Real-Time Human Physical Activity Recognition with Low Latency Prediction Feedback Using Raw IMU Data

Quentin Mascret*, Mathieu Bielmann†‡, Cheikh-Latyr Fall*, Laurent J. Bouyer†‡ and Benoit Gosselin*

***Abstract*— In the realm of Human Activity Recognition (HAR), supervised machine learning and deep learning are commonly used. Their training is done using time and frequency features extracted from raw data (inertial and gyroscopic). Nevertheless, raw data are seldom employed. In this paper, a dataset of able-bodied participants is recorded using 3 custom wireless motion sensors providing embedded IMU and sEMG detection and processing and a base station (a Raspberry Pi 3) running a classification algorithm. A Support Vector Machine with Radius Basis Function Kernel (*RBF-SVM*) is augmented using Spherical Normalization to achieve a motion classification accuracy of 97.35% between 8 body motions. The proposed classifier allows for real-time prediction callback with low latency output.**

I. INTRODUCTION

In remote areas, such as the North of Canada, health services are limited. Preventing falls or abnormal behavior is therefore a crucial issue. Monitoring movement and posture during activities of daily living (ADLs) with wearable technologies is an effective way to identify risky motor behaviors. Few portable and miniature data acquisition systems and analysis software currently support the continuous recording and real time processing of human movement. This pilot study is part of a larger initiative aimed at developing and implementing better miniature portable technologies for the prevention, assessment and treatment of movement disorders in Northern areas.

Human body motions, gestures and posture characteristics vary from one person to another based on age, height, weight, gender and abilities. Activities can be split into three categories: brief event (transition between gestures), basic (static gestures), complex (dynamic gestures) [1], [2] and measured using various type of sensors. Surface Electromyography (sEMG), motion sensors embedded in smartphones [3], [4], inclinometers and goniometers [5], commercial wearable sensors [1], [6] can be used to sense and classify body motions and postures during ADLs [2]. Many studies compare different supervised machine learning models (and some unsupervised models) to classify them [1]. K-nearest neighbour method (k-NN), polynomial SVM or *RBF-SVM* [3], Random Forests (RF) are commonly used. Most of them using frequency and time features extracted from accelerometer and gyroscopic raw data (the combination give better results) [3], [6]. Recent studies focus on Deep Learning Model as Convolutional Neural Networks (CNN), Deep Neural Networks (DNN) or and Deep Convolutional Neural Networks (DCNN) provided superior performances (as in Speech Recognition and image classification). Images are generated from acceleration and angular velocity and processed with DCNN [6]. Even if researchers performed features selection and sample-rate reduction [7] to decrease computational costs, these approaches are attractive for imlpementation on an embedded platform in terms of performance and energy consumption. Unsupervised learning machines using huge training dataset can achieve an accuracy of 96.53% with k-NN method [1].

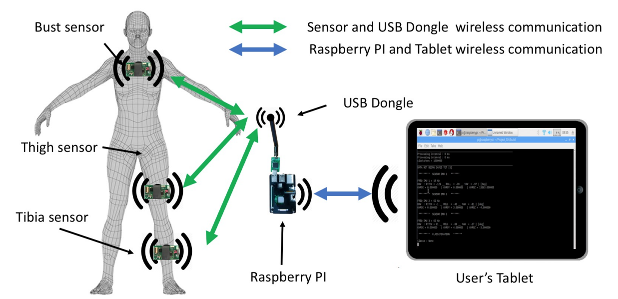


Fig. 1. System overview

This article proposes an accurate *RBF-SVM* using raw data collected from inertial measurement unit (IMU) sensors (acceleration, angular velocity, magnetometer data) and derives the orientation angles (Pitch, Roll, Yaw) in 3D. A new preprocessing method *spherical normalization*, using an hyper-sphere is introduced to normalize input features to improve classification accuracy, and to reduce the size of the required dataset. The overall system, including the data collection, spherical normalization and the classifier, runs on a Raspberry Pi 3 (RPI3) and operates in real time, allowing a maximum output inference latency of 120 ms during prediction phase. Section II describes the data acquisition and signal processing architecture used for gesture prediction. Section III explains the *RBF-SVM* implementation and the proposed *spherical normalization (SN)*. Finally, Section IV compares and analyzes the performance of the proposed system, with and without the *SN* algorithm, to showcase the advantage of using it in terms of performance and computational costs.

II. DATA MEASUREMENT AND SIGNAL PROCESSING

Unobtrusively performing body activity sensing, while performing ADLs, should be fully embedded and wearable.

*Department of Computer and Electrical Engineering. Laval University, Quebec, Canada.
†Center for Interdisciplinary Research in Rehabilitation and Social Integration, Quebec, Canada
‡Department of rehabilitation, Laval University, Quebec, Canada

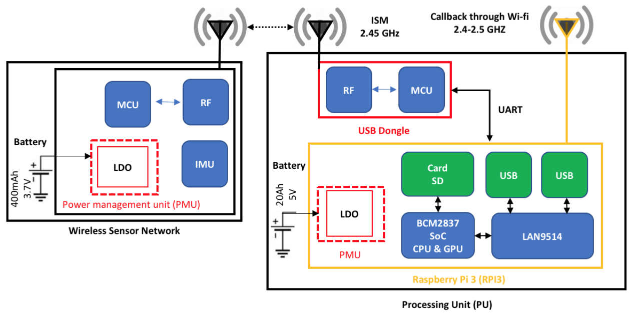


Fig. 2. Diagram of the system describing the sensor node and processing unit architectures.

The system signal processing and software architecture needs to be optimized to enhance power autonomy and lower inference cost. The next subsections provides an overview of the implementation of the proposed system from sensor nodes to signal processing and embedded system architecture.

A. Hardware Architecture

Each participant's physical activity was recorded using custom and low-cost wireless IMU sensors nodes described in [8]. It lies on a 4.0 cm x 2.5 cm printed circuit board with a weight of 12 grammes and features a 9-axis IMU operating at a low sampling rate of 62Hz. Measured data is transmitted with low latency using the *nRF24L01* wireless chip from *Nordic Semiconductor*. They are powered using 3.7V 400 mAh Li-ion rechargeable batteries. Data processing is implemented on a Raspberry Pi 3 Model B connected to a USB dongle receiver which gathers data from all sensors nodes (see Fig.2). Incoming data and callback are shared through a Tablet in real-time using Virtual Network Computing (VNC). The raspberry PI is powered by a nomad battery (AUKEY).

B. Placement of Wearable Sensors

Numerous studies investigated the required numbers of wearable sensors and the most optimal placement for inertial and gyroscopic measurements. Indeed, their position has a direct effect on the measurement results of body motions. Based on the literature, up to 11 sensor locations are recommended for body motion sensing purpose, as depicted in [1]. However, in [9], [10], authors identified 3 locations, namely chest, thigh and ankle, regardless of the body side (right or left), thus requiring only 3 sensors nodes. Ankle sensor was moved to the tibia to measure accurately knee angle for future work during monitoring (see Fig.3).

C. Dataset Recording

This work aims at proposing an embedded and portable system to prevent "risky motor behaviors". A set of 8 activities of interest, described in Table I, have been identified and targeted for use with the recognition algorithm. Participants performed the different motion classes during 2 minutes each, while arbitrarily varying intensity, speed and body position. A non-overlapping window size of 450 ms (1080 samples) is used for training and prediction. The

TABLE I

LIST OF SELECTED ACTIVITIES (FROM A1 TO A8)

Reference	Description of Activity	Category
A41	Standing to seated/seated to standing	Brief
A2	Walking	Complex
A3	Stairs ascend and descent	Complex
A4	Standing	Basic
A5	Seated	Basic
A6	Lying	Basic
A7	Lying to seated/Seated to lying	Brief
A8	Running	Complex

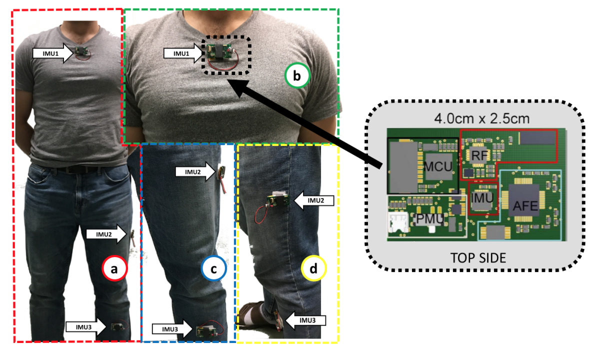


Fig. 3. Wearable IMU Sensors Placements : IMU1 (Bust), IMU2 (Thigh) and IMU3 (Tibia). a) Front view of the subject in *Standing* posture, b) IMU1 attached to the Bust, IMU2 on Left leg, front and side views, c) and d), respectively.

experimental dataset is composed of 8 labeled activities, with 260 arrays of 1080 samples. From each IMU node *IMUi*, a feature vector V^i = [3D Acc, 3D Gyro, 3D Mag, Pitch, Roll, Yaw] is composed from the recorded data. Thus, motion features from Bust, Thigh and Tibia, from *IMU1*, *IMU2* and *IMU3*, respectively, are concatenated to form a 36 dimensional vector ($V_{in} = [V^1 \ V^2 \ V^3]$). Pitch, Roll and Yaw angles are obtained from IMU raw data using a complementary filter [8]. The experimental protocol has been approved by the Laval University Ethics committee (approbation number: #2017-539).

III. CLASSIFIER OVERVIEW

Over-fitting is of the most well-known problem in machine learning and deep learning. Maximizing the margin between all adjacent groups is not the only goal even if very important. If a non-linear kernel function is used, then the smoothness of the kernel function also has an effect on the complexity of the classifier, and hence, on the risk of over-fitting [11]. In the case of Radial Basis Function Kernel, the performance of the SVM depends on the selection of regularization and kernel parameters (C and γ) [13]. It is possible to prevent over-fitting by tuning the hyper-parameters using cross validation.

A. Support Vector Machine With RBF Kernel

Support Vector Machine, introduced by Vapnik [14], is a set of supervised learning methods used for classification, regression analysis and outliers detection. It is a binary classifier. This machine learning technique minimizes empirical

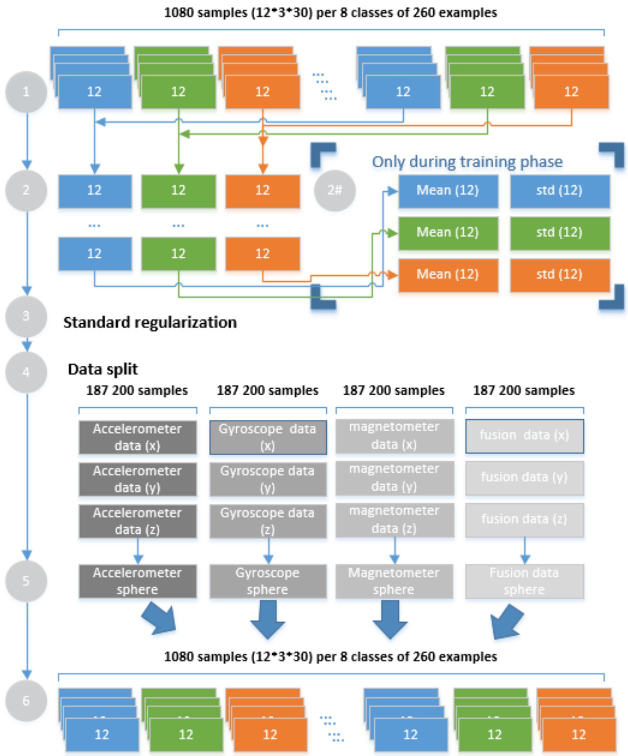


Fig. 4. Flow chart of the sphere normalization process. Blue (IMU1), Green (IMU2); Orange (IMU3). The number ”12” is the number of data inside the buffer of each sensor.

risks and maximizes the margin between the separating hyperplans and the data of two classes. Non-linear classification can be achieved using different kernel methods such as RBF or polynomial. Kernel methods project data from the original data space into a high dimensional space called feature space. Pairwise classifications (One vs All, One vs Rest) ensure classification when using multiple classes.

B. Spherical Normalization

SVM is sensitive to the way features are scaled. Hence, normalizing either the data or the kernel itself is essential. Conventional normalization (subtracting its mean and dividing by its standard deviation) is not suitable when the data is sparse since it removes data sparsity. Spherical Normalization (SN) is a normalization technique introduced in Speaker Verification. It is a preconditioning step employing a transformation that maps each feature vector on the surface of a unit hypersphere [15].

In this work, a new *SN* technique inspired from the *Ellipsoid fit function* [16] and the *spherical vector-length normalization* described in [15], is introduced. Instead of mapping the input vector on a unit sphere, raw data are scaled on the periphery of four unit hyperspheres. The four hyperspheres are generated respectively with acceleration axis, velocity angle axis, magnetometer axis and derived 3D orientation angle axis. Each 3-axis components represent a direction in the hypersphere (see Fig. 5) (x-axis acceleration $\rightarrow$ x direction ; y-axis acceleration $\rightarrow$ y direction, z-axis

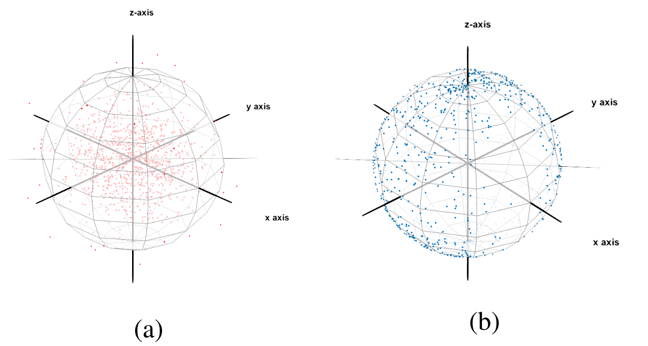


Fig. 5. Raw data (a) before and (b) after spherical normalization proposed

TABLE II
EXPERIMENTAL RESULT OF *RBF-SVM*

	Spherical Normalization	Standard Normalization
Accuracy	97.68	97.02
Precision	97.44	96.95
Recall	97.49	96.60

acceleration $\rightarrow$ z direction and so on on each sphere) and are standardized before being scaled (see Fig. 4). After *SN*, normalized raw data are collected, stored as described in Sec. II-C and given as input arguments to the classifier.

C. Classifier Training

The dataset used to build and evaluate the proposed Spherically normalized Support Vector Machine with RBF kernel is comprised of measurements from 4 able-bodied subjects aged 25 years old on average. The recorded data is split into two sub-datasets, 1) *subdataset 1* for training-validation, and 2) *subdataset 2* for test. *subdataset 1* which comprises recorded signals for 3 subjects (2 for training, 1 for validation), is used to build the classifier model, and select the hyper-parameters γ and C to avoid over-fitting during testing time. Classifier training is performed offline. The model is built using the *scikit-learn python* library [12]. Finally, the confusion matrices provides the measured performance of the classifier (see Fig. 6) .

IV. EXPERIMENTAL RESULT

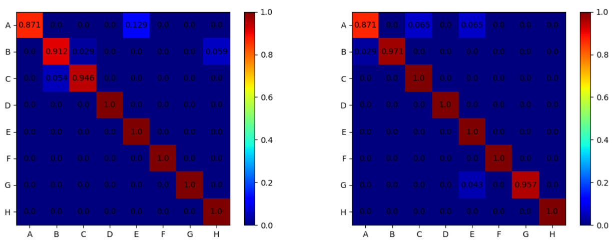
A. RBF-SVM With Spherical Normalization Accuracy

The trained *RBF-SVM* using Spherical normalization provides a better accuracy, precision and recall than standard normalization (see Table II). The confusion matrices are depicted in Fig.6. Even though raw data are used as input features, the performance of the algorithm in increased by 1.57% and by 2.46% the *k-NN* and *RBF-SVM* models described in [1], respectively. Similarly, The polynomial kernel based SVM model described in [3] is 0.89% better (see Table III). Thus, the preprocessing performed with the proposed *SN* method improves *HAR* performance with raw data as well as with filtered features, compared with the conventional normalization approach.

TABLE III

COMPARISON OF EXPERIMENTAL RESULT WITH LITERATURE

Method	Accuracy
K-nn [1] (raw data)	+1.15%
***RBF-SVM* [1] (Raw data)**	+3.46%
***RBF-SVM* [1] (Features)**	+2.13%
Multiclass SVM Polynomial Kernel[3](Features)	-0.89%



(a) (b)

Fig. 6. RBF SVM Normalized confusion matrix of test dataset using standard normalization (a) and *SN* (b). each letter of the alphabet corresponds to an activity (Tab.I). A: A1 - B: A2 - C: A3 - D: A4 - E: A5 - F: A6 - G: A7 - H: A8

B. Latency During Prediction Phase

The main purpose of this work is to design a novel embedded algorithm to perform accurate Human Activity Recognition in real-time. The performance of the proposed algorithm is not only quantified with the accuracy of the classifier, but also by the output latency. To prevent the users from any potential danger, a short latency is the key. The faster the information is transmitted accurately from the device to the user, the faster the risk will be minimized. In order to quantify this ouput latency, real-time testing with the trained classifier were conducted. It consisted of calculating the execution time set between the acquisition of the buffer of 1080 samples and the reception of the class predicted by the *RBF-SVM* during 10 minutes (23 latency values were collected). Finally, the average output latency is 107.1 ms with a standard deviation of 8.5 ms.

C. Measured system performance

Wearable IMU sensors require 31 mW on average to operate at 62 Hz which provides a 48-hour autonomy. The RPI3 which is powered with a 5V battery pack with a 20A capacity provides a 12.5-hour autonomy.

V. CONCLUSIONS

Using raw data from IMU sensors and derived 3D orientation angle pitch, roll and yaw, instead of frequency and time features commonly used for *HAR* reduces computational costs. Experimental results show that *SN* improves accuracy of *RBF-SVM*, when only a small dataset is available. Low computational costs guarantees a latency prediction feedback under 120 ms as well as moderate energy consumption on Raspberry PI 3. The proposed embedded system provides a mean to classify up to 8 ADLs classes using raw data, with accuracy similar to the state of the art. Future work will consist in detecting complex human activities and posture at risk to provide real-time feedback to the user, and prevent injuries. Improvement will be made to reduce energy consumption (sleep/wake-up mode) and latency as well. In addition, the database will be expanded to analyze classifier behavior with a larger dataset.

ACKNOWLEDGMENT

This preliminary study was funded by the Sentinel North Strategy.

REFERENCES

[1] F. Attal, S. Mohammed,M. Dedabrishvili, F. Chamroukhi, L. Oukhellou and Y. Amirat, "Physical Human Activity Recognition Using Wearable Sensors," *Sensors*, 15, pp. 3131431338, Dec. 2015.

[2] Oscar D. Lara and Miguel A. Labrador, "A Survey on Human Activity Recognition using Wearable Sensors," *IEEE Communications surveys and tutorials*, Vol. 15, no. 3, no. 3, pp. 1192-1209, 2013.

[3] Ku Nurhanim, I. Elamvazuthi and L.I. Izhar, T.Ganesan, "Classification of Human Activity based on Smartphone Inertial Sensor using Support Vector Machine," *IEEE 3rd International Symposium in Robotics and Manufacturing Automation (ROMA)*, Kuala Lumpur, 2017, pp. 1-5.

[4] K. Oh, H. S. Park and S. B. Cho, "A Mobile Context Sharing System Using Activity and Emotion Recognition with Bayesian Networks," *7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing*, Xian, Shaanxi, 2010, pp. 244-249.

[5] S.J. Preece, J.Y. Goulermas, L.P Kenney, D. Howard, K. Meijer and R. Crompton, "Activity identification using body-mounted sensorsA review of classification techniques," *Physiological Measurement*, 30. R1-33, 2009.

[6] Wenchao Jiang and Zhaozheng Yin, "Human Activity Recognition using Wearable Sensors by Deep Convolutional Neural Networks," *ACM international conference on Multimedia*, October 2015, pp. 1307-1310.

[7] Jian Cui and Bin Xu, Cost-effective activity recognition on mobile devices, *Proceedings of the 8th International Conference on Body Area Networks*, BodyNets 2013. pp. 90-96.

[8] C. L. Fall, F. Quevillon, A. Campeau-Lecours, S. Latour, M. Blouin, C. Gosselin, Fellow, B. Gosselin, "A multimodal adaptive wireless control interface for people with upper-body disabilities," 2017 *IEEE International Symposium on Circuits and Systems (ISCAS)*, Baltimore, MD, 2017, pp. 1-4.

[9] H. Gjoreski, M. Lustrek and M. Gams, "Accelerometer Placement for Posture Recognition and Fall Detection," 2011 *Seventh International Conference on Intelligent Environments*, Nottingham, 2011, pp. 47-54.

[10] F. Chamroukhi, S. Mohammed , D. Trabelsi , L. Oukhellou and Y. Amirat, "Joint segmentation of multivariate time series with hidden process regression for human activity recognition," *Neurocomputing*, vol. 120, pp. 633644, 2013.

[11] G. C. Cawley and N. L. C. Talbot, "Over-fitting in model selection and subsequent selection bias in performance evaluation," *Journal of Machine Learning Research*, vol. 11, pp. 2079-2107, 2010.

[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, D. Brucher, M. Perrot and E. Duchesnay, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825-2830, 2011.

[13] Chih-Chung Chang and Chih-Jen Lin, "LIBSVM: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, Vol. 2, no. 3, pp. 1-27, May 2011

[14] Vladimir N. Vapnik, "The Nature of Statistical Learning Theory," Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[15] V. Wan and S. Renals, "Speaker verification using sequence discriminant support vector machines," in *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 2, pp. 203-210, March 2005.

[16] Y. Petrov, "Ellipsoid Fit Method", *Matlab Exchange*, 2009.