Human activity recognition using machine learning techniques in a low-resource embedded system

Ilana Stolovas
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería, UdelaR
Montevideo, Uruguay
ilana.stolovas@fing.edu.uy

Santiago Suárez
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería, UdelaR
Montevideo, Uruguay
ssuarez@fing.edu.uy

Diego Pereyra
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería, UdelaR
Montevideo, Uruguay
diego.pereyra@fing.edu.uy

Francisco de Izaguirre
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería, UdelaR
Montevideo, Uruguay
fdeizaguirre@fing.edu.uy

Varinia Cabrera
Instituto de Ingeniería Eléctrica
Facultad de Ingeniería, UdelaR
Montevideo, Uruguay
vcabrera@fing.edu.uy

***Abstract*—Human activity recognition aims to infer a person's actions from a set of observations captured by several sensors. Data acquisition, processing and inference on edge devices add a complexity factor to the task, as they involve a trade-off between hardware efficiency and performance. We present a prototype of a wearable device that identifies a person's activity: walking, running or staying still. The system consists of a Texas Instruments MSP-EXP430G2ET launchpad, connected to a BOOSTXL-SENSORS boosterpack with a BMI160 accelerometer. The designed prototype can take acceleration measurements, process them and either transmit them to a computer or classify the activity in the microcontroller. Additionally, our system has LEDs to display coloured signals according to the inferred activity in real-time. The classification algorithm is based on the calculation of statistical features (mean, standard deviation, maximum and minimum) for each accelerometer axis, the application of a dimensionality reduction algorithm (LDA, Linear Discriminant Analysis) and an SVM (Support Vector Machines) classification model.**

***Index Terms*—Human Activity Recognition, Acceleration Sensor, Linear Discriminant Analysis, Support Vector Machines.**

I. Introduction

The acquisition and interpretation of signals from human activity monitoring have been of special interest for scientific research and development over the last decades. The ability to analyse different circumstances and make decisions based on their correct interpretation has improved society's lifestyle. Human movement has contributed to the treatment of many mobility-impaired disorders, such as osteoarthritis, multiple sclerosis and Parkinson's disease [1], fall detection [2], fitness applications [3] and beyond. These applications use accelerometers connected to devices with varying degrees of wearability, from smartphones [2] to waist-mounted devices [4].

This work aims to prototype a system that applies machine learning techniques for human activity recognition in an embedded system, performing real-time inference in a microcontroller. The main challenge and motivation are to align two seemingly opposite disciplines: the development of low-resource embedded software and machine learning algorithms for activity inference. The first one is related to electronics field in terms of optimising memory usage and power consumption, whereas the second one is often supported by a large amount of data combined with an important processing unit and storage.

Related work has been done in machine learning for applications on edge devices with varying degrees of complexity. The developed models include Decision Trees, Neural Networks, Naive Bayes, Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA), k-Nearest Neighbours and Supporting Vector Machines (SVM) ([1], [5], [6]).

We are strongly based on [7], which performs LDA for dimensionality reduction, and [8] adds SVM for classification. These algorithms are described in detail in section II-B. In contrast to [9], the designed system successfully classifies the activity performed by the user with no need for external (remote) processing.

II. Methods

The following sections describe the approach used to implement an accurate human activity recognition system using machine learning techniques on a microcontroller in real-time.

A. Proposed Solution

The physical system is designed to be wearable by a human being to classify the activity being done by the user in real-time. The MSP430G2553 microcontroller plugged into an MSP-EXP430G2ET launchpad (from Texas Instruments) allows quick prototyping while providing enough processing capabilities. A BOOSTXL-SENSORS Texas Instrument boosterpack provides 3-axis acceleration measurements by including a BMI160 Bosch Inertial Measurement Unit. A 16KB flash memory and 512 bytes RAM are available.

This work was partially funded by Comisión Académica de Posgrado (CAP, UdelaR), Espacio Interdisciplniario (EI, UdelaR) and Comisión Sectorial de Investigación Científica (CSIC, UdelaR) "Proyecto I+D: Sistema electrónico para la caracterización del comportamiento de ovinos".

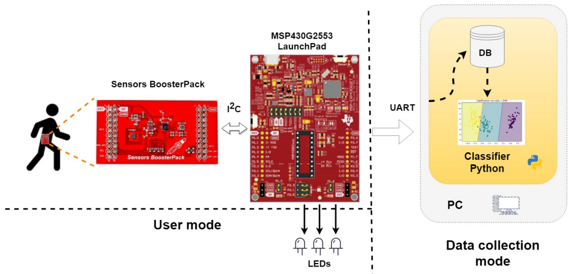


Fig. 1: Proposed solution. Data acquisition and preprocessing are the same for both operation modes. Features go to PC for **data collection mode**, whereas **user mode** classifies the activity and displays the result on the LEDs.

The activities to be classified are: running, walking and staying still. Daily activities detectable insight are important for validation purposes. Moreover, to help validation, three LEDs (included in the launchpad) indicate which activity the system is recognising with obtained data. Each LED indicates a different activity when turned on.

This system uses a machine learning algorithm, so a database is required to train it. This database was created by acquiring accelerometer data, extracting characteristics on the microcontroller and sending them to a PC for training.

The system has two operational modes, one for training and one for classifying. Data gathering happens in **data collection mode**, while the real-time classification and LED display occurs in **User mode**. Figure 1 shows a functional system description.

Some elements are shared between both modes. As an example, the raw acceleration data is sampled at a frequency of 100 Hz from the accelerometer in both modes. The sampling frequency was chosen in accordance to [7]. These measurements are sent one at a time to the microcontroller via the I^2C protocol at a frequency of 100 kHz, allowing the message to be sent completely from the boosterpack to the microcontroller before next sample is taken.

Furthermore, once a data sample has reached the microcontroller, a preprocessing step begins. The data samples information is accumulated until the information corresponding to a time window of 256 data samples is reached. A detailed explanation of this process is presented in section II-C.

Once the samples are preprocessed, statistically significant features are extracted for each axis. These features comprise the mean, standard deviation, maximum and minimum, totaling twelve features from [7] within the machine learning algorithm's single input data vector.

In **data collection mode**, these features will be sent to a PC via UART, at a baud rate of 9600 bps, and a Python-written program receives and stores features in a comma-separated values (csv) file. Two hundred feature vectors were collected for each activity in order to train the algorithm. In **user mode**, features are used to classify the activity the user is doing at the time, and display the results in the LEDs system.

B. Activity Recognition

Once the data has been collected and properly labelled, a supervised machine learning algorithm is trained. In this case, the employed algorithm combines a dimensionality reduction technique and a linear classifier. Linear Discriminant Analysis (LDA) performed dimensionality reduction. Support Vector Machine (SVM) did low-dimension classification. In the relevant literature, LDA is described both as a dimensionality reduction technique [10] and as a classification algorithm [7]. In this work, LDA is used only as its dimensionality-reduction version.

1) LDA: This algorithm finds the optimal subspace in which to project the original data. Each data point is represented by a d-dimensional input feature vector $\mathbf{x} = [x_1, x_2, ..., x_d]^t$ associated with a class $k \in \{1, 2, ..., C\}$. Finding a subspace in which to project this data is equivalent to computing a projection matrix $\mathbf{W}$, so that $\mathbf{y} = \mathbf{W}^t\mathbf{x}$ is in the desired subspace for every point in the original space.

The constraints on the matrix $\mathbf{W}$ define what should be an optimal projection. For LDA, these constraints imply that the projected means of all the classes should be as far as possible from each other (maximising their variance) and that the projected data points of the same class should be as close as possible (minimising their variance). Both constraints form a maximisation problem:

$$\max_{\mathbf{w}:\|\mathbf{w}\|=1} \frac{\mathbf{w}^t\mathbf{S}_b\mathbf{w}}{\mathbf{w}^t\mathbf{S}_w\mathbf{w}} \tag{1}$$

Finding the solution to this problem yields a direction in which to project the original, high-dimension data. If more project directions are required, the general form of the solution allows selecting other specific directions.

Both $\mathbf{S}_b$ and $\mathbf{S}_w$ can be functions of the original data in the following way:

$$\mathbf{S}_b = \sum_{j=1}^{C} n_j(\mathbf{m}_j - \mathbf{m})(\mathbf{m}_j - \mathbf{m})^t \tag{2}$$

It can be seen that equation 2 has the form of a Rayleigh quotient, which has as a solution the eigenvector corresponding to the largest eigenvalue of $\mathbf{S}_w^{-1}\mathbf{S}_b$. Furthermore, more eigenvectors from the matrix $\mathbf{S}_w^{-1}\mathbf{S}_b$ can be used, if more than one direction is required. It is proven in [11] that this matrix has at most, rank $C-1$. Thus, we can project the original data in $C-1$ dimensions.

Finally, for LDA, it is proven in [10] that its time complexity (for its usual implementation) is $O(ndt + t^3)$, being n the number of samples, d the number of features and $t = \min\{n, d\}$. This may be computationally complex. However, our solution only involves LDA for dimensional reduction purposes, performing the projection inside the microcontroller. As this is just a multiplication of a $k \times d$ matrix (being k the number of dimensions selected to project into) by a d-dimensional vector, this operation has a time complexity of $O(dk)$. Therefore, having a moderate size for d and k makes adequate to implement this operation in the microcontroller.

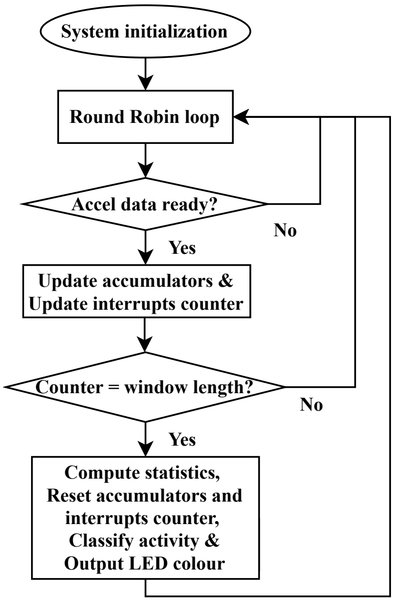


Fig. 2: Main application flow

2) SVM: For the final stage of our activity recogniser, an SVM model has been implemented to classify among the three defined activities. SVM is a supervised machine learning algorithm that aims to find the hyperplane best separating two data classes. The optimal hyperplane can be found by solving an optimisation problem, where each boundary has a separating margin. Violating the margin means a misclassification, which should be minimised. In case more than two classes are present, a one-against-all technique is applied. The model output has numerous decision boundaries for multi-class classification. The dimension of the resulting hyperplane is $d-1$, with d the dimension of the data (the number of features from the input space). For the Python implementation, a linear kernel was used, with an error control of C=1.

We applied the LDA model to 12-dimension feature vectors, with a total of 3 classes. LDA can project the original data on a plane (2D) or a line (1D). In the first case, the decision boundary is a line (2 parameters: slope and y-intercept), and in the second is just a point. Both models were trained in Python using the Scikit-learn [12] library. The data collected via UART was received and stored on a data frame for the training purpose. Fitting the LDA model results on a projection 12x2 matrix, easy to store in the microcontroller flash memory. SVM learnt parameters were also stored to use as a classification threshold in the edge device.

C. Embedded Software

The main application implements a Round Robin with Interrupts architecture. The microcontroller receives data from the accelerometer to process and classify activities generating a LED output, as shown in figure 2.

The most relevant developed modules are the following:

- **Data acquisition**. This module is responsible for the communication between the accelerometer and the microcontroller. It supports I^2C protocol and makes use of the BMI160 sensor driver [13]. The sensor sampling rate is set to $100\,\mathrm{Hz}$ for a range of ± 4 g.
- **Feature calculation**. When receiving data from the accelerometer, relevant information should be stored to compute the statistical features. As mentioned in section II-A, the statistical features computed from the raw accelerometer data are minimum, maximum, mean and standard deviation. These features are extracted from a time window corresponding to 256 samples. The design choice for the window length was based on [14], considering that 2 seconds of sampling is sufficient for an optimal trade-off between recognition speed and accuracy. Considering that hardware resources are limited, estimators of the statistical features were used to allow partial calculations each time a data sample was received from the accelerometer, avoiding storing all sampled data. These estimators are:

$$\hat{a}_{max} = \max_n\{a_n\} \qquad \hat{a}_{min} = \min_n\{a_n\} \tag{3}$$

$$\hat{\mu}_a = \frac{1}{N}\sum_{n=1}^{N} a_n \tag{4}$$

$$\hat{\sigma}_a = \sqrt{\frac{1}{N}\sum_{n=1}^{N} a_n^2 - \left(\frac{1}{N}\sum_{n=1}^{N} a_n\right)^2} \tag{5}$$

  Maximum and minimum values are updated for each accelerometer sample. The axis sum and square-sum are updated, and when a full-time window is sampled, the estimators are computed.
- **Classification**. The train parameters mentioned in section II-B are hard-coded in the flash memory: the projection 12x2 matrix for dimensional reduction and SVM coefficients determining the decision boundary for classification. The inference is achieved simply by multiplying the computed features by the LDA matrix and thresholding the result by the SVM.

Other modules not mentioned above were implemented to send the statistical features to the PC for training (in **Data collection mode**), handling UART protocol.

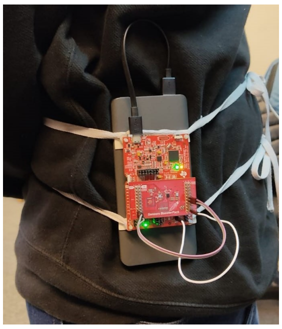
III. Results and Discussion

The system is designed to be tied to the right side of the user's abdomen. For data collection mode, the system is connected to a laptop -carried in a backpack- via a micro-USB to USB cable (figure 3a). For user mode, the launchpad is connected to a powerbank (figure 3b).

The BMI160 acceleration measurements are read from the microcontroller successfully and correctly preprocessed within the time constraints the system had ($100\,\mathrm{Hz}$ sampling rate).

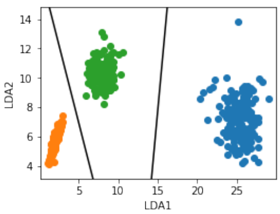
A projection matrix was obtained by training the LDA algorithm, transforming the data into two dimensions. This projection is shown in figure 4a. Because the data clusters are significantly split, we decided to use only one dimension to project the original data into (figure 4b). By these means,

(a) Data collection mode

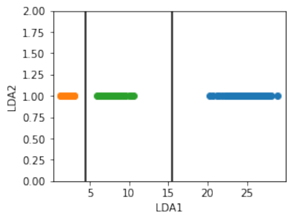


(b) User mode

Fig. 3: Physical system setup



(a) LDA Projection 2D



(b) LDA Projection 1D

Fig. 4: LDA for activity classification. **Blue**: running, **orange**: staying still, **green**: walking.

TABLE I: Classification metrics

Activity	*Precision*	*Recall*	*F1-score*
Staying Still	0.9485	1.0	0.9736
Walking	0.9892	0.9485	0.9684
Running	1.0	0.9897	0.9948
Global	0.9792	0.9793	0.9789

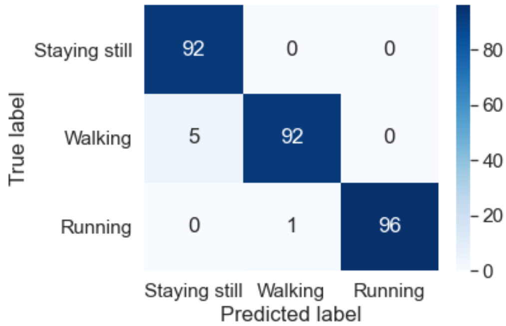


Fig. 5: Testing results on classification performance.

we would make the real-time processing faster, as the computations required for classification would be cut in half.

The embedded software for **user mode** takes less than 6 KB flash memory and 240 B of RAM, less than half of the available resources mentioned in section II-A. Besides, a maximum current consumption of 712 µA was achieved in **user mode**, excluding the LEDs, using low-power mode feature.

Lastly, a successful classification was achieved by the final system, tested on four individuals of different physical appearance (varying height and body shape). We took four minutes sampling for each activity, and video-recorded the output LEDs. The confusion matrix is given in figure 5, reaching an average precision of 97.92% and an average recall of 97.93%. The fact that for staying still we performed 100% precision, and for running we reached 100% recall is a concern. This could be because the training dataset was exclusively comprised of young people. An older person might run extremely similar to a young individual fast walk. Then, results would indicate that samples are biased. Misclassification occurs to "lower movement" states (predicting walking when running, and staying still when walking). Precision, recall and F1-score metrics for each activity are reported in Table I. The global values reported are the macro averages of each metric.

IV. Conclusion

The designed prototype is able to successfully recognise the performed activity through the inertial measurements acquired with the accelerometer and processed in the microcontroller in real-time. An efficient algorithm was designed to compute statistical features of the signal without requiring to store all acquisitions. With these features, an activity database has been correctly built in a PC, training a classification algorithm which was finally embedded in the microcontroller.

The application of LDA and the identification of comparison thresholds with SVM made it possible to achieve a successful activity recognition by computing simple operations on the microcontroller. Moreover, the classification results visible with the LEDs are consistent and distinguished for the selected activities: running, walking and staying still.

Future work would include more activities to recognise (e.g., climbing stairs) for the performed prototype system. Furthermore, the output LEDs could be replaced with a lower power-consumption system feedback. In addition, the dataset could be expanded, including individuals from different age groups and characteristics. The system is intended for a target public of young people, however, it could be interesting to use our system with disabled or elderly people.

References

[1] A. Godfrey, R. Conway, D. Meagher, and G. ÓLaighin, "Direct measurement of human movement by accelerometry," *Medical engineering & physics*, vol. 30, no. 10, pp. 1364–1386, 2008.

[2] M. A. Habib, M. S. Mohktar, S. B. Kamaruzzaman, K. S. Lim, T. M. Pin, and F. Ibrahim, "Smartphone-based solutions for fall detection and prevention: challenges and open issues," *Sensors*, vol. 14, no. 4, pp. 7181–7208, 2014.

[3] J. Qi, P. Yang, M. Hanneghan, S. Tang, and B. Zhou, "A hybrid hierarchical framework for gym physical activity recognition and measurement using wearable sensors," *IEEE Internet of Things Journal*, vol. 6, no. 2, pp. 1384–1393, 2018.

[4] M. J. Mathie, A. C. Coster, N. H. Lovell, and B. G. Celler, "Accelerometry: providing an integrated, practical method for long-term, ambulatory monitoring of human movement," *Physiological measurement*, vol. 25, no. 2, p. R1, 2004.

[5] E. S. Fogarty, D. L. Swain, G. M. Cronin, L. E. Moraes, and M. Trotter, "Behaviour classification of extensively grazed sheep using machine learning," *Computers and Electronics in Agriculture*, vol. 169, p. 105175, 2020.

[6] J. W. Kamminga, H. C. Bisby, D. V. Le, N. Meratnia, and P. J. Havinga, "Generic online animal activity recognition on collar tags," in *Proceedings of the 2017 ACM International Joint Conference on*

Pervasive and Ubiquitous Computing and Proceedings of the 2017 ACM International Symposium on Wearable Computers, 2017, pp. 597–606.

[7] S. P. Le Roux, J. Marias, R. Wolhuter, and T. Niesler, "Animal-borne behaviour classification for sheep (dohne merino) and rhinoceros (ceratotherium simum and diceros bicornis)," *Animal Biotelemetry*, vol. 5, no. 1, pp. 1–13, 2017.

[8] R. Ahmed Bhuiyan, N. Ahmed, M. Amiruzzaman, and M. R. Islam, "A robust feature extraction model for human activity characterization using 3-axis accelerometer and gyroscope data," *Sensors*, vol. 20, no. 23, p. 6990, 2020.

[9] J. Wannenburg and R. Malekian, "Physical activity recognition from smartphone accelerometer data for user context awareness sensing," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 47, no. 12, pp. 3142–3149, 2016.

[10] D. Cai, X. He, and J. Han, "Training linear discriminant analysis in linear time," in *2008 IEEE 24th International Conference on Data Engineering*. IEEE, 2008, pp. 209–217.

[11] P. Xanthopoulos, P. M. Pardalos, and T. B. Trafalis, "Linear discriminant analysis," in *Robust data mining*. Springer, 2013, pp. 27–33.

[12] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[13] BoschSensortec, https://github.com/BoschSensortec/BMI160_driver.

[14] O. Banos, J.-M. Galvez, M. Damas, H. Pomares, and I. Rojas, "Window size impact in human activity recognition," *Sensors*, vol. 14, no. 4, pp. 6474–6499, 2014.