Human activity recognition (HAR) is vital in a wide range of real-life applications such as health monitoring of elderly people, abnormal behaviour detection and smart home management. HAR systems can employ smart human-computer interfaces and be parts of active, intelligent surveillance systems. The increasing use of high-tech mobile and wearable devices, such as smart phones, smart watches and smart bands, can be the key elements in building high accuracy models, as they can provide a tremendous number of signals. This research aims to develop and test a machine learning (ML) model, which can successfully recognize a performed activity using raw signals obtained by wearable devices. Photo-plethysmography – Daily Life Activities (PPG-DaLiA) dataset contains data related to 15 individuals wearing physiological and motion sensors. PPG-DaLiA was used as an input to a custom data segmentation model to obtain the respective training and testing dataset. Overall, 23 ML well-established models were employed. The weighted and the fine k-nearest neighbours, the fine Gaussian support vector machines and the bagged trees were the algorithms that achieved the best performance with a very high accuracy level.

1. Introduction

Human activity recognition (HAR) is currently one of the most popular research fields among the machine learning (ML) applications (Jobanputra et al., 2019). The widespread use of internet of things, the development of capable smart wearable devices, together with the introduction of new ML and deep learning (DL) algorithms are the main key factors that have contributed to the rapid spread of HAR systems over the last years (Hassan et al., 2018; Wan et al., 2020). For an increasing number of people, the ‘always attached to the body’ devices are becoming essential in everyday life. Apart from their main functionality, smart phones, smart watches and smart bands can collect, in almost real-time, vast data related to the activity performed by a specific person (Lara & Labrador, 2012; Psathas et al., 2020). HAR models integrate these data with ML algorithms to...
recognize the underlying activity (Boukhechba et al., 2019; Psathas et al., 2020). Even a simple commercial smart phone or a smart watch may contain a large variety of embedded sensors such as accelerometers, gyroscopes, magnetometers, oximeters, electro cardiographers and light sensors that could provide all data required by an HAR system (Chen et al., 2018; Ronao & Cho, 2016; Wan et al., 2020).

This research used a public dataset acquired by two wearable devices to recognize 9 real-life activities performed by 15 individuals. This dataset was chosen due to its large number of measurements for long periods (2.5 h for each subject) and since it comprises a large number of ‘close to real-life’ activities.

In Psathas et al. (2020) the same dataset has been used. The overall accuracy was as high as 92.8% and it was achieved by combining the bagged trees (BGT) algorithm with the default settings and properties of Photoplethysmography – Daily Life Activities (PPG-DaLiA) dataset.

The research is a major extension of Psathas et al. (2020) and it has resulted in a severely increased level of overall accuracy. It has the following characteristics: (a) a subset of the original dataset has been considered (only the raw data have been used) and (b) variable sampling technique has been employed in the activity distinction and data segmentation processes.

The rest of this paper is organized as follows: Section 2 presents previous indicative HAR research studies. Section 3 describes the PPG-DaLiA dataset, while Section 4 describes the proposed data preprocessing and preparation methodology. Section 5 presents the classification approaches used. Experimental results are presented in Section 5. Section 6 concludes this paper and discusses the results and future work.

2. Literature review

The HAR model is a challenging and growing field of research in ML that employs sensors and computer vision systems (Jobanputra et al., 2019). Over the past years, HAR models mainly relied on sensor-based systems, but the rapid evolution of computer vision (CV) and DL developed sophisticated algorithms that can provide higher accuracy. It is a fact that CV-based algorithms still face challenges due to the complexity caused by the variability of the background in everyday environments. More specifically, there are limitations due to the variable illumination conditions of each scene during the day. This problem reduces the tracking and positioning capability of each subject (Ferrari et al., 2020; Wan et al., 2020; Xu et al., 2019). Moreover, the aforementioned techniques are limited by the computational resources of the used infrastructure, and by the diversified, dynamic environment in which such systems need to operate (Ferrari et al., 2020; Junker et al., 2008).

Pirttikangas et al. (2006), in their pioneer research, tested a model that used several multilayer perceptron and k-nearest neighbours (k-NN) algorithms to recognize 17 activities to achieve an overall accuracy of 90.61%. Casale et al. (2011) used a wearable device and applied a random forest classification algorithm to model five distinct activities (walking, climbing stairs, talking with a person, staying standing and working on the computer). This research achieved an overall accuracy of 90%. A few years later, Ahmed and Louf, (2013) performed an HAR system, using case-based reasoning, support vector machines (SVMs) and neural networks (NN), to achieve an overall accuracy of 0.86, 0.62 and 0.59, respectively, for three specific activities (breathing, walking or running, sitting
and relaxing). Wang et al. (2013) introduced a wearable system using passive tags, which achieved an accuracy of 93.6%. Li et al. (2016) achieved an accuracy of 96% and F-score equal to 0.74 for 10 medical activities. In fact, Li et al. introduced an activity recognition system for complex and dynamic medical settings that use passive radio-frequency identification (RFID) technology. Shinmoto et al. (2016) developed a system with RFID tags that could recognize two simple but very important classes (bed and chair exit alerts for elders), achieving an overall accuracy of 94%. Brophy et al. (2018) proposed a hybrid convolutional neural network and an SVM model with an accuracy of 92.3% for four activities (walking and running on a treadmill, low and high resistance bike exercise). Ryoo et al. (2018) proposed a backscattering activity recognition network of tags, which comprises a network of passive RF tags capable of recognizing daily human activities with an average error of 6%. In Boukhechba et al. (2019), a DL NN was developed for five activities (standing, walking, jogging, jumping and sitting). The F1 score was 0.86. Psathas et al. (2021) developed the wireless identification and sensing platform infrastructure with wearable RFID tags to recognize four activities in a controlled and monitored hospital room with a success rate of 98.93% on the training data.

3. Dataset description

PPG-DaLiA dataset was published by Reiss et al. (2019). It is a publicly available multimodal dataset that contains physiological and motion data recorded from a wrist- and a chest-wearable device. It contains data for seven males and eight females, aged between 21 and 55 years, performing eight distinct activities. The transition time between activities was also recorded and labelled as the ‘zero’ activity. All the activities were performed as close to real-life conditions as possible, and the recorded signals for each subject lasted approximately 2.5 h.

Raw sensor data were recorded using RespiBAN professional (RespiBAN, 2021) and Empatica E4 (Empatica, 2021) wearable devices. The sensor data provided by the two devices are described in Table 1.

For each subject, the following characteristics were also mentioned in the dataset:

1. Age (age of the subject).
2. Gender (male or female).
3. Height (measured in centimetres).
4. Weight (measured in kilograms).
5. Skin type (SKT) according to the Fitzpatrick scale (Fitzpatrick, 1988).
6. Fitness level (FL) which is an index on a scale 1–6 related to how often the subject exercises.

Table 1. Raw data from sensors in RespiBAN and Empatica E4 devices.

| Factor/abbreviation | Device          | Sampling rate (Hz) |
|---------------------|-----------------|--------------------|
| Electrocardiogram   | RespiBAN        | 700                |
| Chest-device        | RespiBAN        |                    |
| Blood volume        | Empatica E4     | 64                 |
| Wrist-device        | Empatica E4     | 32                 |
| Electrodermal activity | Empatica E4   | 4                  |
| Body temperature    | Empatica E4     |                    |
An FL value of 1 is assigned to a person that exercises less than once a month, whereas a value of 6 corresponds to 5–7 times a week. The profile of the subjects that participated in this research is presented in Table 2, where \( i = 1–15, j = \{m \text{ (for Male)}, f \text{ (for Female)}\} \) SKT is skin type, FNT is fitness type. The subjects are ordered by gender.

The recorded activities, the average duration of each activity and their descriptions are given in Table 3.

According to the developers of the dataset, there was only one major hardware issue during the data collection process, due to which the recorded data about person S6 are only valid for the first 1.5 h.

The available number of signal samples per group of factors (as given in Table 1) with the test duration and the number of distinct activity labels for each subject are given in Table 4.

The PPG-DaLiA dataset includes one more factor, which was not originally measured by any sensor, but it was calculated by the dataset developers. The heart rate in beats per minute (bpm) was calculated using the raw signals of the RespiBAN electrocardiogram signals. That extra factor, named ECGbpm, was manually calculated using a sliding window with a length of 8 s and a shift of 2 s. The segmentation technique will be explained in detail in the next section, but it was why the same sliding window and offset were also used in past research of our team (Psathas et al., 2020).

### Table 2. Properties of the subjects considered in this research.

| \( S_{ij} \) | Age  | Height | Weight | SKT | FL | Age  | Height | Weight | SKT | FL |
|-------------|------|--------|--------|-----|----|------|--------|--------|-----|----|
| S\(_{1,m}\) | 34   | 182    | 78     | 3   | 6  | S\(_{5,f}\) | 21     | 180    | 70   | 3   | 4  |
| S\(_{2,m}\) | 28   | 189    | 80     | 3   | 5  | S\(_{6,f}\) | 37     | 176    | 70   | 3   | 1  |
| S\(_{3,m}\) | 25   | 170    | 60     | 3   | 5  | S\(_{7,f}\) | 21     | 168    | 58   | 3   | 2  |
| S\(_{4,m}\) | 25   | 168    | 57     | 4   | 5  | S\(_{9,f}\) | 28     | 167    | 60   | 4   | 5  |
| S\(_{5,m}\) | 43   | 179    | 70     | 3   | 5  | S\(_{10,f}\) | 55     | 164    | 56   | 4   | 5  |
| S\(_{12,m}\) | 43  | 195    | 105    | 3   | 5  | S\(_{11,f}\) | 24     | 168    | 62   | 3   | 5  |
| S\(_{13,m}\) | 28  | 183    | 79     | 2   | 5  | S\(_{13,f}\) | 21     | 170    | 63   | 3   | 6  |

### Table 3. Description of activities included in the PPG-DaLiA dataset.

| Class/Activity | ID | Duration (min) | Description |
|---------------|----|----------------|-------------|
| Sitting still | 1  | 10             | Sitting still while reading (motion artefact-free baseline). |
| Ascending/Descending stairs | 2  | 5              | Climbing six floors up and going down again, repeating this twice. |
| Table soccer  | 3  | 5              | Playing table soccer, 1 vs. 1 with the supervisor of the data collection process. |
| Cycling       | 4  | 8              | Performed outdoors, around research campus, following a defined route of about 2 km length with varying road conditions (gravel and paved). |
| Car driving   | 5  | 15             | Started at the parking ground of our research campus and was carried out within the area nearby. Subjects followed a defined route which took about 15 min to complete. The route included driving on different streets in a small city and driving on country roads. |
| Lunch break   | 6  | 30             | This activity was carried out at the canteen of the research campus. The activity included queuing and fetching food, eating and talking at the table. |
| Walking       | 7  | 10             | Walking back from the canteen to the office, with some detour. |
| Working       | 8  | 20             | Subjects returned to their desk and worked on a computer. |
| Transient periods | 0 |                | Before and after each activity, a transient period was included, to arrive at the starting location of the next activity. |
version, we are not employing the ECGbpm factor to use an advanced variable sampling and segmenting approach.

4. Data segmentation and prepossessing and classification methodology

The dataset chosen in this research was populated by millions of recorded raw signals originally acquired by two wearable devices using different sampling rates per factor and device. It is impossible to use this dataset before aligning and transforming all raw data to homogenized data vectors with the same instances, regardless of the sampling.

4.1. Variable data segmentation using fast Fourier transformation

The Fourier analysis converts a signal from its original domain (often time or space) to a representation in the frequency domain and vice versa. The fast Fourier transform (FFT) is a specific algorithm of low complexity that computes the discrete Fourier transform (DFT) of a sequence. It is an extremely powerful mathematical tool that allows observing the obtained signals from the time domain to the frequency domain inside, which several difficult problems become very simple to analyze (Brigham, 1988; Nussbaum, 1981).

The DFT is obtained by decomposing a sequence of values into components of different frequencies. Any periodic function \( g(x) \) integral in the domain \( D = [-\pi, \pi] \) can be written as an infinite sum of sine and cosine as follows:

\[
g(x) = \sum_{k=-\infty}^{\infty} \tau_k e^{jkx}\tag{1}
\]

\[
\tau_k = \frac{1}{2\pi} \int_D g(x) e^{-jkx} \, dx \tag{2}
\]

where \( e^{i\theta} = \cos(\theta) + j\sin(\theta) \). The idea that a function can be broken down into its constituent frequencies is a powerful one and the backbone of the Fourier transform (Zhang, 2015). An FFT rapidly computes such transformations by factorizing the DFT matrix into

Table 4. The available number of signal samples per group of factors together with activity labels and test duration.

|          | 700 Hz factors | 64 Hz factors | 32 Hz factors | 4 Hz factors | Activity labels | Test duration (min) |
|----------|----------------|---------------|---------------|--------------|-----------------|--------------------|
| Subject 1| 6,448,403      | 589,571       | 294,787       | 36,851       | 36,851          | 154                |
| Subject 2| 5,743,503      | 525,123       | 262,563       | 32,823       | 32,823          | 137                |
| Subject 3| 6,118,703      | 559,427       | 279,715       | 34,967       | 34,967          | 146                |
| Subject 4| 6,405,003      | 585,603       | 292,803       | 36,603       | 36,603          | 153                |
| Subject 5| 6,513,503      | 595,523       | 297,763       | 37,223       | 37,223          | 155                |
| Subject 6| 3,675,003      | 336,003       | 168,003       | 21,003       | 21,003          | 88                 |
| Subject 7| 6,540,103      | 597,955       | 298,979       | 37,375       | 37,375          | 156                |
| Subject 8| 5,656,003      | 517,123       | 258,563       | 32,323       | 32,323          | 135                |
| Subject 9| 5,992,003      | 547,843       | 273,923       | 34,243       | 34,243          | 143                |
| Subject 10| 7,453,603     | 681,475       | 340,739       | 42,595       | 42,595          | 177                |
| Subject 11| 6,333,603     | 579,075       | 289,539       | 36,195       | 36,195          | 151                |
| Subject 12| 5,539,803     | 506,499       | 253,251       | 31,659       | 31,659          | 132                |
| Subject 13| 6,395,203     | 584,707       | 292,355       | 36,547       | 36,547          | 152                |
| Subject 14| 6,270,603     | 573,315       | 286,659       | 35,835       | 35,835          | 149                |
| Subject 15| 5,557,303     | 508,099       | 254,051       | 31,759       | 31,759          | 132                |
a product of sparse (mostly zero) factors. As a result, it reduces the complexity of computing the DFT from \( O(N^2) \), which arises if one applies the definition of DFT to \( O(N \log N) \), where \( N \) is the data size.

To perform FFT to our raw data, we need to define the \( N \) data size and the time shift \( D \) of each segment, as shown in Figure 1.

As mentioned in Section 2, dataset developers used the combination of 8 s for windows size and 2 s for windows sifting, to calculate the ECGbpm factor. The number of instances per activity and per subject obtained using the above parameters is presented in Table 5, while the classification results, shown in Psathas et al. (2020), is presented in Figure 2.

Although the results shown in the above-confusion matrix are really good, it is clear that most of the misclassified cases are related to the activity 'zero' which is the transition activity. There are also a few other misclassified cases, but most of them are related to activity zero. To improve the classification further, a variable window length has been employed and the \textit{windows shift segmentation} approach is proposed in this paper. This is another novelty of this research effort.

According to Ferrari et al. (2020), the length of each segment cannot be a random value. It has to be carefully selected based on the type of the performed activity and its duration. Ferrari et al. (2020) and Micucci et al. (2017) proposed a window length of around 3 s when the performed activity is \textit{slow walking}. They both agree on the facts that (a) the cadence of an average person walking is within 90–130 steps/min (Medrano et al., 2016) and (b) at least a full walking cycle (two steps) is preferred on each window. They also conclude that very small window size could produce several segments 'without any activity at all' if the performed activity is a soft one. On the other hand, big window size could produce segments with high overlapping unless a very high offset is selected. In that case, several activity changes may be lost due to the fast-shifting of windows.

It is more than clear that the selection of window length and offset is a key factor to build an accurate HAR system.

It is not always easy to determine the nature of the monitored activities or their duration, so a \textit{variable segmentation methodology} (VSM) is proposed in this paper. This is one of the main novelties of this research. According to the VSM, the system receives
Table 5. Instances after applying FFT for 8 s windows size and 2 s windows sifting.

| Subject | ACT₀ | ACT₁ | ACT₂ | ACT₃ | ACT₄ | ACT₅ | ACT₆ | ACT₇ | ACT₈ | Instances per subject |
|---------|------|------|------|------|------|------|------|------|------|-----------------------|
| S₁      | 1,141| 349  | 143  | 172  | 205  | 444  | 1,178| 377  | 594  | 4,603                 |
| S₂      | 1,260| 300  | 133  | 152  | 194  | 460  | 610  | 345  | 645  | 4,099                 |
| S₃      | 1,005| 300  | 218  | 146  | 189  | 455  | 1,080| 369  | 605  | 4,367                 |
| S₄      | 1,260| 284  | 261  | 163  | 237  | 503  | 948  | 297  | 619  | 4,572                 |
| S₅      | 1,287| 300  | 232  | 144  | 215  | 415  | 1,127| 295  | 634  | 4,649                 |
| S₆      | 1,145| 317  | 215  | 155  | 266  | 524  | 0    | 0    | 0    | 2,622                 |
| S₇      | 1,095| 309  | 203  | 137  | 251  | 518  | 1,212| 355  | 588  | 4,668                 |
| S₈      | 1,042| 300  | 206  | 161  | 209  | 384  | 900  | 240  | 595  | 4,037                 |
| S₉      | 1,001| 300  | 241  | 142  | 277  | 478  | 921  | 295  | 622  | 4,277                 |
| S₁₀     | 1,772| 298  | 233  | 152  | 243  | 468  | 1,042| 541  | 572  | 5,321                 |
| S₁₁     | 1,014| 300  | 228  | 162  | 245  | 430  | 1,205| 327  | 610  | 4,521                 |
| S₁₂     | 994  | 301  | 242  | 140  | 240  | 445  | 730  | 265  | 597  | 3,954                 |
| S₁₃     | 1,191| 302  | 248  | 145  | 249  | 449  | 1,014| 355  | 612  | 4,565                 |
| S₁₄     | 1,282| 303  | 227  | 162  | 250  | 477  | 866  | 307  | 602  | 4,476                 |
| S₁₅     | 1,026| 306  | 209  | 177  | 203  | 393  | 721  | 329  | 602  | 3,966                 |
| Instances per ACT | 17,515 | 4,569 | 3,239 | 2,310 | 3,473 | 6,843 | 13,554 | 4,697 | 8,497 | 64,697 |

Figure 2. Confusion matrix of the ensemble BGT algorithm for a windows size of 8 s and windows sifting of 2 s.
the sensor’s signals which are buffered. Finally, it creates multiple simultaneous segments using several predefined combinations of window lengths and offsets. As a result, each moment is represented by more than one segment. The number of segments for each moment is equal to the total number of combinations of window lengths and offsets.

The proposed VSM algorithm is presented in the following algorithm 1:

Algorithm 1. Detailed steps of the proposed VSM

1. Open sensor signal stream
2. Start buffering stream with a buffer size up to maximum window length
3. Set execution interval
4. Set accuracy threshold
5. Set window index $w_i$ to 0
6. For each selected window length $l$, and window offset $j$ do
   7. compute instance $\text{instance}_i[w_i]$ using FFT on the selected $\text{Segment}[l,j]$
   8. $w_i \leftarrow w_i + 1$
   9. end for
10. Apply the desired ML algorithm for classifying HAR using all calculated instances $\text{instance}_i[]$
11. If similar HAR results $> \text{accuracy}$
12. then successful HAR
13. else no decision can be made else / no successful HAR
14. At interval moment Go to step 5

One of the key features of the proposed methodology is that at each interval all combinations of window length-offset are used and tested. If the results of all combinations point to the same class, then the system has a strong decision. If this is not the case, then the accuracy threshold is checked to decide if one class can be accepted.

The testing of the proposed methodology is presented in Section 5 using 25 different length-offset combinations, which will be described in the next section.

4.2. Data prepossessing

PPG-DaLiA dataset consists of 15 python serialized files. Each file includes all sensor signals and the performed activity of each subject (Figure 3(a)).

The first step is the development of the necessary Python scripts to de-serialize the provided .pkl files and write the data vectors in plain text files. Python writes the respective content in the *.txt files for each *.pkl file, 135 in total (see Figure 1(b)). All text files have one column, except for the i_ACC_chest.txt and the i_ACC_wrist.txt files that have three columns. The abbreviations are given in Table 1 and i represents the number of the subject ($i = 1, \ldots, 15$). To facilitate the rest of the process, we have developed and executed an original Matlab script named Matlab facilitator (MLB_FCL).

MLB_FCL reads all txt files created by the Python and prepares the data vectors to perform the segmentation procedure, described in Section 4.1. For this specific experiment, a combination of 25 window lengths and offsets was used. Five different values for window length $l = \{4, 5, 6, 7, 8\}$ and five more for window offset $j = \{1, 2, 3, 4, 5\}$ were used. First, we read and converted each *.txt file to a Matlab table. The i_ACC_chest.txt and the i_ACC_wrist.txt (three accelerometers $x, y, z$) have been converted to three
tables, one for every column (Figure 3). Every feature is segmented with the combination of window length $l$ and window offset $j$ (Figure 3(d)) and the corresponding tables were created. For each timeseries segment of Activity$_{i,lj}$, the dominant value is held (Figure 3(d)) FFT is applied on each time-series segment of the rest features (Figure 3). The final dataset, after developing all combinations and merging all tables into one final table, counts 1386,994 data vectors with 11 independent variables, as given in Table 1, and one labelled variable, which is the performed activity.

The procedure of de-serialization and processing of all files to create the final and usable dataset is illustrated in Figure 3.

A total of 23 classification algorithms have been employed, namely fine tree, medium tree, coarse tree, linear discriminant, quadratic discriminant, linear SVM, quadratic SVM, cubic SVM, fine Gaussian SVM, medium Gaussian SVM, coarse Gaussian SVM, cosine KNN, cubic KNN, weighted KNN, fine KNN, medium KNN, Gaussian naive Bayes, Kernel naive Bayes, boosted trees, BGT, subspace discriminant, subspace KNN, random under-sampling boost trees. The following sections present only the architecture and the results of the four most robust algorithms with the highest values of performance indices.
4.3. **Weighted k-nearest neighbours (Wk-NN)**

Classifying query points based on their distance to specific points (or neighbours) in a training dataset can be a simple yet effective process. The k-NN is a lazy and non-parametric learning algorithm. Given a set $X$ of $n$ points and a distance function, k-NN search finds the $k$ closest points to a query point or set of them (Hechenbichler & Schliep, 2004). Dudani (1978) introduced a weighted voting method called the distance weighted (DW) Wk-NN rule. According to this approach, the closer neighbours are weighted more heavily than the farther ones using the DW function. The weight $w_i$ for the $i$-th nearest neighbour of the query $x'$ is defined in the following equation:

$$
 w'_i = \begin{cases} 
 \frac{d(x'x_{NN}^k) - d(x'x_{NN}^1)}{d(x'x_{NN}^k) - d(x'x_{NN}^1)}, & \text{if } d(x'x_{NN}^k) \neq d(x'x_{NN}^1) \\
 1, & \text{if } d(x'x_{NN}^k) = d(x'x_{NN}^1) 
\end{cases}
$$

Finally, the classification result of the query is determined by the majority weighted voting using the following equation:

$$
 y' = \arg \max_y \sum_{(x_{NN}^i, y_{NN}^i) \in T} w'_i \times \delta(y = y_{NN}^i).
$$

Based on Equation (4), a neighbour with a shorter distance is weighted more heavily than one with a greater distance: the nearest neighbour is assigned a weight equal to 1, whereas the furthest one is assigned a weight of 0 and the weights of the others are scaled linearly to the interval in between.

4.4. **Fine k-NN**

The k-NN is a lazy and non-parametric classification method (Cover & Hart, 1967). The k-NN classifier is a traditional classification rule. An unlabelled vector (a query or test point) is classified by assigning the label, which is most frequent among the $k$ training samples nearest to that query point. Given a set $X$ of $n$ points and a distance function, k-NN search finds the $k$ closest points to a query point or set of them (Hechenbichler & Schliep, 2004). A commonly used distance metric for k-NN is Euclidean distance calculated by the following equation, while other distance metrics used, depending on each occasion, are Hamming distance and Minkowski distance (Jaskowiak & Campello, 2011).

$$
 d(x, y) = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}
$$

The best choice of $k$ depends upon the data; generally, larger values of $k$ reduce the effect of noise on the classification (Jaskowiak & Campello, 2011), but that may change boundaries between classes to be less distinct. A good $k$ can be selected by various heuristic techniques. The accuracy of the k-NN algorithm can be negatively affected by the presence of noisy features. Over the years, many attempt have been made to make the algorithm more robust (Nigsch et al., 2006).

Fine k-NN refers to the nearest neighbour classifier that makes finely detailed distinctions between classes with the number of neighbours set to 1 and uses Euclidian distance to determine the nearest neighbours (Johnson & Yadav, 2018).
4.5. Fine Gaussian SVMs

A SVM algorithm classifies the data by finding the best hyperplane that separates all the data of one class from those of the other class (Gutierrez, 2015). It is a supervised learning technique based on the statistical learning theory employed for classification and regression tasks (Boser et al., 1992). For the binary separation of vectors in classes, the function is defined in the following equation:

\[ F(x) = \text{sgn}(f_w(x) + b) = \text{sgn}(w, x + b) \]  

(6)

where \( w \) is the weight vector of the function and \( b \) is the bias, and

\[ \text{sgn}(x) = \begin{cases} 
-1, & \text{for } x < 0 \\
0, & \text{for } x = 0 \\
1, & \text{for } x > 0 
\end{cases} \]  

(7)

In a linearly separable case, given a training dataset, \( \mathbf{x}_i \in \mathbb{R}^n \) and their corresponding label \( y_i \in \{+1, -1\} \), \( i = 1, \ldots, N \), a separator can be defined as follows:

\[ F(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i - b = 0 \]  

(8)

The solution to the problem of finding the hyperplane can be determined using the Lagrange relaxation procedure in the following equations:

\[
\max_{\alpha} \left\{ \sum_{i=1}^{m} a_i y_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} a_i y_i a_j y_j x_i^T x_j \right\}
\]  

(9)

\[
\sum_{i=1}^{m} a_i y_i = 0 \quad \text{and} \quad a_i \geq 0, \quad i = 1, 2, \ldots, m
\]  

(10)

where \( \alpha = [\alpha_1, \ldots, \alpha_N] \) are the non-negative Lagrange multipliers. The solution to the above equation gives the location of the separating hyperplane. In the case data are non-linear, there is a way to create a non-linear class separator by applying the Kernel function (Hofmann et al., 2008). Kernels project the input vector space into a feature vector space of higher dimensions, seeking to make separation easier in space. The solution to the dual problem with the projection of Equations (9) and (10) now transforms to

\[
\max_{\alpha} \left\{ \sum_{i=1}^{m} a_i y_i - \frac{1}{2} \sum_{i=1}^{m} \sum_{j=1}^{m} a_i y_i a_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \right\}
\]  

(11)

\[
\sum_{i=1}^{m} a_i y_i = 0 \quad \text{and} \quad C \geq a_i \geq 0, \quad i = 1, 2, \ldots, m
\]  

(12)

where \( K(\mathbf{x}_i, \mathbf{x}_j) \) is the Kernel function. The mathematical representation of Gaussian Kernel is as follows:

\[ K(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\gamma \mathbf{x}_i^T \mathbf{x}_j^2), \quad \gamma > 0 \]  

(13)

\( c \) and \( \gamma \) represent the hyperplane parameters and need to be optimized.
Fine Gaussian SVM refers to the SVM that makes finely detailed distinctions between classes with Kernel scale set to $\gamma = \sqrt{n}/4$ where $n$ is the number of features (Savas & Dovis, 2019).

### 4.6. Bagged trees

Bagging is a model averaging approach ML method of combining multiple predictors. It generates multiple training sets by sampling with replacement from the available training data (Breiman, 1994). Bagged Trees was deduced from the phrase ‘bootstrap aggregating’ (Breiman, 1997). Bootstrap aggregating improves classification and regression models in terms of stability and accuracy. It also reduces variance and helps to avoid overfitting. It can be applied to any type of classifiers. Bagging is a popular method in estimating bias, standard errors and constructing confidence intervals for parameters. In the case of classification into two possible classes, a classification algorithm creates a classifier $H: D \rightarrow \{-1,1\}$ on the basis of a training set of example descriptions (in our case played by a document collection) $D$. The bagging method creates a sequence of classifiers $H_m, m=1, \ldots, M$ with respect to modifications of the training set. These classifiers are combined into a compound classifier. The prediction of the compound classifier is given as a weighted combination of particular classifier predictions according to the following equation:

$$H(d_i, c_j) = \text{sign} \left( \sum_{m=1}^{M} a_m H_m(d_i, c_j) \right)$$  \hspace{1cm} (15)

Parameters $a_m, m = 1, \ldots, M$ are determined in such a way that more precise classifiers have a stronger influence on the final prediction than less precise classifiers. The precision of base classifiers $H_m$ can be slightly higher than that of a random classification. That is why these classifiers $H_m$ are called weak classifiers.

### 4.7. Evaluation metrics

Accuracy is the overall evaluation index of the developed ML models. This is a multi-class classification research, so the ‘One Versus All’ strategy (Joutsijoki et al., 2016) was used. The calculated validation indices have been calculated and considered are presented in Table 6.

### 5. Experimental results

The experiments were performed using Matlab R2019a. Due to the vast data and the big number of classifiers, a private distributed-parallelized environment of 10 virtual instances

| Table 6. Calculated indices for the evaluation of the multi-class classification approach. |
|---------------------------------------------------------------|
| Index   | Abbreviation | Calculation |
|-----------------|--------------|-------------|
| Sensitivity     | SNS, REC, TPR| $SNS = TP/(TP + FN)$ |
| Specificity     | SPC, TNR     | $SPC = TN/(TN + FP)$ |
| Accuracy        | ACC          | $ACC=(TP + TN)/(TP + FP + FN + TN)$ |
| F1 Score        | F1           | $F1 = 2*TP/(2*TP + FP + FN)$ |
| Precision       | PREC         | $PREC = TP/(TP + FP)$        |

A. PAPALEONIDAS ET AL.
of Matlab was employed. Each instance had access to eight i9-9900k threads and 32GB of memory, while it was equipped with two Matlab workers, which means that a total 20 workers were available.

The accuracy threshold was set to 0, which means that each classification was accepted only if all instances created by VSM had been classified to the same class. The zero threshold could be also called zero error tolerance and it was selected to check the model’s performance under the strictest configuration. The 10-fold cross validation was also applied to tests.

Tables 7–10 present the performance for each of the four best classification algorithms, using Sensitivity (SNS), Specificity (SPC), Accuracy (ACC), F1 Score and Precision (PRC).

The confusion matrix for the classification obtained using the BGT algorithm is shown in Figure 4. Due to the high accuracy and similarity of results, the rest of the confusion matrices are not included.

All employed metrics prove that the proposed model has an excellent performance. In all four algorithms Accuracy is higher than 99%, but all indices are proving a very well-

| Table 7. Activity (AC) classification performance indices for the weighted k-NN algorithm. |
|---------------------------------------------|
|                | AC - 0 | AC - 1 | AC - 2 | AC - 3 | AC - 4 | AC - 5 | AC - 6 | AC - 7 | AC - 8 | Overall |
| SNS            | 0.9985 | 0.9992 | 0.9977 | 0.9998 | 0.9988 | 0.9982 | 0.9999 | 0.9980 | 0.9999 | 0.9987 |
| SPC            | 0.9984 | 0.9996 | 0.9999 | 0.9999 | 0.9999 | 0.9997 | 0.9996 | 0.9990 | 0.9993 | 0.9982 |
| ACC            | 0.9994 | 0.9990 | 0.9999 | 0.9997 | 0.9968 | 0.9998 | 0.9997 | 0.9999 | 0.9997 | 0.9999 |
| F1             | 0.9989 | 0.9980 | 0.9986 | 0.9990 | 0.9983 | 0.9995 | 0.9996 | 0.9983 | 0.9998 | 0.9995 |
| PRC            | 0.9975 | 0.9977 | 0.9992 | 0.9966 | 0.9983 | 0.9989 | 0.9999 | 0.9992 | 0.9992 | 0.9985 |

| Table 8. Activity (AC) classification performance indices for the fine k-NN algorithm. |
|---------------------------------------------|
|                | AC - 0 | AC - 1 | AC - 2 | AC - 3 | AC - 4 | AC - 5 | AC - 6 | AC - 7 | AC - 8 | Overall |
| SNS            | 0.9968 | 0.9996 | 0.9988 | 0.9994 | 0.9994 | 1.0000 | 0.9998 | 0.9993 | 0.9998 | 0.9992 |
| SPC            | 0.9999 | 0.9993 | 0.9998 | 0.9991 | 0.9998 | 0.9987 | 0.9993 | 0.9998 | 0.9999 | 0.9999 |
| ACC            | 0.9995 | 0.9993 | 0.9993 | 0.9999 | 0.9995 | 0.9998 | 0.9999 | 0.9994 | 0.9997 | 0.9990 |
| F1             | 0.9978 | 0.9984 | 0.9976 | 0.9992 | 0.9976 | 0.9988 | 0.9999 | 0.9997 | 0.9995 | 0.9997 |
| PRC            | 0.9985 | 0.9983 | 0.9989 | 0.9968 | 0.9975 | 0.9991 | 0.9979 | 0.9983 | 0.9999 | 0.9990 |

| Table 9. Activity (AC) classification performance indices for the fine Gaussian SVM algorithm. |
|---------------------------------------------|
|                | AC - 0 | AC - 1 | AC - 2 | AC - 3 | AC - 4 | AC - 5 | AC - 6 | AC - 7 | AC - 8 | Overall |
| SNS            | 0.9979 | 0.9999 | 0.9976 | 0.9999 | 0.9993 | 0.9981 | 0.9999 | 0.9990 | 0.9996 | 0.9988 |
| SPC            | 0.9999 | 0.9997 | 0.9998 | 0.9999 | 0.9999 | 0.9996 | 0.9990 | 0.9992 | 0.9983 | 0.9969 |
| ACC            | 0.9989 | 0.9992 | 0.9990 | 0.9999 | 0.9999 | 0.9988 | 0.9977 | 0.9981 | 0.9995 | 0.9994 |
| F1             | 0.9972 | 0.9993 | 0.9978 | 0.9977 | 0.9978 | 0.9998 | 0.9987 | 0.9998 | 0.9973 | 0.9997 |
| PRC            | 0.9975 | 0.9990 | 0.9991 | 0.9967 | 0.9979 | 0.9998 | 0.9995 | 0.9983 | 0.9998 | 0.9993 |

| Table 10. Activity (AC) classification performance indices for the BGT algorithm. |
|---------------------------------------------|
|                | AC - 0 | AC - 1 | AC - 2 | AC - 3 | AC - 4 | AC - 5 | AC - 6 | AC - 7 | AC - 8 | Overall |
| SNS            | 0.9978 | 0.9989 | 0.9984 | 0.9998 | 0.9984 | 0.9991 | 0.9996 | 0.9989 | 0.9995 | 0.9989 |
| SPC            | 0.9993 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9998 |
| ACC            | 0.9989 | 0.9998 | 0.9998 | 0.9999 | 0.9998 | 0.9999 | 0.9999 | 0.9999 | 0.9999 | 0.9997 |
| F1             | 0.9980 | 0.9988 | 0.9984 | 0.9985 | 0.9984 | 0.9992 | 0.9996 | 0.9988 | 0.9994 | 0.9988 |
| PRC            | 0.9982 | 0.9987 | 0.9984 | 0.9973 | 0.9985 | 0.9992 | 0.9996 | 0.9988 | 0.9994 | 0.9987 |
tuned model. The above-confusion matrix also indicates a prodigious performance with only 120 (5, 56, 5 and 54) misclassified cases in between classes 1 and 8. The vast majority of the misclassified cases are related to the ‘transition period 0’ and it is something expected. However, looking at the evaluation indices (Tables 7–10) and the confusion matrix, it is shown that the performance is also extremely high (but not perfect) for class zero. Even for class zero, 375,671 cases are correctly identified, whereas the misclassified cases are less than 1000). It is obvious that during the transition period (which lasts a limited, almost zero period of time) more than one activity are actually performed. However, even for this difficult case, our modelling effort is very efficient.

6. Discussion and conclusions

The research has managed to correctly identify human activities by considering a dataset, obtained using two wearable devices. Using the introduced VSM, we could deploy a high
accurate HAR model and also recognize the transition between activity moments. The introduced modelling approach was extensively tested to distinguish nine distinct and specific activities close to real-life conditions. The results have proved that four classification algorithms can offer excellent models. The performance of the classification for six of the nine classes reached almost the absolute 100%, while it was extremely high for the other two classes, namely, 7 and 8. The efficiency of the proposed methodology is lower (but still very high) only for the case of the transition class, which should be studied further to be improved.

Psathas et al. (2020) introduced an HAR model that could achieve an accuracy of 92.8%. The extended version that has been introduced and tested has achieved an accuracy as high as 99.9% and it could identify the transition moments within the dataset.

Future work will concentrate on the minimization of the used features to build a lighter version of the model that could be trained in a shorter period of time. Also, an important matter is the improvement of the classification of the transition class ‘zero’.

Finally, it would be also interesting to ‘inverse’ the methodology to estimate the heartbeat rate. Together with the estimated activity, the system could be able to provide early warnings in the cases of heart problems.

ORCID
Antonios Papaleonidas http://orcid.org/0000-0002-0545-7638
Anastasios Panagiotis Psathas http://orcid.org/0000-0003-3779-6754
Lazaros Iliadis http://orcid.org/0000-0002-6404-1528

References
Ahmed, M. U., & Loutfi, A. (2013). Physical activity identification using supervised machine learning and based on pulse rate. International Journal of Advanced Computer Sciences and Applications, 4 (7), 210–217. https://www.diva-portal.org/smash/record.jsf?pid=diva2%3A638278&Ldswid=2205
Boser, B. E., Guyon, I. M., & Vapnik, V. N. (1992, July 27–29). A training algorithm for optimal margin classifiers. In Proceedings of the fifth annual workshop on computational learning theory (pp. 144–152). https://doi.org/10.1145/130385.130401
Boukhechba, M., Cai, L., Wu, C., & Barnes, L. E. (2019). ActiPPG: Using deep neural networks for activity recognition from wrist-worn photoplethysmography (PPG) sensors. Smart Health, 14, Article 100082. https://doi.org/10.1016/j.smhl.2019.100082
Breiman, L. (1994). Bagging predictors (Technical Report 421). Department of Statistics, University of California at Berkeley.
Breiman, L. (1997). Arcing the edge (Technical Report 486). Statistics Department, University of California at Berkeley.
Brigham, E. O. (1988). The fast Fourier transform and its applications. Prentice-Hall.
Brophy, E., Veiga, J. J. D., Wang, Z., Smeaton, A. F., & Ward, T. E. (2018). An interpretable machine vision approach to human activity recognition using photoplethysmograph sensor data. arXiv preprint arXiv:1812.00668.
Casale, P., Pujol, O., & Radeva, P. (2011, June 8–10). Human activity recognition from accelerometer data using a wearable device. In Iberian conference on pattern recognition and image analysis (pp. 289–296). Springer.
Chen, Z., Jiang, C., & Xie, L. (2018). A novel ensemble ELM for human activity recognition using smartphone sensors. IEEE Transactions on Industrial Informatics, 15(5), 2691–2699. https://doi.org/10.1109/TII.2018.2869843
Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory, 13*(1), 21–27. https://doi.org/10.1109/TIT.1967.1053964

Dudani, S. A. (1978). The distance-weighted k-nearest neighbor rule. *IEEE transactions on Systems, Man and Cybernetics, 8*(4), 311–313. https://doi.org/10.1109/TSMC.1978.4309958

Empatica. (2021, April). *Empatica E4 wristband*. Retrieved April 10, 2021, from https://www.empatica.com/en-eu/research/e4/

Ferrari, A., Micucci, D., Mobilio, M., & Napoletano, P. (2020). On the personalization of classification models for human activity recognition. *IEEE Access, 8*, 32066–32079. https://doi.org/10.1109/ACCESS.2020.2973425

Fitzpatrick, T. B. (1988). The validity and practicality of sun-reactive skin types I through VI. *Archives of dermatology, 124*(6), 869–871. https://doi.org/10.1001/archderm.1988.01670060015008

Gutierrez, D. D. (2015). *Machine learning and data science: An introduction to statistical learning methods with R*. Technics Publications.

Hassan, M. M., Uddin, M. Z., Mohamed, A., & Almogren, A. (2018). A robust human activity recognition system using smartphone sensors and deep learning. *Future Generation Computer Systems, 81*, 307–313. https://doi.org/10.1016/j.future.2017.11.029

Hechenbichler, K., & Schliep, K. (2004). Weighted k-nearest-neighbor techniques and ordinal classification. Collaborative Research Center 386, Discussion Paper 386.

Hofmann, T., Schölkopf, B., & Smola, A. J. (2008). Kernel methods in machine learning. *The Annals of Statistics, 1171–1220*. https://doi.org/10.1214/009053607000000677

Jaskowiak, P. A., & Campello, R. J. G. B. (2011, August 10–12). Comparing correlation coefficients as dissimilarity measures for cancer classification in gene expression data. Proceedings of the Brazilian symposium on bioinformatics (pp. 1–8), Brasília.

Jobanputra, C., Bavishi, J., & Doshi, N. (2019). Human activity recognition: A survey. *Procedia Computer Science, 155*, 698–703. https://doi.org/10.1016/j.procs.2019.08.100

Johnson, J. M., & Yadav, A. (2018). Fault detection and classification technique for HVDC transmission lines using KNN. In D. K. Mishra, M. Nayak, & A. Joshi (Eds.), *Information and communication technology for sustainable development*. Lecture notes in networks and systems (Vol. 10, pp. 245–253). Springer.

Joutsijoki, H., Haponen, M., Rasku, J., Aalto-Setälä, K., & Juhola, M. (2016). Machine learning approach to automated quality identification of human induced pluripotent stem cell colony images. *Computational and Mathematical Methods in Medicine*. https://doi.org/10.1155/2016/3091039

Junker, H., Amft, O., Lukowicz, P., & Tröster, G. (2008). Gesture spotting with body-worn inertial sensors to detect user activities. *Pattern Recognition, 41*(6), 2010–2024. https://doi.org/10.1016/j.patcog.2007.11.016

Lara, O. D., & Labrador, M. A. (2012). A survey on human activity recognition using wearable sensors. *IEEE Communications Surveys & Tutorials, 15*(3), 1192–1209. https://doi.org/10.1109/SURV.2012.110112.00192

Li, A., Miao, Z., Cen, Y., & Liang, Q. (2016, March). Abnormal event detection based on sparse reconstruction in crowded scenes. In 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) (pp. 1786-1790). IEEE. ISO 690.

Medrano, C., Plaza, I., Igual, R., Sánchez, Á, & Castro, M. (2016). The effect of personalization on smartphone-based fall detectors. *Sensors, 16*(1), Article 117. https://doi.org/10.3390/s16010117

Micucci, D., Mobilio, M., & Napoletano, P. (2017). Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences, 7*(10), Article 1101. doi:10.3390/app7101101

Nigsch, F., Bender, A., van Buuren, B., Tissen, J., Nigsch, E., & Mitchell, J. B. (2006). Melting point prediction employing k-nearest neighbor algorithms and genetic parameter optimization. *Journal of chemical information and modeling, 46*(6), 2412–2422. https://doi.org/10.1021/ci060149f

Nussbaumer, H. J. (1981). The fast Fourier transform. In *Fast Fourier transform and convolution algorithms*. Springer Series in Information Sciences (Vol. 2, pp. 80–111). Springer.

Pirttikangas, S., Fujinami, K., & Nakajima, T. (2006, October 11–13). Feature selection and activity recognition from wearable sensors. In *International Symposium on Ubiquitous Computing Systems* (pp. 516–527). Springer.
Psathas, A. P., Papaleonidas, A., & Iliadis, L. (2020, November 30–December 3). Machine learning modeling of human activity using PPG signals. In *International Conference on Computational Collective Intelligence* (pp. 543–557). Springer.

Psathas, A. P., Papaleonidas, A., & Iliadis, L. (2021, June 25–27). *A machine learning approach for recognition of elders’ activities using passive sensors*. IFIP International conference on artificial intelligence applications and innovations. Springer.

Reiss, A., Indlekofer, I., Schmidt, P., & Van Laerhoven, K. (2019). Deep ppg: Large-scale heart rate estimation with convolutional neural networks. *Sensors*, 19(14), Article 3079. [https://doi.org/10.3390/s19143079](https://doi.org/10.3390/s19143079)

RespiBAN. (2021, April). *RespiBAN professional*. Retrieved April 10, 2021, from [https://biosignalsplux.com/products/wearables/respiban-pro.html](https://biosignalsplux.com/products/wearables/respiban-pro.html)

Ronao, C. A., & Cho, S. B. (2016). Human activity recognition with smartphone sensors using deep learning neural networks. *Expert Systems with Applications*, 59, 235–244. [https://doi.org/10.1016/j.eswa.2016.04.032](https://doi.org/10.1016/j.eswa.2016.04.032)

Ryoo, J., Karimi, Y., Athalye, A., Stanačević, M., Das, S. R., & Djurić, P. (2018, June 10–15). *Barnet: Towards activity recognition using passive backscattering tag-to-tag network*. Proceedings of the 16th annual international conference on mobile systems, applications, and services (pp. 414–427).

Savas, C., & Dovis, F. (2019). The impact of different kernel functions on the performance of scintillation detection based on support vector machines. *Sensors*, 19(23), Article 5219. [https://doi.org/10.3390/s19235219](https://doi.org/10.3390/s19235219)

Shinmoto Torres, R. L., Visvanathan, R., Hoskins, S., Van den Hengel, A., & Ranasinghe, D. C. (2016). Effectiveness of a batteryless and wireless wearable sensor system for identifying bed and chair exits in healthy older people. *Sensors*, 16(4), 546. [https://doi.org/10.3390/s16040546](https://doi.org/10.3390/s16040546)

Wan, S., Qi, L., Xu, X., Tong, C., & Gu, Z. (2020). Deep learning models for real-time human activity recognition with smartphones. *Mobile Networks and Applications*, 25(2), 743–755. [https://doi.org/10.1007/s11036-019-01445-x](https://doi.org/10.1007/s11036-019-01445-x)

Wang, L., Gu, T., Xie, H., Tao, X., Lu, J., & Huang, Y. (2013, December 2–4). A wearable RFID system for real-time activity recognition using radio patterns. In *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services* (pp. 370–383), Springer.

Xu, C., Chai, D., He, J., Zhang, X., & Duan, S. (2019). InnoHAR: A deep neural network for complex human activity recognition. *IEEE Access*, 7, 9893–9902. [https://doi.org/10.1109/ACCESS.2018.2890675](https://doi.org/10.1109/ACCESS.2018.2890675)

Zhang, Z. (2015). Photoplethysmography-based heart rate monitoring in physical activities via joint sparse spectrum reconstruction. *IEEE Transactions on Biomedical Engineering*, 62(8), 1902–1910. [https://doi.org/10.1109/TBME.2015.2406332](https://doi.org/10.1109/TBME.2015.2406332)