A Deep Learning Approach to on-Node Sensor Data Analytics for Mobile or Wearable Devices

Daniele Ravì, Charence Wong, Benny Lo, and Guang-Zhong Yang, *Fellow, IEEE*

***Abstract*—The increasing popularity of wearable devices in recent years means that a diverse range of physiological and functional data can now be captured continuously for applications in sports, wellbeing, and healthcare. This wealth of information requires efficient methods of classification and analysis where deep learning is a promising technique for large-scale data analytics. While deep learning has been successful in implementations that utilize high-performance computing platforms, its use on low-power wearable devices is limited by resource constraints. In this paper, we propose a deep learning methodology, which combines features learned from inertial sensor data together with complementary information from a set of shallow features to enable accurate and real-time activity classification. The design of this combined method aims to overcome some of the limitations present in a typical deep learning framework where on-node computation is required. To optimize the proposed method for real-time on-node computation, spectral domain preprocessing is used before the data are passed onto the deep learning framework. The classification accuracy of our proposed deep learning approach is evaluated against state-of-the-art methods using both laboratory and real world activity datasets. Our results show the validity of the approach on different human activity datasets, outperforming other methods, including the two methods used within our combined pipeline. We also demonstrate that the computation times for the proposed method are consistent with the constraints of real-time on-node processing on smartphones and a wearable sensor platform.**

***Index Terms*—ActiveMiles, deep learning, Human Activity Recognition (HAR), Internet-of-Things (IoT), low-power devices, wearable.**

I. Introduction

DEEP learning is a paradigm of machine learning that uses multiple processing layers to infer and extract information from big data. Research has shown that the use of deep learning can achieve improved performance in a range of applications when compared to traditional approaches [1]–[6]. Conventional learning approaches use a set of predesigned features—also known as "shallow" features—to represent the data for a specific classification task. In image processing and machine vision, shallow features such as SIFT or FAST are often used for landmark detection [7], whereas for time-series analysis, statistical parameters are used [8]–[11].

Human Activity Recognition (HAR), e.g., generally exploits time-series data from inertial sensors to identify the actions being performed. In healthcare, inertial sensor data can be used for monitoring the onset of diseases as well as the efficacy of treatment options [11], [12]. For patients with neurodegenerative diseases, such as Parkinson's, HAR can be used to compile diaries of their daily activities and detect episodes such as freezing-of-gait events, for assessing the patient's condition [13]. Quantifying physical activity through HAR can also provide invaluable information for other applications, such as evaluating the condition of patients with chronic obstructive pulmonary disease (COPD) [14], [15] or evaluating the recovery progress of patients during rehabilitation [16], [17].

Currently, smartphones, wearable devices, and internet-of-things (IoT) are becoming more affordable and ubiquitous. Many commercial products, such as the Apple Watch, Fitbit, and Microsoft Band, and smartphone apps including Runkeeper and Strava, are already available for continuous collection of physiological data. These products typically contain sensors that enable them to sense the environment, have modest computing resources for data processing and transfer, and can be placed in a pocket or purse, worn on the body, or installed at home. Accurate and meaningful interpretation of the recorded physiological data from these devices can be applied potentially to HAR. However, most current commercial products only provide relatively simple metrics, such as step count or cadence. The emergence of deep learning methodologies, which are able to extract discriminating features from the data, and increased processing capabilities in wearable technologies give rise to the possibility of performing detailed data analysis *in situ* and in real time. The ability to perform more complex analysis, such as activity classification on the wearable device would be advantageous for the aforementioned applications.

The rest of the paper is organized as follows: In Section II, we introduce the current state-of-the-art in machine learning for HAR. Our proposed methodology is then described in Section III. Datasets used for performance evaluation are presented in Section IV along with detailed comparison of the different approaches. Our findings and contributions are concluded in Section V.

Manuscript received July 20, 2016; revised October 19, 2016; accepted November 18, 2016. Date of publication December 23, 2016; date of current version January 31, 2017. This research work was supported by EPSRC reference: EP/L014149/1 Smart Sensing for Surgery project and EPSRC-NIHR HTC Partnership Award (EP/M000257/1 and EP/N027132/1).

The authors are with the Hamlyn Centre, Imperial College London, London SW7 2AZ, U.K. (e-mail: d.ravi@imperial.ac.uk; charence@imperial.ac.uk; benny.lo@imperial.ac.uk; g.z.yang@imperial.ac.uk).

Digital Object Identifier 10.1109/JBHI.2016.2633287

II. Related Work

One of the main challenges, when designing a classification method for time-series analysis, is selecting a suitable set of features for subsequent classification. Recent surveys of research in activity recognition show the diverse range of features and classification methods used [18], [19].

In [20], a simple energy thresholding method applied to frequency analysis of the input data is used for the detection of freezing of gait in Parkinson patients. In other applications, statistical parameters [8], basis transform coding [9], and symbolic representation [10] are often used as "shallow" features to describe time-series data. Methods such as decision trees and support vector machines (SVM) are then trained to classify the data using the given features [21]–[23]. Catal *et al.* [24] proposed a method for HAR that combines multiple classification methods, known as an ensemble of classifiers, to maximize the accuracy that can be attained from each classification method.

Using deep learning methods, such as deep belief networks (DBN), restricted Boltzmann machines (RBM), and convolutional neural networks (CNN), a discriminative set of features can be learnt directly from the input data [3]–[5]. However, for HAR, changes in sensor orientation, placement, and other factors require that deep learning approaches for HAR must use complex designs with many layers in order to discover a complete hierarchy of features to properly classify the raw data. Alsheikh *et al.* [6] demonstrate activity recognition using a method based on DBNs and RBMs formed using multiple hidden layers. A hybrid deep learning and hidden Markov model (HMM) approach is finally used with three 1000 neuron layers. While utilizing additional hidden layers and neurons to improve recognition accuracy is not a significant burden for high-performance computer systems, it makes these methods unsuitable for devices with fewer resources.

A deep learning approach optimized for low-power devices presented in [1] uses a spectrogram representation of the inertial input data to provide invariance against changes in sensor placement, amplitude, or sampling rate, thus allowing a more compact method design. However, the results reported in [1] do not always overcome the accuracy obtained from shallow features, which may be due to resource limitations and the simple design of the method. For this reason, we propose to combine a set of shallow features with those obtained from deep learning in this paper. As far as we know, we are the first that propose to combine efficiently both shallow and deep features with a method that can be executed in real time on a wearable device.

III. Methods

As mentioned previously, in Ravì *et al.* [1], it is shown that features derived from a deep learning method performed on devices with limited resources are sometimes less discriminative than a complete set of predefined shallow features. A possible reason for this behavior may lie in the fact that deep learning methods with less computational layers cannot find the entire hierarchy of features. Another possibility is that, since the extraction of features through deep learning is driven by data, if the dataset is not well represented in all the possible modalities (i.e., location of the sensor, different sensor's properties such as amplitude or sampling rate) the deep learning approach is not capable to generalize these data modalities automatically for the classification task. In these scenarios, shallow features may achieve better performance than deep learning approaches. Consequently, we believe that shallow and deep learnt features provide complementary information that can be jointly used for classification.

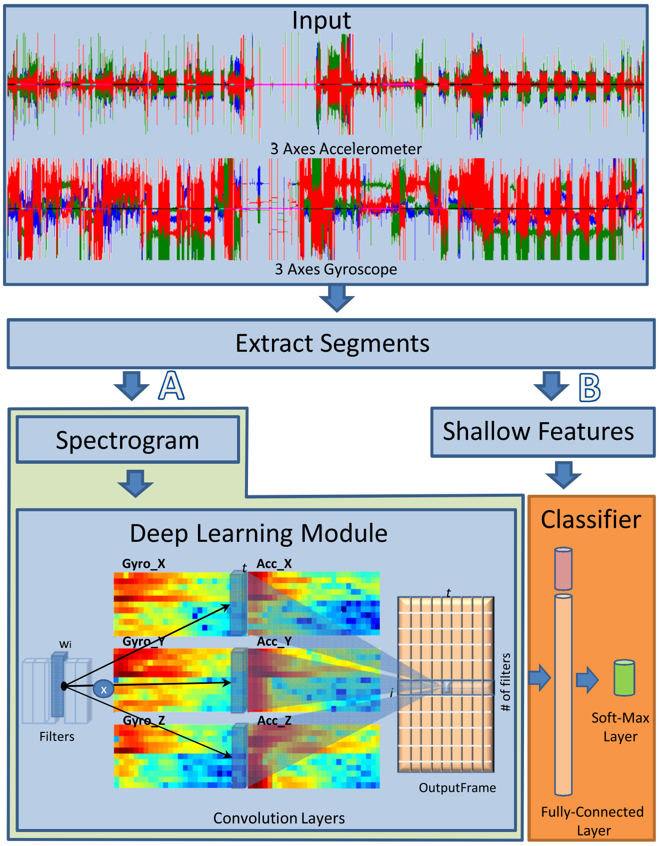


Fig. 1. Schematic workflow of the proposed method: the raw datasets measured by the inertial sensors are collected and divided into segments. The automatically learnt features and the shallow features are extracted in processes A and B, respectively. In the last block, the features are combined together and classified using a fully connected layer and a soft-max layer of the deep learning model.

The pipeline of the proposed approach that combines both shallow and deep learnt features is described in Fig. 1. The first block within the pipeline collects the raw data obtained from the inertial sensors. The second block extracts the input data into segments to be used along both process A and process B of the pipeline where features from a deep learning method and shallow features are computed in parallel. In the final block of the pipeline, these two sets of features are merged together and classified using a fully connected and a soft-max layers. The details of the approach are further explained in Algorithm 1 and each of these blocks are described as follows:

A. Input

For the application of HAR, we will be using inertial sensors, such as accelerometers and gyroscopes for the input block of

```
Algorithm 1: Proposed algorithm.
Input:
    $Acc_X, Acc_Y, Acc_Z$        ▷ Raw Triaxial Acceleration data
    $Gyr_X, Gyr_Y, Gyr_Z$        ▷ Raw Triaxial Gyroscope data
    $kw$                         ▷ Kernel size for 1D convolution
    $sf$                         ▷ Number of frequencies points in the spectrogram
    $st$                         ▷ Number of time-localized points in the spectrogram
    $dw$                         ▷ Step of the convolution
    $wp$                         ▷ Number of filters
    $n$                          ▷ Number of sample in each segment
Output:
    $result$                     ▷ label

                                 ▷ Extract Segment
$a[1], a[2], a[3] \leftarrow Segm(Acc_X, n), Segm(Acc_Y, n), Segm(Acc_Z, n);$
$g[1], g[2], g[3] \leftarrow Segm(Gyr_X, n), Segm(Gyr_Y, n), Segm(Gyr_Z, n);$
                                 ▷ Extract Shallow Features
$s \leftarrow Extract\_shallow\_features(a[1], a[2], a[3], g[1], g[2], g[3])$
                                 ▷ Extract Learnt Features
$S\_a[1], S\_a[2], S\_a[3] \leftarrow Spectr(a[1]), Spectr(a[2]), Spectr(a[3]);$
$S\_g[1], S\_g[2], S\_g[3] \leftarrow Spectr(g[1]), Spectr(g[2]), Spectr(g[3]);$
for $i$=1 to $wp$ do
    for $t$=1 to $sf$ do
        for $z$=1 to 3 do
            $o[t][i]\mathrel{+}=\sum_{j=1}^{st}\sum_{k=1}^{kw} w[i][j][k] * S\_g[z][j][dw*(t-1)+k];$
            $o[t+sf][i]\mathrel{+}=\sum_{j=1}^{st}\sum_{k=1}^{kw} w[i][j][k] * S\_a[z][j][dw*(t-1)+k];$
        end for
    end for
end for
                                 ▷ Combine Features
$cb \leftarrow Combine(s, o);$
                                 ▷ Classification
$fc \leftarrow Fully\_connected(cb);$
$result \leftarrow Soft\_max(fc);$
```

Fig. 1. It is important to note that the described approach also caters for additional time-series data from other sensor types, such as Electrocardiography (ECG) for measuring heart rhythm or Electromyography (EMG) for muscle activity.

B. Extract Segments

After the raw signals are collected, segments of n samples are extracted and forwarded along processes A and B of the pipeline. The number of samples to consider depends on the type of application involved. Of course, increasing the length of the segments can introduce an improvement in recognition accuracy, but at the same time it would cause a delay in response for real-time applications as longer segments of data need to be obtained and the boundary between different activities become less well defined. Typically, segments of 4 to 10 s are used for HAR [6]. The reason that segments rather than single data points are used is motivated by the fact that the highly fluctuating raw inertial measurements make the classification of a single data point impractical [25]. Therefore, segments are obtained using a sliding window applied individually to each axis of the sensor.

C. Spectrogram and Deep Learning Module

In process A of Fig. 1, a set of deep features is automatically extracted using the proposed deep learning module. This module takes advantage of a spectrogram representation and an efficient design to achieve its task. In previous work, Ravì *et al.* [1] show the importance of using a suitable domain when a deep learning methodology is applied to time-series data. Specifically, they show that the spectrogram representation is essential for extracting interpretable features that capture the intensity differences among nearest inertial data points. The spectrogram representation also provides a form of time and sampling rate invariance. This enables the classification to be more robust against data shifting in time and also against changes in amplitude of the signal and sampling rate. Moreover, frequency selection in the spectrogram domain also provides an implicit way to allow noise filtering of the data over time.

A spectrogram of an inertial signal x is a new representation of the signal as a function of frequency and time. Specifically, the spectrogram is the magnitude squared of the short-time Fourier transform (STFT). The procedure for computing the spectrogram is to divide a longer time signal into short segments of equal length and then compute the Fourier transform separately on each shorter segment. This can be expressed as

$$\mathrm{STFT}\{x[n]\}(m,\omega) = X(m,\omega) = \sum_{n=-\infty}^{\infty} x[n]\omega[n-m]e^{-j\omega n} \tag{1}$$

likewise, with signal $x[n]$ and window $w[n]$. The magnitude squared of the STFT yields the spectrogram of the function:

$$\mathrm{spectrogram}\{x(n)\}(m,\omega) = |X(m,\omega)|^2. \tag{2}$$

The resulting spectrogram is a matrix $st \times sf$, where st is the number of different short term, time-localized points and sf is the number of frequencies considered. Therefore, the spectrogram describes the changing spectra as a function of time. In Fig. 2, we show examples of the averaged spectrograms across different activities. As we can see, their representations exhibit different patterns. Specifically, it appears that highly variable activities exhibit higher spectrogram values along all frequencies, instead repetitive activities, such as walking or running, only show high values on specific frequencies. These discriminative patterns can be detected by the deep learning module, which aims to extract features and characterize activities.

Once the spectrograms have been computed, they are processed using the deep learning module. The design of our deep learning module is aimed at overcoming some of the issues typically present in a deep learning framework where on-node computation is required. Specifically, these disadvantages include the following:

1) deep learning modules can contain redundant links between pairs of nodes that connect two consecutive layers of the neural network;
2) correlations in different signal points are usually overlooked; and
3) a large set of layers can be built on top of each other to extract a hierarchy of features from low level to high level.

Deep learning approaches with these designs tend to have high computation demands and are unsuitable for low-power devices being considered in this paper. In our proposed approach, we reduce the computation cost by limiting the connections between the nodes of the network and by computing the features

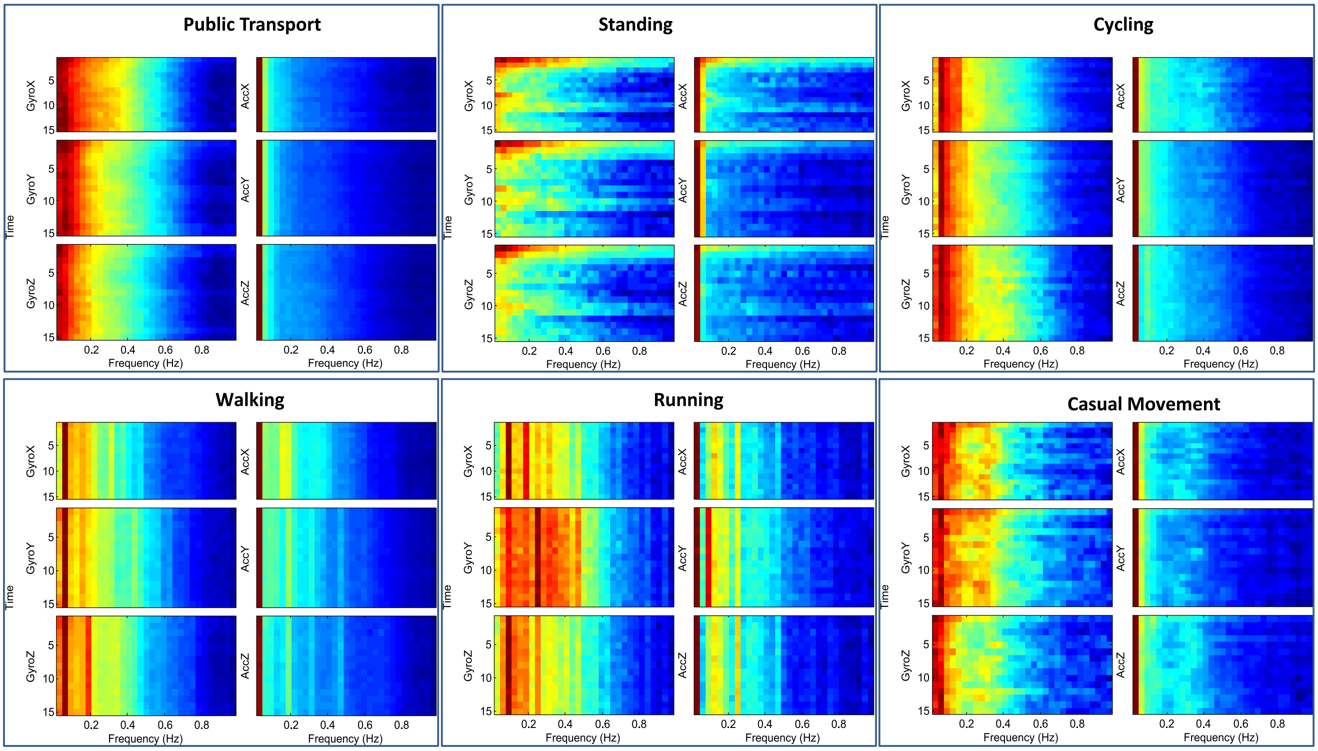


Fig. 2. Examples of averaged spectrograms extracted from different activities of the ActiveMiles dataset. Their representations exhibit different patterns for feature extraction and class recognition.

efficiently through the use of few hidden layers. Specifically, the spectral representations of different axes and sensors are prearranged so that the data represent local correlations and they can be processed using 1-D convolutional kernels with the same principle that CNN [26] follows. These filters are applied repeatedly to the entire spectrogram and the main advantage is that the network contains just a number of neurons equal to a single instance of the filters, which drastically reduces the connections from the typical neural network architecture.

The proposed prearrangement of the spectrograms is shown in the deep learning module of Fig. 1. Here the spectrograms computed on the x, y, and z axes are grouped together column wise while the spectrograms obtained from different sensors are grouped row wise. The processing of our deep learning module is based on the use of sums of local 1-D convolutions over this prearranged input. Since each activity has a discriminative distribution of frequencies, as shown in Fig. 2, the sum is performed in correspondence to each frequency. Specifically, each filter w_i—with size $kw \times st$—is applied to the spectrogram vertically, and the weighted sum of the convolved signal at time t is computed as follows:

$$o[t][i] = \sum_{j=1}^{st} \sum_{k=1}^{kw} w[i][j][k] * \text{input}[j][dw * (t-1) + k] \quad (3)$$

where dw is the stride of the convolution. These convolutions produce an output layer o with size $wp \times \text{OutputFrame}$ with $\text{OutputFrame} = (\text{InputFrame} - kw)/dw + 1$ and wp, the number of filters. The results of the convolution obtained from the x, y, and z axes of an inertial sensor are summed together without any discrimination so that the orientation invariance property is maintained. This helps the proposed deep learning framework to be more generalizable even when variation in the data resulting from different sensor orientation is not well represented in the dataset. The filters applied to the three axes share the same weights, which is important for reducing the number of parameters for each convolution layer.

TABLE I
SHALLOW FEATURES EXTRACTED FROM THE PROPOSED APPROACH AND COMBINED WITH THE LEARNT FEATURES

Input Data	Features		
Raw signal	Interquartile Range Root Mean Square Standard Deviation Mean-cross Zero-cross	Amplitude Variance Skewness Median	Kurtosis Mean Min Max
First derivative	Root Mean Square Standard Deviation	Variance	Mean

D. Shallow Features

In process B of Fig. 1, 17 predefined shallow features are considered. These features, listed in Table I, are extracted separately from each segment of each axis, creating a vector representation for the considered segment. This step is expressed on line 5 in Algorithm 1. In our case, it takes six input segments, $a[1], a[2], a[3], g[1], g[2], g[3]$, representing, respectively, the

TABLE II
SUMMARY OF HUMAN ACTIVITY DATASETS

Dataset	Description	# of Classes	Subjects	Samples	Sampling Rate	Reference
ActiveMiles	Daily activities collected by smartphone in uncontrolled environments	7	10	4,390,726	50 – 200 Hz	[1]
WISDM v1.1	Daily activities collected by smartphone in a laboratory	6	29	1,098,207	20 Hz	[27]
WISDM v2.0	Daily activities collected by smartphone in uncontrolled environments	6	563	2,980,765	20 Hz	[28][29]
Daphnet FoG	Freezing of gait episodes in Parkinson's patients	2	10	1,917,887	64 Hz	[20]
Skoda	Manipulative gestures performed in a car maintenance scenario	10	1	$\sim 701,440$	98 Hz	[30]

accelerometers and the gyroscope data vector along the three axes and produces a final vector of 102 features as output.

E. Classification

Once both deep and shallow features have been computed they are merged together into a unique vector and classified through a fully connected layer and a soft-max layer, as shown by lines 20 and 21 of Algorithm 1.

F. Training Process

Shallow and deep features are trained together in a unified deep neural network. During each stage of the training, errors between the target and obtained values are used in the backward propagation routine to update the weights of the different hidden layers. Stochastic gradient descent (SGD) is used to minimize the loss function defined by the L2-norm. To further improve the training procedure of the weights, we have used three regularizations:

1) Weight decay: it is a term in the weight update rule that causes the weights to exponentially decay to zero if no other update is scheduled. It is used to avoid over fitting.
2) Momentum: it is a technique for accelerating gradient descent and attempting to move the global minimum of the function. It accumulates a velocity vector in directions of persistent reduction in the objective across iterations.
3) Dropout: it is a technique that prevents overfitting and provides a way of combining many different neural network architectures efficiently for consensus purposes. At each iteration of the training, dropout temporarily removes nodes from a neural network, along with all its incoming and outgoing connections. The choice of which units to drop is random and is determined according to a probability p of retaining a node. Training a network with dropout leads to significantly lower generalization error.

IV. EXPERIMENTAL RESULTS

A. Datasets

To evaluate the proposed system, we analyze the performance obtained on complex real world activity data, collected from multiple users. Five public datasets are analyzed using tenfold cross validation. Table II summarizes these datasets. Noteworthy is the release of our dataset, ActiveMiles (available at http://hamlyn.doc.ic.ac.uk/activemiles/), which contains unconstrained real world human activity data from ten subjects collected using five different smartphones. Each subject was asked to annotate the activities they carried out during the day using an Android app developed for this purpose. There are no limitations on where the smartphone is located (i.e., pocket, bag, or held in the hand). Annotations record the start time, end time, and label of a continuous activity. Since each smartphone uses a different brand of sensor, the final dataset will contain data that have many modalities, including different sampling rates and amplitude ranges. It is one of the largest datasets in terms of number of samples with around 30 h of labeled raw data, and it is the first database that groups together data captured using different sensor configurations.

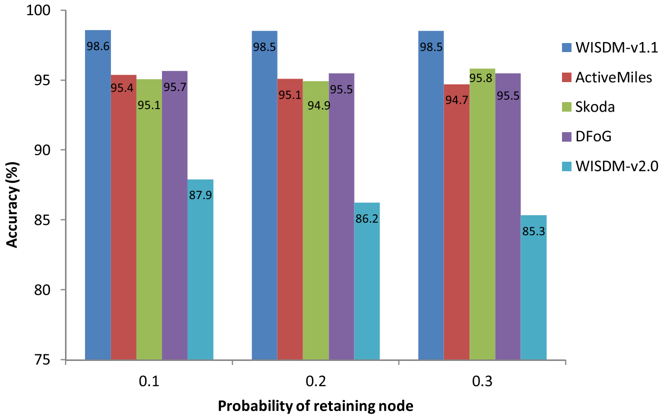


Fig. 3. Behavior of the proposed approach by increasing the probability of retaining a node in the dropout regularization, where size of the convolutional kernel is 2, number of levels is 2, and number of filters is 40.

B. Parameters Optimizations

The proposed deep learning framework contains a few hyperparameters that must be defined before training the final model. An optimization process based on a grid search is proposed to find the best values for the following:

1) the probability of retaining a node during the dropout regularization;
2) the size of the convolutional kernels for all relative convolutional layers;
3) the total number of convolutional layers; and
4) the total number of filters in each convolutional layer.

The behavior of the system when these parameters are systematically tested is shown in Figs. 3, 4, and 5. In Fig. 3, we can infer for datasets that have many classes and large variability, increasing the probability of retaining a node during dropout

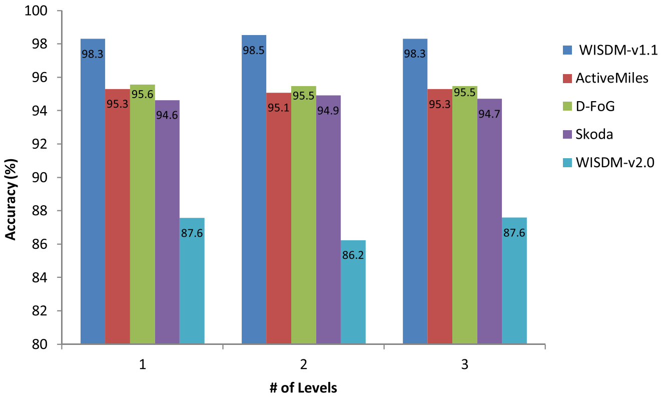


Fig. 4. Behavior of the proposed approach by increasing the number of convolutional levels, where probability of retaining a node is 0.2, size of the convolutional kernel is 2, and number of filters is 40.

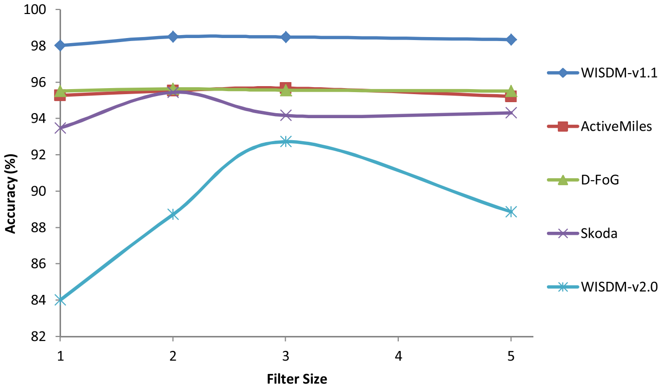
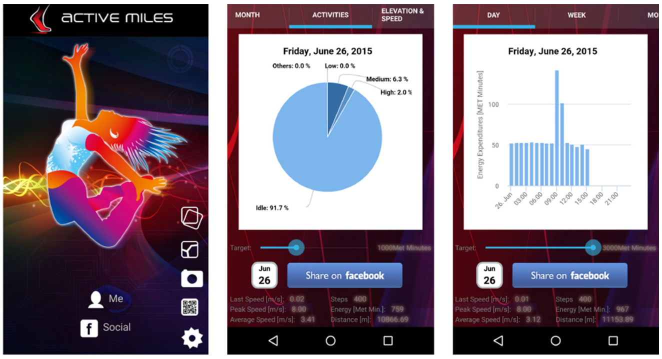


Fig. 5. Behavior of the proposed approach by increasing the size of convolutional kernels in all the convolutional layers, where probability of retaining a node is 0.1, number of levels is 1, and number of filters is 40.

Fig. 6. Intel Edison features a dual-core Intel Atom CPU at 500 MHz, wireless connectivity, and compact physical dimensions at $35.5 \times 25.0 \times 3.9$ mm. It is a small but powerful platform that is well suited for wearable devices.



Target: 1000Met Minutes
Jun 26
Last Speed [m/s]: 0.02
Peak Speed [m/s]: 8.00
Average Speed [m/s]: 3.41
Steps 400
Energy [Met Min.]: 759
Distance [m]: 10866.69

Target: 3000Met Minutes
Jun 26
Last Speed [m/s]: 0.01
Peak Speed [m/s]: 8.00
Average Speed [m/s]: 3.12
Steps 400
Energy [Met Min.]: 967
Distance [m]: 11153.89

(a) (b) (c)

Fig. 7. Graphical user interface used in the app ActiveMiles. (a) Main view presented to the user for browsing all the recorded data of the app. (b) Pie chart shows the percentage of the different activities captured during the day. (c) Histogram is used to show the METS per minute computed in each hour.

makes the results worse. Instead, for datasets with fewer classes and less variability, increasing this probability provides better performance. The results in Fig. 4 show instead that the proposed approach requires only a few levels in order to obtain a complete hierarchy of learnt features and increasing the number of levels does not result in substantial improvement. Finally, Fig. 5 shows that the optimal size of the 1-D convolutional kernels in all the convolutional layers is two or three depending, also in this case, on the complexity of the datasets. For simple datasets, a filter size of two seems to provide the best performance, while for more complex datasets this size needs to be increased up to three. In our experiments, we also discover that 32 filters in each convolutional layer and 80 neuron nodes in the fully connected layer are the optimal values. However, since they show that reducing the number of filters does not affect classification accuracy, the number of filters can be reduced up to ten if there are resource limitations.

C. Implementation

The proposed solution has been implemented for different low-power platforms. The deep learning model is trained offline on a standard workstation (2× Intel Xeon E5-2680v2 CPU, 64GB DDR3 RAM) and then exported onto these platforms. Only the feedforward routine of the deep learning module is used for on-node classification of the data. Specifically, the proposed approach has been deployed as an app for Android devices and also as an embedded algorithm for the Intel Edison Development Platform, Fig. 6. The app is available on the Google Store at the following link: https://play.google.com/store/apps/details?id=org.imperial.activemilespro. The mobile app allows monitoring of human activity throughout the day and a complete summary of the detected activities is presented to the user, as shown in Fig. 7(b). The app also provides an estimation of the metabolic equivalents (METS) per minute captured for each hour of the day [see Fig. 7(c)]. Differently from other energy expenditure metrics, such as Calories or Joules, METS is a biological parameter that expresses a normalized level of physical activity that does not factor in person's weight. In the ActiveMiles app, METS are measured by multiplying the amount of minutes spent for each activity by a relatively well defined constant value and summed together [31], [32] The second implementation of the proposed system is for the Intel Edison (Intel Atom Tangier CPU, 1GB RAM, dimensions: 35.5 × 25.0 × 3.9 mm). This implementation is used to demonstrate the on-node human activity classification. For both architectures, we use the

TABLE III
COMPARISON OF OUR PROPOSED SOLUTION AGAINST EXISTING METHODS IN DIFFERENT DATASET

	Features		Results for each dataset				
Approach	Shallow	Deep	ActiveMiles Accuracy (%)	WISDM v1.1 Accuracy (%)	WISDM v2.0 Accuracy (%)	Skoda (Node 16) Accuracy (%)	Daphnet FoG Accuracy (%)
Shallow-features-only	√	✗	95.0	97.4	92.5	**95.9**	**95.8**
Catal *et al.* [24]	√	✗	91.7	94.3	89.8	86.9	94.8
Alsheikh *et al.*[6]	✗	√	84.5	98.2	82.9	89.4	91.5
Ravì *et al.*[1]	✗	√	95.1	98.2	88.5	91.7	95.7
Ours	√	√	**95.7**	**98.6**	**92.7**	95.3	**95.8**

TABLE IV
PRECISION AND RECALL VALUES OBTAINED BY THE PROPOSED APPROACH IN ALL THE CONSIDERED DATASETS

ActiveMiles

	Running	Walking	Cycling	Casual Movem.	Public Trans.	Idle	Standing
Prec.(%)	95.24	95.46	96.20	96.38	95.82	97.95	74.19
Recall(%)	98.77	96.53	96.64	96.13	95.16	96.52	73.02

WISDM v1.1

	Walking	Jogging	Sitting	Standing	Upstairs	Downstairs
Prec.(%)	99.37	99.64	97.85	98.15	95.52	94.44
Recall(%)	99.37	99.40	98.56	97.25	95.13	95.90

WISDM v2.0

	Walking	Jogging	Stairs	Sitting	Standing	Lying Down
Prec.(%)	97.17	98.01	85.00	87.32	82.05	88.65
Recall(%)	97.19	97.73	76.98	89.28	82.11	85.80

Skoda

	Write on notepad	Open hood	Close hood	Check gaps front door	Open left front door	Close left front door	Close both left door	Check trunk gaps	Open and close trunk	Check steering wheel
Prec.(%)	96.67	97.78	89.47	91.15	100.00	88.89	92.86	98.78	100.00	93.55
Recall(%)	91.34	97.78	94.44	92.79	100.00	80.00	94.20	97.59	98.04	100.00

Daphnet FoG

	No freeze	Freeze
Precision (%)	97.40	67.89
Recall (%)	98.15	59.92

FFTW3 library [33] to extract the spectrogram and the Torch framework [34] to implement the deep learning model.

D. Classification Results

To demonstrate the relative merit of the proposed method, we have compared the performance of the proposed solution against other methods. We have considered a baseline approach that is purely based on shallow features, defined as "shallow-features-only," which corresponds to process B of the pipeline of our combined approach. We also consider approaches that are purely based on deep learning, such as [1], [6] in our evaluation. In particular, the approach in [1] corresponds to process A of our combined pipeline. The classification results are presented in Table III. Nonoverlapping window sizes of 4 s (for Skoda and Daphnet-FoG datasets) and 10 s (for ActiveMiles, WISDM v1.1 and WISDM v2.0 datasets) were used to segment the raw inertial time-series data.

From Table III, it is evident that approaches based entirely on deep learnt features or shallow features do not always provide the best performance. In fact, on the WISDM v2.0 and Skoda datasets, the proposed baseline "shallow-features-only" obtains 4% and 4.2% greater classification accuracy when compared to [1], which only uses deep learnt features. On the other hand, the same approach in [1] obtains 0.1% and 0.8% greater accuracy on the ActiveMiles and WISDM v1.1 datasets than the baseline. Combining both deep learnt and shallow features allows the best of the approaches to be exploited, producing a more generalizable solution that, as shown in our experiments, overcomes the other approaches for all datasets except for the Skoda dataset. While the other deep learning approaches do not perform well on this last dataset, which is most likely due to the small amount of activity segments available (Table I shows there is a low number of samples and a high sampling rate), our combined approach obtains a result that is much closer to the best accuracy, achieved in this case by the "shallow-features-only" method. Therefore, we can conclude that shallow features provide complementary information toward deep learning.

In Table IV, more detailed results about the perclass precision and recall values obtained by the proposed approach are presented. In addition to showing the different classes for each dataset, the table also shows the quality of the multiclass prediction. For most classes, precision and recall is above 85%. For the Daphnet FoG dataset, the under representation of the class "freeze" in the training data is the most plausible reason for the lower precision and recall results observed.

To evaluate if the proposed method could achieve real-time performance on a smartphone or a miniature wearable device, the computation time required to perform the classification task for a 10 s segment of data was measured. Using a LG Nexus

5 smartphone, Samsung Galaxy S5 smartphone, and Intel Edison, to represent the different low-power hardware that can be used for continuous monitoring of human activity, computation times of 53.8, 125.2, and 198.8 ms were obtained, respectively. Marginal differences in computation time exist due to differences in hardware and software, such as CPU, memory, and operating system specification. The computation cost obtained is significantly less than the 10 s segments used, which demonstrates the feasibility of real-time analysis.

V. Conclusion

In this paper, we have presented a method that combines shallow and learnt features from a deep learning approach for time-series data classification. The proposed method is designed to overcome some of the issues typically present in a deep learning framework when on-node computation is required. Thanks to a prearrangement of the data before they are passed into the deep learning framework and thanks to a simplified computation layer the learnt features are computed efficiently through the use of just a few levels. The experiments conducted show that the accuracy of the proposed method is better when compared to the current state-of-the-art approaches. Moreover, the ability of the proposed method to generalize across different classification tasks is demonstrated using a variety of human activity datasets, including datasets collected in unconstrained real world environments. Finally, we show that the computation time obtained from low-power devices, such as smartphones, wearable devices, and IoT, is suitable for real-time on-node HAR.

Acknowledgment

The authors would like to thank Intel Health and Life Sciences (EMEA Innovation Team) for the provision of the Intel Edison Development Platform. The ActiveMiles dataset and code used in this paper are available at http://hamlyn.doc.ic.ac.uk/activemiles/. For data enquiries, please contact hamlyn@imperial.ac.uk.

References

[1] D. Ravi, C. Wong, B. Lo, and G.-Z. Yang, "Deep learning for human activity recognition: A resource efficient implementation on low-power devices," in *Proc. 2016 IEEE 13th Int. Conf. Wearable Implantable Body Sensor Netw.*, Jun. 2016, pp. 71–76.

[2] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst. 25*, 2012, pp. 1097–1105. [Online]. Available: http://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[3] M. Zeng *et al.*, "Convolutional neural networks for human activity recognition using mobile sensors," in *Proc. 2014 6th Int. Conf. Mobile Comput., Appl. Services*, Nov. 2014, pp. 197–205. [Online]. Available: http://dx.doi.org/10.4108/icst.mobicase.2014.257786

[4] J. B. Yang, M. N. Nguyen, P. P. San, X. L. Li, and S. Krishnaswamy, "Deep convolutional neural networks on multichannel time series for human activity recognition," in *Proc. 24th Int. Conf. Artif. Intell.*, 2015, pp. 3995–4001. [Online]. Available: http://portal.acm.org/citation.cfm?id=2832806

[5] Y. Chen and Y. Xue, "A deep learning approach to human activity recognition based on single accelerometer," in *Proc. 2015 IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 1488–1492. [Online]. Available: http://dx.doi.org/10.1109/smc.2015.263

[6] M. A. Alsheikh, A. Selim, D. Niyato, L. Doyle, S. Lin, and H.-P. Tan, "Deep activity recognition models with triaxial accelerometers," Nov. 2015. [Online]. Available: http://arxiv.org/abs/1511.04664

[7] D. G. Lowe, "Object recognition from local scale-invariant features," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*,, 1999, vol. 2. pp. 1150–1157.

[8] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Comput. Survey*, vol. 46, no. 3, Jan. 2014, Art. no. 33. [Online]. Available: http://dx.doi.org/10.1145/2499621

[9] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proc. 2005 Joint Conf. Smart Objects Ambient Intell.: Innovative Context-Aware Services: Usages Technol.*, 2005, pp. 159–163. [Online]. Available: http://dx.doi.org/10.1145/1107548.1107591

[10] J. Lin, E. Keogh, S. Lonardi, and B. Chiu, "A symbolic representation of time series, with implications for streaming algorithms," in *Proc. 8th ACM SIGMOD Workshop Res. Issues Data Mining Knowl. Discovery*, 2003, pp. 2–11. [Online]. Available: http://dx.doi.org/10.1145/882082.882086

[11] G.-Z. Yang, *Body Sensor Networks*, 2nd ed. New York, NY, USA: Springer, 2014.

[12] M. Ermes, J. Prkk, J. Mntyjrvi, and I. Korhonen, "Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions," *IEEE Trans. Inf. Technol. Biomed.*, vol. 12, no. 1, pp. 20–26, Jan. 2008.

[13] P. Angeles, M. Mace, M. Admiraal, E. Burdet, N. Pavese, and R. Vaidyanathan, *A Wearable Automated System to Quantify Parkinsonian Symptoms Enabling Closed Loop Deep Brain Stimulation*. Cham, Switzerland: Springer, 2016, pp. 8–19. [Online]. Available: http://dx.doi.org/10.1007/978-3-319-40379-3_2

[14] L. Atallah, B. Lo, R. Ali, R. King, and G. Z. Yang, "Real-time activity classification using ambient and wearable sensors," *IEEE Trans. Inf. Technol. Biomed.*, vol. 13, no. 6, pp. 1031–1039, Nov. 2009.

[15] F. Pitta, T. Troosters, V. S. Probst, M. A. Spruit, M. Decramer, and R. Gosselink, "Quantifying physical activity in daily life with questionnaires and motion sensors in COPD," *Eur. Respiratory J.*, vol. 27, no. 5, pp. 1040–1055, 2006. [Online]. Available: http://erj.ersjournals.com/content/27/5/1040

[16] K.-H. Chen, P.-C. Chen, K.-C. Liu, and C.-T. Chan, "Wearable sensor-based rehabilitation exercise assessment for knee osteoarthritis," *Sensors*, vol. 15, no. 2, pp. 4193–4211, 2015. [Online]. Available: http://www.mdpi.com/1424-8220/15/2/4193

[17] N. Bidargaddi, A. Sarela, L. Klingbeil, and M. Karunanithi, "Detecting walking activity in cardiac rehabilitation by using accelerometer," in *Proc. 3rd Int. Conf. Intell. Sensors, Sensor Netw. Inf.*, Dec. 2007, pp. 555–560.

[18] M. Shoaib, S. Bosch, O. D. Incel, H. Scholten, and P. J. Havinga, "A survey of online activity recognition using mobile phones," *Sensors*, vol. 15, no. 1, pp. 2059–2085, 2015. [Online]. Available: http://www.mdpi.com/1424-8220/15/1/2059

[19] C. C. T. Clark, C. M. Barnes, G. Stratton, M. A. McNarry, K. A. Mackintosh, and H. D. Summers, "A review of emerging analytical techniques for objective physical activity measurement in humans," *Sports Med.*, pp. 1–9, 2016. [Online]. Available: http://dx.doi.org/10.1007/s40279-016-0585-y

[20] M. Bächlin *et al.*, "Wearable assistant for Parkinson's disease patients with the freezing of gait symptom," *IEEE Trans. Inf. Technol. Biomed.*, vol. 14, no. 2, pp. 436–446, Mar. 2010. [Online]. Available: http://dx.doi.org/10.1109/titb.2009.2036165

[21] Y. Lu, Y. Wei, L. Liu, J. Zhong, L. Sun, and Y. Liu, "Towards unsupervised physical activity recognition using smartphone accelerometers," *Multimedia Tools and Appl.*, pp. 1–19, 2016. [Online]. Available: http://dx.doi.org/10.1007/s11042-015-3188-y

[22] A. Ignatov and V. Strijov, "Human activity recognition using quasiperiodic time series collected from a single tri-axial accelerometer," *Multimedia Tools and Appl.*, Springer, pp. 1–14, 2015. [Online]. Available: http://dx.doi.org/10.1007/s11042-015-2643-0

[23] K. H. Walse, R. V. Dharaskar, and V. M. Thakare, "A study of human activity recognition using AdaBoost classifiers on WISDM dataset," *Inst. Integrative Omics Appl. Biotechnol. J.*, vol. 7, no. 2, pp. 68–76, Jan. 2016.

[24] C. Catal, S. Tufekci, E. Pirmit, and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition," *Appl. Soft Comput.*, vol. 37, pp. 1018–1022, Dec. 2015. [Online]. Available: http://dx.doi.org/10.1016/j.asoc.2015.01.025

[25] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE Commun. Surveys Tut.*, vol. 15, no. 3, pp. 1192–1209, Jul.-Sep. 2013. [Online]. Available: http://dx.doi.org/10.1109/surv.2012.110112.00192

[26] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.

[27] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newslett.*, vol. 12, no. 2, pp. 74–82, Mar. 2011. [Online]. Available: http://dx.doi.org/10.1145/1964897.1964918

[28] J. W. Lockhart, G. M. Weiss, J. C. Xue, S. T. Gallagher, A. B. Grosner, and T. T. Pulickal, "Design considerations for the WISDM smart phone-based sensor mining architecture," in *Proc. 5th Int. Workshop Knowl. Discovery Sensor Data*, 2011, pp. 25–33. [Online]. Available: http://dx.doi.org/10.1145/2003653.2003656

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

[30] P. Zappi *et al.*, "Activity recognition from on-body sensors: Accuracy-power trade-off by dynamic sensor selection," in *Wireless Sensor Networks*. Berlin, Germany: Springer, 2008, vol. 4913, ch. 2, pp. 17–33. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-77690-1_2

[31] B. E. Ainsworth *et al.*, "2011 compendium of physical activities: a second update of codes and met values," *Med. Sci. Sports Exercise*, vol. 43, no. 8, pp. 1575–1581, 2011.

[32] D. Ravi, B. Lo, and G.-Z. Yang, "Real-time food intake classification and energy expenditure estimation on a mobile device," in *Proc. 2015 IEEE 12th Int. Conf. Wearable Implantable Body Sensor Netw.*, Jun. 2015, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/BSN.2015.7299410

[33] M. Frigo and S. G. Johnson, "The design and implementation of fftw3," *Proc. IEEE*, vol. 93, no. 2, pp. 216–231, Feb. 2005.

[34] R. Collobert, S. Bengio, and J. Mariéthoz, "Torch: A modular machine learning software library," IDIAP, Martigny, Switzerland, Tech. Rep. EPFL-REPORT-82802, 2002.

Authors' photographs and biographies not available at the time of publication.