Metadata of the chapter that will be visualized in SpringerLink

Book Title	Activity and Behavior Computing	
Series Title		
Chapter Title	Head-AR: Human Activity Recognition with Head-Mounted IMU Using Weighted Ensemble Learning	
Copyright Year	2021	
Copyright HolderName	The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd.	
Corresponding Author	Family Name	**Gjoreski**
	Particle	
	Given Name	**Hristijan**
	Prefix	
	Suffix	
	Role	
	Division	Faculty of Electrical Engineering and Information Technologies
	Organization	Ss. Cyril and Methodius University
	Address	Skopje, North Macedonia
	Email	hristijang@feit.ukim.edu.mk
Author	Family Name	**Kiprijanovska**
	Particle	
	Given Name	**Ivana**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Jozef Stefan Institute & Jozef Stefan Postgraduate School
	Address	Ljubljana, Slovenia
	Email	ivana.kiprijanovska@ijs.si
Author	Family Name	**Stankoski**
	Particle	
	Given Name	**Simon**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Jozef Stefan Institute & Jozef Stefan Postgraduate School
	Address	Ljubljana, Slovenia
	Email	simon.stankoski@ijs.si
Author	Family Name	**Kalabakov**
	Particle	
	Given Name	**Stefan**
	Prefix	
	Suffix	

	Role	
	Division	
	Organization	Jozef Stefan Institute & Jozef Stefan Postgraduate School
	Address	Ljubljana, Slovenia
	Email	stefan.kalabakov@ijs.si
Author	Family Name	**Broulidakis**
	Particle	
	Given Name	**John**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Emteq Ltd
	Address	Brighton, UK
	Email	john.broulidakis@emteq.net
Author	Family Name	**Nduka**
	Particle	
	Given Name	**Charles**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Emteq Ltd
	Address	Brighton, UK
	Email	charles@emteq.net
Author	Family Name	**Gjoreski**
	Particle	
	Given Name	**Martin**
	Prefix	
	Suffix	
	Role	
	Division	
	Organization	Jozef Stefan Institute & Jozef Stefan Postgraduate School
	Address	Ljubljana, Slovenia
	Email	martin.gjoreski@ijs.si

Abstract	This paper describes the machine learning (ML) method Head-AR, which achieved the highest performance in a competition with 11 other algorithms and won the Emteq Activity Recognition challenge. The goal of the challenge was to recognize eight activities of daily life from a device mounted on the head, which provided data from a 3-axis IMU: accelerometer, gyroscope, and magnetometer. The challenge dataset was collected by four subjects, of which one subject was used as a test for the challenge evaluation. The method processes the stream of sensors data and recognizes one of the eight activities every two seconds. The method is based on weighted ensemble learning, which combines three models: (i) a dynamic time warping classification model, which analyzes raw accelerometer data; (ii) a classification model that uses expert features; (iii) and a classification model that uses features selected by a feature selection algorithm. To compute the final output, the predictions of the three models are combined using a novel weighing scheme. The method achieved an F1-score of 61.25% on the competition's evaluation.

Chapter 10
Head-AR: Human Activity Recognition with Head-Mounted IMU Using Weighted Ensemble Learning

Hristijan Gjoreski, Ivana Kiprijanovska, Simon Stankoski, Stefan Kalabakov, John Broulidakis, Charles Nduka, and Martin Gjoreski

Abstract This paper describes the machine learning (ML) method Head-AR, which achieved the highest performance in a competition with 11 other algorithms and won the Emteq Activity Recognition challenge. The goal of the challenge was to recognize eight activities of daily life from a device mounted on the head, which provided data from a 3-axis IMU: accelerometer, gyroscope, and magnetometer. The challenge dataset was collected by four subjects, of which one subject was used as a test for the challenge evaluation. The method processes the stream of sensors data and recognizes one of the eight activities every two seconds. The method is based on weighted ensemble learning, which combines three models: (i) a dynamic time warping classification model, which analyzes raw accelerometer data; (ii) a classification model that uses expert features; (iii) and a classification model that

H. Gjoreski (✉)
Faculty of Electrical Engineering and Information Technologies, Ss. Cyril and Methodius University, Skopje, North Macedonia
e-mail: hristijang@feit.ukim.edu.mk

I. Kiprijanovska · S. Stankoski · S. Kalabakov · M. Gjoreski
Jozef Stefan Institute & Jozef Stefan Postgraduate School, Ljubljana, Slovenia
e-mail: ivana.kiprijanovska@ijs.si

S. Stankoski
e-mail: simon.stankoski@ijs.si

S. Kalabakov
e-mail: stefan.kalabakov@ijs.si

M. Gjoreski
e-mail: martin.gjoreski@ijs.si

J. Broulidakis · C. Nduka
Emteq Ltd, Brighton, UK
e-mail: john.broulidakis@emteq.net

C. Nduka
e-mail: charles@emteq.net

M. A. R. Ahad et al. (eds.), *Activity and Behavior Computing*, Smart Innovation, Systems and Technologies 204, https://doi.org/10.1007/978-981-15-8944-7_10

496650_1_En_10_Chapter ☑ TYPESET ☐ DISK ☐ LE ☑ CP Disp.:29/9/2020 Pages: xxx Layout: T1-Standard

uses features selected by a feature selection algorithm. To compute the final output, the predictions of the three models are combined using a novel weighing scheme. AQ1 The method achieved an F1-score of 61.25% on the competition's evaluation.

10.1 Introduction

Human activity recognition (HAR) is an integral part of many wearable devices such as smartphones, smartwatches, and fitness trackers. It provides valuable context information that can be utilized in many ways, including tracking physical activities [1], tracking transportation modes [2], and tracking stress levels [3], among others. HAR can also be used as part of disease severity detection methods for Parkinson's disease and depression monitoring.[1]

To advance the field of HAR and to provide a common benchmark for HAR algorithms, several machine learning (ML) challenges have been organized in the HAR community including Challenge-UP 2019,[2] SHL-2018 [4], SHL-2019 [5], EvAAL-2013 [6–9], and Cooking AR Challenge.[3] All of these ML challenges focus on the use of motion capture software and sensors worn below the head. For example, in SHL-2018, the participants developed ML pipelines to classify eight modes of transportation using data from eight smartphone sensors. SHL-2019 was similar to SHL-2018, with one additional complication, i.e., the competitors had to use cross-location transfer learning for their models. Challenge-UP was a HAR and fall-detection challenge in which the participants developed ML pipelines using data from wearable sensors, ambient sensors, and vision devices. The Cooking AR Challenge tasked the competitors with recognizing food preparation activities using motion capture and acceleration sensors.

Differently to those ML challenges, the Emteq HAR challenge[4] tasked the participants with recognizing eight daily life activities using data from inertial sensors (accelerometer, gyroscope, and magnetometer) provided by a head-mounted device, i.e., glasses. The activities of interest were: walking, walking using a smartphone, sitting on a sofa watching a movie, sitting on a sofa using a smartphone, sitting on a chair working on a laptop, sitting on a chair using a smartphone, standing stationary, and standing using a smartphone. The dataset consisted of four subjects, one of whom was used as a test data for the final challenge evaluation.

This paper describes the Head-AR method that was developed for the competition. Head-AR is an IMU ML method that processes streams of sensors data and recognizes one of eight activities every two seconds. Head-AR is an ensemble of three models: (i) a dynamic time warping classification model, which analyzes raw accelerometer data;

[1]Emteq Ltd: https://emteq.net.

[2]https://sites.google.com/up.edu.mx/challenge-up-2019.

[3]https://abc-research.github.oio/cook2020/.

[4]https://github.com/simon2706/Emteq-ARC2019.

(ii) a classification model that uses expert features; (iii) and a classification model that uses features selected from an extensive set of general time-series features, using a feature selection algorithm.

10.2 Relation to Prior Work

HAR using body-worn sensors is a mature field. ML algorithms such as Random Forest (RF), Support Vector Machines (SVM), and K-Nearest Neighbors (KNN) are widely used for building accurate HAR models [10]. For example, Arif et al. [11] constructed a pipeline in which time-domain features are extracted from accelerometer data and are then filtered using a Correlation-based Feature Selection (CFS) method. Before being fed into a KNN model, the data was further simplified by selecting only the most valuable instances. In this way, they were able to achieve an accuracy above 95% when classifying six ambulation activities. Weng et al. [12] used a hierarchical placement of three SVM classifiers to capture activity information in data from an accelerometer with a very low sampling frequency. The first SVM in their architecture is used to determine if the user is stationary or not. The other two are used to distinguish between stationary and dynamic activities, respectively. This architecture achieved an accuracy of above 96% while having a very low power consumption when classifying whether a user is sitting, standing, walking, or running. Zappi et al. [13] implemented a robust system that aimed to be independent of the number of accelerometers that were used or the quality of data. Their solution was based on the use of Hidden Markov Models as base learners for each sensor in the system, whose outputs were later combined using either majority voting or a discrete naive Bayes classifier. When all 57 sensors present in the Skoda Mini Checkpoint dataset are functional, their system achieved an accuracy of up to 96% on ten different activities.

In recent years, deep learning (DL) has emerged as a novel approach in the field of HAR, with methods mainly focusing on the use of Convolutional Neural Networks (CNNs) [14], Recurrent Neural Networks (RNNs) [15] or a combination of the two, with architectures such as the DeepConvLSTM [16]. Although DL has produced some impressive results, in most cases the networks' training has been done using large publicly available datasets such as OPPORTUNITY, PAMAP2, and UCI-Smartphone [17]. However, the Emteq HAR challenge provided a small dataset (only a few hours of data), making the training of end-to-end deep learning models not applicable in this situation. Furthermore, the results of several HAR competitions suggest that, in some situations, classic ML approaches might still be able to produce better results compared to DL [4, 5, 9].

In the field of HAR, sensors are usually placed on the wrists [18–20], ankles [18, 21], hips [2, 11, 12], waist [22] or the torso [23] of the user. Approaches using head-mounted devices are rather scarce. Loh et al. [24] used a head-worn accelerometer, barometer, and GPS sensors with an SVM for fitness activity classification. Ishimaru et al. [25] used head-worn electrooculography (EOG) and accelerometers data, which was segmented and classified by a KNN algorithm. Additionally, Zhang et al. [26] and Farooq et al. [27] proposed the use of head-mounted sensors to detect eating and chewing events. More specifically, Zhang et al. [26] used eyeglasses equipped with electromyography (EMG) sensors in order to monitor muscles' activity. In all of these contributions, the authors suggest using sensors that are either highly specialized to the classification task or are simply more expensive compared to the accelerometer, gyroscope, and magnetometer proposed in our method.

Regarding the activities of interest in HAR, the most common ones for classification are dynamic ones, e.g., walking, running, cycling, and doing housework. This is reflected in HAR datasets such as OPPORTUNITY [28] and PAMAP2 [29]. Classifying activities which differ from each other by very subtle changes in posture or the existence of "micromovements" such as "sitting on a sofa watching a movie" versus "sitting on a sofa using a smartphone" is rarely addressed in related studies, even more so with a head-mounted device. This is of particular interest for Emteq, and therefore it is addressed by our method in this study.

Finally, a state-of-the-art HAR method, which combines a feature-based model and a model based on raw data was recently presented by Gjoreski and Janko et al. [2, 30]. The raw data model was an end-to-end DL model. Compared to that approach, ours does not use an end-to-end DL, but a combination of Dynamic Time Warping (DTW) and KNN, it does not require large amounts of data for training and could be applied to smaller datasets.

10.3 Data

The competition dataset is recorded in a simulated home environment. It is comprised of approximately three hours of labeled data collected from three volunteers, released for training the models, and one hour of unlabeled data from a fourth volunteer used for the final evaluation of the competitors. The activities are performed when the user is either upright (standing stationary vs. walking) or sitting (sitting at a desk on a chair vs. sitting on a sofa). During the recording, the volunteers may or may not be using a smartphone, resulting in 8 subcategories of activities. The eight activities of interest and their distribution are shown in Fig. 10.1. The dataset size is quite limited, which makes the identification of all eight subcategories even more challenging.

The data is collected with an IMU device worn on the head, providing: a 3-axis accelerometer, a 3-axis gyroscope, and a 3-axis magnetometer, sampled at 50 Hz. Also, we calculated the magnitude of each sensor, resulting in 12 sensor streams, overall.

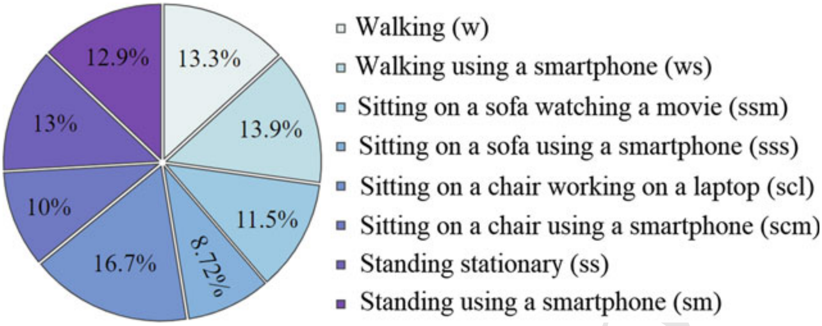

Fig. 10.1 Distribution of the activity data

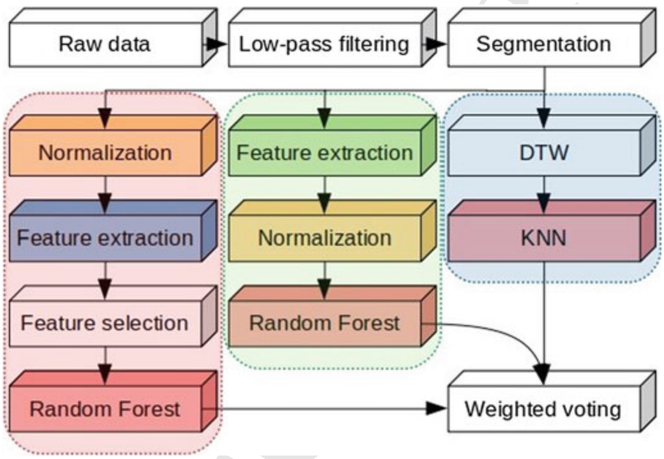

Fig. 10.2 The Head-AR ensemble method

10.4 Method

The proposed Head-AR method (shown in Fig. 10.2) is an ML ensemble of three models: two models are feature-based ML models working with different subsets of features, and one model is a DTW-based model that works with raw sensors' data.

In the first step, the raw data is filtered with a low-pass filter, which acts as a smoothing function in the time domain. This step reduces the influence of high-frequency artifacts, which in this dataset do not carry valuable information since the activities are less dynamic. After the filtering step, the data is segmented using a sliding window of 4 s and a 50% overlap. This way, the model recognizes an activity every 2 s. The windowing parameters were determined empirically. Next, the pipeline separates into three different branches.

In the first branch (left and red in Fig. 10.2), the filtered sensor data is normalized, and a large number of features (12,000 overall) are extracted (see Sect. 10.4.1.). To reduce the number of features, we used a combination of ranking and wrapper feature selection approaches (see Sect. 10.4.2.). Lastly, an RF model for HAR is trained using the selected features.

The second branch (middle and green in Fig. 10.2) is similar to the first one, except that the order of the data normalization and feature extraction is reversed, i.e., we first extract the features, and then we perform normalization. Additionally, in this branch we use expert features which are based on previous HAR work [2, 31] (see Sect. 10.4.1.). The normalized features are then used to train another RF model.

The third branch (right and blue in Fig. 10.2) uses a KNN classification model based on DTW distance rather than the standard Euclidean distance [32]. The dataset contains a transition label that splits the data into trials that consist of data from the same activity (class). Each trial is further segmented using a sliding window. To improve the computational feasibility of determining the DTW distance between the segments, the model considers only the middle segments from each trial. The final predictions are made by taking the majority class of the segments in one trial.

Each model (branch) produces a prediction for each segment. The final prediction for each segment is calculated using weighted voting. For example, the final output O for the ith segment (instance) $\overrightarrow{x_i}$ is determined as follows:

$$O(\overrightarrow{x_i}\,|k,m,n) = \begin{cases} k, & P_{FS_k} > P_{E_m} \wedge P_{FS_k} > P_{D_n} \\ m, & P_{E_m} > P_{FS_k} \wedge P_{E_m} > P_{D_n} \\ n, & P_{D_n} > P_{FS_k} \wedge P_{D_n} > P_{E_m} \end{cases} \tag{10.1}$$

where

$$\begin{aligned} k &= O_{FS}(\overrightarrow{x_l}), k = 1, 2, \ldots, 8 \\ m &= O_E(\overrightarrow{x_l}), m = 1, 2, \ldots, 8 \\ n &= O_D(\overrightarrow{x_l}), n = 1, 2, \ldots, 8 \end{aligned} \tag{10.2}$$

and, P_{FS_k} is the precision of the model in the first branch for the class label k; P_{E_m} is the precision of the model in the second branch for the class label m; and, P_{D_n} is the precision of the model in the third branch for the class label n. In other words, the weighing scheme outputs the prediction of the model that has the highest precision score for its predicted class. The precision for each class is calculated using cross-validation on the model's training data. After having the precision for each class from each model, we can obtain the final weighing scheme as described with Eqs. 10.1, 10.2.

Our weighing scheme is general and can be applied for two or more models. The main idea of the proposed scheme is to utilize multiple classifiers that are able to learn the characteristics of different classes in such a way that we maintain the individual accuracy for those classes when merging the predictions from multiple classifiers.

10.4.1 Feature Extraction

The Python package tsfresh[5] allows general-purpose time-series feature extraction, which we exploited in generating approximately 1000 features per sensor stream. These features include the minimum, maximum, mean, variance, the correlation between axes, their covariance, skewness, kurtosis, quartile values and range between the number of times the signal is above/below its mean, the signal's mean change, and its different autocorrelations (correlations for different delays), among others. Since they are general features, we applied a feature selection algorithm to select the features that are useful for HAR. These features are used for one of the ML models.

The second feature-based model uses expert features, i.e., features based on previous HAR work [2, 31]. These features were calculated using the signal's Power Spectral Density (PSD), which is based on the fast Fourier transform. The features were calculated for each sensor stream. They include PSD magnitude, energy, entropy, binned distribution using ten bins up to 25 Hz, and first four statistical moments of the PSD (mean value, standard deviation, skewness, and kurtosis). The overall number of expert features is 264, which is low enough to be used with most of the ML algorithms without a feature selection.

10.4.2 Feature Selection

We built a feature selection algorithm to select the features that are useful for the specific task. We focused on removing correlated features and features which did not contribute to the model's performance. First, we estimated the mutual information (MI) between each feature and the class. The higher the MI, the stronger the relationship between the class and the corresponding feature. Next, we divided the features into a 100 nonoverlapping subgroups. To begin the feature selection process, we calculated the Pearson correlation between the features in the first subgroup. If the correlation between a pair exceeded a threshold of 0.8 (strong correlation), we removed the feature with the lower MI. Using the remaining features from this subgroup and the features of the next subgroup, we created a new set of features on which the previously described procedure was applied again. This process was repeated until there were no more subgroups to add to the current set.

In the last phase, we used a wrapper algorithm to further reduce the subset of features: (i) we selected the highest ranked feature (by mutual information), we trained an ML model, and we calculated its F1-score; (ii) the next best-ranked feature was added to the subset and the model was re-trained and re-evaluated. If the F1-score increased for more than 1%, the newly added feature was kept in the final feature subset, otherwise, it was rejected. The second step was repeated iteratively for each feature.

[5] https://tsfresh.readthedocs.io/en/latest/text/list_of_features.html.

To avoid overfitting, the feature selection was performed using LOSO evaluation, which resulted in three feature-selection iterations. In each iteration, the data of two subjects was used as a training subset (i.e., to calculate MI, correlation, and to train the ML model). The data of the third subject was used as a test (i.e., to evaluate the ML model during the "wrapper" phase). The final subset of features was calculated as the intersection of the features selected in each LOSO iteration. It contained 226 features.

10.4.3 Feature-Based ML Algorithms

We experimented with a variety of ML algorithms including: Decision Tree [33], RF [34], Naive Bayes [35], KNN [36], SVM [37], Bagging [38], Adaptive Boosting [39], and Extreme Gradient Boosting (XGB) [40]. The models' hyperparameters were tuned using the following procedure: parameter settings were randomly sampled from distributions predefined by an expert. Next, models were trained with the specific parameters and then evaluated using internal k-fold cross-validation on the training data. The best-performing model from the internal k-fold cross-validation was used to classify the test data.

In general, the ensemble models performed better than the single-model algorithms. Additionally, the feature selection was ran both with the RF and the XGB and achieved similar results. We decided to continue with RF because it has fewer hyperparameters and it is faster to train.

10.5 Evaluation Results

We evaluated the performance of the models using LOSO evaluation. All results presented in this section refer to the internal evaluation of the methods.

In Table 10.1, we present the macro F1-score [41], an evaluation metric predefined by the challenge organizers. The first four columns present the results achieved by the DTW model, the RF trained with expert features (RF-E), the RF trained with all general features (RF-A), and the RF trained with features selected by the feature selection algorithm (RF-FS). The next three columns present the results achieved by voting ensembles of two models (single models combined using weighted voting). We disregarded the RF-A model from further experiments, as it showed the lowest results in terms of macro F1-score and its training is time-consuming. The column before the last one presents the results achieved by our method (Head-AR), which is a weighted voting ensemble of the three models: DTW, RF-E, and RF-FS. The last column presents the results achieved by a majority voting ensemble of the same three models.

The internal testing results show that each of the single models is specialized for a subset of classes. For example, the DTW outperformed the other single models for the

Table 10.1 F1-score for: single models (DTW, RF-E, RF-A, RF-FS); two-model-weighted voting ensembles; Head-AR—three-model-weighted voting ensemble; and three-model majority voting ensemble. LOSO evaluation. w-walking, ws-walking using a smartphone, ssm-sitting on a sofa watching a movie, sss-sitting on a sofa using a smartphone, scl-sitting on a chair working on a laptop, scm-sitting on a chair using a smartphone, ss-standing stationary, sm-standing using a smartphone

	DTW	RF-E	RF-A	RF-FS	DTW RF-E	DTW RF-FS	RF-E RF-FS	Head AR	DTW RF-E RF-FS majority
w	0.94	0.88	**0.99**	**0.99**	0.94	0.93	0.93	**0.99**	0.96
ws	0.39	0.83	**0.99**	**0.99**	0.76	**0.99**	**0.99**	**0.99**	0.95
ssm	0.74	**0.92**	0.46	0.52	**0.92**	0.74	0.67	**0.92**	0.90
sss	0.21	0.11	0.01	0.07	**0.27**	0.19	0.31	0.22	0.01
scl	0.51	**0.66**	0.30	0.62	**0.66**	0.57	0.36	**0.66**	0.62
scm	**0.21**	0.14	0.08	0.17	0.00	0.00	0.08	0.06	0.15
ss	0.75	0.83	0.53	0.78	0.83	0.78	0.81	0.83	**0.90**
sm	0.23	0.67	0.18	0.29	0.67	0.29	0.54	**0.67**	0.51
F1	0.50	0.63	0.44	0.56	0.63	0.56	0.59	**0.67**	0.63

classes "sitting-sofa-smartphone" and "sitting-chair-smartphone". Also, the model trained with features selected by the feature selection algorithm (RF-FS) significantly outperformed the model trained with all extracted features (RF-A). The model trained with expert features (RF-E) was the best-performing single model. From the two-model combinations, the combination of DTW and RF-E achieved the highest performance. From the three-model combinations, the Head-AR (weighted ensemble) outperformed the voting ensemble. Most significantly, the Head-AR achieved the highest F1-score for five out of eight classes, and it is second best for two classes, which makes it the best-performing method, overall.

Furthermore, Fig. 10.3 compares the methods by showing the F1-score achieved for each activity and each user, separately. The results of one method on a certain activity are shown as three same-colored dots, each representing one test user in the LOSO evaluation. For example, the three pink dots in each of the columns represent the three F1-scores obtained by the Head-AR method for each activity, when testing on three different users in LOSO evaluation. If we analyze the results of the four best- performing models, the Head-AR, the RF-E model, the DTW + RF-E model and the majority voting ensemble (represented with the colors, pink, orange, red, and gray, respectively) we can see that for the first two activities, the Head-AR model has the most consistent high results across all users. This is not the case for the other three models, whose results are in the range of 0.8–1.0. The Head-AR, RF-E, and DTW + RF-E models show similar results when being compared on the third, fifth, seventh, and eighth activity, with the majority voting ensemble showing larger variance between the results of different users and lower minimum scores

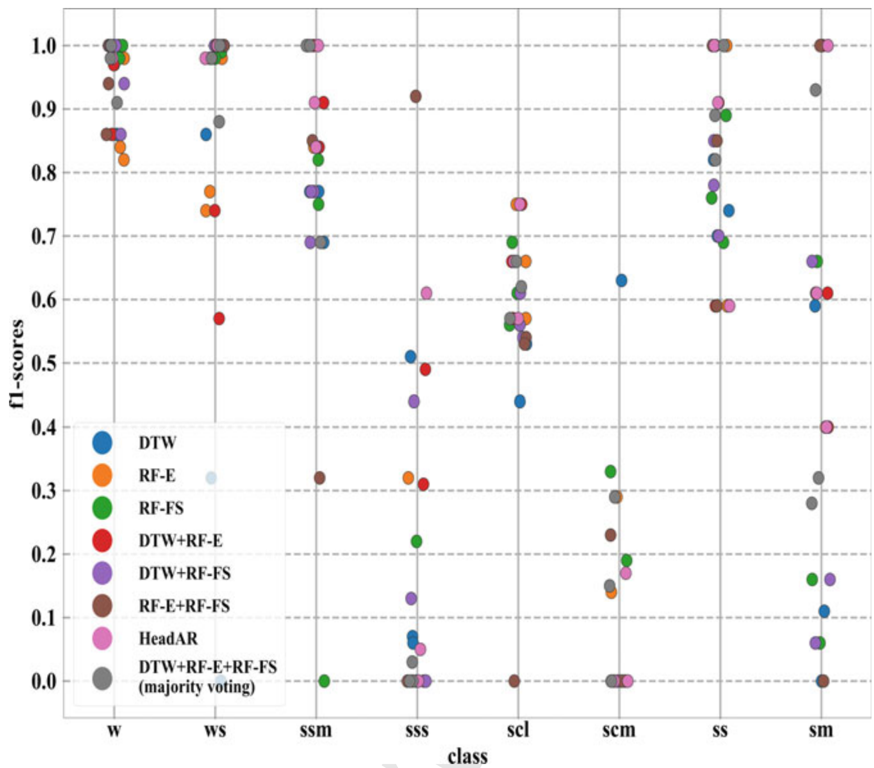

Fig. 10.3 A comparison of the results produced by each of the methods for all activity labels and for every user. w-walking, ws-walking using a smartphone, ssm-sitting on a sofa watching a movie, sss-sitting on a sofa using a smartphone, scl-sitting on a chair working on a laptop, scm-sitting on a chair working on a smartphone, ss-standing stationary, sm-standing using a smartphone

when comparing the "sitting-sofa-movie" and "standing-smartphone" activities. The majority voting model shows higher results compared to the other three models only when looking at the "standing stationary" activity. Finally, when comparing the results on the "sitting-sofa-smartphone" and "sitting-chair-smartphone" activities, the DTW + RF-E model and the majority voting ensemble are the best out of those four, by achieving more consistent results for 2 out of the 3 test users.

Table 10.2 shows the confusion matrix for the Head-AR method. The four classes that involve sitting on sofa or chair, with or without smartphone (ssm, sss, scl, and scm) are often confused. The most problematic classes are "sitting-sofa-smartphone" and "sitting-chair-smartphone".

Table 10.2 Summed and normalized (per row) confusion matrix from the LOSO evaluation for Head-AR. w-walking, ws-walking using a smartphone, ssm-sitting on a sofa watching a movie, sss-sitting on a sofa using a smartphone, scl-sitting on a chair working on a laptop, scm-sitting on a chair working on a smartphone, ss-standing stationary, sm-standing using a smartphone

	predicted							
	w	**ws**	**ssm**	**sss**	**scl**	**scm**	**ss**	**sm**
w	99	1	0	0	0	0	0	0
ws	0	100	0	0	0	0	0	0
ssm	0	0	100	0	0	0	0	0
sss	0	0	0	23	41	33	0	3
scl	0	0	0	0	73	24	0	3
scm	0	0	0	9	42	6	0	43
ss	0	0	18	0	6	0	75	0
sm	0	0	0	1	0	24	0	75

10.6 Discussion

The weighted ensemble learning method, Head-AR, was compared with single-algorithm ensemble methods (e.g., RF) and voting ensemble method, i.e., a method that uses the same models as Head-AR, but computes the final output using majority voting. The results presented in Table 10.1 showed that Head-AR combines multiple models more effectively compared to the other methods and achieves the highest evaluation scores.

Regarding the used algorithms in the weighting scheme, it should be noted that they were chosen based on experimental analysis. Experiments were performed with a variety of algorithms (see Sect. 10.4.3), and this particular combination achieved the highest score. However, Head-AR is algorithm independent, and depending on the domain, different algorithms can be used. Compared to other voting schemes, Head-AR's main advantage is that it can combine models specialized for different classes. By using a specialized weighting scheme, Head-AR decides which model's prediction to output as a final prediction.

Moreover, the obtained results showed that Head-AR could distinguish well the activities when the person is in a standing position (e.g., "standing-stationary" and "standing-smartphone") or when he/she is walking (e.g., "walking" or "walking-smartphone"). However, this was not the case with the sitting-related activities, especially "sitting-sofa-smartphone", "sitting-chair-laptop" and "sitting-chair-smartphone". In particular, "sitting-sofa-smartphone" is confused with the chair-related activities rather than "sitting-sofa-movie", which at first seems like a more similar activity. Nevertheless, this can be explained if the posture of the head during these activities is observed in more detail. When a person uses a smartphone, it is usually held at chest or abdomen height. This results in a slight tilt of the head forward, which does not depend on whether the person is sitting on a chair or sofa. A tilt of the head can also be observed when a person is performing the "sitting-chair-laptop" activity, since the laptop is also at a person's chest height when placed on a table or desk. On the other hand, while a person is performing the "sitting-sofa-movie" activity, no head tilt can be observed—the TV is usually at eye level. This is the only activity where a person is in a sitting position and does not use any device (that would result in a head tilt), so the Head-AR method can distinguish it from the other sitting-related activities. However, it remains a challenge for the model to be able to distinguish the other sitting-related activities when a person is using a device (e.g., smartphone, laptop etc.).

One possible solution for this problem would be to introduce temporal information of the instances. In the experiments presented in the paper, all windows were classified independently from one another. This approach discards all the information on temporal dependencies between them. Nevertheless, if a user, for example, is currently performing "sitting-chair-laptop", but the next window is classified as "sitting-sofa-smartphone", followed by another "sitting-chair-laptop" classification, it is likely for "sitting-sofa-smartphone" to be a misclassification. Such relations can be captured using an additional model after the classification. Example models are Hidden Markov models (HMMs), RNNs, Long Short-Term Memory (LSTM) networks [42], bidirectional LSTMs [43], Gated Recurrent Unit (GRU) networks [44], among others. These models can use past and current predictions as input and output the "corrected" current prediction. However, the temporal information about the instances in the dataset was not available, so this approach was not applicable for this challenge.

10.7 Conclusion and Future Work

We presented the Head-AR method for HAR based on weighted ensemble learning that combines three ML models, each of them specialized for a subset of classes. Two of the models are feature-based, and one works with the raw sensors' data streams. Head-AR processes the sensors' data and recognizes one of the eight activities every two seconds. It was tuned for robustness and real-time performance by combining head-mounted IMU sensors.

The internal evaluation showed that this optimal pipeline configuration achieved an F1-macro score of 60–70% (average 67%) on the three training subjects using LOSO evaluation. In general, Head-AR shows higher minimum scores and lower variance between the results for almost every activity of the three subjects, when compared with the other four best-performing methods.

On the competition's evaluation, Head-AR achieved 61.5% F1-macro score on one unseen test subject. However, the results show that there is still room for improvement, especially for sitting-like activities. The problem with these activities is that they are too similar to each other when looking through the prism of a head-mounted device. Even more, the dataset is too small, thus learning accurate models that will work for unseen users is challenging. One possibility to tackle this problem is to incorporate temporal information of the instances into the HAR method, i.e., to use an additional model after the classification that can capture temporal relations between the classes. Another idea is to train personalized models. They are more likely to effectively learn the user-specific differences that confound general models and significantly improve the results [2]. Another possibility to tackle this problem is to include more data from a variety of subjects. Additionally, one can focus on micro-movements and analyze the accelerometer data using template-matching techniques [45]. The idea is that when analyzing the whole sitting segment, one might find some templates/patterns that are characteristic for each of the activities. Finally, we plan to further analyze the magnetometer data to detect the room's specificities, such as locations of the sofa and chairs, to name a few. Even though this might improve the results for this particular dataset, it has disadvantages because the models may learn a room-specific model and not a general one that will work in any environment.

Acknowledgements We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU used for this research. The authors declare that they have no conflict of interest.

References

1. Kozina, S., Gjoreski, H., Gams, M., Lustrek, M.: Three-layer activity recognition combining domain knowledge and meta-classification author list. J. Med. Biol. Eng. **33**, 406–414 (2013)
2. Janko, V., Gjoreski, M., Slapničar, G., Mlakar, M., Reščič, N., Bizjak, J., Drobnič, V., Marinko, M., Mlakar, N., Gams, M., et al.: Winning the sussex-huawei locomotion-transportation recognition challenge. Human Activity Sensing, pp. 233–250. Springer, Berlin (2019)
3. Gjoreski, M., Luštrek, M., Gams, M., Gjoreski, H.: Monitoring stress with a wrist device using context. J. Biomed. Inf. **73**, 159–170 (2017)
4. Wang, L., Gjoreskia, H., Murao, K., Okita, T., Roggen, D.: Summary of the sussex-huawei locomotion-transportation recognition challenge. In: Proceedings of the 2018 ACM International Joint Conference and 2018 International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers, pp. 1521–1530 (2018)
5. Wang, L., Gjoreski, H., Ciliberto, M., Lago, P., Murao, K., Okita, T., Roggen, D.: Summary of the sussex-huawei locomotion-transportation recognition challenge 2019. In: Adjunct Proceedings of the 2019 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2019 ACM International Symposium on Wearable Computers, pp. 849–856 (2019)
6. Gjoreski, H., Kaluža, B., Gams, M., Milić, R., Luštrek, M.: Context-based ensemble method for human energy expenditure estimation. Appl. Soft Comput. **37**, 960–970 (2015)
7. Gjoreski, H., Gams, M., Lutrek, M.: Human activity recognition: From controlled lab experiments to competitive live evaluation. In: 2015 IEEE International Conference on Data Mining Workshop (ICDMW), pp. 139–145. IEEE (2015)
8. Kozina, S., Gjoreski, H., Gams, M., Luštrek, M.: Efficient activity recognition and fall detection using accelerometers. In: International Competition on Evaluating AAL Systems Through Competitive Benchmarking, pp. 13–23. Springer (2013)

9. Gjoreski, H., Stankoski, S., Kiprijanovska, I., Nikolovska, A., Mladenovska, N., Trajanoska, M., Velichkovska, B., Gjoreski, M., Lustrek, M., Gams, M.: Wearable Sensors Data-Fusion and Machine-Learning Method for Fall Detection and Activity Recognition, pp. 81–96 (2020)
10. Lara, O.D., Labrador, M.A.: A survey on human activity recognition using wearable sensors. IEEE Commun. Surv. Tutor. **15**(3), 1192–1209 (2012)
11. Arif, M., Bilal, M., Kattan, A., Iqbal Ahamed, S.: Better physical activity classification using smartphone acceleration sensor. J. Med. Syst. **38**(9), 95 (2014)
12. Weng, S., Xiang, L., Tang, W., Yang, H., Zheng, L., Lu, H., Zheng, H.: A low power and high accuracy mems sensor based activity recognition algorithm. In: 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM), pp. 33–38. IEEE (2014)
13. Zappi, P., Stiefmeier, T., Farella, E., Roggen, D., Benini, L., Troster, G.: Activity recognition from on-body sensors by classifier fusion: sensor scalability and robustness. In: 2007 3rd International Conference on Intelligent Sensors, Sensor Networks and Information, pp. 281–286. IEEE (2007)
14. Zeng, M., Nguyen, L.T., Yu, B., Mengshoel, O.J., Zhu, J., Wu, P., Zhang, J.: Convolutional neural networks for human activity recognition using mobile sensors. In: 6th International Conference on Mobile Computing, Applications and Services, pp. 197–205. IEEE (2014)
15. Inoue, M., Inoue, S., Nishida, T.: Deep recurrent neural network for mobile human activity recognition with high throughput. Artif. Life Robot. **23**(2), 173–185 (2018)
16. Francisco Javier Ordóñez and Daniel Roggen: Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition. Sensors **16**(1), 115 (2016)
17. Wang, J., Chen, Y., Hao, S., Peng, X., Lisha, H.: Deep learning for sensor-based activity recognition: a survey. Pattern Recognit. Lett. **119**, 3–11 (2019)
18. Hammerla, N.Y., Halloran, S., Plötz, T.: Deep, convolutional, and recurrent models for human activity recognition using wearables (2016). arXiv:1604.08880
19. Chernbumroong, S., Atkins, A.S., Yu, H.: Activity classification using a single wrist-worn accelerometer. In: 2011 5th International Conference on Software, Knowledge Information, Industrial Management and Applications (SKIMA) Proceedings, pp. 1–6. IEEE (2011)
20. Plötz, T., Hammerla, N.Y., Olivier, P.L.: Feature learning for activity recognition in ubiquitous computing. In: Twenty-Second International Joint Conference on Artificial Intelligence (2011)
21. McCarthy, M.W., James, D.A., Lee, J.B., Rowlands, D.D.: Decision-tree-based human activity classification algorithm using single-channel foot-mounted gyroscope. Electron. Lett. **51**(9), 675–676 (2015)
22. Ravi, N., Dandekar, N., Mysore, P., Littman, M.L.: Activity recognition from accelerometer data. In: Aaai, vol. 5, pp. 1541–1546 (2005)
23. Li, A., Ji, L., Wang, S., Wu, J.: Physical activity classification using a single triaxial accelerometer based on hmm (2010)
24. Loh, D., Lee, T.J., Zihajehzadeh, S., Hoskinson, R., Park, E.J.: Fitness activity classification by using multiclass support vector machines on head-worn sensors. In: 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), pp. 502–505. IEEE (2015)
25. Ishimaru, S., Kunze, K., Uema, Y., Kise, K., Inami, M., Tanaka, K.: Smarter eyewear: using commercial eog glasses for activity recognition. In: Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing: Adjunct Publication, pp. 239–242 (2014)
26. Zhang, R., Amft, O.: Monitoring chewing and eating in free-living using smart eyeglasses. IEEE J. Biomed. Health Inf. **22**(1), 23–32 (2017)
27. Farooq, M., Sazonov, E.: Accelerometer-based detection of food intake in free-living individuals. IEEE Sens. J. **18**(9), 3752–3758 (2018)
28. Roggen, D., Calatroni, A., Rossi, M., Holleczek, T., Förster, K., Tröster, G., Lukowicz, P., Bannach, D., Pirkl, G., Ferscha, A., et al.: Collecting complex activity datasets in highly rich networked sensor environments. In: 2010 Seventh International Conference on Networked Sensing Systems (INSS), pp. 233–240. IEEE (2010)

29. Reiss, A., Stricker, D.: Introducing a new benchmarked dataset for activity monitoring. In: 2012 16th International Symposium on Wearable Computers, pp. 108–109. IEEE (2012)
30. Gjoreski, M., Janko, V., Slapničar, G., Mlakar, M., Reščič, N., Bizjak, J., Drobnič, V., Marinko, M., Mlakar, N., Luštrek, M., et al.: Classical and deep learning methods for recognizing human activities and modes of transportation with smartphone sensors. Inf. Fusion (2020)
31. Xing, S., Tong, H., Ji, P.: Activity recognition with smartphone sensors. Tsinghua Sci. Technol. **19**(3), 235–249 (2014)
32. Mitsa, T.: Temporal Data Mining. Chapman and Hall/CRC (2010)
33. Ross Quinlan, J.: Improved use of continuous attributes in c4. 5. J. Artif. Intell. Res. **4**, 77–90 (1996)
34. Ho, T.K.: Random decision forests. In: Proceedings of 3rd International Conference on Document Analysis and Recognition, vol. 1, pp. 278–282. IEEE (1995)
35. Russell, S.J., Norvig, P.: Artificial Intelligence: A Modern Approach. Pearson Education Limited, Malaysia (2016)
36. Aha, D.W., Kibler, D., Albert, M.K.: Instance-based learning algorithms. Mach. Learn. **6**(1), 37–66 (1991)
37. Cristianini, N., Shawe-Taylor, J., et al.: An Introduction to Support Vector Machines and other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
38. Breiman, L.: Bagging predictors. Mach. Learn. **24**(2), 123–140 (1996)
39. Freund, Y., Schapire, R.E.: A desicion-theoretic generalization of on-line learning and an application to boosting. In: European Conference on Computational Learning Theory, pp. 23–37. Springer (1995)
40. Chen, T., Guestrin, C.: Xgboost: a scalable tree boosting system acm sigkdd international conference on knowledge discovery and data mining. ACM, pp. 785–794 (2016)
41. Van Asch, V.: Macro-and micro-averaged evaluation measures [[basic draft]]. Belgium: CLiPS, vol. 49 (2013)
42. Hochreiter, S., Schmidhuber, J.: Long short-term memory. Neural Comput. **9**(8), 1735–1780 (1997)
43. Schuster, M., Paliwal, K.K.: Bidirectional recurrent neural networks. IEEE Trans. Signal Process. **45**(11), 2673–2681 (1997)
44. Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y.: Learning phrase representations using rnn encoder-decoder for statistical machine translation (2014). arXiv:1406.1078
45. Nguyen-Dinh, L.-V., Roggen, D., Calatroni, A., Tröster, G.: Improving online gesture recognition with template matching methods in accelerometer data. In: 2012 12th International Conference on Intelligent Systems Design and Applications (ISDA), pp. 831–836. IEEE (2012)

Author Queries

Chapter 10

Query Refs.	Details Required	Author's response
AQ1	Please check and confirm if the author names and initials are correct.	

MARKED PROOF

Please correct and return this set

Please use the proof correction marks shown below for all alterations and corrections. If you wish to return your proof by fax you should ensure that all amendments are written clearly in dark ink and are made well within the page margins.

Instruction to printer	*Textual mark*	*Marginal mark*
Leave unchanged	under matter to remain	
Insert in text the matter indicated in the margin		New matter followed by or
Delete	through single character, rule or underline or through all characters to be deleted	or
Substitute character or substitute part of one or more word(s)	through letter or through characters	new character or new characters
Change to italics	under matter to be changed	
Change to capitals	under matter to be changed	
Change to small capitals	under matter to be changed	
Change to bold type	under matter to be changed	
Change to bold italic	under matter to be changed	
Change to lower case	Encircle matter to be changed	
Change italic to upright type	(As above)	
Change bold to non-bold type	(As above)	
Insert 'superior' character	through character or where required	or under character e.g. or
Insert 'inferior' character	(As above)	over character e.g.
Insert full stop	(As above)	
Insert comma	(As above)	,
Insert single quotation marks	(As above)	or and/or or
Insert double quotation marks	(As above)	or and/or or
Insert hyphen	(As above)	
Start new paragraph		
No new paragraph		
Transpose		
Close up	linking characters	
Insert or substitute space between characters or words	through character or where required	
Reduce space between characters or words	between characters or words affected	