An Experimental Study on the Energy Efficiency of Feature Selection for Human Activity Recognition with Wrist-worn Devices

Susanna Peretti[1][0009−0000−4162−4301], Chiara Contoli[1][0000−0003−2389−2593], and Emanuele Lattanzi[1][0000−0002−6568−8470]

dept. of Pure and Applied Sciences, University of Urbino, Urbino, Italy
s.peretti@campus.uniurb.it
{chiara.contoli,emanuele.lattanzi}@uniurb.it

Abstract. Incorporating machine and deep learning methodologies into wearable devices has enhanced the capacity to accurately recognize human activity, thus enabling a range of applications including healthcare monitoring and fitness tracking. However, machine and deep learning can be costly in terms of the computational resources and energy consumption required. In this work, we study how a feature selection decision impacts the energy consumption of an ESP32 wearable device by evaluating the best trade-off between classification performance and energy expenditure. Experimental results, conducted on publicly available datasets, demonstrate that the best trade-off between energy consumption and accuracy is reached by selecting between 20 and 25 features, with an accuracy ranging between 73.56% and 87.44%, and an energy consumption between 2340.945 μJ and 3759.270 μJ.

Keywords: Human Activity Recognition · Feature Selection · Energy Efficiency · Wearables · Constrained Devices.

1 Introduction

The field of sensor-based Human Activity Recognition (HAR) has emerged as a pivotal area of research within the domain of wearable computing and ubiquitous sensing. Its applications span a wide range of domains, including health monitoring and the development of smart environments [24, 13]. Wrist-worn devices, such as smartwatches and fitness trackers, have become particularly popular due to their unobtrusive nature and their ability to collect data continuously. These devices frequently employ a multitude of sensors, including accelerometers, gyroscopes, and magnetometers, to capture a comprehensive array of signals that can be processed to infer user activities. Sensor-based activity recognition has yielded excellent results, mainly due to the application of machine learning (ML) techniques, both in shallow and deep approaches [33].

Nowadays, deep neural networks (DNNs) have become state-of-the-art in many machine learning applications, ranging from computer vision to speech recognition, by processing raw sensor data directly. However, the impressive performance of DNNs comes at the cost of significant computational resources required for both training and inference. A common approach to address these computational demands is to delegate the model inference to a cloud-based framework. On the other hand, maintaining inference tasks to wearable devices offers several compelling advantages: (i) it eliminates latency issues associated with cloud communication, thereby enhancing responsiveness; (ii) it enhances privacy and security by keeping data local to the device; and (iii) it can improve energy efficiency by balancing the energy demands of computation and communication [1, 4]. However, the deployment of such devices is often constrained by their limited battery life, necessitating the development of energy-efficient solutions to prolong operational duration without compromising accuracy. In this context, several studies have demonstrated that concerning tiny devices, shallow tools are more energetically advantageous than deep approaches while maintaining comparable classification capabilities [17, 3, 16, 23].

In shallow ML, representative features must be extracted from raw sensor data in a process known as feature selection to identify activities. The extraction can be performed in the time domain, frequency domain, or both to leverage the unique characteristics of each domain. The shallow approach envisages domain experts with specialized knowledge analyzing and selecting hand-crafted features, typically using heuristic algorithms [9]. Hand-crafted features often pertain to statistical information, regardless of the domain. Feature selection consists of discarding features that do not provide helpful information, i.e., irrelevant. Other discarded features are those that do not provide more information compared to currently selected ones, i.e., redundant. According to the literature [21, 26], feature selection strategies can be classified into three categories, which are the most widely used: filter-based, wrapper-based, and embedded-based approaches. More recently, two other categories have been identified, i.e., hybrid and ensemble [36].

Existing literature in the area of feature selection mainly focuses on proposing new algorithms to improve the selection of the best feature subset [30, 14, 37, 2]. Despite the high interest in the topic, feature computational complexity and resulting energy impact are often neglected. In [8], the authors propose a many-objective feature selection algorithm based on the computational complexity of features. However, the complexity of each feature is not determined, and they chose to assign a fixed, random cost value. Moreover, they did not consider the energy impact. Therefore, our work is motivated by the limited literature in exploring the computational complexity of each single feature and the evaluation of the energy consumption on constrained devices in the HAR context, leaving room for investigation. Momeni *et al.* consider features' computational complexity and try to estimate the energy consumption of the features on an ARM Cortex M3 hardware platforms [27]. However, the exploration was carried out for multimodal acute stress monitoring, which entails adopting, placing, and analyzing sensor data different from those used for HAR.

This paper extends our previous work [31] by thoroughly investigating the energy consumed by each feature during the feature selection process, measuring the actual consumption with a device rather than hypothesizing it, considering data collected via wearable devices such as smartwatches and smartphones. In this work, we consider activities of daily living, ranging from hand-based activities such as brushing teeth and preparing sandwiches to locomotion activities such as walking and running. The goal is to find the best trade-off between the energy consumption associated with feature selection and recognition accuracy. We used the Recursive Feature Elimination and Select From Model methods with RidgeCV regularization as feature selection algorithms.

To validate our approach, we consider three datasets to extract features in the time and frequency domains. One is a homemade dataset, *Ad-Hoc DB* [25], and the other two are the public *Watch_HAR* [35] and the *RealWorld2016* [34]datasets. To consider only signals acquired from wrist-worn devices, we filtered traces collected in the *RealWorld2016* dataset with the sensor positioned at the wrist. To assess the recognition accuracy, we used a Random Forest classifier. On the one hand, this choice is motivated by the successful results yielded in HAR and, on the other, by the fact that these algorithms and classifiers are lighter than other deep-learning approaches. Such an approach greatly simplifies the deployment on a lower-power-constrained device because it allows us to avoid the adoption of optimization deployment strategies [10]. Results show that our approach effectively balances energy consumption and recognition accuracy, demonstrating the potential to save energy while preserving accuracy when machine learning models are deployed on wearable devices.

The rest of the paper is organized as follows: Section 2 reports related work on the impact associated with the feature selection process; in Section 3 we describe our proposed approach and discuss achieved results in Section 4. We then summarize the contribution of this work in Section 5.

2 Related Work

Feature selection is a critical phase of the HAR process, which deserves attention from the research community. This attention has increased in the last decade because of the high amount of data fostered by the pervasiveness of mobile and wearable devices that allow for constant monitoring of human activity and biosignals [29]. Indeed, this phase highly impacts the quality of the classification accuracy since it is supposed to select the most relevant features. It also impacts the computational burden since each feature is supposed to have its computational complexity, thus leading to a lighter or heavier impact on devices' resource consumption.

Karagiannaki *et al.* evaluated three feature selection methods: Feature Selection based on Feature Similarity, Relief-F, and Clustering with Node Centrality. These methods were provided by a library, which was subsequently evaluated in terms of execution time and energy consumption. The algorithms were evaluated in the HAR context, and they found that the extraction and selection of features are the most time-consuming operations and that Relief-F is the fastest in terms of maximum execution time, around 28s. In contrast, the energy requirement of their framework is around 20.3 J. However, those results were obtained by implementing their architecture on a Samsung Galaxy Tab4, which differs from low-power wearable devices.

In [15], Ghasemzadeh *et al.* investigated the power optimization of wearable sensor nodes by performing instruction-level energy analysis of the feature selection phase. They used the TI MSP430 microcontroller to process real data collected from three subjects using wearable motion sensors. Specifically, they considered TI MSP430's 'mov' instruction to quantify the energy cost associated with each feature. To perform such a quantification, the authors leveraged the results of the work by Lane and Campbell [22], who assessed the energy consumption due to the execution of different types of the MSP430 processor instructions. However, Ghasemzadeh *et al.* considered only statistical features. Ding *et al.* focused on the reduction of energy consumption of wearable device-based HAR systems by exploring i) the use of hybrid, i.e., non-uniform, window techniques, ii) the use of a mutual information-based feature selection method, and iii) their proposed Random Forest methods [12]. They considered only time-domain features and, despite the focus on wearable devices, the experimental evaluation was performed on a XIAOMI 5 smartphone, which acts as a data collector from integrated sensors, and a Fujitsu SH771 laptop, which runs their proposed feature extraction and recognition algorithm. Moreover, the energy efficiency was evaluated in terms of computational time reduction, i.e., the time taken by the laptop to perform the recognition activity.

Another work exploring non-uniform window segmentation for energy-efficient HAR is the one of Bhat *et al.* [7]. The authors proposed a HAR framework that leverages textile-based stretch sensors and an accelerometer to capture raw data from 9 users. They considered features only from the frequency domain, i.e., Fast Fourier Transform and Discrete Wavelet Transform. They then used different classifiers, such as Support Vector Machine, Random Forest, Decision Tree, k-Nearest Neighbors, and their proposed neural network. The framework was implemented on the TI-CC2650 microcontroller, and the authors measured both the power and the energy consumed during the sensing, feature selection and classification, and communication via Bluetooth phase. However, the authors considered all the sets of selected features as if they were all equally relevant in their contribution to the activity recognition. Subsequently, Bhat *et al.* used a commercial analog front end to propose a fully integrated ultra-low power hardware accelerator for HAR that provides all steps from reading raw sensor data to activity classification. However, the feature selection was not the focus of the work.

Most of the literature does not consider the computational complexity of individual features, thus neglecting their varying complexity and, consequently, their different impact on energy consumption. Some works, such as the one by Barandas *et al.*, focused on providing tools that help users get an idea of the computational cost of feature extraction [6]. The authors developed a Python package named Time Series Feature Extraction Library (TSFEL), which provides a set of implemented feature extraction methods based on Numpy and SciPy. The proposed library allows users to analyze time series data and extract features in the temporal, statistical, and spectral domains. As output, it provides an estimation of the complexity of the feature. Moreover, the library is designed to allow users to write their feature extraction method and its corresponding domain so that other features can be added to those available by default in the library.

Compared to the existing literature, we consider simultaneously time and frequency domain features, and we evaluate the energy impact of each selected feature on a low-power device after considering the computational complexity of each feature.

3 Methodology

In this section, we provide background on the feature selection process, and we describe our proposed methodology to extract, select, and estimate the energy consumption of the features.

3.1 Background

The HAR process comprises four phases, typically: i) data gathering, ii) data pre-processing, iii) feature engineering, and iv) classification. Feature engineering consists of feature extraction and feature selection steps. Feature extraction is about analyzing the raw signal in the time, frequency, and time-frequency domains and subsequently extracting distinctive features. This step can be carried out manually by a so called domain expert, also with the help of optimization algorithms, or can be carried out automatically with the help of deep learning algorithms.

Feature selection is about discarding irrelevant and redundant features to select a subset of meaningful features. Filter-based, wrapper-based, and embedded-based approaches are the most widely used in this step. Filter methods leverage training data characteristics to identify feature importance and do not depend on the learning algorithm. Wrapper methods iteratively search for the relationship between optimal feature subset selection and feature relevance until a stopping criterion is met, and the search relies on the predictive performance of the specific learning algorithm. Dhal and Azad surveyed feature selection techniques and the corresponding area of ML applications [11], concluding that: i) the filter approach is much faster than the other two, but it is less accurate; ii) the wrapper approach is the most accurate, but it suffers high computational time, whereas the embedded approach overcomes the problems of the other two approaches; iii) results of feature selection methods vary depending on the used dataset and adopted ML approach.

3.2 Proposed Approach

Extracting relevant features from both time and frequency domains provides more benefits. Indeed, time domain features are commonly used because of their low computation cost. However, identifying complex dynamics and hidden patterns, such as repetitive movements or other periodic components, may require analyzing signals in the frequency domain to unveil those peculiarities that are not so easily discernible in time domain analysis. Therefore, we extract features from both domains. In particular, we extract statistical data, such as mean and standard deviation, and other relevant metrics, such as interquartile range, autoregressive coefficients, signal magnitude area, and root mean square from the time domain. We used the Fast Fourier Transform (FFT), a mathematical technique that decomposes the data into different frequency components, to extract features such as skewness, kurtosis, peaks, and energy in the frequency domain.

3.3 Feature Energy Characterization

In order to analyze the trade-off between the energy consumption of the considered features and the corresponding classification accuracy, we developed a sensor-based HAR application. The application comprises two distinct tasks. The first task is responsible for data collection from a triaxial accelerometer and gyroscope, while the second task is responsible for computing each feature on the gathered data. The software was compiled for the ESP32 platform and executed on the wearable device while the power-drawn trace was recorded to estimate the energy consumption. An example of a power trace is presented in Figure 1.

In particular, the figure reports the power consumption when computing the FFT followed by the Kurtosis, standard deviation, and Skewness features. The energy bursts corresponding to each feature have been highlighted by dotted rectangles. It is noteworthy that the FFT exhibits a higher power level with respect to the computation of the feature. This is likely due to the fact that it has been implemented using the Espressif ESP32 library which involves the usage of the internal digital signal processor (DSP). Moreover, the periodic bursts corresponding to the reading of the sensor samples are also clearly visible. Notice that to make the bursts in the figure more visible, both the FFT and feature calculations were consecutively repeated ten times. The features considered in this work, together with the corresponding energy consumption, are listed in Table 1.

Notice that FFT appears to be the most expensive in terms of energy, reaching almost 1400 μ J, despite using a dedicated co-processor (DSP). Moreover, features involving simple operations such as multiplications, sums, and comparisons record a consumption lower than one μ J. An intermediate range is shown by features that involve more complex operations, such as sorts, exponentials, and square roots.

Starting from the characterization of the proposed features, we leverage both wrapper and embedded approaches as feature selection strategies. In particular, we use the Recursive Feature Elimination (LR-RFE) wrapper-based method that

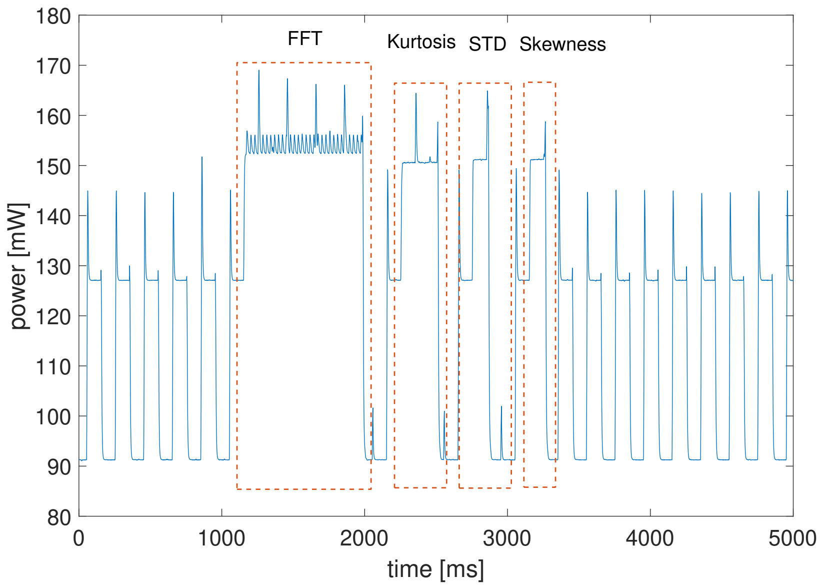


Fig. 1. Power consumption trace of the ESP32 collected during features computation.

Table 1. Characterization of the features energy consumption.

Feature	Formula	Energy [μJ]
Mean	$\frac{1}{n}\sum_{i=1}^{n} s_i$	0.21
Min	$min(s_1, s_2, ...s_n)$	0.19
Max	$max(s_1, s_2, ...s_n)$	0.19
Median	$median(s_1, s_2, ...s_n)$	152.66
Standard Deviation	$\sqrt{\frac{1}{n}\sum_{i=1}^{n}(s_i - \mu_s)^2}$	409.81
N Peaks	The number of signal peaks	0.15
Peak-to-Peak Amplitude	$max(s) - min(s)$	0.17
Interquartile Range	$perc(s, 75) - perc(s, 25)$	119.32
Autocorrelation	$\frac{1}{n}\sum_{i=1}^{n-k}(s_i - \bar{s}) * (s_{i+k} - \bar{s})$	0.17
Energy	$\frac{\sum_{i=1}^{n} s_i^2}{length(s_i)}$	0.15
Autoregressive coefficients	$\sum_{i=1}^{n} \alpha_i x(n-i) + \epsilon(n)$	0.15
Signal Magnitude Area	$\frac{1}{n}\sum_{i=1}^{n-1}(\lvert x_i\rvert + \lvert y_i\rvert + \lvert z_i\rvert)$	0.15
Root Mean Square	$\sqrt{\frac{1}{n}\sum_{i=1}^{n} s_i^2}$	385.47
FFT	$\sum_{n=0}^{N-1} x(n) \cdot e^{-j2\pi\frac{kn}{N}}$	1399.61
Spectral Mean	$\frac{\sum_{i=1}^{n} k\lvert S_i\rvert^2}{\sum_{i=1}^{n} \lvert S_i\rvert^2}$	0.15
Skewness	$\frac{\mathcal{E}[(s_i - \bar{s})^4]}{\mathcal{E}[(s_i - \bar{s})^2]^2}$	189.11
Kurtosis	$\mathcal{E}[(\frac{s_i - \bar{s}}{\sigma})^3]$	186.98
Growth factor	$\frac{max(\lvert s\rvert)}{\sqrt{mean(s^2)}}$	10.32

gradually reduces the number of features combined with logistic regression. This method considers all features and, subsequently, removes iteratively the least relevant ones. We also use the Select From Model with RidgeCV regularization (SFM-RidgeCV) embedded-based method, which involves a set of RidgeCV models. Each RidgeCV model has a different set of selected features and chooses the best feature subset based on cross-validated performance metrics. The method then leverages the optimal feature subset to train the final model.

4 Experimental Evaluation

In this section, we provide a thorough description of the experimental setup, and we present the results of several experiments.

4.1 Setup

Feature selection algorithms, model training, and testing have been executed on a personal computer equipped with an Intel Core i7-4712MQ $\times$ 8 processor and 8 GB of RAM using Python. The energy characterization of each feature was done by executing it on an ESP32 connected to an MPU6050 triaxial accelerometer and gyroscope [20].

To estimate the device's energy consumption, we measured the voltage drop across a (9.8Ω) sensing resistor placed in series with the device's power supply. The device was powered at 3.3V using an NGMO2 Rohde & Schwarz dual-channel power supply [32]. During the experiments, we sampled the monitored signals using a National Instruments NI-DAQmx PCI-6251 16-channel data acquisition board [28].

The solidness of our methodology has been evaluated using three representative datasets collected via wrist-worn devices have been evaluated, namely Watch_HAR, Ad-hoc DB, and RealWorld2016.

Watch_HAR [5, 35]: in this dataset, 13 subjects (both male and female) wearing a smartwatch on their dominant hand performed activities (commonly executed at home or at work) in a laboratory environment. Data are collected from the accelerometers, gyroscopes, and magnetometers. Each subject, in its own style, carried out 16 activities for approximately 1 to 3 minutes: *brushing teeth, preparing sandwiches, reading a book, typing, using a phone, using remote control, walking freely, walking holding a tray, walking with a handbag, walking with hands in pockets, walking with objects underarm, washing face and hands, washing mug, washing plate, writing*. Note that we removed the walking activities from this dataset because we wanted to focus solely on the recognition of hand-based activities.

Ad-hoc DB [25]: we created this dataset specifically to study the recognition of *handwashing* and *handrubbing* activities performed during the day. Data are collected from the triaxial accelerometer and gyroscope of a smartwatch positioned on the wrist of the dominant hand of four participants during real-life activities. Each subject wore the smartwatch for several hours on different days and was asked to annotate the start and the end of each handwashing or hand-rubbing activity performed during the day. Together with the activities of interest, we also collected *Unknown Activities* (*UAs*) data by randomly sampling the sensors during the day. For each subject, we collected about 2 hours spent washing hands, about 2 hours and 30 minutes spent rubbing, and about 3 hours of *UAs*.

RealWorld2016 [34]: the dataset includes movements from 15 different subjects, comprising 8 males and 7 females, engaged in various activities such as standing, lying down, sitting, jumping, climbing up, climbing down, walking, and running. Data for each activity were gathered using six sensors that measured acceleration, GPS, gyroscope, light, magnetic field, and sound levels. These sensors were positioned at various body locations, including the chest, forearm, head, shin, thigh, upper arm, and waist. Additionally, each movement was recorded with a video camera for further analysis. The subjects had an average age of 32 years, an average weight of 74 kg, and an average height of 171 cm, with each activity lasting approximately 10 minutes on average.

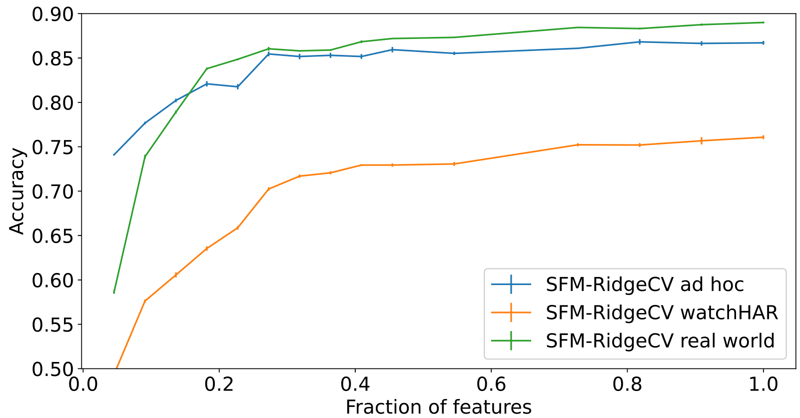
The signals of the three datasets have been divided into a 2.56 seconds time window. Choosing the time window size and the percentage of overlap is non-trivial. The choice motivation is manifold: on the one hand, according to the literature, in HAR tasks, different window lengths have been used, from 1 second up to 30 seconds [18, 19]. On the other, in our preliminary exploration, we found that the best value for the datasets resulted in 2 seconds without overlap. Last but not least, using the FFT requires a power of 2 sample length to allow a more efficient computation. Therefore, we opted for 256 samples, which resulted in 2.56 seconds, giving a sampling frequency of 100 Hz.

The dataset contains accelerometer and gyroscope signals, each measuring movement along three axes (x, y, z). The Signal Magnitude Area (SMA) is computed across all three axes together as a single value for each sensor type. For the time domain, there are 12 features per 3 axes and 2 types of sensors (accelerometer and gyroscope), thus resulting in $12 \times 3 \times 2$ features. For the frequency domain, there are 8 features per axis, resulting in $8 \times 3 \times 2$ features. As result, the total number of features is 122 features.

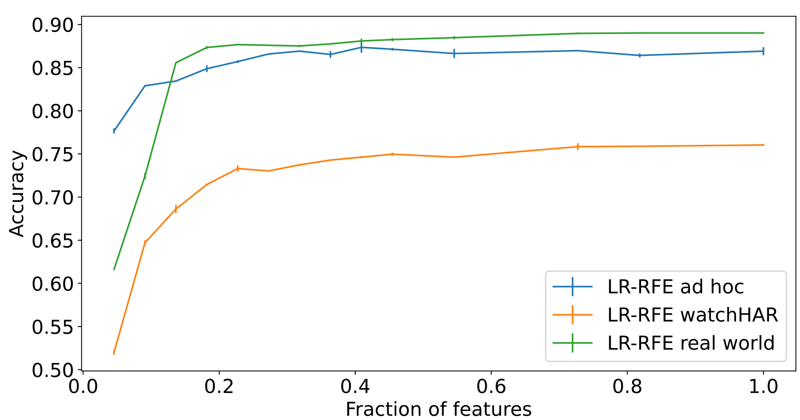
4.2 Accuracy results

By varying the number of features, the two selection methods, SFM-RidgeCV and LR-RFE, have been applied to the three datasets. The best set of selected features was then used to train and test the recognition accuracy of a Random Forest classifier.

Figure 2 shows the classification accuracy obtained with the SFM-RidgeCV and LR-RFE selection methods when varying the number of the selected features, expressed as a fraction of totals. Results are shown for Ad-hoc DB (blue line), Watch_HAR (orange line), and RealWorld2016 (green line) datasets. Each point of the plots is obtained by averaging 5 executions using different random seeds, and vertical bars represent the standard deviation.



(a) SFM-RidgeCV



(b) LF-RFE

Fig. 2. Classification accuracy with respect to the number of selected features expressed as a fraction of totals.

The Ad-hoc DB dataset reaches a stable accuracy value close to the maximum using about 50% of the available features with both selection methods. Although the other datasets require a higher percentage of available features to get closer to the maximum accuracy, it is interesting to note that a high level of accuracy (i.e., greater or equal to 80%) is reached with less than 20% of available features for the Ad-hoc DB and RealWorld2016 datasets with both selection methods. This means that, depending on the application requirements, it is possible to save energy without sacrificing accuracy too much.

4.3 Energy-aware feature selection

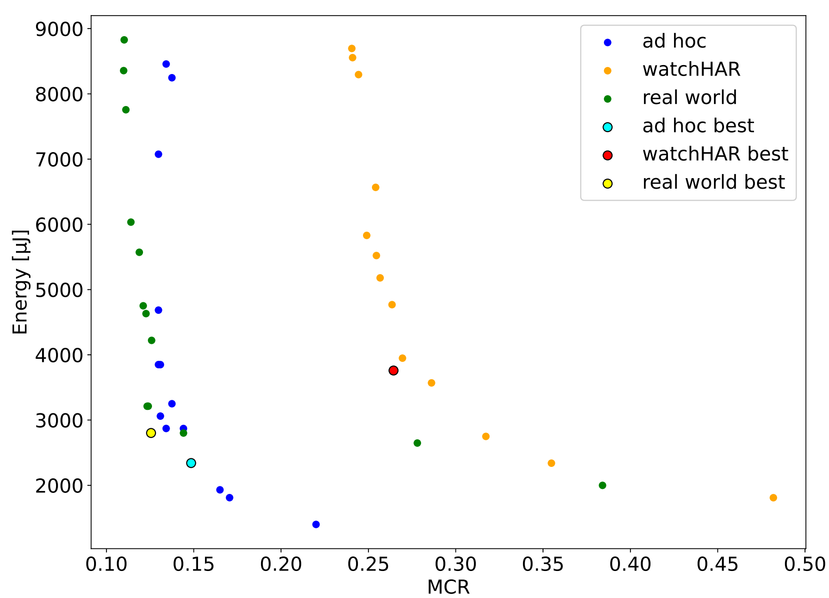


Fig. 3. Pareto plan reporting the trade-off between the miss-classification rate (MCR) and the total energy consumption of the corresponding selected features.

Figure 3 shows the Pareto plan, which plots each feature selection point obtained in the previous experiments. These points are characterized in terms of total energy consumption in micro joules and the corresponding misclassification rate (MCR) of the Random Forest Classifier. The results for the three datasets are shown in three different colors: blue for AD-hoc DB, orange for WatchHAR, and green for RealWorld2016.

For each dataset, the best trade-off point, which minimizes both energy consumption and MCR, is highlighted in specific colors: light blue for Ad-hoc DB, red for WatchHAR, and yellow for RealWorld2016. The optimal point for WatchHAR includes 25 features, while the other two datasets count 20 features.

(a) Ad-hoc DB

(b) WatchHAR

(c) RealWorld2016

Fig. 4. Visual representation of the selected features in the Pareto optimal points by means of word clouds.

To better understand which features were selected at each optimal point and how they contribute to global energy consumption, we constructed the three word clouds corresponding to the three optimal points highlighted in the Pareto plot. Each cloud is a visual representation of the occurrence of each feature tag in the way that the words appear bigger the more often they are used. For instance, considering a feature named `gyroY_fft_npeaks`, which is the number of peaks calculated over the spectral representation of the y component of the gyroscope, it contributes to the word cloud by increasing both the `gyroY`, `fft` and `npeaks` counters.

Figure 4 shows the three resulting word clouds corresponding to the experiments conducted on top of the three reference datasets. It is interesting to note that the proposed method provides an optimal feature list in all three datasets that mainly contains features computed from the FFT despite its high energy cost. This demonstrates the fact that to have a good compromise between accuracy and energy consumption, it is still necessary to spend the energy needed to shift to the frequency domain instead of calculating features directly on top of the time series data. An additional observation derived from an analysis of the figure indicates that, in all three datasets, the signals derived from the accelerometer appear to be more significant than those derived from the gyroscope. Consequently, the computed features primarily originate from it.

5 Conclusion

High dimensional sensor data poses challenges in terms of pattern recognition and computational complexity for sensor-based human activity recognition, which also impact the system energy expenditure. This paper proposes a method to extract, select, and characterize the energy associated with each feature involved in activity recognition. The method aims to select the most informative features while considering their energy impact. We tested our methodology on three datasets, and the results show that it is possible to fine-tune feature selection to find the best compromise between accuracy and energy consumption. The analysis also highlights that a good compromise is reached when features are selected from the frequency domain, although this implies selecting features with a higher energy cost. Our proposed methodology is general enough to be applied in different classification scenarios, and it can help tune the energy impact of machine learning models deployed on wearable devices.

References

1. Abdel-Basset, M., Hawash, H., Chang, V., Chakrabortty, R.K., Ryan, M.: Deep learning for heterogeneous human activity recognition in complex iot applications. IEEE Internet of Things Journal pp. 1–1 (2020). https://doi.org/10.1109/JIOT.2020.3038416
2. Ahadzadeh, B., Abdar, M., Safara, F., Khosravi, A., Menhaj, M.B., Suganthan, P.N.: Sfe: A simple, fast and efficient feature selection algorithm for high-dimensional data. IEEE Transactions on Evolutionary Computation (2023)

3. Alam, F., Mehmood, R., Katib, I., Albeshri, A.: Analysis of eight data mining algorithms for smarter internet of things (iot). Procedia Computer Science **98**, 437–442 (2016)
4. Alessandrini, M., Biagetti, G., Crippa, P., Falaschetti, L., Turchetti, C.: Recurrent neural network for human activity recognition in embedded systems using ppg and accelerometer data. Electronics **10**(14), 1715 (2021)
5. Alevizaki, A., Trigoni, N.: watchHAR: A Smartwatch IMU dataset for Activities of Daily Living (Sep 2022). https://doi.org/10.5281/zenodo.7092553, https://doi.org/10.5281/zenodo.7092553
6. Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., Liu, H., Schultz, T., Gamboa, H.: Tsfel: Time series feature extraction library. SoftwareX **11**, 100456 (2020)
7. Bhat, G., Deb, R., Chaurasia, V.V., Shill, H., Ogras, U.Y.: Online human activity recognition using low-power wearable devices. In: 2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD). pp. 1–8. IEEE (2018)
8. Bidgoli, A.A., Ebrahimpour-Komleh, H., Rahnamayan, S.: A many-objective feature selection algorithm for multi-label classification based on computational complexity of features. In: 2019 14th International Conference on Computer Science & Education (ICCSE). pp. 85–91. IEEE (2019)
9. Chen, Z., Zhang, L., Cao, Z., Guo, J.: Distilling the knowledge from handcrafted features for human activity recognition. IEEE Transactions on Industrial Informatics **14**(10), 4334–4342 (2018)
10. Contoli, C., Lattanzi, E.: A study on the application of tensorflow compression techniques to human activity recognition. IEEE Access (2023)
11. Dhal, P., Azad, C.: A comprehensive survey on feature selection in the various fields of machine learning. Applied Intelligence **52**(4), 4543–4581 (2022)
12. Ding, G., Tian, J., Wu, J., Zhao, Q., Xie, L.: Energy efficient human activity recognition using wearable sensors. In: 2018 IEEE Wireless Communications and Networking Conference Workshops (WCNCW). pp. 379–383. IEEE (2018)
13. Diraco, G., Rescio, G., Siciliano, P., Leone, A.: Review on human action recognition in smart living: Sensing technology, multimodality, real-time processing, interoperability, and resource-constrained processing. Sensors **23**(11), 5281 (2023)
14. El-Hasnony, I.M., Barakat, S.I., Elhoseny, M., Mostafa, R.R.: Improved feature selection model for big data analytics. IEEE Access **8**, 66989–67004 (2020)
15. Ghasemzadeh, H., Amini, N., Saeedi, R., Sarrafzadeh, M.: Power-aware computing in wearable sensor networks: An optimal feature selection. IEEE Transactions on Mobile Computing **14**(4), 800–812 (2014)
16. Gupta, C., Suggala, A.S., Goyal, A., Simhadri, H.V., Paranjape, B., Kumar, A., Goyal, S., Udupa, R., Varma, M., Jain, P.: Protonn: Compressed and accurate knn for resource-scarce devices. In: International conference on machine learning. pp. 1331–1340. PMLR (2017)
17. Haigh, K.Z., Mackay, A.M., Cook, M.R., Lin, L.G.: Machine learning for embedded systems: A case study. BBN Technologies: Cambridge, MA, USA (2015)
18. Hassan, M.M., Uddin, M.Z., Mohamed, A., Almogren, A.: A robust human activity recognition system using smartphone sensors and deep learning. Future Generation Computer Systems **81**, 307–313 (2018)
19. Hou, C.: A study on IMU-based human activity recognition using deep learning and traditional machine learning. 2020 5th International Conference on Computer and Communication Systems (ICCCS) pp. 225–234 (2020)

20. InvenSense Inc.: Mpu-6050 product specification (2023), https://invensense.tdk.com/products/motion-tracking/6-axis/mpu-6050/, last accessed 2024-06-14
21. Kumar, V., Minz, S.: Feature selection. SmartCR **4**(3), 211–229 (2014)
22. Lane, N., Campbell, A.: The influence of microprocessor instructions on the energy consumption of wireless sensor networks. In: Third Workshop on Embedded Networked Sensors (EmNets 2006). vol. 34 (2006)
23. Lattanzi, E., Calisti, L.: Energy-aware tiny machine learning for sensor-based handwashing recognition. In: Proceedings of the 2023 8th International Conference on Machine Learning Technologies. pp. 15–22 (2023)
24. Lattanzi, E., Calisti, L., Freschi, V.: Unstructured handwashing recognition using smartwatch to reduce contact transmission of pathogens. Ieee Access **10**, 83111–83124 (2022)
25. Lattanzi, E., Calisti, L., Freschi, V.: Unstructured handwashing recognition using smartwatch to reduce contact transmission of pathogens. IEEE Access **10**, 83111–83124 (2022). https://doi.org/10.1109/ACCESS.2022.3197279
26. Li, J., Cheng, K., Wang, S., Morstatter, F., Trevino, R.P., Tang, J., Liu, H.: Feature selection: A data perspective. ACM computing surveys (CSUR) **50**(6), 1–45 (2017)
27. Momeni, N., Valdés, A.A., Rodrigues, J., Sandi, C., Atienza, D.: Cafs: cost-aware features selection method for multimodal stress monitoring on wearable devices. IEEE Transactions on Biomedical Engineering **69**(3), 1072–1084 (2021)
28. National.Instruments: Pc-6251 datasheet (2020), http://www.ni.com/pdf/manuals/375213c.pdf, last accessed 2024-06-14
29. Nguyen, B., Coelho, Y., Bastos, T., Krishnan, S.: Trends in human activity recognition with focus on machine learning and power requirements. Machine Learning with Applications **5**, 100072 (2021)
30. Peng, H., Ying, C., Tan, S., Hu, B., Sun, Z.: An improved feature selection algorithm based on ant colony optimization. Ieee Access **6**, 69203–69209 (2018)
31. Peretti, S., Contoli, C., Lattanzi, E.: Complexity-aware features selection for wrist-worn human activity recognition. In: 2024 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops). pp. 799–804. IEEE (2024)
32. Rohde&Schwarz: Ngmo2 datasheet (2020), https://www.rohde-schwarz.com/it/brochure-scheda-tecnica/ngmo2/, last accessed 2024-06-14
33. Suto, J., Oniga, S., Lung, C., Orha, I.: Comparison of offline and real-time human activity recognition results using machine learning techniques. Neural computing and applications **32**, 15673–15686 (2020)
34. Sztyler, T., Stuckenschmidt, H.: On-body localization of wearable devices: An investigation of position-aware activity recognition. 2016 IEEE International Conference on Pervasive Computing and Communications (PerCom) (2016)
35. Trigoni.Zenodo, A.A.N.: Watchhar: A smartwatch imu dataset for activities of daily living. (2022), http://dx.doi.org/10.5281/zenodo.7092553 [Accessed: October 20th, 2023]
36. Zebari, R., Abdulazeez, A., Zeebaree, D., Zebari, D., Saeed, J.: A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. Journal of Applied Science and Technology Trends **1**(1), 56–70 (2020)
37. Zhou, H., Zhang, J., Zhou, Y., Guo, X., Ma, Y.: A feature selection algorithm of decision tree based on feature weight. Expert Systems with Applications **164**, 113842 (2021)