Personalized Online Learning of Whole-Body Motion Classes Using Multiple Inertial Measurement Units

Viktor Losing[1,2], Taizo Yoshikawa[3], Martina Hasenjaeger[2], Barbara Hammer[1] and Heiko Wersing[2]

***Abstract*— Online action classification is an important field of research, enabling the particularly interesting application scenario of controlling wearable devices which actively support the user's motions. The majority of machine learning applications of real-world systems are based on pre-trained average-user models without any personalization. Our long-term goal is to provide a system that adapts to its user's personal behavior patterns on the fly and in real-time. Ideally, we want to initiate a continuous collaboration between the system and the user where both alternatively adjust to each other to maximize the system's utility. Such tasks are not feasible with static models. In this paper, we investigate the potential and benefits of personalized online learning in the task of online action classification. We record motion sequences of different subjects wearing the Xsens bodysuit, which incorporates multiple inertial measuring units, enabling a fine-grained discrimination of motions. On this basis, we first perform a feature selection, showing that only a few sensors are necessary to achieve a high classification performance. Subsequently, we compare the recognition capabilities of offline average user models against personalized models trained in an online way. Our experiments conclude that personalized models require only few data to outperform average user systems and are particularly valuable for applications with limited computational hardware which rely on the raw sensor inputs only.**

I. INTRODUCTION

The classification of human actions is crucial for diverse application scenarios such as surveillance, human-machine interactions and pervasive health care [1], [2]. In contrast to the tasks of activity recognition or offline action classification, where the classification is performed on the basis of a fully observed sequence, online action classification aims to recognize the motion on the fly and as quickly as possible. This crucial difference opens up a spectrum of further application scenarios. One that we are particularly interested in is the control of wearable devices which actively support the motion of users. Such hardware may support physical rehabilitation or assist handicapped persons by means of an improved prosthesis control [3], [4] for example. Another application area are working environments where repetitive and strenuous motions such as bending and kneeling are frequently demanded [5]–[7].

Usually, action classification is done with skeleton data extracted from vision-based sensors [8]–[12]. However, the accessibility of visual data is limited to environments equipped with the necessary sensors. In case of wearable devices, integrating inertial measuring units (IMU) is a viable and elegant solution to obtain a continuous data stream independent from environmental properties. IMUs have been mostly used for activity recognition tasks, which aim to discriminate high-level actions such as walking, cycling and swimming, often on the basis of commercial everyday hardware like smart-phones, smart-watches or fitness-tracker wristbands [13]–[15]. In contrast, we apply the Xsens bodysuit [16], [17] which incorporates seventeen IMUs spread across the body, providing a rich measurement of postures with a high sample rate, enabling motion recognition in a fine-grained way.

Commonly, online action classification has been tackled with offline machine learning methods. A static model is generated by a large amount of labeled examples and then applied to a hold-out set [8]–[12]. However, we aim for a system that adapts to the personal behavior patterns of its user in real-time and on the fly based on inertial measurements. Ideally, a continuous collaboration between the system and user is triggered where both alternatively adjust to each other and maximize the utility of the system. A static model is simply not suitable in such a scenario. Hence, we apply online machine learning algorithms which continuously adapt to the incoming data stream. Since the data is processed one-by-one, these algorithms are able to handle infinite streams and thereby guarantee a low time and space complexity. Therefore, even limited computational hardware is able to process the stream locally without access to cloud services, allowing an application independent from internet connectivity and offering complete data privacy at the same time.

Personalization denotes the modification of a system towards the characteristics of an individual user. Two different modes of personalization have been distinguished [18], [19]: i) Active customization by the user, e.g. by making selections and setting parameters and ii) adaptive systems where the usage history is employed to estimate user preferences and situation statistics to adjust parameters and behavior. The general idea behind personalized learning is that the focus on one person drastically reduces the variance within the data, enabling a better performance with a smaller amount of data. Furthermore, the major problem of inter-person generalization is completely avoided, facilitating the task as well as the computational complexity, since post-processing steps such as normalization or temporal integration can often be omitted.

The potential of personalization in the context of motion recognition was mostly analyzed with static models. Weiss

[1]Bielefeld University, Universitätsstr. 25, 33615 Bielefeld, Germany
[2]HONDA Research Institute Europe, Carl-Legien-Str. 30, 63073 Offenbach am Main, Germany
[3]HONDA R&D Co., Ltd. R&D Center X, 8-1 Honcho, Wako-shi, Saitama, 351-0188 Japan

and Lockhart investigated personalization in an activity recognition task on the basis of accelerometer-data obtained from smart phones [20]. In their experiments, personalized models were more accurate with a clearly smaller amount of training data. Medrano et al. used personalized models for the task of human fall-detection and achieved a higher performance in comparison to average user models [21]. Our contribution differs from the mentioned work, since we combine personalization with online learning to classify motion classes on the basis of a large data foundation obtained with multiple IMUs. We first perform a forward feature selection to determine a small sub-set of IMUs which are the most valuable ones in terms of classification performance. Subsequently, we thoroughly compare average offline machine learning models with online personalized ones. Our experiments show that online models require a small amount of data to outperform average user systems, particularly if only the raw sensor data is used, often a necessity for applications with strictly limited computational resources. The advantage of the online models increases further with more training data.

II. Framework

Our focus is the evaluation of off- and online learning models in the online action classification task. In the following, we introduce the classification task and describe the characteristics of both learning schemes.

A. Online action classification

Our application is an online action classification problem [9]. A stream $\{\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_t}\}$ of feature vectors (IMU sensor measurements in our case) arrives one after another. The goal is to determine whether a frame $\mathbf{x_t}$ at time t belongs to an action among the predefined C action classes. Algorithms are allowed to use not only the current feature vector $\mathbf{x_t}$ but also those of the past $(\mathbf{x_{t-1}} \ldots \mathbf{x_1})$. The classification is done for each feature vector presented in the order of the stream. In contrast to action detection tasks, there are no situations in which $\mathbf{x_t}$ does not belong to any predefined action. A model predicts the action class in the form of:

$$y_t^\star = \underset{y_t \in \{1,\ldots,C\}}{\arg\max} \quad P(y_t | x_1, \ldots, x_t).$$

Naturally, online action classification is more challenging than offline action classification, since methods are not allowed to peek in the future and must instantaneously determine the action of x_t.

B. Offline learning

In the offline learning setting, an algorithm generates a model function $h : \mathbb{R}^n \mapsto \{1, \ldots, c\}$ based on a training set $D_{\text{train}} = \{(\mathbf{x}_i, y_i) \,|\, i \in \{1, \ldots, j\}\}$. In the subsequent test phase, the model is applied on another set $D_{\text{test}} = \{(\mathbf{x}_i, y_i) \,|\, i \in \{j+1, \ldots, k\}\}$, whose labels are kept hidden. The model provides a label $\hat{y}_i = h(\mathbf{x}_i)$ for every point $x_i \in D_{\text{test}}$ and the 0-1 loss $\mathcal{L}(\hat{y}_i, y_i) = \mathbb{1}(\hat{y}_i \neq y_i)$ is calculated. The test error

$$E(D_{\text{test}}) = \frac{1}{k} \sum_{i=j+1}^{k} \mathcal{L}(h(\mathbf{x}_i), y_i) \tag{1}$$

is the commonly used performance metric. In our case, an offline average user model is generated by using the data of all but one subjects for training and tested to classify the actions of the hold-out subject.

C. Online learning

The online learning setting is more challenging, since the data is accessed one-by-one in a predefined order and the algorithm provides a model after each datapoint. Therefore, online algorithms initially tend to deliver a lower performance compared to their offline counterparts. However, they provide the benefits of a low time and space complexity, are able to process datasets of arbitrary sizes and allow particular tuning to a special problem domain.
Formally, a potentially infinite sequence $S_t = (s_1, s_2, \ldots, s_t)$ of tuples $s_i = (\mathbf{x}_i, y_i)$ arrives one after another. In contrast to the offline setting, a model function is generated after each tuple. As t represents the current time stamp, the classification $\hat{y}_t = h_{t-1}(\mathbf{x}_t)$ is done according to the previously learned model h_{t-1}. After the true label y_t is revealed, the applied learning algorithm generates a new model $h_t = \text{train}(h_{t-1}, s_t)$ on the basis of the current tuple s_t and the previous model h_{t-1}. Usually, the interleaved test train error is used for performance evaluation and is defined as:

$$\hat{E}(S_t) = \frac{1}{t} \sum_{i=1}^{t} \mathcal{L}(h_{i-1}(x_i), y_i). \tag{2}$$

We train our online personalized models iteratively from the scratch. Precisely, each action of one subject is first classified by the model and subsequently used for training.

The principle difference between off- and online approaches is that the offline methods have generally a much larger set of training data available, whereas the online algorithms have the capability to adapt to the actual test data. The natural consequence is that online methods using few data are only applicable if the variation in the test condition is not too high, which is particularly the case for personalized learning. Online algorithms are even able to adapt to non-stationary environments and efficient methods have been recently published [22], [23], however, this is beyond the scope of this contribution.

Our goal is to evaluate the potential of online learning schemes for model individualization as just formalized in the context of online action classification based on IMUs. Thereby, we will rely on state-of-the-art online learning schemes as exist in the literature, in particular the Online Random Forest [24] algorithm.

III. Dataset

The data recording as well as the resulting dataset is described in this chapter.

Fig. 1: The XSENS body suit with seventeen IMU sensors. The sternum IMU is occluded due to the image perspective. We used the wireless version. Source: `https://www.xsens.com/`.

A. Recording setup

We used the popular Xsens bodysuit with seventeen IMUs, measuring linear and angular motions with a triad of gyroscopes and accelerometers, distributed on different body locations as shown in Figure 1. We used a sample rate of of 60 Hz, which is usually sufficient to track highly dynamic motions as common in sports [25]. Six additional sensor positions are interpolated, resulting in altogether 23 different *segments*. Altogether, thirteen different *feature types* such as acceleration, joint-angles or orientation are provided for each segment and are encoded with three or four dimensions. Various filters are used to provide also integrated feature types such as velocity and position. However, these are prone to drift due to the inevitable sensor bias [26].

B. Feature extraction and normalization

We translated the position of all segments such that the pelvis is in the origin of the X-Y coordinates. Xsens offers kinematic data as 3-dimensional vectors. We enriched the data by adding further feature types which encode only the magnitude of these. Furthermore, a normalized version of each feature is added which is obtained by mean-subtraction and scaling to unit-variance. This has been done per person to improve generalization across different subjects. In total, up to 32 different feature types are available for each of the 23 segments, resulting in 1472 dimensions per sample.

C. Action variety

Four different subjects performed nine movement sequences consisting of several single actions. These sequences were repeated 10-20 times. In total, sixteen fine-grained action classes are present in the data[1]. The recordings were done in one session for each subject and the data was manually labeled. Figure 2 depicts some action sequences. The action class distribution of the dataset is depicted in Figure 3. The data is mainly dominated by five different walking actions as well as the standing class, which frequently occurs in-between. Altogether, the dataset encodes 2755 actions represented by 329021 single instances.

[1] The action classes are : stand, walk forward, walk backwards, walk sideways, walk curve, turn, squat down, squat up, lunge down, lunge up, stand with object, walk forward with object, walk backwards with object, turn with object, put object down, squat up with object)

Fig. 3: The imbalanced class distribution of the recorded data. For reasons of clarity, only the dominating classes are labeled in the chart. ST = stand, WB = walk backwards, WF = walk forward, WS = walk sideways, TU= turn around.

IV. Experiments

We compare an offline average user model against an online personalized one, which we denote as *average* (AVG) and *personalized* (PERS) from now on. The *average* model is trained in leave-one-subject-out scheme. Precisely, it is tested with the data of one specific subject, whereas data of the remaining three subjects is used for training. This is done repeatedly such that each subject is used for testing once. Personalized models are evaluated in the online learning setting, as described in Section II-C. The model classifies first the label of one sample and uses it afterward for model adaption. This is done for all samples in the dataset. However, the order of each instance within one performed action is predefined by the recording time, and therefore, there is a high degree of label auto-correlation, since each action consists of a number of samples with the same class. In this case, the ordinary online scheme is misleading in because a naive classifier, simply using the previously seen label for classification, achieves a very low error rate without learning anything. Therefore, we perform an action-wise evaluation. Precisely, the model classifies all samples of one action, before the corresponding labels get revealed. The personalized models are trained from scratch, one online model for every subject in single pass. Consequently, online models are utilized without any form of pre-training and only access the data of one subject. Please note, that we calculate both errors (off- and online) using the same data for testing, but the online algorithms continuously adapt their model to the test subject.

We apply on- and offline variants of the popular Random Forest (RF) [24], [27] to enable a possibly fair comparison. The RF is a well known state-of-the-art learning algorithm, delivering highly competitive results [28], [29] and is easy to apply out of the box. Concretely, we use decision forests consisting of 50 trees and rely on the class entropy as impurity function [30].

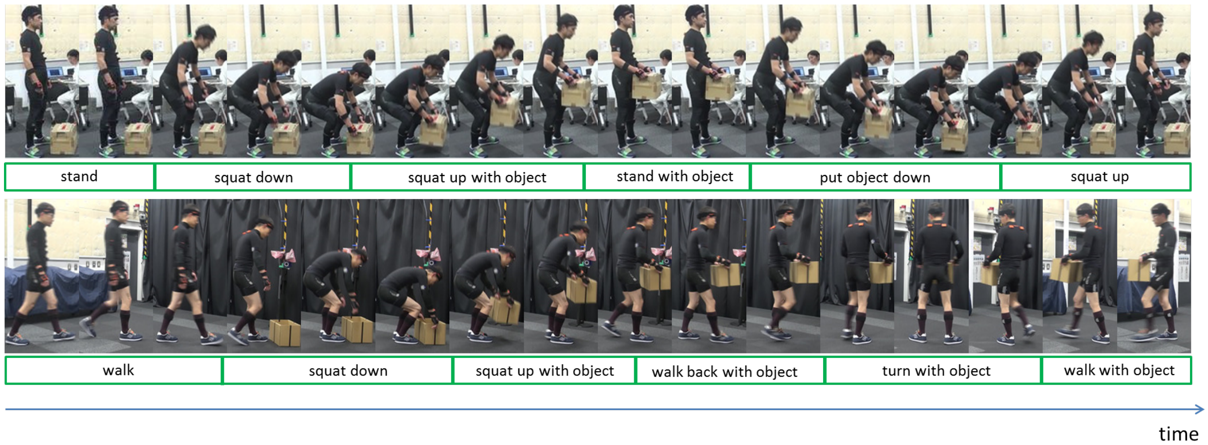

Fig. 2: Exemplary sub-parts of sequences with different subsequent actions. The ground truth is depicted in the green boxes. The classes are often ambiguous in the transition period from one action to another.

A. Encoding recent motion data

Action classification is known to be more accurate when the feature vector encodes not only the current sensor state, but the recent motion history. We performed preliminary experiments to compare the effect of different feature configurations. We evaluate the performance for encoding only the current sensor state versus stacking the features of the last 30 frames ($\sim$0.5 s)) versus encoding the features of the last 30 frames via the five highest values of the discrete cosine transformation (DCT) [31]. The highest performance is achieved by the DCT encoding even though it uses a substantially lower amount of dimensions in comparison to stacking. Based on these results, we use the DCT encoding for the rest of the experiments[2]. The detailed results are listed in the appendix (Table II).

B. Feature selection

The Xsens body suit offers various feature types for each segment. To reduce the number of input dimensions, we performed for each model feature selection, determining the most valuable dimensions in two steps. We use forward feature selection [33] to minimize the classification error. Concretely, we start with an empty set and iteratively add the feature type / segment which minimizes the error the most until all of them are used. In the first step, we extract the most valuable feature types (see section III), thereby using all segments and in the second we determine the most valuable segments using only the previously extracted feature types. The normalized feature types were not offered to the *personalized* model, since they have no effect on person-specific and scale-invariant models. Figure 4 shows the error rate depending on the number of feature types / segments. It can be seen, that only few feature types and segments are necessary to reach a high performance. Using all feature types is even harmful which is probably caused by overfitting. The personalized model relies on less data and is therefore more affected. Based on these results, we decide to use the three best feature types together with the five most valuable segments. Table I lists the most valuable feature types for both models. The chosen types by the *average* model are mostly temporally integrated and normalized, highlighting that the person-specific normalization improves the inter-person generalization. In contrast, the *personalized* model prefers rather the raw sensor data. Figure 5 displays

Fig. 4: Forward feature selection in two steps: First, the best feature types are determined using all segments (top). Subsequently, the most valuable segments are determined on the basis of the three best feature types (bottom). The *personalized* model had no access to the normalized features because they have no effect on its performance. The number of evaluated segments depends on the chosen feature types in the first step. Hence, the number of evaluated segments is different for both models.

[2] The discrete Fourier transformation (DFT) [32] was evaluated as well, however, it performed worse than the DCT

TABLE I: The three most important feature types.

Priority	AVG	PERS
1	position (scaled)	sensor-acceleration
2	magnitude-angular-velocity (scaled)	sensor-angular-velocity
3	velocity (scaled)	center-of-mass

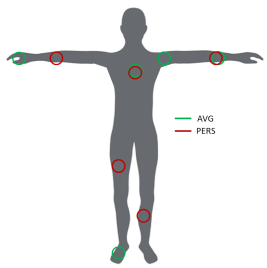

Fig. 5: The favored sensor locations of both models. There is no excessively covered body part. Instead, the sensors are equally distributed across the whole body.

the preferred sensors. Both models equally distribute the segments across the body, indicating that our motion dataset profits from an encoding of the whole body posture.

C. Results

The reported results are based on the selected features (Section IV-B) which are encoded via the five highest values of the corresponding DCT (Section IV-A). Figure 6 shows the average error rate of the *personalized* and *average* model depending on the relative action progress. As expected, the models are the most uncertain during the transition period from one action to another, since the data is very similar then for consecutive action classes. Also the ground truth is most inconsistent then because it is hard to define the exact moment one action ends and another starts. The models are particularly uncertain at the beginning of motions because the feature vector encodes the recent past. Consequently, a lot of variance is within the data at this stage, since different action classes can transition into the same class.
Both models perform similarly on average with the *average* one being slightly better. However, the *personalized* model is continuously adapting, therefore, we analyze it in more detail. Assuming the data of a given subject contains l single

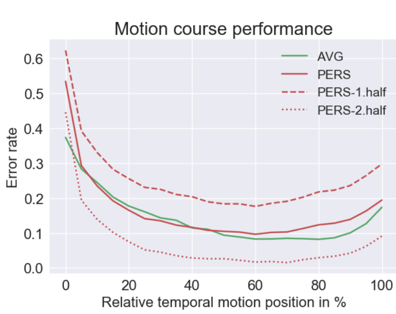

Fig. 6: The classification performance depending on the relative progress of the action. The models are less accurate in the transitions between the actions.

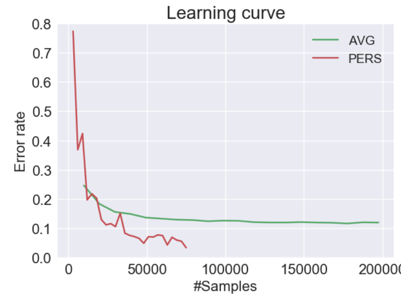

Fig. 7: The learning curve of both models. The *personalized* model achieves a lower error with significantly less data.

motions, we break the set into approximately equally sized sets l_1 and l_2, i.e. $|l_1| \approx |l_2| \approx \frac{|l|}{2}$, and measure the performance on each of those independently. The performance of the first half is the incremental error achieved on l_1 by an online model starting from scratch without any prior knowledge and continuously adapting on the basis of l_1. Averaging the results over all subjects leads to the curve in the plot. As expected, it performs clearly worse in comparison to the *average* model. The performance of the second half is the incremental error achieved on the second half of the motions l_2, but this time the model has already seen the data of the first half l_1. In this case, we observe that the *personalized* model significantly outperforms the *average* one as soon it has seen enough motions of the respective person. The average performance of the *personalized* model can be expected to converge at least towards the error rate achieved for the second half of the data with additional motion sequences.
The learning curves are depicted in Figure 7. The performance of the *personalized* model is measured by averaging the performance over the last 3600 instances ($\sim$1 min), whereas the performance of the *average* model is always measured by classifying the whole data of the hold-out-subject. The *average* model has access to significantly more data because it is trained on three subjects. Nonetheless, the learning curve of the *personalized* model is not only steeper, but it reaches a distinctly lower error rate, highlighting the benefits of personalized learning in this field.

D. Raw sensor data

It is a matter of time until the integration of even low sensor bias signals leads inevitably to drift, if no recalibration is periodically performed. One way to avoid the continuous bias accumulation is to use the raw sensor signals without any integration. Also applications where the integration is not possible due to limited computational resources have to rely on the raw data. Figure 8 depicts the classification performance when only the raw sensor data is used in comparison to the one using also integrated features[3].

The inter-subject generalization ability of the *average* model significantly drops if only the sensor data is used,

[3] Similar to the selection process in Section IV-B, we determined the most relevant features of the raw sensor data.

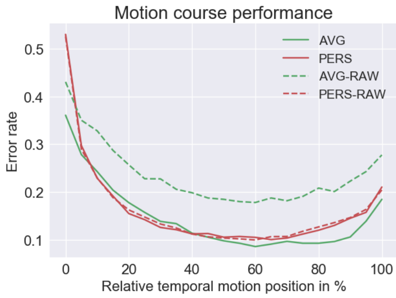

Fig. 8: The error rate depending on the relative progress of the action. The performance achieved with the raw features is compared with the one including integrated information. Varying sensor signals across the subjects reduce the generalization ability of the *average* model.

because of the person-specific nature of the raw signals. The *personalized* model is basically not affected, since it uses in both cases mostly the raw signals. Hence, personalized learning is even more valuable when the input is limited to the raw sensor data.

V. Obtaining the ground truth online

In this paper, we simulated personalized online learning with pre-labeled data. However, the application of such a system in real-world scenarios requires online ground truth, which is often hard to obtain. However, in most cases, a labeling delay is tolerable and therefore ground truth can be provided in retrospective. One possibility is to let humans label the recorded data, which has often been done in human-machine interaction scenarios [34]. Additionally, active learning approaches can be integrated to reduce the labeling burden [35], [36]. A feasible way to get the labels without human interaction is to use another model which classifies the event after it has occurred. In other words, this model has the advantage to peek in the future, which often drastically facilitates the task. In various prediction tasks, the classification of the event is very simple in retrospective such that the ground truth quality is comparable to manually annotated data [37]. Though, action classification is a hard task even after the complete motion has been performed, it is still less challenging than doing it online. Hence, this semi-supervised learning approach may be a reasonable choice, however it is just a concept we have not applied, yet. Figure 9 shows the corresponding system architecture.

VI. Conclusion

In this paper, we analyzed the application of personalized online machine learning models in the task of online action classification. Using the Xsens body suit, we recorded a large motion database with over 2750 actions. Four different subjects repetitively performed various motion sequences which are categorized into sixteen different classes. The data was enriched with normalization techniques to improve the inter-subject generalization. We performed a forward feature selection, which showed that only a few IMUs are necessary

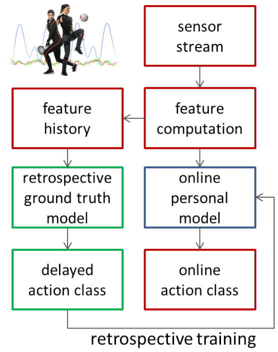

Fig. 9: Online learning system architecture. The ground truth is determined with an additional model which is allowed to buffer a some data and classify actions with delay. These delayed labels are used to train the online model.

TABLE II: Average error rates of all experiments.

Feature set	#Dimensions AVG	#Dimensions PERS	AVG	PERS	PERS-1.half	PERS-2.half
Single frame	33	27	0.177	0.186	0.173	0.986
Stacked	990	810	0.116	0.145	0.229	0.607
DCT	165	135	0.115	0.134	0.213	0.558

(a) The results are based on all available features including post-integrations and normalizations. Three feature types and five segments were used for both models.

Feature set	#Dimensions AVG	#Dimensions PERS	AVG	PERS	PERS-1.half	PERS-2.half
Single frame	35	35	0.246	0.172	0.237	0.108
Stacked	1050	1050	0.190	0.148	0.218	0.794
DCT	175	175	0.179	0.142	0.215	0.703

(b) Only the raw sensor signals were used in these experiments. Three feature types and five segments were used for both models.

to obtain a high performance. Using state-of-the-art machine learning algorithms, we compared personalized online models against average-user offline models. It turns out that personalized models are very efficient learners yielding better results with a small amount of data which indicates that the motions are indeed performed in a personalized way. The performance gain is particularly pronounced when only the raw sensor signals are used without any integration. The advantage is likely to increase with additional data.
In the future, we want to continue this analysis with additional data and particularly evaluate the personal data variance between different recording sessions. We already started to fuse off- and online personalized models to avoid a cold start at the beginning. The first results show that the hybrid model indeed provides a performance gain. Our long-term goal is to utilize personalized models in a real-world application as described in Section V.

Appendix

For the sake of completeness, we report the average error rates of all experiments in Table II.

REFERENCES

[1] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, vol. 43, no. 3, pp. 16:1–16:43, Apr. 2011. [Online]. Available: http://doi.acm.org/10.1145/1922649.1922653

[2] R. Poppe, "A survey on vision-based human action recognition," *Image Vision Comput.*, vol. 28, no. 6, pp. 976–990, June 2010. [Online]. Available: http://dx.doi.org/10.1016/j.imavis.2009.11.014

[3] B. Paaen, A. Schulz, J. Hahne, and B. Hammer, "An EM transfer learning algorithm with applications in bionic hand prostheses," in *Proceedings of the 25th European Symposium on Artificial Neural Networks (ESANN 2017)*, M. Verleysen, Ed. i6doc.com, 2017, pp. 129–134.

[4] K. Kong and D. Jeon, "Design and control of an exoskeleton for the elderly and patients," *IEEE/ASME Transactions on mechatronics*, vol. 11, no. 4, pp. 428–432, 2006.

[5] J. E. Pratt, B. T. Krupp, C. J. Morse, and S. H. Collins, "The roboknee: an exoskeleton for enhancing strength and endurance during walking," in *IEEE International Conference on Robotics and Automation, 2004. Proceedings. ICRA '04. 2004*, vol. 3, April 2004, pp. 2430–2435 Vol.3.

[6] Y. Muramatsu, H. Kobayashi, Y. Sato, H. Jiaou, and T. Hashimoto, "Quantitative performance analysis of exoskeleton augmenting devices - muscle suit - for manual worker," *International Journal of Automation Technology*, vol. 5, pp. 559–567, 07 2011.

[7] Honda. [Online]. Available: http://asimo.honda.com/innovations/feature/body-weight-support-assist/

[8] G. Garcia-Hernando and T.-K. Kim, "Transition forests: Learning discriminative temporal transitions for action recognition and detection," *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 407–415, 2017.

[9] Y. Li, C. Lan, J. Xing, W. Zeng, C. Yuan, and J. Liu, "Online human action detection using joint classification-regression recurrent neural networks," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 203–220.

[10] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom sequence models for efficient action detection," in *CVPR 2011*, June 2011, pp. 3201–3208.

[11] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *2013 IEEE International Conference on Computer Vision*, Dec 2013, pp. 3551–3558.

[12] M. Jain, J. v. Gemert, H. Jgou, P. Bouthemy, and C. G. M. Snoek, "Action localization with tubelets from motion," in *2014 IEEE Conference on Computer Vision and Pattern Recognition*, June 2014, pp. 740–747.

[13] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," in *Proceedings of the Fourth International Workshop on Knowledge Discovery from Sensor Data*, 2010, pp. 10–18.

[14] G. M. Weiss, J. L. Timko, C. M. Gallagher, K. Yoneda, and A. J. Schreiber, "Smartwatch-based activity recognition: A machine learning approach," in *2016 IEEE-EMBS International Conference on Biomedical and Health Informatics (BHI)*, Feb 2016, pp. 426–429.

[15] E. C. Nelson, T. Verhagen, and M. L. Noordzij, "Health empowerment through activity trackers: An empirical smart wristband study," *Computers in human behavior*, vol. 62, pp. 364–374, 2016.

[16] D. Roetenberg, H. Luinge, and P. Slycke, "Xsens mvn: Full 6dof human motion tracking using miniature inertial sensors," vol. 3, 01 2009.

[17] Xsens. [Online]. Available: https://www.xsens.com/research/published-papers/

[18] G. Fischer, "User modeling in human-computer interaction," *User Modeling and User-Adapted Interaction*, vol. 11, no. 1-2, pp. 65–86, 2001.

[19] M. Hasenjger and H. Wersing, "Personalization in advanced driver assistance systems and autonomous vehicles: A review," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 1–7.

[20] G. M. Weiss and J. W. Lockhart, "The impact of personalization on smartphone-based activity recognition," in *AAAI Workshop on Activity Context Representation: Techniques and Languages*, 2012, pp. 98–104.

[21] C. Medrano, I. Plaza, R. Igual, Á. Sánchez, and M. Castro, "The effect of personalization on smartphone-based fall detectors," *Sensors*, vol. 16, no. 1, p. 117, 2016.

[22] V. Losing, B. Hammer, and H. Wersing, "KNN Classifier with Self Adjusting Memory for Heterogeneous Concept Drift," in *16th International Conference on Data Mining (ICDM)*. IEEE, 2016.

[23] A. Bifet, G. Holmes, and B. Pfahringer, "Leveraging bagging for evolving data streams," *Machine Learning and Knowledge Discovery in Databases*, pp. 135–150, 2010.

[24] A. Saffari, C. Leistner, J. Santner, M. Godec, and H. Bischof, "On-line random forests," in *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*. IEEE, 2009, pp. 1393–1400.

[25] B. Tessendorf, F. Gravenhorst, B. Arnrich, and G. Trster, "An imu-based sensor network to continuously monitor rowing technique on the water," in *2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing*, Dec 2011, pp. 253–258.

[26] R. G. J. Damgrave and D. Lutters, "The drift of the xsens moven motion capturing suit during common movements in a working environment," in *Proceedings of the 19th CIRP Design Conference–Competitive Design*. Cranfield University Press, 2009.

[27] L. Breiman, "Random forests," *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[28] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems," *J. Mach. Learn. Res*, vol. 15, no. 1, pp. 3133–3181, 2014.

[29] V. Losing, B. Hammer, and H. Wersing, "Choosing the Best Algorithm for an Incremental Learning Task," in *European Symposium on Artificial Neural Networks (ESANN)*, 2016.

[30] U. M. Fayyad and K. B. Irani, "The attribute selection problem in decision tree generation," in *AAAI*, 1992, pp. 104–110.

[31] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," *IEEE Transactions on Computers*, vol. C-23, no. 1, pp. 90–93, Jan 1974.

[32] F. J. Harris, "On the use of windows for harmonic analysis with the discrete fourier transform," *Proceedings of the IEEE*, vol. 66, no. 1, pp. 51–83, Jan 1978.

[33] S. Abe, "Feature selection and extraction," in *Support Vector Machines for Pattern Classification*. Springer, 2010, pp. 331–341.

[34] S. Amershi, M. Cakmak, W. B. Knox, and T. Kulesza, "Power to the people: The role of humans in interactive machine learning," *AI Magazine*, December 2014.

[35] B. Settles and M. Craven, "An analysis of active learning strategies for sequence labeling tasks," in *Proceedings of the conference on empirical methods in natural language processing*. Association for Computational Linguistics, 2008, pp. 1070–1079.

[36] C. Limberg, H. Wersing, and H. Ritter, "Improving active learning by avoiding ambiguous samples," in *International Conference on Artificial Neural Networks (ICANN)*. Springer, October 2018.

[37] V. Losing, B. Hammer, and H. Wersing, "Personalized maneuver prediction at intersections," in *2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC)*, Oct 2017, pp. 1–6.