not only by computational complexity but also by architectural design. The excessive use of LSTM layers and self-attention in the DCL-Attn, Attend, and ALAE-TAE models,

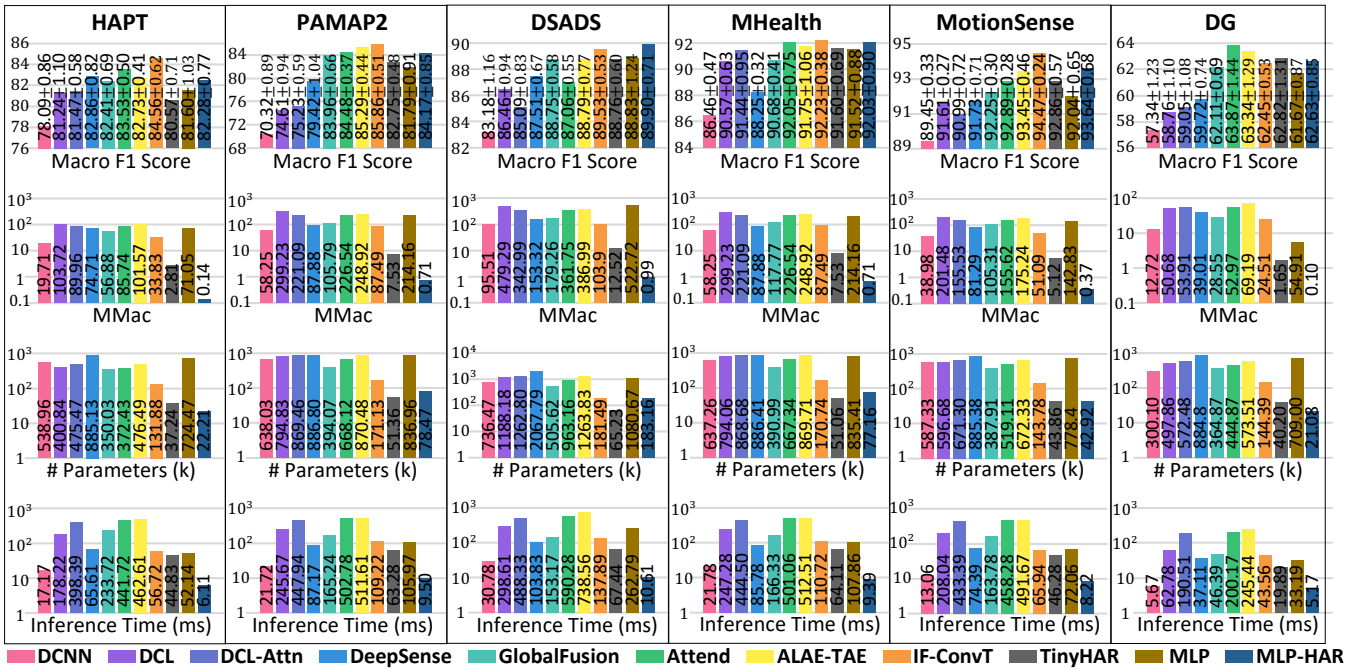


**Figure 2: Classification performance on six datasets.**

due to the sequential calculations inherent in LSTM layers which lack efficient parallelism, further slows down their inference times.

This observation that structural complexity can slow down inference is underscored by comparing the DeepSense and GlobalFusion models. Despite having lower model complexity per dataset, GlobalFusion exhibits significantly slower inference times. This slowdown is due to its complex multi-branch structure, which incorporates self-attention layers for global information fusion. This example highlights the importance of our proposed model's design philosophy: its plain topology is deployment-friendly, emphasizing efficiency and simpler architectural choices that facilitate faster processing speeds.

Among all models examined, the DCNN model is the least effective. Despite its higher computational complexity and greater number of learnable parameters, it achieves the second-fastest inference time on the device. For example, when processing the DG dataset on a smartwatch, its inference time is comparable to that of the proposed MLP-HAR, even with a higher parameter count and complexity. This underscores that a plain topology, combined with the use of CNN and FC layers, significantly aids in the efficient deployment of models. This observation is further supported by the performance of the MLP model, which is fast despite having many parameters and high computational complexity due to its configuration. However, since the MLP model completely ignores the characteristics of the HAR task, its performance is much worse than that of the proposed MLP-HAR model.

## 4.3 Post-Processing

In the deployment of HAR models, transitions between different activities are typically gradual. To enhance the robustness and accuracy of predictions, a post-processing technique known as majority voting is frequently utilized. To assess the performance of IF-ConvT and our proposed MLP-HAR model during deployment, we applied post-processing to all their predictions. The device operates with a double buffering mechanism, where one thread collects and buffers data, and another thread manages model inference. The window size for majority voting was set to 10. Table 3 illustrates the results of both models before and after post-processing. The results indicate that post-processing generally improves model performance, especially when the models already demonstrate good initial performance. For example, on the MotionSense and MHealth datasets, both models showed significant improvements. On the HAPT dataset, the performance gap between IF-ConvT and MLP-HAR narrowed from 2.28% to 0.35%. It is crucial to emphasize that models with faster inference times hold a distinct advantage in actual deployment. A model with shorter inference times can update its predictions more frequently, allowing it to process more data windows within the same amount of time when using a majority voting strategy, thereby solidifying the results. Conversely, if a model's inference time is long, a large voting window can lead to response delays.

| | HAPT | PAMAP2 | DSADS | MHealth | MotionSense | DG |
|---|---|---|---|---|---|---|
| IF-ConvT | 84.56 | 85.86 | 89.53 | 92.23 | 94.47 | 62.45 |
| IF-ConvT+P | 87.28 | 87.42 | 92.49 | 97.45 | 98.96 | 63.04 |
| MLP-HAR | 82.28 | 84.17 | 89.90 | 92.03 | 93.64 | 62.63 |
| MLP-HAR+P | 86.93 | 87.50 | 92.81 | 97.29 | 98.73 | 62.11 |

**Table 3: Comparison of performance before and after post-processing. The model name + P stands for post-processing.**

## 4.4 Ablation Study and Parameter Analysis

To validate the impact of different representations in the data embedding module and the effect of skip connections in the mixer module, we conducted an extensive ablation study. We evaluated the model using only *frequency* representation, only *temporal* representation, and *both* representations in the data embedding module.

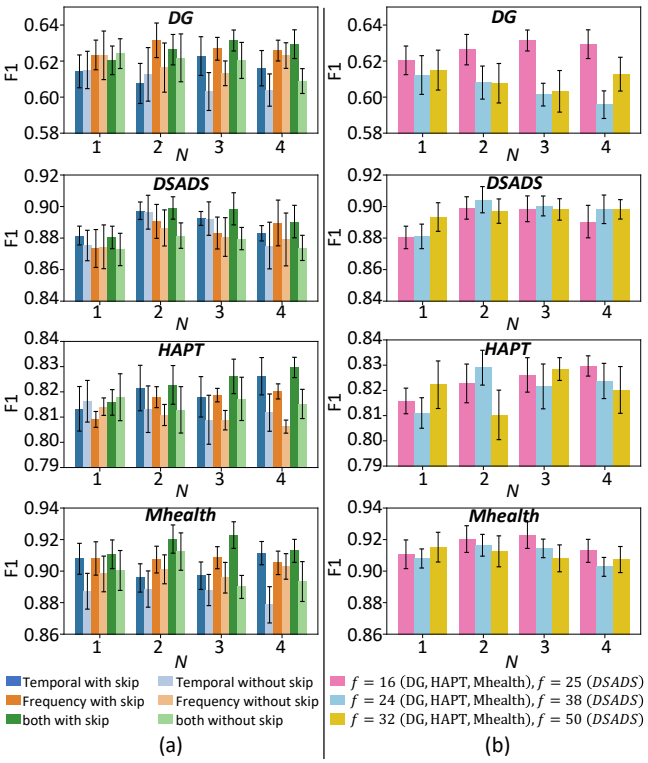


**Figure 3: (a) Ablation study for validating the contributions of different representations and skip connections, as well as the impact analysis of parameter $N$. (b) Impact analysis of parameter $f$ and $N$.**

Additionally, we examined the impact of including (*True*) or excluding (*False*) skip connections in the mixer module. This resulted in six different model configurations. We also investigated the effect of the number of blocks $N$ on performance, training and evaluating the six model configurations with $N$ set to 1, 2, 3, and 4. The results are presented in Figure 3(a). The results show that using skip connections generally improves model performance. The performance of using only frequency or temporal representation varies across datasets. For example, on the DG and MHealth datasets, models using frequency representation outperform those using temporal representation, whereas on the DSADS and HAPT datasets, the opposite is true. Using both representations leverages the strengths of each, leading to improved performance. The model's performance is also influenced by changes in $N$, varying by dataset. For instance, on the HAPT dataset, more blocks lead to better performance, while on the DSADS dataset, $N = 2$ achieves the best results. Increasing $N$ raises model complexity and the number of parameters, introducing different over-fitting risks depending on the dataset. However, models with $N \geq 2$ consistently outperform those with $N = 1$. We speculate this is because $N \geq 2$ allows the model to perform multiple fusions of information across temporal and sensor channel dimensions.

After confirming the benefits of using both representations and skip connections, we examined the impact of parameter variations in the FFT transformation. Specifically, we varied the size of $f$ and the number of blocks $N$ while employing both representations and skip connections. Figure 3(b) shows the range of $f$ and $N$ values and their corresponding model performance. The results indicate that there is no universal combination of $f$ and $N$ that consistently improves performance across all datasets. However, optimizing these parameters for individual datasets does enhance performance. For instance, on the DSADS and HAPT datasets, adjusting $f$ and $N$ values resulted in significant performance improvements.

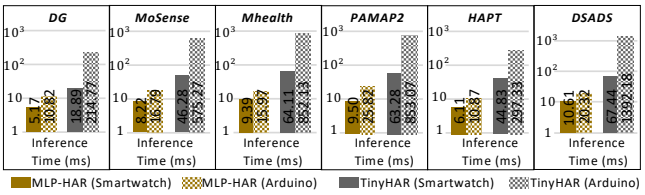


**Figure 4: Comparison of model inference time on two devices. Y-axis is logarithmically scaled (base 10).**

## 4.5 Deployment on Hardware

In this section, we explore the deployment of our model on the Arduino Portenta H7 LITE, a board with more limited computing capabilities. As previously shown, because the TinyHAR model demonstrates the lowest inference time and computational complexity aside from our proposed MLP-HAR model, we report only the inference times for TinyHAR and MLP-HAR on the Arduino Portenta H7 LITE. The results are illustrated in Figure 4. From the figure, it is evident that due to the reduced computational power of the device, the inference times of both models are longer compared to those on the smartwatch. However, it is noteworthy that the increase in inference time for TinyHAR is substantial, slowing down by at least 10×. In contrast, the increase in inference time for the proposed MLP-HAR model is approximately 2× to 3× between the two devices. This outcome underscores the superiority of our proposed model when deployed on devices with more restricted computing capabilities, which can be attributed to MLP-HAR's plain topology composed solely of FC layers.

.

## 5 CONCLUSION

In this work, we introduced a purely FC model architecture, thoughtfully designed not only to leverage the different saliencies of multi-modalities and temporal information extraction, but also to facilitate efficient deployment on edge devices. Experimental results demonstrate that compared to current SOTA HAR models, our model delivers comparable performance while boasting the smallest model size, minimal computational complexity, and the fastest inference time. When deployed on edge devices with limited computational capacity, the proposed model's superior capabilities were further showcased, highlighting its potential for practical real-world applications where computational resources are at a premium.

## REFERENCES

[1] Alireza Abedin, Mahsa Ehsanpour, Qinfeng Shi, Hamid Rezatofighi, and Damith C Ranasinghe. 2021. Attend and discriminate: Beyond the state-of-the-art for human activity recognition using wearable sensors. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 5, 1 (2021), 1–22.

[2] Nafees Ahmad and Ho-fung Leung. 2023. ALAE-TAE-CutMix+: Beyond the State-of-the-Art for Human Activity Recognition Using Wearable Sensors. In

*2023 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 222–231.

[3] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450* (2016).

[4] Marc Bachlin, Daniel Roggen, Gerhard Troster, Meir Plotnik, Noit Inbar, Inbal Meidan, Talia Herman, Marina Brozgol, Eliya Shaviv, Nir Giladi, et al. 2009. Potentials of enhanced context awareness in wearable assistants for Parkinson's disease patients with the freezing of gait syndrome. In *2009 International Symposium on Wearable Computers*. IEEE, 123–130.

[5] Oresti Banos, Rafael Garcia, Juan A Holgado-Terriza, Miguel Damas, Hector Pomares, Ignacio Rojas, Alejandro Saez, and Claudia Villalonga. 2014. mHealthDroid: a novel framework for agile development of mobile health applications. In *Ambient Assisted Living and Daily Activities: 6th International Work-Conference, IWAAL 2014, Belfast, UK, December 2-5, 2014. Proceedings 6*. Springer, 91–98.

[6] Billur Barshan and Murat Cihan Yüksek. 2014. Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units. *Comput. J.* 57, 11 (2014), 1649–1667.

[7] Marius Bock, Alexander Hölzemann, Michael Moeller, and Kristof Van Laerhoven. 2021. Improving deep learning for HAR with shallow LSTMs. In *Proceedings of the 2021 ACM International Symposium on Wearable Computers*. 7–12.

[8] Kaixuan Chen, Dalin Zhang, Lina Yao, Bin Guo, Zhiwen Yu, and Yunhao Liu. 2021. Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities. *ACM Computing Surveys (CSUR)* 54, 4 (2021), 1–40.

[9] Si-An Chen, Chun-Liang Li, Nate Yoder, Sercan O Arik, and Tomas Pfister. 2023. Tsmixer: An all-mlp architecture for time series forecasting. *arXiv preprint arXiv:2303.06053* (2023).

[10] Yuwen Chen, Kunhua Zhong, Ju Zhang, Qilong Sun, and Xueliang Zhao. 2016. LSTM networks for mobile human activity recognition. In *2016 International conference on artificial intelligence: technologies and applications*. Atlantis Press, 50–53.

[11] Yves Luduvico Coelho, Francisco de Assis Souza dos Santos, Anselmo Frizera-Neto, and Teodiano Freire Bastos-Filho. 2021. A lightweight framework for human activity recognition on wearable devices. *IEEE Sensors Journal* 21, 21 (2021), 24471–24481.

[12] Xiaohan Ding, Xiangyu Zhang, Ningning Ma, Jungong Han, Guiguang Ding, and Jian Sun. 2021. Repvgg: Making vgg-style convnets great again. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 13733–13742.

[13] Sture Holm. 1979. A Simple Sequentially Rejective Multiple Test Procedure. *Scandinavian journal of statistics* (1979), 65–70.

[14] Jie Hu, Li Shen, and Gang Sun. 2018. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7132–7141.

[15] Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).

[16] Shengzhong Liu, Shuochao Yao, Jinyang Li, Dongxin Liu, Tianshi Wang, Huajie Shao, and Tarek Abdelzaher. 2020. Globalfusion: A global attentional deep learning framework for multisensor information fusion. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 1 (2020), 1–27.

[17] Haojie Ma, Wenzhong Li, Xiao Zhang, Songcheng Gao, and Sanglu Lu. 2019. AttnSense: Multi-level attention mechanism for multimodal human activity recognition.. In *IJCAI*. 3109–3115.

[18] Xiao Ma, Shengfeng He, Hezhe Qiao, and Dong Ma. 2024. DiTMoS: Delving into Diverse Tiny-Model Selection on Microcontrollers. In *2024 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, 69–79.

[19] Saif Mahmud, M Tonmoy, Kishor Kumar Bhaumik, AK Mahbubur Rahman, M Ashraful Amin, Mohammad Shoyaib, Muhammad Asif Hossain Khan, and Amin Ahsan Ali. 2020. Human activity recognition from wearable sensor data using self-attention. *arXiv preprint arXiv:2003.09018* (2020).

[20] Mohammad Malekzadeh, Richard G Clegg, Andrea Cavallaro, and Hamed Haddadi. 2019. Mobile sensor data anonymization. In *Proceedings of the international conference on internet of things design and implementation*. 49–58.

[21] Sebastian Münzner, Philip Schmidt, Attila Reiss, Michael Hanselmann, Rainer Stiefelhagen, and Robert Dürichen. 2017. CNN-based sensor fusion techniques for multimodal human activity recognition. In *Proceedings of the 2017 ACM international symposium on wearable computers*. 158–165.

[22] Vishvak S Murahari and Thomas Plötz. 2018. On attention models for human activity recognition. In *Proceedings of the 2018 ACM international symposium on wearable computers*. 100–103.

[23] Ronald Mutegeki and Dong Seog Han. 2020. A CNN-LSTM approach to human activity recognition. In *2020 international conference on artificial intelligence in information and communication (ICAIIC)*. IEEE, 362–366.

[24] Kamsiriochukwu Ojiako and Katayoun Farrahi. 2023. MLPs Are All You Need for Human Activity Recognition. *Applied Sciences* 13, 20 (2023), 11154.

[25] Francisco Javier Ordóñez and Daniel Roggen. 2016. Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. *Sensors* 16, 1 (2016), 115.

[26] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. 2019. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems* 32 (2019).

[27] Attila Reiss and Didier Stricker. 2012. Introducing a new benchmarked dataset for activity monitoring. In *2012 16th international symposium on wearable computers*. IEEE, 108–109.

[28] Jorge-L Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-aware human activity recognition using smartphones. *Neurocomputing* 171 (2016), 754–767.

[29] Sarbagya Ratna Shakya, Chaoyang Zhang, and Zhaoxian Zhou. 2018. Comparative study of machine learning and deep learning architecture for human activity recognition using accelerometer data. *Int. J. Mach. Learn. Comput* 8, 6 (2018), 577–582.

[30] Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. 2021. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems* 34 (2021), 24261–24272.

[31] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems* 30 (2017).

[32] Frank Wilcoxon. 1992. Individual comparisons by ranking methods. In *Breakthroughs in Statistics: Methodology and Distribution*. Springer, 196–202.

[33] Zhi-Qin John Xu, Yaoyu Zhang, and Tao Luo. 2022. Overview frequency principle/spectral bias in deep learning. *arXiv preprint arXiv:2201.07395* (2022).

[34] Jianbo Yang, Minh Nhut Nguyen, Phyo Phyo San, Xiaoli Li, and Shonali Krishnaswamy. 2015. Deep convolutional neural networks on multichannel time series for human activity recognition.. In *Ijcai*, Vol. 15. Buenos Aires, Argentina, 3995–4001.

[35] Shuochao Yao, Shaohan Hu, Yiran Zhao, Aston Zhang, and Tarek Abdelzaher. 2017. Deepsense: A unified deep learning framework for time-series mobile sensing data processing. In *Proceedings of the 26th international conference on world wide web*. 351–360.

[36] Ye Zhang, Longguang Wang, Huiling Chen, Aosheng Tian, Shilin Zhou, and Yulan Guo. 2022. IF-ConvTransformer: A framework for human activity recognition using IMU fusion and ConvTransformer. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 6, 2 (2022), 1–26.

[37] Yu Zhao, Rennong Yang, Guillaume Chevalier, Ximeng Xu, and Zhenxing Zhang. 2018. Deep residual bidir-LSTM for human activity recognition using wearable sensors. *Mathematical Problems in Engineering* 2018 (2018), 1–13.

[38] Yexu Zhou, Michael Hefenbrock, Yiran Huang, Till Riedel, and Michael Beigl. 2021. Automatic remaining useful life estimation framework with embedded convolutional LSTM as the backbone. In *Machine Learning and Knowledge Discovery in Databases: Applied Data Science Track: European Conference, ECML PKDD 2020, Ghent, Belgium, September 14–18, 2020, Proceedings, Part IV*. Springer, 461–477.

[39] Yexu Zhou, Tobias King, Yiran Huang, Haibin Zhao, Till Riedel, Tobias Röddiger, and Michael Beigl. 2024. Enhancing Efficiency in HAR Models: NAS Meets Pruning. In *2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*. IEEE, 33–38.

[40] Yexu Zhou, Haibin Zhao, Yiran Huang, Till Riedel, Michael Hefenbrock, and Michael Beigl. 2022. TinyHAR: A lightweight deep learning model designed for human activity recognition. In *Proceedings of the 2022 ACM International Symposium on Wearable Computers*. 89–93.