Considerations for the Design of an Activity Recognition System Using Inertial Sensors

Athanasios I. Kyritsis, Michel Deriaz, and Dimitri Konstantas
Information Science Institute, GSEM/CUI
University of Geneva
Geneva, Switzerland
Email: $\{athanasios.kyritsis, michel.deriaz, dimitri.konstantas\}$@unige.ch

***Abstract*—The last decade there has been an increasing research interest in the field of human activity recognition in the frame of designing context-aware applications. There is a plethora of parameters that affect the performance of an activity recognition system. However, designers of such systems often either ignore some factors or even neglect their importance. In this paper, we present and discuss in detail research challenges in human activity recognition using inertial sensors, and we analyse the significance of the existent parameters during the design and the evaluation of such systems. We exemplify the role of the aforementioned parameters with an experiment that was conducted, in which 11 people performed 5 different activities. Data were recorded from the inertial sensors of a wrist-worn smartwatch. We illustrate how various parameters of the system can be configured and demonstrate how they impact the whole system's performance. This work aims to be used as a concise reference for future endeavours in the field of human activity recognition using inertial sensors of mobile devices in general, and of wrist-worn smartwatches in particular.**

***Keywords*—Activity recognition, feature extraction, machine learning, pattern recognition, smart devices, wearable sensors.**

I. Introduction and Related Work

Over the past decade, there have been great efforts towards Activity Recognition (AR) methods and techniques both by researchers and the industry. There are many applications that either require or would benefit from AR. Healthcare monitoring systems use sensors to track Activities of Daily Living (ADL) of older adults and assist the work of caregivers [1]. Besides the healthcare sector, other domains that benefit from AR include sports [2], entertainment [3] and the industrial sector [4].

Several commercial products rely on AR. All major video game console manufacturers have developed such systems. Nintendo recognises gestures using the inertial sensors of the handheld controllers starting from the Nintendo Wii console [5]. Microsoft identifies activities by monitoring full-body movement using the Kinect camera [3]. Sony uses inertial sensor data from the controllers and tracks them in space using a camera with the PlayStation Move system. All these systems, while initially developed for entertainment, have also been used by researchers for rehabilitation purposes [6]. Modern Virtual Reality (VR) consumer products, like the HTC Vive, use both Inertial Measurement Units (IMU) in hand-held controllers and cameras that detect user gestures and activities. Although these devices have initially targeted gaming, researchers have exploited their capabilities so that they can be used in other domains, like in health applications [7].

The proposed approaches for AR systems can be roughly divided into two categories, the inertial sensor-based ones [8] and the camera-based ones [9]. In the sensor-based methods, one or more inertial sensors, such as accelerometers and gyroscopes, are attached to the human body. Time-series techniques are applied to the collected signals to extract useful information. In the camera-based methods, different computer vision techniques are employed to obtain analytical results.

Camera-based AR methods use video sequences recorded by video cameras to detect human gestures and activities. AR is an important domain of research in computer vision, and its applications include patient monitoring systems and video surveillance systems [10]. Processing and feature extraction from raw videos target the finding of specific characteristics such as colours, shapes and body motions that can describe human activity. These features can also be used for body model reconstruction [11]. Despite all the progress made for AR using vision-based methods, they pose particular limitations. They are intrusive and thus can not be used in applications where privacy is a requirement. Moreover, due to the fixed locations of the cameras, these techniques can not be used for real-time applications where constant monitoring is required.

Inertial sensor-based AR techniques overcome these last limitations. The increased availability of such sensors due to the omnipresence of smartphones and smartwatches has enabled the use of AR techniques in ubiquitous computing. Sensor-based AR systems are either knowledge-driven or data-driven. Knowledge-driven approaches use prior domain knowledge to build an abstract model and apply the model to the recorded data [12]. On the contrary, data-driven approaches work by extracting correlations between data and gestures and eventually build a model for classification [8].

While there are already a lot of works and applications of AR, very few of them [13] discuss the parameters and the choices that impact the performance of an AR system. Usually,

This work was supported by the Commission for Technology and Innovation CTI, of the Swiss federal government, (CTI grant 18566.2 PFES-ES) in the frame of the Recover@home project.

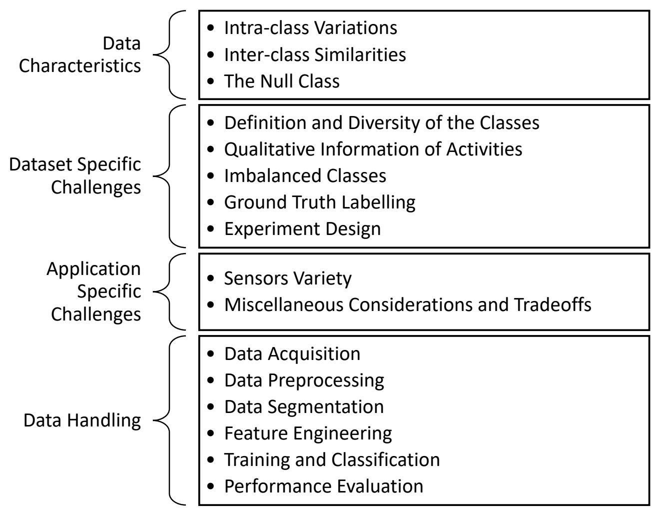

Fig. 1. Summary of the design considerations of an activity recognition system.

those works present a single application specific best solution. The scope of this paper is to discuss the various design parameters that are crucial when performing AR. An AR system is developed to show how different variables impact the performance of it. Developing the optimal AR system would require a massive dataset, a lot of experiments and computing power for parameter tuning, and is out of the scope of this publication. There is a large variety in every step of creating an AR system, starting from the data acquisition and the feature engineering up to the training and classification phases, and every single step can significantly impact the system as a whole.

The rest of the paper is organised as follows. In Section II we discuss the considerations and the parameters that affect every AR system in the design, implementation, testing and evaluation phases. In Section III we present the experiment that we have conducted and in Section IV we evaluate a variety of tests and show how each parameter tuning may impact the performance of the overall system. Finally, we conclude our work in Section V.

II. Design Considerations

There are many challenges when designing an AR system. In this section, we are presenting a list of considerations and parameters that impact every AR system during the design, implementation, testing and evaluating phase. Fig. 1 summarises the considerations that are further discussed.

A. Data Characteristics

1) Intra-class Variations: The first challenge of any AR system is to be robust to intra-class variations. Intra-class variations mainly exist because the different activities are performed differently by various people. For example, if inertial sensors are attached to the wrists of two people running, it is very improbable that we observe a similar swing in both cases. Intra-class variations can also exist among activities performed by the same person. For example when a person is walking to catch a bus and when the same person is taking an after-work walk. Theoretically, the prediction from both cases should belong to the "walking" class, but in practice, also depending on where the inertial sensors are placed, the recorded data might significantly differ. To tackle with this issue, the designer of the system should capture a significant amount of training data, both from a single person but also from several others, for the dataset to capture as much variability as possible. It is crucial for a model to avoid overfitting and be robust across different people.

2) Inter-class Similarities: Another challenge arises when a system aims to identify classes that are different, but the recorded sensor data show very similar characteristics. An example would be to differentiate between the activity of walking with a stroller and the one of walking with a shopping cart. To tackle this issue, the designer can either take more and different sensors into account or observe other correlated activities that take place in parallel [14].

3) The Null Class: In a continuous data collection setting, like in an all-day running AR system, the indifferent to the designer classes for identification may be more and may form the majority of the collected dataset. In that case, the irrelevant activities that have similar characteristics to the relevant ones form the null class. The null class is extremely hard to model since it encapsulates an arbitrary and theoretically infinite number of activities. The simplest way would be to identify the null class with the data patterns that significantly differ from those of the desired activity. Other methods like self-training [15], may allow the designer to use the null class during the training of the AR model.

B. Dataset Specific Challenges

1) Definition and Diversity of the Classes: When designing an AR system, it is essential to define the classes of the activities that are of interest to the specific application. Although this task seems to be of little importance, it can be tricky because human activities are highly diverse, can be performed in many ways and sometimes even the definition of them can vary. There have been researches on the definition of a taxonomy of activities [16] that can prove a good reference for AR systems designers.

2) Qualitative Information of Activities: While the vast majority of research for AR systems aims at detecting the activity that is being performed at a specific moment, little progress has been made on the extraction of its qualitative information. It would be interesting for example for physiotherapists to be able to know if their patients are executing the prescribed activities correctly, and if not, to understand the source of the problem. Such research has so far only targeted the sports sector, and the settings were too constrained [17].

3) Imbalanced Classes: Another challenge of any AR system arises when the classes to be modelled do not exist in similar quantities in the training dataset. This problem is profound in long-term monitoring settings because only a few activities take place frequently, e.g. walking, while others scarcely, e.g. doing squats. Possible ways for a researcher to ameliorate this situation include the recording of additional training data of the underrepresented classes, the generation of simulated training data and oversampling the smaller classes [18].

4) Ground Truth Labelling: A necessary, laborious and time-consuming task for all supervised AR systems is the annotation of the training data. Post hoc annotation is possible for data captured from cameras by labelling the footage but is difficult to achieve with inertial sensor data. In laboratory settings, the researcher can annotate the data in real time, but in daily life situations, the user has to label the data with the ground truth with techniques like the experience sampling method [19].

5) Experiment Design: An essential aspect of any AR system is the data collection and the overall design of the experiment. So far there are only a few general purpose datasets [20] that can be used for activity recognition and there is no commonly agreed way to collect data. The recorded data depend on the designer of the experiment, and usually, every study has different priorities, e.g. a large number of participants, a large dataset, clean data, etc. Datasets are also not always publicly available to be reused in other experiments. To be able to have comparable and reproducible scientific results and focus more on the methods for data analysis, it is crucial for the scientific community to commonly agree on some standard data acquisition guidelines and datasets.

C. Application Specific Challenges

1) Sensors Variety: There is considerable variability in the available sensing equipment. Apparently, every sensor has different specifications given by the manufacturer of it, such as sampling frequency, accuracy, precision and operating temperature range. Moreover, mobile devices can be used in different ways. For example, for the same activity, one would expect different recordings from the inertial sensors of a smartwatch worn on the wrist than from a smartphone kept in the pocket.

Smart devices embedded three-axis accelerometers and three-axis gyroscopes are most commonly used in inertial sensor-based AR systems [21]. Fusing both accelerometer and gyroscope data usually leads to a better recognition performance than when only using a single source of data. A three-axis magnetometer can be also used in conjunction with the aforementioned sensors in order to optimize the detection of the orientation of the user in space [22]. A nine-axis inertial sensor refers to a three-axis accelerometer, a three-axis gyroscope and a three-axis magnetometer enclosed in a single module.

2) Miscellaneous Considerations and Tradeoffs: There are many tradeoffs that each AR system designer should consider. Some computer applications that rely on gesture recognition should run in real time, while for others such as monitoring long-term behaviour, for example, an offline analysis may suffice. There are also tradeoffs associated with the accuracy, the power consumption and the latency of the system. In case the application runs on a mobile device, power efficiency should be taken into account sometimes even opposed to the accuracy and the latency of it [23].

D. Data Handling

1) Data Acquisition: The first step in any IMU-based AR system is to capture raw data using different inertial sensors, attached to different locations of the user's body. There are also advanced systems that use even more environment sensors to record additional data [24]. Some sensors provide multiple values, like for example the accelerometer that gives three recordings, one for each of the x, y and z-axes. Each sensor is sampled at regular time intervals, and the recorded raw data correspond to a multivariate time-series dataset. The sampling frequency is different per sensor and sometimes can be set according to the requirements of the application, e.g. for power saving.

2) Data Preprocessing: Before proceeding to extract features from the available raw dataset, the data need to be preprocessed and cleaned. Among the raw data, there might be artifacts caused by electromagnetic interference that need to be filtered out. Also, data streams from different sensors should be synchronised at that point. In case the streams have a different sampling rate, they should be resampled with the same frequency. Data may also be calibrated according to the sensor characteristics and normalised. Data preprocessing is a generic step, and the same actions should be applied to all input data with no exceptions [25].

3) Data Segmentation: In this part, we are identifying the segments of the preprocessed data that contain useful information about the activities to be detected. Each segment has a duration and is defined by the start time and the end time within the time-series dataset. It is hard to segment the dataset ideally because in daily life there is not always a concrete pause between two consequent activities and the boundaries are tough to define.

A widely used data segmentation method is the sliding time window one. In the sliding window approach, a window of a specific width is moved across the data and defines the start and the end of the segment. The higher the width, the higher the lag, since the AR system has to "wait" for a specific amount of time before a segment is full. The optimal size of the window is not known, is inferred during the testing phase and can influence the performance of the system [26]. Another variable of the segmentation phase is the step size. While a small step size will increase the number of segments and create some that potentially better contain information about an activity, it will also increase the computing load of the application, since some entries should be computed more than once.

Another way to segment the preprocessed data may take advantage of the fact that different activities have different intensities and so the energy level of the IMU signals are distinctive. Other methods for data segmentation include the definition of either a rest position or a specific gesture [27].

Finally, segmentation can occur using external sources, like calendar entries with the start and duration of different activity sessions.

4) Feature Engineering: This step is about deriving features from the raw time series data. These features will form the feature vector that will be used for machine learning. Depending on the features, they can be either extracted on the segmented windows or the entire activity. The most widely used features in AR research consist of signal based ones. These can be either time domain based statistical ones like the mean, the median, the variance, etc. or frequency-domain features, like the energy in specific frequency bands [28]. The more features, the more training data are needed to classify activities more accurately, and the more computational resources are required for classification. On real-time and mobile systems, for example, it is imperative to use a minimum amount of features that do not significantly degrade the performance of the system. There are methods to automatically reduce the conventionality of the feature space by ranking and selecting the most important ones [29].

5) Training and Classification: Before predicting activities based on newly recorded sensor data, we must first train the selected model. For supervised learning, a training set is needed that consists of N entries of feature vectors X with corresponding output labels y. The selected model is defined by a parameter set θ, which during the training phase is learnt to minimise the classification error on the training set. Then for classification, the selected model with the trained parameter set θ is used to map a feature vector X to a set of confidence values y, that corresponds to the scores for every class that exists in the class set. With the scores vector and using either confidence thresholds or multiobjective optimisation techniques, a single prediction class is selected by the trained model.

Over the years, machine learning researchers have proposed a large number of algorithms. Many of those have already been used by the AR community to classify activities and to solve application-specific problems. The used methods include among others Hidden Markov Models [30], dynamic Bayesian networks [31], Decision Trees [32], Support Vector Machines [33], etc. There have also been researchers that have implemented deep learning neural networks techniques [34] for AR. So far no research derives one or a set of best machine learning algorithms for AR. The decision on that depends on the characteristics of the data. For example, if the dataset consists of many observations in a low dimensional space, then even the k-Nearest Neighbours (kNN) method may perform sufficiently well, while in other cases a more complex model may be necessary. There are also techniques to fuse the results of multiple classifiers to create a model that performs better than the submodels that it consists of [35].

6) Performance Evaluation: The last step when training a model for AR is to evaluate its performance [36]. Various performance metrics can be used for evaluation, metrics such as True Positive (TP) and False Positive (FP) rate, precision, accuracy, F-scores and recall. The confusion matrix summarises how many instances of the training set, either in absolute terms or a percentage, were correctly classified and how many were not. The Receiver Operating Characteristic (ROC) and the Precision-Recall (PR) curves also provide an insightful view on the predictive performance of the model.

The evaluation scheme that is typically used to evaluate AR models is the k-fold cross-validation scheme. According to this, the training set is partitioned into multiple folds. All of them but one are used to train the model and then the remaining fold is used to evaluate the performance of the model. This procedure is repeated until all folds have been left out once. The performance results are then averaged to evaluate the performance of the predicted model. A hold-out strategy can also be used, where the model is trained once on a percentage of the available dataset and then evaluated on the rest of it.

III. Experimental Design

We conducted an experiment in which we will showcase how different design decisions of an AR system compare and impact the overall performance of the system. We focused on recognising mobility activities from wrist-worn inertial sensors. Activity tracking applications running on smartphones use this kind of AR. At this point, however, we should note that in real life scenarios the problem of AR appears to be more demanding mostly because of the noise in the data and the variability of the way the activities can be performed.

A. Setup

We recorded the wrist movements of 11 male participants of ages 25-35 that performed 5 different mobility activities. The mobility activities include walking, running, idling in the office, going up and going down the stairs. Each participant was asked to perform each of the 5 activities for around 35 seconds for 4 times in total. With this, we guaranteed that the recorded dataset would be balanced. For the going up and going down the stairs activities, the 35 seconds target per recording could not always be satisfied, because the time of going up or down the available staircase varied according to the pace of each user. The resulting total dataset was roughly 128 minutes.

B. Sensors and Data

Wrist measurements were recorded using the inertial sensors of the Sony SmartWatch 3 watch running Wear OS. The watch was worn on the dominant hand of every participant which was the right for ten participants and the left for one of them. The available IMUs were a three-axis accelerometer and a three-axis gyroscope. All recordings were timestamped and the sampling rate for the accelerometer and the gyroscope was 150 Hz. During all recordings, the main researcher was observing and instructing the participants in order to correctly annotate with the ground truth and guarantee the cleanliness of the data. We used Matlab for feature extraction from the time series data and Python with the Scikit-learn module [37] for the machine learning experiments.

TABLE I
LIST OF EXTRACTED FEATURES PER SENSOR

Domain	Group	Features	No of features
Time	1	Mean Standard deviation	2
	2	Median Skewness Kurtosis 25th percentile 75th percentile Sq. sum of < 25th perc. Sq. sum of < 75th perc.	7
Frequency	3	Maximum frequency Sum of 5 Hz Number of peaks	9

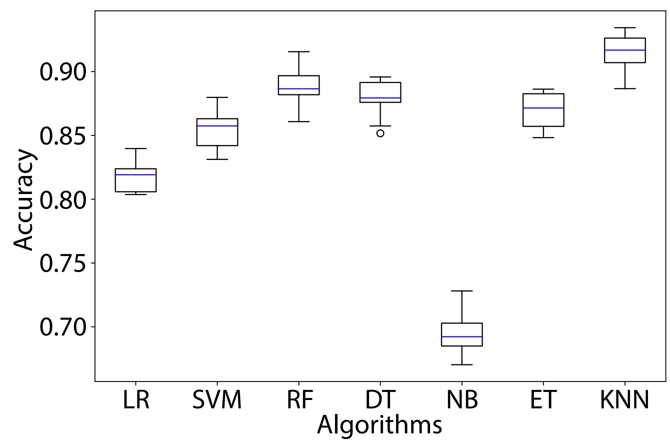

Fig. 2. Box plot of multiple person-independent classifiers.

IV. EVALUATION

As previously discussed, there are many components of an AR system that can be implemented in a variety of ways, and each such decision impacts the overall performance of the system. In this section, we will methodically evaluate our system with a plethora of choices regarding its parameters. We have searched up to a certain extent, also considering that all steps are interdependent and need to be configured jointly to achieve optimal results. This challenge becomes even more prevalent in real time AR systems that need to be regularly optimised based on user feedback and need to adapt continuously.

Since the IMUs that we have used did not provide a constant sampling rate throughout the recordings, the raw sensor data were resampled with a sampling frequency of 60 Hz. This frequency was selected for this study as it is higher than the 20 Hz commonly required to assess daily living [38] and also lower than what typical off-the-shelf IMU components can achieve.

Both time and frequency domain features were computed for both sensors. The time domain features include the mean, the standard deviation, the median, the skewness, the kurtosis, the 25th and the 75th percentile, and the squared sum of the components under the 25th and the 75th percentile. Those were derived from the resultant vector. For the frequency domain features, a Fast Fourier Transform (FFT) was performed after normalisation on the windows, and the features were computed per axis. Those features include the maximum frequency, the sum of heights of frequency components below 5 Hz and the number of peaks in the spectrum below 5 Hz, as it was noticed that most of the signal strength lied between 0-5 Hz. All the features extracted for this study are summarised and grouped in Table I. Group 1 includes two basic time domain features, Group 2 contains the rest of them, and Group 3 includes all the computed frequency domain ones.

To evaluate the performance of our systems, we split the available dataset into a training set (80%) and a test set (20%). The 10-fold cross-validation scheme was used on the training set in order to train the model, the performance of which was evaluated on the aforementioned test set.

A. Machine Learning Algorithms and Person Dependence

For the initial test, we fed the features of Group 1 of both the accelerometer and the gyroscope sensors, so four features in total, into multiple machine learning algorithms. The classifiers that we have evaluated are Logistic Regression (LR), Support Vector Machines (SVM), Random Forest (RF), Decision Tree (DT), Naive Bayes (NB), Extra Trees (ET) and k-Nearest Neighbours (kNN). The features were computed over a time window of 5s with a step size of 1s, so there was a 4s overlap between consecutive windows.

We tested both a user-independent and a user-dependent approach. Fig. 2 presents the box plot for all trained classifiers for the user-independent tests. Table II presents the confusion matrix for the kNN classifier, the best performing classifier for this case. We notice that the idling and the running activities achieve a perfect classification rate, even with a few features and a simple kNN classifier. We also notice that the classes that mostly suffer from misclassification are the going down and up ones, with misclassification rates of 15.1% and 13.3% respectively.

For the user-dependent approach, the results of all the 11 users for all selected classifiers are presented in Table III. As expected, personalised models have on average a better accuracy than person-independent ones, because they adapt to a specific person. There is even a case that a perfect classification rate was achieved (user 6, ET classifier). However, as in the case of user 3, when intra-class variations exist, that is when the user does not perform every activity in a consistent way, the model does not perform equally well even though it is a personalised one.

B. Sliding Time Window Size

One of the parameters selected during the feature extraction phase is the size of the time window. To investigate how the time window size affects the performance of our AR system, we swept the time window for the values $T_w = 0.5, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20s$. We run all the classifiers for all these cases using the same four features of group 1 of

TABLE II
CONFUSION MATRIX OF THE PERSON-INDEPENDENT KNN CLASSIFIER

		Predicted class				
		Down	Idle	Up	Walk	Run
True class	Down	84.9%	0%	10.7%	4.4%	0%
	Idle	0%	100%	0%	0%	0%
	Up	7.5%	0%	86.7%	5.8%	0%
	Walk	4%	0%	6.3%	89.7%	0%
	Run	0%	0%	0%	0%	100%

TABLE III
ACCURACY OF MULTIPLE PERSON-DEPENDENT CLASSIFIERS

User	Algorithms						
	LR	SVM	RF	DT	NB	ET	KNN
1	89.7%	96.4%	97%	97%	90.3%	98.2%	95.2%
2	95.5%	97.7%	97%	95.5%	93.9%	97%	97%
3	78.1%	83.2%	90.5%	85.4%	79.6%	86.1%	86.1%
4	87.3%	90.3%	89.6%	88.1%	79.9%	92.5%	91.8%
5	96.3%	97.8%	97.8%	94.9%	94.9%	95.6%	98.5%
6	97.7%	99.2%	99.2%	99.2%	96.9%	100%	99.2%
7	98.5%	99.3%	98.5%	98.5%	95.6%	98.5%	97.8%
8	96.3%	97%	94%	92.5%	91%	97%	97.8%
9	82.5%	81.7%	84.1%	86.5%	69%	83.3%	85.7%
10	92.6%	95.6%	93.3%	89.6%	81.5%	94.8%	91.9%
11	91.3%	96.4%	96.4%	94.2%	94.9%	95.7%	94.9%
Avg.	91.4%	94.1%	94.3%	92.9%	88%	94.4%	94.2%

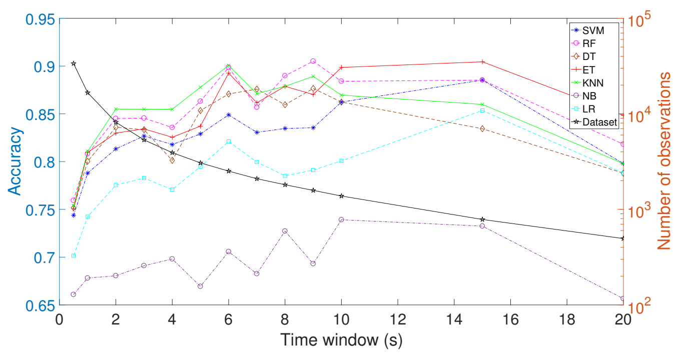

Fig. 3. Recognition performance for different time window sizes.

Table I for both sensors. On all tests, the step and thus the overlap between two consecutive windows was equal to half the time window size. For example, for the time window of 0.5s the selected step was 0.25s, for the time window of 1s the overlap was 0.5s, and so on. The results are presented in Fig. 3.

Naturally, as the time window increases, the available observations in the observations set that can be extracted from the raw data decrease. We notice that the time window size affects the recognition performance of the system, but up to a point. When the time window is tiny, it usually can not contain useful information about the executed activity.

C. Sensors and Features

In this part, we have experimented with multiple machine learning algorithms and with various combination of the input features. For each classifier, we tried all possible combinations of the groups of features of Table I from both available sensors.

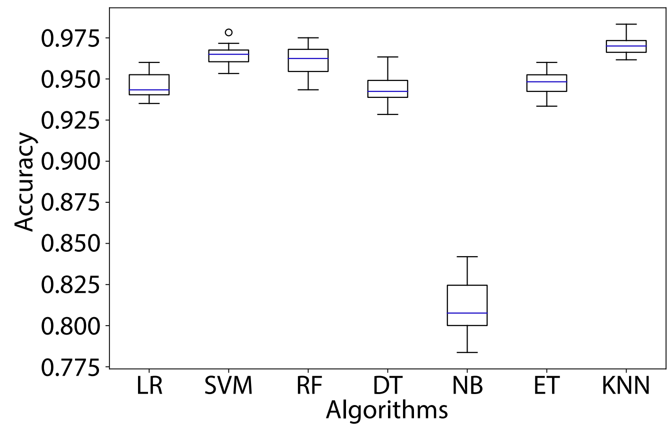

Fig. 4. Box plot of multiple person-independent classifiers using all available features from both sensors.

The results are presented in Table IV. Fig. 4 presents the box plot for all trained classifiers for the user-independent tests using all available features from both sensors.

The designer of an AR system usually has to decide on a single classifier. This choice is usually not only based on the performance of the model. Some classifiers tend to be more computationally complex and sometimes their superiority in performance may not justify the difference in their requirements concerning computing power. In our tests, for the last test case, the SVM and the kNN classifiers achieved predictive performances of 97.3% and 97.1%. However, the former is more computationally demanding than the latter and may not be suitable for a resource-restricted environment.

Moreover, the more features we are feeding a classifier, the more computationally complex it becomes. Therefore, it is a good practice to apply feature selection techniques to reduce the feature set to the most significant features. In our case, all automatic feature selection techniques ranked highest the features extracted from accelerometer data. This result was expected, given that according to the results of Table IV, accelerometer features performed better than the equivalent gyroscope ones in most cases when used independently.

V. CONCLUSION AND FUTURE WORK

This paper demonstrates different considerations for the design of a sensor-based activity recognition system. After discussing many parameters of such a system, we presented an experiment of a smartwatch-based AR system. Depending on the selection of the different parameters of the AR system, the predictive performance of it changes. After several tests, a predictive performance of up to 97.3% for a person-independent model was achieved.

As with any real-world application, there were some limitations regarding the collected dataset. For the study, participants were asked to perform the required activities in their work environment were conditions were not ideal. For example, when going up or down the staircase, the participants had to make a sharp turn to continue to the next chunk of stairs and

TABLE IV
PREDICTIVE PERFORANCES OF VARIOUS ALGORITHMS FOR VARIOUS INPUT FEATURE COMBINATIONS

Feature groups	Sensors	Algorithms						
		LR	SVM	RF	DT	NB	ET	KNN
1	A	79.7%	80.2%	83.3%	83.7%	71.2%	81.9%	83.4%
	G	54%	53.9%	61.9%	61.9%	51.5%	60%	60.6%
	A+G	81.2%	84.3%	87.6%	86.8%	68.8%	85.8%	90.7%
2	A	84%	84.9%	85.8%	85.6%	73%	83.8%	87%
	G	62.8%	65.2%	73.6%	72.2%	53.9%	70.1%	75%
	A+G	88.1%	89.3%	89.5%	90.3%	70.4%	88%	93.9%
3	A	77%	80.6%	89.1%	91.4%	76.7%	87.4%	88.8%
	G	75.5%	82.9%	91.1%	89.2%	72%	88.1%	90.3%
	A+G	89%	92.8%	93.3%	95.1%	80%	94.1%	93.7%
1+2	A	83.7%	84.6%	86.3%	86.2%	74.4%	84.4%	87.7%
	G	63.1%	66%	75%	71.3%	53.6%	71.7%	77%
	A+G	87.8%	89.8%	90%	90.7%	70.6%	89.8%	94.3%
1+3	A	86.4%	90.6%	93.7%	94.7%	79.4%	91.5%	92.5%
	G	78.4%	84.8%	91.9%	91.1%	70.4%	89%	91.5%
	A+G	92.6%	96.4%	95.3%	94.9%	80.6%	94.8%	95.5%
2+3	A	90%	92.8%	92.4%	94%	82%	90.5%	93.9%
	G	82%	88.2%	92.8%	91%	70.8%	90.4%	93.6%
	A+G	94.9%	97.3%	94.7%	94.5%	81.7%	94.7%	96.3%
1+2+3	A	90%	92.8%	93.5%	93.5%	81.3%	90.9%	94.1%
	G	82.6%	88.6%	93.1%	90.9%	70.3%	90%	94%
	A+G	95.2%	97.3%	95.5%	95%	81.3%	95.3%	97.1%

some time up to 1s was spent walking during this transition. Moreover, the participants were all men, and of a small age range, so the data are skewed regarding these aspects.

To extend this study, deep learning techniques will be evaluated. So far the analysis was performed offline on a desktop computer. It would be of interest to evaluate how different choices would affect a real-time system running on a smart-watch and what performance can be achieved having restricted power and computing resources.

REFERENCES

[1] J. Pansiot, D. Stoyanov, D. McIlwraith, B. P. Lo, and G.-Z. Yang, "Ambient and wearable sensor fusion for activity recognition in healthcare monitoring systems," in *4th international workshop on wearable and implantable body sensor networks (BSN 2007)*. Springer, 2007, pp. 208–212.

[2] F. Kosmalla, F. Daiber, and A. Krüger, "Climbsense: Automatic climbing route recognition using wrist-worn inertia measurement units," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 2015, pp. 2033–2042.

[3] K. K. Biswas and S. K. Basu, "Gesture recognition using microsoft kinect®," in *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*. IEEE, 2011, pp. 100–103.

[4] T. Stiefmeier, D. Roggen, G. Ogris, P. Lukowicz, and G. Tröster, "Wearable activity tracking in car manufacturing," *IEEE Pervasive Computing*, vol. 7, no. 2, 2008.

[5] J. C. Lee, "Hacking the nintendo wii remote," *IEEE pervasive computing*, vol. 7, no. 3, 2008.

[6] K. Tanaka, J. Parker, G. Baradoy, D. Sheehan, J. R. Holash, and L. Katz, "A comparison of exergaming interfaces for use in rehabilitation programs and research," *Loading...*, vol. 6, no. 9, 2012.

[7] J. Egger, M. Gall, J. Wallner, P. Boechat, A. Hann, X. Li, X. Chen, and D. Schmalstieg, "Htc vive mevislab integration via openvr for medical applications," *PloS one*, vol. 12, no. 3, p. e0173972, 2017.

[8] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, "Sensor-based activity recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 42, no. 6, pp. 790–808, 2012.

[9] R. Bodor, B. Jackson, and N. Papanikolopoulos, "Vision-based human tracking and activity recognition," in *Proc. of the 11th Mediterranean Conf. on Control and Automation*, vol. 1, 2003.

[10] X. Wang and Q. Ji, "A hierarchical context model for event recognition in surveillance video," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 2561–2568.

[11] A. Baak, M. Müller, G. Bharaj, H.-P. Seidel, and C. Theobalt, "A data-driven approach for real-time full body pose reconstruction from a depth camera," in *Consumer Depth Cameras for Computer Vision*. Springer, 2013, pp. 71–98.

[12] L. Chen, C. D. Nugent, and H. Wang, "A knowledge-driven approach to activity recognition in smart homes," *IEEE Transactions on Knowledge and Data Engineering*, vol. 24, no. 6, pp. 961–974, 2012.

[13] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, p. 33, 2014.

[14] T. Huynh, M. Fritz, and B. Schiele, "Discovery of activity patterns using topic models," in *Proceedings of the 10th international conference on Ubiquitous computing*. ACM, 2008, pp. 10–19.

[15] M. Längkvist, L. Karlsson, and A. Loutfi, "A review of unsupervised feature learning and deep learning for time-series modeling," *Pattern Recognition Letters*, vol. 42, pp. 11–24, 2014.

[16] B. E. Ainsworth, W. L. Haskell, A. S. Leon, J. D. Jacobs, H. J. Montoye, J. F. Sallis, and J. R. Paffenbarger, "Compendium of physical activities: classification of energy costs of human physical activities." *Medicine and science in sports and exercise*, vol. 25, no. 1, pp. 71–80, 1993.

[17] M. Kranz, A. MöLler, N. Hammerla, S. Diewald, T. PlöTz, P. Olivier, and L. Roalter, "The mobile fitness coach: Towards individualized skill assessment using personalized mobile devices," *Pervasive and Mobile Computing*, vol. 9, no. 2, pp. 203–215, 2013.

[18] A. FernáNdez, V. LóPez, M. Galar, M. J. Del Jesus, and F. Herrera, "Analysing the classification of imbalanced data-sets with multiple classes: Binarization techniques and ad-hoc approaches," *Knowledge-based systems*, vol. 42, pp. 97–110, 2013.

[19] M. Csikszentmihalyi and R. Larson, "Validity and reliability of the experience-sampling method," in *Flow and the foundations of positive psychology*. Springer, 2014, pp. 35–54.

[20] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *ESANN*, 2013.

[21] A. Wang, G. Chen, J. Yang, S. Zhao, and C.-Y. Chang, "A comparative study on human activity recognition using inertial sensors in a smartphone," *IEEE Sensors Journal*, vol. 16, no. 11, pp. 4566–4578, 2016.

[22] A. Cereatti, D. Trojaniello, and U. Della Croce, "Accurately measuring human movement using magneto-inertial sensors: techniques and challenges," in *Inertial Sensors and Systems (ISISS), 2015 IEEE International Symposium on*. IEEE, 2015, pp. 1–4.

[23] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015.

[24] H. Lu, W. Pan, N. D. Lane, T. Choudhury, and A. T. Campbell, "Soundsense: scalable sound sensing for people-centric applications on mobile phones," in *Proceedings of the 7th international conference on Mobile systems, applications, and services*. ACM, 2009, pp. 165–178.

[25] Z. S. Abdallah, M. M. Gaber, B. Srinivasan, and S. Krishnaswamy, "Activity recognition with evolving data streams: A review," *ACM Computing Surveys (CSUR)*, vol. 51, no. 4, p. 71, 2018.

[26] G. Okeyo, L. Chen, H. Wang, and R. Sterritt, "Dynamic sensor data segmentation for real-time knowledge-driven activity recognition," *Pervasive and Mobile Computing*, vol. 10, pp. 155–172, 2014.

[27] A. Zinnen, C. Wojek, and B. Schiele, "Multi activity recognition based on bodymodel-derived primitives," in *International Symposium on Location-and Context-Awareness*. Springer, 2009, pp. 1–18.

[28] A. Dutta, O. Ma, M. P. Buman, and D. W. Bliss, "Learning approach for classification of geneactiv accelerometer data for unique activity identification," in *Wearable and Implantable Body Sensor Networks (BSN), 2016 IEEE 13th International Conference on*. IEEE, 2016, pp. 359–364.

[29] S. Pirttikangas, K. Fujinami, and T. Nakajima, "Feature selection and activity recognition from wearable sensors," in *International Symposium on Ubiquitious Computing Systems*. Springer, 2006, pp. 516–527.

[30] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.

[31] K. P. Murphy and S. Russell, "Dynamic bayesian networks: representation, inference and learning," 2002.

[32] L. Bao and S. S. Intille, "Activity recognition from user-annotated acceleration data," in *International Conference on Pervasive Computing*. Springer, 2004, pp. 1–17.

[33] T. Huynh, U. Blanke, and B. Schiele, "Scalable recognition of daily activities with wearable sensors," in *LoCA*, vol. 7, 2007, pp. 50–67.

[34] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert Systems with Applications*, vol. 59, pp. 235–244, 2016.

[35] L. I. Kuncheva, "A theoretical study on six classifier fusion strategies," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 24, no. 2, pp. 281–286, 2002.

[36] J. A. Ward, P. Lukowicz, and H. W. Gellersen, "Performance metrics for activity recognition," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 1, p. 6, 2011.

[37] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg *et al.*, "Scikit-learn: Machine learning in python," *Journal of machine learning research*, vol. 12, no. Oct, pp. 2825–2830, 2011.

[38] C. V. Bouten, K. T. Koekkoek, M. Verduin, R. Kodde, and J. D. Janssen, "A triaxial accelerometer and portable data processing unit for the assessment of daily physical activity," *IEEE Transactions on Biomedical Engineering*, vol. 44, no. 3, pp. 136–147, 1997.