IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

SPECIAL SECTION ON TRENDS, PERSPECTIVES AND PROSPECTS OF MACHINE LEARNING APPLIED TO BIOMEDICAL SYSTEMS IN INTERNET OF MEDICAL THINGS

Received April 8, 2018, accepted May 10, 2018, date of publication May 24, 2018, date of current version June 26, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2840327

Meta-Classifiers in Huntington's Disease Patients Classification, Using iPhone's Movement Sensors Placed at the Ankles

FRANCISCO DIEGO ACOSTA-ESCALANTE[1], ELÍAS BELTRÁN-NATURI[1], MARIE CATHERINE BOLL[2], JOSÉ ADÁN HERNÁNDEZ-NOLASCO[1], AND PABLO PANCARDO GARCÍA[1]

[1]División Académica de Informática y Sistemas, Universidad Juárez Autónoma de Tabasco, Cunduácan 86690, Mexico
[2]Laboratorio de Investigación Clínica, Instituto Nacional de Neurología y Neurocirugía MVS, Mexico City 14269, Mexico

Corresponding authors: Francisco Diego Acosta-Escalante (francisco.acosta@ujat.mx) and Elías Beltrán-Naturi (ebnaturi@gmail.com)

ABSTRACT Machine learning methods have been used to classify neurodegenerative diseases using gait data. Recent works with Huntington's disease (HD) patients have reported results up to 88.2% of correct classification based on a probabilistic modeling approach for gait assessment. The aim of this paper was to improve HD patients' classification results while reducing the number of sensor devices to capture gait data and identifying the related gait features. The proposed method is based on general assembles (Meta) classifier algorithms' approach, for the classification of HD affected gait versus healthy control (HC) subjects normal gait. The proposed methodology was tested on gait data recorded on HD patients and HC subjects using raw data from smart-phones movement sensors placed at both ankles. Best partial results of individual classifier algorithms are taken at each iteration of the meta-classifier, to predict the final result by averaging results and majority votes. Several instances of this combined approach were tested and validated. Obtained results confirm an improvement in accuracy, since 13 subjects out of a total of 14 were correctly classified. All seven Huntington's disease patients were correctly selected with *Logitboost & RandomForest* combination.

INDEX TERMS Motion measurement, patients monitoring, pattern recognition, data mining and performance analysis and evaluation.

I. INTRODUCTION

Huntington's disease is an autosomal dominant hereditary disorder characterized clinically by a triad of motor, cognitive and psychiatric symptoms. Motor disorders are usually the most frequent and the most notorious [1], [2]. They are progressive and worsen with the severity of the disease, they affect precision and speed of movement, cause loss of balance and normal gait mechanics, and lead to falls [3], [4]. Features of this disorders include excessive, spontaneous movements, irregularly timed, randomly distributed and abrupt. Disorders severity may vary from restlessness with mild, intermittent exaggeration of gesture and expression, fidgeting movements of the hands, unstable, dance-like gait to a continuous flow of disabling, violent movements (chorea) [5]. The chorea is usually the earliest abnormality of visible movement in adults; it is confused with other ailments when movements are isolated and incipient form [6]. Notwithstanding, chorea in HD does not appreciably affect the center of gravity during ambulation, and the consistency of gait profiles at heel strike shows that the ultimate target is achieved in each step despite random and frequent variability during the gait cycle [7]. Patients with more advanced disease have poorer balance and decreased gait mechanics, resulting in a state of nonambulation [8].

The onset average age of the disease varies from 30 to mid-50 years (fourth decade of life) [8]. After diagnosis, a progressive worsening of symptoms is observed in a period of 15 to 30 years until death [9]. There is no therapy or intervention available that demonstrates delayed onset or slows disease progression [10]. Disease mean duration to death is estimated between 15 and 20 years after the onset of chorea. Actual duration is probably much longer, based on biomarkers and clinical observations [11].

Medical evaluations to assess motor control include gait tests with long walkways that reveal longer posture, tendency to lean back on heels, decreased speed and stride length variability; alterations in gait are used as a predictor of disease progression. Technological tools have been developed to

evaluate such alterations as: image processing, walking bands with sensors, pressure sensors and portable sensors based on musculoskeletal models [12], [13].

Some recent works have focused to find evidence of alterations in gait patterns, and compared with medical scales to establish the Neurodegenerative (ND) disease progress such as Parkinson Disease (PD) [14], [15], Huntington's Disease (HD) [16], [17], Hereditary Ataxias (HA) [18], [19]. PD patients have been identified completely with recording speech signals using supervised learning with classification algorithms [20].

In recent years, technology has played an important role in supporting health-care specialists to perform neurodegenerative diseases diagnosis. This diagnosis is made by direct medical observation of how people walk. Gait analysis from movement sensors data has been extensively used with healthy people, however its use in HD patients is not widely documented. Movement sensors are implemented to quantify and evaluate gait characteristics as in [21]–[23]; Its use includes devices with different measuring capacities, from devices with single sensor [24] to smartphones with multiple sensor types [25], [26]. Recent research shows that sensors of iPhone smartphones, are "*sufficiently reliable and accurate to evaluate and identify the kinematic gait patterns*" [26], [27], studies related to gait evaluation and healthcare have demonstrated sensors-enabled iPhone capacity to accurately acquire quantified gait parameters with a sufficient level of consistency, specifically in ankle position, and in a comfortable, portable, and wearable manner [28]–[30].

Raw data coming directly from the sensors must be prepared for processing to improve classification results. This pre-processing of data has been done with one of the following approaches: extracting gait (kinematic) features related to a single step or a sequence of them such as length, frequency, speed, cadence, etc. [14], [17], [31], treating sensors data as a digital signal (flow of information from a source) and taking values as signal frequency, sampling frequency, maximum and minimum values, etc. [14], [32] and taking directly the stride cycles from raw data [33], [34], to establish when a subject belongs to a pathology using classifier algorithms; in this work we test the strength of the last approach, in improving the accuracy of the classification, by reducing human intervention and data preprocessing in data collection protocol.

The classification stage may include the use of one or more algorithms at a time. The use of **meta, multiple or assembly classifiers** allows the improvement of classification algorithms results by combining predictions of individual classifiers set in some way to classify new examples. A meta-classifier is a classifier that doesnt implement a classification algorithm on its own, but uses another classifier to do the actual work. The meta-classifier adds another processing step that is performed before the actual base-classifier sees the data. Examples of meta-classifiers are Logitboost, RandomCommittee and MultiBoostAB.

In this work we show that the use of LogitBoost as meta-classifier with RandomForest as base-classifier have better performance compared with other tree decision algorithms.

Basic characteristics of this algorithms are include in this section in order to clarify how well suited they are to classify stride cycle data.

A. CLASSIFICATION AND REGRESSION

The term "*regression*" commonly refers to a particular kind of parametric model (or process) for estimating a (numeric) target variable. Classification problem can be solved with a learner that can only produce estimates for a numeric target variable.

Given a class variable G that takes on values $1, \ldots, J$. The idea is to transform this class variable into J numeric "*indicator*" variables $G_1, \ldots, G_J$ to which the regression learner can be fit.

The indicator variable G_j for class j takes on value 1 whenever class j is present and value 0 everywhere else. A separate model is then fit to every indicator variable G_j using the regression learner. When classifying an unseen instance, predictions $u_1, \ldots, u_J$ are obtained from the numeric estimators fit to the class indicator variables, and the predicted class is: $j^* = \underset{j}{argmax}\, u_J$. This transformation process is used several times, for example when using model trees for classification. Transforming a classification task into a regression problem can be done using standard linear regression model. Linear regression fits a parameter vector β to a numeric target variable to form a model $f(x) = \beta^T x$. where x is the vector of attribute values for the instance (assuming a constant component in the input vector to accommodate the intercept). Linear logistic regression Pr (1), models the posterior class probabilities for the J classes via linear functions in x and ensures that they sum to one and remain in [0, 1].

$$Pr(G = j|X = x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}, \text{ where } F_j(x) = \beta_j^T \cdot x \tag{1}$$

Fitting a logistic regression model means estimating the parameter vectors β_j. The statistics standard procedure is to look for the maximum likelihood estimate, i.e. to choose the parameters that maximize the probability of the observed data points. For the logistic regression model, there are no closed-form solutions for these estimates. Instead, numeric optimization algorithms have to be used to approach the maximum likelihood solution iteratively and reach it in the limit. One such iterative method is the LogitBoost algorithm [35], shown in Figure 1, the LogitBoost algorithm has being proposed for fitting additive logistic regression models by maximum likelihood.

In each iteration, it fits a least-squares regressor to a weighted version of the input data with a transformed target variable. Here, y_{ij}^* (2) are the binary pseudo-response variables which indicate group membership of an observation

LogitBoost (J classes)

1. Start with weights $w_{ij} = 1/n,\ i = 1, \ldots, n,\ j = 1, \ldots, J,\ F_j(x) = 0$ and $p_j(x) = 1/J\ \ \forall j$
2. Repeat for $m = 1, \ldots, M$:
   (a) Repeat for $j = 1, \ldots, J$:
      i. Compute working responses and weights in the jth class
$$z_{ij} = \frac{y^*_{ij} - p_j(x_i)}{p_j(x_i)(1 - p_j(x_i))}$$
$$w_{ij} = p_j(x_i)(1 - p_j(x_i))$$
      ii. Fit the function $f_{mj}(x)$ by a weighted least-squares regression of z_{ij} to x_i with weights w_{ij}
   (b) Set $f_{mj}(x) \leftarrow \frac{J-1}{J}(f_{mj}(x) - \frac{1}{J}\sum_{k=1}^{J} f_{mk}(x)),\ \ F_j(x) \leftarrow F_j(x) + f_{mj}(x)$
   (c) Update $p_j(x) = \frac{e^{F_j(x)}}{\sum_{k=1}^{J} e^{F_k(x)}}$
3. Output the classifier $\underset{j}{\text{argmax}}\ F_j(x)$

FIGURE 1. LogitBoost meta-classifier algorithm [35].

like this:

$$y^*_{ij} = \begin{cases} 1 & \text{if } y_i = j \\ 0 & \text{if } y_i \neq j \end{cases}, \quad \begin{array}{l} \text{where } y_i \text{ is the observed class} \\ \text{for instance } x_i \end{array} \tag{2}$$

When constraining f_{mj} to be linear in x, linear logistic regression achieve if the algorithm is run until convergence. When further constraining f_{mj} to be a linear function of only the attribute that results in the lowest squared error, then the algorithm performs automatic attribute selection. By using cross-validation to determine the best number of LogitBoost iterations M, only those attributes that improve the performance on unseen instances are included.

B. CLASSIFICATION ALGORITHMS

Classification algorithms used in this study are based on decision tree learning method, which goal is to create a model that predicts the value of a target variable based on several input variables. In data mining a decision tree is a predictive model, where each interior node corresponds to one of the input variables; there are edges to children for each of the possible values of that input variable. Each leaf represents a value of the target variable, given the values of the input variables represented by the path from the root to the leaf. Decision tree learning uses the decision tree to go from observations about an input variable to conclusions about the variable's target value. Tree models, where the target variable can take a discrete set of values are called classification trees. Values are classified by navigating from the root of the tree down to a leaf, according to the outcome of the tests along the path [36].

C. RANDOM FORESTS ALGORITHM

Random forests or random decision forests [37], [38] are an ensemble learning method for classification, regression and other tasks. Random forest classification algorithm, as the name suggest, creates a forest with a number of trees. The idea behind the algorithm is that building a small decision-tree with few features is a computationally cheap process. A strong learner can be formed, combining many small, weak decision trees by averaging or taking the majority vote. This combination has to be made in such a way that the model produced by several learners into an ensemble, performs better than the original one. One way of combining learners is bootstrap aggregating or bagging, which shows each learner a randomly sampled subset of the training points, so that learners produces different models that can be sensibly averaged. In bagging, one samples training points with replacement from the full training set.

Random forest algorithm uses ensemble method bootstrap aggregated (or bagged) to build multiple decision trees by repeatedly re-sampling training data with replacement, and voting the trees for a consensus prediction [39]. Random decision forests correct for decision trees' habit of over-fitting to their training set. This is because trees that are grown very deep tend to learn highly irregular patterns: they over-fit their training sets, i.e. have low bias, but very high variance (bias-variance tradeoff). Random forests overcome this by averaging multiple deep decision trees, trained on different parts of the same training set, with the goal of reducing the variance. This comes at the expense of a small increase in the bias and some loss of interpretability, but generally greatly boosts the performance in the final model [40].

Random Forests algorithm in fig. 2 works in 2 steps (functions): in first step, a bootstrap sample from training set S is selected for each tree in the forest, where $S^{(i)}$ denotes the *ith* bootstrap. In second step a decision-tree is learned, using a modified decision-tree learning algorithm *RamdomizedTreeLearn*. The modified algorithm, instead of examining all possible feature-splits at each node of the tree, selects some subset of the features $f \subseteq F$, where F is the set of features. The node then splits on the best feature in f rather than F. In practice f is much, much smaller than F. Deciding on which feature to split is oftentimes the most computationally expensive aspect of decision tree learning. By narrowing the set of features, the learning of the tree is drastically sped up.

Precondition: A training set $S := (x_1, y_1), \ldots, (x_n, y_n)$, features F, and number of trees in forest B.

```
function RandomForest(S, F)
    H ← ∅
    for i ∈ 1, ..., B do
        S^(i) ← A bootstrap sample from S
        h_i ← RandomizedTreeLearn(S^(i), F)
        H ← H ∪ {h_i}
    end for
    return H
end function
function RandomizedTreeLearn(S, F)
    At each node:
        f ← very small subset of F
        Split on best feature in f
    return The learned tree
end function
```

FIGURE 2. Random forests algorithm.

TABLE 1. Results of classification algorithms for HD obtained in previous works.

Date	Ref.	Dataset	Type	Methods	HD	Accuracy
2011	[44]	Public	Pressure sensors	Classification algorithms	20	85.71%
2012	[43]	Public	Pressure sensors	Classification algorithms	15	50.00%
2014	[45]	Public	Pressure sensors	Meta-classifier algorithms	10	88.67%
2015	[34]	Ownership	Accelerometers	Meta-classifier algorithms	13	78.78 %
2016	[35]	Ownership	Accelerometer and Gyroscope	Classification algorithms	17	88.20%

In order to better understand why this algorithms combination is well suited to our study, let us review the base notions of machine learning. In supervised learning, we are given a data set and already know what our correct output should look like, having the idea that there is a relationship between the input and the output.

Supervised learning problems are categorized into ''regression'' and ''classification'' problems. In a regression problem, we are trying to predict results within a continuous output, meaning that we are trying to map input variables to some continuous function. In a classification problem, we are instead trying to predict results in a discrete output. In other words, we are trying to map input variables into discrete categories. Thus, while Logitboost approaches by means of a linear logistic regression, the most likely solution by selecting only those attributes that improve performance in unseen instances; Random forests performs the classification, predicting whether a subject is control or sick.

Published results from recent works on neurodegenerative disease patients classification, were obtained with pressure sensors data from public dataset ''PhysioNet'' [41] or they have captured and build their own private dataset.

Results based on the public dataset ''PhysioNet'', include those from Iram *et al.* [43] who found that *Quadratic Bayesian Normal* classifier had a lower error rate overall 65% (23/40) and their best results in classifying Huntington's Disease patients was 50% (5/10); Banie *et al.* [44] have obtained 86.957% of recognition of all classes with *Quadratic Bayesian Normal classifier* and for Huntington's Disease patients reached (85.714%) with *Decision Tree classifier*. A result of 88.674% of precision in classifying Huntington's Disease patients was obtained with combined meta-classifiers algorithms: *RandomSubSpace & Bagging*, *Bagging & PART* and *CVParameterSelection & Bagging* [44].

Recent results obtained with five movement sensors data from ownership dataset can be found in [33], where 81.04% of correctly classified data was obtained, recognizing the largest number of patients according to their pathology, result for HD patients was 78.78% using *Logitboost & RandomSubspace* meta-classifiers. Mannini *et al.* [35] used Hidden Markov Models (HMM) and Support Vector Machine (SVM) classifier, with a dataset of movement sensors placed at waist and shanks to reach 90.5% of recognition gait data for all groups, however, for HD patients was lower scored (88.2%).

We can observe in Table 1, how results obtained in previous studies have been continuously improving depending on dataset and treatment algorithms. In last work with pressure sensors (row 3) best results were obtained with meta-classifier algorithms. Results with classification algorithms (row 5) were better than those treated with meta-classifiers (row 4), when richest input information from movement sensors dataset were used.

Based on the idea of the sum of strengths obtained from the combination of individual algorithms, we wanted to verify the performance of classifiers combination when classifying HD patients with raw gait data dataset. We show that in Binary classification, the combination of *Logitboost & RandomForest* had outcomes high overall accuracy (94.4%) compared with those reviewed in the literature (88.674%) and the performance in correct classifying HD patients reached 96.6%.

This work represents the first step towards achieving the long-term goal of continuous real-time monitoring disease progression of HD patients. Aiming in that direction, the objective of this work is threefold, the improvement of recent results in HD patients classification versus Healthy Control (HC) subjects, while reducing the number of sensor devices to capture gait data and identifying the gait features involved in the process. To achieve this objective we used gait dataset obtained from Mexican patients and healthy people, using two wearable movement sensors devices attached to each of the subjects' ankles. Also, real-time continuous monitoring needs automatic data acquisition procedures with no human intervention and if possible using algorithms with good classification results over raw data.

II. SUBJECTS, MATERIALS AND DATA COLLECTION

Gait laboratory was installed at Instituto Nacional de Neurología y Neurocirugía ''Manuel Velasco Suárez'' (INNN-MVS) in a space of 20m long by 3m wide, this is large enough to capture gait characteristics.

Raw gait data was collected with the movement sensors of two Smartphones iPhone 5S (accelerometer and gyroscope three axes, rate 100Hz) [45] affixed to each subject ankle. Raw data were processed on a Dell computer with a Xenon Intel processor, 12 GB of ram memory, running Linux(Fedora 25) operating system. Processing time for raw gait dataset with two algorithms was between 24 and

36 hours, depending on the pair of algorithms used in each experiment.

There were 14 voluntary participants from INNN-MVS: seven with Huntington's Disease (HD) and seven Healthy Control (HC) subjects. Patients have clinical evaluations using the unified scale, (UHDRS, Unified Huntington's Disease Rating Scale). Controls are age and gender matched healthy subjects. Characteristics of participants are listed in Table 2.

TABLE 2. Characteristics of the study population.

Population		
Variable	Patients(n=7)	Controls(n=7)
Age (y, mean, ± sd)	48.8 ± 19.7	47.8 ± 11.4
Sex (male:female)	4 : 3	3 : 4
Weight (kg, mean± sd)	61.4 ± 9	62.4 ± 12.7
Height (cm, mean± sd)	162.4 ± 8.4	162.7 ± 8.0
Time evolution disease (y,mean±sd)	3.3 ± 2.2	Does not apply

Data was collected in a seven days period, during patients medical examination visit. INNN medical staff supervised patients while gathering gait data, in order to prevent accidents. Patients lack of availability and motor disorders (loss of balance, abnormal gait, precision and speed of movements) were taken into account in data collect planning.

This allowed to capture information from patients with different severity of the disease. Data collect protocol was designed to take this into account by placing movement sensors on the subjects' ankles at each walk (fig. 3). This arrangement is well suited to reduce gait alterations that may occur when walking, due to discomfort of wearing sensor devices and severity of motor alterations presented by patients. Each device recorded raw data sensor in individual files while walking. Physiological patient information (age, sex, health condition, etc.) was registered in separate file.

FIGURE 3. Data capture of patients with neuro-degenerative diseases in INNN-MVS.

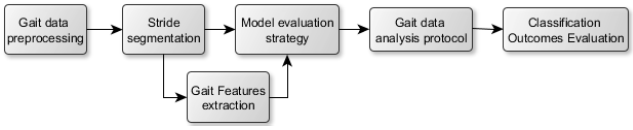

FIGURE 4. Gait data collection and classification schema.

III. METHODS

This research involves sampling, processing and evaluation of data to recognize patients with HD using meta-classifiers. After data were collected, we work with the information into six stages represented in Figure 4. (1) The preparation of the raw data to be used is described in the processing gait data (sec. III-A); (2) strides finding and extraction set out at stride segmentation (sec. III-B); (3) with processed data, the characteristics of each patient were extracted with the procedure of gait features extraction (sec. III-C); (4) the training configuration of the meta-classifiers and the evaluation of the performance of the models are described in model evaluation strategy (sec. III-D); (5) the performance of the meta-classifiers are compared in gait data analysis protocol (sec. III-E); and (6) comparison of classification errors and selection of the meta-classifier with better performance are explained in section with Classification outcomes evaluation (III-F).

A. GAIT DATA PREPROCESSING

Gait raw data was captured and stored on the devices (iPhones) and then it was extracted for processing. It needs to be prepared to identify stride information and features extraction, this preprocessing protocol was based on procedures implemented in [46]–[48]. Preprocessing protocol starts with sensor data calibration step to fix timing differences, then a zero normalization is made to eliminate signal constant effects on device accelerometer, followed by the calculation of Signal Acceleration Magnitude (*mag*) also known as Signal Vector Magnitude (SVM) to get an invariant measure to accelerometer rotation, the protocol ends with the elimination of noise from data and the definition of a window with 10-stride data.

1) SENSOR DATA CALIBRATION

Accelerometer data is captured at a variable sampling rate despite setting a capture frequency, this is due to the time

difference in which the capture call is made and the recording time on the device. Linear interpolation is used to calibrate the data to a sampling of constant time intervals. Linear interpolation equation is of the form,

$$a\prime = a_1 + \frac{(a_2 - a_1)(t\prime - t_1)}{t_2 - t_1} \tag{3}$$

where a_i the current acceleration, t_i is the capture time and $t\prime$ and $a\prime$ are the new sample obtained.

2) ZERO NORMALIZATION

When the device is immobilized the average acceleration values from the accelerometer should be equal to gravity force, however the data recorded from the devices are not stable over time, the acceleration is in a changing state. To eliminate these signal constant effects, zero normalization equation (4) was applied to acceleration values of the three axes,

$$A_i(t) = A_i(t) - \mu_i, i \in \{a_x, a_y, a_z\} \tag{4}$$

where a is acceleration in time and μ is the average acceleration.

3) ORIENTATION INDEPENDENCE

Sensor axes values depend on the position and the way in which the device is attached to the subject's body as they reflect its movements and gait alterations. To determine information relevant to strides, the Signal Acceleration Magnitude (mag_a) equation (5) that is invariant to the accelerometer orientation was used,

$$mag_a(i) = \sqrt{a_x(i)^2 + a_y(i)^2 + a_z(i)^2} \tag{5}$$

with *x, y and z* axis from each sample (i).

4) NOISE REMOVAL

Data smoothing refers to techniques for eliminating unwanted noise or behaviors in data, while outliers detection identifies data points that are significantly different from the rest of the data. The moving average is a common data smoothing technique that slides a window along the data, computing the mean of the points inside of each window. This can help to eliminate insignificant variations from one data point to the next. A moving average filter smooths data by replacing each data point with the average of the neighboring data points defined within the span. This process is equivalent to lowpass filter with the response of the smoothing given by the difference of equation 6, where $y_s(i)$ as the smoothed value for the *ith* data point,

$$y_s(i) = \frac{1}{2N+1}(y(i+N) + y(i+N1) + \cdot + y(iN)) \tag{6}$$

where N is the number of neighboring data points on either side of $y_s(i)$, and $2N + 1$ is the span [49].

B. STRIDE SEGMENTATION

Information corresponding to 10 strides (walking or gait cycles) was extracted in accordance with the minimum of 10 strides established as a sufficient measure to capture gait characteristics [50].

Gait cycles were identified with the acceleration signal from accelerometers, recognizing the changes in the signal (positive to negative), corresponding to the contact of the heel with the ground. It can be observed that when stride starts, the acceleration increases and when stride ends the acceleration decreases.

The peak to peak algorithm 1 (inspired from [51]) was used to find the maximum and minimum local value of the acceleration magnitude from signal acceleration magnitude (mag_a). Only those peaks that were greater than the standard deviation of values from each subject were considered. The *findpeaks* function finds all peaks greater than the minimum value (*minpeak*) and return the peaks value found (*pks*) and its index (*locs*). The same algorithm is used to calculate the minimum values, taking the same data with the inverted signs as input. To find the beginning of each stride, the minimum peaks below the standard deviation of all the values were considered. Each peak selected as the start of stride should be followed and preceded by a maximum peak, higher and lower than its standard deviation respectively.

Algorithm 1 Finding the Start of Each Stride

```
data ← mag_a
minpeak ← std(data)
pks, locs ← findpeaks(data, minpeak)
data_i ← −data
minpeak_i ← std(data_i)
pksi_i, locs_i ← findpeaks(data_i, minpeak_i)
j ← 0
for i ← 1, length(pks_i) do
    if pks_i(i) ≤ std(pks_i) and pks(i + 1) ≥ std(pks) and
  pks(i − 1) ≤ std(pks) then
        start(j) ← locs_i(i)
        j ← j + 1
    end if
end for
```

C. GAIT FEATURES EXTRACTION

Gait features were extracted with a computer-aided gait assessment tool (IGAIT), from gait data collected by an accelerometer. This tool provides an interactive and user-friendly platform to visualize acceleration data. A total of four types of gait features: Spatio-temporal, frequency domain, regularity, and symmetry can be derived with this tool [52], Twenty-eight gait-related characteristics were extracted: six space-time (equation 7), fifteen related to frequency (equations 8, 9 and 10) and seven of regularity and symmetry of step. The information was obtained in a 10-meter walking space and it was captured at a sampling rate of 100Hz. A maximum acceleration threshold of 0.4 was set to determine gait

characteristics. Autocorrelation coefficients are necessary to scale the regularity and symmetry of the walking, therefore, they are determined automatically or they are established manually by the maximum limits of acceleration for each one of the axes of the accelerometers (equation 11).

The Root Mean Square (RMS) is a measure of the magnitude of a data set. For acceleration measurements it indicates the intensity of the movement. The RMS values of the three acceleration directions (VT, AP and ML) are calculated respectively using Eq. 7

$$RMS_b = \sqrt{\sum_{i=1}^{N}(x_{di} - \bar{x}_d)^2/N} \tag{7}$$

Frequency signal analysis (spectral analysis) is used to calculate the magnitude of energy or power of movements corresponding to the frequency with which they were repeated. This analysis is important for identifying groups of people with specific movements and is independent of the movement sensors location. This includes the estimation of the Integral Power Spectral Density (IPSD) using Eq. 8 and 9. Where $0 \leq \omega \leq \pi$ is the angular frequency, x_i is the acceleration in either the VT, AP or ML axis, and N is the total number of acceleration samples. The frequency with the maximum PSD value is the main frequency. Other important feature in this analysis is the Cumulative Power Spectral Density CPSD, which is calculated with Eq. 10 for 50%, 75%, 90% and 100%. There are fifteen characteristics in the frequency domain, five for each direction.

$$PSD(e^{j\omega}) = \frac{1}{2\pi N}\left|\sum_{i=1}^{N} x_i e^{-j\omega i}\right|^2 \tag{8}$$

$$IPSD = \int_0^{\pi} PSD(\omega)d\omega \tag{9}$$

$$CPSD(\omega) = \int_0^{\omega} PSD(\omega)d\omega \tag{10}$$

$$f_c(t) = \frac{1}{N+|t|}\sum_{i=1}^{N-|t|} x_i x_{i+t} \tag{11}$$

D. MODEL EVALUATION STRATEGY

1) CLASSIFICATION ALGORITHMS

Metaclassifiers have demonstrated a good performance in features recognition of gait people with neurodegenerative diseases previous works (Table 1). A special feature for accuracy estimation of boosting meta-classifier algorithm is the availability to specify the number of iterations on execution. This feature allows the generation of multiple classifiers algorithm from one base classifier, each new model looking to avoid the mistakes of its predecessor in order to improve the classification result.

All possible combinations of the meta-classifiers with trees classifiers of Table 3 were tested with "Weka tool" to get the combinations with best results.

TABLE 3. Meta-classifiers and tree classifiers algorithms.

Meta-classifiers (Assemblies)	Tree Classifiers
Decorate (Decrt)	ADTree
Bagging (Bagg)	DesicionStump (DcStp)
LogitBoost(LogBst)	ExtraTree(XtrTr)
RandomSubSpace (RSS)	FT
Multiboost (MltBst)	HoeffdingTree (HfdTr)
Ordinalclassclassifier (RdlClsf)	J48
RotationForest (RotFst)	J48Consolidate (J48Cnst)
Dagging (Dagg)	J48Graft
Grading (Gradg)	LADTree
RealAdaBoost (RlaBst)	LMT
Multiclassifier (MulClr)	NBTree
RandomCommittee (RanCom)	RandomForest (RndFrt)
CVParameterSelection (CVPS)	RandomTree (RndTr)
MultiBoostAB (MBstAB	REPTree (RpTr)
RecedIncrementalLogitBoost (RILogbst)	SimpleCart (SmpCrt)

2) MODEL VALIDATION ACCURACY

One method of judging the quality of a particular model is by *residuals*. That means that the model is fit using all the data points, and the prediction for each data point is compared with its actual output. The absolute value of each error is taken and the mean of those values is computed to arrive at the mean absolute residual error. Models with lower values of this measure are deemed to be better. The problem with residual evaluations is that they do not give an indication of how well the learner will do, when it is asked to make new predictions for data that it has not already seen. One way to overcome this problem is do not use the entire data set when training a learner. Some of the data is removed before training begins. Then when training is done, the data that was removed can be used to test the performance of the learned model on "new" data. This is the basic idea for a whole class of model evaluation methods called cross validation.

In *k*-fold cross-validation, the training data is divided into a set of k disjoint subsets in the same size D_1, D_2, D_k. One of the k subsets is used for testing, whereas the other (*k-1/k*) subsets are used for training. This process is repeated k times (the folds) by using each of the *1/k* subsets as the test set, and the error is averaged over all possibilities. The k results from the folds can then be averaged to produce a single estimation. This has the advantage that all examples in the labeled data have an opportunity to be treated as test examples and each observation is used for validation exactly once. This does not substantially increase the variance of performance estimates because in k iterations all available instances are used for model evaluation. In a sense, the *k-fold* cross-validation procedure effectively virtualizes the training and validation or test sets. Values *k-folds* with 5 and 10 are particularly the most used [53]. When we have access to an unlimited number of examples we can estimate the performance, by choosing the model that provides the lowest "true" error rate on the entire population. In real applications, we only have access to a finite set of samples, usually

smaller than we would like, with a small and little participatory study population. Due to these shortcomings, the concept of cross-validation provides the greatest accuracy of implementation [54].

Leave-one-out (LOOCV) is a special type of k-fold cross-validation where k is set to the number of all data. That is, only one sample is left out at a time for the test set. In stratified cross-validation, the folds are stratified so that the class distribution of the tuples in each fold is approximately the same as that in the initial data [55]; for small sample sizes, LOOCV is much more accurate than other methods [56].

E. GAIT DATA ANALYSIS PROTOCOL

Classification accuracy is given in terms of several measures, the most commonly used for performance analysis are:

- **TP rate (TPR)**, also referred to as *sensitivity* is $TP/ActualPositives$, instances correctly classified as a given class.
- **FP rate (FPR)**, rate of false positives (instances falsely classified as a given class).
- **Precision** is $TP/PredictedPositives$. i.e. the proportion of instances that are truly of a class divided by the total instances classified as that class.
- **Recall** is the proportion of instances classified as a given class divided by the actual total in that class (equivalent to TP rate).
- **F-Measure** is a combined measure for precision and recall calculated as:

$$F - Measure = \frac{(2 * Precision * Recall)}{(Precision + Recall)} \tag{12}$$

- **Weighted average** that uses weights proportional to class frequencies of the data in average calculation. The weighted average of TP rate has been used to determine algorithm performance, because it stands for the algorithm success to correctly classify the member of each class.

In addition to accuracy, other measures commonly used to assess classifiers performance are:

- **ROC area measurement:** The standard ROC (Receiver Operating Characteristic) curve is a plot of TPR (true positive rate) against FPR (false positive rate) although alternatives have been recommended, and plotting Sensitivity vs Specificity is equivalent. Note that $Recall = TPR = Sensitivity$ and $FPR = (1 - Specificity)$.
- **Confusion matrix:** A confusion matrix is a technique for summarizing the performance of a classification algorithm, by definition a confusion matrix C is such that $C_{i,j}$ is equal to the number of observations known to be in group i but predicted to be in group j.
  - $C_{1,1}$ is True Positive (**TP**): the number of elements that were correctly predicted in the class they actually belong,
  - $C_{0,1}$ is False Positive (**FP**): the number of elements that were erroneously predicted to be members of one class but they actually belong to another one,

TABLE 4. Confusion matrix definition.

a	b	<—	Classified as
$C_{1,1}$	$C_{1,0}$	a =	Actual values for class a
$C_{0,1}$	$C_{0,0}$	b =	Actual values for class b

  - $C_{0,0}$ is True Negative (**TN**): the number of elements that were correctly predicted not belonging to a class and they actually do not belong to it and
  - $C_{1,0}$ is False Negative (**FN**): the number of elements that were erroneously predicted not being member of the class they actually belong.
- **Kappa statistic:** Kappa is often used as a measure of reliability between two human raters. Regardless, columns correspond to one ''rater'' while rows correspond to another ''rater''. In supervised machine learning, one ''rater'' reflects ground truth (the actual values of each instance to be classified), obtained from dataset labeled data, and the other ''rater'' is the machine learning classifier used to perform the classification. Kappa measures the agreement of prediction with the true class, kappa value of 1 means full agreement, while Kappa value of 0 is comparable to ''random guessing'' (similar to ROC value of 0.5). equation used is:

$$Kappa = \frac{(ObservedAccuracy - ExpectedAccuracy)}{(1 - ExpectedAccuracy)} \tag{13}$$

Kappa is an indication that a classifier is guessing randomly even if we have accuracy, precision, and recall near the unit. It is a metric that compares an Observed Accuracy with an Expected Accuracy (random chance). Considering random chance (agreement with a random classifier), generally means that it is less misleading than simply using accuracy as a metric (an Observed Accuracy of 80% is a lot less impressive with an Expected Accuracy of 75% versus an Expected Accuracy of 50%).

F. CLASSIFICATION OUTCOMES EVALUATION

There is a wealth of criteria by which the algorithms can be evaluated and compared. Error rates are used to assess how well the prediction results satisfied the real values distribution. The Mean Absolute Error (MAE) measures the average magnitude of the errors in a set of forecasts, without considering their direction. It measures accuracy for continuous variables. The Root Mean Squared Error (RMSE) measures the average magnitude of the error. MAE and RMSE can be used together to diagnose the variation in the errors in a set of forecasts. Let's denote the actual value as α and the estimated value using some algorithm as $\hat{\alpha}$. All error statistics compare true values to their estimates, but do it in a slightly different way. This means ''how far away'' are estimated values from the true value of α. We can see in equation 14, that sometimes square roots are used and sometimes absolute values - this is because when using square roots the

extreme values have more influence on the result. Indicators associated with classification error can be calculated with formulas 14 and 15.

The Relative Absolute Error (RAE) indicates how the model residuals (mean) are related to the values or the variability of the target function (mean deviation) itself, directly within a performance measure. The RAE should be less than 1 for any reasonable model, and preferably close to 0. The Root Relative Squared Error (RRSE) is relative to what would have been if a simple predictor had been used. Therefore, RRSE takes the total squared error and normalizes it by dividing the total squared error of the simple predictor, the square root of the relative squared error, reduces the error to the same dimensions as the predicted quantity ([53]).

$$3MAE = \frac{1}{N}\sum_{i=1}^{N}|\hat{\alpha}_i - \alpha_i| \quad RAE = \frac{\sum_{i=1}^{N}|\hat{\alpha}_i - \alpha_i|}{\sum_{i=1}^{N}|\bar{\alpha}_i - \alpha_i|} \tag{14}$$

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(\hat{\alpha}_i - \alpha_i)^2} \quad RRSE = \sqrt{\frac{\sum_{i=1}^{N}(\hat{\alpha}_i - \alpha_i)^2}{\sum_{i=1}^{N}(\bar{\alpha}_i - \alpha_i)^2}} \tag{15}$$

The RMSE will always be larger or equal to the MAE; the greater the difference between them, the greater the variance in the individual errors in the sample. If the RMSE = MAE, then all the errors are of the same magnitude. In RAE and RRSE those differences are divided by the variation of α so they have a scale from 0 to 1 and if we multiply this value by 100 we get similarity in 0-100 scale (i.e. percentage). The values of $\sum(\bar{\alpha}_i - \alpha_i)^2$ or $\sum|\bar{\alpha}_i - \alpha_i|$ tell us how much α differs from it's mean value. It is about how much α differs from itself (compared to variance). Because of that, the measures are named relative - they give us a result related to the scale of α.

Correlation is the measure of how much α and $\hat{\alpha}$ are related. It gives values between -1 and 1, where 0 is no relation, 1 is very strong, linear relation and -1 is an inverse linear relation (i.e. bigger values of α indicate smaller values of $\hat{\alpha}$, or vice versa).

IV. RESULTS

According to the proposed method, we conducted two experiments: the first one aims to improve the classification results based on raw data from iPhone movement sensors placed at the ankles, while the second one aims to identify the gait features involved in the process.

A. EXTRACTING STRIDES FROM HD GAIT DATASET

The **HD gait dataset** was built by extracting raw data from movement sensors (accelerometer and gyroscope) data from each file (with *CSVKit tool*). We plotted each data file and removed outliers information (using *R tool*), missing and meaningless borders data; afterward data calibration, removing of linear data and calculate the signal acceleration magnitude (*mag*) were carried out.

The moving average filter was applied on *mag* data to locate a 10-stride window in accordance to *Stride segmentation protocol* (section III-B), the obtained values are illustrated in figure 5.

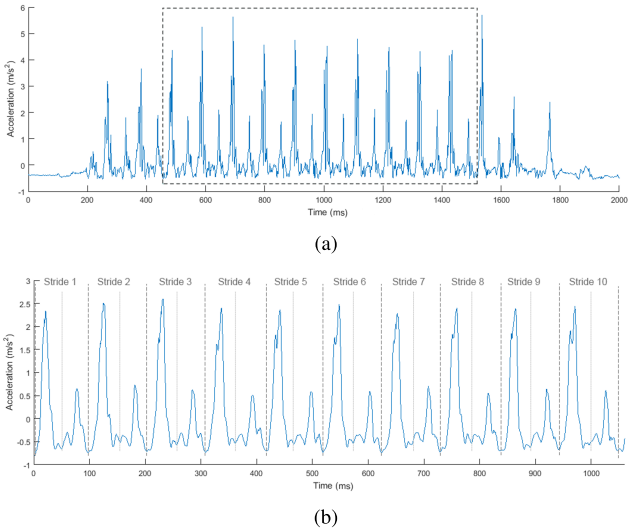

FIGURE 5. Stride segmentation protocol results. (a) Signal acceleration magnitude from raw data before preprocessing. (b) 10-stride window obtained after applying stride segmentation.

The 10-stride window from gait data is shown on (figure 5(b)).

B. DETERMINING GAIT FEATURES FROM HD GAIT DATASET

HD gait dataset was the input to iGAit software to acquire 28 gait features, with Sample Rate = 10 ms and Distance = 10 m. Since an irregular gait pattern is being analyzed, we test different threshold values to find the best one according to the gait event detection results, as suggested in [52]. Threshold parameter values were set to 0.4 for control subjects and to 0.3 for HD patients. Approximately 56 features were derived for each patient by taking data from the right (R) and left (L) sensors.

The 28 iGait resulting features were input to several attribute selection algorithms. CfsSubsetEval & BestFirst and CfsSubsetEval & GreedyStepwise algorithms selected the same 11 gait features, using a 14 fold cross-validation (stratified). Average and standard deviation values of these features for both classes are shown in Table 5.

iGait selected gait features were as follow:

- **RMS (Root Mean Square)** is the average acceleration along each three-dimensional axis: Anterior-Posterior (AP), Medium Lateral (ML), and Vertical (VER) during the walking period. All RMS were selected for both ankles, except the ML axis of the left ankle sensor.
- **Integral IPSD Frequency at 75% in VER, 90% in AP and 100% in ML** were taken from right ankle sensor (R) and frequency AP at 90% were selected from both sensors.

TABLE 5. Selected features including both, left (L) and right (R) sensors data.

Gait feature	Control		HD	
	Mean	Std Dev	Mean	Std Dev
RMS in AP (R)	0.44	0.14	0.32	0.16
RMS in AP (L)	0.44	0.15	0.32	0.78
RMS in ML (R)	0.70	0.20	0.42	0.97
RMS in VER (R)	0.56	0.11	0.43	0.97
RMS in VER (L)	0.62	0.11	0.45	0.11
IPSD at 75% in VER (R)	6.56	2.75	4.44	0.96
IPSD at 90% in AP (R)	21.43	2.78	23.88	7.35
IPSD at 90% in AP (L)	20.42	4.02	20.84	2.61
IPSD at 100% in ML (R)	25.98	4.59	28.10	4.13
Symmetry in VER (L)	0.64	0.22	0.34	0.23
Stride regularity in AP (L)	-0.36	0.06	0.21	0.19

- **Symmetry in VER**. Gait symmetry is estimated as the harmonic ratio for each of the three axis and used to measure gait symmetry [57]
- **Stride regularity in AP** is estimated for each axis as the normalized auto-covariance for a lag of exactly one estimated stride time [58]

C. ALGORITHMS PREDICTIVE ACCURACY ESTIMATION

Each meta-classifier was tested with a classification tree indicating the iterations necessary to achieve the best accuracy. The algorithms with the best results were: *Logitboost & Randomforest* (45 iterations), *randomcommittee & ExtraTree* (100 iterations) and *multiboost-AB & simplecart*, *multiboost-AB & J48* (100 iterations).

Model validation for these results was confirmed by running the LOOCV method, over the gait datasets; we ran the classifier 14 times stratified, by each fold 13 subjects used for training and 1 subject leaving out for test, so that each subject was used as test only once.

The classification Weighted Average is reported in Tables 6 and 7, the percentages indicate that 13 of 14 subjects were correctly recognized.

TABLE 6. Results of binary classification between sick and healthy subjects using raw data.

Algorithms	Weighted Avg.
Logitboost & RandomForest	94.4402%
RandomCommittee & ExtraTree	93.8688%
Multiboost-AB & SimpleCart	93.8633%
Multiboost-AB & J48	93.7479%

TABLE 7. Results of binary classification between sick and healthy subjects using 11 gait features.

Algorithms	Weighted Avg.
Logitboost & RandomForest	92.8571%
RandomCommittee & ExtraTree	85.7143%
Multiboost-AB & SimpleCart	78.5714%
Multiboost-AB & J48	92.8571%

Algorithms results in Table 6, were very closed, with a maximum difference of only 0.7%: *Logitboost & RandomForest* (94.44%), *RandomCommittee & ExtraTree* (93.86%), *MultiBoostAB & SimpleCart*(93.86%) and *MultiboostAB & J48* (93.74%). These results are based on the ability of the algorithms to correctly discriminate the members of each class. It is a sufficient criterion for algorithms selection, but not to justify which is the best performing.

Reported results in Tables 6 and 7 from both experiments were very close for *Logitboost & RandomForest* algorithm, because one subject of 14 (7.143%) was misclassified. *Multiboost-AB & J48* algorithm got a better performance with gait features than with raw data, because it kept up one misclassified subject. *RandomCommittee & ExtraTree* algorithm had lower performance than with raw data, with 2 misclassified subjects and *Multiboost-AB & SimpleCart* with 3 misclassified subjects.

D. BEST PERFORMING ALGORITHMS ANALYSIS

1) DETAILED ACCURACY BY CLASS

Performance Weka output for selected algorithms are shown in Tables 8 and 9. It can be noticed that accuracy is given in terms of several measures, and we take here the most commonly used for performance analysis. We can observe that the results obtained with gait features are not as smooth as those obtained with raw data, this is because the first are staggered by the number misclassified subjects. In the case of *Logitboost & RandomForest* algorithm for example, the value 1.0 in TP indicates that all elements of the HD class were correctly classified, while for the HC class the value 0.85 indicates that one subject was erroneously classified (there are no values in between).

TABLE 8. Detailed accuracy by class with raw data.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
Logitboost & RandomForest						
0.921	0.034	0.961	0.921	0.941	0.988	HC
0.966	0.079	0.930	0.966	0.948	0.988	HD
0.944	0.057	0.945	0.944	0.944	0.988	Avg.
RandomCommittee & ExtraTree						
0.924	0.048	0.947	0.924	0.935	0.985	HC
0.952	0.076	0.932	0.952	0.942	0.985	HD
0.939	0.062	0.939	0.939	0.939	0.985	Avg.
MultiBoost-AB & SimpleCart						
0.928	0.052	0.943	0.928	0.936	0.981	HC
0.948	0.072	0.935	0.948	0.941	0.978	HD
0.939	0.062	0.939	0.939	0.939	0.979	Avg.
MultiBoost-AB & J48						
0.927	0.053	0.942	0.927	0.934	0.976	HC
0.947	0.073	0.934	0.947	0.94	0.972	HD
0.937	0.063	0.938	0.937	0.937	0.974	Avg.

TP rate, also referred to as **sensitivity** is TP/actual positives instances correctly classified as a given class, results were over 0.921 for raw data and over 0.571 for gait

TABLE 9. Detailed accuracy by class with 11 gait features.

TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
Logitboost & RandomForest						
0.857	0.000	1.000	0.857	0.923	0.939	HC
1.000	0.143	0.875	1.000	0.933	0.939	HD
0.929	0.071	0.938	0.929	0.928	0.939	Avg.
RandomCommittee & ExtraTree						
0.857	0.143	0.857	0.857	0.857	0.898	HC
0.857	0.143	0.857	0.857	0.857	0.898	HD
0.857	0.143	0.857	0.857	0.857	0.898	Avg.
MultiBoost-AB & SimpleCart						
1.000	0.429	0.700	1.000	0.824	0.816	HC
0.571	0.000	1.000	0.571	0.727	0.816	HD
0.786	0.214	0.850	0.786	0.775	0.816	Avg.
MultiBoost-AB & J48						
0.857	0.000	1.000	0.857	0.923	0.929	HC
1.000	0.143	0.875	1.000	0.933	0.929	HD
0.929	0.071	0.938	0.929	0.928	0.929	Avg.

features, while for the rate of false positives (or **FP rate**, the instances falsely classified as a given class), results were under 0.079 for raw data and under 0.429 for gait features. It is important to notice that in the HD class the result was 1 in TP which means that all members of that class were correctly classified, while the FP result is 0.143 as one element of the HC class was erroneously classified as HD.

For **Precision** (TP/predicted positive. i.e. the proportion of instances that are truly of a class divided by the total instances classified as that class), all results are over 0.93 for raw data and over 0.70 for gait features. For **Recall** (the proportion of instances classified as a given class divided by the actual total in that class), it can be noticed that it is equivalent to TP rate.

The weighted average of TP rate has been used to determine algorithm performance, because it stands for the algorithm success to correctly classify the member of each class. All algorithm results have very close values for all measures. For instance, weighted average of TP rate, Precision, Recall and F-Measure have a maximum distance of 0.07 for all algorithms with raw data, which means how well-adapted this tree meta-classifiers are for raw HD gait dataset classification. However, the weighted average results for gait characteristics are more separate from each other, even if it is only one misclassified subject of difference.

Detailed accuracy values are ordered with higher values at top of Table 8. Based on weighted average, higher accuracy values are for *Logitboost & RandomForest* meta-classifier algorithm. Similar results are reported in Table 7 for *Logitboost & RandomForest* algorithm, but here *Multiboost-AB & J48* algorithm got the same result.

2) ROC AREA MEASUREMENT

The area under the curve (AUC) metric, measures the performance of a binary classification. In two-class (binary) classification, this will capture the probability threshold changes in

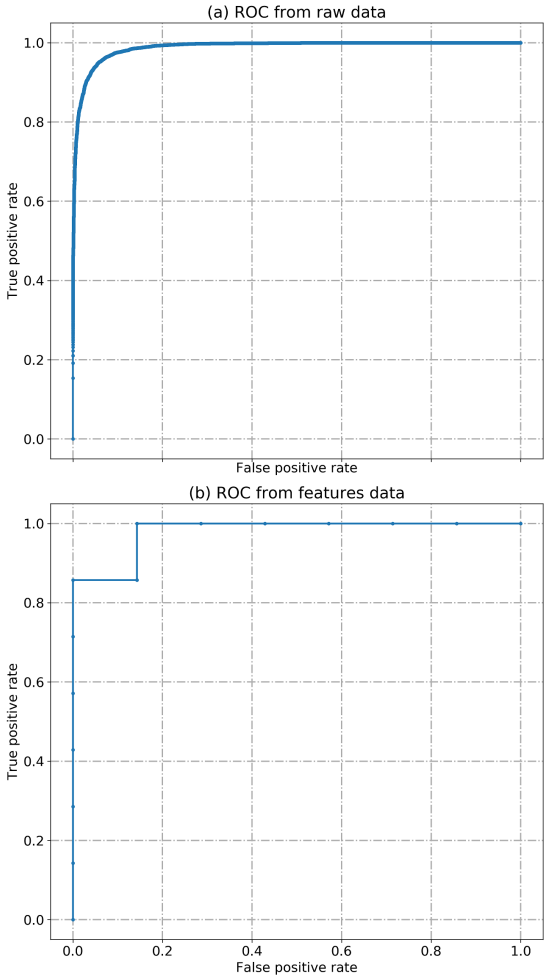

FIGURE 6. ROC graph for algorithm *Logitboost & RandomForest*.

a ROC curve. Normally the threshold for two classes is 0.5. An algorithm that is performing better than chance will lie above the diagonal (the chance line TPR = FPR). An "optimal" classifier will have ROC area values approaching 1. ROC graph values in tables 8 and 9, were plotted in fig. 6, as we can observe they are very close to 1. Each of the experiments have excellent prediction and there's no relevant difference between them for *Logitboost & RandomForest* algorithm.

3) CONFUSION MATRIX

In Weka Confusion Matrix, the row indicates the true class (real data), the column indicates the classifier output, with *Control(a)* and *Huntington(b)* representing class labels. Each entry gives the number of instances of *row* that were classified as *column*. In Table 10 there is a confusion matrix for each meta-classifier algorithm. For example, the Confusion Matrix for *Logitboost & RandomForest* algorithm can be read as: 8042 instances of Control were correctly classified as Control whereas 326 Huntington instances were erroneously classified as Control. Top-left and bottom-right of the matrix are showing instances the algorithms gets right. Bottom-left

TABLE 10. Meta-classifiers confusion matrix for raw data.

Meta-Classifier		HC	HD
Logitboost & RandomForest	HC	8042	686
	HD	326	9148
RandomCommittee & ExtraTree	HC	8068	660
	HD	456	9018
MultiBoost-AB & SimpleCart	HC	8102	626
	HD	491	8983
MultiBoost-AB & J48	HC	8090	638
	HD	500	8974

and top-right of the matrix are showing where the algorithm is confused. All correct classifications are on the top-left to bottom-right diagonal. Everything off that diagonal is some kind of incorrect classification. The total number of instances for this case is 18,200. Each model ranks a slightly higher number of instances for HD due to the impaired gait in patients. Correctly classified values are much higher than mistaken ones. The average sample value for every 10-strides is: for HC is 1,247.00 while the average for HD is 1,353.00. The HD class obtains the highest number of instances correctly classified for all algorithms. *Logitboost & RandomForest* scored the highest percentages, 94.44% of instances were correctly classified.

Likewise in Table 11 there is a confusion matrix for each meta-classifier algorithm. The total number of instances are 14, (7 for each class). Classification results are the same for *Logitboost & RandomForest* and *Multiboost-AB & J48* algorithms, with one instance of HC misclassified as HD. *RandomCommittee & ExtraTree* got two misclassified instances, one of each class and *MultiBoost-AB & SimpleCart* that misclassified 3 instances of HD as HC.

TABLE 11. Meta-classifiers confusion matrix for gait features data.

Meta-Classifier		HC	HD
Logitboost & RandomForest	HC	6	1
	HD	0	7
RandomCommittee & ExtraTree	HC	6	1
	HD	1	6
MultiBoost-AB & SimpleCart	HC	7	0
	HD	3	4
MultiBoost-AB & J48	HC	6	1
	HD	0	7

4) KAPPA STATISTICS

Observed Accuracy is simply the number of instances that were classified correctly throughout the entire confusion matrix; i.e. Tables 10 and 11 present instances that were labeled as Control and predicted as Huntington by the classifier and the contrariwise. The last column of Tables 12 and 13 contain the observed accuracy by meta-classifier based on outcomes of Tables 10 and 11 (number of instances that the classifier agreed with the dataset, divided by the total number of instances).

TABLE 12. Weka kappa statistic and accuracy for raw data.

Meta-classifier & Classifier	Kappa statistic	Accuracy
LogitBoost & RandomForest	0.8884	94.44%
RandomCommittee & ExtraTree	0.8771	93.86%
Multiboost-AB & SimpleCart	0.8770	93.86%
Multiboost-AB & J48	0.8747	93.74%

TABLE 13. Weka kappa statistic and accuracy for gait features.

Meta-classifier & Classifier	Kappa statistic	Accuracy
LogitBoost & RandomForest	0.8571	92.86%
RandomCommittee & ExtraTree	0.7143	85.71%
Multiboost-AB & SimpleCart	0.5714	78.57%
Multiboost-AB & J48	0.8571	92.86%

The Expected Accuracy is directly related to the number of instances of each class (Control and Huntington), along with the number of instances that the machine learning classifier agreed with the dataset. The Expected Accuracy is calculated by multiply the marginal frequency of Control for one ''rater'' by the marginal frequency of Control for the second ''rater'', and divided by the total number of instances; the marginal frequency for a class by a ''rater'' is just the sum of all instances the ''rater'' indicated were that class. For confusion matrix (Table 10), *Logitboost & RandomForest* classifier reconized 8,368 instances as Control of 8,728 instances labeled as Control in dataset. Marginal values were $(8368 * 8728/18202)$ 4,012.5208 for Control and $(9834 * 9474/18202)$ 5,118.5208 for Huntington; the Expected Accuracy is 0.50 $((4012.52081 + 5118.5208)/18202)$. Expected Accuracy turned out to be 50%, as will always be the case when either ''rater'' classifies each class with the same frequency in a binary classification (both Control and Huntington classes contained a very close number of instances according to the confusion matrix). For confusion matrix in Table 11 Expected Accuracy is 0.5 also, because the marginal values for HC and HD are 3 and 4 respectively and the total number of instances is 14.

The kappa statistic is calculated with the equation 13 using the Observed Accuracy and the Expected Accuracy calculated previously. Weka's results for Kappa measure of all meta-classifiers are shown in Tables 12 and 13. We can observe that the higher agreement of prediction with the true class is for *LogitBoost & RandomForest*, as well as the highest accuracy. However, we can point out that in the case of gait features in Table 13, *Multiboost-AB & J48* algorithm also obtains the highest values.

E. EVALUATION MEASURES (MAE, RMSE, RAE, RRSE)

Error measures are the indicators of how well prediction results fits real values distribution, the greater the difference between them, the greater the variance in the individual errors. Resulting values obtained with formulas 14 and 15

show that RMSE values are always larger than MAE values, the greatest difference between RMSE and MAE is for J48 algorithm (0.184) with raw data and for SimpleCart algorithm (0.234) with gait features. The smallest differences between them were for ExtraTree algorithm in both cases, 0.39 with raw data and 0.49 with gait features.

Table 14 show that values for *LogitBoost & RandomForest* algorithm are: MAE = 0.05 for Mean Absolute Error, that measures accuracy for continuous variables and RMSE = 0.22 for Root Mean Squared Error, that measures the average magnitude of the error. The difference between them gives a variation in prediction of 0.16, this is the lowest value of all classifiers.

TABLE 14. Score in MAE, RMSE, RAE and RRSE for raw data.

Meta-classifier & Classifier	MAE	RMSE	RAE %	RRSE %	MCC
LogitBoost & RandomForest	0.056	0.223	11.37	44.63	0.886
RandomCommittee & ExtraTree	0.252	0.291	50.60	58.33	0.877
Multiboost-AB & SimpleCart	0.061	0.244	12.31	48.83	0.877
Multiboost-AB & J48	0.063	0.247	12.57	49.57	0.875

[1] MAE (Mean Absolute Error), RMSE (Root Mean Squared Error), RAE (Relative Absolute Error), RRSE (Root Relative Squared Error)

Weka output for Matthews Correlation Coefficient (MCC) that are closed to 1 indicates strong relation between actual values in dataset and predicted ones. MCC value is 0.88 for *LogitBoost & RandomForest*, in Table 14, but all other values are very close to this one, meaning a strong relation between the actual values in dataset and those predicted by the algorithm.

Based on this indicators we can say that the best performing algorithm is *LogitBoost & RandomForest* which had the bigger correlation and smaller error estimates. We can see in Table 14 that the results obtained for accuracy, place this algorithm as the best to predict Huntington's disease on people using movement sensor devices at the ankles.

V. DISCUSSION

We propose a method to address two important facts in HD gait classification: the first is that results from previous studies have shown that to date there is no single algorithm that provides the best results with any data set and the second is that classification algorithms results, obtained with walking data from patients with Huntington's disease, can be improved. Proposed method strength is shown in obtained results, they are better than those in recent publications on a 10-point scale.

We worked with meta-classifiers as in [44], but we added tree classifiers as in [43], this combination got out better results in accuracy than related works. Meta-classifiers and

TABLE 15. Score in MAE, RMSE, RAE and RRSE for gait features.

Meta-classifier & Classifier	MAE	RMSE	RAE %	RRSE %	MCC
LogitBoost & RandomForest	0.074	0.267	13.85	50.14	0.866
RandomCommittee & ExtraTree	0.329	0.378	61.74	70.86	0.714
Multiboost-AB & SimpleCart	0.233	0.467	43.60	87.15	0.632
Multiboost-AB & J48	0.071	0.267	13.39	50.11	0.929

accelerometer data from several devices in [33] provided an accuracy of 78.78%, however, we used additional gyroscopes data that provided a greater impact on the classification results, in a similar way than in [34]. Using boosting algorithms (meta-classifier & classifier) allowed us to significantly improve the results: from 50% using classic algorithms (the lowest) and 88.674% using meta-classifiers (the highest reported) to 94.5% combining prior techniques;

The major finding of our proposal is that the use of meta-classifier with tree classifiers reached a global classification Precision of 94.5% with raw data and 93.8% with gait features for all groups and a selection of 96.6% with raw data and 100% of HD patients. *Bagging, RandomSubspace, and LogitBoost* meta-classifiers provided good performance with pressure sensors in [44], however, when single classifier algorithms were implemented with data from movement sensors, the performance decreased [33], for this reason, we decided to prove the meta-classifiers strength with tree classifiers. This decision let us obtain a higher accuracy. To our knowledge, the best classification performance for HD obtained until now. Tree classifiers have been implemented in recognition of other neuro-degeneratives diseases with excellent results, such as Parkinson [20], [59]. This allows confirming the initial hypothesis in which a combination of two algorithms produces better results than those obtained with the individual algorithms.

The second finding was that just two movement sensors (accelerometer and gyroscope), one per ankle, were enough to obtain better results than those in [35], where three devices were used. The smaller number of devices makes our proposal less resource consuming and less obtrusive than others recently proposed [34], who have had the best classification result to date. It is important to remark that the reduction in the number of devices to capture gait movement reduces the effect that they may have in patients walking gait and enable the possibility of continuous patient monitoring to oversee disease progression.

The third finding was that 11 gait features allows a good HD gait patients recognition. This fact is important because even when raw data analysis is well suited for continuous monitoring, it does not allow to know further details as gait patterns or subjects identification. Obtained classification results with gait features were very close to those obtained

with raw data. Results difference in Precision for *LogitBoost & RandomForest* was of 0.007.

The use of several types of movement sensors captured more significant data during the walking of the subjects than when only one type of sensor was used, which led to a better characterization of the walking pattern; Sánchez-Delacruz *et al.* [34] used five accelerometers in patients' body but their approach offered a lower accuracy than ours.

We assume that the classification results for HD subjects were higher than healthy ones, given that these patients have exaggerated and uncontrolled movements, that were reflected in captured data, helping the classifiers to a better separation.

VI. CONCLUSIONS

Based on previous results, we assumed that meta-classifiers are well placed to improve the results so far published in the classification of patients with HD disease, while reducing the number of devices used to capture the movements of gait. To confirm this assumption we used data collected with accelerometer and gyroscope values of two movement sensors placed in both ankles of HD patients and HC subjects. With this data we build the HD gait dataset composed of raw data, and another one with gait features extracted from raw data.

We found out that 11 gait features using 2 attribute selection algorithms were the most representative of the gait patterns of the subjects.

We tested 10 meta-classifiers algorithms with their corresponding classifiers, over HD gait dataset. Algorithms results in Table 8 and 9 were higher to 90% of correctly predicted classes and with detailed accuracy values very narrowly closest. This confirms the assumption that meta-classifier have better performance than single classifiers, even when they have been modified to suit a specific disease.

Best performing meta-classifier algorithm, based on the number of HD patients that were correctly classified was *Logitboost & RandomForest* with 96.6% with raw data and 100% with gait features.

Obtained results encourage future work in continuous monitoring, since binary classification (comparison of two models) may serve, for example, to better monitor the progression of the disease by detecting an aggravation that is reflected in changes in the way the study subjects walk. This line of study needs a monitoring system with real time data transfer to be designed.

Another line of study that promotes the results of the classification is the early detection of the disease. The study of algorithms allows to discover the patterns of gait in several neurodegenerative diseases, and therefore it can be very useful in the differential diagnosis for example of a Huntington's disease, versus a spinocerebellar ataxia, clinically similar. The results of this study could be exploited for the monitoring of patients undergoing some type of rehabilitation or other therapeutic intervention. It should be noted that these results could be favored by very ill patients because at the advanced stage of the disease, more alterations in gait are presented, hence future works have to respond to the question of what is going to be the meta-classifiers performance when predicting HD subjects at different stages of the disease?

ACKNOWLEDGMENT

We would like to thank Dr. Miguel Angel Celis, Director of the INNN-MVS for his support in carrying out the measurements for this project. We are very grateful to the staff and employees of INNN-MVS for their collaboration. We are also grateful to all the patients who participated and the accompanying family members. We would also like to express our gratitude to Universidad Juárez Autónoma de Tabasco for supporting the academic resources needed for this research.

REFERENCES

[1] M. Singh, M. Singh, and S. Paramjeet, "Neuro-degenerative disease diagnosis using human gait: A review," *Comput. Sci. Electron. J.*, vol. 7, no. 1, pp. 16–20, 2013. [Online]. Available: http://tudr.thapar.edu:8080/jspui/bitstream/10266/2741/4/2741.pdf

[2] A. Delval and P. Krystkowiak, "Locomotion et maladie de Huntington," *Revue Neurol.*, vol. 166, no. 2, pp. 213–220, 2010.

[3] R. Du, "Impact of caryolanemagnolol on gait and functional mobility on individuals with Huntington's disease," *Tropical J. Pharmaceutical Res.*, vol. 14, no. 9, pp. 1713–1717, 2015.

[4] T. Cruickshank, A. Reyes, L. Peñailillo, J. Thompson, and M. Ziman, "Factors that contribute to balance and mobility impairments in individuals with Huntington's disease," *Basal Ganglia*, vol. 4, no. 2, pp. 67–70, 2014.

[5] A. Barbeau, R. Duvoisin, F. Gerstenbrand, J. P. Lakke, C. D. Marsden, and G. Stern, "Classification of extrapyramidal disorders. proposal for an international classification and glossary of terms," *J. Neurol. Sci.*, vol. 51, no. 2, pp. 311–327, 1981.

[6] D. Justo *et al.*, "Is non-recognition of choreic movements in Huntington disease always pathological?" *Neuropsychologia*, vol. 51, no. 4, pp. 748–759, 2013.

[7] N. C. Reynolds, J. B. Myklebust, T. E. Prieto, and B. M. Myklebust, "Analysis of gait abnormalities in Huntington disease," *Arch. Phys. Med. Rehabil.*, vol. 80, no. 1, pp. 59–65, 1999.

[8] A. J. McGarry, K. Biglan, and F. Marshall, "Huntington disease," in *Rosenberg's Molecular and Genetic Basis of Neurological and Psychiatric Disease*. Amsterdam, The Netherlands: Elsevier, 2015, ch. 75. [Online]. Available: https://www.sciencedirect.com/science/article/pii/B9780124105294000759

[9] M. P. Parsons and L. A. Raymond, "Huntington disease," in *Neurobiology of Brain Disorders*. Amsterdam, The Netherlands: Elsevier, 2015, pp. 303–320.

[10] Y. M. Bordelon, "Huntington disease," *Neurologic Clin.*, vol. 31, no. 4, pp. 1085–1094, Nov. 2013.

[11] L. Seeberger, "Huntington's disease," in *Encyclopedia of Movement Disorders*. Amsterdam, The Netherlands: Springer, 2003, pp. 1–14.

[12] A. I. A. Mendoza, T. J. B. Santamaria, V. G. Urrego, J. P. R. Restrepo, and M. C. Z. García, "Marcha: Descripción, métodos, herramientas de evaluación y parámetros de normalidad reportados en la literatura," *CES Movimiento y Salud*, vol. 1, no. 1, pp. 29–43, 2013.

[13] C. Cifuentes and F. Mart "inez, and E. Romero, "Análisis teórico y computacional de la marcha normal y patológica: Una revisión," *Revista Med*, vol. 18, no. 2, pp. 182–196, 2010.

[14] J. Klucken *et al.*, "Unbiased and mobile gait analysis detects motor impairment in Parkinson's disease," *PLoS ONE*, vol. 8, no. 2, p. e56956, Feb. 2013. [Online]. Available: http://dx.plos.org/10.1371/journal.pone.0056956

[15] D. Trojaniello, A. Ravaschio, J. M. Hausdorff, and A. Cereatti, "Comparative assessment of different methods for the estimation of gait temporal parameters using a single inertial sensor: Application to elderly, post-stroke, Parkinson's disease and Huntington's disease subjects," *Gait Posture*, vol. 42, no. 3, pp. 310–316, 2015.

[16] A. Dalton, H. Khalil, M. Busse, A. Rosser, R. van Deursen, and G. ÓLaighin, "Analysis of gait and balance through a single triaxial accelerometer in presymptomatic and symptomatic Huntington's disease," *Gait Posture*, vol. 37, no. 1, pp. 49–54, 2013.

[17] D. Trojaniello, A. Cereatti, A. Ravaschio, M. Bandettini, and U. D. Croce, "Assessment of gait direction changes during straight-ahead walking in healthy elderly and Huntington Disease patients using a shank worn MIMU," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2014, pp. 2508–2511.

[18] S. Shirai, I. Yabe, M. Matsushima, Y. M. Ito, M. Yoneyama, and H. Sasaki, "Quantitative evaluation of gait ataxia by accelerometers," *J. Neurol. Sci.*, vol. 358, nos. 1–2, pp. 253–258, 2015.

[19] A. Matsushima *et al.*, "Clinical assessment of standing and gait in ataxic patients using a triaxial accelerometer," *Cerebellum Ataxias*, vol. 2, p. 9, Aug. 2015.

[20] A. K. Tiwari, "Machine learning based approaches for prediction of Parkinson's disease," *Mach. Learn. Appl., Int. J.*, vol. 3, no. 2, pp. 33–39, Jun. 2016.

[21] J. Rueterbories, E. G. Spaich, B. Larsen, and O. K. Andersen, "Methods for gait event detection and analysis in ambulatory systems," *Med. Eng. Phys.*, vol. 32, no. 6, pp. 545–552, 2010.

[22] R. Martinez-Mendez, M. Sekine, and T. Tamura, "Detection of anticipatory postural adjustments prior to gait initiation using inertial wearable sensors," *J. Neuroeng. Rehabil.*, vol. 8, no. 1, p. 17, 2011.

[23] M. Iosa, P. Picerno, S. Paolucci, and G. Morone, "Wearable inertial sensors for human movement analysis," *Expert Rev. Med. devices*, vol. 13, no. 7, pp. 641–659, 2016. [Online]. Available: http://www.ncbi.nlm.nih.gov/pubmed/27309490

[24] T. Shirakawa, M. Kamiura, A. Takagi, and H. Sato, "Analysis for the gait patterns of healthy subjects during March," *Procedia Comput. Sci.*, vol. 24, pp. 167–174, Dec. 2013. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050913011824?via%3Dihub

[25] Y. Watanabe, "Toward application of immunity-based model to gait recognition using smart phone sensors: A study of various walking states," *Procedia Comput. Sci.*, vol. 60, pp. 1856–1864, Sep. 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050915024230

[26] B. Sun, Y. Wang, and J. Banda, "Gait characteristic analysis and identification based on the iPhone's accelerometer and gyrometer," *Sensors*, vol. 14, no. 9, pp. 17037–17054, 2014. [Online]. Available: http://www.mdpi.com/1424-8220/14/9/17037

[27] A. Galán-Mercant, F. J. Barón-López, M. T. Labajos-Manzanares, and A. I. Cuesta-Vargas, "Reliability and criterion-related validity with a smartphone used in timed-up-and-go test," *Biomed. Eng. OnLine*, vol. 13, p. 156, Dec. 2014. [Online]. Available: https://doi.org/10.1186/1475-925X-13-156

[28] Y. Fujiki, P. Tsiamyrtzis, and I. Pavlidis, "Making sense of accelerometer measurements in pervasive physical activity applications," in *Proc. CHI Extended Abstracts Hum. Factors Comput. Syst. (CHI EA)*, New York, NY, USA: ACM, 2009, pp. 3425–3430. [Online]. Available: http://doi.acm.org/10.1145/1520340.1520497

[29] R. LeMoyne, T. Mastroianni, M. Cozza, C. Coroian, and W. Grundfest, "Implementation of an iPhone as a wireless accelerometer for quantifying gait characteristics," in *Proc. Annu. Int. Conf. IEEE Eng. Med. Biol.*, Aug./Sep. 2010, pp. 3847–3851.

[30] R. LeMoyne, T. Mastroianni, A. Hessel, and K. Nishikawa, "Ankle rehabilitation system with feedback from a smartphone wireless gyroscope platform and machine learning classification," in *Proc. IEEE 14th Int. Conf. Mach. Learn. Appl. (ICMLA)*, Dec. 2015, pp. 406–409.

[31] A. Mannini, D. Trojaniello, U. D. Croce, and A. M. Sabatini, "Hidden Markov model-based strategy for gait segmentation using inertial sensors: Application to elderly, hemiparetic patients and Huntington's disease patients," in *Proc. 37th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Aug. 2015, pp. 5179–5182. [Online]. Available: http://ieeexplore.ieee.org/document/7319558/

[32] D. Joshi, A. Khajuria, and P. Joshi, "An automatic non-invasive method for Parkinson's disease classification," *Comput. Methods Programs Biomed.*, vol. 145, pp. 135–145, Jul. 2017.

[33] E. Sánchez-Delacruz, F. Acosta-Escalante, C. Boll-Woehrlen, F. J. Álvarez-Rodríguez, A. Hernández-Nolasco, M. A. W. y Pablo Pancardo, "Categorización de enfermedades neurodegenerativas a partir de biomarcadores de la marcha," *Komput. Sapiens*, vol. 2, pp. 17–20, Aug. 2015.

[34] A. Mannini, D. Trojaniello, A. Cereatti, and A. M. Sabatini, "A machine learning framework for gait classification using inertial sensors: Application to elderly, post-stroke and Huntington's disease patients," *Sensors*, vol. 16, no. 1, p. 134, Jan. 2016.

[35] J. Friedman, T. Hastie, and R. Tibshirani, "Additive logistic regression: A statistical view of boosting (with discussion and a rejoinder by the authors)," *Ann. Statist.*, vol. 28, no. 2, pp. 337–407, 2000.

[36] L. Rokach and O. Maimon, *Data Mining With Decision Trees: Theory and Applications*. Singapore: World Scientific, 2014.

[37] T. K. Ho, "Random decision forests," in *Proc. 3rd Int. Conf. Document Anal. Recognit.*, vol. 1. Aug. 1995, pp. 278–282.

[38] T. K. Ho, "The random subspace method for constructing decision forests," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 8, pp. 832–844, Aug. 1998.

[39] L. Breiman, "Bagging predictors," *Mach. Learn.*, vol. 24, no. 2, pp. 123–140, 1996.

[40] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning* (Springer Series in Statistics), vol. 1. New York, NY, USA: Kluwer, 2001. [Online]. Available: http://www.mdpi.com/1424-8220/14/9/17037

[41] *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. Accessed: Nov. 2017. [Online]. Available: https://www.springer.com/gb/book/9780387848570

[42] S. Iram, D. Al-Jumeily, P. Fergus, M. Randles, and A. Hussain, "Computational data analysis for movement signals based on statistical pattern recognition techniques for neurodegenerative diseases," in *Proc. 13th Annu. Post Graduate Symp. Converg. Telecommun., Netw. Broadcast.*, 2012, pp. 1–6.

[43] M. Banaie, M. Pooyan, and M. Mikaili, "Introduction and application of an automatic gait recognition method to diagnose movement disorders that arose of similar causes," *Expert Syst. Appl.*, vol. 38, no. 6, pp. 7359–7363, 2011.

[44] E. Sánchez-Delacruz, F. Acosta-Escalante, M. A. Wister, J. A. Hernández-Nolasco, P. Pancardo, J. J. Méndez-Castillo, "Gait recognition in the classification of neurodegenerative diseases," in *Ubiquitous Computing and Ambient Intelligence. Personalisation and User Adapted Services*, R. Hervás, S. Lee, C. Nugent, and J. Bravo, Eds. Cham, Switzerland: Springer, 2014, pp. 128–135. [Online]. Available: https://link.springer.com/chapter/10.1007/978-3-319-13102-3_23

[45] Apple. (2014). *iPhone 5s—Technical Specifications*. [Online]. Available: https://support.apple.com/kb/SP685?locale=en_GB and https://www.apple.com/uk/iphone-5s/specs/

[46] M. Muaaz and R. Mayrhofer, "Orientation independent cell phone based gait authentication," in *Proc. 12th Int. Conf. Adv. Mobile Comput. Multimedia (MoMM)*, 2014, pp. 161–164.

[47] V. Chandel, A. D. Choudhury, A. Ghose, and C. Bhaumik, "AcTrak—Unobtrusive activity detection and step counting using smartphones," in *Mobile and Ubiquitous Systems: Computing, Networking, and Services* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 131. Cham, Switzerland: Springer, 2014, pp. 447–459. https://link.springer.com/chapter/10.1007%2F978-3-319-11569-6_35

[48] M. Muaaz and R. Mayrhofer, "Cross pocket gait authentication using mobile phone based accelerometer sensor," in *Computer Aided Systems Theory—EUROCAST* (Lecture Notes in Computer Science), vol. 9520. Cham, Switzerland: Springer, Feb. 2015, pp. 731–738.

[49] *Cross Pocket Gait Authentication Using Mobile Phone Based Accelerometer Sensor*. Accessed: Jan. 2018. [Online]. Available: https://link.springer.com/chapter/10.1007%2F978-3-319-27340-2_90

[50] J. H. Hollman, K. B. Childs, M. L. McNeil, A. C. Mueller, C. M. Quilter, and J. W. Youdas, "Number of strides required for reliable measurements of pace, rhythm and variability parameters of gait during normal and dual task walking in older individuals," *Gait Posture*, vol. 32, no. 1, pp. 23–28, May 2010.

[51] U. Ryu *et al.*, "Adaptive step detection algorithm for wireless smart step counter," in *Proc. Int. Conf. Inf.Sci. Appl. (ICISA)*, Jun. 2013, pp. 1–4.

[52] M. Yang, H. Zheng, H. Wang, S. McClean, and D. Newell, "iGAIT: An interactive accelerometer based gait analysis system," *Comput. Methods Programs Biomed.*, vol. 108, no. 2, pp. 715–723, 2012. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S016926071200106X

[53] P. Cichosz, "Classifcation model evaluation," in *Data Mining Algorithms Explained Using R*. Chichester, U.K.: Wiley, 2015, pp. 189–233.
[54] C. C. Aggarwal, *Data Classification: Algorithms and Applications*. Boca Raton, FL, USA: CRC Press, 2015. [Online]. Available: https://www.crcpress.com/Data-Classification-Algorithms-and-Applications/Aggarwal/p/book/9781466586741
[55] H. Jiawei, M. Kamber, J. Han, M. Kamber, and J. Pei, *Data Mining: Concepts and Techniques*. 2012.
[56] A. M. Molinaro, R. Simon, and R. M. Pfeiffer, "Prediction error estimation: A comparison of resampling methods," *Bioinformatics*, vol. 21, no. 15, pp. 3301–3307, 2005.
[57] J. L. Bellanca, K. A. Lowry, J. M. Van Swearingen, J. S. Brach, and M. S. Redfern, "Harmonic ratios: A quantification of step to step symmetry," *J. Biomech.*, vol. 46, no. 4, pp. 828–831, Feb. 2013.
[58] R. Moe-Nilssen and J. L. Helbostad, "Estimation of gait cycle characteristics by trunk accelerometry," *J. Biomech.*, vol. 37, no. 1, pp. 121–126, 2004. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0021929003002331
[59] R. G. Ramani, G. Sivagami, and S. G. Jacob, "Feature relevance analysis and classification of parkinson disease tele-monitoring data through data mining techniques," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, no. 3, pp. 298–304, 2012.

FRANCISCO DIEGO ACOSTA-ESCALANTE received the master's degree in computer science from the University of Grenoble National Polytechnic and University Joseph Fourier, France, and the Ph.D. degree (Hons.) in computer science from the University of Sciences and Techniques of Languedoc, Montpellier, France. He has been a Research Professor with the Universidad Juárez Autónoma de Tabasco for over 20 years. He is currently a Lecturer in computer science with the Universidad Juárez Autónoma de Tabasco for both undergraduate and graduate students. He has also given lectures at scientific events. He has authored over 30 undergraduate and graduate theses and 20 scientific publications in computing science. He recently focused on data science and Internet of Things. His research interests cover topics like system modeling, distributed activities on Ad Hoc networks, and ambient intelligence.

ELÍAS BELTRÁN-NATURI received the B.Sc. degree in informatics and the M.Sc. degree in computer system from the Orizaba Technological Institute, Mexico. He is currently pursuing the Ph.D. degree in computer science with the Universidad Juárez Autónoma de Tabasco, México.

MARIE CATHERINE BOLL received the Ph.D. degree in medical sciences from the Universit Nacional Autnoma de Mexico. She is currently a Neurologist at the Instituto Nacional de Neurologia y Neurocirugia MVS (INNN-MVS), Mexico. She is a Specialist in movement disorders and responsible for the Clinical Investigation Laboratory. She created the first Gait Laboratory of INNN. She carries out the institutional registry of prion diseases, ataxias, and chorea. She has authored several publications in indexed journals on biomarkers, mechanisms, and new treatments for neurodegenerative diseases. She is a member of the National System of Researchers level II.

JOSÉ ADÁN HERNÁNDEZ-NOLASCO received the bachelor's degree in electronic and communications engineering from the Autonomous University of Nuevo Len in 1996, the M.Sc. degree in electronic engineering (telecommunications) from the Monterrey Institute of Technology and Higher Education in 2003, and the Ph.D. degree in optics from the National Institute for Astrophysics, Optics and Electronics in 2012. He has been a Research Professor with the Universidad Juárez Autónoma de Tabasco for 20 years. He has authored or co-authored over 20 publications in the areas of optics and ambient intelligence and over 30 participations in conferences. His areas of interest are ubiquitous systems, ambient intelligence, telecommunications infrastructure, and the study of the propagation of communication signals.

PABLO PANCARDO GARCÍA received the Ph.D. degree in computer science from the Universidad Juárez Autónoma de Tabasco (UJAT) in 2016 and the M.Sc. degree in information technology from the Monterrey Institute of Technology and Advanced Studies in 1998. He is currently an Associate Professor and a Researcher with the Distributed Systems Group, UJAT. He has authored or co-authored over 40 publications in relevant international conferences and journals on ubiquitous computing and novel AmI applications related to welfare and healthcare. His main research interests are ambient intelligence, pervasive computing, IoT, and HCI. He has led/taken part in several research projects involving the adoption of mobile ubiquitous computing and sensor networks.