Towards paó: Wearable Wellness Companion

Philipp Durnay (4622227)
Sergio Soto Munoz Ledo (4627288)
Ozan Dogu Tuna (4198034)

***Abstract*— A person's posture may have a strong correlation her or his physical and mental state. However, being aware of one's own posture at any given time proves to be difficult in everyday life. As wearable technology is advancing, computer based systems can be used to increase multiple aspects of body awareness, including posture. This work investigates the design space of a wearable posture tracking device: paó. The software for an embedded system using a 9-axis motion sensor is developed together with an accompanying smartphone application. Different posture/activity detection algorithms are evaluated on a collected dataset and implemented in the smartphone app. In several experiments on this offline dataset an activity detection accuracy of 83% is achieved, together with a 92% accuracy of posture detection. Different design possibilities for the smartphone application are investigated in context of invoking behavioral change in the user.**

I. INTRODUCTION

Mind and body awareness relates to one's ability to strip away any physical and mental interferences; lack of which often surface in poor body habits and impaired concentration. This concept is more than a simple understanding of the parts that make up one's body and where they are located. It is also about understanding the condition of body and mind at a given time, and being aware of what they can do together. In this sense, mind and body awareness is strongly related to mindfulness, as mindfulness is about "bringing one's attention to present experience on a "moment-to-moment" basis. There is a vast amount of research that suggests the practice of mindfulness is strongly correlated with one's well-being and perceived health. One can argue that mind and body awareness is an essential part of true mindfulness.

Wearable technology space is getting more and more crowded with devices that provide information to the user about different aspects of body condition such as activity and fitness level, heart rate, etc. However, most of these devices treat the data as independent silos, lacking a cohesive approach for determining one's true wellbeing. Moreover, almost none of them put posture into account, despite a number of recent studies suggest strong correlation between people's posture and their mood, body and mind wellbeing as well as self confidence[1]. Tracking only fitness or activity without considering posture and mood is at least open to criticism for resolving on insufficient data. A holistic approach to body and mind wellness including mindfulness, self confidence and body awareness may lead to tangible health benefits and better user satisfaction. This paper is based on the vision of developing a product that would enable such a holistic approach: paó.

The main value addition of paó comes from its attempt to fit accurate posture detection into an attractive package that the users would feel comfortable wearing on a daily basis. Posture, combined with commodity biometric data such as activity levels and heartrate monitoring, can enable the device to have important functionality that is uncommon in what is currently offered in the market. Among such functionality are providing:

- corrective feedback to invoke behavioral change and help with posture related back issues,
- a way for users to relate different indicators of mood and self esteem and help improving their condition,
- assistance in meditation and other physical activities where a person's posture matters.

The system consists of two main parts: a wearable device running embedded software and a smartphone running a companion application. Ultimately, these two pieces of the system should work seamlessly to provide the functionality described above. The embedded software in the wearable device shall be responsible for data acquisition and realtime posture feedback, whereas the companion application shall be there to store data history and use it in smart ways (visualization and coaching) to invoke behavioral change, as well as provide user friendly ways of device configuration, calibration and training.

The scope of this paper is a feasibility study for whether satisfactory performance and robustness can be achieved by implementing different pattern recognition methods for posture detection of a user. For this purpose, the following method has been taken:

1) A simple data collection and pre-processing algorithm is implemented on a wearable size device containing a 9-axis motion sensor and a micro controller.
2) Different pattern recognition training and classification methods have been implemented into a smartphone application for comparison of their accuracy and computational load in identifying posture.
3) A simple smartphone app interface was designed for real-time data visualization, data history visualization and simpler usage of classifier training and debugging capabilities.

II. RELATED WORK

Incorrect posture can lead to strain on musculoskeletal system, particularly on the spinal column, which can result a host of side effects on one's body such as strain on muscles, herniated disc, pinched nerve pain, depression, stress, digestive issue, breathing problem, back and neck pain, shoulder pain, headache, slouching, rounded shoulders, hunchback, bent knees, tilting head forward, locked knees, arching of low belly and many more[6]. Recent studies have shown seating in a wrong position over extended periods cause new health problems, but also worsen the ones that an individual already may have[7]. There is ongoing research that tries to determine whether the so called good postures actually provide a clinical advantage[8].

Research also shows that body postures can influence one's attitude both via association[2] and self perception[3]. It can also impact one's persuasiveness by influencing the likelihood of having positive thoughts about the subject[4]. According to one study conducted in 1983[5], not only the direction of thinking, but mere amount of thinking and elaboration on a subject can be influenced by the posture taken during the effort.

In the past decades, a growing body of research on embodiment has demonstrated that not only bodily sensations, but also bodily postures, gestures and expressions are inherent components of emotional experience and they influence the evaluation of surroundings, as well as memory recall. To provide some examples: several studies suggest that posture is not only an attribute but something that can be trained to get better [9] [10]. The effect of posture based can be so prominent that exercising high power pose and posture prior to high-stakes social evaluation (such as job interviews) can influence the outcome[11].

One of the latest approaches to posture recognition/differentiation is using image pattern recognition/extraction techniques[12]. The results show high accuracy, but the methods require use of multiple cameras, thus imposing high costs and reducing convenience. Realtime posture recognition and monitoring can also be accomplished through the use of wearable motion sensors. Such a system could potentially allow the user to live and work normally with minimal interference, with data being stored and transmitted automatically over wireless solutions such as Bluetooth. There are commercially available products that implement this sort of idea (Lumo Lift, Upright, Prana to name a few), but one can easily argue that a solution that gained wide market traction has not been achieved yet, despite obvious interest from customers and potential early adopters.

III. REQUIREMENTS AND CHALLENGES

paó is designed to help users conduct their everyday business while unobtrusively detecting and logging their posture. Although functionality enables a variety of use cases, realtime posture detection and corrective feedback is a tentpole feature. Specifically, paó detects whether the user has good posture at any moment, and reports a prolonged wrong posture via a gentle buzz, where each buzz is taking increasing time intervals and 'snooze' after a point to stop distracting the user. paó needs to meet the following requirements:

1) To ensure wide adoption in practice, the device should have a high level of accuracy with low variance, classifying right posture from wrong given different circumstances and activities. It is easy for the user to dismiss the added value of the device if he/she encounters too many false positives (and false negatives) in a short amount of time, even if the device statistically may demonstrate have high degree of accuracy.
2) As posture is important throughout the waking life of a user, it should be unobtrusive to day by day activities and should not distract user to the point of creating friction to its adoption into daily life.
3) It needs to detect the posture while the user is sitting, standing or moving in a robust way, i.e, across different body shapes, different garments that it is attached to and different positions of attachment around the upper body.
4) Calibration and training process of system should be intuitive and require the minimum amount of effort/time. If possible, the device should be able to operate with a usable level of accuracy out of the box, without any specific training period, and increase its accuracy over time with data generated collectively by (possibly all) paó users.

First, paó must be able to detect changes in mode (not moving, moving) and posture, using accelerometer and gyroscope readings in the presence of interference from the movement of the user as well as disturbance from movement of the the garment that the device is attached to. Moreover, there can be subtle differences between a right posture in one mode (sitting) and wrong posture in another mode (standing). Also, a leaning motion to the front to tie someone's shoelaces or pick up an object from the ground can be identified as wrong posture during movement. As a result, the user's posture may be falsely classified.

Additionally, it is challenging to design the training process for posture due to several reasons. The notion of good posture and bad posture do not have rigorous definitions. Ground truth could be collected making user actively label modes and postures, but that could easily mean too much training effort put on the user's side.

Moreover, the data collected is highly dependent on how the device is worn and what it is worn on. A set of training data relevant for one clothing and/or one attachment point can be different enough from another one to invalidate an important chunk of training data. The attachment point can change throughout the day, even throughout an activity, as the user may feel to change clothes or simply decide to wear paó in a different way.

IV. SYSTEM DESIGN

paó is a wearable sensing device with a companion app, that can accurately track a user's posture and activity. Specifically, it detects whether the user has correct posture at a given point in time, which is an enabling primitive for various applications of body and mind awareness as well as posture related activity coaching. To this end, paó senses the orientation of a person's upper body using a 9-axis motion sensor equipped wearable device on a user's torso. A series of algorithms running on the device collect relevant information from the sensors and use a classifier for decision of posture state. The device can do posture detection standalone, however the experience is enhanced by use of a companion app. Figure 3 shows the high level system overview of paó.

From a pure technical point of view, the most interesting parts of paó can be categorized into three (somewhat functional) categories:

- sensor data acquisition and preprocessing the relevant part of the data for classification,
- pattern recognition based posture detection algorithms,
- companion app for training, debugging and preliminary data visualization.

The following subsections will investigate these three categories in detail.

A. *Sensor Data Acquisition and Preprocessing*

The 9-axis motion sensor in paó is at the heart of posture detection functionality. Ideally, a mechanical system as complex as human spinal column can be represented with high accuracy using a number of sensors located at different parts of the body. However, an important design driver for any wearable including paó is unobtrusiveness. A wearable device should be designed around the idea of easy wearability. Moreover, multiple sensors mean increased cost and energy overhead. paó aims to get around these challenges by tracking posture using only one sensor. Nevertheless, an accurate representation of the orientation of the user's upper body is a serious challenge.

Invensense MPU9250 that is used as paó's sensor has a number of capabilities that are enabled by 9 axis sensor fusion. These capabilities can be used in different combinations to obtain as accurate representation of posture as possible. More specifically, one can get raw sensor readings of accelerometer, gyroscope and magnetometer as well as more complex geometric information supplied by an on board Digital Motion Processor (DMP) of the sensor package. The most convenient way of accessing DMP data is through the Embedded Motion Processor Library (eMPL) supplied by Invensense; an embedded software stack of the sensor driver layer that leverages the features of the DMP and is supported by MPU9250. Via eMPL, many of the features of the hardware and the on board Digital Motion Processor (DMP) are encapsulated into modular APIs which can be used and referenced.

The device orientation can be represented in many different mathematical forms such as the Euler angles, quaternions, rotation matrix, etc. These forms are also convertible between one and another. The most useful among them for the purpose of posture detection are the Euler angles.

The Euler angles are three angles introduced by Leonard Euler to describe the orientation of a rigid body. Euler angles are commonly used in navigation for aircraft orientation and robotics for robotic arm movement. Euler's rotation theorem tells that any orientation can be described as three consecutive rotations. Figure 2 illustrates the geometric definition of these rotations in angles.

Assuming the rigid body in question is a person's upper body, the most relevant one among these angles is the one that is defined with respect to the axis that is parallel to the gravitational vector, i.e θ. This is logical, as posture is mostly associated with a relatively straight spinal column, where a perfectly straight position would have an angle θ close to zero. Of course, in reality, this may seem as an oversimplified definition. However, the importance of θ as an input parameter for posture classification is obvious. However, eMPL only provides quaternion representation directly from DMP, whereas APIs that provide other representations are implemented on the driver layer, executed on the microprocessor. A design decision has been taken to implement custom code for quaternion to Euler angle transformations, rather than relying on proprietary code. Conversion from quaternions to Euler angles are governed by the following equations:

$$\phi = \arctan\big(q_1q_3 + q_2q_4, q_1q_4 - q_2q_3\big)$$
$$\theta = \arccos\big(-q_1^2 - q_2^2 + q_3^2 + q_4^2\big)$$
$$\psi = \arctan\big(q_1q_3 - q_2q_4, q_2q_3 + q_1q_4\big)$$

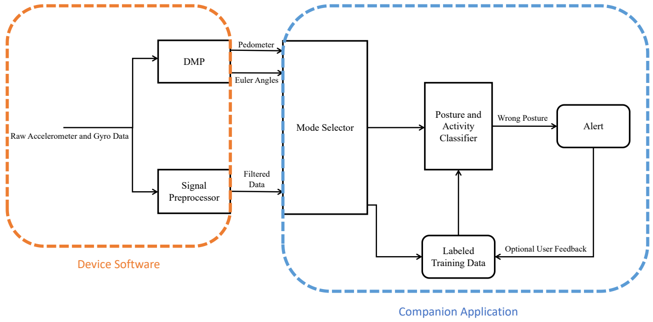
Another perk of using the eMPL is that, through DMP, one can access algorithms that calculate pedometer and tap gesture obtained using sensor data. Although custom algorithms could be written for both functionalities, exploiting DMP for them are chosen as the efficient path, keeping the main focus of paó as posture detection.

B. *Posture Classifier*

Once the relevant data is extracted from the sensors, it is fed further to the classifier to identify the posture at a given time . The current status of implementation is that the posture classifier is part of the companion app instead of the device software. The device of paó is intended to be standalone, thus the final implementation of the posture classifier will be on the device. However, temporarily having the posture classifier in the app has a number of advantages:

1) Evaluation between a number of different pattern recognition algorithms is essential before deciding on which one to use on device software. This evaluation involves a rapid prototyping approach which would take more time and effort using an embedded toolchain. Prototyping the posture classifier in an application environment is much more efficient.
2) Implementation of certain pattern recognition algorithms are time consuming for embedded environments. High level languages offer a more convenient way of programming.
3) The computational expensive part of a classifier is the training phase. Having the respective algorithms implemented in the application allows training the classifier on the phone and sending only the tuned parameters to the device. Thus the training can be

Fig. 1. System overview for paó.



done faster and energy of the device can be saved for classification.

4) The application provides a fast and developer friendly debugging environment, as dedicated interfaces for this function can be built with relative ease.

As indicated above, evaluation of different classification algorithms in terms of accuracy and power efficiency is an important part of this study. In a first step the classification problem is further defined. After having a clear understanding of what needs to be classified, suitable algorithms can be chosen and evaluated.

The goal of paó is to give users valuable feedback about their posture and improve their well-being. Hence, knowing the posture at any given activity is a fundamental requirement. On top of that knowledge about the general user activity can be exploited to further advise the user. However, implementing a whole activity recognition system would be beyond the scope of this work. Thus the following six classes are defined:

1) sitting with good posture,
2) sitting with bad posture,
3) standing with good posture,
4) standing with bad posture,
5) walking with good posture,
6) walking with bad posture.

Most pattern recognition algorithms have the implicit assumption that the measured samples are independently identical distributed (iid). This means that the probability of observing a certain sample at a given time is not related to the observation of another sample at another time. For paó this would mean that after observing "Sitting" at timestep t it is equally likely to observe "Walking","Standing" and "Sitting" at timestep t+1. Obviously, this assumption is not met (assuming that the sampling period is smaller than a couple of seconds).

Generally classification methods are quite robust against violations of this assumption so they can still be used in this application. However, this insight can be further exploited in the further design and evaluation of the posture classifier.

A mathematical model that generally works quite well on sequential (non-iid) data is the (Hidden) Markov Model [13]. Here the relation between observations is estimated based on example data and formulated as a Markov Chain. However, this model requires to estimate the transition probabilities between the different activities. Without a very big amount of data this model is unlikely to perform well in practice and is therefore not further investigated. Instead more simplistic approaches that exploit the fact that users don't change their posture at a high pace are evaluated. An example is the combination of sequential outputs of classifiers in a (weighed) majority vote.

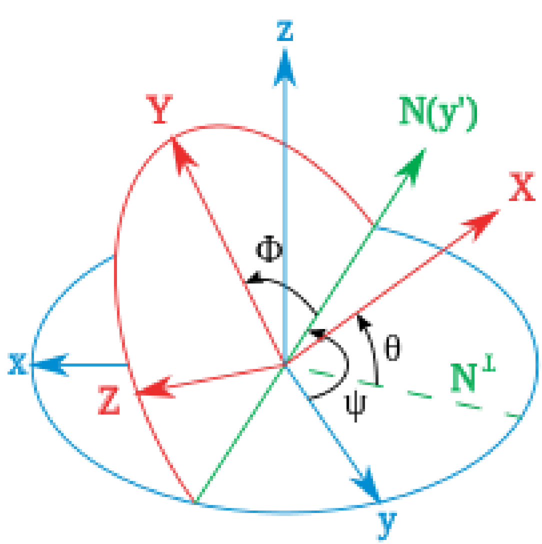
To this end, a number of supervised learning methods are evaluated for the posture classifier. These, along with a small description of each algorithm's concept are provided below [14]:

- **k-Nearest Neighbors:** k-Nearest Neighbors is a one of the simplest pattern recognition algorithms. A set of labeled examples is stored in the training phase. Newly classified samples are assigned to their closest neighbors in the training set. Different distance measures can be used and the amount of neighbors taken into account can be varied. These parameters will be chosen based on the evaluation.
  The kNN classifier can be easily implemented and form a non-linear and thus more flexible decision boundary. However, in the simple implementation the distance to every training object has to be calculated when classifying new samples. Thus the decision time gets prohibitive when having a lot of samples in the training set. Methods to reduce the classification time are available but require more implementation effort.
- **Decision Tree:** A decision tree is an acyclic connected graph, where internal nodes represent a "test" on an attribute, each branch from the node represents the outcome of the test, and each leaf node represents a decision taken after taking all test nodes into account. In an informal way a decision tree can be described as a set of questions. At each node a binary question is asked until a leaf is reached. The sample is then assigned depending on the amount of training samples in that specific leaf. The advantage of decision trees is their transparency. In contrast to most other classification methods a human can actually see what the classifier learned and why it decided in a way it decided.

Additionally are decision trees computationally cheap at classification time. A problem of decision tree is that they easily overfit to the training data. This can be limited to some extend by limiting the maximum tree size or by using random forests.

- **Quadratic Discriminant Analysis:** Quadratic discriminant analysis belongs to the category of Bayes classifiers. Each class is assumed to have a Gaussian distribution. The covariance matrix, mean and prior of these distributions is estimated at training time. With Bayes rule the class posterior probability for each class can be evaluated and the sample can be assigned to the highest one. Quadratic Discriminant Analysis can work well when the data is Gaussian distributed. However, a problem is the amount of parameters that need to be tuned. For an l-dimensional states and n classes $2n + n * l^2$ parameters need to be estimated. This can be difficult with limited amount of training data.
- **Artificial Neural Network:** Artificial neural networks are pattern recognition systems inspired by the biological neural networks in human brain. Neural networks continuously learn and improve their performance to do tasks by considering examples, generally without task-specific programming.

Fig. 2. A geometric representation of Euler angles.



Artificial neural networks tend to work quite well on a lot of classification problems. However, the big amount of parameters that need to be tuned usually requires a lot of training data. Thus the training time is long and also the classification time can - depending on the network size- be prohibitive for and embedded device. Also the implementation is cumbersome. Next to the basic structure of interacting neurons the implementation of optimization algorithms like stochastic gradient descent is required.
- **Linear Support Vector Machines** Support Vector Machine is linear classifier that tries to find the hyperplane that maximized the margin between two classes. It does so by using samples of the individual classes as so called ”support vectors”. SVMs are widely used in text and image classification as they tend to achieve good performance even when the feature dimension is much higher than the number of observations. Challenging however especially for this application is the large amount of time and memory needed for training.
- **AdaBoost** AdaBoost is a boosting method that can improve the classification performance of individual learners. In this application one level decision trees (decision stumps) are boosted. This leads to a classifier that has low memory and CPU requirements during training and classification.

Another evaluation topic for the posture classifier is the data representation. In total the measurements of accelerometer, gyroscope and DMP are available. These can be further processed e.g by calculating the max/min values or doing a Fourier transformation of a certain window. However, more features does not necessarily mean an increase of classification performance. As more features are used, more parameters need to be tuned and dimensionality of the feature space increases. If the training data is not large enough this can lead to a decrease in performance at some point. To counteract this issue feature selection or dimension reduction can be applied. Different data representations are evaluated in section V.

The goal of paó is to alert the user when being in a wrong posture. However, the device should not respond to abrupt movements or short periods of wrong posture. Therefore data from the sensors can be filtered for noise and outliers. Additionally several sequential outputs of the classifier can be combined for postprocessing. Different pre- and postprocessing methods evaluated as further described in section V The described algorithms are mostly evaluated offline using the sklearn framework in python. However, as the algorithms also need to be evaluated for energy consumption, training time and performance in practice. Several of the most potential algorithms are implemented in the smartphone application. These are the nearest-neighbor classifier, decision trees, AdaBoost and quadratic discriminant analysis. Although neural networks and linear svms are likely to perform well, their energy consumption and memory/cpu requirements is expected to be prohibitive. Also their implementation is quite cumbersome. Thats why these algorithms will only be implemented if they prove to work substantially better than other methods.

C. Companion Application

As explained in section IV-B, there are a number of advantages in taking development and debugging related functionality on the application rather than on device or on a cross-compiler platform. This observation resulted initially taking a ’developer friendly’ approach rather than a ’consumer friendly approach’ in the app’s design and functionality.

Fig. 3. Views from the companion app. From left to right: main, configuration, debug and training views.

The companion smartphone app was developed for iOS 11, using the Xcode IDE and all graphics and assets were drawn and programmed in PaintCode. It is composed of multiple views through which the user navigates. The views designed are as follows:

- **Main View:** Main view contains constantly updated posture and activity recognition data from the currently active classifier type in a representation of whether the user is identified as sitting, standing or moving, as well as whether his/her posture is right at any given moment and pedometer count. A horizontal bar at the top represents the real-time posture correctness identified by the classifier. Four circular bars represent the completeness of a desired goal set in the configuration. In the case of the 3 activities (sitting, standing and moving), this represents a percentage of time spent in a good posture over the total amount of time spent performing the corresponding activity. In the case of step counter, it represents the percentage of steps walked in comparison to the total step goal set. Moreover, a plot of posture recognition part of the classification is provided, where it is possible to reach historical data by 'sliding through' this plot. Finally, two buttons provide a 'Snooze' functionality so the user can silence the vibration alerts from pa, and a 'Calibrate' function that initializes an offset with which the DMP sensor values are adjusted to an initial position.
- **Configuration View:** Configuration view provides functionality of setting different parameters of the system, and for now it is used for high level goal settings configuration. These goals represent the percentage of time the user intends to achieve in a good posture for each of the activities. In the future, more advanced configuration options for this view are in the pipeline, such as the classifier that is used, signal preprocessing window length, etc.
- **Debug View:** Debug view provides developers an environment that enables them to check realtime raw sensor and DMP readings along with terminal output of the embedded software running on the device in real time. Moreover, using this view, it is possible to ping the device for connection as well as send raw messages to it in the form of hex codes. Finally, the developers can export training data and logs by using the respective button.
- **Training View:** Training view takes the application into training mode. In this mode, the six possible labels are presented in buttons where a user can 'label' the current state that he/she is in to the sensor data collected at that given moment. This labeled data is then used to train posture classifiers.

V. EVALUATION

In this section the system performance in terms of classification accuracy is evaluated. Data of different users are collected and performances of different posture detection methods are compared offline. This study gives an insight on the best performing algorithms and configuration parameters. However, the amount of users that are available to collect data is limited for the scope of this project. Hence, any performance evaluated offline may not be representative enough for a final design decision. Also it is not really possible to evaluate the usability and energy consumption offline. That's why the most potential posture detection methods are implemented in the application to be tested in everyday use.

Next to the different posture detection methods described in subsection IV-B, further questions shall be answered in this experiment. In total this leads to the following:

1) What is the overall potential of posture detection using pattern recognition algorithms in combination with data attainable from 9-axis motion sensor?
2) Is it possible to train a classifier for multiple users so the system works without further calibration and training for the individual user?
3) What is an appropriate representation of the measured data?
4) What is the best performing classification algorithm?
5) Can filtering further improve classification performance?

The data for the experiment is collected from 3 users spending 3 minutes on each of activities defined in subsection IV-B. The performance of the classification methods is evaluated based on the classification error in a 20-folded cross-validation. The results of the different experiments are summarized in Table I and further described below.

A. *Experiment I*

In a first experiment different pattern recognition algorithms are run on the raw data set without any preprocessing applied. Unfortunately, the results show that most classifiers perform not very good on the dataset. Accuracies are in the range of 50% for most classifiers and the variance of the results is quite high. This means the results are not very accurate and vary a lot depending on what kind of data is used for training.

However, decent performance could be obtained by the six nearest neighbors classifier (6NN) and the neural network. Especially the 6NN achieves an accuracy of 81% and has comparatively low variance. Looking at the confusion matrix gives further insights on how the different samples were classified. A confusion matrix displays the true data that is fed in as rows, and the columns as the classifier output, normalized to 1. Thus, a perfect classification would mean an identity matrix, and any deviation from that indicates the level of 'confusion' that the classifier has.

Figure 4 displays the confusion matrix of the 6NN classifier. The results show that a wrong posture during sitting can be classified correctly in 94% of the cases. Also the detection of wrong postures while standing works with 89%. Furthermore, it seems the classifier can correctly identify the user as moving, but has the biggest level of confusion to decide on right or wrong posture during moving activity state.

Fig. 4. Confusion matrix for 6-NN Classifier.

	MovOk	StandOk	StandNok	MovNok	SitOk	SitNok
MovOk	0.79	0.06	0.05	0.07	0.03	0.0
StandOk	0.01	0.98	0.01	0.0	0.0	0.0
StandNok	0.04	0.03	0.89	0.03	0.01	0.0
MovNok	0.12	0.05	0.1	0.68	0.03	0.01
SitOk	0.01	0.01	0.07	0.0	0.88	0.03
SitNok	0.01	0.0	0.0	0.03	0.02	0.94

Disregarding the activity recognition and looking only at the posture detection performance yields the confusion matrix in Table II. The false negative rate is at 17% while the false positive rate is at 8%. Hence, it is more likely the device does not trigger an alarm although there is a wrong posture than the opposite.

B. *Experiment II*

In the second experiment different preprocessing methods are evaluated. A median and an average filter is applied with variable window size. Additionally we compute the peak to peak magnitude of the accelerometer across a certain window size as an additional feature.

The best results are displayed in Table I and show that the classification of individual algorithms could be improved. For example the decision tree and the neural network achieve better results than in the first experiment. However, none of the newly applied methods top the so far best result of the 6NN in experiment I.

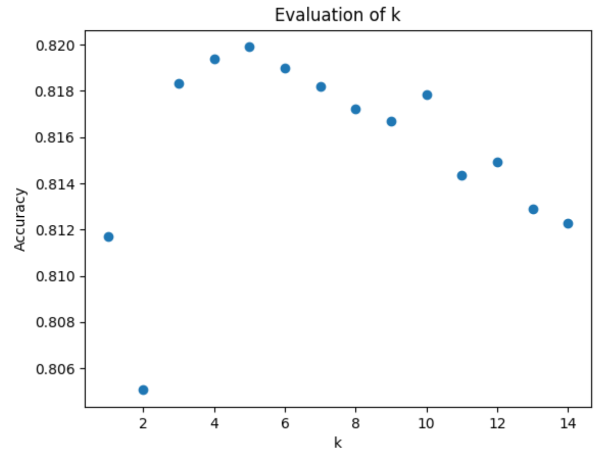


Fig. 5. Performance of kNN with different amount of neighbors.

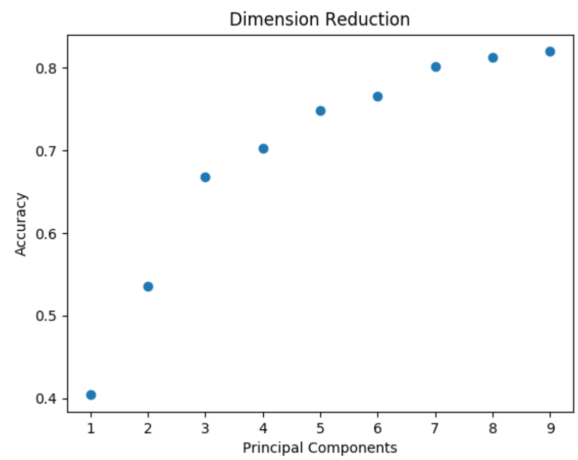


Fig. 6. Performance of 5NN at different dimensions.

C. *Experiment III*

In the third experiment the best performing classifier so far is fine tuned further. Dimension reduction with principal component analysis (PCA) is performed and a varying amount of neighbors for the kNN classifier is chosen. Further tuning can be applied by using different distance measures. The metrics Minkowski,"Manhattan", "Chebyshev" and "Mahalanobis" are evaluated.

Figure 5 shows the performance of the kNN classifier at different amount of neighbors. The graph shows the best performance can be obtained using 5 neighbors. This configuration even yields the so far best result, an accuracy of 82.02 %.

Figure 6 shows the performance of the 5NN classifier at different dimensions. The graph shows that the best performance is obtained using all 9 dimensions.

A slight improvement could be obtained using the "Manhattan" distance metric. It leads to a new best result of 83.35%

TABLE I
CLASSIFICATION PERFORMANCE IN DIFFERENT EXPERIMENTS.

	6NN	AdaBoost	Decision Tree	Random Forest	Neural Network	SVM	QDA
Experiment I	0.81± 0.07	0.44 ± 0.14	0.61 ± 0.15	0.61 ± 0.13	0.75 ± 0.10	0.50 ± 0.12	0.56 ± 0.11
Experiment II							
Average (w=18)	0.68± 0.08	0.36 ± 0.11	0.60 ± 0.14	0.62 ± 0.12	0.69 ± 0.13	0.47 ± 0.15	0.62 ± 0.12
Median (w=30)	0.69 ±0.12	0.41 ± 0.17	0.71 ± 0.13	0.61 ± 0.12	0.78 ± 0.11	0.47 ± 0.13	0.69 ± 0.12

TABLE II
POSTURE DETECTION PERFORMANCE.

	Posture Ok	Posture Not Ok
Posture Ok	0.92	0.08
Posture Not Ok	0.17	0.83

D. Experiment IV

So far all classification was evaluated on the data of three users, as the goal was to find a one-fits-all solution. However, another possibility is to train the device only on individual users thus empowering the user to adapt the device to himself.

In this experiment the posture detection algorithm is trained and evaluated only on the individual user data. This leads to an average accuracy of 90% with a standard derivation of 0.05%. It shows that with additional effort of the user, the performance of the device can be improved.

E. Experiment V

Although the offline evaluation gives clear insights on the performance of different methods on a collected dataset, this does not mean the methods will work equally well in practice. That's why in this experiment the practical usage of the system is tested. A full evaluation for practical usage should contain energy consumption, accuracy, training time from different users. However, such an experiment is beyond the scope of this work and will be evaluated in the future. Instead a smaller experiment with two users is conducted.

It turns out that the result of the offline evaluation can not be repeated. Better results are obtained using short training phases of 1 minute, an average filter of 30 samples and a 7 nearest neighbors classifier.

VI. DISCUSSION AND FUTURE WORK

In the first evaluation step different posture detection and data representation methods are compared on a collected dataset. The results are used to answer the questions posed for evaluation.

1) **What is the overall potential of posture detection using pattern recognition algorithms in combination with data attainable from 9-axis motion sensor?**
Generally, the posture detection algorithms can identify wrong postures in most static cases, e.g. when sitting or standing. Here the accuracy lays over 90%. More problematic is the detection of wrong postures while moving. Although the best performing methods are capable of detecting a movement, they can not accurately distinguish between moving with a healthy posture and moving with a unhealthy posture.
The false negative rate is at 17% and the false positive rate at 8%. Assuming a wrong posture for a short amount of time is not life threatening but false alarms can be quite annoying for the user this is an acceptable result.
2) **Is it possible to train a classifier for multiple users so the system works without further calibration and training for the individual user?**
In the experiments the accuracy could be improved by 7 % by training only on the individual user. Therefore we can conclude that there is a trade-off between usability (no need for individual training) and accuracy. However, a dataset of three persons is not really representative. Repeating the experiment with a larger amount of people should be part of future work.
3) **What is an appropriate representation of the measured data?**
Principal component analysis was applied to reduce the data dimension. Additionally other features like a peak-to-peak measurement of an accelerometer window was tested. The best results were obtained using sensor data of all 9-axis.
4) **What is the best performing classification algorithm?**
The best result were obtained using a 5NN-classifier with manhattan distance metric.
5) **Can filtering further improve classification performance?**
Filtering could not further improve the classification performance.

Throughout the study, the challenges that were anticipated in the beginning turned out to be true. Namely, the biggest struggle is with coming up with a robust and one fits all definition and classification method of posture using only one motion sensor that is not necessarily attached to one's body in a rigid manner. The ambiguities hit the performance in two main ways: defining right posture, especially during movement, is a hard task even from a medical point of view. The decision boundaries can get really tight and boundaries that belong to different activities can overlap with each other. This can be mitigated by rigorous training and calibration algorithms which tailor algorithms to a particular person wearing the device in a particular way, but going far with solutions of such kind can easily deviate the product away from user friendliness. Overall, what paó is trying to achieve remains an interesting technical and design study.

REFERENCES

[1] Thrasher M., Van der Zwaag M.D., Bianchi-Berthouze N., Westerink J.H.D.M. (2011). Mood Recognition Based on Upper Body Posture and Movement Features. Affective Computing and Intelligent Interaction. ACII 2011. Lecture Notes in Computer Science, vol 6974. Springer, Berlin, Heidelberg.

[2] Priester, J. M., Cacioppo, J. T., & Petty, R. E. (1996). The influence of motor processes on attitudes toward novel versus familiar semantic stimuli. Personality and Social Psychology Bulletin, 22, 442447.

[3] Laird, J. D., & Bresler, C. (1992). The process of emotional experience: A self-perception theory. In M. S. Clard (Ed.), Review of personality and social psychology (Vol. 13, pp. 213234). Newbury Park, CA: Sage.

[4] Neumann, R., Forster, J., & Strack, F. (2003). Motor compatibility: The bi-directional link between behavior and evaluation. The psychology of evaluation: Affective processes in cognition and emotion (pp. 371391). Mahwah, NJ: Lawrence Erbium Associates.

[5] Petty, R. E., Wells, G. L., Heesacker, M., Brock, T. C., & Cacioppo, J. T. (1983). The effects of recipient posture on persuasion: A cognitive response analysis. Personality and Social Psychology Bulletin.

[6] S. Chopra, M. Kumar and S. Sood, Wearable posture detection and alert system. 2016 International Conference System Modeling and Advancement in Research Trends (SMART), Moradabad, 2016, pp. 130-134.

[7] X. Yuan, S. Yu, Q. Dan, G. Wang and S. Liu, "Fall detection analysis with wearable MEMS-based sensors. Electronic Packaging Technology (ICEPT), 2015 16th International Conference on, Changsha, 2015, pp. 1184-1187.

[8] B. S. Somesh, A. Mukherjee, S. Sen and P. Karmakar. Constant current control of stepper motor in microstepping mode using PlC I6F877A. Devices, Circuits and Systems (ICDCS), 2014 2nd International Conference on, Combiatore, 2014.

[9] Johnson, W. (1996). The posture of meditation. Boston: Shambhala, 22-26.

[10] Fuchs T., Koch S. C. (2014). Embodied affectivity: on moving and being moved. Front. Psychol. 5:508 10.3389/fpsyg.2014.00508.

[11] Cuddy A. J. C., Wilmuth C. A., Carney D. R. (2012). The benefit of power posing before a high-stakes social evaluation, in Harvard Business School Working Paper, No. 13-027.

[12] Cha, Youngsu, Kihyuk Nam, and Doik Kim. Patient Posture Monitoring System Based on Flexible Sensors. Ed. Hyun-Joong Chung and Tae-il Kim. Sensors (Basel, Switzerland) 17.3 (2017): 584. PMC. Web. 14 Nov. 2017.

[13] Rabiner, Lawrence, and B. Juang. "An introduction to hidden Markov models." ieee assp magazine 3.1 (1986): 4-16.

[14] Theodoridis, Sergios, and Konstantinos Koutroumbas. "Pattern recognition." IEEE Transactions on Neural Networks 19.2 (2008): 376.