

Standardizing Your Training Process for Human Activity Recognition Models – A Comprehensive Review in the Tunable Factors^{*}

Yiran Huang^[0000–0003–3805–1375], Haibin Zhao^[0000–0001–7018–1159], Yexu Zhou^[0000–0002–8866–7998], Till Riedel^[0000–0003–4547–1984], and Michael Beigl^[0000–0001–5009–2327]

Telecooperation Office, Karlsruhe Institute of Technology, Karlsruhe, Germany
yhuang, hzhao, zhou, riedel, beigl@teco.edu

Abstract. In recent years, deep learning has emerged as a potent tool across a multitude of domains, leading to a surge in research pertaining to its application in the Wearable Human Activity Recognition (WHAR) domain. Despite the rapid development, concerns have been raised about the lack of standardization and consistency in the procedures used for experimental model training, which may affect the reproducibility and reliability of research results. In this paper, we provide an exhaustive review of contemporary deep learning research in the field of WHAR and collate information pertaining to the training procedure employed in various studies. Our findings suggest that a major trend is the lack of detail provided by model training protocols. Besides, to gain a clearer understanding of the impact of missing descriptions, we utilize a control variables approach to assess the impact of key tunable components (e.g., optimization techniques and early stopping criteria) on the inter-subject generalization capabilities of HAR models. With insights from the analyses, we define a novel integrated training procedure tailored to the WHAR model. Empirical results derived using five well-known WHAR benchmark datasets and three classical HAR model architectures demonstrate the effectiveness of our proposed methodology: in particular, there is a significant improvement in macro F1 Leave-One-Subject-Out (LOSO) Cross-Validation (CV) performance.

Keywords: human activity recognition · deep learning model.

1 Introduction

Wearable Human Activity Recognition (WHAR) is a process that aims to classify human actions within a specific temporal boundary based on measurements, e.g., acceleration, rotation speed, and geographical coordinates, gathered from personal digital devices [49]. It is central to numerous applications and services. For

^{*} Supported by the Ministry of Science, Research and the Arts Baden-Wuerttemberg as part of the SDSC-BW and by the Carl-Zeiss-Foundation as part of "stay young with robots" (JuBot) project.

instance, in the smart homes [36,5,9], accurately detecting a user’s activity, such as watching television, can enable the automation of derivative tasks like adjusting room light levels. Similarly, in industrial manufacturing [31,1,14], recognizing a worker’s current task could aid in suggesting sequential actions for workflow optimization. In the context of road safety [11,55,58], identifying a driver’s present driving behavior or mental states can catalyze measures to improve road safety. These applications place high-performance demands on a WHAR system, necessitating the guarantees of sufficient data and high-performance models.

Wearable technology has experienced significant growth in recent years, with nearly 500 million units shipped globally by the end of 2022, a 500% increase from the 82 million units shipped in 2016. Smartwatches are projected to be the most popular, accounting for 216 million units in 2022. Smart trackers and glasses follow, with 105 million and 32 million units, respectively. Equipped with various sensors, these devices offer advantages such as unobtrusive continuous data collection, direct user contact, and long battery life. Consequently, they provide detailed insights into human behavior, making wearables a key platform for developing applications that automatically recognize human activity.

On the other hand, since the introduction of AlexNet in 2012 [29], deep learning has made significant strides, yielding outstanding results across multiple domains. Early advancements in these domains have inspired developments in WHAR. These include the utilization of Convolution Neural Network (CNN) [30] for feature extraction [22], the integration of Long Short-Term Memory (LSTM) [17] with CNN for capturing temporal relationships (CNNLSTM) [38], and the incorporation of transformer [52] for managing information interaction between sensors [57]. In addition, newer technologies like attention mechanisms [52] are also leveraged in WHAR [60]. According to Google Scholar ¹, from 2021 to the present, a total of 1,660 academic articles have contained the keywords ”human activity recognition” and ”neural networks” in their titles.

Although these factors have contributed to the rapid growth of WHAR, they also pose challenges in terms of reproducing recent work and benchmarking research. There is a noticeable lack of attention given to explicating the process by which experimental models are trained. The training process of a model, encompassing the trajectory of model development from initialization to gaining expertise in a specific task, plays a pivotal role in determining the model’s final performance.

A comprehensive description of the training process should include the following components:

(i) **Dataset Description:** WHAR data is typically multimodal and may include data collected by several different sensors. This data may vary in terms of sampling frequency, data quality and position of wearing. Therefore, the dataset description is intended to provide a brief overview of the dataset, including aspects such as data source, number of features, sensor type and sensor position.

(ii) **Model Parameters Setting if present:** These parameters pertain to the specific architecture of the model, like the number of layers and neurons per

¹ <https://scholar.google.com>.

layer. Selecting the appropriate network architecture parameters impacts the model’s representational capability and training efficiency.

(iii) **Data Preprocessing:** WHAR data is often noisy and may contain missing data due to factors like the quality of the sensors and data storage and transmission methods. Data preprocessing refers to a series of operations that cleanse or transform raw data to make it suitable for model training before it is fed into the deep learning model. It also typically involves dividing the data set into training, validation and test sets for model training, tuning and evaluation. Due to the high subject dependency of WHAR data, the way the dataset is partitioned will greatly affect the results of model training and testing.

(iv) **Batch Size and Maximum Epoch Setting:** The former is the quantity of training samples used in each parameter update. Its selection will directly affect the efficiency and stability of training. Larger batches can speed up the training process, but may require more memory resources; while smaller batches can better control parameter updates, but the training process may become noisier and require more iterations to reach convergence. The latter refers to the number of epochs the dataset undergoes during the training process. Setting it too small may cause the model to fail to converge, while setting it too large may cause the model to overfit.

(v) **Optimizer Setting:** In the training of deep learning models, the optimizer plays a crucial role in guiding the update of network parameters by computing gradients via backpropagation, thereby aligning the network’s output more closely with the expected values. Different optimizers have distinct characteristics, and therefore, may be specifically selected for the different target domains [45]. Stochastic gradient descent (sGD) [44] is referred to as one of the most straightforward approaches, but may stuck in local minimum and requires more delicate hyperparameter tuning. Adam [28] is the most commonly used optimizer in WHAR field and can adaptively adjust the step size and momentum, but it is more sensitive to input noise. Adagrad [12] adjusts the learning rate and momentum by utilizing historical training information, making it more suitable in case the gradients are sparse.

(vi) **Learning rate strategy:** The learning rate is used to control the step size of the model parameters update. Its setting directly affects the efficiency of model training and the speed of convergence. When training a deep model, we usually set an initial value for the learning rate and then gradually adjust the learning rate through the scheduler during the learning process to improve the convergence efficiency and final performance of the model [3]. Step [40] is the most popular approach in WHAR field, which decreases the learning rate by a certain ratio after a predefined number of epochs. More advanced schedulers, such as CosineAnnealingLR (Cos) [35] can accelerate model convergence and reduce the risk of overfitting by warming-up the learning rate. Cosine annealing with warm restarts (Cos&Restart) [35] helps the model jump out of local minima by restarting it periodically to find the global optimal solution.

(vii) **Early stopping strategy:** Early-stopping [33] is a method that can mitigate model overfitting and reduce training time as well as resource con-

sumption. It involves monitoring model performance on a validation set (early stopping base) and stopping training once validation performance no longer improves within a certain number of epochs (patience).

(viii) **Final model selection strategy:** Strategies for selecting the end model, e.g., the last epoch model, model with the smallest evaluation loss, etc.

(ix) **Validation&Test setting:** Rigorous evaluation to measure model effectiveness.

(x) **Additional details:** These may pertain to other relevant aspects not covered by the above categories. For instance, certain training schemes may incorporate bespoke techniques, such as switching between different loss functions during the training phase.

By comprehensively articulating each stage of the training process, the intention of the training process description is to provide the reader with the ability to accurately replicate the experimental result described in the publication. Such documentation reinforces the transparency and integrity of the research, elements that are paramount in the field of deep learning science.

Nonetheless, our analysis points out that while nearly all publications deliver comprehensive details about dataset descriptions, model parameters setting, data preprocessing, and validation and test settings, there is a notable lack of exhaustive detail in other aspects.

Until now, the majority of review articles in the realm of WHAR have mainly concentrated on outlining various machine learning and deep learning architectures, with an emphasis on model design. Contrary to this trend, our research pivots its focus towards a comprehensive analysis of the model training process in the context of WHAR. To the best of our knowledge, this survey stands as a pioneering effort specifically targeting descriptions of the model training process. Through this exploration, we endeavor to shed light on the current state of model training process narratives in the relevant literature and proffer recommendations to guide future scholarly pursuits in this domain.

The key contributions of our work can be summarized below:

- To the best of our knowledge, we are the first in the WHAR field to write a review that focuses on the model training procedure in various research papers.
- This study delivers insights into how models have been trained in recent years. It identifies prevailing issues about model training in the field and highlights the potential consequences of these problems using control variables.
- In response to these identified issues, we propose a countermeasure and provide experimental evidence of its efficacy.
- We are releasing the proposed training procedure as an open-source package to aid in further research in this field.

Table 1. Summarize of the hyperparameters and the corresponding range in model training procedure.

Parameter	Range
Optimizer	{sGD [44], ADAM [28], Adadelta [56], Rmsprop}
Learning Rate	[$1e-1$, $1e-5$]
Weight Decay	[$1e-1$, $1e-8$]
Scheduler	{Step [51], LRPlateau [47], Cos [35], Cos+restart [35]}
Scheduler Base	{Train Loss, Validation Loss, Train Metric, Validation Metric}
Scheduler Patience (epoch)	[1, 100]
Batch Size	[16, 512]
Maximum Epoch	[10, 1500]
Early Stopping	{Yes, No}
Early Stopping Base	{Train Loss, Validation Loss, Train Metric, Validation Metric}
Early Stopping Patience (epoch)	[1,100]
Model Selection Base	{Last, Train Loss, Validation Loss, Train Metric, Validation Metric}

2 Search Strategy

We followed Preferred Reporting Items for Systematic review and Meta-Analysis Protocols (PRISMA-P) to select relevant and important articles for our research target. We completed this search using two protocols: the search protocol and the three-level exclusion protocol.

Search Protocol: We utilized Google Scholar, a free academic search engine provided by Google, as our search platform. It provides an extensive coverage of academic literature across numerous disciplines worldwide, such as journal articles, theses, conference papers, and more. Its database is regularly updated, ensuring access to the most recent academic literature. Google Scholar supports keyword-based searches to retrieve related articles. In alignment with our research goals, we initially used the following three keyword sets for our exploration: ("Human Activity Recognition" OR "HAR"), ("Machine learning" OR "Deep learning" OR "Neural network"), and ("Wearable" OR "Smartphone" OR "Smartwatch" OR "Smartglasses" OR "Smartband"), combined with the platform's limited search statements. In light of our objective to analyze the latest developments, we limited the search time frame to 2021 through June 2023. Through this procedure, we sourced a total of 4,960 articles.

Three-Level Exclusion Protocol: We prioritized relevance and ranked papers by their impact and subsequently excluded duplicates, non-English articles, non-conference/journal papers, and those lacking references. We filtered out articles whose titles did not align with our research purpose. This initial inspection helped us narrow down to 274 articles. By reading the abstracts of these 274 articles, we removed survey articles and those that focused on other types of Human Activity Recognition (HAR) tasks. The second filtering phase yielded

208 papers. On downloading these papers and swiftly reviewing their contents, we discarded those primarily focusing on machine learning models, data collection, and data processing. The third and final filtering round culminated in a total of 102 papers.

3 Observation

In deep learning, beyond the inherent model parameters optimized using gradient descent, several hyperparameters critically influence a model’s performance. These include the number of layers, kernel sizes, and training aspects like optimizer selection and batch size. These hyperparameters are not directly learned through training but profoundly affect the model’s structural complexity, training speed, and overall efficacy.

Table 2. Summarize of the hyperparameters tuning result.

Dataset	Optimizer	Learning Rate	Batch Size	Epoch
[13]	$1e - 3$	/	/	50
[8]	Adam	$1e - 3$	64	50
[23]	/	$1e - 3$	512	/
[10]	Adam	$1e - 3$	400	120
[42]	/	/	50	/
[53]	/	$1e - 4$	400	/
[39]	/	$1e - 6$	/	/
[48]	Adam	$1e - 2$	64	/
[34]	/	$3e - 4/6e - 4/8e - 5^2$	/	/
[23]	/	$1e - 3$	512	/
[27]	Adam	/	/	/
[6]	/	$1e - 3$	/	/
[15]	/	$1e - 3$	64	50/
[2]	/	$1e - 3$	64	75
[24]	/	$1e - 3$	64	50
[37]	Adam	$2.5e - 3$	64	50

The description of the training process in a research paper is essentially a record of the values taken for these parameters. It ensures that the entire training process of the model is unambiguous. Table 1 summarizes the hyperparameters that describe the training process, along with their common ranges within the WHAR domain as derived from 102 studied papers.

For a specific task, hyperparameter tuning aims to identify the optimal configuration of hyperparameters leading to the best model performance e.g. through Bayesian Optimization [54], reinforcement learning [20] or Monte Carlo [19,18]. However, as demonstrated in Table 1, the search space for training model hyperparameters is vast. It leads to substantial computational costs associated with hyperparameter tuning, especially for tasks like WHAR that involve large datasets and complex feature extraction using complex models. Individual model

training on the classic oppo [7] dataset can take approximately one hour on Tesla P100 ³. Performing systematic hyperparameter searches incurs significant time and computational resource expenses. Furthermore, considering typical limitations in paper length and the focus of WHAR-related papers on introducing new model structures, algorithms, or theories, hyperparameter tuning is generally not the primary focus. Among the 102 papers we analyzed, only 16 of them (15.7%) involved hyperparameter tuning. Table 2 summarizes their tuning results.

We observe that all these works only explored a subset of the hyperparameter space. Most concentrated on seeking for the learning rate and batch size, followed by the number of epochs and the choice of optimizer. There is a lack of exploration in other hyperparameter dimensions. Additionally, we found that the results of the search efforts were relatively consistent regarding optimizer and learning rate. All searches pertaining to the optimizer ultimately converged on selecting the Adam optimizer [28]. The majority of chosen learning rates were around $1e - 3$, accounting for 64.3% of similar searches. Epoch values mostly centered around 60. Batch size results were polarized, with half of the studies selecting a size of 64, while the other half opted for batch sizes above 400. This apparent inconsistency in the search results further emphasizes the significance of hyperparameters. It underscores that the required hyperparameters can vary significantly based on differences in models and data.

Apart from the 16 articles that engaged in hyperparameter tuning, the remaining 86 articles addressed the potential adverse effects of neglecting hyperparameter tuning through various strategies. Some adopted parameter settings from similar works, while the majority rely on empirical choices [59,21]. Figure 1 illustrates the hyperparameter setting made by these articles in terms of the four mentioned parameters. Similar to the tuning results, these studies also predominantly favored the Adam optimizer [28]. Concerning learning rates, in addition to $1e - 3$, $1e - 4$ was also a popular choice. The setting of the epoch number deviated from the tuning results. Most of the works leaned towards larger epoch values. This trend could be attributed to the application of techniques like early stopping. In early stopping scenarios, even if a large number of epochs is set, training can be terminated ahead of time if specific conditions are met. For instance, if the validation loss ceases to decrease, training can halt prematurely. These strategies aim to mitigate the negative impact of neglecting exhaustive hyperparameter optimization. While they might not yield the optimal performance, they offer a practical approach given the resource constraints, limited paper space, and the specific focuses of the papers in the WHAR domain.

Additionally, we observed in Figure 1 that the "none" category has a high occurrence rate across all plots. This signifies that, using batch size as an example, in these papers, 45% of them did not provide a description of the batch size. This indicates an incompleteness in the descriptions of their model training process. This scenario is likewise seen in other hyperparameter plots.

Figure 2 presents a summary of the descriptions related to parameter schedulers across all reviewed papers. It was found that 81% of the articles lacked

³ The performance is tested with <https://colab.research.google.com/>.

descriptions about schedulers. Moreover, only 6% of the articles described early stopping procedures, and only 9% outlined the final model selection process.

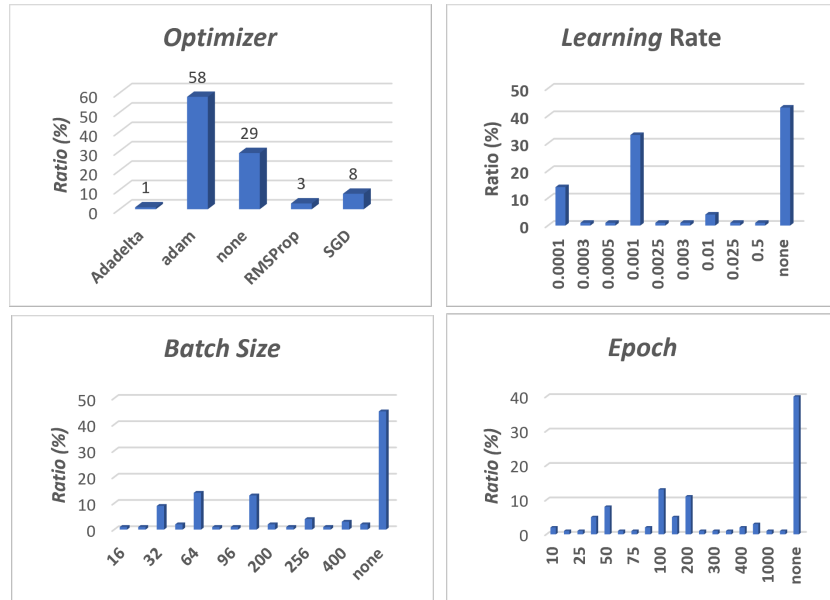


Fig. 1. Distribution on the four hyperparameters Optimizer, Learning Rate, Batch Size, and Epoch across all reviewed papers. 'none' indicates that there is no corresponding description in the paper.

Intuitively, it is clear that challenges due to descriptive deficiencies can significantly affect the reliability and reproducibility of the corresponding articles, underscoring the need for a standardized and comprehensive description of model training. However, the actual strength of the impact of the parameters describing the model training process, for instance, optimizer settings (type, learning rate, weight decay), early stopping strategies and model selection strategies, on the WHAR model end performance remains an open question, which is discussed in the next section.

4 Control-Variates Analysis

This section aims to investigate the independent impact of the aforementioned factors on the final model performance in WHAR context. To achieve this, we employ the control-variates method and run experiments with CNNLSTM [38] model on the HAPT [43] dataset. The baseline training procedure is derived from the collected papers by identifying the most frequently utilized hyperparameters (as described in section 3) in the training procedure. Specifically, an Adam optimizer [28] is employed with an initial learning rate set at $1e - 3$ and

no weight decay. Cross-entropy is selected as the loss function, along with batch training set to a batch size of 64. The maximum number of epochs is capped at 60. A step scheduler is also utilized to reduce the learning rate by a factor of 0.1 every 10 epochs. For early stopping, we use a 10-epoch patience also based on validation loss. The test model selected runs on the least validation loss post-training. Figure 3 visualizes the validation loss during the training in control-variates experiment for hyperparameter: optimizer, learning rate, weight decay and batch size.

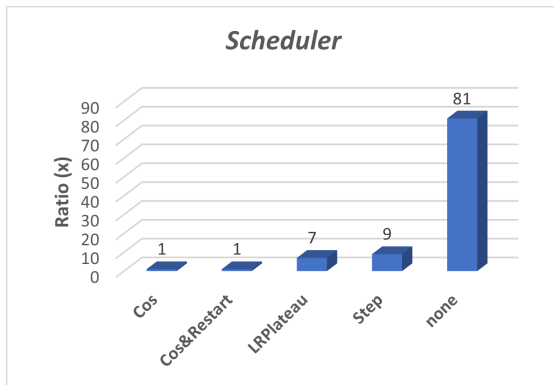


Fig. 2. Distribution of choices for scheduler hyperparameters across all reviewed papers.

Figure 3(a) depicts the impact of diverse optimizers (i.e., sGD [44], Adam [12], and Adam [28]) on model training. Our analysis finds notable variance in results across these optimizers. Of the three, Adam [28] yields the best performance. Conversely, sGD [44] does not converge due to its hyperparameter sensitivity and inappropriate initialization.

Figure 3(b) illustrates the effect of different initial training rates. Notably, although the baseline Adam optimizer [28] can adaptively adjust the step size, which is almost equivalent to adjusting the learning rate, the initial learning rate still plays a crucial role in the model training. In particular, setting an improper initial learning rate may hinder the model from converging (e.g., the black curve), leading to suboptimal performance. In addition, the best model performance is achieved by setting the initial learning rate $1e - 3$.

Despite the strong capability of weight decay for improving model generalization in other domains [32,16], Figure 3(c) draws a different conclusion in WHAR domain. We can observe that the utilization of different weight decays does not significantly influence the validation loss. An excessive weight decay may even increase the validation loss, conversely, setting the weight to $1e - 4$ seems to facilitate a more stable training process.

Figure 3(d) summarizes the validation loss with different batch sizes. Consistent with the conclusion from another independent work [26], a smaller batch

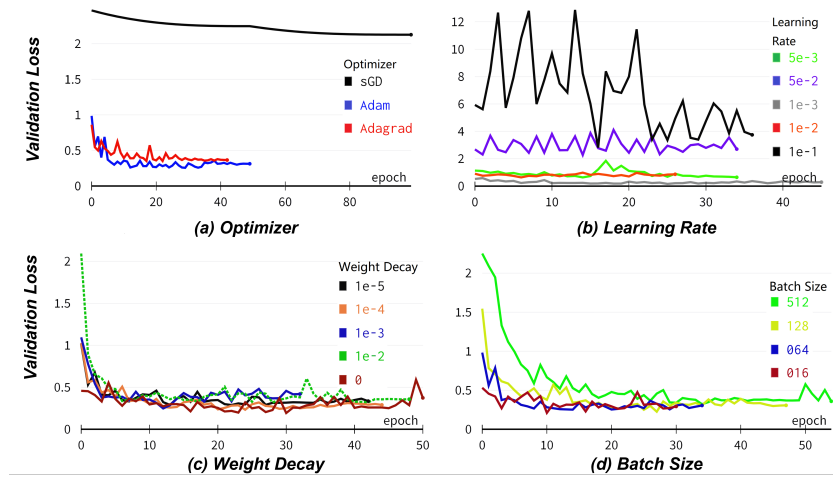


Fig. 3. Validation loss during model training in control-variates experiment.

size leads to the better model generalization. However, as reported in Figure 1, it is not a common option in recent WHAR articles.

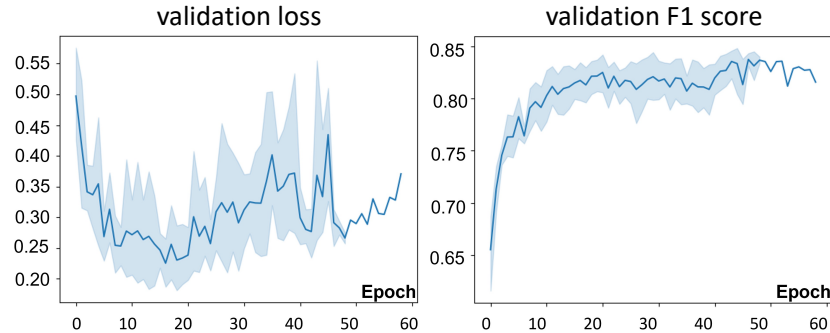


Fig. 4. Mean and standard deviation of the validation loss (left) and macro F1 score (right) of the models trained on HAPT dataset.

Figure 4 depicts the validation loss and macro F1 score over the epoch during the training on HAPT [43] dataset. The solid line represents the mean F1 score w.r.t. subjects, while the shading indicates the corresponding variance. Our analysis reveals that the validation loss decreases gradually and reaches its minimum at approximately the 15th epoch. However, the loss tends to increase progressively thereafter. In contrast, the F1 score continues to improve gradually even after the 15th epoch. This implies that employing validation loss for early-stopping may hinder the model from reaching a higher F1 score, which is the actual metric for evaluating the model performance. Additionally, the model with early-stopping does not perform as well as the model obtained by terminating the training at the maximal training epochs.

Based on the analysis of the control-variates approach, we can conclude that all the aforementioned factors have a discernible impact on the final performance of WHAR models. Therefore, their inclusion in academic papers is crucial to ensuring the reproducibility of experiments.

Table 3. Summary of the datasets used in the experiment. The abbreviations acc, gyro, mag denote 3d accelerometers, gyroscopes and magnetometers, respectively.

Name	#Subjects	#Channels	#Length	#Classes	Sensors type	Freq
DSADS	8	45	126	19	acc, gyro, mag	25 Hz
HAPT	30	6	128	12	acc, gyro	50 Hz
OPPO	4	77	30	18	acc, gyro, mag	30 Hz
PAMAP2	9	18	168	12	acc, gyro	100 Hz
RWHAR	15	21	128	8	acc	50 Hz

5 Experiment

It has been observed that frequently used training hyperparameters derived from reviewed articles do not yield an optimal WHAR model. Accordingly, without using hyperparameter tuning, how should we set the parameters of the training process to get relatively close to optimal results? To address this, we propose a hypothesis: by employing the optimal setting for each factor (determined during the control-variates analysis) for model training (denoted as *new*), the performance of the trained model could be enhanced, surpassing the effectiveness of the baseline used in section 4 (denoted as *comm*). Specifically, our approach involves using an Adam optimizer [28] with an initial learning rate of $1e - 3$ and a weight decay of $1e - 4$. In addition, a Cosine Scheduler [35] is employed, along with the application of the cross-entropy loss function [46]. We utilize batch training with a batch size of 16 and apply the F1 score on the validation set for early stopping, with a patience of 30 epochs. After training, the optimal model is chosen based on the best F1 score on the validation set. In this section, we conduct an experiment to test this hypothesis.

Benchmark WHAR datasets. To ensure experimental fairness, we conducted experiments on five most widely used benchmark datasets, i.e., DSADS [4], HAPT [43], OPPO [7], PAMAP2 [41], and RW [50]. As summarized in Table 3, these datasets differ significantly in terms of sensor types, sampling rates, number of subjects, number of channels, sequence length and application scenarios. As data preprocessing, we applied sliding windows to split data in segments, specifically, we employed a 50% overlap between adjacent windows for the training and validation sets. While for the test set, we used a 90% overlap between adjacent windows [25].

Model To ensure the generalizability of our new training process, three models with different structures are utilized as our target models, namely, CNN-based model (MCNN) [30], hybrid CNN-LSTM model (CNNLSTM) [38], and transformer-based model (Transformer) [57].

Evaluation and Metric To evaluate the performance of the final obtained models and demonstrate their generalizability across subjects, we employ the

LOSO-CV method on all datasets. In response to the category imbalance present within the WHAR data, we apply the F1 score as the evaluation metric. Moreover, to ensure the robustness of our results, we repeat the experiment five times using different random seeds and report the average results for all experiments.

Baseline We identified the most frequently utilized training procedure in the collected academic papers and use them as our baseline (comm). Concretely, we use an Adam optimizer [28] with an initial learning rate of $1e-3$ without weight decay. We use the cross-entropy loss to guide the batch training with a batch size of 256. We employ an LRScheduler that reduces the learning rate by a factor of 0.1 based on the validation loss with patience of 10 epochs. We apply early-stopping with the patience of 30 epochs also on validation loss. Finally, we consider the model with the lowest validation loss as the optimal model.

5.1 Result

Table 4 presents the results of our experiments. We can observe that, the new training procedure achieved significantly better results compared to the baseline procedure on all datasets and across all models. This enhancement confirms the effectiveness of the proposed training procedure. In addition, it is noteworthy to mention that, this enhancement can be gained without any supplementary demands on the training resource and any increasing model inference time.

Table 4. LOSO-CV performance (mean F1 score) comparison.

Model	Method	DSADS	HAPT	OPPO	PAMAP2	RW
MCNN	comm	0.801	0.779	0.405	0.708	0.691
	new	0.865	0.802	0.432	0.734	0.747
CNNLSTM	comm	0.854	0.806	0.392	0.735	0.706
	new	0.900	0.820	0.395	0.764	0.768
Transformer	comm	0.812	0.767	0.402	0.654	0.598
	new	0.872	0.805	0.440	0.758	0.731

6 Conclusion and future work

In this study, we highlight the prevalent trend that, the majority of academic papers are deficient in providing a comprehensive description of the model training process. Employing the control-variables analysis, we affirm that, these overlooked factors exert a significant influence on the model performance. Therefore, the lack of these descriptions can markedly affect the reproducibility of the achievements reported in those academic papers. In addition, based on the control-variables analysis, we propose a new training procedure and test its effectiveness by conducting extensive experiments on five widely used benchmark datasets using three models with different architectures. The experiments indicate that, the new training procedure can achieve significantly better results compared to the most common training procedure on all datasets and all models.

References

1. Aehnelt, M., Gutzeit, E., Urban, B., et al.: Using activity recognition for the tracking of assembly processes: Challenges and requirements. *WOAR* **2014**, 12–21 (2014)
2. Alawneh, L., Alsarhan, T., Al-Zinati, M., Al-Ayyoub, M., Jararweh, Y., Lu, H.: Enhancing human activity recognition using deep learning and time series augmented data. *Journal of Ambient Intelligence and Humanized Computing* pp. 1–16 (2021)
3. Arbib, M.A.: *The Handbook of Brain Theory and Neural Networks*. MIT press (2003)
4. Barshan, B., Altun, K.: *Daily and Sports Activities*. UCI Machine Learning Repository (2013), DOI: <https://doi.org/10.24432/C5C59F>
5. Bianchi, V., Bassoli, M., Lombardo, G., Fornacciari, P., Mordonini, M., De Munari, I.: Iot wearable sensor and deep learning: An integrated approach for personalized human activity recognition in a smart home environment. *IEEE Internet of Things Journal* **6**(5), 8553–8562 (2019)
6. Buffelli, D., Vandin, F.: Attention-based deep learning framework for human activity recognition with user adaptation. *IEEE Sensors Journal* **21**(12), 13474–13483 (2021)
7. Chavarriaga, R., Sagha, H., Calatroni, A., Digumarti, S.T., Tröster, G., Millán, J.d.R., Roggen, D.: The Opportunity Challenge: A Benchmark Database for On-body Sensor-Based Activity Recognition. *Pattern Recognition Letters* **34**(15), 2033–2042 (2013)
8. Dirgová Luptáková, I., Kubovčík, M., Pospíchal, J.: Wearable sensor-based human activity recognition with transformer model. *Sensors* **22**(5), 1911 (2022)
9. Du, Y., Lim, Y., Tan, Y.: A novel human activity recognition and prediction in smart home based on interaction. *Sensors* **19**(20), 4474 (2019)
10. Dua, N., Singh, S.N., Semwal, V.B., Challa, S.K.: Inception inspired cnn-gru hybrid network for human activity recognition. *Multimedia Tools and Applications* **82**(4), 5369–5403 (2023)
11. Duan, S., Yu, T., He, J.: Widriver: Driver activity recognition system based on wifi csi. *International Journal of Wireless Information Networks* **25**, 146–156 (2018)
12. Duchi, J., Hazan, E., Singer, Y.: Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of machine learning research* **12**(7) (2011)
13. Gumaei, A., Hassan, M.M., Alelaiwi, A., Alsalman, H.: A hybrid deep learning model for human activity recognition using multimodal body sensing data. *IEEE Access* **7**, 99152–99160 (2019)
14. Günther, L.C., Kärcher, S., Bauernhansl, T.: Activity recognition in manual manufacturing: Detecting screwing processes from sensor data. *Procedia CIRP* **81**, 1177–1182 (2019)
15. Hamad, R.A., Kimura, M., Yang, L., Woo, W.L., Wei, B.: Dilated causal convolution with multi-head self attention for sensor human activity recognition. *Neural Computing and Applications* **33**, 13705–13722 (2021)
16. Hinton, G.E., Salakhutdinov, R.R.: Reducing the Dimensionality of Data with Neural Networks. *science* **313**(5786), 504–507 (2006)
17. Hochreiter, S., Schmidhuber, J.: Long Short-Term Memory. *Neural computation* **9**(8), 1735–1780 (1997)
18. Huang, Y., Schaal, N., Hefenbrock, M., Zhou, Y., Riedel, T., Beigl, M.: Mxai: Local model-agnostic explanation as two games. In: *2023 International Joint Conference on Neural Networks (IJCNN)*. pp. 01–08. IEEE (2023)

19. Huang, Y., Zhou, Y., Hefenbrock, M., Riedel, T., Fang, L., Beigl, M.: Automatic feature engineering through monte carlo tree search. In: Joint European Conference on Machine Learning and Knowledge Discovery in Databases. pp. 581–598. Springer (2022)
20. Huang, Y., Zhou, Y., Hefenbrock, M., Riedel, T., Fang, L., Beigl, M.: Universal distributional decision-based black-box adversarial attack with reinforcement learning. In: International Conference on Neural Information Processing. pp. 206–215. Springer (2022)
21. Huang, Y., Zhou, Y., Riedel, T., Fang, L., Beigl, M.: randomhar: Improving ensemble deep learners for human activity recognition with sensor selection and reinforcement learning. arXiv preprint arXiv:2307.07770 (2023)
22. Ito, C., Shuzo, M., Maeda, E.: Cnn for Human Activity Recognition on Small Datasets of Acceleration and Gyro Sensors Using Transfer Learning. In: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers. pp. 724–729 (2019)
23. Jain, Y., Tang, C.I., Min, C., Kawsar, F., Mathur, A.: Collossl: Collaborative self-supervised learning for human activity recognition. Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies **6**(1), 1–28 (2022)
24. Javed, A., Faheem, R., Asim, M., Baker, T., Beg, M.: A smartphone sensors-based personalized human activity recognition system for sustainable smart cities. *sustain. cities. soc.* **71**, 102970 (2021) (2021)
25. Jordao, A., Nazare Jr, A.C., Sena, J., Schwartz, W.R.: Human Activity Recognition Based on Wearable Sensor Data: A Standardization of the State-of-the-art. arXiv preprint arXiv:1806.05226 (2018)
26. Keskar, N.S., Mudigere, D., Nocedal, J., Smelyanskiy, M., Tang, P.T.P.: On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima. arXiv preprint arXiv:1609.04836 (2016)
27. Khatun, M.A., Yousuf, M.A., Ahmed, S., Uddin, M.Z., Alyami, S.A., Al-Ashhab, S., Akhdar, H.F., Khan, A., Azad, A., Moni, M.A.: Deep cnn-lstm with self-attention model for human activity recognition using wearable sensor. *IEEE Journal of Translational Engineering in Health and Medicine* **10**, 1–16 (2022)
28. Kingma, D.P., Ba, J.: Adam: A Method for Stochastic Optimization. arXiv preprint arXiv:1412.6980 (2014)
29. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems* **25** (2012)
30. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet Classification with Deep Convolutional Neural Networks. *Communications of the ACM* **60**(6), 84–90 (2017)
31. Kubota, A., Iqbal, T., Shah, J.A., Riek, L.D.: Activity recognition in manufacturing: The roles of motion capture and semg+ inertial wearables in detecting fine vs. gross motion. In: 2019 International Conference on Robotics and Automation (ICRA). pp. 6533–6539. IEEE (2019)
32. LeCun, Y., Bottou, L., Bengio, Y., Haffner, P.: Gradient-Based Learning Applied to Document Recognition. *Proceedings of the IEEE* **86**(11), 2278–2324 (1998)
33. LeCun, Y., Bottou, L., Orr, G.B., Müller, K.R.: Efficient Backprop. In: *Neural networks: Tricks of the trade*, pp. 9–50. Springer (2002)
34. Li, Y., Wang, L.: Human activity recognition based on residual network and bilstm. *Sensors* **22**(2), 635 (2022)
35. Loshchilov, I., Hutter, F.: SGDR: Stochastic Gradient Descent with Warm Restarts. arXiv preprint arXiv:1608.03983 (2016)

36. Mehr, H.D., Polat, H.: Human activity recognition in smart home with deep learning approach. In: 2019 7th International Istanbul Smart Grids and Cities Congress and Fair (ICSG). pp. 149–153. IEEE (2019)
37. Mohsen, S., Elkaseer, A., Scholz, S.G.: Industry 4.0-oriented deep learning models for human activity recognition. *IEEE Access* **9**, 150508–150521 (2021)
38. Mutegeki, R., Han, D.S.: A CNN-LSTM Approach to Human Activity Recognition. In: 2020 international conference on artificial intelligence in information and communication (ICAIIIC). pp. 362–366. IEEE (2020)
39. Ni, Q., Fan, Z., Zhang, L., Zhang, B., Zheng, X., Zhang, Y.: Daily activity recognition and tremor quantification from accelerometer data for patients with essential tremor using stacked denoising autoencoders. *International Journal of Computational Intelligence Systems* **15**(1), 1 (2022)
40. Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al.: PyTorch: An Imperative Style, High-Performance Deep Learning Library. *Advances in neural information processing systems* **32** (2019)
41. Reiss, A., Stricker, D.: Introducing A New Benchmarked Dataset for Activity Monitoring. In: 2012 16th international symposium on wearable computers. pp. 108–109. IEEE (2012)
42. Requena, S.H., Nieto, J.M.G., Popov, A., Delgado, I.N.: Human activity recognition from sensorised patient’s data in healthcare: A streaming deep learning-based approach. *IJIMAI* **8**(1), 23–37 (2023)
43. Reyes-Ortiz, J.L., Oneto, L., Samà, A., Parra, X., Anguita, D.: Transition-Aware Human Activity Recognition Using Smartphones. *Neurocomputing* **171**, 754–767 (2016)
44. Robbins, H., Monro, S.: A Stochastic Approximation Method. *The annals of mathematical statistics* pp. 400–407 (1951)
45. Rosenblatt, F.: The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain. *Psychological review* **65**(6), 386 (1958)
46. Shannon, C.E.: A mathematical theory of communication. *ACM SIGMOBILE mobile computing and communications review* **5**(1), 3–55 (2001)
47. Simonyan, K., Zisserman, A.: Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556* (2014)
48. Singla, R., Mittal, S., Jain, A., Gupta, D.: Convlstm for human activity recognition. In: International Conference on Innovative Computing and Communications: Proceedings of ICICC 2021, Volume 2. pp. 335–344. Springer (2022)
49. Strackiewicz, M., James, P., Onnela, J.P.: A systematic review of smartphone-based human activity recognition methods for health research. *NPJ Digital Medicine* **4**(1), 148 (2021)
50. Szttyler, T., Stuckenschmidt, H.: On-body Localization of Wearable Devices: An Investigation of Position-Aware Activity Recognition. In: 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom). pp. 1–9. IEEE (2016)
51. Tang, Y., Zhang, L., Min, F., He, J.: Multiscale deep feature learning for human activity recognition using wearable sensors. *IEEE Transactions on Industrial Electronics* **70**(2), 2106–2116 (2022)
52. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention Is All You Need. *Advances in neural information processing systems* **30** (2017)
53. Verma, U., Tyagi, P., Aneja, M.K.: Multi-branch cnn gru with attention mechanism for human action recognition. *Engineering Research Express* (2023)

54. Wu, J., Chen, X.Y., Zhang, H., Xiong, L.D., Lei, H., Deng, S.H.: Hyperparameter optimization for machine learning models based on bayesian optimization. *Journal of Electronic Science and Technology* **17**(1), 26–40 (2019)
55. Xing, Y., Lv, C., Wang, H., Cao, D., Velenis, E., Wang, F.Y.: Driver activity recognition for intelligent vehicles: A deep learning approach. *IEEE transactions on Vehicular Technology* **68**(6), 5379–5390 (2019)
56. Zeiler, M.D.: Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701* (2012)
57. Zhang, Z., Wang, W., An, A., Qin, Y., Yang, F.: A Human Activity Recognition Method Using Wearable Sensors Based on Convtransformer Model. *Evolving Systems* pp. 1–17 (2023)
58. Zhao, H., Scholz, A., Beigl, M., Ni, S., Singaraju, S.A., Aghassi-Hagmann, J.: Printed Electrodermal Activity Sensor with Optimized Filter for Stress Detection. In: *Proceedings of the 2022 ACM International Symposium on Wearable Computers*. pp. 112–114 (2022)
59. Zhao, H., Zhou, Y., Riedel, T., Hefenbrock, M., Beigl, M.: Improving human activity recognition models by learnable sparse wavelet layer. In: *Proceedings of the 2022 ACM International Symposium on Wearable Computers*. pp. 84–88 (2022)
60. Zhou, Y., Zhao, H., Huang, Y., Riedel, T., Hefenbrock, M., Beigl, M.: Tinyhar: A Lightweight Deep Learning Model Designed for Human Activity Recognition. In: *Proceedings of the 2022 ACM International Symposium on Wearable Computers*. pp. 89–93 (2022)