A Study on Hyperparameters Configurations for an Efficient Human Activity Recognition System

Paulo J.S. Ferreira*
João Mendes-Moreira*
João M.P. Cardoso*
up201305617@fe.up.pt
jmoreira@fe.up.pt
jmpc@fe.up.pt
INESC TEC, Faculty of Engineering, University of Porto
Porto, Portugal

ABSTRACT

Human Activity Recognition (HAR) has been a popular research field due to the widespread of devices with sensors and computational power (e.g., smartphones and smartwatches). Applications for HAR systems have been extensively researched in recent literature, mainly due to the benefits of improving quality of life in areas like health and fitness monitoring. However, since persons have different motion patterns when performing physical activities, a HAR system would need to adapt to the characteristics of the user in order to maintain or improve accuracy. Mobile devices, such as smartphones, used to implement HAR systems, have limited resources (e.g., battery life). They also have difficulty adapting to the device's constraints to work efficiently for long periods. In this work, we present a *k*NN-based HAR system and an extensive study of the influence of hyperparameters (window size, overlap, distance function, and the value of k) and parameters (sampling frequency) on the system accuracy, energy consumption, and response time. We also study how hyperparameter configurations affect the model's performance for the users and the activities. Experimental results show that adapting the hyperparameters makes it possible to adjust the system's behavior to the user, the device, and the target service. These results motivate the development of a HAR system capable of automatically adapting the hyperparameters for the user, the device, and the service.

CCS CONCEPTS

• **Human-centered computing → Mobile devices**.

KEYWORDS

human activity recognition, *k*NN, energy consumption, adaptability, smartphones, hyperparameters, sampling frequency

*All authors contributed equally to this research.

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for third-party components of this work must be honored. For all other uses, contact the owner/author(s).
iWOAR 2023, September 21–22, 2023, Lübeck, Germany
© 2023 Copyright held by the owner/author(s).
ACM ISBN 979-8-4007-0816-9/23/09.
https://doi.org/10.1145/3615834.3615851

ACM Reference Format:
Paulo J.S. Ferreira, João Mendes-Moreira, and João M.P. Cardoso. 2023. A Study on Hyperparameters Configurations for an Efficient Human Activity Recognition System. In *8th international Workshop on Sensor-Based Activity Recognition and Artificial Intelligence (iWOAR 2023), September 21–22, 2023, Lübeck, Germany*. ACM, New York, NY, USA, 6 pages. https://doi.org/10.1145/3615834.3615851

1 INTRODUCTION

Recently, Human Activity Recognition (HAR) [5, 8, 9, 11, 15] has been a popular research field due to the wide spread of mobile devices with sensors (e.g., smartphones and smartwatches). HAR aims at recognizing human activities at runtime through the analyses of sensing data and other observations acquired by in situ devices [25].

Figure 1 shows the two phases of a Machine Learning (ML) based HAR system: the offline training view (Preparation Phase), and the online classification view (Use Phase). The Preparation Phase is used to train the model to be later used in the Use Phase and includes dataset preparation, feature selection and extraction, normalization, and training [21]. In the Use Phase, which typically occurs online, the trained classifiers infer user activities. This Phase includes data acquisition, feature extraction, normalization, and classification [21]. In this phase, the data are acquired directly from the sensors of the target device. This phase can also include an incremental learning step.

In order to be able to recognize different activities accurately, a HAR system typically uses ML algorithms capable of inferring activities from the sensor data. One of the most used ML algorithms in HAR is *k*-Nearest Neighbours (*k*NN) [4]. Besides, *k*NN has also been proved to obtain very high accuracy in the HAR field (see, e.g., [2, 5, 12, 13, 16, 17, 20, 22, 24]). Being *k*NN an instance-based lazy learning algorithm, not requiring a computationally expensive training stage, it facilitates the implementation of incremental/online learning HAR systems in mobile devices. With *k*NN, online/incremental learning only needs to add, remove, or update the instances in the *k*NN memory. Updating *k*NN is fast and requires low energy consumption because there is no need to retrain the algorithm, which is essential in devices with energy limitations [5].

Typically, an ML algorithm has two types of parameters: model parameters and hyperparameters [14]. Model Parameters are used inside the model and are estimated or learned from the data as a part of the learning process. On the other hand, hyperparameters

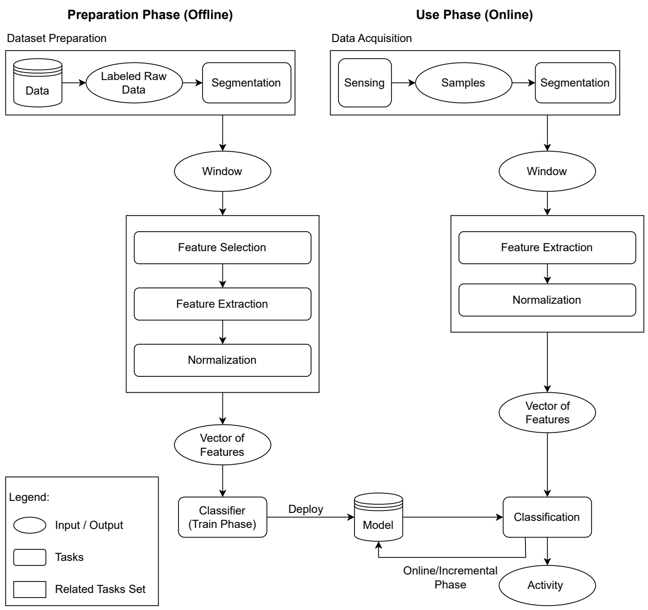

Figure 1: Block diagram of the HAR system considered.

are external parameters that are not part of the model and thus can not be predicted from the data set but can be configured by subject matter experts or by trial and error until an acceptable accuracy is achieved. Hyperparameters control the learning process and can have wildly varying effects on the resulting model and its performance [3]. For the same training dataset, with different hyperparameters, an ML algorithm might learn models with significantly different performance on the testing dataset [23].

This paper studies the influence of hyperparameters on a HAR system's optimal performance, specifically analyzing window size, overlap, k value, distance function, and sampling frequency in a kNN-based approach. Accuracy, response time (inference time), and energy consumption are evaluated as performance metrics. Prompt response time is crucial for time-sensitive applications that require real-time activity recognition, while energy consumption is vital for mobile and wearable devices with limited battery life. The results of this study provide valuable insights into the optimal hyperparameter configurations for HAR systems, contributing to the design of more efficient, effective, and adaptable systems for real-world applications. We present in [6] an extended version of this study.

2 RELATED WORK

HAR is a research field focused on automatically identifying and classifying human activities from sensor data. ML algorithms are commonly used in HAR systems for activity classification. Hyperparameters are essential for building accurate HAR models. They can significantly influence the model's performance, so choosing them carefully is important. As hyperparameters affect the algorithms' performance in terms of prediction and computation, many researchers studied their impact on HAR systems. Below are some examples of HAR studies that address hyperparameters.

For example, Garcia et al. [7] study the impact of window size and overlapping. They used the PAMAP2 HAR dataset and built an ensemble classifier with kNN, VFDT, and Naive Bayes algorithms. The study explored variations in window size (from 100 to 1000 samples with increments of 100) and overlap (from 0.0 to 0.9 with increments of 0.1). Evaluation metrics included accuracy, energy consumption, and execution time. They use the ODROID-XU+E6 board to measure energy consumption. Smaller window sizes showed lower accuracy, while larger sizes improved accuracy. The overlap factor had fluctuations in accuracy, with optimal results between 0.1 and 0.5. Smaller windows consumed less energy and had shorter execution times due to reduced feature calculation effort. Increasing the overlap factor raised energy consumption due to more calculations and classifications. Higher accuracies were associated with increased energy consumption.

Although not considered a hyperparameter, sampling frequency plays a crucial role in the performance of a HAR system. It directly affects inference time, energy consumption, and the performance of machine learning algorithms. Higher sampling frequencies, meaning more samples per second, generally increase energy consumption. Several studies have addressed the impact of sampling frequency on HAR systems.

For example, Zheng et al. [26] evaluate the impact of the sampling frequency on accuracy and energy consumption with SVM as an inference algorithm. They use a dataset collected at 1 Hz, 5 Hz, 10 Hz, and 50 Hz sampling frequencies. The data consists of the following activities: Sitting, Standing, Walking, Running, Upstairs, and Downstairs. The results show that the accuracy has only improved slightly with the sampling rate increase from 1 Hz to 50 Hz. In terms of energy consumption, there is an increase with increased sampling frequency, which is more significant when the sampling rate changes from 10 Hz to 50 Hz.

3 EXPERIMENTAL SETUP

Dataset: The *PAMAP2* [18, 19] HAR dataset contains sensor data from 9 different users and 12 different activities. Data were collected from 3 Inertial Measurement Units (IMU) positioned in different body areas (wrist, chest, and ankle), at a sampling frequency of 100 Hz. Each IMU has 3 3-axis embedded sensors: an accelerometer, a gyroscope, and a magnetometer. The activities are organized as basic activities (walking, running, nordic walking, and cycling); posture activities (lying, sitting, and standing); everyday activities (ascending and descending stairs); household (ironing and vacuum cleaning), and fitness activities (rope jumping).

Embedded computing platform: The energy consumption and response time values presented herein report measurements of the execution of the HAR system in an ODROID-XU+E6 [1] board.

Evaluation: To measure the performance of the models, we use the following ML metrics: accuracy and F1-Score. We also measure the response time and energy consumption. The values for time and energy include the following phases: reading data from the files, feature extraction, and inference. We use the fANOVA (Functional ANOVA (Analysis of Variance)) [2] [10] algorithm to

[1] https://www.hardkernel.com/
[2] https://www.automl.org/ixautoml/fanova/

assess the importance of the selected hyperparameters. We follow the leave-one-subject-out (LOSO) approach.

Feature Extraction: Using fixed-size sliding windows, we extracted 10 features for each 3D sensor: x-axis Mean, y-axis Mean, z-axis Mean, Mean of the sum of the x, y, and z axes, x-axis Standard Deviation, y-axis Standard Deviation, z-axis Standard Deviation, x and y axes Correlation, x and z axes Correlation, and y and z axes Correlation. The features are extracted from 9 sensors: 3 sensors (accelerometer, gyroscope, and magnetometer) for each body placement (wrist, chest, and ankle). This results in 90 features. The features are normalized using the Min-Max technique.

Inference ML Algorithm: k-Nearest Neighbour (kNN) [4] is an instance-based classifier based on the majority voting of its k neighbors for classifying an instance. The value of k defines how many nearest neighbor instances contribute to the classification of each instance. kNN does not use any model to fit, and it is only based on memory. kNN is a lazy learning algorithm because it does not have a learning phase; instead, it "memorizes" the training dataset. In this study, the kNN implementation is based on the MOA Java library.

Configurations Search: The goal of the first experiment was to explore the configuration space (see Table 1). We use the multi-objective search in Optuna [1] to select the best 702 configurations that optimize two different objectives: maximize the accuracy and minimize response time.

Table 1: Selected Hyperparameters for the 1st exploration.

Hyperparameters	Search Space	# Values
k	[1; 10]	10
distance function	Euclidean, Manhattan, Chebyshev	3
window size	[50; 900] with steps of 50	18
overlap	[0.0; 0.9] with steps of 0.1	10

Downsampling the Dataset: We reduce the sampling frequency of the dataset by downsampling via removing samples until we achieve the intended sampling frequency.

Sampling Frequency: We conducted two experiments to evaluate the effect of the sampling frequency. In the first experiment, we varied the sampling frequency of the train set and tested it against all considered frequencies. For the second experiment, we set the training set frequency at 100 Hz and tested with frequencies of 25 Hz and 1 Hz. When adjusting the sampling frequency of the dataset, we ensure that each sliding window contains the same number of seconds of data regardless of the frequency. However, specific constraints related to window size and overlap arise when dealing with a sampling frequency of 1 Hz. Consequently, only 189 configurations are usable for this particular frequency due to these constraints.

4 RESULTS

4.1 Hyperparameters

The results show that the same configurations have different behaviors depending on the user under test, resulting in different Pareto-Fronts for each user. Table 2 summarizes the variation in the accuracy, response time, and energy consumption of the Pareto-Front of each user.

Table 2: Pareto-Front values (PP) for accuracy (Acc), response time (RT), and energy consumption (EC) for all users.

User	Acc (%)	EC (mJ)	RT (ms)	PP
1	[33.05 ; 81.97]	[10.15 ; 121.12]	[5.70 ; 62.92]	57
2	[26.89 ; 95.15]	[9.27 ; 116.58]	[5.11 ; 59.64]	64
3	[20.46 ; 93.95]	[9.16 ; 123.62]	[5.07 ; 61.84]	93
4	[27.85 ; 94.71]	[9.08 ; 99.85]	[4.99 ; 51.40]	77
5	[21.48 ; 91.06]	[9.63 ; 73.22]	[5.29 ; 38.04]	68
6	[27.52 ; 92.80]	[8.92 ; 105.72]	[4.93 ; 53.11]	71
7	[34.26 ; 95.35]	[9.30 ; 130.30]	[5.10 ; 66.97]	51
8	[11.08 ; 86.51]	[8.56 ; 86.55]	[4.64 ; 44.40]	65

Table 2 shows that the highest accuracy measured in this experiment is 95.35% and for User 7 using the configuration 305 (window size = 900, overlap = 0%, distance = Manhattan, k = 9). However, this config. also results in the highest energy consumption (130.30 mJ) and response time (66.97 ms), considering all the Pareto-Front of all users under analysis. On the other hand, for User 1 and config. 52 (window size = 850, overlap = 0%, distance = Manhattan, k = 10), the accuracy does not exceed 82%. Despite User 1 having registered the lowest maximum accuracy, it is the third user with the highest energy consumption (121.12 mJ) and the second with the highest response time (62.92 ms). There are users with higher accuracies than User 1 and lower energy consumption and response times. For example, User 5, with config. 909 (window size = 900, overlap = 50%, distance = Manhattan, k = 10), achieved 91.06% accuracy (more 9.09% than User 1) with the lowest energy consumption (73.22 mJ, less 47.90 mJ than User 1) and response time (38.04 ms, less 24.88 ms than User 1).

Figure 2 shows the importance of the hyperparameters when considering the Pareto-Front configurations for User 5. Figure 2 shows that window size and overlap have the most impact on the system, with the distance function being the hyperparameter with the most negligible impact. The most important factor for accuracy is the overlap. The value of k and window size also has some importance. The importance of the distance function for accuracy is much lower. The window size and overlap are the most important hyperparameters for energy consumption and response time. The results show that the studied hyperparameters have different levels of impact on the different performance metrics and this should be considered when optimizing the model for a specific task.

We already concluded that the hyperparameters could impact the model's overall performance and computational cost of the HAR system. However, the hyperparameters can significantly impact the model's performance at an activity level. For example, a configuration allows a global F1-Score of 0.80, but at an activity level, the model's performance for activities sitting, standing, and descending stairs is significantly worse than the remaining activities. Some activities are not affected by the hyperparameter's values (e.g. running, lying, and cycling). These results reinforced the importance of adaptability in HAR systems because although

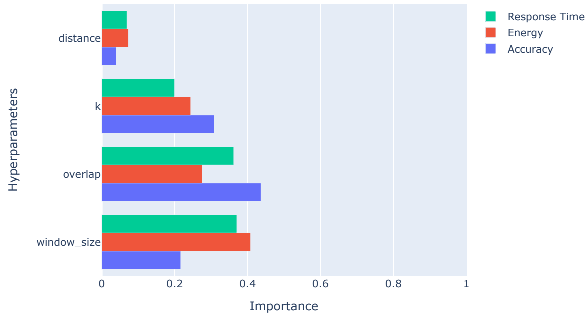

Figure 2: Hyperparameters Importance for the Pareto-front configurations.

configuration provides a global high performance for the system, some activities may be negatively affected by the configuration.

4.2 Sampling Frequency

The results show that changing the sampling frequencies of the training set and of the test set significantly impacted the model's accuracy. Overall, all users show a decreased accuracy when the sampling frequency is reduced. However, the decrease in the sampling frequency impacted all users differently. There are users for whom this impact is more significant than others. The biggest drop in the accuracy is shown for User 8, with a decrease in accuracy of about 17%. There are also other significant drops in the accuracy for Users 3 and 4, where we obtained a decrease of 12.40% and 12.43%, respectively. On the other hand, User 7 reports a slight reduction in accuracy, dropping 4.27%, and User 5 shows a decrease of 5.63%.

The combination of the sampling frequency for the train and test sets that resulted in the lowest accuracy obtained is when both sets have a sampling frequency of 1 Hz. This combination obtains lower accuracy for 50% of the users. On average, the highest accuracy values are obtained for sampling frequencies greater or equal to 12.5 Hz for the train and test sets. Overall, the accuracy benefits from sampling frequencies greater or equal to 12.5 Hz, while sampling frequencies equal to or below 5 Hz show significant decreases in accuracy.

We also evaluate the effect of the sampling frequency on the Pareto-Front of User 5. We use a sampling frequency of 100 Hz for the train set and then test them using sampling frequencies of 100 Hz, 25 Hz, and 1 Hz. Figure 3 presents the results of this experiment. It shows that the Pareto-Fronts are different for each of the frequencies under study. Focusing only on the sampling frequency of 100 Hz, we observe a significant increase in the energy consumption and response time for accuracies greater or equal to 85%, with a slight increase in the accuracy. However, this behavior is absent for 25 Hz and 1 Hz sampling frequencies. When the sampling frequency is reduced, the energy consumption and response time variation become smaller.

Comparing the Pareto-Front for the sampling frequencies of 100 Hz and 25 and focusing only on the configuration that allows us to obtain the maximum accuracy, we observe a significant reduction in energy consumption and response time. We measured a decrease of 58% in energy consumption and 59% in response time. At the same time, the accuracy does not suffer significant modifications. Considering now the Pareto-Front for the sampling frequencies of 100 Hz and 1 Hz, and focusing only on the configuration that allows us to obtain the maximum accuracy, higher energy consumption and response time reductions are observed. Although we observe a decrease of 80% in energy consumption and 79% in response time, the sampling frequency of 1 Hz also results in a reduction in the accuracy of about 11%.

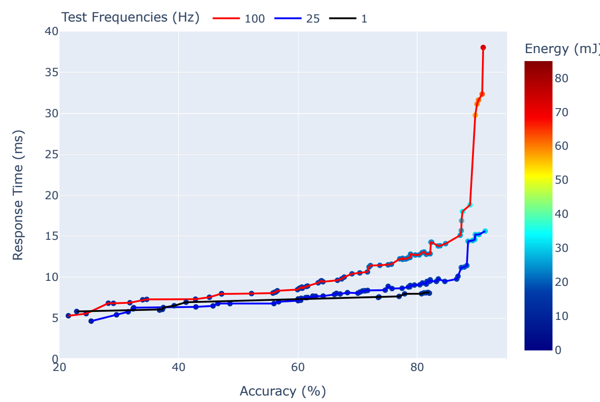

Figure 3: Pareto-frontier of User 5 for different test frequencies (100 Hz, 25 Hz, and 1 Hz).

5 CONCLUSION

Human activity recognition (HAR) is a research area that has been increasingly addressed due to the widespread of mobile devices, such as smartphones, with enough processing power and memory. HAR systems must be able to adapt to the device constraints, the user, and the environment to be more efficient.

As one way to change the HAR system's behavior is by changing its hyperparameters, this paper presented a study of the impact of window size, overlap, value of k, distance function, and sampling frequency on a *k*NN-based HAR system, in terms of accuracy, response time, and energy consumption using the PAMAP2 dataset.

The obtained results show that hyperparameters significantly impact the system's accuracy, response time, and energy consumption. Each user and activity is characterized by different configurations belonging to the Pareto-Front and different range values of the performance metrics. Overall, window size and overlap were the hyperparameters with a more significant impact on the system.

The sampling frequency also significantly impacted the HAR system and had different behaviors depending on the user and activity. Although lowering the sampling frequency leads to energy consumption savings and faster response time, its reduction must be carefully analyzed due to its impact on the system's accuracy.

The significant HAR system impact of certain hyperparameters, reinforces the need to carefully study them before implementing a HAR system. These conclusions also motivate the necessity of a self-adaptive HAR system for the user, the device, and the target service.

REFERENCES

[1] Takuya Akiba, Shotaro Sano, Toshihiko Yanase, Takeru Ohta, and Masanori Koyama. 2019. Optuna: A Next-Generation Hyperparameter Optimization Framework. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining* (Anchorage, AK, USA) *(KDD '19)*. Association for Computing Machinery, New York, NY, USA, 2623–2631. https://doi.org/10.1145/3292500.3330701

[2] Yunus Celik, Samuel Stuart, Wai Lok Woo, Liam T. Pearson, and Alan Godfrey. 2022. Exploring human activity recognition using feature level fusion of inertial and electromyography data. In *2022 44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*. IEEE, Glasgow, Scotland, 1766–1769. https://doi.org/10.1109/EMBC48229.2022.9870909

[3] Marc Claesen and Bart De Moor. 2015. Hyperparameter Search in Machine Learning. *CoRR* abs/1502.02127 (2015), 5 pages. arXiv:1502.02127 http://arxiv.org/abs/1502.02127

[4] T. Cover and P. Hart. 1967. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory* 13, 1 (1967), 21–27.

[5] Paulo J. S. Ferreira, João M. P. Cardoso, and João Mendes-Moreira. 2020. kNN Prototyping Schemes for Embedded Human Activity Recognition with Online Learning. *Computers* 9, 4 (2020), 20 pages.

[6] Paulo J. S. Ferreira, João Mendes Moreira, and João M. P. Cardoso. 2023. A Study on Hyperparameters Configurations for an Efficient Human Activity Recognition System. arXiv:2308.13314 [cs.HC]

[7] Kemilly D. Garcia, Tiago Carvalho, João Mendes-Moreira, João M. P. Cardoso, and André C. P. L. F. de Carvalho. 2020. A Study on Hyperparameter Configuration for Human Activity Recognition. In *14th International Conference on Soft Computing Models in Industrial and Environmental Applications (SOCO 2019)*, Francisco Martínez Álvarez, Alicia Troncoso Lora, José António Sáez Muñoz, Héctor Quintián, and Emilio Corchado (Eds.). Springer International Publishing, Cham, 47–56.

[8] Ashish Gupta and Hari Prabhat Gupta. 2021. YogaHelp: Leveraging Motion Sensors for Learning Correct Execution of Yoga With Feedback. *IEEE Transactions on Artificial Intelligence* 2, 4 (2021), 362–371. https://doi.org/10.1109/tai.2021.3096175

[9] Vincent Hernandez, Tomoya Suzuki, and Gentiane Venture. 2020. Convolutional and recurrent neural network for human activity recognition: Application on American sign language. *PLOS ONE* 15 (02 2020), e0228869. https://doi.org/10.1371/journal.pone.0228869

[10] Frank Hutter, Holger Hoos, and Kevin Leyton-Brown. 2014. An Efficient Approach for Assessing Hyperparameter Importance. In *Proceedings of the 31st International Conference on Machine Learning (Proceedings of Machine Learning Research, Vol. 32)*, Eric P. Xing and Tony Jebara (Eds.). PMLR, Bejing, China, 754–762.

[11] Ankita Jain and Vivek Kanhangad. 2018. Human Activity Classification in Smartphones Using Accelerometer and Gyroscope Sensors. *IEEE Sensors Journal* 18, 3 (2018), 1169–1177. https://doi.org/10.1109/JSEN.2017.2782492

[12] Sahak Kaghyan and Hakob Sarukhanyan. 2012. Activity recognition using K-nearest neighbor algorithm on smartphone with Tri-axial accelerometer. *International Journal of Informatics Models and Analysis (IJIMA), ITHEA International Scientific Society, Bulgaria* 1 (2012), 146–156.

[13] Mustafa Kose, Ozlem Durmaz Incel, and Cem Ersoy. 2012. Online human activity recognition on smart phones. *Workshop on Mobile Sensing: From Smartphones and Wearables to Big Data* 16, 2012 (2012), 11–15.

[14] Gang Luo. 2016. A review of automatic selection methods for machine learning algorithms and hyper-parameter values. *Network Modeling Analysis in Health Informatics and Bioinformatics* 5, 1 (2016), 1–16.

[15] Saad Mohamad, Moamar Sayed-Mouchaweh, and Abdelhamid Bouchachia. 2020. Online active learning for human activity recognition from sensory data streams. *Neurocomputing* 390 (2020), 341–358. https://doi.org/10.1016/j.neucom.2019.08.092

[16] Gael S. Mubibya and Jalal Almhana. 2022. Improving Human Activity Recognition using ML and Wearable Sensors. In *ICC 2022 - IEEE International Conference on Communications*. IEEE, Seoul, Republic of Korea, 165–170. https://doi.org/10.1109/ICC45855.2022.9839267

[17] P. Paul and T. George. 2015. An effective approach for human activity recognition on smartphone. In *2015 IEEE International Conference on Engineering and Technology (ICETECH)*. IEEE, Coimbatore, India, 1–3.

[18] Attila Reiss and Didier Stricker. 2012. Creating and Benchmarking a New Dataset for Physical Activity Monitoring. In *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments* (Heraklion, Crete, Greece) *(PETRA '12)*. Association for Computing Machinery, New York, NY, USA, Article 40, 8 pages. https://doi.org/10.1145/2413097.2413148

[19] A. Reiss and D. Stricker. 2012. Introducing a New Benchmarked Dataset for Activity Monitoring. In *2012 16th International Symposium on Wearable Computers*. IEEE, Newcastle, UK, 108–109.

[20] Sadiq Sani, Nirmalie Wiratunga, and Stewart Massie. 2017. Learning deep features for kNN-based human activity recognition.. In *ICCBR*, Antonio A. Sanchez-Ruiz and Anders Kofod-Petersen (Eds.). CEUR Workshop Proceedings, Trondheim, Norway, 95–103.

[21] Muhammad Shoaib, Stephan Bosch, Ozlem Incel, Hans Scholten, and Paul Havinga. 2015. A Survey of Online Activity Recognition Using Mobile Phones. *Sensors* 15 (01 2015), 2059–2085. https://doi.org/10.3390/s150102059

[22] Pekka Siirtola and Juha Röning. 2012. User-Independent Human Activity Recognition Using a Mobile Phone: Offline Recognition vs. Real-Time on Device Recognition. In *Distributed Computing and Artificial Intelligence*, Sigeru Omatu, Juan F. De Paz Santana, Sara Rodríguez González, Jose M. Molina, Ana M. Bernardos, and Juan M. Corchado Rodríguez (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 617–627.

[23] B. Wang and N. Z. Gong. 2018. Stealing Hyperparameters in Machine Learning. In *2018 IEEE Symposium on Security and Privacy (SP)*. IEEE, San Francisco, CA, USA, 36–52. https://doi.org/10.1109/SP.2018.00038

[24] Yiwei Xia, Jun Ma, ChuYue Yu, XunHuan Ren, Boriskevich Anatoliy Antonovich, and Viktar Yurevich Tsviatkou. 2022. Recognition System Of Human Activities Based On Time-Frequency Features Of Accelerometer Data. In *2022 International Conference on Intelligent Systems and Computer Vision (ISCV)*. IEEE, Fez, Morocco, 1–5. https://doi.org/10.1109/ISCV54655.2022.9806107

[25] Shugang Zhang, Zhiqiang Wei, Jie Nie, Lei Huang, Shuang Wang, and Zhen Li. 2017. A Review on Human Activity Recognition Using Vision-Based Method. *Journal of Healthcare Engineering* 2017 (07 2017), 1–31. https://doi.org/10.1155/2017/3090343

[26] Lingxiang Zheng, Dihong Wu, Xiaoyang Ruan, Shaolin Weng, Ao Peng, Biyu Tang, Hai Lu, Haibin Shi, and Huiru Zheng. 2017. A Novel Energy-Efficient Approach for Human Activity Recognition. *Sensors* 17, 9 (2017), 21 pages. https://doi.org/10.3390/s17092064

A TABLES

Table 3: Detailed information regarding the Related Work.

Study	Hyperparameters	Dataset	Number of Activities	Metrics	Inference Algorithm
Garcia et al. [7]	Window Size (samples) (100 to 1000) Overlap (0.0 to 0.9)	PAMAP2 [18, 19]	12	Accuracy, Energy, Execution Time	*k*NN, VFDT, Naive Bayes, Ensemble
Zheng et al. [26]	Sampling Frequency (Hz) (1, 5, 10, 50)	Private	6	Accuracy Energy	SVM
This Work	Sampling Frequency (Hz) (1, 5, 12.5, 25, 50, 100) Window Size (samples) (50 to 900) Overlap (0 to 0.9) k (1 to 10) Distance (E, M, C)	PAMAP2 [18, 19]	12	Accuracy, F1-Score, Energy, Inference Time	*k*NN