WIJEKOON, A., WIRATUNGA, N., SANI, S. and COOPER, K. 2020. A knowledge-light approach to personalised and open-ended human activity recognition. *Knowledge-based systems* [online], 192, 105651. Available from: http://doi.org/10.1016/j.knosys.2020.105651

A knowledge-light approach to personalised and open-ended human activity recognition.

WIJEKOON, A., WIRATUNGA, N., SANI, S. and COOPER, K.

2020

OpenAIR @RGU

This document was downloaded from https://openair.rgu.ac.uk

CC BY NC ND

A Knowledge-Light Approach to Personalised and Open-ended Human Activity Recognition

Anjana Wijekoon[a,*], Nirmalie Wiratunga[a], Sadiq Sani[a], Kay Cooper[b]

[a]School of Computing and Digital Media, Robert Gordon University, Aberdeen, UK
[b]School of Heath Sciences, Robert Gordon University, Aberdeen, UK

Abstract

Human Activity Recognition (HAR) is a core component of clinical decision support systems that rely on activity monitoring for self-management of chronic conditions such as Musculoskeletal Disorders. Deployment success of such applications in part depend on their ability to adapt to individual variations in human movement and to facilitate a range of human activity classes. Research in personalised HAR aims to learn models that are sensitive to the subtle nuances in human movement whilst Open-ended HAR learns models that can recognise activity classes out of the pre-defined set available at training. Current approaches to personalised HAR impose a data collection burden on the end user; whilst Open-ended HAR algorithms are heavily reliant on intermediary-level class descriptions. Instead of these "knowledge-intensive" HAR algorithms; in this article, we propose a "knowledge-light" method. Specifically, we show how by using a few seconds of raw sensor data, obtained through micro-interactions with the end-user, we can effectively personalise HAR models and transfer recognition functionality to new activities with zero re-training of the model after deployment. We introduce a Personalised Open-ended HAR algorithm, MN^{Z}, a user context aware Matching Network architecture and evaluate on 3 HAR data sources. Performance results show up to 48.9% improvement with personalisation and up to 18.3% improvement compared to the most common

*Corresponding author
Email address: `a.wijekoon@rgu.ac.uk` (Anjana Wijekoon)

"knowledge-intensive" Open-ended HAR algorithms.

Keywords: Human Activity Recognition, Personalised HAR, Open-ended HAR, Zero-Shot Learning, Matching Networks

1. Introduction

Physical 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) remain restricted to a set of pre-defined activities modelled on a general population. Personal physical activity traits such as activity preferences and patterns, gait or posture cannot be incorporated in to these applications. In addition, when tracking new user-defined activities these applications rely on self-reporting by user which often lead to unreliable and inconsistent entries. Further, a study conducted in 2015 concluded that out of 58% of smart phone users in the US who downloaded healthcare fitness applications on their mobile phones, 47% of them stopped using these apps due to the high burden of data entry and loss of interest [1].

A Machine Learning model that performs Human Activity Recognition (HAR) is the main computation module that underpins these activity monitoring applications and they utilise available wearable sensor data to perform a classification task to recognise activities in real time. These models are pre-trained on sensor data gathered from a general population and remain restricted to a pre-defined number of activity classes.

An important consideration for HAR is classifier training, where training examples can either be acquired from a general population (user-independent), or from the target user of the system (user-dependent). Previous research has shown that using user-dependent data results in superior performance [2, 3, 4, 5]. The relatively poor performance of user-independent models can be attributed to variations in activity patterns, gait or posture between different individuals [6]. However, training a classifier exclusively with user provided data is not practical in a real-world configuration as this places a significant burden on the user to provide sufficient amounts of training data required to build a personalised model. Minimising this data collection burden whilst maintaining comparable performance is challenging and recent work in few-shot learning is directly relevant to addressing this challenge [7].

The ability to incorporate new activities elegantly into pre-trained models after deployment also remains an open challenge. Accordingly, researchers have recognised the need for Open-ended HAR [8] 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 user input. Recently researchers have explored the area of Zero-Shot Learning (ZSL) [9] where the model transfers its learning to unseen classes after deployment, utilising an intermediary level of class descriptions. In the domain of wearable sensor based HAR, these class descriptions are built manually through expert domain knowledge (such as class-attribute mappings) [10, 11]. Visual data (such as video) based HAR commonly follow unsupervised learning approaches where the class descriptions are learnt from a knowledge base such as a text corpus [12, 13]. These approaches are "knowledge-intensive"– that is, they are highly reliant on the availability of intermediary semantic knowledge that is acquired through a demanding knowledge acquisition task, which is undesirable in real-world settings.

In this article, we introduce an approach to personalised Open-ended HAR using Matching Networks (MN). MN is a neural network architecture that was introduced for the task of one-shot learning by [7]. The capability of this network to learn from few examples is exploited here to minimise the demand on users to provide training data for personalisation. Furthermore by extending the one-shot method to a zero-shot method we are able to transfer the learnt matching model to activity classes that were unseen during training. We refer to this as a "knowledge-light" Open-ended HAR approach, and conduct a comparative study to establish its utility as a promising contender for real-world deployment.

In essence, adopting MN for personalised and Open-ended HAR, would require the user to provide only a small number of examples for each activity he/she wants regardless of whether or not these activities were all part of the model training phase.

Accordingly we make the following three contributions:

- Introduce a "knowledge-light" personalisation algorithm MN^P for HAR that minimises the burden of data collection on the end-user.
- Introduce a "knowledge-light" ZSL algorithm MN^Z for personalised Open-ended HAR that relies only on data obtained through micro-interactions with the end-user.
- Provide a comprehensive evaluation of MN^Z for a wide range of human activities across three HAR application data sources.

The rest of the article is organised as follows: Section 2 discusses current research and challenges in the areas of Personalised HAR and Open-ended HAR. Section 3 introduces our approach to personalisation and Open-ended HAR with a use case scenario. Section 4 introduces Matching Networks and formulates our approaches with MN^P and MN^Z architectures. We present our experiments (Section 5) and our findings(Section 6) in subsequent sections; followed by the Discussion in Section 7 and planned future work and conclusions in Section 8.

2. Related Work

In this section we outline related literature in personalisation and Open-ended Human Activity Recognition (HAR) with focus on data and knowledge requirements.

2.1. Personalised Human Activity Recognition

Personalising a HAR algorithm is desirable for physical activity monitoring applications where personal nuances such as gait patterns and posture can be integrated in to recognition tasks. In literature there are two common approaches to personalising HAR: user-dependent modelling which utilises significant amounts of end-user data for model training and semi-supervised learning which utilises a limited amount of end-user data to bootstrap a generic pre-trained model.

We find that most conventional approaches typically adopt training with user-dependant data. Early literature from [5, 3] and [2] report performance improvements of 39.3%, 9.7% and 19.0% respectively with classification algorithms trained with user-dependent data over the same algorithms trained with user-independent data. More recently, online Multi-task Learning (OMTL) approaches have reported further improvements in performance [14]. OMTL's treat each individual user as a task and all tasks are trained together as a multi-task classifier in order to influence each other. However with all these approaches performance gains are offset by the demand for end-user data (e.g. in order to cater for the increased number of tasks), resulting in limited applicability for real-world personalised HAR deployment.

More recently, semi-supervised learning has been explored as an alternative to user-dependent personalised models where smaller sets of personal data are used to re-train the model after deployment. Self-learning, co-learning and active learning, are a few semi-supervised learning approaches that have been used successfully to personalise HAR [6]. For instance an active learning framework that employs heuristics, uses feedback from the user to bootstrap a personalised HAR model [15], achieved an 8.5% performance improvement compared to non-personalised models. With this approach, a classification model is re-trained in real-time when new data instances are encountered. This is computationally intensive and the consumption of significant power makes them a less desirable solution for mobile platforms. In addition, performance gains were only observed when compared to weaker baselines with hardly any improvement observed against stronger baselines [6].

In this paper we use Matching Networks (MN) [7] as an alternative to address challenges related to user-dependant training and personalisation. MN was introduced by [7] for One-shot and Few-shot Learning in image recognition where MN out-performs the state-of-the-art. MN has also comparatively out-performed k-NN, SVM and MLP algorithms in the HAR task [16] and here we further enhance MN as a method of personalisation and for Open-ended HAR. In particular new training and test strategies are introduced for MN to enable learning from few examples thereby eliminating the requirement for large data collections. Furthermore it is trained to generalise learning from a few data instances from a given user to address the personalisation requirement. Importantly end-user data can be conveniently integrated within the classifier following deployment with zero re-training of the model.

2.2. Open-ended Human Activity Recognition

Open-ended Human Activity Recognition (HAR) aims to develop models that are able to recognise new activities encountered after deployment, and that were not observed during training [8]. Existing methods reported in literature fall under unsupervised and supervised approaches; where the former relies on concept change detection algorithms to recognise new activities whilst the latter relies on semantic knowledge to describe unseen classes.

Unsupervised methods such as clustering, by nature do not rely on labelling and are naturally suited for Open-ended applications. Incremental updates to the clusters allow integration of new classes as instances are folded-in [17] 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 rest. Here, 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. We expect that having a mixed range of different activity types of this nature is likely to require different sensitivity thresholds to be accommodated and will naturally benefit from some limited supervision.

Recognising classes not seen during training as a supervised learning problem is often referred to as Zero-shot Learning (ZSL). ZSL exploits semantic knowledge of classes in both HAR [18, 10, 19, 12, 13] and computer vision [20, 21]. Acquisition of semantic knowledge is explored mainly in two methods; manually produced by an expert or learnt via an unsupervised method using an expert knowledge base such as a text corpus.

An activity-attribute matrix is the most common intermediary semantic knowledge space seen when achieving Open-ended HAR [10, 19] with wearable sensor data. An activity-attribute matrix is an intermediary semantic knowledge-base used in achieving open-ended HAR [10, 19, 11]. Such a matrix provides domain expert knowledge in which a high-level activity class is described by a sequence of intermediate-level activity attributes (hence intermediary semantic knowledge-base). With an activity-attribute matrix, the Open-ended HAR functionality is facilitated by adding a new, mapping heuristic, each time a new activity class is encountered. These algorithms use lower-level classification models to predict attributes of the semantic knowledge space, then aggregate those predictions in to a high-level class using similarity based algorithms. For instance, in [19] a new unseen activity such as a chest-press exercise can be added (at deployment) by describing it as a sequence of known action primitives (such as Arms side, Arms curl and Arms forward). This approach was later improved to incorporate temporal aspects of attribute sequences [10]. More recently researchers [11] applied Open-ended HAR for industrial pose recognition, where they introduced a similarity based ZSL algorithm. They used a deep convolutional model to predict a set of lower level semantic features consisting of intermediary human movement classes (or movement primitives). Thereafter pose recognition involved the mapping of aggregated predictions to individual poses using a set of heuristics with no re-training after deployment. The key idea is that models are learnt to predict the primitives and as long as new activities can be described using a sequence of these learnt primitives then the open-ended functionality of HAR is supported. Clearly the challenge with such a strategy is to ensure that all potential action primitives are included and thereafter ensuring a representation formalism is available to describe activities using these primitives.

Unsupervised semantic knowledge acquisition is commonly used in video based Open-ended HAR [12, 13]. The intermediary semantic knowledge space consists of text descriptions for all activity classes, that is learnt in an unsupervised manner from a domain knowledge source. Both [12] and [13] learn the semantic embedding space using a Google News dataset with over 100 billion words. A lower level supervised learning component converts video data in to text (similar to caption generation) and later a similarity based algorithm selects the activity class from the semantic space that best matches the generated text. Again these algorithms are build on the assumption that the semantic embedding space essentially includes all possible activity classes.

Although unsupervised approaches to Open-ended HAR is comparatively less burdensome in knowledge acquisition (compared to the manual task), it is still challenging to adapt this approach to wearable sensor based HAR due to the non-visual nature of sensor data. In addition, the performance of an unsupervised approach depends on the completeness of the intermediary semantic space and therefore provides no opportunity for personalisation. Evaluation of these existing Open-ended HAR algorithms is challenging due to their "knowledge-intensive" nature. Different evaluation approaches that are being adopted by researchers has resulted in non-reproducible and non-comparative studies [22, 23]. We argue that one of the main drawbacks of existing Open-ended HAR algorithms is the dependency on an intermediary semantic knowledge space. Specifically for wearable based Open-ended HAR, we recognise that the application of existing algorithms on new datasets (for comparative purposes) is limited due to the unavailability of expert domain knowledge needed to derive the required semantic knowledge space unless explicitly made available.

Our Open-ended HAR algorithm 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 a limited amount of raw sensor data from the user (through micro-interactions). We redefine the mapping task (low-level attributes and intermediary semantic space) as a matching task between feature spaces, to have better generalisable feature engineering from model training to deployment. Consequently our algorithm can also be conveniently evaluated with any HAR dataset with no burden of acquiring semantic mapping knowledge.

We look at similarity based learning algorithms to implement personalised Open-ended HAR in a "knowledge-light" manner. Similarity based learning or Metric Learning was first explored with Siamese Neural Networks [24, 25] where the network learns from positive and negative instance pairs to iteratively refine an embedding function that learns a metric space. Later Triplet Networks [26] and Matching Networks [7] incorporated multiple negative and positive instances in to training examples which improved the training efficiency and the diversity of the resulting feature embedding function. Multiple advancements were made to Matching Networks which introduced variations such as Prototypical Networks [27] and MAML [28], but the fundamental concept of similarity based matching remained constant. Accordingly in this work we exploit Matching Networks and its capability to find similar instances in a multi-class feature space to achieve a knowledge-light approach to Personalised Open-ended HAR.

3. Use case

In this section we will present a detailed use case of our solution to building a personalised fitness application that recognise custom activities according to user preference. This use case is illustrated in stages in Figure 3 where blue icons indicate personalisation of existing activities; green icons indicate introducing new activities to the model with personalisation; yellow icons indicates an incoming query in real-time for classification.

Imagine a user who is physically active and a gym enthusiast, downloads the Open-ended HAR application to her mobile phone. The user wants to per-

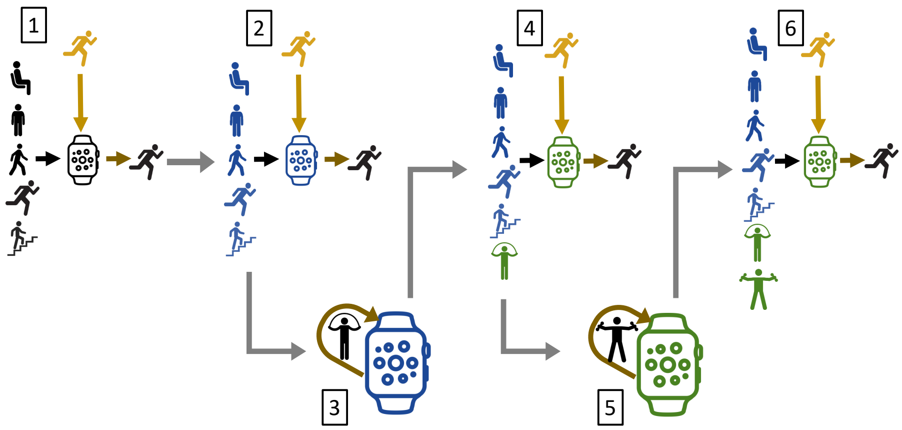

Figure 1: Performing Open-ended HAR utilising few calibration data obtained from micro-interactions with the end-user

sonalise the application and 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 existing activity and new activities using sensors available on the wearable device. Subsequently the application is personalised and extended to recognise these new activities using calibration data (sensor data and corresponding activity labels) in the future.

Stage 1: At this stage the application is only able to recognise a set of five common activities (walking, running, sitting, standing and ascending stairs) modelled on a general population.

Stage 2: She starts personalising the application by recording 30 seconds of herself performing each of the pre-packaged 5 activities that is already supported by the application. With this personal data the application incorporates personal user traits when recognising activities already modelled by the application.

Stage 3: She realises that rope jumping is not one of the activities automatically recognised by the application. She performs 30 seconds of rope jumping while wearing the wearable devices that are connected to the mobile application, and at the end, she labels the data as rope jumping.

Stage 4: Thereafter the application automatically recognises rope jumping in addition to the five activities it originally recognised.

Stage 5: After a while the user adds bicep curls to her daily routine of exercises and wonders if the mobile application can keep track of her performance. She performs 30 seconds of bicep curls while wearing the wearable devices that are connected to the mobile application, and at the end, she labels the data as bicep curls.

Stage 6: Once again the application automatically recognises bicep curls in addition to the six activities it was already recognising before.

Importantly the new data is minimal (i.e., knowledge-light) and is seamlessly integrated without updating the reasoning model (i.e., no model re-training after stage 1), which minimises the computational requirements and energy consumption of the mobile application.

4. Method

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

4.1. Matching Networks

Matching Networks (MN) [7] can be viewed as an end-to-end neural implementation of the otherwise static kNN algorithm. The network learns to generate a disjoint feature space by iteratively matching a query instance to a support set, which contains both positive and negative matches to the query instance. It is essentially “training to match” which is what sets it apart from conventional supervised learning models. Further this training characteristic is what makes addition of new examples possible with no re-training of the model.

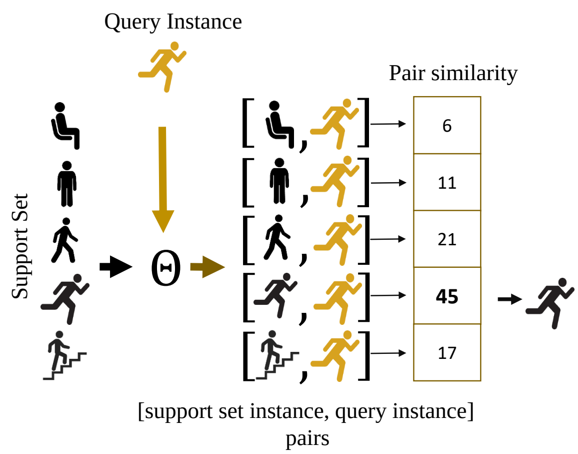

Figure 2: Classification with Matching Networks

Lets consider a dataset with a set of $\mathcal{X}$ activity instances belonging to a set of $\mathcal{L}$ activity classes. The 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 $n_{tr} \leq |\mathcal{L}|$.

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

Given an MN training set, $\{(q_1, S_1), (q_2, S_2), \ldots, (q_m, S_m)\}$, with m instances, we can observe that each MN instance consists of a query, q_i and an associated support set, s_i, such that $q_i \notin S_i$. Here q is a pair (what we would normally refer to as a training instance in conventional supervised learning), (x, y), where x is a raw feature vector and y its class label. The feature embedding function, θ (a neural network model), transforms all support set instances and the query instance into feature vectors (Equation 2).

$$\theta(x) = x' \tag{2}$$

Similarity between all query instance and support set instance pairs are calculated with an appropriate similarity metric. (For instance, Cosine Similarity is shown in Equation 3)

$$sim(x', x'_i) = \frac{\sum x'_j x'_{i,j}}{\sqrt{x'^2_j}\sqrt{x'^2_{i,j}}} \tag{3}$$

Finally an attention mechanism in the form of similarity weighted majority vote estimates the class distribution, $\hat{y}$. (Equation 4 and 5).

$$a(x', x_i') = \frac{e^{sim(x', x_i')}}{\sum^{|s|} e^{sim(x', x_i')}} \tag{4}$$

$$\hat{y} = \sum^{|S|} a(x', x_i') \times y \tag{5}$$

During training, the network iteratively updates weights of θ to maximise the pair similarity between the query instance and support set instances that belong to the same activity class. This is enforced by the loss function, categorical cross-entropy, which quantifies the difference between the estimated and actual class distributions (Equation 6).

$$Loss = \sum_{j}^{n_{tr}} y_j log(\hat{y}_j) \tag{6}$$

Essentially the concept of "learning to match" is facilitated by the attention layer where attention is focused on pair-wise similarity computations; which in turn influences the network's back propagation and consequent weight updates. This means that the embedding function that is learnt is optimised for matching which is a proxy to class prediction.

After deployment (Figure 2), the model predicts the label $\hat{y}$ for a query instance $\hat{x}$ with respect to its support set $\hat{S}$ (Equation 7). In other words, the network learns to retrieve the best match from the support set elements, thereafter using them with weighted voting to predict the class.

$$\hat{y} = argmax_y P(y|\hat{x}, \hat{S}) \tag{7}$$

4.2. Personalised Matching Networks for HAR

In comparison to computer vision applications, HAR has an additional dimension to its data which is the user. We plan to incorporate this additional knowledge in order to personalise the classification task. We update the MN training and test methods to incorporate this additional dimension and as a

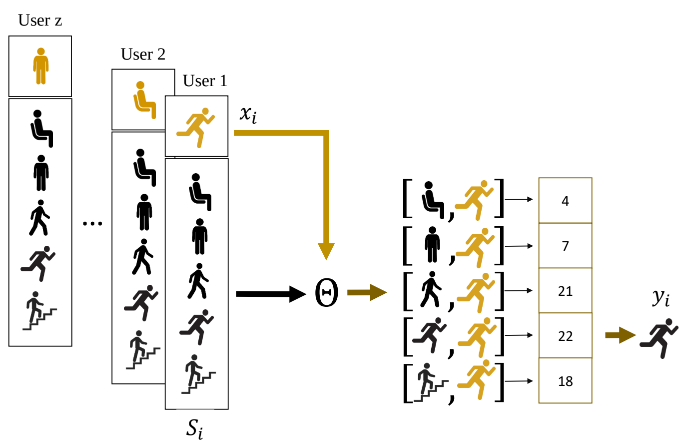

Figure 3: Training Personalised Matching Networks

result build a personalised version of MN that is better suited for HAR. We re-design the training and test sets such that a (q_i, S_i) pair is always constructed with data belonging to a single user. Accordingly the support set S_i will contain positive and negative instances from the same user to whom q_i belongs to, and in this way the model is trained to learn matching for personalisation. Note that by having to focus on query and support sets from the same user the model is forced to focus on traits that are important for recognising activities given user nuances. The resulting network will classify a particular user's activities using a small set of examples provided by the same user.

4.3. Matching Networks in an Open-ended Setting

In an Open-ended environment, after deployment, we expect a situation where the model can have 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. Thereafter the model is expected to recognise all activity classes in both $\mathcal{L}$ and $\hat{\mathcal{L}}$.

With the original MN definition [7], n_{te} is restricted to n_{tr}. In an Open-ended setting, 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.

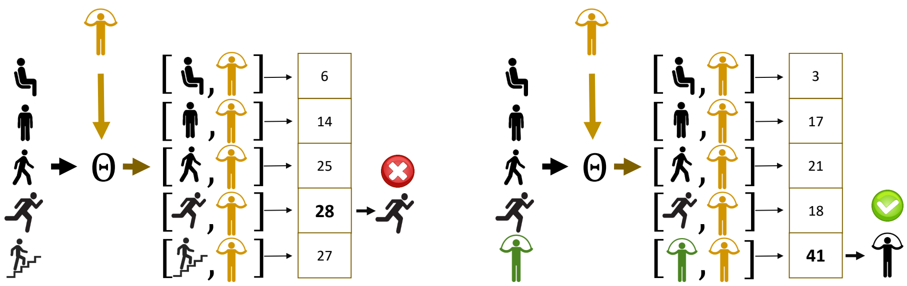

Figure 4: Two scenarios of random support set selection to perform Open-ended HAR with original Matching Network (fixed support set length) at deployment; left - support set does not contain the activity class query belongs to; right - support set contains the activity class query belongs to

Figure 4 illustrates how the original MN fails with a fixed length support when used for ZSL. Here the green coloured icon indicates the new activity class introduced post-deployment. Now there are 6 possible ways ($nCr = n!/r! \times (n-r)!$) to select the support set and Figure 4 shows two scenarios. It is evident that 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 set size to include as many as the expected number of classes that are available after deployment. We explore the second option in the next section where the number of classes in the support set size is dynamic.

4.4. Open-ended Matching Networks

We introduce a condition on Equation 8, which facilitates inclusion of all available classes in the support set, as new classes are introduced to the model after deployment (Equation 9), where the cardinality of set $\hat{S}$ is now $k \times n_{te}$.

With this refinement we are able to use the originally trained network for Open-ended HAR after deployment.

$$n_{te} \leq |\mathcal{L}| + |\hat{\mathcal{L}}| \tag{8}$$

$$\hat{S} = \{(x, y)|x \in (\mathcal{X} \cup \hat{\mathcal{X}}), y \in (\mathcal{L} \cup \hat{\mathcal{L}})\} \tag{9}$$

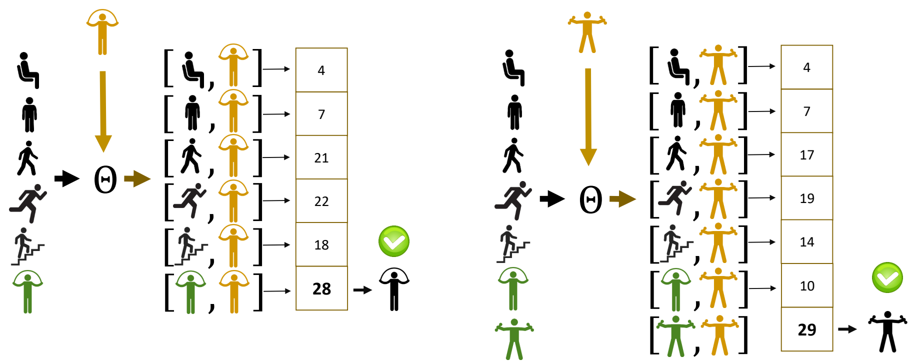

Figure 5: Open-ended HAR with Open-ended Matching Networks at deployment; support set length is variable to include all known activity classes as they are introduced by the user

Figure 5 illustrates the Open-ended Matching Networks architecture after deployment. New activities (the green activity icons) are introduced to the model with a few calibration data from the user. Ideally for personalisation purposes calibration data can be requested for each activity (if this is found to be feasible given the operational context). Importantly, all classes (seen during training and introduced after deployment) are represented in the support set and the model θ does not use the additional calibration data to update itself, but instead uses them as "descriptors" for new classes. As further classes are introduced, the support set includes them all when matching the query instances for classification.

5. Evaluation

Next we discuss our datasets, data pre-processing, model architectures and experiment methodologies.

5.1. Datasets

We consider three HAR data sources to evaluate our methods; we have selected these datasets as they collectively represent a wide range of human activities.

5.1.1. HDPoseDS Dataset

The human pose classification dataset HDPoseDS [1] is a sensor-rich dataset published in 2018 by [11]. The dataset contains 22 activities (poses and sedentary activities) recorded with 10 participants, wearing 31 Inertial Measurement Units (IMU) over the full body. The data was recorded at 60Hz where each IMU consists of a 3-axis accelerometer, gyroscope and magnetometer. This is a sensor-rich dataset which can be challenging to replicate in real-world applications. Therefore we plan to evaluate our methods against more restricted sensor configurations derived form this dataset. We remove sensors considering their redundancy and intrusiveness in the real-world while maintaining the full body sensor coverage. Accordingly we create two versions; we first exclude all 14 sensors on the fingers, resulting dataset with 17 sensors is the first version. We further eliminate 11 sensors to create the second version. We will use the following notation to refer to the two datasets.

- $HDPoseDS_{17}$: Dataset with 17 sensors after removing all 14 sensors placed on fingers.
- $HDPoseDS_{6}$: Dataset with only 6 sensors: on right and left hands, right and left feet, head and hip.

5.1.2. PAMAP2 Dataset

PAMAP2 [2] is a Physical Activity Monitoring dataset which contains data from 3 IMUs located on wrist, chest and ankle. Data was recorded with 9 users approximately at $9Hz$ for 18 activity classes by following a pre-defined

[1] [11] –Public dataset available at http://projects.dfki.uni-kl.de/zsl/data/

[2] [29] –Public dataset available at http://archive.ics.uci.edu/ml/datasets/pamap2+physical+activity+monitoring

protocol. Activities include that are ambulatory, sedentary and activities of daily living [29]. One user and 10 activities were filtered out of this dataset due to insufficient data. The refined dataset contained 8 users and 8 activity classes.

5.1.3. SelfBACK Dataset

SelfBACK dataset for HAR [3] was compiled with a tri-axial accelerometer data streams for 9 ambulatory and sedentary activities. Each activity was performed for approximately 3 minutes and data recorded at $100Hz$ sampling rate. We consider following two versions of the dataset, one with two sensors and the other with one sensor.

- SelfBACK$_{W,T}$- Data from 34 users where 2 accelerometers were mounted on the right wrist and the right thigh.
- SelfBACK$_{W}$- Data from 50 users where an accelerometer was mounted on the right wrist.

5.2. Pre-processing

Following pipeline was used to pre-process and form instances where an input raw signal is progressively converted to a vector, x (a single sensor pre-processing scenario is illustrated in Figure 6).

1. Use a sliding window with no overlap to segment the original raw sensor signal.
2. Extract 3-dimensional (x, y, z) raw accelerometer data from each sensor.
3. Apply Discrete Cosine Transformation (DCT) and extract most significant features from each dimension.
4. Concatenate all DCT feature vectors from each dimension of all sensors to form the final feature vector.

Some differences to hyper parameter settings were needed (such as values for sliding window size and DCT feature vector length) to accommodate the

[3] [30] –Public dataset available at https://github.com/rgu-selfback/Datasets

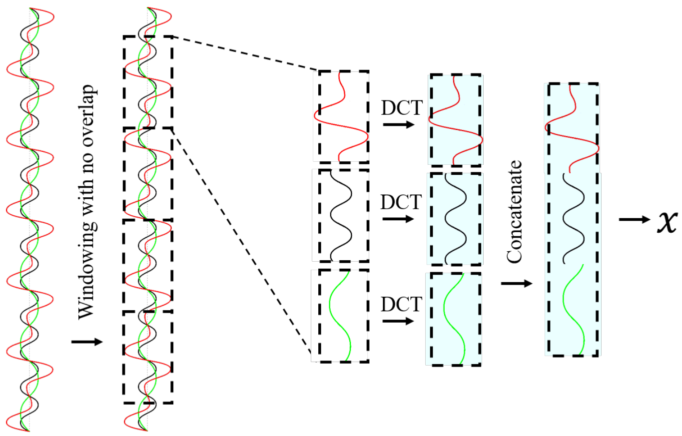

Figure 6: Pre-processing steps for a single sensor stream

inherent differences between activity types in each dataset (based on prior work in [16] and [30]). Refer to Table 1 for these details on each of our datasets.

Table 1: Datasets and pre-processing

Property	HDPoseDS		PAMAP2	SelfBACK	
	HDPoseDS$_{17}$	HDPoseDS$_{6}$		SelfBACK$_{W,T}$	SelfBACK$_{W}$
Number of Sensors	17	6	3	2	1
Number of Activities (n)	22	22	8	9	9
Number of Users	10	10	8	34	50
Sampling Frequency	60Hz	60Hz	9Hz	100Hz	100Hz
Sliding Window (timestamps)	60	60	500	500	500
DCT feature length	30	30	60	60	60
Final feature length	4590	540	540	360	180

5.3. Matching Networks Architecture and hyper-parameters

A set of empirical experiments were conducted to determine the most effective hyper-parameters for the original MN architecture in the HAR domain. We maintain these hyper-parameters constant across MN, MN^{P} and MN^{Z} in our comparative studies. Our choice of hyper-parameters is influenced by performance gains whilst maintaining moderate computational overhead.

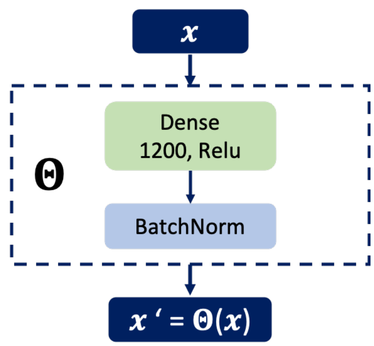

Figure 7: Feature embedding function for MN

Firstly, θ in our model, consists of a single-hidden layer (1200 units) fully connected neural network with Batch Normalisation (Figure 7). We use Keras [4] python libraries to implement our algorithms and a Batch Normalisation layer to normalise output which also acts as a regulariser against over-fitting [31]. The network is trained for 20 epochs with an "Adam" optimiser (learning rate = 0.001) using categorical cross-entropy as the loss function. Cosine similarity is used as the metric in the attention layer.

Secondly we explore the most effective k value for the MN architecture. We perform an empirical study with eight distinct values ranging from 1 to 10 (on all five datasets). We observe a consistent improvement of performance with higher k values, which also increases the computational overhead. We select k=5 as it exhibit the best compromise between them. Full details of this study is presented in Appendix A and is based on prior work in [16].

5.4. Evaluation Methodology

We performed each experiment as a user hold-out experiment on the chosen train and test split ratios of 2/3 and 1/3, repeated five times with a random

[4] https://keras.io

selection of test users. We re-use the same evaluation methodology from [11] and [21] to produce comparable results against the baseline algorithms. The user hold-out test strategy also ensures the performance of the test user is not influenced by the users' personal traits learnt during model training. We report mean accuracy or mean F-measure with statistical significance testing for experiments with an existing baseline using one-tailed t-test at 95% confidence level.

We performed a set of experiments with different train and test split ratios with a view to understanding the MN over-fitting behaviour with limited training examples. We observe up-to 9.79% degradation of accuracy when train set ratio was reduced from 2/3 to 1/3, which suggests that the model exhibits a considerable over-fitting to training data even with the regularisation used in the feature embedding function. Full details of these experiments are included in Appendix B.

5.5. Personalised Human Activity Recognition

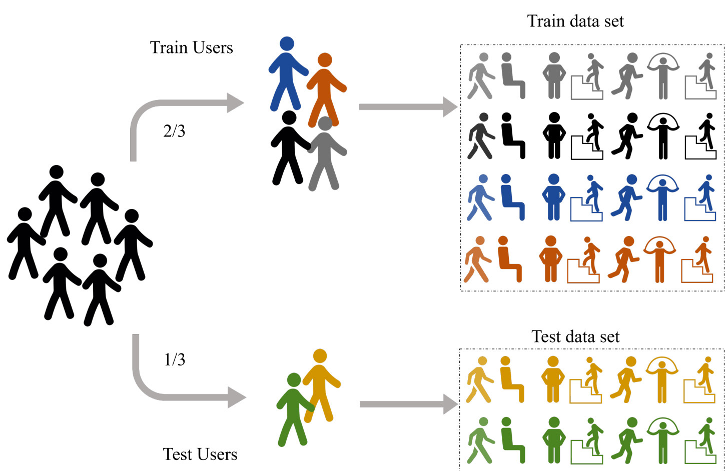

Figure 8: User hold-out validation strategy for Personalised Matching Networks

We compare original (MN [7]) vs. our Personalised Matching Networks (MN^P) architectures for HAR. The aim of this experiment is to observe the effect of personalisation for HAR with Matching Networks as illustrated in Section 3 stage 1 and 2.

Figure 8 shows how each of the randomly formed holds ensure no overlap in users (i.e. users are shown in different colours and colours do not overlap in train and test). Further all classes (e.g. walking, running) in training also appear in testing. Once the training users are separated from the test users; we create instances per user for the personalised and non-personalised matching network configurations as follows:

MN [7]: A training set contains $500 \times number\_of\_train\_users$ number of query instances. This means for each user we have 500 instances where each instance is created by randomly sampling a query instance and thereafter randomly sampling its paired disjoint support set (without replacement) from the train user population. We ensure that all classes are represented by k instances within each support set where $k = 5$. To form the test instance we first sample the test support set from test user population and use the rest of the test instances as query instances. Later we pair each test query instance with the test support set to create complete test instances.

MN^{P} The main difference when creating training instances for the personalised version is that when forming instances for the matching network, we ensure that the 500 instances created for each train user is sampled by accessing data from that user alone. This means that both the query and the paired support set are sampled from a subset of the training data associated with the same user. As before we ensure that the query and support set pairs are disjoint and we use $k = 5$. For creating test instances, we first sample a test support set for each user, then pair it with its own query instances. This way we ensure the support set and a query instance is dis-joint and each query, support set pair does belong to the same test user.

5.6. Personalised Open-ended Human Activity Recognition

With the open-ended HAR experiments we need to simulate the encountering unseen classes at test time. Accordingly we adopt the conventional setting described in [22] where we train with a subset of classes (classes seen during training) and we test with a mutually exclusive subset of classes (unseen classes). Details of our evaluation strategy is illustrated in Figure 9.

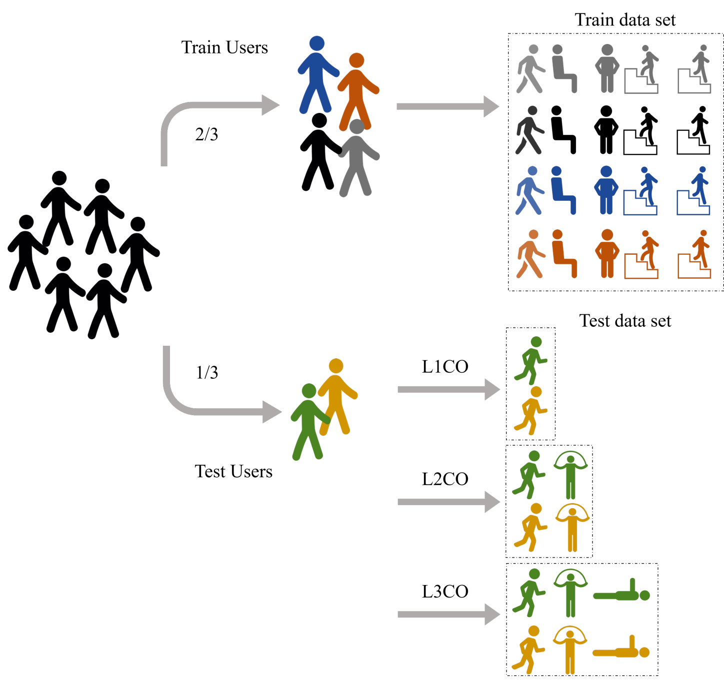

Figure 9: User hold-out validation strategy for Personalised Open-ended Matching Networks

We enforce the influence of personalisation on Open-ended HAR by adapting the same process outlined in Section 5.5 to form train and test instances for each of the open-ended matching network configurations but ensuring exclusivity of classes as follows:

Instances for training set: A training set contains n_{tr} number of classes where $n_{tr} = |\mathcal{L}|$; and for each train user, 500 query instances are selected stratified across all training classes. Each query is paired with a disjoint support set sampled without replacement from the same user to create the complete instance, where 5 instances per class are sampled ($k = 5$). In total we create $500 \times number\_of\_train\_users$ amount of train data.

Instances for test set: For each test user, 5 instances (k=5) are sampled from each training class (seen), as well as each test class (unseen) to create the test support set of size $k \times n_{te}$ where $n_{te} = |\mathcal{L}| + |\hat{\mathcal{L}}|$ and $k = 5$. The sampled test support set simulates the provision of calibration data from test users for unseen activity classes. Remaining test instances are considered as query test instances and each is paired with the test support set to create complete test instances. Note that now the support set can contain both new calibrated data as well as other data for previously existing activities.

5.6.1. Leave-one-class-out (L1CO) Experiments

We create n number of experiments for each dataset where $n = |\mathcal{L}| + |\hat{\mathcal{L}}|$ such that each activity class will serve as the test class. Accordingly we create 22, 22, 8, 9 and 9 experiments for datasets HDPoseDS$_{17}$, HDPoseDS$_6$, PAMAP2, SelfBACK$_{W,T}$ and SelfBACK$_W$ respectively. We will refer to these experiments as L1CO as illustrated in Figure 9. For instance we can see that the single jogging activity appears only in the test set (L1CO) but is not included in the 5 train activities from the 2/3 of the train users. Essentially this setup ensures that both users as well as classes are disjoint between training and testing.

5.6.2. Leave-N-class-out (LNCO) Experiments

In a real-world deployment, an open-ended HAR algorithm should evolve robustly as the user introduces new unseen classes. Here we want to explore how performance might vary as increasing numbers of unseen activity classes are folded-in with, MN^Z. Basically we evaluate MN^Z by leaving out approximately up-to one third of its total number of classes and treating them as unseen classes. We refer to these as LNCO (Leave-N-Class-Out) experiments. See for instance Figure 9 for examples of leaving out 2 (L2CO) and 3 (L3CO) classes.

HDPoseDS$_{17}$: We created 7 LNCO experiments with the HDPoseDS dataset where $N = \{1, 2, 3, 4, 5, 6, 7\}$ (n = 22). For instance, experiment *L6CO* will use data from 16 classes as train data and data from 6 classes as test data. We repeated each experiment 20 times with a random set of test classes each time.

PAMAP2, SelfBACK$_{W,T}$, SelfBACK$_{W}$: We created 3 LNCO experiments with each dataset where $N = \{1, 2, 3\}$ (n = 8,9,9). For instance, experiment *L3CO* for PAMAP2 will use data from 5 classes as train data and data from 3 classes as test data. We repeated each experiment 10 times with random set of test classes each time.

6. Results

In this section we first study the impact of personalisation on HAR and thereafter move onto Personalised Open-ended HAR results.

6.1. Personalised Human Activity Recognition

Table 2: Personalised Matching Networks Results

Datasets	Accuracy (%)		Difference
	MN [7]	MN^P (Ours)	
HDPoseDS$_{17}$	76.781	**98.365**	+21.684
HDPoseDS$_{6}$	42.917	**91.856**	+48.939
PAMAP2	87.148	86.900	-0.248
SelfBACK$_{W,T}$	73.403	**91.689**	+18.286
SelfBACK$_{W}$	63.196	**85.633**	+22.437

Table 2 summarises comparative results for non-personalised (MN [7]) vs. personalised Matching Network (MN^P) experiments from Section 5.5. With the MN^P architecture, we observe a significant performance improvement with four of our datasets with accuracy improvements in the range of 18-48% (statistically significant at 95% confidence level). This confirms that there is a clear advantage to using personalised support sets for HAR using matching networks. Looking at performances on individual datasets, we observe that results on both the HDPoseDS datasets have been improved with as much as a 48% recorded with one of them; followed by the SelfBACK$_W$ and SelfBACK$_{W,T}$ datasets. However the expected improvements were not evident with the PAMAP2 dataset. This might be explained by the different characteristics observed in each dataset, in particular we have fewer users and fewer data instances in PAMAP2. Nevertheless given the non-conclusive result obtained for personalisation with PAMAP2 (unlike with the other 4) we plan to study this dataset more closely in the follow-on personalised Open-ended HAR sections; whereby results for both personalised and non-personalised versions of Open-ended HAR will be explored with PAMAP2; whilst only personalised versions will be taken forward with the rest of the datasets.

6.2. Personalised Open-ended Human Activity Recognition

In this section we first look at L1CO results from HDPoseDS and compare them against the two most commonly used Open-ended HAR algorithms as baselines. In subsequent sections, we further validate our approach with two other datasets and finally we present LNCO results in detail.

6.2.1. L1CO on HDPoseDS

We first compare our method against two most commonly used "knowledge-intensive" ZSL algorithms for open-ended pose classification with HDPoseDS dataset. 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 while achieving comparable performance.

- DAP [21]: Direct Attribute Prediction, mostly commonly used ZSL algorithm based on class-attribute matrix, introduced by [21].

- AI [11]: ZSL algorithm proposed by [11] that utilises a class-attribute matrix and attribute importance.

Table 3: L1CO on $\mathrm{HDPoseDS}_{17}$ and $\mathrm{HDPoseDS}_{6}$

Test Class	DAP [21]	AI [11]	MN^Z (Ours)	
			$\mathrm{HDPoseDS}_{17}$	$\mathrm{HDPoseDS}_{6}$
WaistTwistingR	0.364	0.293	**1.000**	**1.000**
StretchingForward	0.370	0.871	**1.000**	0.982
Sitting	0.407	0.744	**1.000**	0.896
WaistTwistingL	0.424	0.264	**1.000**	0.997
FoldingArm	0.477	0.439	**1.000**	0.990
Skiing	0.528	0.783	**1.000**	**1.000**
BaseballHitting	0.586	0.774	**1.000**	**1.000**
Boxing	0.655	0.749	**1.000**	0.997
StretchingCalfL	0.665	0.807	**1.000**	0.712
Standing	0.715	0.694	**1.000**	0.993
Thinking	0.824	0.823	**1.000**	**1.000**
Squatting	0.892	**1.000**	**1.000**	0.942
DeepBreathing	0.906	0.980	**1.000**	**1.000**
StretchingCalfR	0.957	0.890	0.985	0.911
PointingR	0.963	0.995	**1.000**	**1.000**
StretchingUp	0.991	**1.000**	**1.000**	0.970
HeelToBackR	0.993	0.973	**1.000**	**1.000**
PointingL	0.994	0.972	**1.000**	0.992
RaiseArmR	**0.997**	0.952	0.968	0.945
WaistBending	**1.000**	0.961	**1.000**	**1.000**
HeelToBackL	**1.000**	0.979	0.997	0.986
RaiseArmL	**1.000**	0.985	**1.000**	0.853
Mean	0.760	0.815	0.998	0.962

Table 3 presents L1CO evaluation results with HDPoseDS dataset in detail. It is sorted by increasing performance of DAP [21]. We have used bold text to indicate the best result achieved for each experiment. The baseline DAP [21] achieves performance that ranges from 0.364 to 1.000 with an average F-measure of 0.760; the baseline AI [11] achieves performance that ranges from 0.293 to 1.000 with an average of 0.815. Overall we can see that MN^{Z} consistently outperforms both DAP [21] and AI [11] in both sensor configurations (statistically significant at 95% confidence level against both DAP [21] and AI [11]). With the 17 sensor configuration, MN^{Z} achieves a maximum F-measure of 1.0 with 85% of the experiments; the minimum performance is as high as 0.968, and the average F-measure is 0.998. With the more restricted 6 sensor configuration, MN^{Z} again achieves a maximum F-measure of 1.0, with the minimum performance of 0.712, and an average F-measure of 0.962.

Considering the range of F-measures obtained across all experiments, it is evident that the performance of MN^{Z} is highly reliable over all activity classes compared to both baselines. Consistent results obtained for restricted sensor configuration suggests that our algorithm performs well with minimised sensor requirements. This is an important insight for when developing robust Open-ended HAR algorithms that are user-friendly for real-world deployment. We continue this investigation further with three other datasets that are further restricted in sensor requirements.

6.2.2. L1CO on PAMAP2, SelfBACK$_{W,T}$ and SelfBACK$_{W}$

We have selected three datasets, PAMAP2, SelfBACK$_{W,T}$ and SelfBACK$_{W}$ with 3, 2 and 1 sensors respectively, that are compiled for sedentary activities, ambulatory activities and activities of daily living. With this evaluation we further investigate the robustness of our approach in circumstances that use fewer sensors to determine a wide range of activities.

We will present standalone results for these datasets as there is no appropriate baseline in literature - existing ZSL algorithms such as DAP [21] or AI [11] demands for a domain knowledge acquisition task (in the form of a class-attribute matrix) that is not available for these two data sources. With the PAMAP2 dataset we will evaluate MN^{Z} in both personalised and non-personalised settings; in order to better understand the role of personalisation on this dataset given our non-conclusive results in the previous section.

Table 4: L1CO on PAMAP2

Test class	Non-personalised MN^Z	MN^Z
Descending stairs	0.554	**0.817**
Sitting	0.638	**0.824**
Ascending stairs	0.406	**0.843**
Vacuum cleaning	**0.898**	0.875
Ironing	0.670	**0.915**
Standing	0.879	**0.932**
Lying	0.943	**0.961**
Walking	0.958	**0.969**
Mean	0.743	0.892

Table 4 presents L1CO evaluation results for PAMAP2 dataset; it is sorted by increasing performance of MN^Z. Unlike the results we obtained previously for personalised vs. non-personalised HAR with PAMAP2 dataset, here we see a far more conclusive outcome in favour of personalisation when faced with Open-ended HAR tasks (with statistical significance at 95% confidence level). With non-personalised MN^Z, the performance ranges from 0.406 to 0.958 where the average f-measure is 0.743. With MN^Z the performance ranges from 0.817 to 0.969 where the average F-measure is 0.892. We achieve consistently good performance with MN^Z across all experiments with minimum performance being over 0.810. These results suggest that there is a significant advantage in using personalisation for Open-ended HAR even with the PAMAP2 dataset. In addition the consistency of results over different experiments are comparatively stable when using a personalised approach.

It is worth noting that PAMAP2 results here compared to that of HDPoseDS results in Table 3 are somewhat lower (i.e. PAMAP2 has a mean value of 0.892 instead of 0.998 as with $\mathrm{HDPoseDS}_{17}$ or 0.962 with $\mathrm{HDPoseDS}_6$). This can be explained by noting the difference between the number of sensors used in each of the datasets. For instance PAMAP2 uses just 3 sensors (located on the wrist, chest and the ankle) whilst HDPoseDS use as much as 17 in one dataset and 6 in the other. Accordingly the overall lower mean performance with PAMAP2 is to be expected since with fewer number of sensors it is less likely to be better able to capture all necessary movements to support HAR. However even with half the number of sensors used in the HDPoseDS$_6$ dataset, MN^Z still achieves 0.892 F-measure with PAMAP2.

Table 5: L1CO on SelfBACK$_{W,T}$ and SelfBACK$_W$

Test class	MN^Z	
	SelfBACK$_{W,T}$	SelfBACK$_W$
Walking downstairs	0.731	0.544
Walking fast	0.796	0.707
Walking moderate pace	0.857	0.719
Walking upstairs	0.916	0.703
Walking slow	0.946	0.830
Standing	0.958	0.927
Jogging	0.985	0.986
Sitting	0.990	0.973
Lying	0.994	0.938
Mean	0.908	0.814

In Table 5 we presents MN^Z results for ZSL experiments with the SelfBACK$_{W,T}$ and SelfBACK$_W$ datasets. Firstly looking at the SelfBACK$_{W,T}$ results with MN^Z we note that it is in the range 0.731 to 0.994 with an average F-measure of 0.908. We observe a fairly consistent performance across experiments but the algorithm struggles with classes such as walking downstairs and walking fast. This is reasonable given that there are five variations of walking as activity classes in this dataset, which draws us to the conclusion that similarity based MN^Z algorithm performs better with activities that are naturally significantly different. Similar to personalisation results we observe that even with 2 sensors SelfBACK$_{W,T}$ outperforms PAMAP2 results, which suggests that not only the number of sensors but their placement has a major impact on Open-ended HAR performance.

Unlike SelfBACK$_{W,T}$ with SelfBACK$_{W}$ we have data from just a single wrist sensor, which arguably is the most user friendly sensor configuration for a wearable based Open-ended HAR application. With MN^{Z} the results range from 0.544 to 0.986 where the average F-measure is 0.814. We observe that the consistency of performance across different experiments drop as the number of sensors present are limited. Experiments where the test class is a variation of walking such as walking downstairs or walking fast are again found to be further challenging with a single sensor. Naturally a single sensor on the wrist can capture only a limited form of the full body movement, which is likely to result in a more ambiguous sensor data stream. Accordingly we would expect that the similarity-based attention mechanism used in MN^{Z} to struggle to differentiate between feature representations from different ambulatory activities.

6.2.3. LNCO Results

The aim of this evaluation is to further validate the robustness of our approach and observe how our approach evolves when multiple unseen classes are introduced to the application after deployment (as we saw on stage 5 and 6 on Section 3). We will report standalone results for these experiments as we cannot find an appropriate baseline in literature due to the novelty of our approach and inherent challenges of reproducibility of existing knowledge-intensive Open-ended HAR algorithms.

Table 6 presents results we obtained for Leave-N-class-out experiments with all four datasets. We have reused the mean L1CO results on column “L1CO”. First row refers to results obtained with the HDPoseDS dataset with 17 sensor configuration. We observe that MN^{Z} maintain nearly 1.000 F-measure as we keep introducing up to 7 new classes after deployment. With PAMAP2, SelfBACK$_{W,T}$ and SelfBACK$_{W}$ datasets we again observe that the F-measure

Table 6: LNCO results

Datasets	MN^Z						
	L1CO	L2CO	L3CO	L4CO	L5CO	L6CO	L7CO
$HDPoseDS_{17}$ (n=22)	0.998	1.000	0.999	0.996	0.998	0.998	0.996
PAMAP2 (n=8)	0.892	0.898	0.874	N/A	N/A	N/A	N/A
$SelfBACK_{W,T}$ (n=9)	0.908	0.928	0.937	N/A	N/A	N/A	N/A
$SelfBACK_{W}$ (n=9)	0.813	0.844	0.862	N/A	N/A	N/A	N/A

from L1CO is maintained even as we increase the number of new and unseen classes that are introduced in Open-ended HAR (up to 3 new classes). We also observe that there are miner random increments of performance as we introduce new unseen classes, this is due to the random selection of test classes in the experiment design. Overall, we conclude that our algorithm, MN^Z maintain consistent performance as new classes are introduced to the application.

Considering all experiment results, we recognise the need for strategic placement of multiple sensors to capture full body movement to preserve reliable performance and user-friendliness of the personalised Open-ended HAR application. It is highly significant for Open-ended HAR, since at design time, we are unable to anticipate the activity preferences of the end-user.

7. Discussion

It is evident that the similarity based "knowledge-light" methods for Personalised Open-ended HAR is performing consistently superior to the state-of-the-art knowledge-intensive methods. In this section we draw insights to explain the reasons behind this performance improvement by discussing the limitations of "knowledge-intensive" methods and then explore potential implications of our method.

7.1. Limitations of Knowledge-intensive Methods

As mentioned in Section 2, performance of knowledge-intensive methods, depends on several aspects. For instance with the AI algorithm [11], firstly each sensor determines a lower level action of the user and secondly, these lower level actions from different sensors are combined together to derive the pose on the basis of a set of rules. We believe that the completeness and the accuracy of this rule set are important contributing factors. There are mainly two approaches to design these rules; firstly, designed manually with the knowledge of an expert [11] or secondly learnt as mentioned in Section 2.

Completeness can only be achieved by ensuring every possible pose is covered by one or more rules prior to the deployment of the model. None of the rule acquisition methods have the ability to induce new rules and cannot guarantee that the lower level actions are sufficient to describe all possible future poses; i.e. they do not have a granular intermediary feature space as do the MN methods. The accuracy of the rule set is determined by the the ability to explain a pose using lower level actions predicted by different sensors. For instance, how accurately can we describe the pose "Pointing with Right hand" using sensors on elbow and hand [11]. Although knowledge-intensive methods have been found to performs well on open-ended image recognition tasks, where an image can be described with objects in the image, in contrast, we believe it is challenging with human activities, where we cannot reduce an activity to a set of movements due to the complex nature of movements as well as the personal variations in human movement.

In this paper we have proposed a different approach to open-ended HAR by exploiting similarity knowledge. As discussed in Section 4 the MN model learns a feature space where data from different classes are distinctly separated. This property is considerably preserved when new classes are introduced at the test time (i.e. MN^{Z}), and it still produces a feature space where the instances from different classes are substantially separated.

7.2. Implications of Personalised Open-ended Matching Networks

Our method relies on few instances of data for each activity provided by the user eliminating the need for building a complete knowledge base (or rule set) prior to model deployment. In this way it eliminates the need to represent an activity with multiple intermediary feature representations. We use few examples of an activity obtained during test time to represent the activity that was not seen during training. We argue that obtaining a sample of recorded sensor data is not an overhead in human activity recognition (As explored in Section 3); thus obtaining a few calibration examples during test time is not a limitation of our method. Therefore we believe that the methods introduced in this paper have great potential in the area of Open-ended human activity recognition.

A clear advantage of Open-ended MN is that no additional training is required when new activity classes are introduced to the model. This is advantageous to operate on edge devices that are limited in memory and computational capacity. The evaluation on adding multiple unseen classes (in Section 6.2.3) demonstrated the scalabil ty of this algorithm. However we expect there to be at least two potential scenarios where model re-training policies will be necessary; firstly, when unseen classes are similar to one or more of the existing set of classes; and secondly when changes in user circumstances (e.g. weight, disabilities, gait) are likely to invalidate previously provided data. In the former situation the system will be required to re-learn the changed class boundaries in order to differentiate the new from the previous classes by integrating few instances of data from the new class. Whilst in the latter scenario, the user will be required to re-train with new instances for all activity classes.

8. Conclusions

This article has introduced a neural matching architecture that can support both Personalisation and Open-ended HAR. Results from our comparative studies suggest that the proposed methods are able to address the challenge of wearable devices being restricted to recognising from a fixed set of given activities (e.g. walking, running cycling) pre-modelled based on a general population. The fixed nature of these models can be attributed to the conventional training strategy adopted for supervised Human Activity Recognition (HAR) algorithms i.e.activities and persons that are to be recognised must appear in the initial training data. Existing personalisation algorithms impose a burden on individual users to produce excessive amounts of calibration data; existing Open-ended HAR algorithms depend on expert knowledge acquisition to recognise unseen classes and are not well suited for mobile platforms.

The proposed Personalised Open-ended Matching Network (MN^Z) is "knowledge-light", where we use a few seconds of raw calibration data obtained through micro-interactions with the end-user to personalise and to introduced new activity classes after deployment. We first evaluate the effectiveness of personalisation by comparing our personalised algorithm with the original Matching Networks architecture where the results suggests personalisation contributes to major performance improvements. We further evaluate our algorithm for personalised Open-ended HAR; first against the two most common ZSL algorithms which by nature are knowledge-intensive, and our results confirm that the proposed knowledge-light approach to Open-ended HAR outperforms both and is consistently reliable over a wide range of activity classes, with zero knowledge engineering cost. In addition our evaluation with multiple unseen classes resulted in consistent performance confirming the robustness of MN^Z. We observe that the number of sensors and their placement is a major contributing factor to the performance of Open-ended HAR.

In future work, we plan to explore methods to minimise sensor requirements after deployment through methods such as Translators proposed in [32]. This will enable us to train the model in an unrestricted sensor-rich setting with high accuracy and deploy with fewer sensors with minimum compromise on performance. Thereafter we plan to integrate this solution in to a wearable based mobile application. Finally we encourage the research community to improve "knowledge-light" approaches to personalised Open-ended HAR as it eliminates multiple challenges with existing HAR algorithms and to produce more re-producible research in terms of both open access to algorithms and data.

9. Acknowledgement

This work was part funded by SelfBACK. The SelfBACK project is funded by the European Union's H2020 research and innovation programme under grant agreement No. 689043. More details available at http://www.selfback.eu.

References

References

[1] P. Krebs, D. T. Duncan, Health app use among us mobile phone owners: A national survey, JMIR mHealth and uHealth 3 (4).

[2] M. Berchtold, M. Budde, D. Gordon, H. R. Schmidtke, M. Beigl, Actiserv: Activity recognition service for mobile phones, in: International Symposium on Wearable Computers (ISWC) 2010, IEEE, 2010, pp. 1–8.

[3] L. C. Jatoba, U. Grossmann, C. Kunze, J. Ottenbacher, W. Stork, Context-aware mobile health monitoring: Evaluation of different pattern recognition methods for classification of physical activity, in: 2008 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, IEEE, 2008, pp. 5250–5253.

[4] S. Sani, N. Wiratunga, S. Massie, K. Cooper, knn sampling for personalised human activity recognition, in: International Conference on Case-Based Reasoning, Springer, 2017, pp. 330–344.

[5] E. M. Tapia, S. S. Intille, W. Haskell, K. Larson, J. Wright, A. King, R. Friedman, Real-time recognition of physical activities and their intensities using wireless accelerometers and a heart rate monitor, in: 2007 11th IEEE international symposium on wearable computers, IEEE, 2007, pp. 37–40.

[6] B. Longstaff, S. Reddy, D. Estrin, Improving activity classification for health applications on mobile devices using active and semi-supervised

learning, in: 2010 4th International Conference on Pervasive Computing Technologies for Healthcare, IEEE, 2010, pp. 1–7.

[7] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, in: Advances in Neural Information Processing Systems, 2016, pp. 3630–3638.

[8] N. Kawaguchi, N. Nishio, D. Roggen, S. Inoue, S. Pirttikangas, K. Van Laerhoven, 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, ACM, 2016, pp. 690–695.

[9] H. Larochelle, D. Erhan, Y. Bengio, Zero-data learning of new tasks., in: AAAI, Vol. 1, 2008, p. 3.

[10] H.-T. Cheng, F.-T. Sun, M. Griss, P. Davis, J. Li, D. You, Nuactiv: Recognizing unseen new activities using semantic attribute-based learning, in: Proceeding of the 11th annual international conference on Mobile systems, applications, and services, ACM, 2013, pp. 361–374.

[11] H. Ohashi, M. Al-Naser, S. Ahmed, K. Nakamura, T. Sato, A. Dengel, Attributes' importance for zero-shot pose-classification based on wearable sensors, Sensors 18 (8) (2018) 2485.

[12] X. Xu, T. Hospedales, S. Gong, Transductive zero-shot action recognition by word-vector embedding, International Journal of Computer Vision 123 (3) (2017) 309–333.

[13] J. Qin, L. Liu, L. Shao, F. Shen, B. Ni, J. Chen, Y. Wang, Zero-shot action recognition with error-correcting output codes, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 2833–2842.

[14] X. Sun, H. Kashima, N. Ueda, Large-scale personalized human activity recognition using online multitask learning, IEEE Transactions on Knowledge and Data Engineering 25 (11) (2013) 2551–2563.

[15] T. Miu, P. Missier, T. Plötz, Bootstrapping personalised human activity recognition models using online active learning, in: 2015 IEEE International Conference on Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing, IEEE, 2015, pp. 1138–1147.

[16] S. Sani, N. Wiratunga, S. Massie, K. Cooper, Personalised human activity recognition using matching networks, in: International Conference on Case-Based Reasoning, Springer, 2018, pp. 339–353.

[17] H. Gjoreski, D. Roggen, Unsupervised online activity discovery using temporal behaviour assumption, in: Proceedings of the 2017 ACM International Symposium on Wearable Computers, ACM, 2017, pp. 42–49.

[18] J. Liu, B. Kuipers, S. Savarese, Recognizing human actions by attributes, in: Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition, IEEE Computer Society, 2011, pp. 3337–3344.

[19] H.-T. Cheng, M. Griss, P. Davis, J. Li, D. You, 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, ACM, 2013, pp. 355–358.

[20] C. H. Lampert, H. Nickisch, S. Harmeling, Learning to detect unseen object classes by between-class attribute transfer, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 951–958.

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

[22] Y. Xian, C. H. Lampert, B. Schiele, Z. Akata, Zero-shot learning-a comprehensive evaluation of the good, the bad and the ugly, IEEE Transactions on Pattern Analysis & Machine Intelligence (1) 1–1.

[23] J. Song, C. Shen, Y. Yang, Y. Liu, M. Song, Transductive unbiased embedding for zero-shot learning, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 1024–1033.

[24] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, R. Shah, Signature verification using a" siamese" time delay neural network, in: Advances in neural information processing systems, 1994, pp. 737–744.

[25] S. Chopra, R. Hadsell, Y. LeCun, Learning a similarity metric discriminatively, with application to face verification, in: Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on, Vol. 1, IEEE, 2005, pp. 539–546.

[26] E. Hoffer, N. Ailon, Deep metric learning using triplet network, in: International Workshop on Similarity-Based Pattern Recognition, Springer, 2015, pp. 84–92.

[27] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems, 2017, pp. 4077–4087.

[28] C. Finn, P. Abbeel, S. Levine, Model-agnostic meta-learning for fast adaptation of deep networks, in: Proceedings of the 34th International Conference on Machine Learning-Volume 70, JMLR. org, 2017, pp. 1126–1135.

[29] A. Reiss, D. Stricker, Introducing a new benchmarked dataset for activity monitoring, in: Wearable Computers (ISWC), 2012 16th International Symposium on, IEEE, 2012, pp. 108–109.

[30] S. Sani, S. Massie, N. Wiratunga, K. Cooper, Learning deep and shallow features for human activity recognition, in: International Conference on

Knowledge Science, Engineering and Management, Springer, 2017, pp. 469–482.

[31] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, in: International Conference on Machine Learning, 2015, pp. 448–456.

[32] A. Wijekoon, N. Wiratunga, S. Sani, S. Massie, K. Cooper, Improving knn for human activity recognition with privileged learning using translation models, in: International Conference on Case-Based Reasoning, Springer, 2018, pp. 448–463.

Appendix A - Search for supportset size

Number of instances to be included per class in the support set, k, is a crucial hyper-parameter that affects the performance and efficiency of Matching Networks. For comparative study purposes we wish to use a k value consistently across all datasets. We considered 8 k values 1, 2, 3, 4, 5, 6, 8 and 10; using repeated user hold-out validation, where 2/3 of the users are in training data and the rest in test data. We repeat each hold-out experiment 5 times with a random selection of test users and calculate the mean accuracy as the performance metric.

Table 7: Evaluation for different k values

Datasets	Samples per class (k)							
	1	2	3	4	5	6	8	10
$HDPoseDS_{17}$	65.63	73.89	75.65	71.63	76.78	77.56	76.23	**79.09**
$HDPoseDS_{6}$	35.10	38.09	41.82	43.82	44.96	41.03	**47.54**	42.20
PAMAP2	67.19	81.48	82.91	86.45	86.73	86.83	**87.67**	86.10
$SelfBACK_{W,T}$	65.89	73.35	72.08	73.15	71.41	72.27	71.87	**74.91**
$SelfBACK_{W}$	55.97	58.73	64.03	61.67	64.86	62.64	64.82	**65.34**

All five datasets show considerable performance improvements with increasing k. Specifically, three datasets achieve best performance with $k = 10$ and two datasets with $k = 8$. Increasing k, increases the support set size, which in turn increases the number of pair-wise similarity computations needed by the attention layer of the MN architecture. For instance if $k = 1$ and $n_{tr} = 9$ the support set size is 9 and with equation 3 we need to calculate similarity for 9 pairs. If $k = 10$, and $n_{tr} = 9$, the support set size is 90 and we need to calculate similarity for 90 pairs. Similarity computation is expensive having time complexity that increases linearly with the number of instances in the support sets and the dimensionality of each instance. To validate this observation, we measured the mean time spent on similarity computations by maintaining the output length of the feature embedding function (x'_i) constant at 1200 (Please refer to Figure 7) across all experiments. The mean time taken for similarity calculation of one pair was recorded as $16.4567ms$. Accordingly, when $k = 1$ and $k = 10$, we recorded mean times of $148.1ms$ and $1.481s$ respectively when processing an instance by the MN model. Accordingly the choice of k is a trade-off between model performance and model train/test efficiency. On the basis of the results in 7 and computational overhead, our choice is $k = 5$ across all experiments.

Appendix B - Matching Networks - Robustness to over-fitting

We conducted an experiment to observe the performance of MN when the size of the training data set is gradually reduced. Here we are keen to explore the ability of MN to generalise and its robustness to over-fitting. For this we choose three train/test split ratios where test set ratio is 1/3, 1/2 and 2/3 respectively. We keep "samples per class" at 5 (k=5) and repeated each hold-out experiment 5 times with a random selection of test users and calculate the mean accuracy as the performance metric.

Table 8: Test for over-fitting

Datasets	Test set ratios		
	1/3	1/2	2/3
HDPoseDS_{17}	76.78	69.89	66.61
HDPoseDS_{6}	42.92	38.48	36.05
PAMAP2	87.15	83.62	83.08
$\text{SelfBACK}_{W,T}$	73.40	72.84	73.21
SelfBACK_{W}	63.20	61.20	63.05

Table 8 presents the results. With three datasets the performance decline when the size of the test set is increased. Two datasets maintain the performance across different split ratios. Declined performance is as expected as the model is not exposed to adequate training instances to generalise itself to all possible test scenarios. It is noteworthy that two datasets maintain their performances with limited access to training data. Inherent nature of similarity based learning of MN and the batch normalisation used in the feature embedding function mainly contributes toward regularisation of the model when training with limited data. In summary as expected the best performance was recorded with 2/3 and 1/3 train/test split ratios and the test accuracy is either maintained or declined with increasing test set sizes. Therefore we select the most common user hold-out train/test split ratios of 2/3 and 1/3 for the experiments in this article.