Open-Ended Human Activity Recognition using Matching Networks

Abstract

Open-ended Human Activity Recognition (HAR) addresses the challenge of recognising activities outside of a limited pre-defined set, which improves the robustness of a recognition algorithm that is used in real-world applications. Access to training data where every possible activity is enumerated before deployment is impractical. Recent work in Machine Learning research has explored learning generalisable models from few examples (i.e., Few-Shot Learning), and beyond that to learning models that can adapt to recognise classes not seen during training, or Zero-Shot Learning (ZSL). However, existing ZSL algorithms in the context of Open-ended HAR are heavily reliant on expert domain knowledge, which introduces a demanding knowledge acquisition task. In this paper, we propose to extract a few seconds of raw calibration sensor data, which can be conveniently obtained from micro-interactions with the user, to replace expert domain knowledge. We introduce a ZSL algorithm, MN^Z, which is based on Matching Network architecture, that exploits similarities between this calibration data to perform Open-ended HAR. To prove the effectiveness of MN^Z, we conduct a comparative study on three HAR datasets. Our results confirm that MN^Z, with just 5 seconds of calibration data, performs 14-18% better when compared to recent ZSL algorithms that utilise domain expert input. In addition, our results emphasise the need for strategic selection of multiple sensors for reliable Open-ended HAR.

Introduction

Activity monitoring with wearable sensors is a popular digital health intervention strategy used in many health and well-being mobile applications. However automated recognition of human activities in current fitness applications (e.g. Google Fit, Apple Health) remains restricted to a set of predefined activities. When tracking new user-defined activities these applications rely on self-reporting by users which often leads to unreliable and inconsistent entries. Further, a study conducted in 2015 concluded that 58% of smart phone users in US downloaded health-care fitness applications on their mobile, and 47% of those who downloaded, stopped using them due to the high burden of data entry and loss of interest (Krebs and Duncan 2015).

The ability to incorporate new activities elegantly into pre-trained models after deployment remains an open challenge. Accordingly, researchers have recognised the need for Open-ended HAR (Kawaguchi et al. 2016) with a view to creating robust HAR applications that can be personalised to an individual's preferred set of activities. An important aim for Open-ended HAR is to extend HAR capacity for automated recognition to new activities with minimal calibration or input from the user.

Lazy learners (Aha, Kibler, and Albert 1991) e.g. kNN, are particularly well suited to open-ended classification due to their non-reliance on pre-trained models. However lazy learners are heavily reliant on feature engineering, typically using feature selection and weighting heuristics, to ensure that instances are represented appropriately (Lampert, Nickisch, and Harmeling 2014). Recent work in One-Shot Learning (Bart and Ullman 2005) and Transfer Learning (Weiss, Khoshgoftaar, and Wang 2016) has led researchers to explore the area of Zero-Shot Learning (ZSL) (Larochelle, Erhan, and Bengio 2008), where the model transfers its learning to unseen classes utilising high-level class descriptions. The above approaches to Open-ended applications are "knowledge-intensive"–that is, highly reliant on expert input to provide semantic knowledge. We explore approaches to reduce this contention for expert knowledge(i.e., knowledge-light).

With recent advances in deep metric learners (Hoffer and Ailon 2015) it is common practice to exploit feature embeddings that are iteratively refined during model learning. Matching Network (MN) is one such method that can be viewed as an end-to-end learner that generates a feature embedding function as a by product of learning to match instances (Vinyals et al. 2016). Specifically this function learns to generate disjoint feature representations for classes in a multi-class feature space. We plan to exploit this characteristic of MN to address Open-ended HAR as a ZSL problem.

Accordingly we make the following contributions:

- Introduce a new ZSL algorithm MN^Z that utilises raw calibration sensor data as a high level class descriptor;
- Compare and validate that our ZSL algorithm outperforms recent ZSL algorithms that use expert domain knowledge; and

- Confirm generalisability and superior performance with three HAR datasets.

The rest of this paper is organised as follows: Related Work section discusses current research in the area of Open-ended HAR and ZSL. Methodology section introduces our approach to Open-ended HAR and provides details of the MN^Z algorithm. We evaluate and present our findings in two subsequent sections A Comparison with knowledge-intensive ZSL and A Comparison with knowledge-light ZSL; followed by Conclusion.

Related Work

Open-ended Human Activity Recognition (HAR) aims to develop models that are able to recognise new activities encountered after deployment, that were not observed during training (Kawaguchi et al. 2016). Existing methods reported in literature fall under supervised and unsupervised approaches; where the former relies on some user supervision at deployment whilst the latter relies on concept change detection algorithms to recognise new activities.

Unsupervised methods by nature do not rely on labelling and are naturally suited for Open-ended HAR. Work in Online Leaning (Karp 1992) and Incremental Learning (Jantke 1993) has demonstrated, how with no supervision, new activities can be recognised. Here incremental updates to the clusters allow integration of new classes as instances are folded-in (Gjoreski and Roggen 2017) even after model deployment. However the absence of any supervision means that it is harder to recognise both long and short bursts of new activity classes with similar levels of recognition performance. Each activity type requires different sensitivity thresholds to be set depending on their expected activity cyclic length or duration of observed activities. Consequently, recognition is focused on one type at the expense of ignoring the other. In this paper we work with a spectrum of human activities: from short pose detection to; longer ambulatory activity recognition (such as walking and running); through to activities of daily living. Such a mixed range of different activity types requires different sensitivity thresholds to be accommodated and will naturally benefit from some limited supervision.

One of the most recent work on Open-ended HAR is an industrial pose recognition application (Ohashi et al. 2018) which introduces a k-Nearest Neighbour (kNN) based Zero-Shot Learning (ZSL) algorithm. They use deep convolutional models to predict a set of higher-level semantic features which are intermediary human movement classes. Thereafter pose recognition involves the mapping of aggregated predictions to individual poses using a set of heuristics. Here Open-ended HAR functionality is facilitated by adding a new, mapping heuristic, each time a new pose is encountered. Accordingly ZSL is enabled by integrating the new heuristic knowledge within the recognition pipeline with the pre-trained convolution models (i.e., no re-training is done after deployment).

This reliance on additional domain knowledge(i.e., knowledge-intensive) is the common approach to ZSL in multiple domains (Lampert, Nickisch, and Harmeling 2009; Liu, Kuipers, and Savarese 2011; Cheng et al. 2013; Lampert, Nickisch, and Harmeling 2014). Specifically for Open-ended HAR, the manual knowledge acquisition burden is less desirable. Our work also adopts the ZSL paradigm, but advocates instead, a "knowledge-light" approach for integrating new class knowledge. More specifically, instead of integrating mapping heuristics; we acquire limited amount of raw calibration data from the user (through micro-interactions). Instead of predicting intermediary movement classes, we redefine the mapping task as a matching task, to have better generalisable feature engineering from model training to deployment.

Supervised learning with few labelled data has been recently explored extensively in computer vision domain (Vinyals et al. 2016; Snell, Swersky, and Zemel 2017; Yang et al.). As an emerging topic in the domain of HAR, most recently it has been adopted successfully in (Sani et al. 2018), where only a few instances of activity data is used as training data. They impose personalisation in addition to learning with few examples with Matching Networks (MN) (Vinyals et al. 2016), which out-performed traditional HAR models. In this paper we also use MN, but explore how to adapt it for ZSL in the context of Open-ended HAR.

Methodology

In this section we introduce and formalise our approach to Open-ended HAR as a knowledge-light ZSL method inspired by Matching Networks.

Proposed Open-Ended HAR Method

Figure 1: Open-ended HAR with calibration data from micro-interactions with the user

Figure 1 illustrates our approach to Open-ended HAR with ZSL. Imagine user *A* who is physically active and a gym enthusiast downloads the Open-ended HAR application to her mobile phone. At this stage the application is only able to recognise a few common activities (e.g. walking, running, sedentary) modelled on a general population at design time. User *A* wants the application to automatically recognise activities she performs regularly but are not packaged in the generic design. She records a few seconds of calibration data for each new activity (e.g. rope jumping and dumbbell exercise) using sensors available on the wearable device. Subsequently the application extend its functionality into recognising these new activities using calibration data (sensor and activity label) in the future. Importantly the new data is minimal (i.e., knowledge-light) and is seamlessly integrated without updating the reasoning model. We formalise this approach next.

Matching Networks for Zero-Shot Learning

Matching Network (MN) can be viewed as an end-to-end neural implementation of the otherwise static kNN algorithm. The network iteratively learns to match a given query instance to one or few elements in a small set of instances called a support set (Vinyals et al. 2016). A support set contains both positive and negative matches to the query instance. Accordingly a MN train or test instance comprises both the query instance and its support set, in comparison to conventional supervised machine learning. An attention mechanism in the form of similarity weighted majority vote estimates the class distribution. The loss function is driven by the difference between the estimated and actual class distributions (quantified using cross entropy). This ensures that the network eventually learns a feature embedding function committed to generating feature representations that highlights the matches between query and support set instances. In other words the network learns to match.

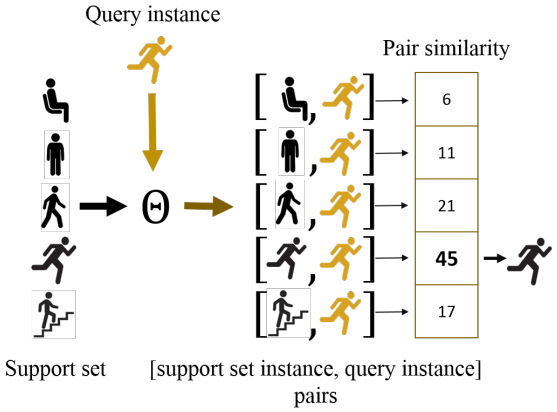

Figure 2: Matching Networks

Lets consider a dataset with a set of $\mathcal{X}$ activity instances belonging to set of $\mathcal{L}$ activity classes. Support set S is defined as in Equation 1. Cardinality of the support set is $k \times n_{tr}$, where k is the number of instances per class. n_{tr} is the number of classes in the support set and $n_{tr} \leq |\mathcal{L}|$.

$$S = \{(x, y)|x \in \mathcal{X}, y \in \mathcal{L}\} \tag{1}$$

The MN's training set, $\{(q_1, S_1), (q_2, S_2), \ldots, (q_N, S_N)\}$, has N elements, where each query instance, q_i, is a training instance pair, (x, y), and is never featured in its' support set, S_i (i.e. $q_i \notin S_i$). The MN classifier model, θ, learns to recognise the activity class, y, for a given query instance, x, relative to support set, S. At test time, the learnt model predicts label $\hat{y}$ for a query instance $\hat{x}$ guided by a support set S.

After deployment, the model has access to a few example instances, $\hat{\mathcal{X}}$, for a set of new activity classes, $\hat{\mathcal{L}}$, that were not seen during training of the model. We can view this as the user providing a small set of instances for calibration. We predict the label, $\hat{y}$, for test activity instance, $\hat{x}$, relative to a support set, $\hat{S}$. Accordingly we define, $\hat{S}$, as in Equation 2. Cardinality of set $\hat{S}$ is $k \times n_{te}$, where n_{te} is the number of classes in the support set after deployment.

$$\theta(\hat{x}, \hat{S}) \rightarrow \hat{y}$$

$$\hat{S} = \{(x, y)|x \in (\mathcal{X} \cup \hat{\mathcal{X}}), y \in (\mathcal{L} \cup \hat{\mathcal{L}})\} \tag{2}$$

With the original MN definition (Vinyals et al. 2016), n_{te} is restricted to the size of the training support set, ($n_{te} = n_{tr}$). With ZSL this forces the network to select a subset of classes from both training classes($\mathcal{L}$) and test classes($\hat{\mathcal{L}}$). This has the undesirable property that the set of possible combinations, grows exponentially with increasing numbers of new classes at deployment. As a result the support set may not include the class($\hat{y}$), which $\hat{x}$ belongs to, resulting in poor performance.

Figure 3 illustrates how the original MN fails with a fixed length support when used for ZSL. Here the green coloured icon denotes a new activity class encountered post-deployment. As shown in the figure, the absence of the expected class in the support set results in an incorrect classification outcome. One way around this is to try out several class combinations within the support set (potential for combinatorial explosion). The alternative is to expand the support size to cover as many as the expected number of classes that are encountered after deployment.

We explore the second option where the number of classes in the support set size is dynamic. Accordingly we introduce condition, $n_{te} \leq |\mathcal{L}| + |\hat{\mathcal{L}}|$. This facilitates inclusion of all available classes in the support set, as new classes are introduced to the model after deployment. This allows the classifier to make an informed decision when predicting activity, $\hat{y}$, relative to the support set. With this refinement we are able to use the network for Open-ended HAR.

Figure 4 illustrates our approach in detail. A new activity (the green activity icon) is introduced to the model with calibration data from the user. Ideally for personalisation purposes calibration data can be requested for every activity (if this is found to be feasible given the operational context). Importantly all classes (seen during training and testing) are represented and the model does not use the additional calibration data to update the model, but instead uses it as a "descriptor" for the new class. As further classes are introduced, the support set can grow to include them all when matching the query instances for classification.

A Comparison with knowledge-intensive ZSL

We first compare our method with one of the latest ZSL methods, based on kNN, which uses heuristics to aggregate and map predictions of movements to new activity classes (Ohashi et al. 2018). Two versions of this baseline algorithm are compared here: kNN based ZSL with mapping heuristics and kNN based ZSL with mapping heuristics and attribute importance. Note that mapping heuristics and attribute importance are externally acquired data from an expert. We use the following notation to refer to each algorithm:

- kNN^{Z+}: nearest-neighbour algorithm with mapping heuristics;

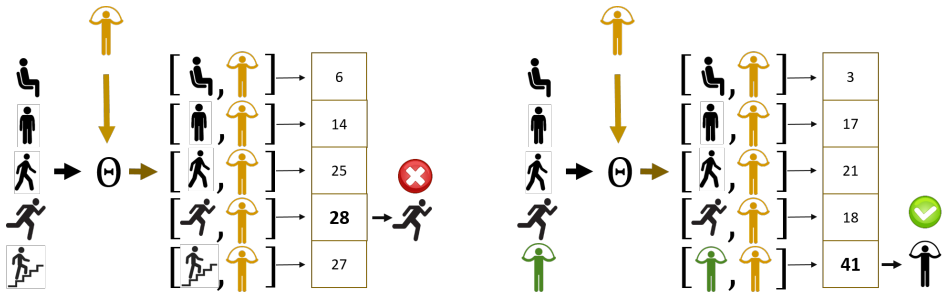

Figure 3: Zero-Shot Learning with Matching Networks

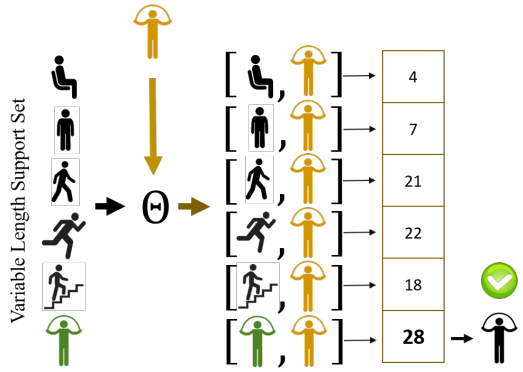

Figure 4: Matching Networks with a variable length support set

- KNN^{Z++}: same as KNN^{Z+}, additionally using attribute importance; and
- MN^{Z}: our algorithm formulated as described in the Methodology section, with no additional knowledge about unseen classes, except for the 5 seconds of calibration data for each new class.

Our aim in this comparison is to explore whether by learning to match, as in MN^{Z}, we can help reduce the burden on expert knowledge needed for aggregation, mapping and weights. We expect that the calibration data will be necessary and sufficient to reach comparable (if not better) performance to that with knowledge-intensive methods.

We use a simple feature embedding function for the MN^{Z}, with one fully connected layer of 1200 hidden units and batch normalisation. The similarity function used at the attention layer is Cosine similarity. We report F-measure as the performance metric. All hyper-parameters were selected empirically considering performance gain vs. computational intensity.

HDPoseDS Dataset

The human pose classification dataset HDPoseDS [1] is a rich dataset of wearable sensors, published in 2018 (Ohashi et al. 2018). The dataset contains 22 poses recorded with 10 participants, wearing 31 Inertial Measurement Units (IMU) over the full body. The data was recorded at 60Hz. Each IMU consist of a 3-axis accelerometer, gyroscope and magnetometer.

A sliding window of size 60 timestamps with no overlap is used to create instances; adapted from the publication for this dataset (Ohashi et al. 2018). Thereafter features are extracted from the 3-dimensional (x, y, z) raw accelerometer data instances to form a 1-dimensional Discrete Cosine Transform (DCT) feature vectors of size 90; adapting from (Sani et al. 2018) which simplifies feature representation.

Experimental Design

Experiments for both KNN^{Z+} and KNN^{Z++} use all 31 original sensors; with MN^{Z} we begin with 17 of the original sensors (approximately 54% of the sensors in the dataset) by excluding 14 sensors that are placed on the fingers of the subject. We exclude these finger sensors because they are comparatively more intrusive and redundant in a real-world setting. Three further experiments were designed for MN^{Z}, by gradually decreasing the number of sensors used as features. Table 1 lists the four sets of experiments; each referenced by the percentage of sensors used together with details of the sensors that are excluded. The intention behind this experimental design is two fold. First we wish to evaluate the trade-off between performance and number of sensors used, keeping in-mind that fewer sensors are more desirable in a practical application. Secondly, we compare our approach which intuitively minimises sensors with the rule-based approach (attribute importance) that is used by KNN^{Z++}.

To allow comparison of results withKNN^{Z++} and KNN^{Z+}, a Leave-One-Class-Out (LOCO) evaluation is used, maintaining consistency with those reported in (Ohashi et al. 2018). We remove data from one activity class when creating training instances and use data from the left-out activity as test data. This means that the number of experiments are equal to the number of human activities in the dataset. Accordingly we create $88 (= 22 * 4)$ experiments for the HDPoseDS dataset.

[1] A collaboration between Research & Development Group, Hitachi Ltd, and German Research Center for Artificial Intelligence (DFKI). More details and public dataset is available at http://projects.dfki.uni-kl.de/zsl/data/

Table 1: MN^Z Experiments with sensor set exclusions (short forms used for R-Right, L-Left sensors)

Experiment	Description of excluded sensor sets
54%	All 14 sensors placed on fingers
41%	All finger sensors + R&L UpLeg, R&L ForeArm
32%	All finger sensors + R&L UpLeg, R&L ForeArm, R&L Arm and Spine
19%	All finger sensors + R&L UpLeg, R&L ForeArm, R&L Arm, Spine, R&L Leg, R&L Shoulder

With MN^Z the support set and associated query is sampled from the same user to form each instance as described in (Sani et al. 2018). Note that each MN^Z instance is a query and support set pair, unlike with KNN^{Z+} and KNN^{Z++}. Accordingly the following strategy is used to create its train and test set instances:

- **Instances for training set:** A training set contains 21 of the 22 classes; and for each person, 500 queries are selected with stratified sampling from the HDPoseDS dataset. Each query is paired with a disjoint stratified support set to create the complete instance. Here 5 instances per class are sampled resulting in a support set of size 105 $(21 * 5 = 105)$.
- **Instances for test set:** 5 instances are sampled from each training class, as well as the test (unseen class) to create a support set of 110 instances ($5 * 21$ classes). The sampled support set simulates user-provided calibration data. This support set is used with each of the remaining test instances (not used in the support set). Accordingly, the number of test instances is 5 less than the total number of available instances in the test class.

Results

Table 2 is sorted by increasing performance of KNN^{Z++}. We use bold text to indicate the best result achieved for each experiment. Overall we can see that MN^Z consistently outperforms both KNN^{Z+} and KNN^{Z++}. MN^Z with 54% of the sensors achieves a maximum f-measure of 1.0 (on 85% of the experiments), with a minimum performance as high as 0.968, and an average f-measure of 0.998. Thereafter with only 6 sensors the average f-measure drops only by 3.6%, with the minimum performance with 6 sensors down to 0.712 (see column 19%).

Looking at each experiment in detail, we see that MN^Z with just 19% and 32% of sensors out-performs both baselines on 16 and 20 experiments respectively. This confirms that our approach of intuitively excluding sensors outperforms the rule-based approach used in KNN^{Z++}. In contrast, only on one experiment does a baseline marginally outperform MN^Z (i.e. activity HeelToBackL). Furthermore, the performance of MN^Z is much more reliable over all activity classes compared to both baselines, considering the range of f-measures obtained across all experiments.

Table 2: Comparison of KNN^{Z+} and KNN^{Z++} vs. MN^Z

Test class	KNN^{Z+}	KNN^{Z++}	MN^Z 54%	MN^Z 41%	MN^Z 32%	MN^Z 19%
WaistTwistingL	0.217	0.264	**1.000**	**1.000**	**1.000**	**0.997**
WaistTwistingR	0.155	0.293	**1.000**	**1.000**	**1.000**	**1.000**
FoldingArm	0.535	0.439	**1.000**	**0.995**	**1.000**	**0.990**
Standing	0.695	0.694	**1.000**	**1.000**	**1.000**	**0.993**
Sitting	0.680	0.744	**1.000**	**0.990**	**1.000**	**0.896**
Boxing	0.675	0.749	**1.000**	**1.000**	**0.990**	**0.997**
BaseballHitting	0.660	0.774	**1.000**	**1.000**	**1.000**	**1.000**
Skiing	0.780	0.783	**1.000**	**1.000**	**1.000**	**1.000**
StretchingCalfL	0.546	0.807	**1.000**	**1.000**	**0.986**	0.712
Thinking	0.780	0.823	**1.000**	**1.000**	**0.985**	**1.000**
StretchingForward	0.878	0.871	**1.000**	**1.000**	**1.000**	**0.982**
StretchingCalfR	0.596	0.890	**0.985**	**0.985**	**1.000**	**0.911**
RaiseArmR	0.966	0.952	**0.968**	0.965	**0.995**	0.945
WaistBending	0.974	0.961	**1.000**	**1.000**	**1.000**	**1.000**
PointingL	0.954	0.972	**1.000**	**1.000**	**0.984**	**0.992**
HeelToBackR	0.871	0.973	**1.000**	**1.000**	0.945	**1.000**
HeelToBackL	**1.000**	0.979	0.997	0.986	0.989	0.986
DeepBreathing	0.973	0.980	**1.000**	**1.000**	**1.000**	**1.000**
RaiseArmL	0.979	0.985	**1.000**	**1.000**	**1.000**	0.853
PointingR	0.999	0.995	**1.000**	**1.000**	**1.000**	**1.000**
Squatting	0.975	1.000	**1.000**	**1.000**	**1.000**	0.942
StretchingUp	0.986	1.000	**1.000**	**1.000**	**1.000**	0.970
Average	0.769	0.815	0.998	0.997	0.994	0.962

A Comparison with knowledge-light ZSL

In this section we evaluate MN^Z on two HAR datasets: PAMAP2 and SelfBACK. The purpose of this evaluation is two fold. Firstly, we wish to evaluate the performance of Open-ended Matching Networks in different domains of human activities, using two additional HAR datasets that contain activities of daily living, ambulatory activities and sedentary activities. Secondly, we improve our evaluation technique to simulate MN^Z being used in the real-world where the test user population is different from the training user population. As a baseline, we compare MN^Z with a knowledge-light lazy learner. We use the following notation to refer to each algorithm:

- KNN: k-Nearest Neighbour algorithm in an ZSL setting; and
- MN^Z: our algorithm formulated as described in the Methodology section.

Datasets

PAMAP2 Dataset PAMAP2 [2] is a Physical Activity Monitoring dataset which contains data from 3 IMUs located on wrist, chest and ankle recorded with 9 subject. Data was recorded approximately at $9Hz$ for 18 activity classes that are ambulatory, sedentary and activities of daily living (Reiss and Stricker 2012). We filter out one subject and

[2] https://archive.ics.uci.edu/ml/datasets/PAMAP2+Physical+Activity+Monitoring

10 activities with insufficient data. The refined dataset contained 8 subjects and 8 activity classes.

SelfBACK Dataset SelfBACK dataset for HAR [3] was compiled with a tri-axial accelerometer data streams for 9 ambulatory and sedentary activities. Activities were performed by 50 individuals where accelerometer was mounted on the right-hand wrist of each subject. Each activity was performed for approximately 3 minutes and data was recorded at $100Hz$ sampling rate.

As with the HDPoseDS dataset, we use a similar pipeline to pre-process and form paired instances for MN^Z. Some differences to hyper parameter settings were needed (such as values for sliding window size and DCT feature vector length) to accommodate inherent differences between activity types in each dataset. Here we use 500 timestamps with no overlap as the sliding window for both datasets; and a DCT feature vector length of 180 and 540 for SelfBACK and PAMAP2 respectively.

Experimental Design

We design our evaluation strategy by incorporating iterative hold-out testing to our LOCO evaluation approach. For each dataset we create n number of experiments with LOCO where n=number of classes. The train set and test set creation is similar to previous evaluation section, only difference is that the test data will belong to users not seen during training. We use data from $2/3$ of users selected randomly for training and rest of the users for testing (hold-out) and we repeat each hold-out test for 5 iterations.

We use number of instances per class (k) as 5, the same feature embedding function and same similarity function, Cosine similarity from the previous evaluation section for MN^Z. In a ZSL problem setting the baseline, kNN, will only have access to calibration data provided by the user for classes seen during training and classes not seen during training. Accordingly we use the support set for test data ($\hat{S}$) as possible neighbours for thekNN algorithm. This yields $n * k$ possible neighbours (40 for PAMAP2 and 45 for SelfBACK) for the kNN algorithm where we select majority voted class from first 5 neighbours. We report F-measure averaged over 5 repetitions as the performance measure.

Results

Here we present results we obtained for two HAR datasets PAMAP2 and SelfBACK and compare it against the baseline kNN algorithm. Results are presented in graphs where the columns are sorted by increasing performance of baseline kNN. We perform statistical significance test on these results to confirm their performance over the baseline algorithm. The error bars indicate the variability of performance for 5 iterations.

Figure 5 presents results for PAMAP2 dataset. The baseline algorithm's performance ranges from 0.678 to 0.962

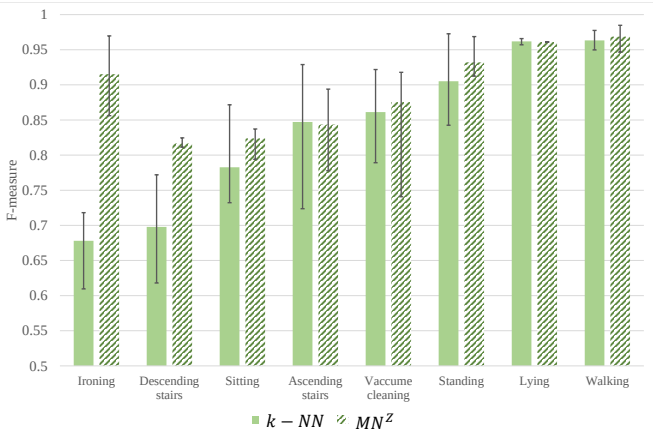

Figure 5: Comparison of kNN vs. MN^Z with PAMAP2 dataset

where the average f-measure is 0.837. With MN^Z the performance ranges from 0.817 to 0.969 where the average f-measure is 0.892. Two sample t-test performed at confidence level 95% confirm that our algorithm significantly outperform the kNN baseline. Individual performances with MN^Z algorithm show that 6 out of 8 experiments out perform kNN algorithm. In addition we achieve consistently good performance with MN^Z across all experiments with minimum performance being over 0.80.

It is evident that there is a significant decline of performance for PAMAP2 compared to HDPoseDS. We note that we are using a more rigorous evaluation strategy with PAMAP2 dataset; with hold-out test strategy, the users in the test set were not seen during training. More importantly the number of sensors available in the PAMAP2 dataset is only 3, located on the wrist, chest and ankle. This is 50% less sensor data compared to the minimum sensor configuration we experimented on with HDPoseDS dataset. With these results we recognise that number of sensors plays a significant role in Open-ended activity recognition. To further investigate the above observation, we next present results obtained with SelfBACK HAR dataset which has only one sensor data stream.

Figure 6 presents MN^Z results for ZSL experiments with the SelfBACK dataset compared against baseline kNN algorithm. The baseline algorithm results range from 0.484 to 0.978 where the average f-measure is 0.828. With MN^Z the results range from 0.544 to 0.986 where the average f-measure is 0.814. Although individual performances of MN^Z algorithm show majority of experiments out perform kNN algorithm, average f-measure suggests that kNN outperforms MN^Z. we confirm with two sample t-test that this difference is not statistically significant at confidence level 95%.

Once again our algorithm maintains fairly consistent performance across different experiments compared to the kNN algorithm. However, we observe that experiments where the test class is a form of walking such as walking downstairs or walking fast are challenging for the MN^Z al-

[3]The SelfBACK project is funded by European Union's H2020 research and innovation programme under grant agreement No. 689043. More details available: http://www.selfback.eu. The dataset associated with this paper is publicly accessible from https://github.com/selfback/activity-recognition

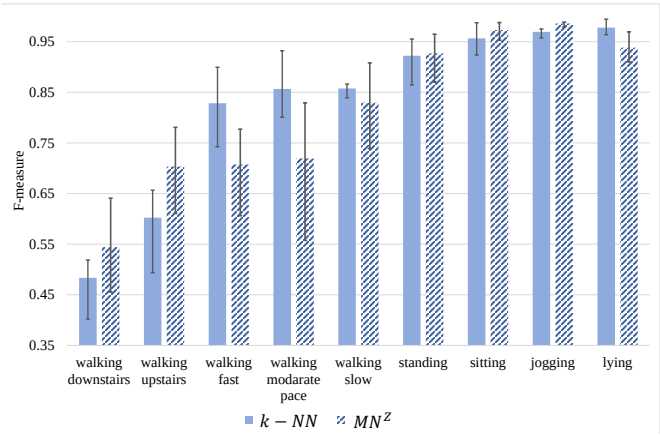

Figure 6: Comparison of KNN vs. MN^Z with SelfBACK dataset

gorithm. In SelfBACK dataset only one sensor was used to capture subject movements which was placed on the wrist. Naturally it only captured certain aspects of full body movement, resulting in ambiguous sensor data streams. Thus we realise that the similarity attention mechanism in MN^Z is struggling to differentiate between feature representations from different ambulatory activities. We also confirm this assumption by referring to results from traditional classification algorithms evaluated on this dataset (Sani et al. 2017). Accordingly we recognise the need for multiple sensors to capture multiple facets of full body movement to preserve reliable performance of Open-ended HAR.

Conclusion

Need for Open-ended Human Activity Recognition (HAR) has been emphasised in the last few years and Zero-Shot Learning (ZSL) has emerged as a viable approach for achieving Open-ended HAR. Current ZSL algorithms for Open-ended HAR exploit expert knowledge to build semantic descriptions for classes not seen during training (or knowledge-intensive). We propose using a few seconds of calibration data as high level descriptors for new classes after deployment (i.e., knowledge-light). To utilise this calibration data we propose a new ZSL learning algorithm inspired by Matching Networks. We evaluate our method against one of the latest knowledge-intensive ZSL learning algorithms and a knowledge-light lazy learner. Our results confirm that our knowledge-light approach to ZSL is consistently reliable for a wide range of new activities. Our approach eliminates the need for expensive expert knowledge input, which makes it a confident candidate for real-world Open-ended HAR applications. We also recognise that number of sensors and their placement is a major contributing factor to the performance of Open-ended HAR. It is important to find the right balance between number of sensors and model performance, as the success of such applications depends on the usability in a real-world setting.

References

Aha, D. W.; Kibler, D.; and Albert, M. K. 1991. Instance-based learning algorithms. *Machine Learning* 6(1):37–66.

Bart, E., and Ullman, S. 2005. Cross-generalization: Learning novel classes from a single example by feature replacement. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 1, 672–679. IEEE.

Cheng, H.-T.; Griss, M.; Davis, P.; Li, J.; and You, D. 2013. Towards zero-shot learning for human activity recognition using semantic attribute sequence model. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, 355–358. ACM.

Gjoreski, H., and Roggen, D. 2017. Unsupervised online activity discovery using temporal behaviour assumption. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*, 42–49. ACM.

Hoffer, E., and Ailon, N. 2015. Deep metric learning using triplet network. In *International Workshop on Similarity-Based Pattern Recognition*, 84–92. Springer.

Jantke, P. 1993. Types of incremental learning. In *AAAI Symposium on Training Issues in Incremental Learning*, 23–25.

Karp, R. M. 1992. On-line algorithms versus off-line algorithms: How much is it worth to know the future? In *IFIP Congress (1)*, volume 12, 416–429.

Kawaguchi, N.; Nishio, N.; Roggen, D.; Inoue, S.; Pirttikangas, S.; and Van Laerhoven, K. 2016. 4 th workshop on human activity sensing corpus and applications: towards open-ended context awareness. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct*, 690–695. ACM.

Krebs, P., and Duncan, D. T. 2015. Health app use among us mobile phone owners: a national survey. *JMIR mHealth and uHealth* 3(4).

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2009. Learning to detect unseen object classes by between-class attribute transfer. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, 951–958. IEEE.

Lampert, C. H.; Nickisch, H.; and Harmeling, S. 2014. Attribute-based classification for zero-shot visual object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 36(3):453–465.

Larochelle, H.; Erhan, D.; and Bengio, Y. 2008. Zero-data learning of new tasks. In *AAAI*, volume 1, 3.

Liu, J.; Kuipers, B.; and Savarese, S. 2011. Recognizing human actions by attributes. In *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 3337–3344. IEEE.

Ohashi, H.; Al-Naser, M.; Ahmed, S.; Nakamura, K.; Sato, T.; and Dengel, A. 2018. Attributes importance for zero-shot pose-classification based on wearable sensors. *Sensors* 18:2485.

Reiss, A., and Stricker, D. 2012. Introducing a new benchmarked dataset for activity monitoring. In *Wearable Computers (ISWC), 2012 16th International Symposium on*, 108–109. IEEE.

Sani, S.; Wiratunga, N.; Massie, S.; and Cooper, K. 2017. knn sampling for personalised human activity recognition. In *International Conference on Case-Based Reasoning*, 330–344. Springer.

Sani, S.; Wiratunga, N.; Massie, S.; and Cooper, K. 2018. Personalised human activity recognition using matching networks. In *International Conference on Case-Based Reasoning*. Springer.

Snell, J.; Swersky, K.; and Zemel, R. 2017. Prototypical networks for few-shot learning. In *Advances in Neural Information Processing Systems*, 4077–4087.

Vinyals, O.; Blundell, C.; Lillicrap, T.; Wierstra, D.; et al. 2016. Matching networks for one shot learning. In *Advances in Neural Information Processing Systems*, 3630–3638.

Weiss, K.; Khoshgoftaar, T. M.; and Wang, D. 2016. A survey of transfer learning. *Journal of Big Data* 3(1):9.

Yang, F. S. Y.; Zhang, L.; Xiang, T.; Torr, P. H.; and Hospedales, T. M. Learning to compare: Relation network for few-shot learning.