Improving Activity Recognition Accuracy in Ambient Assisted Living Systems by Automated Feature Engineering

Eftim Zdravevski, Petre Lameski, Vladimir Trajkovik, Andrea Kulakov, Ivan Chorbev,
Rossitza Goleva, Nuno Pombo and Nuno Garcia

***Abstract*—Ambient Assisted Living (AAL) is promising to become a supplement of the current care models, providing enhanced living experience to people within context-aware homes and smart environments. Activity recognition based on sensory data in AAL systems is an important task because: it can be used for estimation of levels of physical activity; can lead to detecting changes of daily patterns that may indicate an emerging medical condition, or can be used for detection of accidents and emergencies. To be accepted, AAL systems need to be affordable while providing reliable performance. These two factors hugely depend on optimizing the number of utilized sensors and extracting robust features from them. This study proposes a generic feature engineering method for selecting robust features from variety of sensors, which in turn can be used for generating reliable classification models. From the originally recorded time series and some newly generated time series (i.e. magnitudes, first derivatives, delta series, and FFT-based series), a variety of time and frequency domain features are extracted. Then, using two-phase feature selection, the number of generated features is greatly reduced. Finally, different classification models are trained and evaluated on an independent test set. The proposed method was evaluated on five publicly available datasets and on all of them it yielded better accuracy than when using hand-tailored features. The benefits of the proposed systematic feature engineering method are: quickly discovering good feature sets for any given task than manually finding ones suitable for a particular task; selecting a small feature set that outperforms manually determined features in both execution time and accuracy; and identification of relevant sensor types and body locations automatically. Ultimately, the proposed method could reduce the cost of AAL systems by facilitating execution of algorithms on devices with limited resources and by using as few sensors as possible.**

***Index Terms*—feature extraction, time series analysis, ambient intelligence, wearable sensors, sensor fusion, pattern recognition, data mining, data preprocessing, body sensor networks**

I. Introduction

The increasing age of the citizens in the developed countries demands an optimization of health care systems. Expenses for the care of ageing citizens is increasing, while the number of people contributing to health care funds is decreasing. In [1] the projections of the future health expenditures in the European Union are elaborated. The authors conclude that the growing proportion of elderly population will increase the demand for health care services, thus increasing the costs in resources, both human and financial. The report also suggests that the investment in medical science, technology and treatment techniques has the potential to reduce the cost of health care services in the mid and long term.

In that direction, Ambient Assisted Living (AAL) is a relatively new ICT trend to enable new products, services, and processes to transparently support people (mostly elderly) in their preferred living environment aiming to improve their quality of life [2]. In studies related to AAL, recognition of Activities of Daily Living (ADLs) based on sensory recordings has become very popular [3], [4], [5], [6], [7] because of several reasons. First, health benefits related to physical activity (PA) depend on the intensity, duration, and frequency of PA [8], [9], [10], therefore its correct estimation is important. Next, changes in ADLs can indicate an emerging medical condition before it becomes critical [11], therefore ADL recognition is important. Finally, human activity recognition is essential for detecting accidents and immediate emergencies, such as falls.

Generally, ADL recognition approaches are based on signal processing and statistical methods for feature extraction from the raw sensory measurements. Additionally, they employ machine learning algorithms for generating models that can automatically recognize and classify different activities, such as walking, jogging, standing, sitting, lying, cooking, etc. Most approaches are currently based on hand-tailored features based on literature recommendations and are specific to the activities of interest [3], [4], [5], [6], [7], [12], [13], [14], [15], [16]. Some features based on basic statistics (e.g. mean, maximum, minimum, etc.) are encountered in most studies, however others are used only in some studies, so there is lack of clear understanding which features are truly useful for activity recognition. Another confusion arises from the fact that some studies use more sensors from various types than others. Likewise, some studies use noise filtering and smoothing before extracting features, while others do not. These discrepancies are expected, considering the variety of available sensors or the diversity of goals in different studies.

The study presented in this article aims to eliminate the need of manually designing features for each activity recognition problem, while considering the variations originating from the large number of users and the diversity of sensors and activities. A systematic and generic method for feature engineering that can be used in any application for activity recognition based on sensory data is proposed. This is accomplished by a variety of feature extraction and selection techniques, which eventually yield feature sets that are most suitable for each task. To verify the method, we have used five publicly available datasets and compared the results to the original studies that published the datasets, which used manually engineered features. The ultimate goal of our algorithm is to provide compelling evidence on which sensor locations and which features are optimal for a particular activity recognition problem. Hopefully, this will lead to lowering the cost of AAL systems, reducing intrusiveness to subjects by using smaller number of ubiquitous sensors, facilitating execution on hardware with limited resources due to the low number of features, while having reliable performance. All of these properties of AAL systems are the main prerequisites for their successful deployment and acceptance.

II. Methods and Analysis

In this study, we use several publicly available datasets related to activity recognition using body sensors. For each dataset, first we apply segmentation with sliding windows, and then extract time and frequency domain features from the raw sensory readings. Next, with feature selection the number of features is reduced to obtain more robust models and to shorten the model building and recognition time [17]. Finally, using several machine learning algorithms we generate classification models using the reduced feature sets. The datasets, the systematic feature engineering, the feature selection process, and the used machine learning algorithms are described in details in the forthcoming subsections.

A. Data fusion and segmentation

Before proceeding to feature extraction and selection and consequently training machine learning algorithms, the data needs to be segmented in a suitable way. In [18] is provided an overview of the data preparation steps commonly used in activity recognition systems. Additionally, the window size impact in human activity recognition is discussed.

The process of segmenting data streams with sliding windows is shown in Fig. 1. The data fusion consists of parsing, formatting, time alignment and data-type mapping of the incoming data. After the fusion, all data streams are unified into one stream, which is afterwards stored in a database.

The variety of sensors can produce data at different rates. For each sensor measurement, there is a timestamp t associated with it. This timestamp is essential for properly transforming the unified data stream into segments of windowed data. For simplicity, let us assume that all sensor measurements are stored in one entity R which has the following columns: the timestamp t, the value v and the sensor s.

In Fig. 1, the k-th window (w_k) consists of all data that was collected during the period $(ts_k,\ te_k]$. We denote the set of measurements from the i-th sensor that belong in the window w_k with m_i^k, as expressed with (1). Likewise, the range $[ts_{k+1}, te_k]$ defines the overlapping segment between windows w_k and w_{k+1}.

$$m_i^k = \sigma_{s=i \text{ and } t>ts_k \text{ and } t\leq te_k}(R) \tag{1}$$

Let δ_i denote the time difference between consecutive measurements m_i of the i-th sensor. Further, let μ denote the minimum time difference between consecutive measurements of all sensors, as defined with (2). Let a and b denote the windowing and overlapping coefficients, respectively, where $a > b$ and $a, b \in \mathbb{N}^+$. Thus, the window length is $w = a \times \mu$ and the overlapping segment length is $o = b \times \mu$.

$$\mu = \min_i \delta_i \tag{2}$$

Next, in order to determine the number of samples per window (n_i) for the i-th sensor, we use (3), which performs ceiling rounding and returns an integer.

$$n_i = \lceil w/\delta_i \rceil \tag{3}$$

Generally, various sensors can generate data at different rates. If sensor l generates data very rarely (i.e. has low frequency), and sensor p generates data very often (i.e. has high frequency), then $\delta_l \gg \delta_p$. As a result, also $\delta_l \gg \mu$ stands. In some cases, $\delta_l \gg a \times \mu$ may additionally stand, which means the low-frequency sensors may not produce a measurement within a time window. However, even for such sensors l, (3) would round the number of measurements per window to 1. This equation also implies that all sensors generate a constant number of data samples in each window, which may not be correct in general. We are introducing these requirements to ensure that in the window there is at least one data sample from each source, and that the number of samples within a window is constant. The reason for that is because they significantly simplify the automatic feature engineering and also guarantee that all sensors are considered in each window. On the other hand, it violates (1). Therefore, we are relaxing this requirement, and instead we redefine it with relational algebra. In order to retrieve the n_i measurements that belong in window k, which ends at time te_k, we use the relational algebra expression (4). This expression selects all data points from the i-th source in the k-th window for which the timestamp t is less than or equal to the end time of the k-th window te_k, and retrieves only the n_i most recent data points.

$$dp_i^k = \sigma_{\text{rownum()}\leq n_i}(\tau_{t \text{ desc}}(\sigma_{s=i \text{ and } t\leq te_k}(R))) \tag{4}$$

Usually the overlapping (o) is greater than the minimum time between readings μ, but is in the same order of magnitude. The size of windows (w) is generally about one order of magnitude larger than μ. Again, this is just a rule of thumb based on approaches like [19], [20], [21], [14].

Finally, for a given window size w and overlapping o, and total time period during which data was collected T, and the number of training instances N can be calculated with (5).

$$N = \left\lfloor \frac{T-(w-o)}{w-o} \right\rfloor \tag{5}$$

B. Sliding window length

The window length (w) and the size of overlapping segments (o) need to be defined before applying the segmentation. According to the survey presented in [22], the window size depends on the target class, i.e., the activity which should be recognized. For example, activities like “walking” have smaller windows than “cooking”. Lower sensor frequencies

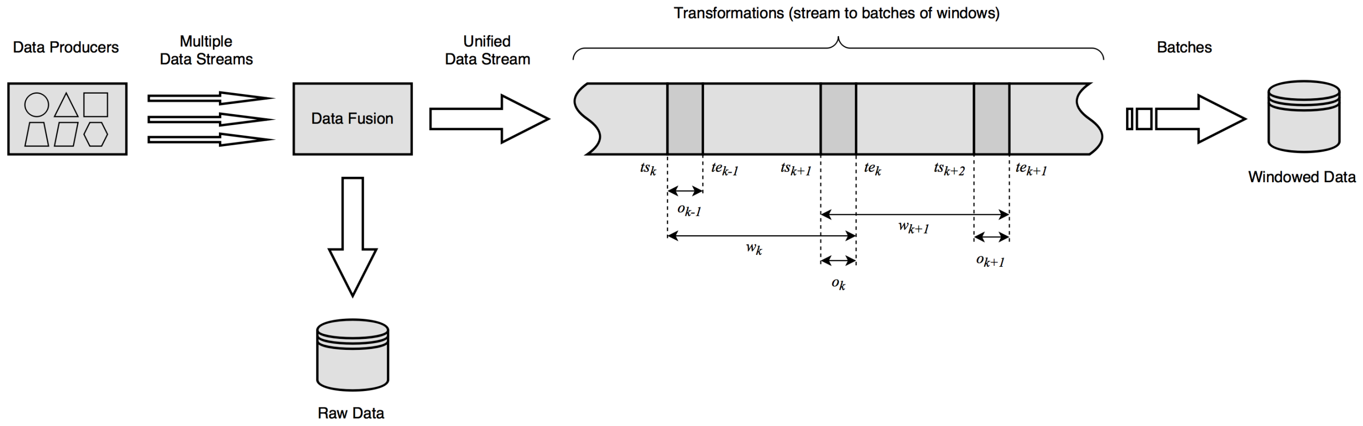


Fig. 1: Segmenting data streams with sliding windows

or more complex activities entail longer windows [18]. We acknowledge that when extracting features for recognition of predefined activities, having varying window length for each activity could be beneficial. To facilitate this, at each moment a segmentation in different windows is required. After the feature extraction from the data in each different window, the system would make several predictions, one for each window, which eventually need to be aggregated to make a final prediction of the activity at particular moment. Such flow becomes much more complicated, thus we are not evaluating it in this study. Instead, we argue that a systematic feature engineering can lead to robust features that facilitate high classification performance. We reason that the windows length of the longest activity can be appropriate, if the generated features are robust and informative.

C. Classification algorithms

In this subsection, we describe the algorithms for building classification models, which were provided by the *scikit-learn* library [23].

We use *logistic regression* (LR) [24] because it is a simple, fast and portable algorithm. Furthermore, LR provides an easy interpretation of models and the importance of features, and it is easily parallelizable. It is essential part of the feature selection methods, which are described later in the manuscript.

Random Forest (RF) [25] and *Extremely Randomized Trees* (ERT) [26] are very efficient ensemble algorithms based on decision trees [27] and provide feature importance estimates, which is useful for feature ranking. Both of these algorithms provide similar estimates for the feature importance, and we have chosen ERT for feature ranking due to its better speed. RF generates multiple decision trees by randomly sampling training instances from the dataset and also randomly selecting m features from each sample, where M is the total number of features per instance and $m < M$. The tree branching is performed by finding the best split from the m features on each node. In the process of classification, each tree votes for the class and the majority class is chosen. On the other hand, ERT chooses the split points in the features randomly. This increases the training speed because the number of calculations per node is decreased. Both algorithms provide excellent classification performance and can train models on very large datasets very fast. The value of the m parameter used in RF and ERT was the default one per their implementation in the scikit-learn library [23]. We did not notice any significant gain by tuning this parameter. During the evaluation, the RF and ERT classification models were built with 100 or 1000 trees.

Additionally, we have also used the *Support Vector Machine* (SVM) classifier [28] with Gaussian kernel. It is a very powerful classifier, but for larger datasets it requires a lot of time for building models, a problem which is exacerbated when parameter tuning is performed. Be that as it may, parameter tuning of SVMs is recommended because it significantly increases their classification accuracy and reduces overfitting [29]. When using an SVM, the datasets were normalized so that each feature will have mean 0 and standard deviation of 1 for the dataset, which can also improve their performance [43].

Other classification algorithms used in similar studies [11], [30], [31] are k-Nearest Neighbors (kNN) [32] and Naive Bayes (NB) [33]. We have used k-Nearest Neighbors with k=5 in all the experiments. We have initially tested using k=3 or k=5, but the latter was always providing better results, therefore we concluded to always use k=5 and not to tune this parameter. When using a kNN, the datasets were also normalized so that each feature will have mean 0 and standard deviation of 1 for the dataset. For the NB classifier we have used Gaussian and Bernoulli distribution functions for the experiments with the validation test set. Only the one that resulted in better accuracy for the validation dataset was used for the final test set.

D. Feature Extraction and Selection

Many approaches perform frequency filtering to remove noise in raw readings or to separate the acceleration into gravitational and body components [3], [15], [16], [34]. For similar reasons, some approaches use moving average smoothing of raw data prior feature extraction [35], [36]. Because sensor orientation is important, these filtering techniques are valuable [3]. However, the feature extraction framework we propose is more general and not specialized for particular sensor type, therefore it is not applying such filtering techniques. We assert

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2684913, IEEE Access

that without specialized filtering, the feature selection process will still identify informative features that will result in robust classification models.

In cases when the datasets had a nominal attribute (e.g. gender, handedness, etc.), several features were generated from them: one or more dummy numeric features (e.g. a flag for each gender) and more numeric features depending on the number of activities, generated with the Weight of Evidence (WOE) technique [37]. As it turned out after performing the experiments with all datasets, all features originating from nominal attributes were discarded.

Due to the recorded data diversity, the key tasks to be performed are feature extraction and identification of the best feature set for automatic activity recognition. A challenge for building robust features is dealing with drift in the data as a result of either data generated by different sensors, data collected from different participants, or loss of accuracy of sensors over time. Ideally, such variations should have little to no effect on the trained models.

Frequently, depending on the problem domain, the types of extracted features have been previously successfully applied to the same or a similar domain. This often is subjective and depends on the researcher's experience. To alleviate this, we propose systematically engineering new time series derived from the original time series from all sensors, aiming to extract variety of informative and robust features suitable for identification of different activities.

The process of feature extraction is shown in Fig. 2 and consists of several steps involving the originally recorded time series and the newly generated time series. First, in step 1, the stream of data is preprocessed with sliding windows generating batches of windows, as described in the previous subsection (for details per evaluated dataset see Table I).

Next, in step 2, our framework generates *magnitudes* time series from multi-axial sensors (e.g. gyroscopes and accelerometers) or multi-channel units (e.g. electro cardiogram - ECG).

From all original and magnitudes time series, step 3 extracts the following types of features, which have been proven to be effective predictors in recent competitions [21], [38] related to feature extraction from diverse time series data. The number of measurements within one window is denoted by n.

- *Basic statistics*: minimum, maximum, range, arithmetic mean, harmonic mean, geometric mean, mode, standard deviation, variance, skewness, kurtosis, signal-to-noise ratio, energy, and energy per sample, which results in 14 features per time series.
- *Equal-width histogram* calculated with $\lceil log_2 n + 1\rceil$ intervals, based on the Sturges rule [39].
- *Quantile based features*: first quartile, median, third quartile, inter-quartile range and some other percentiles (5, 10, 20, 30, 40, 60, 70, 80, 90, 95), used also in [36], [40]. From one time series, it generates 14 features.
- *Auto-correlation* of the measurements within one sliding window [41], [42]. Let τ denote the amount of shift in the domain defined as: $\tau \in [1, \lfloor\sqrt{n}\rfloor]$. For exponentially increasing values of τ in that range, classical auto-correlation and Pearson correlation are calculated. Additionally, both correlations using the first and second half of measurements within one sliding window are calculated.
- *Pearson correlations* between pairs of time series, used also in [41], [42].
- *Linear and quadratic fit coefficients*. There are 2 linear fit and 3 quadratic fit coefficients, yielding 5 features in total per time series [42].

Then, from the original time series and their magnitudes, the system generates new time series: *first derivatives*; *delta series*, which calculate the relative deviations from the mean value of original readings within one window [43]; and series derived from Fast Fourier Transformation (FFT) (henceforth referred to as *FFT-derived* time series): frequencies, amplitudes and magnitudes [3], [20], [36].

Step 4 generates *delta series*, which calculate the relative deviations from the mean value of original readings within one window [43]. First it calculates the mean value $\overline{x}_i$ of the measurements x_i^j within a sliding window of the i-th time series ($0 \leq j < n$, where n is the number of measurements within one sliding window). Then it calculates the differences $\Delta x_i^j = \overline{x}_i - x_i^j$ between the original measurement x_i^j and the calculated mean $\overline{x}_i$. Note that the mean is calculated separately for each sliding window and each time series. Thus, each original measurement x_i^j is mapped to a new value Δx_i^j. As a result, this step generates as many delta series as there are original and magnitude time series.

Subsequently, from the delta series, step 5 extracts only histogram based and quantile based features in the same way as from the original time series.

Auto-correlations features are omitted because they are redundant to the auto-correlation features extracted from the original time series. The redundancy is a direct consequence of the definition of classical auto-correlation and Pearson correlation. The curve-fitting features are also not extracted because the delta series is only a linear translation of the original series, thus these features are also redundant. For the same reason, most basic statistics are also redundant, therefore they are not computed as well. This step generates 21 features from each delta time series.

Comparable to step 4, step 6 generates *first derivatives* time series from the original and magnitude time series [43]. Given that the measurements are recorded at constant frequency, the first derivative is defined as difference between consecutive measurements within one time window. Thus, from a time series with n measurements within one sliding window segment, the first derivative time series has $n - 1$ values.

Step 7 extracts features from the first derivative time series in an analogous manner as from the original time series. The only difference is that it does not compute auto-correlation features for the reasons described in step 5.

Next, step 8 calculates Fast Fourier Transformation (FFT) of the original and magnitude time series, thus generating 3 FFT-derived time series from each of them: i.e. the series of frequencies, amplitudes and magnitudes [3], [36].

From each of the FFT-derived time series, step 9 calculates the following features: minimum, maximum, mean, standard deviation, range, first and third quartile, inter-quartile range,

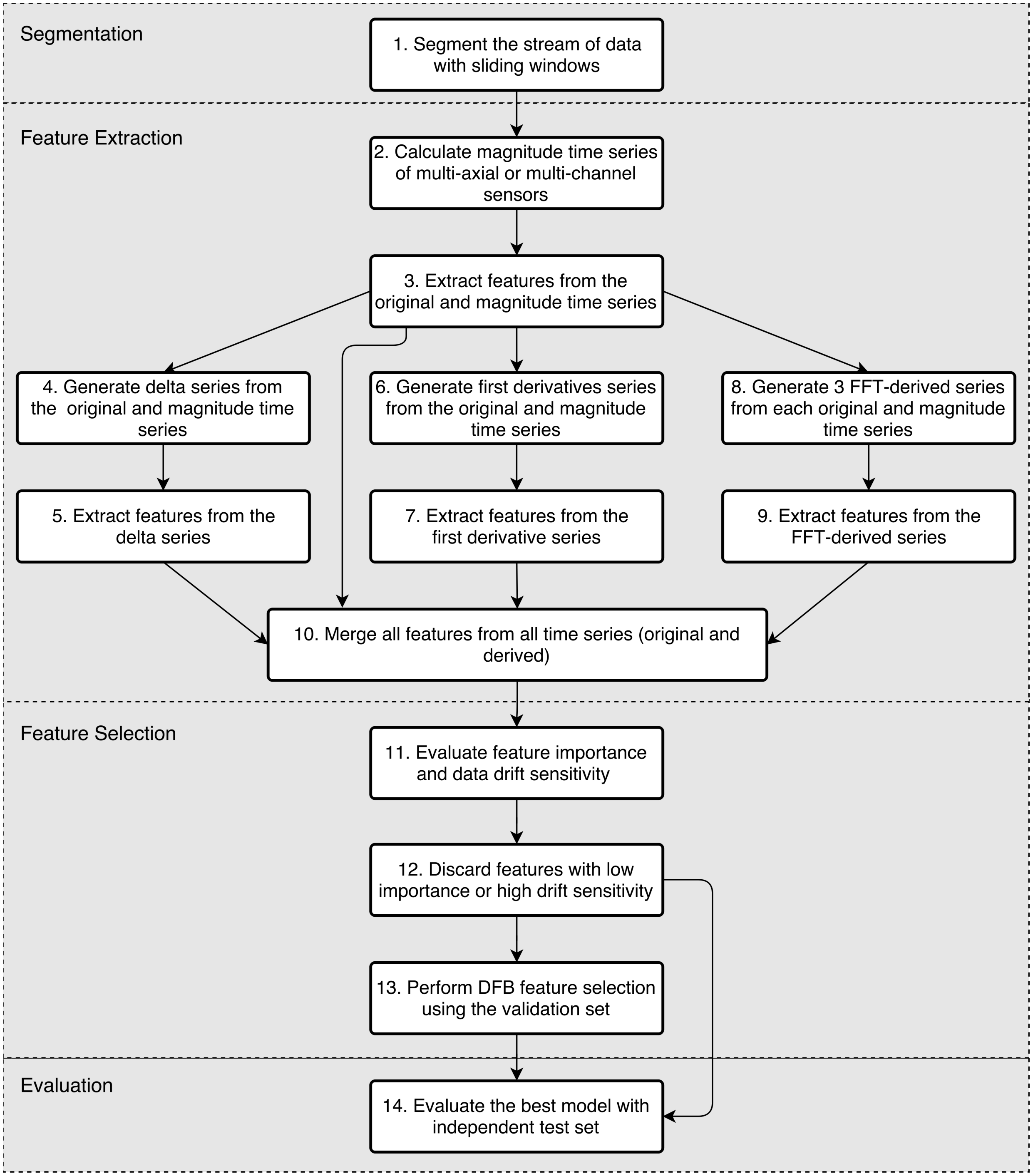

Segmentation
1. Segment the stream of data with sliding windows
Feature Extraction
2. Calculate magnitude time series of multi-axial or multi-channel sensors
3. Extract features from the original and magnitude time series
4. Generate delta series from the original and magnitude time series
6. Generate first derivatives series from the original and magnitude time series
8. Generate 3 FFT-derived series from each original and magnitude time series
5. Extract features from the delta series
7. Extract features from the first derivative series
9. Extract features from the FFT-derived series
10. Merge all features from all time series (original and derived)
Feature Selection
11. Evaluate feature importance and data drift sensitivity
12. Discard features with low importance or high drift sensitivity
13. Perform DFB feature selection using the validation set
Evaluation
14. Evaluate the best model with independent test set

Fig. 2: Algorithm for feature extraction, selection and classification

median and the 10-th, 40-th, 60-th and 90-th percentile. The rationale is that these 13 features would sufficiently describe the distribution of values of the FFT-derived time series. The spectral centroid is another feature generated by FFT from each original and magnitude time series. To summarize, at the end of step 9 there are $3 \times 13 + 1 = 40$ frequency domain features from each original or magnitude time series.

Step 10 merges all features generated by steps 2, 5, 7 and 9. As a result, it unifies them in one large feature set which may potentially contain redundant or non-informative features.

Next is step 11, which calculates the *feature importance* in the merged feature set by training an ERT classifier. All feature importance estimations are performed by training an ERT classifier with 1000 trees and using its feature importance estimates. Additionally, with the method proposed in [44] it estimates the concept distribution drift of the features. Concept drift denotes a change in the probability distribution of a feature. For instance, the feature could be normally distributed in all datasets, but could have considerably different mean and standard deviation in the validation and test sets than

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2684913, IEEE Access

the training sets. Moreover, a feature could have a completely different probability distribution functions in the validation or test subsets, such as skewed normal, log-normal, multi-modal, uniform, etc. These changes in the data distribution often result in model overfitting to the training dataset [44].

To mitigate this, the proposed framework attempts to identify drift sensitive features by generating an artificial dataset containing all rows of the training and validation datasets. Likewise, an artificial target label (i.e. class) is generated, which denotes from which dataset the corresponding row originates. On this artificial dataset an ERT classifier is trained and the importance of each feature evaluated. The latter feature importance in fact defines the *data drift sensitivity* estimate of the feature. To clarify, the very informative features in the artificial dataset are actually very sensitive to data distribution drift and therefore lead to model overfitting [44]. To summarize, at the end of step 11 for each feature, the actual feature importance and the data drift sensitivity estimate are available.

Then step 12 performs coarse-grained feature selection. From the feature importances and data drift sensitivity of all features, the following 9 percentiles are calculated: 0, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100. Note that there are 11 values calculated as percentiles of the feature importances and additional 11 values calculated as percentiles of data drift sensitivity estimates. The algorithm formalized in [45] uses these 2 sets of values as feature importance thresholds and data drift sensitivity thresholds. It evaluates all $11 \times 11 = 121$ combinations of thresholds for discarding features which have low feature importance (i.e. importance < threshold) or high data drift sensitivity (i.e. drift sensitivity > threshold). For each of the 121 feature sets it builds classification models using the training dataset and evaluates them with the validation dataset. The test set is not utilized at this stage at all. Thus, only the feature set that results in best classification accuracy is retained. To summarize, the purpose of this step is to significantly reduce the feature set size by discarding features with low importance or high data drift sensitivity. The applied wrapper classification algorithm for this step is ERT with 100 trees because of its speed and generally good classification performance. RF provides similar performance, albeit somewhat more slowly, depending on the dataset and feature set. The best feature set obtained after this step is hereafter referred to as *score-drift screened* feature set.

Step 13 performs *diversified forward-backward (DFB) feature selection* using a modified version of the algorithm described in [46]. It is a wrapper feature selection method, which trains a model with the training set and evaluates it with the validation set, aiming to evaluate the quality of a feature set. The major difference of the version in our framework is that it is parallelized and uses a greedy heuristic to narrow down the search space. It uses logistic regression or Naive Bayes as a wrapper algorithm. As a first step, our approach ranks by importance the retained features after step 12 (i.e. the best score-drift screened feature set) and only these features are considered during step 13. Next, in one iteration of the forward pass, starting from an empty feature set, features are added to the current best feature set, and multiple feature sets are evaluated in parallel. Only the features whose addition improved the predictive accuracy are retained. The heuristic is that features which did not improve the score when added to some feature set, will not be considered for addition to other feature sets tested later. The forward pass ends if after all eligible features for addition were considered during one forward pass, there is no improvement of the best accuracy. Next, the backward phase follows, which tests if the removal of any feature from the best feature set improves the score. In case a removal of a feature improved the accuracy, it starts a new backward iteration. Otherwise, when all features were tested for removal, the backward iteration ends. In case of an improvement during the forward or the backward phase, the algorithm starts a new cycle of forward and backward passes. Otherwise, the search converges and stops. The algorithm also terminates if it evaluated the maximum number of allowed feature sets (2000 in this study). The feature score-drift screening performed in step 12 is particularly important because without it, the search space for step 13 would be significantly more complex. The feature set obtained after step 13 is henceforth referred to as DFB feature set.

We acknowledge that the selected best feature set may be biased towards the wrapper classification algorithm (i.e. LR or Naive Bayes) used in step 13. Nevertheless, owing to the simplicity and speed of the wrapper algorithms, the system evaluates multiple feature sets in parallel, which is very efficient time-wise. Namely, the algorithm converges or evaluates a maximum of 2000 feature sets in less than 5 minutes when executing 12 threads in parallel. At the end of this step, the feature set that resulted in the highest accuracy is marked as the "best" feature set.

Ultimately, once the best feature set is determined, the evaluation of the automated method for identification of ADLs is performed in two phases. The first phase refers to step 14 in Fig. 2. This step initially determines the optimal values for the SVM parameters C (i.e. cost) and γ. It achieves this by testing various combinations of values for the SVM parameters. More precisely, it tests one default configuration when C is set to one, while the gamma parameter to one divided by the number of features. Additionally, it performs grid search [29], testing various combinations, where C varies in the range 0.1, 1, 10 and 100, and γ varies in the range 0.1, 0.01, 0.001, 0.0001 and 0.00001. The training and validation datasets using the "best" feature set are employed for the grid search evaluation. Finally, in the second phase, all classification algorithms (Logistic regression, Naive Bayes, kNN, ERT, RF and SVM with optimal parameters) are evaluated using the independent test dataset.

E. Evaluation of sensor usefulness

Different studies use different sensors placed at various body locations for activity recognition. Usually the choice of sensors and body placement is based on findings from studies such as [47], [48]. They recommend using sensors closely attached to the body's center of gravity, such as on the chest, trunk or hip. Different studies [3], [12], [14], [19], [20], [49], [50] use sensors placed at the hip. Activities primarily performed by the upper body are successfully classified by sensors on the wrist [4], [13], [14], [20], [49], [50], [51], [52]. In some studies [4], [14], [15], [16], aiming to resemble application of a smart phone for activity detection, a sensor on the waist or in the pocket has been used. Gait analysis [53] and activity recognition [19], [20] also heavily rely on sensors on the ankle. Identification of which sensors and which body positions lead to best performance for detecting activities of interest can lower the intrusiveness to subjects in AAL and general health applications, while reducing the cost. Therefore, the proposed system analyzes all combinations of sensors and body locations in each evaluated dataset in terms of overall accuracy and precision per activity. This, in turn, can lead to identifying key sensors or body locations for identification of particular activities. Therefore, for n sensors placed at a particular body location, Eq. (6) calculates the number of all sensor combinations:

$$\sum_{1 \leq k \leq n} \binom{n}{k} = 2^n - 1 \tag{6}$$

For each sensor combination, the proposed system only takes into consideration the features derived from those sensors and discards all features deriving from other sensors. Then, it performs steps 12 and 13 in Fig. 2, which determine the best feature set from the considered sensors, and finally evaluates them with step 14.

For the SBHAR dataset there was only one sensor location, therefore instead of investigating the impact on sensor locations combinations, we have analyzed the contribution of sensor types. Gyroscope and accelerometer measurements are provided in this dataset. It also includes the body acceleration, which is separated out of the total acceleration with frequency filtering.

F. Datasets

In this study, we use the segmentation strategy that the authors of the original study and dataset used. The segmentation into windows was performed only within a single activity, thus excluding the border intervals when the activity changes from one activity to another.

The subjects in each dataset are divided into three distinct subsets for: training, validation and testing. If the authors of a dataset divided the subjects in training and tests sets already, then we only divide the original training set into train and validation subsets, while keeping the test set completely independent. The training dataset is used for model building, the validation for evaluating the models during feature selection and tuning parameters of classifiers. After the best feature sets and classifier parameters are determined, the union of the training and validation sets are used for building final models and their performance is evaluated on the test (i.e. holdout) set. In addition to this strategy, other studies, such as [3], [19], [20], [50], use the leave-one-subject-out strategy or the 10-fold cross validation strategy [12], [14]. From these strategies, the independent test set (i.e. holdout) is the most pessimistic [54], [55]. For that reason and because the wrapper feature selection method would be much more complicated with a leave-one-subject-out strategy, in this study we use an independent test set for evaluation of the models.

1) DaLiAc dataset: This study described in [3] proposes a hierarchical, multi-sensor based classification of Activities of Daily Living (ADL) that reached an overall mean classification rate of 89.6% and provided a benchmark dataset, hereafter referred to as the DaLiAc (Daily Living Activities) dataset. Nineteen healthy subjects performed thirteen activities. Four Shimmer sensor nodes [56] were placed on right hip, chest, right wrist and left ankle. Each sensor node consisted of three accelerometer axes and three gyroscope axes. The range of the accelerometer was $\pm 6g$. The range of the gyroscopes was $\pm 500 deg/s$ for the sensor nodes wrist, chest, hip and $\pm 2000 deg/s$ for the sensor node on the ankle. The sampling rate was set to 204.8Hz. The window length for segmentation was 5 seconds (i.e. 1024 measurements from each sensor) with 50% overlapping between adjacent windows. Because the original study uses leave-one-subject-out evaluation procedure, we had to randomly divide the subjects into three subsets. Thus, the information about the three subsets is shown in Table I. The list of activities which were analyzed in this dataset have non-blank precision and recall values in Table IV.

2) mHealth dataset: In [12] is described the design, implementation and validation of an open framework for agile development of mobile health applications. As a result, the mHealth dataset was created. It comprises of body motion and vital signs recordings, for ten volunteers of diverse profile, while performing twelve physical activities. Shimmer2 wearable sensors [56] were used for the recordings. The sensors were placed on the subject's chest, right wrist and left ankle. The use of multiple sensors permitted measuring motion experienced by diverse body parts, namely, the acceleration, the rate of turn and the magnetic field orientation, thus better capturing the body dynamics. The sensor positioned on the chest also provided 2-lead ECG measurements, which were not used in the original study [12], however they are analyzed in our study. The sampling rate used for all sensing activities was 50 Hz, and the window length and overlapping were 4s, respectively. The reason why we performed oversampling by using a larger overlap than the common 50% (2s in this case), was to increase the number of segments, considering the low number of subjects in this dataset. The summary of this dataset along with the others is presented in Table I. The twelve activities performed in the mHealth dataset have valid precision and recall values in Table IV.

3) FSP dataset: The study presented in [14] explains an experiment where ten participants performed seven different activities carrying smart phones at five different body positions: right jeans pocket, left jeans pocket, belt position towards the right leg, right upper arm and right wrist. The first three positions are regularly used by people carrying smartphones, the fourth position is commonly used when jogging and the fifth position simulates a smart-watch. The activities (listed in Table IV) are mainly used in the related studies and they are the basic motion activities in daily life. The experiments were carried out indoors in one of the university buildings, except biking. For walking, and jogging,

TABLE I: Information about the datasets used in this study. Abbreviations: Instances is the number of sliding window segments; Train, Valid. and Test are the training, validation and test subsets, respectively; Freq. is the Sampling frequency in Hz; Wind. is the window length in seconds; Overlap is the overlapping between consecutive windows in seconds; Orig. is the number of original time series; Magn. is the number of magnitude time series; Delta is the number of delta series; FD is the number of first derivative time series; FFT stands for the number of FFT-derived time series.

	Instances			Subjects						Number of Time Series				
Dataset	Train	Valid.	Test	Train	Valid.	Test	Freq.	Wind.	Overlap	Orig.	Magn.	Delta	FD	FFT
DaLiAc [1]	3356	2299	3150	7	5	7	204.8	5	2.5	24	8	32	32	96
mHealth [2]	2662	1241	2549	4	2	4	50	4	3	23	8	31	31	91
FSP [2]	5012	2506	5012	4	2	4	50	2	1	60	20	80	80	240
HAR [3]	5132	2220	2947	14	7	9	50	2.56	1.28	6	2	8	8	24
SBHARPT [3]	5468	2374	3192	14	7	9	50	2.56	1.28	6	2	8	8	24

the department corridor was used. For walking upstairs and downstairs, a 5-floor building with stairs was used. This study shows that multi-sensor combination improves the recognition performance in some cases. Henceforth, we refer to this dataset as FSP (Five Smart Phones) in this study and more information about it is provided in Table I.

4) SBHAR dataset: The Smartphone-based Human Activity Recognition (SBHAR) dataset was created as part of the study described in [57], [15]. Recordings of thirty subjects with age from 19 to 48 years doing six ADL while carrying a waist-mounted smartphone (Samsung Galaxy S II) with embedded inertial sensors comprise this dataset. Using the phone's embedded accelerometer and gyroscope, they captured 3-axial body and total acceleration and 3-axial angular velocity at a constant rate of 50Hz. The experiments have been video-recorded to label the data manually. The authors randomly partitioned the subjects into two sets, thus obtaining two sets of 21 and 9 subjects. We further divided the set of 21 subjects into two subsets, which we refer to as training and validation datasets, while keeping the set of 9 subjects as test dataset intact, thus obtaining three subsets in total, as summarized with Table I. The six activities analyzed in this dataset have non-blank precision and recall values in Table IV.

5) SBHARPT dataset: The same authors of the SBHAR dataset described an extended version of the same dataset to include postural transitions in [16], denoted by SBHARPT. This dataset consists of twelve human activities and postural transitions in total, and more information about it is presented in Table I. The evaluated activities in the dataset are shown in Table IV. The only difference is that for the SBHARPT dataset they did not provide the body acceleration as in the SBHAR dataset, rather they provided just the total acceleration and the gyroscope measurements. The six activities and transitions between them analyzed in this dataset have non-blank precision and recall values in Table IV.

TABLE II: Number of features and best accuracy per dataset and feature selection method. Feat. is number of features and Acc. is Accuracy using the best performance obtained on a particular feature set (FS). No feature selection (No FS) columns refer to the number of features and accuracy before the feature selection starts (at the end of step 10 in Fig. 2). Score-Drift FS stands for the feature set that resulted in best accuracy after the feature screening (Step 12 in Fig. 2). DFB FS stands for the feature set obtained by DFB feature selection (Step 13 in Fig. 2) that resulted in best accuracy.

	No FS		Score-drift FS		DFB FS	
Dataset	Feat.	Acc.	Feat.	Acc.	Feat.	Acc.
DaLiAc	4871	0.916	1083	0.934	60	0.912
mHealth	3232	0.972	620	0.997	39	0.998
FSP	11418	0.996	2409	0.998	44	0.997
SBHAR	1415	0.922	316	0.959	47	0.945
SBHARPT	1236	0.925	475	0.958	60	0.941

III. Results

The number of original and generated time series for each dataset is shown in Table I. Next, the total number of generated features, as well as the final number after the score-drift feature screening and DFB feature selection was executed (see Steps 12 and 13 in Fig. 2), is shown in Table II. For all datasets, the score-drift feature screening method significantly reduced the number of features, while also improving the accuracy. Furthermore, the DFB feature selection additionally discarded most of the features, thus reducing them to less than 60 for all datasets. This drastic reduction of features, significantly lowered the classification models building time while having little or no negative effect on the accuracy.

The accuracy per classification algorithm and best feature set per sensor location combination on the DaLiAc dataset is shown in Fig. 3. It is evident that the SVM, RF and ERT had consistently better accuracy for all combinations of sensors than the other classifiers, while LR had worst accuracy.

On the mHealth dataset, the highest accuracy was very high at 0.998, as shown in Fig. 4. It was evident that the ECG sensor did not contribute to the activity recognition at all. The accuracy without the ECG sensor was better or at the same as when it was additionally used in all combinations of other sensors. Likewise, it individually had poor performance for activity recognition (0.337), which is by far the worst when compared to the other sensor locations used individually (over

[1] DaLiAc dataset subjects. Train: 1,2,3,4,5,6,7; Validation: 8,9,10,13,16; Test: 11,12,14,15,17,18,19.

[2] mHealth and FSM dataset subjects. Train: 1,2,3,4; Validation: 5,6; Test: 7,8,9,10.

[3] SBHAR and SBHARPT subjects. Train: 14, 15, 16, 17, 19, 21, 22, 23, 25, 26, 27, 28, 29, 30; Validation: 1, 3, 5, 6, 7, 8, 11; Test: 2, 4, 9, 10, 12, 13, 18, 20, 24.

TABLE III: Accuracy per combination of sensor locations and classifier for the FSP dataset. Num. Loc. stands for number of locations where sensors are placed; Sensor Loc. Comb. stands for Sensor location combination; PL stands for left jeans pocket and PR for right jeans pocket. The accuracies with gray background are the best for a particular sensor location. The accuracy in bold is highest in general. The sensor location combination with gray background has highest accuracy from all sensor location combinations with same number of sensor locations. The sensor location in bold resulted in highest accuracy in general.

Num. Loc.	Sensor Loc. Comb.	kNN	LR	NB	RF	ERT	SVM
5	Arm+Belt+PL+PR+Wrist	0.9942	0.9848	0.9918	0.9960	0.9952	0.9980
4	Arm+Belt+PL+PR	0.9932	0.9789	0.9912	0.9948	0.9942	0.9966
	Arm+Belt+PR+Wrist	0.9940	0.9122	0.9709	0.9934	0.9950	0.9970
	Arm+Belt+PL+Wrist	0.9936	0.9818	0.9800	0.9958	0.9948	0.9978
	Arm+PL+PR+Wrist	0.9882	0.9284	0.9932	0.9954	0.9940	**0.9984**
	Belt+PL+PR+Wrist	0.9956	0.9595	0.9920	0.9962	0.9962	0.9980
3	Arm+Belt+PL	0.9908	0.8434	0.9796	0.9948	0.9942	0.9954
	Arm+Belt+PR	0.9888	0.9372	0.9619	0.9932	0.9930	0.9960
	Arm+Belt+Wrist	0.9709	0.7861	0.9044	0.9581	0.9499	0.9948
	Arm+PL+PR	0.9938	0.9495	0.9936	0.9942	0.9934	0.9966
	Arm+PL+Wrist	0.9890	0.9449	0.9860	0.9936	0.9914	0.9974
	Arm+PR+Wrist	0.9585	0.8817	0.9826	0.9918	0.9870	0.9970
	Belt+PL+PR	0.9900	0.9336	0.9880	0.9936	0.9932	0.9960
	Belt+PL+Wrist	0.9932	0.8601	0.9864	0.9960	0.9950	0.9972
	Belt+PR+Wrist	0.9950	0.9102	0.9792	0.9866	0.9868	0.9966
	PL+PR+Wrist	0.9765	0.8625	0.9936	0.9958	0.9946	0.9954
2	Arm+Belt	0.9409	0.8232	0.8773	0.9481	0.9529	0.9858
	Arm+PL	0.9910	0.9751	0.9808	0.9832	0.9876	0.9946
	Arm+PR	0.9884	0.9629	0.9838	0.9918	0.9902	0.9962
	Arm+Wrist	0.9579	0.8988	0.9280	0.9379	0.9320	0.9665
	Belt+PL	0.9735	0.8394	0.9344	0.9922	0.9902	0.9928
	Belt+PR	0.9565	0.7877	0.9260	0.9850	0.9852	0.9697
	Belt+Wrist	0.9469	0.8378	0.8508	0.9421	0.9547	0.9910
	PL+PR	0.9900	0.9723	0.9838	0.9888	0.9878	0.9898
	PL+Wrist	0.9922	0.8917	0.9862	0.9940	0.9926	0.9974
	PR+Wrist	0.9862	0.9515	0.9828	0.9944	0.9940	0.9970
1	Arm	0.9401	0.9348	0.9018	0.9372	0.9417	0.9475
	Belt	0.8889	0.8057	0.7985	0.8382	0.9054	0.9242
	PocketLeft (PL)	0.9804	0.9779	0.9372	0.9671	0.9635	0.9938
	PocketRright (PR)	0.9850	0.9725	0.9810	0.9916	0.9928	0.9938
	Wrist	0.9603	0.9372	0.9218	0.9284	0.9276	0.9635

TABLE IV: Precision and recall per dataset and activity with the classifier that resulted in best test accuracy.

	DaLiAc		mHealth		FSP		SBHAR		SBHARPT	
Activity	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.	Prec.	Rec.
Walking	0.906	0.987	1	1	0.994	0.994	0.974	0.980	0.970	0.990
Standing	0.836	0.670	1	0.983	1	1	0.903	0.932	0.998	0.800
Sitting	0.979	0.887	1	1	1	1	0.924	0.892	0.750	0.714
Walking upstairs	0.991	0.979	0.983	1	0.997	0.997	0.979	0.979	0.690	0.556
Walking downstairs	0.912	0.918	-	-	0.997	1	0.978	0.971	0.833	0.968
Lying	1	1	1	1	-	-	1	1	0.979	0.993
Running	0.943	0.953	1	1	-	-	-	-	-	-
Jogging	-	-	1	1	1	0.997	-	-	-	-
Biking	-	-	1	1	1	1	-	-	-	-
Bicycling EM (50W)	0.992	1	-	-	-	-	-	-	-	-
Bicycling EM (100W)	1	0.991	-	-	-	-	-	-	-	-
Washing dishes	0.727	0.870	-	-	-	-	-	-	-	-
Vacuuming	1	1	-	-	-	-	-	-	-	-
Sweeping	1	1	-	-	-	-	-	-	-	-
Rope jumping	1	0.969	-	-	-	-	-	-	-	-
Waist bends forward	-	-	1	1	-	-	-	-	-	-
Frontal elevation of arms	-	-	1	1	-	-	-	-	-	-
Knees bending (crouching)	-	-	1	1	-	-	-	-	-	-
Jump front & back	-	-	1	1	-	-	-	-	-	-
Stand to Sit	-	-	-	-	-	-	-	-	0.943	0.909
Sit to Stand	-	-	-	-	-	-	-	-	0.922	0.955
Sit to Lie	-	-	-	-	-	-	-	-	1	1
Lie to Sit	-	-	-	-	-	-	-	-	0.870	0.870
Stand to Lie	-	-	-	-	-	-	-	-	1	1
Lie to Stand	-	-	-	-	-	-	-	-	0.703	0.813

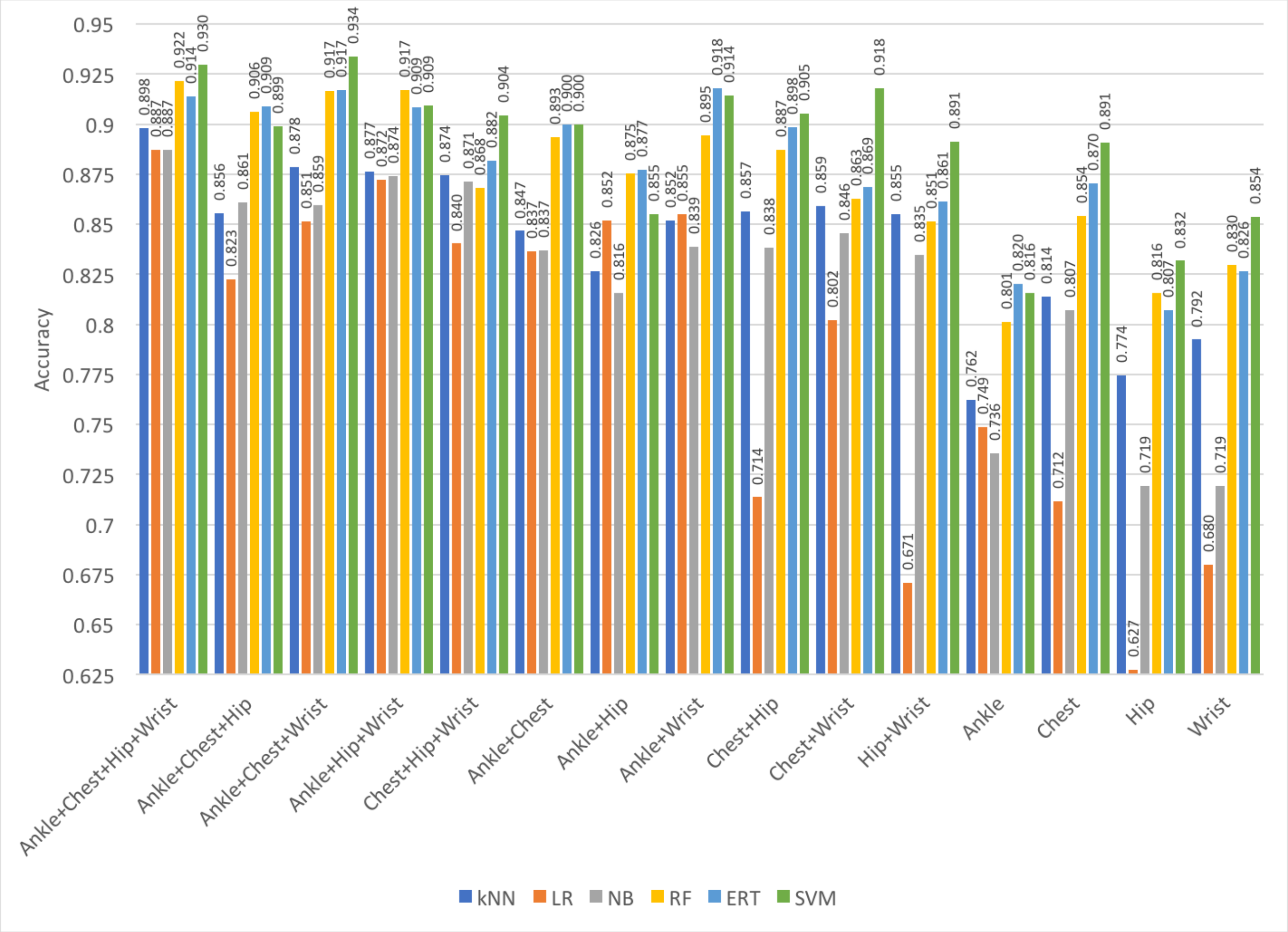


Fig. 3: Accuracy per classification algorithm and best feature set per sensor location combination on the DaLiAc dataset

0.74). To simplify the charts, all combinations that included the ECG sensor were omitted from Fig. 4. The best classifiers for all combinations of sensor locations were a kNN with 5 neighbors and an SVM.

The results of the experiments with the FSP dataset with different sensor locations and various classifiers are shown in Table III. The highest accuracy was very high (0.998). It is also evident that the redundancy of sensors on different body locations did not significantly improve the classification performance.

In Fig. 5 and Fig. 6 are shown the results of the experiments with the SBHAR and SBHARPT datasets, respectively. We can note that the same accuracy of about 0.958 was achieved for both datasets, and that the Naive Bayes and kNN classifier have inconsistent accuracy for all sensor type combinations, unlike the other classifiers, which perform consistently better.

In Fig. 7 the maximum accuracy depending on the sensor locations for the different datasets is shown. The purpose of this figure is to simplify comparison of the different combinations of sensor locations in different datasets and to point out whether some of them are consistently better than others. From the two sensor locations, the ankle plus wrist combination results in best accuracy for both the DaLiAc and mHealth datasets.

Next, Table IV shows the precision and recall for each activity and each dataset on the independent test sets. The activities which were not analyzed in some study, have blank values for the precision and recall columns. Both the precision and accuracy for most activities and dataset were very high, often even 100%. However, for some of the more common activities, such as Sitting and Walking Upstairs or Downstairs, it is notable that the precision of the SBHARPT dataset was significantly lower, even when compared to the SBHAR dataset, although it was based on the same sensors.

Fig. 8 shows the differences of the feature selection methods in terms of accuracy per classification algorithm and dataset. From the DFB feature selection the logistic regression algorithm benefited the most almost on all datasets, while all other classification algorithms had either similar or slightly worse accuracy than with the best score-drift screening feature set. However, in terms of time for building models and making predictions, the benefit of the DFB feature selection is clearly visible in Fig. 9. The duration includes the time needed for model building with the union of the training and validation sets and making predictions on the independent test set. The duration of all classification algorithms on all datasets has

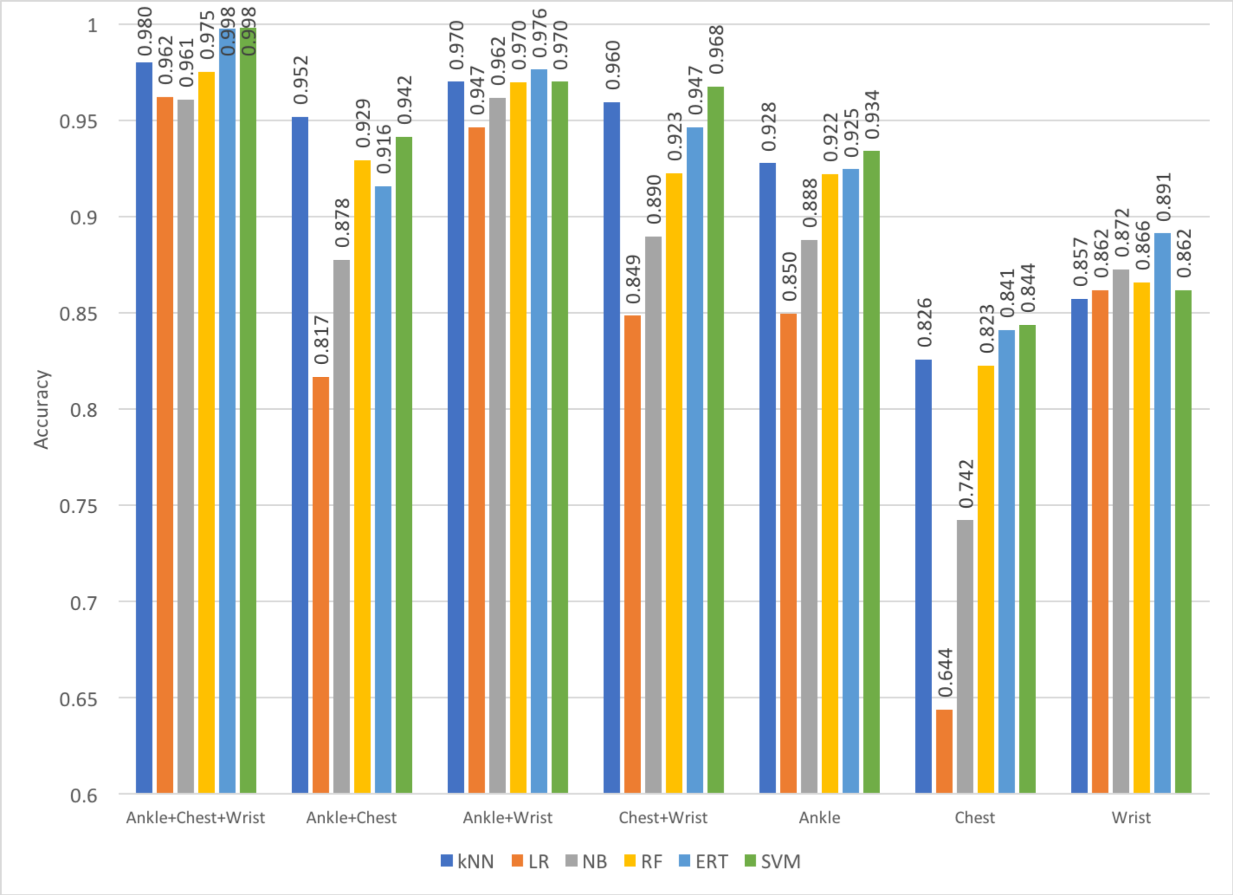


Fig. 4: Accuracy per classification algorithm and best feature set per sensor location combination on the mHealth dataset

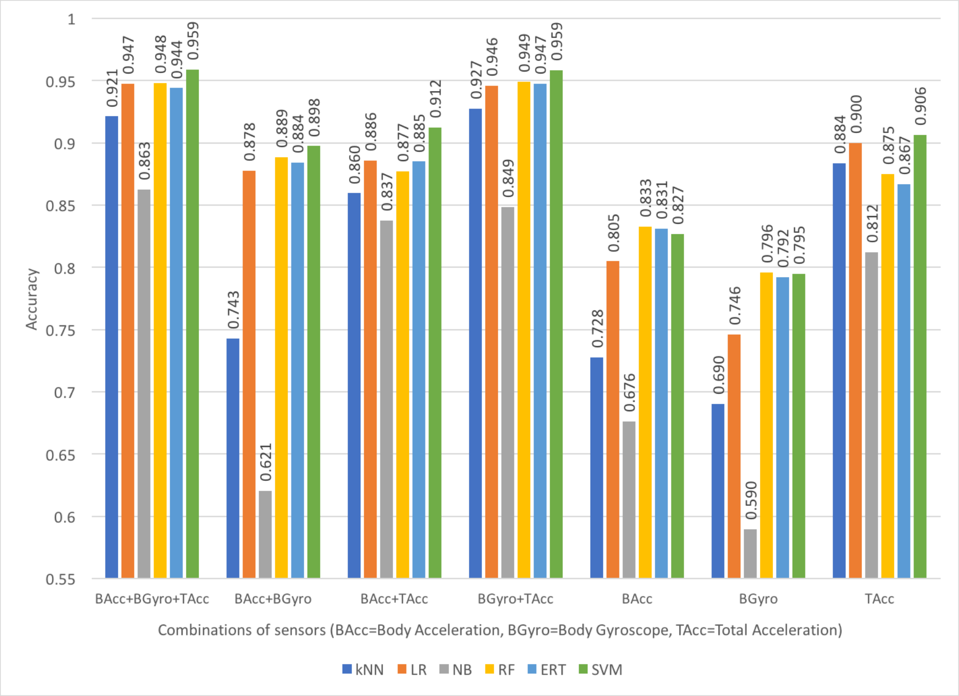


Fig. 5: Accuracy per classification algorithm and best feature set per sensor combination on the SBHAR dataset

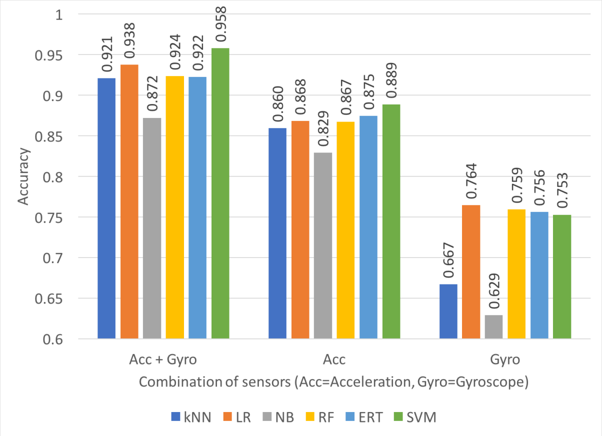


Fig. 6: Accuracy per classification algorithm and best feature set per sensor combination on the SBHARPT dataset

decreased considerably after the DFB feature selection.

All feature extraction and machine learning algorithms were implemented in Python 3 and were executed on a 2.3 GHz quad-core Intel Core i7 processor (with Turbo Boost up to 3.3GHz) with 6MB L3 cache and 16 GB RAM, which should provide context for the execution times. The segmentation (step 1 in Fig. 2) was peformed sequentially, and the maximum time for all datasets was two minutes. The feature extraction (steps 2 to 10 Fig. 2) required at most two hours in total (for the FST dataset). The feature importance and data drift sensitivity estimation (step 11 in Fig. 2) required 10 minutes at most. Step 12 in Fig. 2, which performed score-drift screening, executed for 30 minutes for the largest dataset (i.e. FST). The DFB feature selection (step 13 in Fig. 2) took at most 5 minutes for all datasets, considering that it started with an empty set and gradually added and removed features until maximum of 2000 feature sets were evaluated. Finally, step 14 required usually less than one minute and at most 5 minutes to create a classification model and make predictions for the test set (see Fig. 9). The reported times for steps 12, 13 and 14 are when using all available sensors for all datasets. These three steps were repeated for each sensor location combination, thus increasing the total time needed to perform all experiments.

IV. Discussion

This study presented a systematic feature engineering process from body worn sensors which resulted in reliable recognition of human activities of daily living. Different classifiers with different feature selection methods have been applied on several publicly available datasets containing sensory data from multiple body positions, and the influence of sensor locations on the classification accuracy was analyzed.

From Fig. 7, which shows the maximum accuracy depending on the combination of sensor locations, it is evident that some combinations are consistently better than others. Clearly, from the two sensor locations the ankle plus wrist combination results in best accuracy for both the DaLiAc and mHealth datasets. For these two datasets additionally including a sensor on the chest improves the accuracy. Another peculiar discovery is that for different datasets, different isolated sensor locations offered best results. For the DaLiAc dataset it was the chest sensor, for the mHealth it was the ankle sensor, and for the FSB dataset it was the left or right jeans pocket. For the SBHAR and SBHARPT datasets only one sensor location was utilized, therefore we cannot make comparisons.

Another difference between datasets that becomes apparent from Fig. 7 is that the same sensor in different datasets yields very different accuracies. For instance, the sensor combinations shown on the left in Fig. 7 result in considerably different accuracy in the DaLiAc and mHealth datasets, as do the wrist sensor for DaLiaAc, mHealth and FSP datasets. We attribute this difference to the different experimental setup and somewhat different activities analyzed in the different studies (see Table IV).

The score-drift feature screening method (Step 12 in Fig. 2), which discarded low-informative features or ones with high data distribution sensitivity, had huge impact on the accuracy and time, while considerably reducing the number of features (see Table II). All algorithms benefited from the DFB feature selection (see Steps 12 and 13 in Fig. 2) in terms of execution time (see Fig. 9), while some of them experienced a minor reduction in accuracy (see Fig. 8). The algorithms that benefited the most from the reduced feature sets are LR, kNN and SVMs, whereas NB, RF and ERT still benefited, but not as much. As expected, the LR benefited the most in terms of accuracy from the DFB feature selection when it was used as a wrapper algorithm. Similarly, the NB accuracy on the test set was also increased when it was used as a wrapper algorithm during DFB feature selection. Other algorithms were not as applicable for wrapper algorithms in DFB feature selection. The ERT and RF evaluate multiple feature and data subsets at random, thus making the addition of removal of one feature non-transparent and the impact on accuracy could be neglected. Because the performance of the SVMs is highly sensitive to the values of their hyper parameters, each feature set can potentially have different optimal values, making the search of best feature set dependent additionally on the hyper parameters, and therefore much more complex.

With the DaLiAc dataset the best accuracy was 0.934, obtained using three (i.e. ankle, chest, wrist) out of the four sensor locations (i.e. ankle, chest, hip, wrist), as shown in Fig. 3. From all two-sensor combinations, the best accuracy was 0.918, obtained from the either the ankle and wrist sensors, or the chest and wrist sensors. When only one sensor location was used, the chest sensor provided best accuracy (0.891) and it was considerably better than the other sensors. It is evident that the SVM, RF and ERT models were consistently better for all combination of sensors than the other classifiers, while LR was the worst in terms of accuracy. Compared to the study [3], which published this dataset and obtained 0.896 accuracy, our approach resulted in higher accuracy (0.934). Note that their study used leave-one-subject-out evaluation strategy, whereas ours used an independent test set.

On the mHealth dataset the highest obtained accuracy was 0.998, using the three sensors placed at left ankle, right wrist and chest (see Fig. 4). The study [12], which originally published this dataset, did not use the ECG sensor for activity

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2684913, IEEE Access

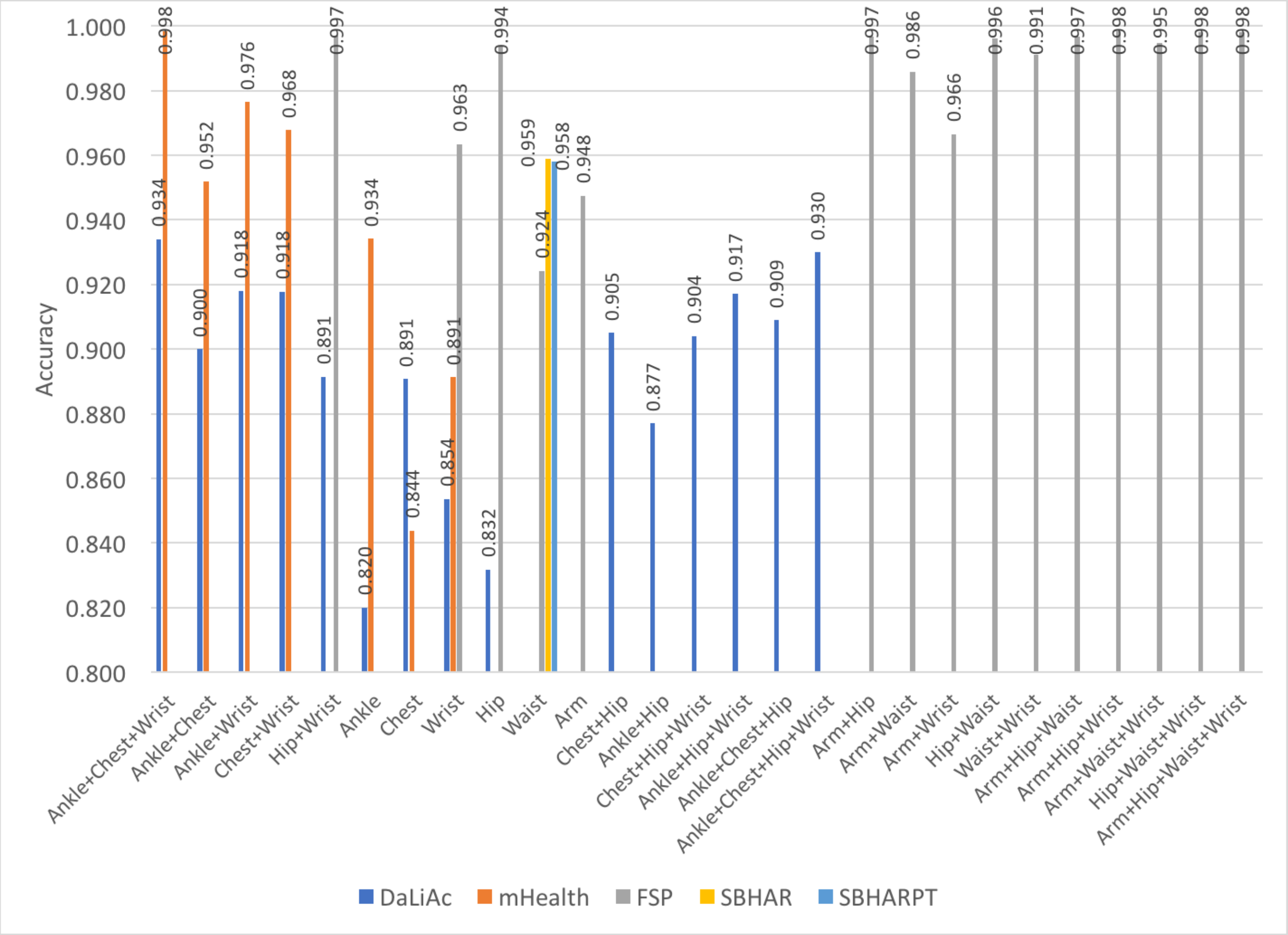
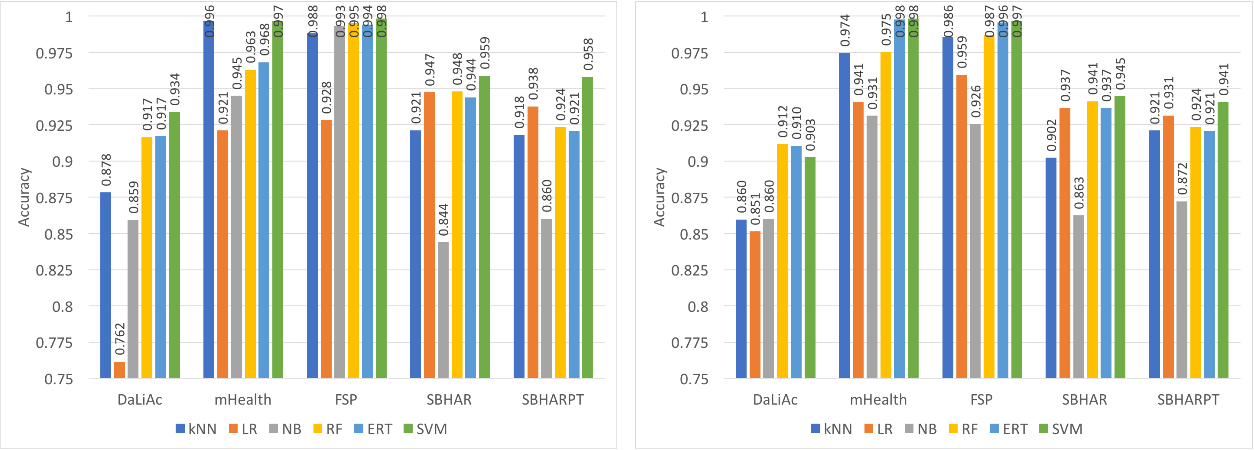


Fig. 7: Accuracy per combination of sensor locations and dataset. Equivalence of sensor locations: mHealth Left Ankle => Ankle; FSP Belt => Waist, FSP Right Jeans Pocket and FSP Left Jeans Pocket => Hip.

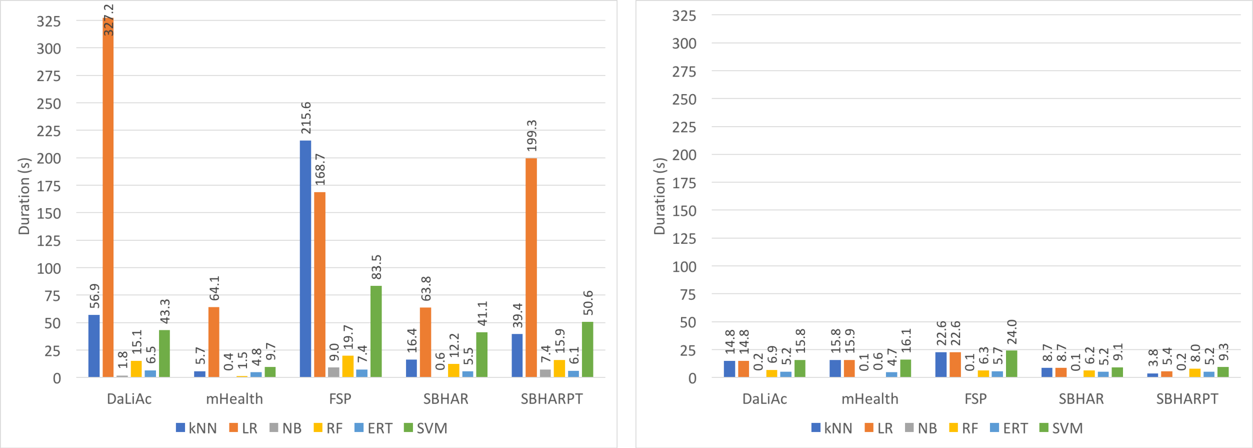


(a) Best feature set obtained with score-drift feature screening

(b) Best feature set obtained with DFB feature selection

Fig. 8: Accuracy per classification algorithm and dataset on the two final feature sets.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2017.2684913, IEEE Access



(a) Best feature set obtained with score-drift feature screening

(b) Best feature set obtained with DFB feature selection

Fig. 9: Time needed to build models on the union of the training and validation sets and make predictions on the test set, per classification algorithm and dataset on the two final feature sets.

recognition. We have also shown that this sensor has very limited ability to recognize actions and is not useful neither in isolation nor in combination with other sensors (see Fig. 4). From the two sensor combinations, the left ankle plus right wrist provided the highest accuracy (0.97). Individually, the left ankle sensor resulted in highest accuracy (0.934). The best classifiers for all combination of sensor locations were a kNN with 5 neighbors and an SVM. Similar to the very high accuracy (see Fig. 4) and precision and recall per activity (Table IV) obtained by our approach, [12] also reports almost perfect precision and recall.

The FSP dataset resulted in very high accuracy for different sensor location combinations (see Table III). Interestingly, using only 2 sensor locations (e.g. on the wrist and in one of the pockets), the maximum obtained accuracy was 0.9975. When using sensors on 3, 4 or even 5 locations, the maximum accuracy was 0.9984, meaning that the more sensors did not improve the accuracy significantly. Another peculiar discovery is that even one sensor (i.e. smart phone) in either of the pockets provides a very high accuracy 0.9938, which is significantly better than the highest accuracy obtained from sensors on any of the other locations (e.g. wrist - 0.9635, arm - 0.9475, belt - 0.9242). Another interesting discovery are the pairs of redundant sensor locations. Sensors on the arm and wrist resulted in accuracy of 0.9665, which is only marginally better than using only a sensor on the wrist (0.9935). Likewise, using sensors only in both pockets (accuracy 0.9898) is even worse than using only one in either pocket (accuracy 0.9938). We attribute this to the more effort required to find an optimal feature subset from larger feature sets, which is a consequence of using more sensors.

The fact that a sensor in one of the pockets and on the wrist result in almost the best accuracy is very important, because it resembles a very realistic use-case when a user carries a smart phone in one of his pockets and wears a smart watch. This encouraging realization means that with the least intrusive sensor location (a smartphone in the pocket and a smart watch), human activity recognition can be performed very reliably. The best precision per activity obtained in our study is significantly better than in the original study [14] for all activities. In fact, for all activities we obtained a precision of at least 0.9947. Note that, we used an independent test set from users that were not used for building models, while the original study used 10-fold cross validation (which generally is considered as a more optimistic strategy [54]).

The highest accuracy for the SBHAR dataset obtained in the studies [57], [15] was 0.96, similar to the accuracy of 0.959 obtained in our study. We have showed that separating the body acceleration out of the total acceleration recorded by the sensor with frequency filtering was not necessary, as our feature extraction framework was able to come up with robust features (see Fig. 5). This is an important finding that can be used to simplify the feature extraction process in variety of pattern recognition applications based on sensory data because we showed that there is no need of specialized noise filtering. Individually, the total acceleration yielded an accuracy of 0.906, whereas the gyroscope resulted in considerably lower accuracy of 0.796.

For the SBHARPT dataset, which is an extended version of the SBHAR dataset that includes postural transitions in addition to the activities already available, we have obtained an accuracy of 0.958 (see Fig. 6). It is somewhat higher than the accuracy of 0.926, reported in [16]. Another advantage of our approach is that it is generic, unlike the approach used in [16], which requires a special assessment method specific to the different postural transitioning conditions that can appear.

The impact of the feature selection methods on number of features can be analyzed from Table II and Fig. 8. Clearly, both methods reduce the number of features significantly. The fast feature ranking method that screens features reduces the number of features by 3 to 5 times, while notably increasing the accuracy. The finer grained DFB feature selection further reduces the number of features and for all datasets the number of selected features is at most 60, which considerably improves the execution time (see Fig. 9).

In regards to the classification algorithms, generally SVMs provided best accuracy, albeit requiring significant effort for parameter tuning, while also being the slowest, considering the time needed for building classification models and making predictions. ERTs performed consistently good on all datasets while also being very fast. Furthermore, redundant and uninformative features did not affect their accuracy significantly. RFs performed comparably to ERTs, even though they were somewhat slower. Both ERTs and RFs were very fast, owing to the parallel implementation. In terms of accuracy, next were kNNs, which were also fast for smaller feature sets, but could be even the slowest in cases of large feature sets. In general, LR and NB performed considerably worse than the other algorithms in terms of accuracy, but were the fastest. However, in case of small number of classes (i.e. activities of interest), LR yielded very good accuracy, comparable to the best on those datasets. Anyhow, LR and NB were very useful for the feature selection process as wrapper algorithms due to their speed, low memory requirements and ability to asses the impact on the accuracy of single features when added to or removed from a feature set. Additionally, they are most suitable for execution on devices with limited memory and computational requirements.

V. Conclusion

ADL recognition is the most common task in AAL systems, therefore it receives great attention from the scientific communities. For successful deployment of AAL systems there are three main factors: cost, reliable performance and acceptance by end users. To address them, in this paper we have proposed a method for determining the optimal number of sensors, sensor positions and optimal number of features. This is performed by a framework for automatic feature engineering for robust recognition of ADLs, regardless of the number of sensors, their types, or body placement. The proposed method reduces the time to determine which features are best suited for a particular task because no manual effort is required, rather a variety of features are automatically generated and the best ones are selected. By determining the best possible feature set, the recognition performance is boosted. Using a small but robust feature set can facilitate execution on hardware with limited resources.

By performing experiments on five publicly available datasets, we have evaluated the proposed framework with a holdout set, consisted of data from subjects that were not used for building classification models. Due to the variety of generated features and the diligent feature selection, for each dataset we could select small feature sets that yielded better or at least the same classification accuracy as the original studies that published the datasets. In line with this, all combinations of sensor locations were automatically analyzed, which identified the essential sensor locations and the locations that are redundant, for each dataset separately. Additionally, a comparison between the different datasets was performed in order to identify the precision and recall per activity, depending on the used sensor locations.

The common application of the proposed method would be during the design phase of AAL systems for offline model building and evaluation. Subsequently, the findings from the experiments could be used for identifying key features and appropriate machine learning algorithm that can be later employed for activity recognition in online mode, even on mobile devices.

Using the proposed approach and the evaluated datasets, we were able to identify that using only a smartphone and a smartwatch, a very high accuracy of activity recognition can be achieved. This finding is very important and encouraging because these two devices are very common nowadays, and are not perceived as too-intrusive. Therefore, they may be employed for ADL recognition without reluctance and aversion from the subjects, as they could be already carrying them, which may reduce the cost of AAL systems for ADL recognition because specialized hardware would not be required.

Conflict of Interests

The authors declare that there is no conflict of interest regarding the publication of this paper.

Acknowledgment

This work was partially financed by the Faculty of Computer Science and Engineering at the Ss.Cyril and Methodius University, Skopje, Macedonia and is supported by the networking activities provided by the ICT COST Action IC1303 AAPELE. Authors affiliated with the Instituto de Telecomunicações also acknowledge the funding for the research by means of the program FCT project UID/EEA/50008/2013.

References

[1] B. Przywara *et al.*, “Projecting future health care expenditure at european level: drivers, methodology and main results,” Directorate General Economic and Monetary Affairs (DG ECFIN), European Commission, Tech. Rep., 2010.

[2] H. Sun, V. D. Florio, N. Gui, and C. Blondia, “Promises and challenges of ambient assisted living systems,” in *2009 Sixth International Conference on Information Technology: New Generations*, April 2009, pp. 1201–1207.

[3] H. Leutheuser, D. Schuldhaus, and B. M. Eskofier, “Hierarchical, multi-sensor based classification of daily life activities: Comparison with state-of-the-art algorithms using a benchmark dataset,” *PLoS ONE*, vol. 8, no. 10, p. e75196, 10 2013. [Online]. Available: http://dx.doi.org/10.1371%2Fjournal.pone.0075196

[4] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, “Complex human activity recognition using smartphone and wrist-worn motion sensors,” *Sensors (Basel, Switzerland)*, vol. 16, no. 4, p. 426, 04 2016. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4850940/

[5] E. Fortune, V. Lugade, S. Amin, and K. Kaufman, “Step detection using multi- versus single tri-axial accelerometer-based systems,” *Physiological measurement*, vol. 36, no. 12, pp. 2519–2535, 12 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4838513/

[6] M. Munoz-Organero and A. Lotfi, “Human movement recognition based on the stochastic characterisation of acceleration data,” *Sensors*, vol. 16, no. 9, 2016. [Online]. Available: http://www.mdpi.com/1424-8220/16/9/1464

[7] T. Bastian, A. Maire, J. Dugas, A. Ataya, C. Villars, F. Gris, E. Perrin, Y. Caritu, M. Doron, S. Blanc, P. Jallon, and C. Simon, “Automatic identification of physical activity types and sedentary behaviors from triaxial accelerometer: laboratory-based calibrations are not enough,” *Journal of Applied Physiology*, vol. 118, no. 6, pp. 716–722, 2015. [Online]. Available: http://jap.physiology.org/content/118/6/716

[8] P. C. Hallal, L. B. Andersen, F. C. Bull, R. Guthold, W. Haskell, and U. Ekelund, "Global physical activity levels: surveillance progress, pitfalls, and prospects," *The Lancet*, vol. 380, no. 9838, pp. 247–257, 2017/01/24. [Online]. Available: http://dx.doi.org/10.1016/S0140-6736(12)60646-1

[9] S. Kahlmeier, T. M. A. Wijnhoven, P. Alpiger, C. Schweizer, J. Breda, and B. W. Martin, "National physical activity recommendations: systematic overview and analysis of the situation in european countries," *BMC Public Health*, vol. 15, p. 133, 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4404650/

[10] W. H. Organization *et al.*, *Global recommendations on Physical Activity for health.* Geneva, Switzerland: World Health Organization, 2010.

[11] E. M. Tapia, S. S. Intille, and K. Larson, "Activity recognition in the home using simple and ubiquitous sensors," in *International Conference on Pervasive Computing*. Springer, 2004, pp. 158–175.

[12] O. Banos, C. Villalonga, R. Garcia, A. Saez, M. Damas, J. A. Holgado-Terriza, S. Lee, H. Pomares, and I. Rojas, "Design, implementation and validation of a novel open framework for agile development of mobile health applications," *BioMedical Engineering OnLine*, vol. 14, no. Suppl 2, pp. S6–S6, 2015. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4547155/

[13] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu, "Elderly activities recognition and classification for applications in assisted living," *Expert Syst. Appl.*, vol. 40, no. 5, pp. 1662–1674, Apr. 2013. [Online]. Available: http://dx.doi.org/10.1016/j.eswa.2012.09.004

[14] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. M. Havinga, "Fusion of smartphone motion sensors for physical activity recognition," *Sensors (Basel, Switzerland)*, vol. 14, no. 6, pp. 10 146–10 176, 06 2014. [Online]. Available: http://www.ncbi.nlm.nih.gov/pmc/articles/PMC4118351/

[15] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Energy efficient smartphone-based activity recognition using fixed-point arithmetic," *Journal of Universal Computer Science*, vol. 19, no. 9, pp. 1295–1314, may 2013. [Online]. Available: http://www.jucs.org/jucs_19_9/energy_efficient_smartphone_based

[16] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754 – 767, 2016. [Online]. Available: //www.sciencedirect.com/science/article/pii/S0925231215010930

[17] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *J. Mach. Learn. Res.*, vol. 3, pp. 1157–1182, Mar. 2003. [Online]. Available: http://dl.acm.org/citation.cfm?id=944919.944968

[18] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/4/6474

[19] L. Bao and S. S. Intille, *Activity Recognition from User-Annotated Acceleration Data.* Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 1–17. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-24646-6_1

[20] S. J. Preece*, J. Y. Goulermas, L. P. J. Kenney, and D. Howard, "A comparison of feature extraction methods for the classification of dynamic activities from accelerometer data," *IEEE Transactions on Biomedical Engineering*, vol. 56, no. 3, pp. 871–879, March 2009.

[21] M. Meina, A. Janusz, K. Rykaczewski, D. Slezak, B. Celmer, and A. Krasuski, "Tagging firefighter activities at the emergency scene: Summary of aaia'15 data mining competition at knowledge pit," in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*, Sept 2015, pp. 367–373.

[22] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Communications Surveys Tutorials*, vol. 15, no. 3, pp. 1192–1209, Third 2013.

[23] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[24] D. W. Hosmer Jr, S. Lemeshow, and R. X. Sturdivant, *Applied logistic regression*. John Wiley & Sons, 2013, vol. 398.

[25] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001. [Online]. Available: http://dx.doi.org/10.1023/A:1010933404324

[26] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine learning*, vol. 63, no. 1, pp. 3–42, 2006.

[27] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[28] C. Cortes and V. Vapnik, "Support-vector networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995. [Online]. Available: http://dx.doi.org/10.1007/BF00994018

[29] P. Lameski, E. Zdravevski, R. Mingov, and A. Kulakov, "Svm parameter tuning with grid search and its impact on reduction of model overfitting," in *Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*. Springer, 2015, pp. 464–474.

[30] N. Ravi, N. Dandekar, P. Mysore, and M. L. Littman, "Activity recognition from accelerometer data," in *Aaai*, vol. 5, no. 2005, 2005, pp. 1541–1546.

[31] M. Keally, G. Zhou, G. Xing, J. Wu, and A. Pyles, "Pbn: towards practical activity recognition using smartphone-based body sensor networks," in *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2011, pp. 246–259.

[32] D. T. Larose, "k-nearest neighbor algorithm," *Discovering Knowledge in Data: An Introduction to Data Mining*, pp. 90–106, 2005.

[33] I. Rish, "An empirical study of the naive bayes classifier," in *IJCAI 2001 workshop on empirical methods in artificial intelligence*, vol. 3, no. 22. IBM New York, 2001, pp. 41–46.

[34] D. M. Karantonis, M. R. Narayanan, M. Mathie, N. H. Lovell, and B. G. Celler, "Implementation of a real-time human movement classifier using a triaxial accelerometer for ambulatory monitoring," *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 156–167, Jan 2006.

[35] A. M. Khan, A. Tufail, A. M. Khattak, and T. H. Laine, "Activity recognition on smartphones via sensor-fusion and kda-based svms," *International Journal of Distributed Sensor Networks*, vol. 10, no. 5, p. 503291, 2014. [Online]. Available: http://dx.doi.org/10.1155/2014/503291

[36] J. Lasek and M. Gagolewski, "The winning solution to the aaia'15 data mining competition: Tagging firefighter activities at a fire scene," in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. P. M. Ganzha, L. Maciaszek, Ed., vol. 5. IEEE, 2015, pp. 375–380. [Online]. Available: http://dx.doi.org/10.15439/2015F418

[37] E. Zdravevski, P. Lameski, A. Kulakov, and S. Kalajdziski, "Transformation of nominal features into numeric in supervised multi-class problems based on the weight of evidence parameter," in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sept 2015, pp. 169–179.

[38] A. Janusz, M. Sikora, Ł. Wróbel, S. Stawicki, M. Grzegorowski, P. Wojtas, and D. Ślęzak, *Mining Data from Coal Mines: IJCRS'15 Data Challenge*. Cham: Springer International Publishing, 2015, pp. 429–438.

[39] H. A. Sturges, "The choice of a class interval," *Journal of the American Statistical Association*, vol. 21, no. 153, pp. 65–66, 1926.

[40] P. Siirtola and J. Röning, "Recognizing human activities user-independently on smartphones based on accelerometer data," *International Journal of Artificial Intelligence and Interactive Multimedia*, vol. 1, no. 5, pp. 38–45, 2012.

[41] H. Martín, A. M. Bernardos, J. Iglesias, and J. R. Casar, "Activity logging using lightweight classification techniques in mobile devices," *Personal and Ubiquitous Computing*, vol. 17, no. 4, pp. 675–695, 2013. [Online]. Available: http://dx.doi.org/10.1007/s00779-012-0515-4

[42] A. Zagorecki, "A versatile approach to classification of multivariate time series data," in *Proceedings of the 2015 Federated Conference on Computer Science and Information Systems*, ser. Annals of Computer Science and Information Systems, M. P. M. Ganzha, L. Maciaszek, Ed., vol. 5. IEEE, 2015, pp. 407–410. [Online]. Available: http://dx.doi.org/10.15439/2015F419

[43] E. Zdravevski, P. Lameski, R. Mingov, A. Kulakov, and D. Gjorgjevikj, "Robust histogram-based feature engineering of time series data," in *Computer Science and Information Systems (FedCSIS), 2015 Federated Conference on*, ser. Annals of Computer Science and Information Systems, M. P. M. Ganzha, L. Maciaszek, Ed., vol. 5. IEEE, Sept 2015, pp. 381–388. [Online]. Available: http://dx.doi.org/10.15439/2015F420

[44] M. Boullé, "Tagging fireworkers activities from body sensors under distribution drift," in *2015 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sept 2015, pp. 389–396.

[45] E. Zdravevski, P. Lameski, and A. Kulakov, "Automatic feature engineering for prediction of dangerous seismic activities in coal mines," in *2016 Federated Conference on Computer Science and Information Systems (FedCSIS)*, Sept 2016, pp. 245–248. [Online]. Available: http://dx.doi.org/10.15439/2016F152

[46] D. Ruta, "Robust method of sparse feature selection for multi-label classification with naive bayes," in *2014 Federated Conference on Computer Science and Information Systems*, Sept 2014, pp. 375–380.

[47] D. S. WARD, K. R. EVENSON, A. VAUGHN, A. B. RODGERS, and R. P. TROIANO, “Accelerometer use in physical activity: Best practices and research recommendations,” pp. S582–S588, 2005. [Online]. Available: http://journals.lww.com/acsm-msse/Fulltext/2005/11001/Accelerometer_Use_in_Physical_Activity__Best.11.aspx

[48] S. G. TROST, K. L. MCIVER, and R. R. PATE, “Conducting accelerometer-based activity assessments in field-based research,” pp. S531–S543, 2005. [Online]. Available: http://journals.lww.com/acsm-msse/Fulltext/2005/11001/Conducting_Accelerometer_Based_Activity.6.aspx

[49] M. Ermes, J. PÄrkkÄ, J. MÄntyjÄrvi, and I. Korhonen, “Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, no. 1, pp. 20–26, Jan 2008.

[50] S. Liu, R. X. Gao, D. John, J. W. Staudenmayer, and P. S. Freedson, “Multisensor data fusion for physical activity assessment,” *IEEE Transactions on Biomedical Engineering*, vol. 59, no. 3, pp. 687–696, March 2012.

[51] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, “Activity classification using realistic data from wearable sensors,” *IEEE Transactions on Information Technology in Biomedicine*, vol. 10, no. 1, pp. 119–128, Jan 2006.

[52] E. Garcia-Ceja, R. F. Brena, J. C. Carrasco-Jimenez, and L. Garrido, “Long-term activity recognition from wristwatch accelerometer data,” *Sensors*, vol. 14, no. 12, pp. 22 500–22 524, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/12/22500

[53] B. Mariani, C. Hoskovec, S. Rochat, C. Büla, J. Penders, and K. Aminian, “3d gait assessment in young and elderly subjects using foot-worn inertial sensors,” *Journal of Biomechanics*, vol. 43, no. 15, pp. 2999–3006, 2017/01/31. [Online]. Available: http://dx.doi.org/10.1016/j.jbiomech.2010.07.003

[54] R. Kohavi, “A study of cross-validation and bootstrap for accuracy estimation and model selection,” in *Proceedings of the 14th International Joint Conference on Artificial Intelligence - Volume 2*, ser. IJCAI’95. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1995, pp. 1137–1143. [Online]. Available: http://dl.acm.org/citation.cfm?id=1643031.1643047

[55] S. Arlot, A. Celisse *et al.*, “A survey of cross-validation procedures for model selection,” *Statistics surveys*, vol. 4, pp. 40–79, 2010.

[56] A. Burns, B. R. Greene, M. J. McGrath, T. J. O’Shea, B. Kuris, S. M. Ayer, F. Stroiescu, and V. Cionca, “Shimmer x2122; x2013; a wireless sensor platform for noninvasive biomedical research,” *IEEE Sensors Journal*, vol. 10, no. 9, pp. 1527–1534, Sept 2010.

[57] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition using smartphones.” in *European Symposium on Artificial Neural Networks (ESANN)*, ser. Computational Intelligence and Machine Learning, Apr 2013.