WHAR Datasets: An Open Source Library for Wearable Human Activity Recognition

Maximilian Burzer
maximilian.burzer@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Tobias King
tobias.king@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Till Riedel
riedel@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Michael Beigl
michael.beigl@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Tobias Röddiger
tobias.roeddiger@kit.edu
Karlsruhe Institute of Technology
Karlsruhe, Germany

Abstract

The lack of standardization across Wearable Human Activity Recognition (WHAR) datasets limits reproducibility, comparability, and research efficiency. We introduce WHAR datasets, an open-source library designed to simplify WHAR data handling through a standardized data format and a configuration-driven design, enabling reproducible and computationally efficient workflows with minimal manual intervention. The library currently supports 9 widely-used datasets, integrates with PyTorch and TensorFlow, and is easily extensible to new datasets. To demonstrate its utility, we trained two state-of-the-art models, TinyHar and MLP-HAR, on the included datasets, approximately reproducing published results and validating the library's effectiveness for experimentation and benchmarking. Additionally, we evaluated preprocessing performance and observed speedups of up to 3.8× using multiprocessing. We hope this library contributes to more efficient, reproducible, and comparable WHAR research.

CCS Concepts

• **Human-centered computing → Human computer interaction (HCI)**; **Ubiquitous and mobile computing**; • **Computing methodologies → Machine learning**.

Keywords

Human Activity Recognition, HAR, Dataset Standardization, Data Preprocessing, Open Source Library, Machine Learning

ACM Reference Format:
Maximilian Burzer, Tobias King, Till Riedel, Michael Beigl, and Tobias Röddiger. 2025. WHAR Datasets: An Open Source Library for Wearable Human Activity Recognition. In *Companion of the 2025 ACM International Joint Conference on Pervasive and Ubiquitous Computing (UbiComp Companion '25), October 12–16, 2025, Espoo, Finland.* ACM, New York, NY, USA, 8 pages. https://doi.org/10.1145/3714394.3756254

UbiComp Companion '25, Espoo, Finland

ACM ISBN 979-8-4007-1477-1/2025/10
https://doi.org/10.1145/3714394.3756254

1 Introduction

Wearable Human Activity Recognition (WHAR) research in areas such as healthcare, sports analytics, and smart environments [17] has led to the creation of a diverse and growing number of datasets [5, 18, 29]. However, this growth has also exposed a significant challenge within the WHAR community: the lack of standardization. Datasets are often characterized by significant variability in file structures and formats, heterogeneous data, and general preprocessing requirements [3, 5, 18, 29]. As a result, researchers frequently rely on custom dataset-specific code to handle and experiment with each dataset [3]. This leads to repetitive work, hinders the reproducibility of published results, and complicates fair comparisons across models, even when evaluated on the same dataset. Unlike other fields such as natural language processing, where centralized benchmarks are common, the lack of standardization in WHAR makes such efforts difficult. Consequently, researchers still spend considerable time and effort on the tedious and error-prone task of data handling, diverting focus from their primary research goals and novel contributions. Although the need for standardization has been acknowledged in the literature [3, 16], only limited steps have been taken to tackle these issues systematically.

To overcome these challenges, we introduce a novel open-source library designed to standardize and streamline data handling in WHAR research. The library is available on GitHub under the MIT license [8], encouraging community collaboration and wide adoption. Our library allows the conversion of a growing number of datasets from the literature into a standardized format using dataset-specific parsers, which can subsequently be preprocessed and loaded without requiring manual work. Using multiprocessing and caching, the library achieves both computational and memory efficiency, minimizing redundant recomputation. It is framework-agnostic at its core, to allow integration with different deep learning frameworks such as PyTorch [24] and TensorFlow [1] using framework-specific adapters. A core strength of our library is its configuration-driven design, enabling consistent and unified data handling while remaining flexible to the unique requirements of each dataset. We designed the library in such a way that custom WHAR datasets, which are not yet supported, can easily be integrated and used with the library by simply providing a dataset-specific configuration including a parser. By abstracting away the complexities and inconsistencies of individual dataset formats, this approach aims to provide a unified data handling solution for WHAR. We hope that the introduction of this standardized library will lead to the following advancements within the WHAR research community:

(1) **Increased Efficiency**: By providing ready-to-use data handling, the library aims to allow researchers to focus on building and evaluating models instead of spending time on data handling and integration.
(2) **Improved Reproducibility**: By enforcing consistent data configurations, our library is expected to help researchers replicate experiments more reliably, reducing the variability introduced by inconsistent data handling.
(3) **Fair Comparability**: Standardized data handling has the potential to enable fair and direct comparisons across models and datasets, supporting the development of standardized WHAR benchmarks.

2 Background and Motivation

WHAR relies heavily on the availability and reuse of open datasets [3]. However, numerous challenges hinder effective utilization, as identified by [3] through a comprehensive literature review and a questionnaire survey of researchers with expertise in HAR. Their study presents a conceptual framework encompassing four phases in the open dataset lifecycle: construction, sharing, searching, and usage. Across these phases, several recurring issues were identified, that significantly impact research productivity and reproducibility.

One of the most pressing challenges highlighted is the absence of a standardized data format for HAR datasets. This lack of consistency complicates data sharing, hinders reproducibility, and adds unnecessary effort when benchmarking or applying novel methods. Researchers reported that datasets often come with missing values, errors, or in unstructured, non-presentable formats, discouraging reuse and increasing the preprocessing burden. In fact, when using existing datasets, nearly 67.7% of participants indicated needing either "some" or "a lot" of preprocessing effort before the data could be used experimentally. Moreover, while a large majority (96.9%) of respondents download open datasets for experimentation, their ability to make use of them is limited by poor metadata, inconsistent annotations, and idiosyncratic formats. Key selection criteria for dataset reuse include the availability of code or scripts for data processing (48.4%), the data format itself (54.8%), and the presence of clear metadata (45.2%). These concerns highlight the community's desire for datasets that are easy to integrate, well-documented, and usable with minimal manual intervention.

Our work directly addresses these gaps by introducing a novel open-source library designed to standardize and streamline dataset handling in WHAR research. It tackles the root causes behind many of the data-sharing and usability issues, such as lack of standardization, metadata inconsistency, and preprocessing overhead, ultimately promoting collaboration and reproducibility in the WHAR community.

3 Related Work

Efforts to standardize dataset access and improve usability have gained significant traction across machine learning domains. Prominent general-purpose libraries and platforms such as Hugging Face [19] and OpenML [13] have made substantial contributions in this area. Hugging Face's Datasets library enables seamless loading of datasets for natural language processing, computer vision, and audio tasks via a unified API. Integration with the Hugging Face Hub further simplifies sharing and collaboration within the community. Similarly, OpenML serves as an open platform for sharing datasets, provides standardized APIs, and encourages reproducibility and reuse through its collaborative design. While both platforms promote dataset accessibility and interoperability, their primary focus is on general-purpose usage across domains. They neither offer domain-specific preprocessing pipelines nor support the specialized requirements of WHAR datasets, e.g. resampling or windowing.

Several domain-specific libraries address adjacent challenges. For instance, [15] and sktime [20] offer comprehensive frameworks for time series learning, supporting forecasting, classification, and transformation pipelines. While including preprocessing functionality relevant to WHAR, their focus remains on general time series tasks. As such, these frameworks fall short in addressing the domain-specific challenges of WHAR pipelines and cannot be directly applied without extensive adaptation.

Other initiatives such as MLCroissant [2] and DCAT-AP [12] focus on standardizing dataset metadata and improving dataset discoverability, particularly in the context of FAIR (findability, accessibility, interoperability, reusability) data principles [33]. While these formats contribute to making datasets more "AI-ready," they are primarily concerned with semantic interoperability rather than the structural readiness required for WHAR data handling.

The lack of standardization in WHAR has also been addressed by prior work such as [16], which focuses primarily on standardizing training methodologies and evaluation protocols. However, their contribution centers on the modeling and training process rather than on dataset access and preprocessing.

In contrast to these prior efforts, our proposed library is purpose-built for WHAR data handling. It unifies dataset access, preprocessing, and deep learning integration through a modular architecture centered around a standardized data format. Crucially, our focus on WHAR-specific requirements distinguishes our library from general-purpose solutions. Over time, this framework can serve as a foundation for benchmarking and sharing WHAR datasets in a consistent and reproducible manner.

4 Requirements and Features

To help achieve the goals of increased efficiency, improved reproducibility, fair comparability, the WHAR datasets library must satisfy several key design requirements and offer a rich set of features. We begin by outlining the non-functional requirements.

(R1) **Usability**: The library should offer a simple, intuitive interface for preprocessing and loading WHAR datasets, enabling users to start experiments with just a few lines of code.
(R2) **Reproducibility**: A standardized, dataset-agnostic configuration schema should define data handling parameters transparently, ensuring consistent and reusable experimentation.
(R3) **Dataset-Agnosticism**: The library should support diverse WHAR datasets by converting them into a standardized format. Integrating new datasets should require minimal effort, using a parser to map to this format, ensuring extensibility and backward compatibility.

(R4) **Framework-Agnosticism**: The library should be compatible with major deep learning frameworks like PyTorch [24] and TensorFlow [1], allowing flexible model development.
(R5) **Computational Efficiency**: To support large datasets and changing hyperparameters, the standardized data format should be optimized for computational efficiency, enabling multiprocessing and caching to ensure scalability.

In addition to these non-functional requirements, the following functional features are critical. They are tailored specifically to the WHAR domain and underscore the necessity of a dedicated library designed to address its unique challenges.

(F1) **Subject-Wise Splitting**: The library must support subject-disjoint train/validation/test splits to prevent data leakage and better reflect real-world deployment. It should enable protocols like subject-wise and leave-one-subject-out (LOSO) cross-validation.
(F2) **Normalization**: Built-in support for common normalization methods, including min-max, z-score, and robust scaling, is required. Both per-window and global (train-set-based) normalization modes should be supported.
(F3) **Sample Loading**: To accommodate varying memory and runtime constraints, the library should provide two sample-loading strategies: on-demand (loading data from disk as needed) for large datasets, and preload (loading all windows into memory) for faster access when dealing with smaller datasets, which is common for WHAR.
(F4) **Class Weighting**: Given the prevalence of class imbalance in WHAR due to varying activity durations, the library should compute class weights from the training data for integration into loss functions.

5 Design and Implementation

Building on the non-functional requirements and functional features outlined in the previous section, the WHAR datasets library is designed with a strong focus on modularity, extensibility, and computational efficiency. As illustrated in Figure 1, the architecture cleanly separates dataset-specific configuration and parsing, as well as framework-specific adapters, from the core library. This separation enables straightforward integration of new WHAR datasets (R3), ensures compatibility with major deep learning frameworks (R4), thereby enhancing overall usability (R1), and offers robust support for WHAR-specific data handling needs (F1–F4).

5.1 Standardized Data Format

To support scalable integration of diverse WHAR datasets (R3), the library employs a standardized data format based on a session-centric representation. Each session corresponds to a single subject performing a single activity and is stored as an individual file containing timestamp-indexed, multivariate time series data from inertial measurement units (IMUs) and other sensors. This design enables multiprocessing of sessions, enhancing the library's scalability to very large datasets (R5).

Parquet was selected as the storage format due to its columnar structure, built-in compression, and strong compatibility with scalable data processing frameworks such as Dask [28], all of which enhance computational efficiency and scalability (R5). Its suitability

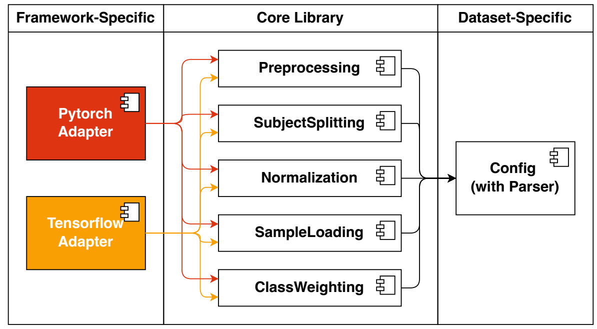

Figure 1: Structure of the library showing clear separation of dataset- and framework-specific components from feature components of the core library.

for large-scale time series machine learning tasks has also been demonstrated in a prior evaluation by OpenML [22].

To support features such as subject-wise data splitting (F1) and class weighting (F4) without repeated reads of raw sensor data, metadata required for subject identification and activity labeling is stored separately in centralized, structured tables. This metadata layer, illustrated in Figure 2, enables efficient filtering and partitioning during preprocessing, further contributing to overall computational efficiency (R5).

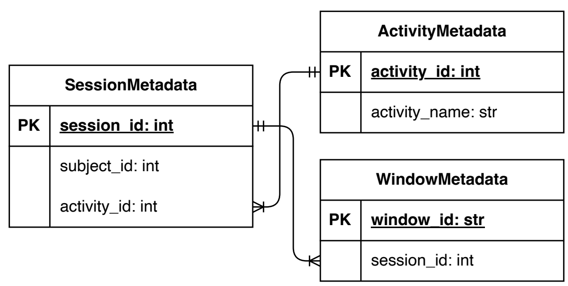

Figure 2: Entity-relationship diagram illustrating the metadata schema for the standardized data format.

5.2 Configuration and Parsing

WHAR datasets are integrated through lightweight configurations containing dataset-specific metadata, data handling hyperparameters, and a parser. While implementing a parser requires an initial manual effort, it guarantees dataset-agnosticism (R3) by converting raw dataset formats into the standardized data format described earlier. Once integrated, the dataset is fully compatible with the core library's preprocessing pipeline and feature components discussed in the next section. Users can easily adapt preprocessing by modifying configuration values, thereby supporting usability (R1). Moreover, using the same configuration with the library ensures consistent results, which promotes reproducibility (R2).

To further improve usability and ensure configuration correctness (R1), the configuration schema is implemented using Pydantic,

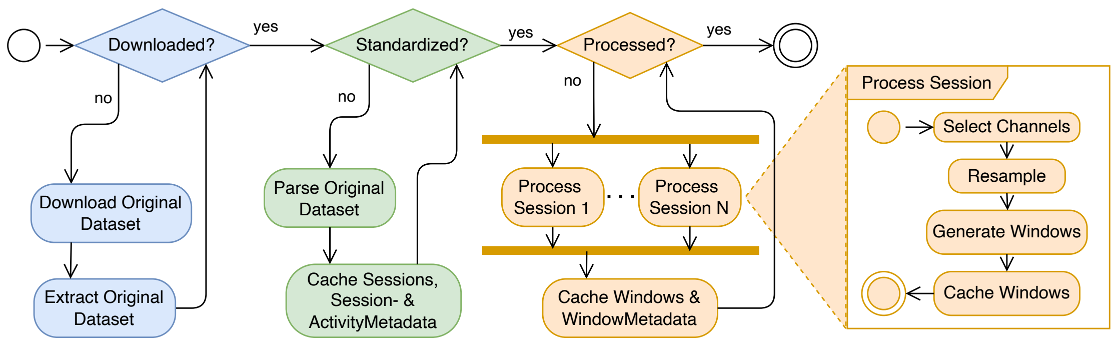

Figure 3: Activity diagram illustrating the preprocessing workflow. The process is divided into three main steps: (1) downloading the dataset, (2) converting it into a standardized data format, and (3) processing sessions to produce windows, optionally using multiprocessing. Each step is cached to prevent redundant computations.

enabling automatic validation. This allows users to catch misconfigurations early and enhances the overall reliability of data handling. An example configuration is shown in Figure 4.

5.3 Feature Components

The core library implements a modular preprocessing pipeline alongside components that realize various functional features. These components operate on the standardized data format and are fully driven by dataset-specific configurations described in the previous section, promoting reproducibility across experiments (R2). Functional features, including subject-wise splitting (F1), normalization (F2), sample loading (F3), and class weighting (F4), are implemented as individual components. This design ensures consistent behavior across datasets while effectively addressing WHAR-specific data processing requirements.

The preprocessing pipeline handles the complete dataset preparation workflow, from downloading and parsing into the standardized format to executing a series of user-defined operations such as activity filtering, channel selection, resampling, and windowing. It validates the output of parsing against constraints of the standardized data format, utilizes caching to prevent redundant processing, and employs optional multiprocessing, ensuring efficiency and scalability (R4). The resulting data windows are saved as individual Parquet files, each linked to its session via a centralized metadata table (see Figure 2), ensuring full data traceability. An overview of this pipeline is shown in Figure 3.

5.4 Framework Integration

To meet the requirement of framework-agnosticism (R4), the library provides dedicated adapters for the popular deep learning frameworks PyTorch [24] and TensorFlow [1]. These adapters utilize the core library's components while presenting familiar, framework-specific interfaces, enabling easy integration with existing training workflows and thus enhancing usability (R1). As illustrated in Figure 5, all core library functionality is fully abstracted and driven by configuration, ensuring consistency and reproducibility across experiments.

```
cfg = WHARConfig(
    # Info
    dataset_id="example",
    download_url="https://example.zip",
    sampling_freq=50,
    num_of_subjects=30,
    num_of_activities=6,
    num_of_channels=9,
    datasets_dir="./datasets",
    # Parsing
    parse=parse_example,
    # Preprocessing
    activity_names=["walking", "sitting", "laying"],
    sensor_channels=["acc_x", "acc_y", "acc_z"],
    window_time=2.56,
    window_overlap=0.5,
    in_parallel=True,
    resampling_freq=None,
    # Training
    batch_size=64,
    learning_rate=1e-4,
    num_epochs=100,
    seed=0,
    in_memory=True,
    given_train_subj_ids=list(range(0, 24)),
    given_test_subj_ids=list(range(24, 30)),
    subj_cross_val_split_groups=[
        [0, 1, 2, 3, 4],
        [5, 6, 7, 8, 9],
        [10, 11, 12, 13, 14],
        [15, 16, 17, 18, 19],
        [20, 21, 22, 23, 24],
        [25, 26, 27, 28, 29],
    ],
    val_percentage=0.1,
    normalization=NormType.STD_GLOBALLY,
)
```

Figure 4: An example configuration containing metadata and hyperparameters, organized into sections for information, parsing, preprocessing, and training.

```python
from whar_datasets.adapters.pytorch import PytorchAdapter
from whar_datasets.support.getter import WHARDatasetID
from whar_datasets.support.getter import get_whar_cfg

cfg = get_whar_cfg(WHARDatasetID.UCI_HAR)
dataset = PytorchAdapter(cfg, override_cache=False)

loaders = dataset.get_dataloaders(train_batch_size=32)
train_loader, val_loader, test_loader = loaders
```

Figure 5: Example usage of the library with PyTorch.

5.5 Supported Datasets

To ensure immediate usability, the library includes built-in support for a set of WHAR datasets commonly used in the literature. These datasets vary widely in the number of subjects, activities, and sampling rates, as summarized in Table 1. Each dataset comes with a complete configuration and a dedicated parser, enabling out-of-the-box use and enhancing usability (R1). These implementations also serve as practical templates for users who wish to integrate additional datasets. Note that our current focus has been on integrating a broad range of datasets by converting them into the standardized format. While we have applied basic data cleaning, we intend to enhance these efforts by refining and updating the parsers in the future.

Table 1: WHAR datasets currently supported by the library.

WHAR Dataset	Number of Subjects	Number of Activities	Sampling Rate (Hz)
UCI-HAR [5]	30	6	50
WISDM [18]	36	6	20
MHEALTH [6]	10	12	50
PAMAP2 [25]	9	18	100
OPPORTUNITY [29]	4	3	30
MotionSense [21]	9	6	50
DSADS [4]	8	19	25
Daphnet [23]	10	2	64
HARSense [11]	12	6	25

Thanks to the library's modular architecture, extending support to new datasets generally requires only implementing a parser and specifying a configuration, without modifying the core codebase. Potential datasets that could be integrated by the community to expand this collection include SHL [32], RealLifeHar [14], ExtraSensory [31], RealWorld [30], UTD-MHAD [9], USC-HAD [34], HuGaDB [10], w-HAR [7], HAPT [27], and WISDM-19 [26].

6 Experiments

6.1 Model Training and Evaluation

To implicitly demonstrate the functionality of the library, its dataset support, and its usability for benchmarking, we trained two popular WHAR models, TinyHAR [36] and MLP-HAR [35], on the 9 natively supported datasets (see Table 1). Results are shown in Figure 6. We did not perform hyperparameter tuning or other performance optimizations such as data cleaning, as the goal was simply to illustrate the usability and ease of benchmarking with the library. As a result, the reported performance is not directly comparable to results from papers that involve model-specific tuning or preprocess the dataset differently. Due to the library's design, the training and evaluation process only requires to implement a new model together with a training script as all the necessary components to obtain the dataloader are included in the library, therefore making this a fair comparison between the two tested models. Training was performed on a single NVIDIA A100 GPU with 40GB of memory, using a batch size of 256, the Adam optimizer, and a learning rate of 0.001. To prevent overfitting, early stopping was implemented, halting training after 15 consecutive epochs without improvement in validation loss. Evaluation followed a Leave-One-Subject-Out (LOSO) cross-validation protocol, with test subjects for each split predefined in the dataloader to ensure consistent and reproducible splits in line with the library's framework.

6.2 Preprocessing Performance Analysis

Furthermore, we assessed the impact of multiprocessing on preprocessing performance in comparison to sequential execution across the 9 natively supported datasets. As illustrated in Figure 7, we report both absolute time differences and speedup factors, based on measurements conducted on an M2 MacBook Pro with 10 CPU cores. For 8 out of the 9 datasets, multiprocessing achieved speedups between approximately 2.1× and 3.8×. The only exception was HARSense, the smallest dataset, where the overhead of multiprocessing outweighed its benefits. These findings demonstrate notable time savings when preprocessing multiple datasets, underscoring the library's computational efficiency and scalability to larger datasets.

7 Conclusion and Future Work

This paper introduces WHAR Datasets, an open-source library designed to standardize and streamline data handling for Wearable Human Activity Recognition (WHAR). By addressing challenges related to inconsistencies in data structures and formats, which often require dataset-specific handling, the library aims to improve research efficiency, enhance reproducibility, and enable fairer comparability within the WHAR community.

Key contributions of the library include a standardized data format, a configuration- and parser-based approach for dataset integration, WHAR-specific preprocessing and other functional features built on this format, and framework-agnostic adapters compatible with popular deep learning frameworks. These features allow researches to concentrate on application development rather than data handling, facilitating faster experimentation and more reliable comparison across different approaches. The design emphasizes extensibility, enabling straightforward integration of new datasets and ensuring the library's long-term applicability.

Currently, the library includes a curated collection of 9 widely used WHAR datasets. Future efforts focus on expanding the repository by adding more datasets from the literature, along with advanced data cleaning, augmentation and preprocessing techniques. New functional features are planned, such as window-level auxiliary feature generation (e.g., spectrograms, statistical summaries)

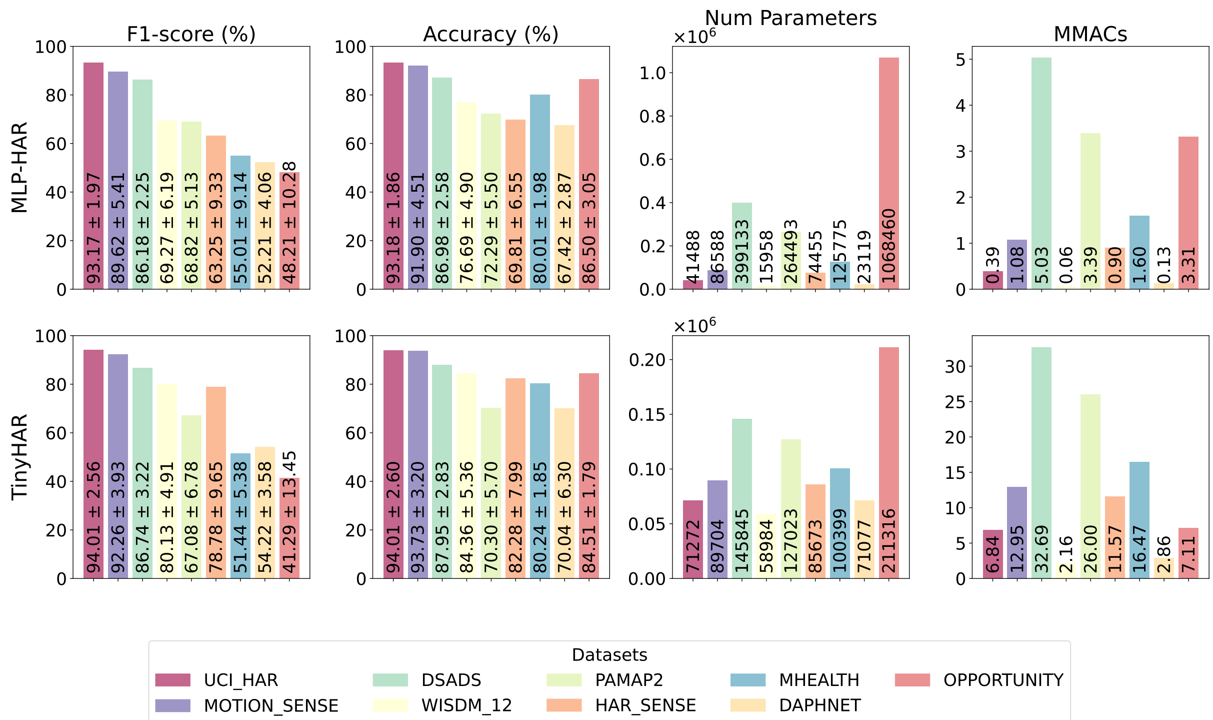

Figure 6: Experimental results evaluated on all ten natively supported datasets for two HAR-models: TinyHAR [36] and MLP-HAR [35]. Reported metrics include Accuracy, Macro F1-Score, the number of parameters and the Multiply-Accumulate (MAC) operations in the millions. The reported results are the mean results of the Leave-One-Subject-Out cross-validation strategy.

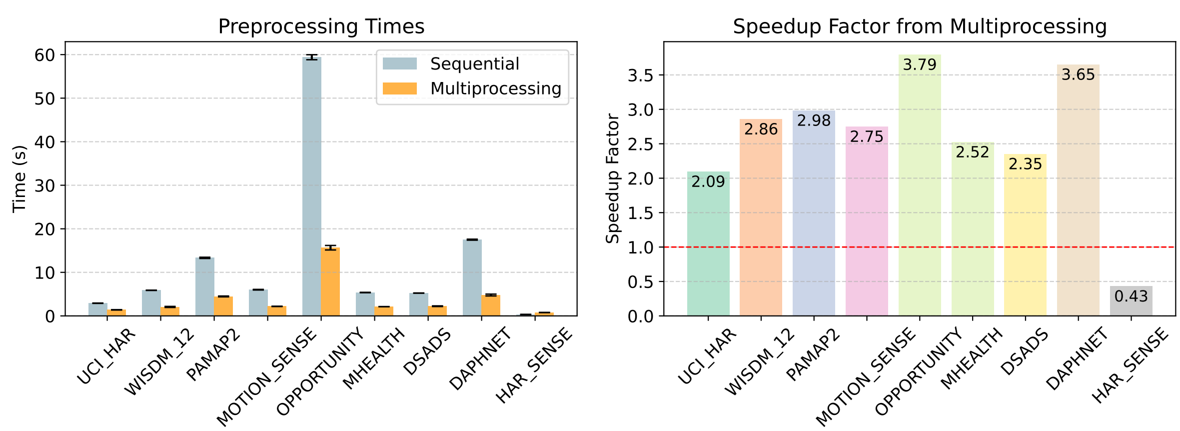

Figure 7: Preprocessing performance across the 9 natively supported datasets. The left plot displays the absolute time differences between sequential and multiprocessing execution, while the right plot shows the corresponding speedup factors. Speedup values above 1 (indicated by the red line) represent performance gains from multiprocessing. All measurements were conducted on an M2 MacBook Pro with 10 CPU cores.

alongside raw sensor data, enabling hybrid modeling approaches. We also aim to encourage community contributions to further enrich the library and support the development of standardized benchmarks for WHAR research.

Acknowledgments

This work is supported by the Helmholtz Association Initiative and Networking Fund on the HAICORE@KIT partition.

References

[1] Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dan Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.

[2] Mubashara Akhtar, Omar Benjelloun, Costanza Conforti, Pieter Gijsbers, Joan Giner-Miguelez, Nitisha Jain, Michael Kuchnik, Quentin Lhoest, Pierre Marcenac, Manil Maskey, Peter Mattson, Luis Oala, Pierre Ruyssen, Rajat Shinde, Elena Simperl, Goeffry Thomas, Slava Tykhonov, Joaquin Vanschoren, Jos van der Velde, Steffen Vogler, and Carole-Jean Wu. 2024. Croissant: A Metadata Format for ML-Ready Datasets. In *Proceedings of the Eighth Workshop on Data Management for End-to-End Machine Learning (SIGMOD/PODS '24)*. ACM, Amsterdam, Netherlands, 1–6. doi:10.1145/3650203.3663326

[3] Gulzar Alam, Ian McChesney, Peter Nicholl, and Joseph Rafferty. 2023. Open Datasets in Human Activity Recognition Research—Issues and Challenges: A Review. *IEEE Sensors Journal* 23, 22 (Nov. 2023), 26952–26980. doi:10.1109/jsen.2023.3317645 Publisher: Institute of Electrical and Electronics Engineers (IEEE).

[4] Kerem Altun, Billur Barshan, and Orkun Tunçel. 2010. Comparative study on classifying human activities with miniature inertial and magnetic sensors. *Pattern Recognit.* 43 (2010), 3605–3620. https://api.semanticscholar.org/CorpusID:18488847

[5] Davide Anguita, Alessandro Ghio, Luca Oneto, Xavier Parra, and Jorge L. Reyes-Ortiz. 2013. A Public Domain Dataset for Human Activity Recognition Using Smartphones. In *21st European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*. CIARP, Bruges, Belgium, 437–442. https://www.esann.org/sites/default/files/proceedings/legacy/es2013-84.pdf

[6] Oresti Banos, Rafael García, Juan 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 *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services. 8th International Conference, UCAmI 2014, Belfast, UK, December 2-5, 2014. Proceedings*, Vol. 8868. Springer, Cham, Switzerland, 91–98. doi:10.1007/978-3-319-13105-4_14

[7] Ganapati Bhat, Nicholas Tran, Holly Shill, and Ümit Y. Ogras. 2020. w-HAR: An Activity Recognition Dataset and Framework Using Low-Power Wearable Devices. *Sensors* 20, 18 (2020), 5356.

[8] Maximilian Burzer. 2025. WHAR Datasets. https://github.com/teco-kit/whar-datasets. MIT License.

[9] Cheng Chen, Roozbeh Jafari, and Nasser Kehtarnavaz. 2015. UTD-MHAD: A multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In *2015 IEEE International Conference on Image Processing (ICIP)*. IEEE, IEEE, Quebec City, QC, Canada, 168–172.

[10] Roman Chereshnev and Igor Kotenko. 2015. HuGaDB: Human Gait Database for Activity Recognition from Wearable Inertial Sensor Networks. *Proceedings of the 11th International Conference on Machine Learning and Data Mining in Pattern Recognition (MLDM)* 10716 (2015), 131–145.

[11] Nurul Amin Choudhury, Soumen Moulik, and Diptendu Sinha Roy. 2021. HARSense: Statistical Human Activity Recognition Dataset. doi:10.21227/9pt3-2m34

[12] European Commission. 2015. *DCAT Application Profile for data portals in Europe (DCAT-AP)*. Implementation Guide. European Commission. https://joinup.ec.europa.eu/collection/semantic-interoperability-community-semic/solution/dcat-application-profile-data-portals-europe Version 1.1, March 2015.

[13] Matthias Feurer, Jan N. van Rijn, Arlind Kadra, Pieter Gijsbers, Neeratyoy Mallik, Sahithya Ravi, Andreas Müller, Joaquin Vanschoren, and Frank Hutter. 2021. OpenML-Python: an extensible Python API for OpenML. *Journal of Machine Learning Research* 22, 100 (2021), 1–5. http://jmlr.org/papers/v22/19-920.html

[14] Daniel Garcia-Gonzalez, Daniel Rivero, Enrique Fernandez-Blanco, and Miguel R. Luaces. 2020. A Public Domain Dataset for Real-Life Human Activity Recognition Using Smartphone Sensors. *Sensors* 20, 8 (2020), 2200.

[15] Julien Herzen, Francesco Lässig, Samuele Giuliano Piazzetta, Thomas Neuer, Léo Tafti, Guillaume Raille, Tomas Van Pottelbergh, Marek Pasieka, Andrzej Skrodzki, Nicolas Huguenin, Maxime Dumonal, Jan Kościsz, Dennis Bader, Frédérick Gusset, Mounir Benheddi, Camila Williamson, Michal Kosinski, Matej Petrik, and Gaël Grosch. 2022. Darts: User-Friendly Modern Machine Learning for Time Series. *Journal of Machine Learning Research* 23 (2022), 1–6. arXiv:2110.03224 doi:10.48550/arXiv.2110.03224

[16] Yiran Huang, Haibin Zhao, Yexu Zhou, Till Riedel, and Michael Beigl. 2024. Standardizing Your Training Process for Human Activity Recognition Models: A Comprehensive Review in the Tunable Factors. doi:10.48550/arXiv.2401.05477 arXiv:2401.05477 [cs].

[17] Yiran Huang, Yexu Zhou, Haibin Zhao, Till Riedel, and Michael Beigl. 2024. A Survey on Wearable Human Activity Recognition: Innovative Pipeline Development for Enhanced Research and Practice. In *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, Yokohama, Japan, 1–10. doi:10.1109/ijcnn60899.2024.10650060

[18] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. 2011. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.* 12, 2 (March 2011), 74–82. doi:10.1145/1964897.1964918

[19] Quentin Lhoest, Albert Villanova del Moral, Yacine Jernite, Abhishek Thakur, Patrick von Platen, Suraj Patil, Julien Chaumond, Mariama Drame, Julien Plu, Lewis Tunstall, Joe Davison, Mario Šaško, Gunjan Chhablani, Bhavitvya Malik, Simon Brandeis, Teven Le Scao, Victor Sanh, Canwen Xu, Nicolas Patry, Angelina McMillan-Major, Philipp Schmid, Sylvain Gugger, Clément Delangue, Théo Matussière, Lysandre Debut, Stas Bekman, Pierric Cistac, Thibault Goehringer, Victor Mustar, François Lagunas, Alexander Rush, and Thomas Wolf. 2021. Datasets: A Community Library for Natural Language Processing. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Association for Computational Linguistics, Online and Punta Cana, Dominican Republic, 175–184. arXiv:2109.02846 [cs.CL] https://aclanthology.org/2021.emnlp-demo.21

[20] Markus Löning, Anthony Bagnall, Sajaysurya Ganesh, Viktor Kazakov, Jason Lines, and Franz J. Király. 2019. sktime: A Unified Interface for Machine Learning with Time Series. arXiv:1909.07872 [cs.LG] https://arxiv.org/abs/1909.07872

[21] 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* (Montreal, Quebec, Canada) *(IoTDI '19)*. ACM, New York, NY, USA, 49–58. doi:10.1145/3302505.3310068

[22] Mitar Milutinovic, Prabhant Singh, Joaquin Vanschoren, Pieter Gijsbers, Andreas Mueller, Matthias Feurer, Jan van Rijn, Marcus Weimer, Marcel Wever, Gertjan van den Burg, and Nick Poorman. 2020. Finding a standard dataset format for machine learning. https://blog.openml.org/openml/data/2020/03/23/Finding-a-standard-dataset-format-for-machine-learning.html. OpenML Blog.

[23] Steven T. Moore, Hamish G. MacDougall, and William G. Ondo. 2008. Ambulatory monitoring of freezing of gait in Parkinson's disease. *Journal of Neuroscience Methods* 167, 2 (Jan. 2008), 340–348. doi:10.1016/j.jneumeth.2007.08.023

[24] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. 2019. PyTorch: An Imperative Style, High-Performance Deep Learning Library. In *Advances in Neural Information Processing Systems 32*. Curran Associates, Inc., Red Hook, NY, USA, 8024–8035. http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[25] Attila Reiss and Didier Stricker. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring. *2012 16th International Symposium on Wearable Computers* 2012 (2012), 108–109. https://api.semanticscholar.org/CorpusID:10337279

[26] Jorge Luis Reyes-Ortiz, Luca Oneto, Xavier Parra, Davide Anguita, and Albert Samà. 2019. WISDM: Smartphone and Smartwatch Activity and Biometrics Dataset. UCI Machine Learning Repository. Dataset.

[27] Jorge-L. Reyes-Ortiz, Luca Oneto, Albert Samà, Xavier Parra, and Davide Anguita. 2016. Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* 171, 1 (2016), 754–767. doi:10.1016/j.neucom.2015.07.085 Preprint submitted March 2015.

[28] Matthew Rocklin. 2015. Dask: Parallel computation with blocked algorithms and task scheduling. In *Proceedings of the 14th Python in Science Conference*, Kathryn Huff and James Bergstra (Eds.). Citeseer, SciPy, Austin, Texas, 130–136.

[29] Daniel Roggen, Alberto Calatroni, Mirco Rossi, Thomas Holleczek, Kilian Förster, Gerhard Tröster, Paul Lukowicz, David Bannach, Gerald Pirkl, Alois Ferscha, Jakob Doppler, Clemens Holzmann, Marc Kurz, Gerald Holl, Ricardo Chavarriaga, Hesam Sagha, Hamidreza Bayati, Marco Creatura, and José del R. Millán. 2010. Collecting complex activity datasets in highly rich networked sensor environments. In *2010 Seventh International Conference on Networked Sensing Systems (INSS)*. IEEE, Kassel, Germany, 233–240. doi:10.1109/INSS.2010.5573607

[30] Timo Sztyler and Heiner Stuckenschmidt. 2016. On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition. In *2016 IEEE International Conference on Pervasive Computing and Communications (PerCom)*. IEEE, IEEE Computer Society, Piscataway, NJ, USA, 1–9.

[31] Yonatan Vaizman, Katherine Ellis, and Gert Lanckriet. 2018. Recognizing Detailed Human Context In-the-Wild from Smartphones and Smartwatches. *IEEE Journal of Biomedical and Health Informatics* 22, 5 (2018), 1427–1437.

[32] Zhiwu Wang, Tony Lee, Shuo Rui, Björn Stenger, Qin Yang, Wei Wang, Dawei Meng, Jun Wang, and Nick Barnes. 2019. The Sussex-Huawei Locomotion Dataset for Human Activity Recognition. In *Proceedings of the 2019 International Symposium on Wearable Computers*. ACM, London, UK, 50–53.

[33] Mark D. Wilkinson, Michel Dumontier, IJsbrand Jan Aalbersberg, Gabrielle Appleton, Myles Axton, Arie Baak, Niklas Blomberg, Jan-Willem Boiten, Luiz Bonino da Silva Santos, Philip E. Bourne, Jildau Bouwman, Anthony J. Brookes, Tim Clark, Mercè Crosas, Ingrid Dillo, Olivier Dumon, Scott Edmunds, Chris T. Evelo, Richard Finkers, Alejandra Gonzalez-Beltran, Alasdair J.G. Gray, Paul Groth, Carole Goble, Jeffrey S. Grethe, Jaap Heringa, Peter A.C. 't Hoen, Rob Hooft, Tobias Kuhn, Ruben Kok, Joris Kok, Scott J. Lusher, Maryann E. Martone, Albert Mons, Abel L. Packer, Bengt Persson, Philippe Rocca-Serra, Marco Roos, Rene van Schaik, Susanna-Assunta Sansone, Erik Schultes, Thierry Sengstag, Ted Slater, George Strawn, Morris A. Swertz, Mark Thompson, Johan van der Lei, Erik van Mulligen, Jan Velterop, Andra Waagmeester, Peter Wittenburg, Katherine Wolstencroft, Jun Zhao, and Barend Mons. 2016. The FAIR Guiding Principles for scientific data management and stewardship. *Scientific Data* 3 (2016), 160018. doi:10.1038/sdata.2016.18

[34] Meina Zhang and Alexander A. Sawchuk. 2012. USC-HAD: A daily activity dataset for ubiquitous activity recognition using wearable sensors. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*. ACM, New York, NY, USA, 1036–1043.

[35] Yexu Zhou, Tobias King, Haibin Zhao, Yiran Huang, Till Riedel, and Michael Beigl. 2024. MLP-HAR: Boosting Performance and Efficiency of HAR Models on Edge Devices with Purely Fully Connected Layers. In *Proceedings of the 2024 ACM International Symposium on Wearable Computers* (Melbourne VIC, Australia) *(ISWC '24)*. Association for Computing Machinery, New York, NY, USA, 133–139. doi:10.1145/3675095.3676624

[36] 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* (Cambridge, United Kingdom) *(ISWC '22)*. Association for Computing Machinery, New York, NY, USA, 89–93. doi:10.1145/3544794.3558467

Received 17 July 2025; accepted 21 July 2025