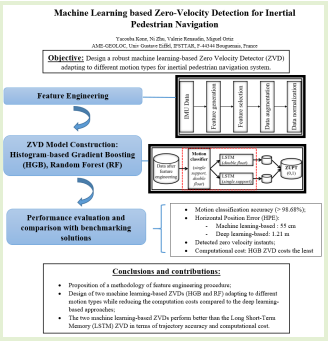
Machine Learning-Based Zero-Velocity Detection for Inertial Pedestrian Navigation

Yacouba Kone, Ni Zhu, Valerie Renaudin, *Member, IEEE*, and Miguel Ortiz

***Abstract*—Zero velocity update is a common and efficient approach to bound the accumulated error growth for foot-mounted inertial navigation system. Thus a robust zero velocity detector (ZVD) for all kinds of locomotion is needed for high accuracy pedestrian navigation systems. In this paper, we investigate two machine learning-based ZVDs: Histogram-based Gradient Boosting (HGB) and Random Forest (RF), aiming at adapting to different motion types while reducing the computation costs compared to the deep learning-based detectors. A complete data pre-processing procedure, including a feature engineering study and data augmentation techniques, is proposed. A motion classifier based on HGB is used to distinguish "single support" and "double float" motions. This concept is different from the traditional locomotion classification (walking, running, stair climbing) since it merges similar motions into the same class. The proposed ZVDs are evaluated with inertial data collected by two subjects over a 1.8 km indoor/outdoor path with different motions and speeds. The results show that without huge training dataset, these two machine learning-based ZVDs achieve better performances (55 cm positioning accuracy) and lower computational costs than the two deep learning-based Long Short-Term Memory methods (1.21 m positioning accuracy).**



***Index Terms*— Pedestrian navigation, inertial sensors, IMU, machine learning, zero-velocity detection.**

I. Introduction

FOR more than 15 years, research in pedestrian navigation has focused on solutions based on inertial signals to provide accurate and continuous indoor/outdoor positioning solutions. The advantage of inertial measurements is that they are possible in all environments including indoor spaces and they do not rise privacy problems as it is the case with radiotelecom based technologies [1]. However, the drift of the inertial sensors very quickly introduces large positioning errors that must be mitigated [2]. The use of walking phases derived from biomechanical knowledge [3], [4] was quickly adopted to replace the lack of GNSS (Global Navigation Satellite System) measurements in buildings. They are used to periodically bound the propagation of the inertial errors in the positioning algorithms [5]. The most used gait period is the single support mid-stance phase during which the speed of the inertial measurement unit (IMU) mounted on the foot equals zero. As walking is cyclic by nature, this observation, called ZUPT (Zero velocity UPdaTe), is applied regularly to ensure a good positioning accuracy [6]–[8]. The technique of ZUPT is also proved to be efficient in many other application domains such as robotic positioning [9].

Existing IMU based ZUPT detectors identify these periods using statistical features. They often depend on thresholds that must be calibrated to account for the different motions. Indeed, even if the use of ZUPT has a long history in pedestrian navigation, its implementation remains complex due to the large variations of human dynamics. When the foot movement is slow, e.g. when walking at comfortable speed, ZUPT detectors work well. But when the human dynamic increases, e.g. running, the performance of existing detectors tends to decrease [5], [6], resulting in false or missed detected ZUPT. More recently, a threshold-less approach [10] and methods based on artificial intelligence (AI) have been proposed to meet the challenge of adapting the ZUPT detectors to the diversity of human dynamics [11], [12]. They alleviate the need for threshold tuning. The most advanced approaches rely on deep leaning techniques that are sometimes combined with parallel motion classification methods. They provide more robust ZUPT detection performances [13] but rise questions about the need for large training databases and even heavy computation costs, preventing real time applications.

This work aims at achieving robust ZUPT detection for walking, running and stairs climbing/descending at different speeds using machine learning-based methods to reduce the computation cost. Real-time implementation of pedestrian

Manuscript received April 18, 2020; revised May 20, 2020; accepted May 20, 2020. Date of publication June 3, 2020; date of current version September 17, 2020. This work was supported in part by the National Research Agency (ANR) of France and in part by the French Defence Procurement Agency (DGA). The associate editor coordinating the review of this article and approving it for publication was Prof. Dongsoo Har. *(Corresponding author: Ni Zhu.)*

The authors are with AME-GEOLOC, IFSTTAR, University Gustave Eiffel, F-44344 Bouguenais, France (e-mail: ni.zhu@univ-eiffel.fr).

Digital Object Identifier 10.1109/JSEN.2020.2999863

positioning filter for complex motions is targeted. An innovation is to make no distinction between walking and stair climbing/descending for classification thanks to two new classes inspired by biomechanics. The performance is assessed with a 1.8 km indoor/outdoor test scenario combining all 4 types of motions ran by 2 persons in a 3-floor building. Performance is also assessed with the positioning error computed for these scenarios and compared with two LSTM approaches. A features engineering study completes the paper to identify the best parameters for zero-velocity detection with didactic parts. It is also found that these features enable robust detection of ZUPT even with rapidly changing movements. This is less true for LSTM trained on globally small databases, since LSTM does not allow random segmentation of the learning database at the sample level to prevent the risk of breaking temporal correlations.

II. State of the Art

A. Existing Classic Zero-Velocity Detection Methods

Zero-velocity detectors (ZVDs) find the instants that the IMU sensors are static corresponding to the stance phase of a walking gait. By detecting the ZUPT, the accumulated errors of foot-mounted inertial sensors can efficiently be bounded. In the current literature, most ZVDs use signal energy (e.g. from acceleration and angular rate) and their variances to distinguish the zero velocity instants. These kinds of detectors include:

- Acceleration or Angular Rate Moving Variance [14] or Moving Average [15] Detector;
- Acceleration Magnitude Detector [16];
- Angular Rate Energy Detector [17];
- The Stance Hypothesis Optimal Detector (SHOE) [6]: this detector is derived within a General Likelihood Ratio Test (GLRT) framework, in which a test statistic is compared with a predefined threshold to make the decision.

Among the above detectors, the moving variance detector can be combined with the magnitude or energy detector to obtain a better performance [18]. However, the limitation of these traditional detectors is their dependence on fixed thresholds. These thresholds vary significantly according to motion types (e.g. walking, running, climbing stairs, etc) as well as individuals. It is not possible to find a universal threshold value for all kinds of motions neither all the subjects. Therefore much research work about adaptive zero-velocity detection methods and some threshold-less ZVD emerged in the past years.

Usually, the adaptive ZVD is realized by establishing the relationship between the user motion and the optimal threshold. These approaches need abundant data collection as well as high computational cost. Based on the traditional SHOE detector, [19] proposed an adaptive ZVD by combining a pressure sensor with accelerometer and gyroscope. It introduced a new test statistic to add pressure sensor records. By enlarging the difference of the test statistics between motion instants and zero velocity instants, it can effectively mitigate the false alarm detection. Then, by using the angular rate magnitude, motion classifications are realized. Finally, two threshold functions for walking/running and ascending/descending stairs are established based on second-order polynomial fitting. These functions are applied according to the motion classification results. A limitation of this adaptive method is its dependency on extra sensors (pressure sensor). Although this sensor could help improve the detection accuracy performance, at the same time, it increases the complexity, the cost and the sensitivity of the complete system. This would make the system less ubiquitous, which limits the user case in some extreme environments such as fire-fighters. Another limitation is linked to the adaptive function established that is specific for one subject. Thus the computation cost will increase when applied to several subjects.

Reference [5] proposed an adaptive ZVD solely based on inertial signals. It distinguishes the swing and the stance phases of a gait cycle with a clustering algorithm. It partitions the potential gait phases into true and false cluster thereby yielding an efficient elimination of the false gait phase. The threshold is adapted to these different recognized gait cycle stages but the corresponding parameters for each gait phase still need to be tuned.

Different from the previous ZVDs which depend on predefined thresholds, [10] proposes a threshold-less ZVD. Based on the shape of the simplified sensor signals through signal processing procedures, it fixes first the most likely zero velocity interval by detecting consecutive two positive pics of angular rates. Then, the zero velocity instants are fixed within the previous interval by using a synthesized signal constructed from the accelerometer signals. It is shown that this algorithm can effectively detect the zero velocity moments for different pedestrians but only the motion types of walking, side-walking and backward walking at different speed are tested and validated in [10]. Its performance and robustness for other motion types are still unknown.

B. Artificial Intelligence (AI)-Based Zero Velocity Detection Methods

A promising way to solve the ZVD problem for different motion types is using Artificial Intelligence (AI)-based methods. Some research work has already been conducted on ZVD using AI approaches showing encouraging results. Generally, AI-based approaches need a variety of data collected from different subjects and different motion types. These data should be labeled to train the model in a supervised way. Existing research work performs the data labelling with extra sensors (e.g., pressure sensor [6]), manual annotation according to certain criteria [20], or with the help of a high accuracy motion capture system [13]. Then the training and validation procedures is done either by machine learning or deep learning approach. Most existing IA-based ZVDs can be divided into two parts: a motion classification module and a stance phase detection module. They can be both or partly realized by IA methods.

Reference [21] proposed a ZVD by Support Vector Machine (SVM) using inertial sensor signals. The motion classification is firstly realized by SVM, where 24 motion types are involved and the classification can reach an accuracy of 98.27%. Knowing the motion type, the stance phase detection is done again with SVM and an additional glitch removal technique is

performed to delete the false positive detection. This proposed algorithm is tested for one subject with different motion types. A 99% accuracy is obtained for ZVD. Its robustness, however, needs to be further validated for various subjects and tested in the navigation algorithm.

As a series of research work concerning the IA-based ZVD, [22] trained an SVM classifier using foot mounted IMU to classify six motion types: walking, jogging, running, sprinting, crouch-walking and ladder-climbing. The mean test classification accuracy is 90% on a dataset with five different subjects. Then, according to the predicted motion type, the traditional SHOE detector is associated at the next step by using the pre-fixed optimal thresholds for each motion. The entire dataset consisted of 30000 training and 30000 test samples in six motion classes collected from five subjects. The complete algorithm is only tested for walking, running and a combined walking/running test. Then, [12] trained a long short term memory (LSTM) neural network to directly detect the stance phase of a gait cycle without prior motion classification and the threshold tuning.

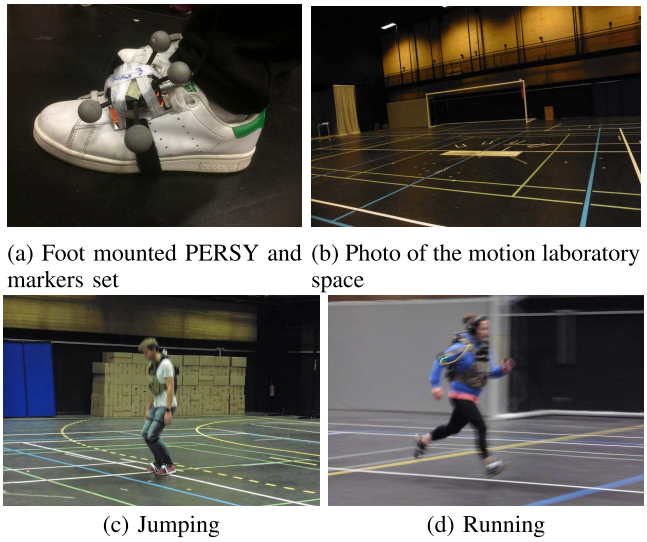
Reference [13] synthesizes the previous two research work and expands the motion types to a stair-climbing case. It also provides a quantitative comparison between the previous two approaches and several classic detectors. Training and test data are collected from five subjects and each subject traveled approximately 110 m for each motion type. Every subject repeated each motion type three times, which results in a total of 45 motion trials covering approximately 6.6 km. The test results show that both the proposed two approaches outperform the existing threshold-dependent ZVD. But the computational cost was not reported here and the deep learning methods are generally computationally expensive.

III. Description of the Training Dataset

A. Inertial Signals Collected in a Large Motion Laboratory

1) Experimental Equipment: The hardware used to collect data for building the training dataset has been designed by GEOLOC laboratory [23]. It is the PERSY unit (PEdestrian Reference System) that is used as a positioning reference system for pedestrian navigation research. Mounted on the foot, it outputs location estimates relative to a starting position with a 0.3 % positioning mean error of the traveled distance [14]. These performances depend on the good recognition of quasi-static phases of the acceleration and the magnetic field records that are used to mitigate the impact of inertial sensor errors in the positioning algorithm. Outdoors and in light indoor spaces, GNSS phase measurements are added to the strapdown Extended Kalman Filter to improve the positioning accuracy. PERSY comprises a tri-axis accelerometer, a tri-axis gyroscope, a tri-axis magnetometer and a single-frequency, dual-constellation GNSS receiver. All data are synchronized with GPS time and recorded at a 160 Hz sampling frequency.

The training database was collected with PERSY mounted on the foot in a large motion laboratory facility. The cluster of markers, needed by the motion laboratory tracking system, was also fixed on PERSY (Fig. 1a). It is used by the motion capture system to estimate the position and the orientation



(a) Foot mounted PERSY and markers set (b) Photo of the motion laboratory space

(c) Jumping (d) Running

Fig. 1. Illustrations of the motion laboratory experiment.

of the cluster's center at 250 Hz with a millimeter level accuracy. These estimates are used as ground truth in this research. As shown in Fig. 1b, the motion capture system is installed in a large gymnasium (30 × 20 m) with two force plates (120 × 60 cm) in its centre allowing realistic human motion analysis.

2) Description of the Scenarios: 7 test subjects of variable size, weight and age participated in the data collection. Among them were 4 women and 3 men who performed running, walking and sidestepping motions during the experiment as shown in Fig. 1c and Fig. 1d. The motions were performed at different speeds using a headset with different beats-per-minute (bpm) rates. The data were recorded for 9 different bpm: 100, 115, 130, 145, 160, 175, 190, 205, 220. In total, the database comprises signals for 63 runs, which corresponds to 9 tracks per person at the different bpm.

3) Two Classes to Describe the Recorded Human Gaits: The walking, running and sidestepping motions were classified in two classes, which are the "single support" and "double float" classes. The single support class corresponds to human gait where at least one foot touches the ground during motion. It is inspired by the biomechanics definition [24] of walking but includes more motions such as standing, stairs climbing/descending and sidestepping at low and comfortable speeds. The class double float corresponds to movements with an aerial phase: a period when both feet are off the ground. Running belongs to this class. Globally, the double float class corresponds to short contact times of the feet on the ground, whereas the single support class corresponds to longer contact times.

B. Pre-Processing of the Inertial Data

Data preprocessing is the first step in machine learning. It transforms and structures the data into a state more appropriate for the learning phase that eases the interpretation of the data characteristics. Our preprocessing process comprises the three following steps:

- Synchronization of the data
- Labeling of the data
- Calibration of the data

1) Data Synchronization: Our two data sources are recorded at different sampling frequencies. The IMU data is recorded at 160 Hz while the motion capture data is recorded at 250 Hz. Data must first be time-synchronized before being segmented and used for identifying the ZUPT instants. Let's note that these instants vary according to the individuals and the motion dynamics. The more segmented data there is, the more accurate will be the estimated ZUPT instants at higher dynamics. The synchronization was done manually for each subject and bpm. This step is even more important for high-speed movements, such as those belonging to the "double float" class defined in section V.

2) Labeling: is a critical step in machine learning as it supports the learning phase to avoid false positives and true negatives. To be as close as possible to the final positioning estimation process and to ensure good labeling, we used the PERSY algorithm suite [14] to identify the true ZUPT instants in the inertial signals. The reference positions of the foot-mounted IMU, provided by the motion laboratory, were used for comparison with the positions estimated by the PERSY Extended Kalman Filter. This was done manually and repeated for each test subject using the associated reference data set from the motion laboratory. Globally, the ZUPT labeling was done using two statistics for single support and three for double float. For single support, they are the norm of the accelerations and its variance for a 0.25 s sliding window. For the double float, the norm of the gyroscope is added to the two previous statistics for 10 samples or 0.0625 s sliding window.

3) Data Calibration: The IMU embedded in PERSY is affected by noise and must be properly calibrated. It was calibrated for misalignment and bias terms using a 3D rotating platform whose horizontality is guaranteed by a spherical level in its middle. The main idea of this calibration is to consider that a perfect accelerometer only measures the gravity force while being static. Two parameters are calculated by this calibration:

- a rotation matrix for misalignment and scale factor
- and a bias vector.

1-minute static observation files were recorded along each axis of the body frame of PERSY in both up and down directions. This 6 positions calibration procedure is based on least-squares estimation. Misalignment and bias estimates are used to calibrate all raw inertial data collected with PERSY in this study.

IV. Feature Engineering for Detecting Zero Velocity Foot Motion Periods

Machine learning algorithm can be fed either by raw data or processed data using feature engineering methods. These methods transform raw data into features to better represent the problem underlying predictive models and improve the accuracy of the model on new data. It aims at producing features as representative as possible for a given problem and consists in the following tasks (Fig. 2):

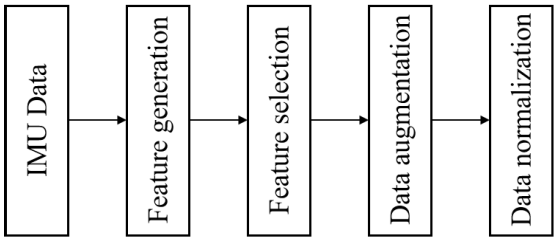


Fig. 2. Feature engineering process.

- generating new features,
- selecting among the set of these features the most relevant for our model,
- increasing the data by adding reduced centered white Gaussian noise on a part of the data,
- finally in normalizing the data.

A. Construction of the Features Based Learning Database

The construction of the learning database is essential for machine learning methods since it significantly impacts the performance. The use of raw data may improve the results while other times they degrade the performance of the model. We adopted a features engineering strategy to derive the most explanatory features of our model. The database (section III) dedicated to learning the ZUPT instants with inertial signals is divided into two sub-dataset linked to the classes single support and double float. Each sub-dataset includes 13 features. The sizes of the sub-bases are respectively, 30MB with 77193 observations of which 21% are ZUPT instant for single support and 9MB with 83571 observations corresponding to 25% of ZUPT instant in double float.

B. Selection of the Best Features for ZUPT Detection

1) Features Generation: The most interesting statistics are those highly correlated with the variable of interest: ZUPT times. Here is a list of the selected features that were estimated over a 0.03125 s manually tuned sliding window, corresponding to 5 steps.

The mean value $\overline{y}$ of the signal y over the N samples sliding window is given by

$$\overline{y} = \frac{\sum_{i=1}^{N} y_i}{N} \tag{1}$$

The variance measures the dispersion of the signal and is given by

$$Var(y) = \frac{\sum_{i=1}^{N} (y_i - \overline{y})^2}{N} \tag{2}$$

The Root Mean Squared Error (RMSE) of the signal is given by

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (y_i - \overline{y})}{N-1}} \tag{3}$$

The Mean of Absolute Error (MAE) is the average of the absolute values of the residuals.

$$MAE = \frac{\sum_{i=1}^{N} |y_i - \overline{y}|}{N} \tag{4}$$

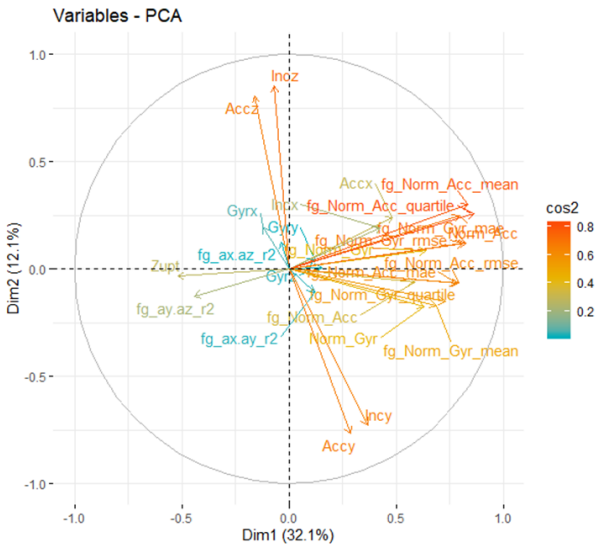


Fig. 3. PCA-based correlation circle.

The inter-axis correlation or correlation coefficient measures the statistical relationship between two variables (y, x). Reference [25] used it to differentiate running from cycling with good results.

$$r_{\vec{X},\vec{Y}} = \frac{N(\sum_{i=1}^{N} x_i y_i) - (\sum_{i=1}^{N} x_i)(\sum_{i=1}^{N} y_i)}{\sqrt{[N\sum_{i=1}^{N} x_i^2 - (\sum_{i=1}^{N} x_i)^2][N\sum_{i=1}^{N} y_i^2 - (\sum_{i=1}^{N} y_i)^2]}} \quad (5)$$

The third quartile ($Q3$) splits off the highest 25% of the data from the lowest 75%.

Except for the inter-axis correlation that uses all three axes of the linear acceleration, other features were estimated using either the norm of the acceleration or the norm of the gyroscope signals.

2) Features Selection: This task selects the most informative features for the ZUPT detection in a set of statistics. It is important since the use of many non-informative variables often results in over-fitting and unnecessary large computing costs. Many features selection techniques exist: the correlation coefficient, RF, Principal Component Analysis (PCA), Logistic Regression, RFE (Recursive Feature Extraction without Cross-validation) or RFECV (Recursive Feature Extraction with Cross-validation) and Independence Test (Chi-square). Reference [26] details the theory of feature generation and selection.

a) Correlation index: The correlation coefficient quantifies the degree of relationship between features. The ZUPT variable is considered as the probability that the velocity at time t equals zero. Being considered as a discrete quantitative variable, the use of the correlation coefficient becomes possible. The PCA can be used to reduce the dimensions. Fig. 3 illustrated the PCA-based correlation circle used to visually assess the degree of relationship between different variables along the factorial axes. The cosine of the angle separating the various variables illustrates its influence. Fig. 3 shows the highest correlation for the ZUPT detection.

We observed that the most correlated variable with ZUPT is r_{a_y,a_z}, which is the inter-axis correlation between the accelerations along the Y and Z axis respectively (with a correlation coefficient $< 30\%$). Although this link is linear, it is too weak to properly model our phenomenon. Other variables are needed to increase the percentage of the total variance for ZUPT phenomenon.

b) Logistic regression: This supervised learning technique was adopted to identify more relevant statistics. It studies the link between a qualitative variable Y and a set of explanatory variables $X_j, j = 1, \ldots, n$. The variables whose p-value is lower than 1‰, i.e. very significant, are considered as influential for the binary ZUPT variable $\{0, 1\}$. The logistic regression analysis identified the following variables:

- the norm of the acceleration $||a||$
- the norm of the gyroscope $||w||$
- $MAE_{||a||}$, $MAE_{||w||}$ and $RMSE_{||a||}$
- $Q3_{||a||}$ and $Q3_{||w||}$
- $Var(||a||)$ and $Var(||w||)$
- r_{a_y,a_z}: the correlation coefficient between a_y and a_z
- r_{a_x,a_z}: the correlation coefficient between a_x and a_z

3) Data Augmentation: Data augmentation was applied to enrich the database using rotations and additive noises. It increases robustness to new data sets that have not been used to train the algorithm [27]. It was only applied to subset of the database.

4) Data Normalisation: Normalisation is used to mitigate the impact of the variation between training and testing and maintain good prediction. Its use for real-time implementation is sometimes difficult. Furthermore, although quite good in practice, this technique is less effective for database whose variation is small, as is the case for high-frequency acceleration data. Standard normalization and Z-score are common normalization techniques in machine learning. Our data being recorded every 0.00625s, which gives a rather small variation between two records, we did not apply normalization.

C. Partitioning of the Data

Efficient learning requires adopting a good data partitioning strategy. Indeed, a too large training database can lead the algorithm to over-fitting [28], [29]. In this case, the algorithm produces very good results on the training data but the prediction error on new data is quite high. Also, a weak database is a source of under-fitting [29] where the prediction error strongly varies from one set of data to another. The balance between under-fitting and over-fitting is the optimum for data partitioning. Another source of problem is the non-representative aspect, in the statistical sense, between training and testing data. Several techniques based on cross-validation exist to mitigate this. Considering the size of our database, we considered 70% of the data for learning and 30% for testing. We also used the k-fold cross-validation method [30] to counteract the effect of non-representative data. This non-exhaustive method divides the base into k classes of equal size. One class serves as a test and the k-1 other classes as a data training. This process is repeated k times. We used the root mean square error (RMSE) as the loss function associated for validation.

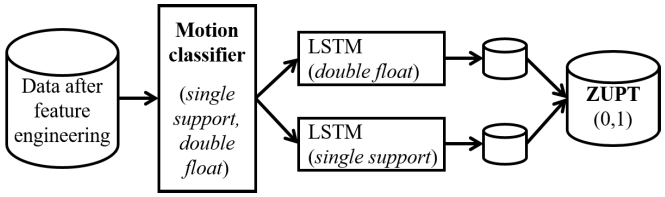


Fig. 4. Architecture of the LSTM based ZUPT detection method.

V. Model Construction for ZUPT Detection

Machine learning and deep learning are the most widely used methods in artificial intelligence (AI). They aim at predicting observations on new data sets that have not been used to train the algorithm. But their computation cost differs depending on the type of method used. In this section, we present two machine learning algorithms whose performance approaches or surpasses that of deep learning (LSTM) with a lower computational cost. The search for reduced computation costs is essential for real-time implementation on smart-devices. Let's note that the training of these two models was done with the same dataset previously mentioned.

To better detect ZUPT instants, the process is divided into two main steps. The first step classifies the human dynamics into the two classes: "single support" and "double float". Knowing the motion class, the second step predicts the ZUPT instants using AI-based techniques. This process is applied to all three algorithms: Random Forest (RF), Histogram-based Gradient Boosting (HGB) and Long short-term memory network (LSTM). Fig. 4 shows this process to predict ZUPT with the LSTM deep learning algorithm.

A. Machine Learning Based Approach

1) Random Forest Classifier Training: Random Forest [31] classifier is a classification algorithm that reduces the variance of forecasts in a decision tree. To improve the performance, it combines many decision trees in a bagging approach. Multiple randomly constructed decision trees are trained on different subsets of data. Concretely, each tree of the RF is trained on a random subset of data according to the bagging principle, with a random subset of features according to the "random projection" principle. The predictions are then used to make a vote. We implemented the RF with the Scikit-learn package whose parameters are given in Table I.

2) Histogram-Based Gradient Boosting Training: The gradient boosting classifier is a supervised machine learning technique based on two parts: gradient descent and boosting. Its loss function is the logarithm of the mean squared error (MSE). It is a variant of light GBM, which is inspired by the Gradient Boosting classifier. These histogram-based estimators are faster than Gradient Boosting Classifier when the number of samples is greater than ten of thousands of samples, as in our case. Similarly to light GBM, HGB can handle large amount of data and requires less memory to run. Also, it emphasizes the accuracy of the results. We chose this algorithm because of its good performance on machine learning competition platforms such as Kaggle. It is an asset in terms of memory usage and speed of execution, which gives a real advantage for real-time implementation [32]. Like for the RF, the Scikit-learn package has been used.

TABLE I
Parameters for Single Support(Walking)/ Double Float(Running)

Algorithm	Type of motion		
	Parameters	*Single Support*	*Double Float*
RF	n_estimators	1000	1477
	min_samples_split	2	2
	min_samples_leaf	1	1
	max_features	auto	sqrt
	max_depth	50	94
	bootstrap	True	False
HGB	validation_fraction	0.9	0.9
	min_samples_leaf	66	2
	max_iter	200	1600
	max_depth	30	90
	max_bins	50	200
	loss	auto	binary CE
LSTM	Number of layer	5	5
	1er layer	64u bidir	64u bidir
	2nd layer	64u	64u
	3rd layer	64u	64u relu
	4th dopout	0.5	0.5
	Last layer	1u sigmoid	1u sigmoid
	Optimizer	Adam	Adam
	Loss	binary CE	binary CE
	metrics	binary_accuracy	binary_accuracy
	epochs	5	50
	batch_size	100	100

CE : Cross Entropy

B. Deep Learning Based Approach

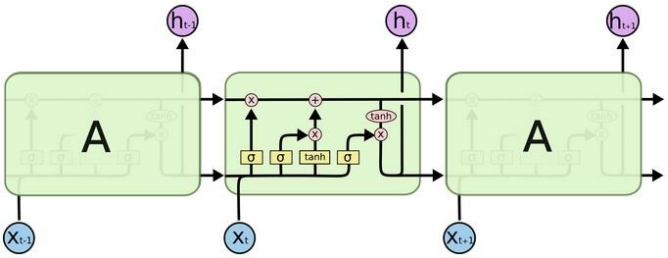
The Recurrent Neural Network (RNN) is a generalization of the feed-forward neural network that has internal memory. The RNN is recurrent by nature because it fulfills the same function for each data input while the output of the current input depends on the last calculation. After producing the output, it is copied and returned to the recurrent network. To make a decision, it considers the current input and the output, learned from the previous input. Although it has a memory, this network is rather limited in practice. Indeed, it faces the explosion or disappearance of gradients and it is difficult to form.

Long Short Term Memory (LSTM) networks are RNN capable of learning long term dependencies. Remembering information for long periods is done by default using back-propagation method to train the model. The prediction of ZUPT faces the same long-term dependency phenomenon. This explains the growth of recent research targeting pedestrian inertial navigation for this deep learning approach [33]. We trained an LSTM to compare the performance of our machine learning based methods with the later.

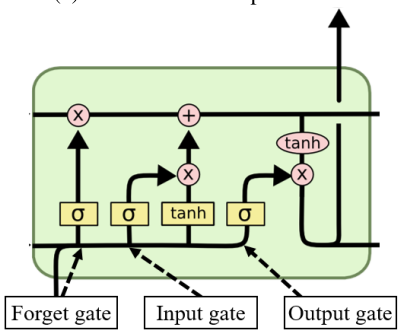
Python Keras was used to train the LSTM models for both single support and double float classes. Cross-entropy was used in both types of movement as a loss function and Adam as an optimizer. The different models converged for both classes, respectively, after 50 and 5 epochs with a batch size of 100 for each.

C. Hyperparametrization of Models

The model parameters are used to link the input data to the desired results. They are learned during model training. These hyperparameters determine the overall structure of our model.



(a) Overview of simple LSTM



(b) Focus on the main gate

Fig. 5. Architecture of LSTM from [36].

The settings of the machine learning models are linked to cost functions. Thus hypermediatization consists in running a statistical optimization. Based on a set of hyperparameters, this task searches for the best combination of their values to find either the minimum (e.g. the loss) or the maximum (e.g. the precision) of a function [34]. Several hypermediatization techniques exist in machine learning as well as in deep learning. The most common ones for machine learning are RandomSearch and GridSearch. For deep learning, they are Adam, Adagrad, momentum, RMSProp.

The hyperparameters estimated with RandomSearch and deep learning Adam as optimizers for our ZUPT detection problem are summarized Table I. RandomSearch was chosen because of its rapid execution speed and convergence time. Adam optimizer was found to be the best for our model.

D. Computation Cost Estimation

LSTM performance strongly depends on its construction. The more hidden layers a network has, the better its prediction capability will be. However, the more layers we add, the more complex the network becomes. this implies a many parameters to estimate which has a considerable impact on the computation cost. Let's consider the case of the simple LSTM network (Fig. 5a) proposed by [35]. The main part has three main gates (Fig. 5b). They are the input, the forget and the output gates that are regulating the flow of information. For each gate, we have at least one linear combination, two parameters to estimate, element by element products or addition and a tangent or sigmoid calculations. Computation is repeated as many times as there is hidden layers.

Boosting is a clustering technique that consists of aggregating classifiers (models) developed sequentially on a learning sample whose individual weights are corrected as they are learned. The classifiers are weighted according to their performance. As for the descent of the gradient, it is an iterative technique close to an optimization problem. Gradient Boosting Tree Classifier is a technique that combines the notions of boosting, gradient descent and regression tree to minimize the error between the true value and the predicted value [37].

TABLE II
TRAINING COMPUTATION COST IN SECONDS

Algorithm	*Training Times (s)*
HGB	21.499
RF	324.921
LSTM	539.726

The construction of a Gradient Boosting requires sorting the samples at each node and for each variable. Sorting is necessary to calculate efficiently the potential gain of a separation point. At each node division, the complexity equals $\theta(n_{features} * nlog(n))$, n being the number of samples at the node. HGB, in contrast, doesn't require sorting the feature values. Instead, it uses a data-structure called a histogram, where the samples are implicitly ordered. The complexity of building a histogram is $\theta(n)$. The complexity of the node splitting procedure is $\theta(n_{feature} * n)$, which is much smaller than the previous one. In addition, instead of considering split points, we consider only max_bins (i.e. the number of bins that the values will be grouped in) split points, which is much smaller [38]. Also, to build histograms, the input data needs to be binned into integer-value bins. This binning procedure requires sorting the feature values, but it only happens once at the very beginning of the boosting process and not at each node, like in Gradient Boosting Classifier and Gradient Boosting Regressor.

The parallelization of calculations is an inconstant asset nowadays with very large amounts of data. Indeed, HGB uses this very powerful calculation method to calculate its parameters. Consequently, all following HGB tasks are parallelized.

- Mapping samples from real values to integer-valued bins (finding the bin thresholds is however sequential).
- Building histograms is parallelized over features.
- Finding the best split point at a node is parallelized over features.
- during the adjustement of parameters, mapping samples into the left and right children is parallelized over samples.
- Gradient and hessians computations are parallelized over samples.
- Predicting is parallelized over samples.

Table II gives the training time for each model. It corresponds to the execution time of the code without including the hyperparametrization time. Let's note that these times are not directly linked to real-time performances of the proposed algorithms since training is computed on a computer different from the final target (smartwatch/IoT/small integrated motherboard).

VI. PERFORMANCE EVALUATION ON INDOOR/ OUTDOOR PEDESTRIAN TRACKS

A. Description of the Experimental Scenarios

New data sets were collected by two new test subjects of different sizes (2 m and 1.66 m height) to assess the

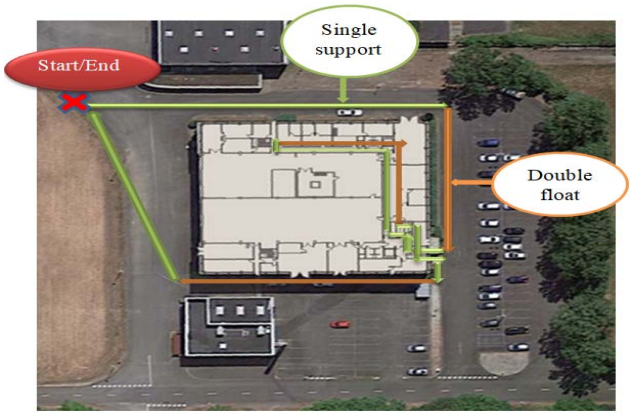


Fig. 6. Test scenario.

(a) Corridors (b) Starting position (c) Staircases

Fig. 7. Indoor/outdoor 1.8 km experiment.

performance of our models. They consist in about 450 m long indoor outdoor scenario in a 3 floor building located in the campus of our university. The scenarios were repeated two times by each person leading to a 1.8 km test. The complete path is shown in Fig. 6. Each scenario comprises single support (green in Fig. 6) and double float motions (orange in Fig. 6). The test includes stair ascent/descent (Fig. 7c), static phases, walking at comfortable speed (Fig. 7a), speed walking and fast running. The experiment started outdoors (Fig. 7b) and continued indoors from the 1st floor up to the 3rd floor alternating single support and double float motions.

B. Computation of Two Benchmarking Solutions

Two foot-mounted inertial navigation benchmarking solutions are used to assess the performance of our approach. The first one is the recently published AI based ZUPT detection method in [13], labeled LSTM utiasSTARS in the rest of the paper. The second solution is the Extended Kalman Filter based software suite [14], labeled PERSY.

To run LSTM utiasSTARS on our evaluation dataset, we used the model provided by [13] in the github. Since all training parameters are not provided (e.g. learning rate), the model was directly applied on this paper's dataset. Training with several learning rates was attempted without success leading to overfitting. We transformed the evaluation dataset into arrays following the instruction of the demo code. Finally, we used the LSTM class to predict the ZUPT before exporting them to estimate the coordinates of the scenarios. To run PERSY algorithms, which is used as a reference trajectory

TABLE III
MOTION CLASSIFICATION ACCURACY

Algo.	Subject 1		Subject 2	
	simple support	double float	simple support	double float
HGB	99.90 %	99.70 %	99.98%	98.68%

TABLE IV
STATISTICAL RESULTS OF THE HPE FOR THE FOUR IA-BASED METHODS APPLIED TO THE 2 SUBJECTS

HPE [m]	Subject 1			Subject 2		
	mean	rms	std	mean	rms	std
HGB	0.491	0.597	0.339	0.413	0.508	0.296
RF	0.448	0.534	0.292	0.415	0.542	0.350
LSTM	0.698	1.033	0.761	0.688	0.990	0.712
LSTM utiasSTARS	1.414	1.705	0.953	0.979	1.122	0.550

with a 0.3% accuracy on horizontal positioning error (HPE), the ZUPT detector was manually tuned to minimize the loop closure error.

C. Evaluation Criteria

The performance of the proposed ZUPT detectors is evaluated according to the following criteria:

- the accuracy of the activity classification;
- the comparison of artificial intelligence based detected ZUPT with PERSY's labels;
- the HPE of the trajectories with IA-based ZUPT detector;
- and the computational cost.

D. Activity Classification Accuracy

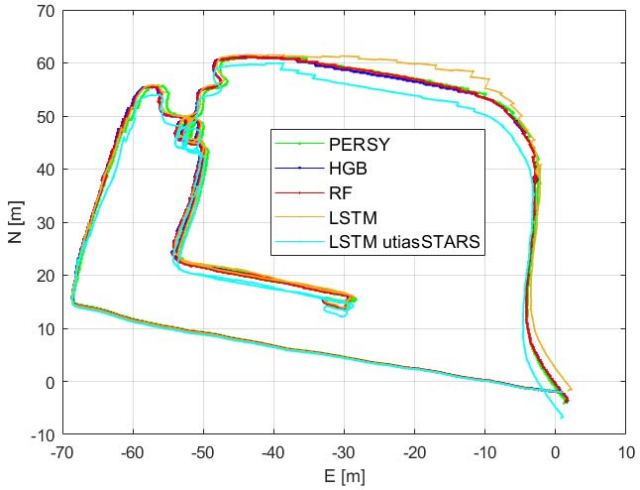
HGB method is adopted for motion classification whose performance is given in Table III. The lowest accuracy for the two dataset is 98.68%, which highlights the efficiency of this HGB based motion classifier. This result comforts the definition of the two motion classes: single support and double float, instead of the traditional walking and running classes, since the stair-climbing case is treated as single support here instead of creating a third class.

E. Horizontal Position Error (HPE)

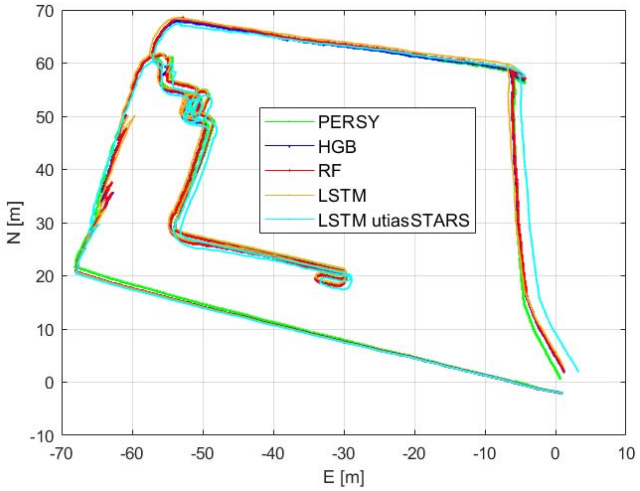
Fig. 8 shows the estimated trajectories of the four IA-based ZVDs as well as the reference trajectory estimated by PERSY (Fig. 8a for the subject 1 and Fig. 8b for the subject 2. Fig. 9 plots the HPE of the trajectories aided by IA-based ZVDs compared to PERSY reference trajectory. The statistical results of the HPE are reported in Table IV.

We observe that the two machine learning-based ZVDs (HGB and RF) have similar performances concerning trajectory accuracy. The root mean square (RMS) HPE of the HGB and RF are respectively 0.597 m and 0.534 m for the subject 1 and 0.508 m and 0.542 m for the subject 2.

Looking at the deep-learning-based methods, the accuracy of the trajectories estimated by LSTM and LSTM utiasSTARS are found to be lower than the previous one. LSTM and LSTM utiasSTARS give an RMS HPE of 1.033 m and 1.705 m



(a) Estimated trajectories of the subject 1



(b) Estimated trajectories of the subject 2

Fig. 8. Trajectories of PERSY and four IA-based ZUPT detectors.

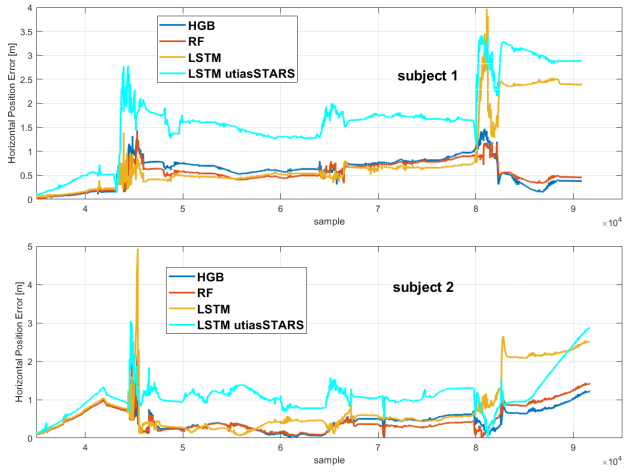
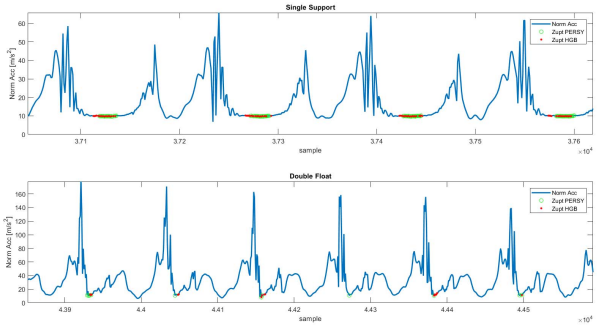
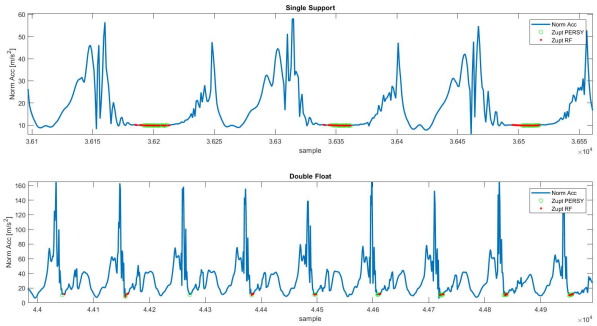


Fig. 9. HPE of trajectories aided by four IA-based ZUPT detectors.

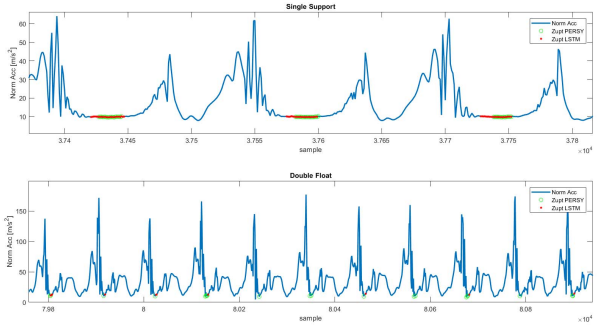
respectively for the subject 1 and 0.990 m and 1.122 m for subject 2. Fig. 8 shows that the LSTM-based estimated trajectory is biased especially during the double float part,



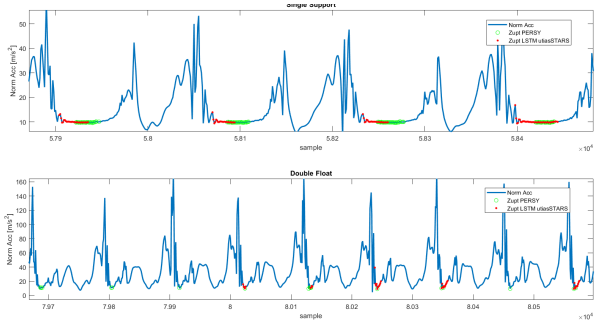
(a) ZUPT PERSY *vs* ZUPT HGB



(b) ZUPT PERSY *vs* ZUPT RF



(c) ZUPT PERSY *vs* ZUPT LSTM



(d) ZUPT PERSY *vs* ZUPT LSTM utiasSTARS

Fig. 10. Comparison of the detected zero velocity instants.

which indicates possible miss detections of the zero velocity instants. This is further analyzed in the next subsection.

F. Comparison of Detected Zero Velocity Instants

Fig. 10 compares the zero velocity instants detected by the four IA-based ZVDs as well as those detected by the manual tuning of the PERSY detector. In each sub-figure, the upper one represents the detected zero velocity instants for single

TABLE V
COMPUTATION COSTS IN SECONDS FOR THE TEST DATASET

Algorithm	*Subject 1*	*Subject 2*
HGB	1.547	1.4
LSTM	5.551	5.197
RF	12.607	13.108
LSTM utiasSTARS	68.09	75.009

support and the lower one represents the detected zero velocity instants for double float.

Similar performance are achieved by the four IA-based detectors for the single support motion type. Although the IA-based detectors detect in general more zero velocity instants than the reference detector, it does not impact much on the estimated trajectories (as shown in the single support part of the trajectories in Fig. 8). However, when looking at the detected zero velocity instants for the double float motion type, all the four IA-based ZVDs suffer from miss-detection. Among these four IA-based detectors, the LSTM-based methods are the most impacted leading to the propagation of more unbounded errors in the estimated trajectory.

In the RNN family, LSTM is widely adopted for its memory effect, i.e. its ability to learn a time dependency. However, this capacity is very specific to the dataset since the chaining of the sequences remains the same from the learning to the application dataset. Therefore, while algorithms that were not designed to include time dependence can be trained with a random sampling of data samples, this is not possible for LSTM. In the case of an LSTM, sampling must be performed using sets of records that are arranged in a specific order. This has a direct impact on the use of AI-based methods for pedestrian navigation.

Since human gait can change rapidly but following specific sequencing dictated by physical and/or surrounding constraints, when LSTM is trained with a specific motion sequencing, it will produce fewer mistakes for the same type of sequencing. Whereas, a sudden change in this sequencing will affect the LSTM predictions. This can explain the reduced detection success rate in our LSTM model. Contrary to LSTM, machine learning algorithms are not designed to incorporate this dependency. However, it can be added in the model thanks to features. Consequently, machine learning-based methods tend to be more robust to the specificity of inertial time-series processing for pedestrian navigation including complex and rapidly changing movements.

G. Computation Costs

The computation time was estimated with a HP-ZBook computer with a 16 GB RAM and an intel i7 2.90 GHz processor on the two databases containing respectively 90950 rows, 96998 rows and 13 features. Table V gives the time for both evaluation datasets. A significant difference between the four learning methods is observed. HGB has the lowest computation time and is 3 times smaller than our LSTM model. LSTM utiasSTARS has the highest computation time. HGB is found to be suitable for real-time implementation.

VII. CONCLUSION

This paper presents two machine learning-based ZVDs, i.e., Histogram-based Gradient Boosting (HGB) and Random Forest (RF). A methodology of feature engineering procedures is described to find the most explanatory features related to zero velocity detector (ZVD). Then, the model is constructed by combing a motion classifier and previously mentioned machine learning methods. Novel human gait classification classes for pedestrian navigation application is adopted. They are the "single support" and the "double float" classes inspired by biomechanics. This strategy avoids distinguishing walking from stair climbing and improves the classification results.

The performances of the proposed machine learning-based ZVDs are evaluated over a 1.8 km indoor/outdoor scenario involving two subjects. The test includes walking, running and stair climbing/descending in a 3-floor building. Detected zero velocity instants are compared with those of two LSTM based approaches for benchmarking. The accuracy of our motion classification can achieve more than 98.68%. Pedestrian trajectories are also estimated using the same Extended Kalman Filter and the zero velocity instants detected by the HGB, RF, the two LSTM methods. The two machine learning-based ZVDs achieve similar positioning accuracy (55 cm RMS) and a better one than LSTM based methods (1.21 m RMS) due to a lower ZVDs success rate. Furthermore, HGB and RF-based ZVDs have lower computational costs than the deep learning-based LSTM ZVDs. Globally, with the lowest computational cost and the best performances, HGB-based ZVD is found promising to be implemented for real-time applications.

ACKNOWLEDGMENT

This work was conducted with the contribution of the CyborgLoc team members for the competition MALIN (MAîtrise de la Localisation INdoor).

REFERENCES

[1] V. Renaudin, A. Dommes, and M. Guilbot, "Engineering, human, and legal challenges of navigation systems for personal mobility," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 1, pp. 177–191, Jan. 2017.

[2] Y. Wang, A. Chernyshoff, and A. M. Shkel, "Error analysis of ZUPT-aided pedestrian inertial navigation," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2018, pp. 206–212.

[3] J. Rose and J. G. Gamble, *Human Walking*, 3rd ed. Philadelphia, PA, USA: Lippincott Williams & Wilkins, 2005.

[4] J. Perul and V. Renaudin, "Learning individual models to estimate the walking direction of mobile phone users," *IEEE Sensors J.*, vol. 19, no. 24, pp. 12306–12315, Dec. 2019.

[5] Z. Wang, H. Zhao, S. Qiu, and Q. Gao, "Stance-phase detection for ZUPT-aided foot-mounted pedestrian navigation system," *IEEE/ASME Trans. Mechatronics*, vol. 20, no. 6, pp. 3170–3181, Dec. 2015.

[6] I. Skog, P. Handel, J. O. Nilsson, and J. Rantakokko, "Zero-velocity detection—An algorithm evaluation," *IEEE Trans. Biomed. Eng.*, vol. 57, no. 11, pp. 2657–2666, Nov. 2010.

[7] R. Harle, "A survey of indoor inertial positioning systems for pedestrians," *IEEE Commun. Surveys Tuts.*, vol. 15, no. 3, pp. 1281–1293, 3rd Quart., 2013.

[8] N. Yu, Y. Li, X. Ma, Y. Wu, and R. Feng, "Comparison of pedestrian tracking methods based on foot- and waist-mounted inertial sensors and handheld smartphones," *IEEE Sensors J.*, vol. 19, no. 18, pp. 8160–8173, Sep. 2019.

[9] H. Yang *et al.*, "Research on the strategy of motion constraint-aided ZUPT for the SINS positioning system of a shearer," *Micromachines*, vol. 8, no. 11, p. 340, Nov. 2017.

[10] S. Y. Cho and C. G. Park, "Threshold-less zero-velocity detection algorithm for pedestrian dead reckoning," in *Proc. Eur. Navigat. Conf. (ENC)*, Apr. 2019, pp. 1–5.
[11] T. Feigl, S. Kram, P. Woller, R. H. Siddiqui, M. Philippsen, and C. Mutschler, "A bidirectional LSTM for estimating dynamic human velocities from a single IMU," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2019, pp. 1–8.
[12] B. Wagstaff and J. Kelly, "LSTM-based zero-velocity detection for robust inertial navigation," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Nantes, France, Sep. 2018, pp. 24–27.
[13] B. Wagstaff, V. Peretroukhin, and J. Kelly, "Robust data-driven zero-velocity detection for foot-mounted inertial navigation," *IEEE Sensors J.*, vol. 20, no. 2, pp. 957–967, Jan. 2020.
[14] J. Le Scornec, M. Ortiz, and V. Renaudin, "Foot-mounted pedestrian navigation reference with tightly coupled GNSS carrier phases, inertial and magnetic data," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*. Sapporo, Japan: IEEE, Sep. 2017, pp. 1–8.
[15] K. Abdulrahim, C. Hide, T. Moore, and C. Hill, "Integrating low cost IMU with building heading in indoor pedestrian navigation," *J. Global Positioning Syst.*, vol. 10, no. 1, pp. 30–38, Jun. 2011.
[16] X. Meng, S. Sun, L. Ji, J. Wu, and W.-C. Wong, "Estimation of center of mass displacement based on gait analysis," in *Proc. Int. Conf. Body Sensor Netw.*, May 2011, pp. 150–155.
[17] R. Feliz Alonso, E. Zalama Casanova, and J. Gómez García-Bermejo, "Pedestrian tracking using inertial sensors," *J. Phys. Agents*, vol. 3, no. 1, pp. 35–43, Jan. 2009.
[18] H. Fourati, "Heterogeneous data fusion algorithm for pedestrian navigation via foot-mounted inertial measurement unit and complementary filter," *IEEE Trans. Instrum. Meas.*, vol. 64, no. 1, pp. 221–229, Jan. 2015.
[19] M. Ma, Q. Song, Y. Gu, Y. Li, and Z. Zhou, "An adaptive zero velocity detection algorithm based on multi-sensor fusion for a pedestrian navigation system," *Sensors*, vol. 18, no. 10, p. 3261, Sep. 2018.
[20] I. Skog, J.-O. Nilsson, and P. Handel, "Evaluation of zero-velocity detectors for foot-mounted inertial navigation systems," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat.*, Sep. 2010, pp. 1–6.
[21] S. Y. Park, H. Ju, and C. G. Park, "Stance phase detection of multiple actions for military drill using foot-mounted IMU," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Alcalá de Henares, Spain, Oct. 2016.
[22] B. Wagstaff, V. Peretroukhin, and J. Kelly, "Improving foot-mounted inertial navigation through real-time motion classification," in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1–8.
[23] V. Renaudin. (2018). *ULISS and PERSY: Research Equipment for the Positioning and Navigation of Travelers*. [Online]. Available: https://www.geoloc.ifsttar.fr/fileadmin/redaction/1_institut/1.20_sites_integres/COSYS/GEOLOC/plaquette_equipement/UlissPersy-plaquette-en.pdf
[24] J. Perry and J. Burnfield, *Gait Analysis: Normal and Pathological Function*, 2nd ed. Thorofare, NJ, USA: SLACK Incorporated, 2010.
[25] L. Bao and S. S. Intille, *Activity Recognition From User-Annotated Acceleration Data*, vol. 3001. Berlin, Germany: Springer, 2004.
[26] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge, U.K.: Cambridge Univ. Press, 2014.
[27] A. Mikolajczyk and M. Grochowski, "Data augmentation for improving deep learning in image classification problem," in *Proc. Int. Interdiscipl. PhD Workshop (IIPhDW)*, Swinoujscie, Poland: IEEE, May 2018, pp. 117–122.
[28] T. Dietterich, "Overfitting and undercomputing in machine learning," *ACM Comput. Surv.*, vol. 27, no. 3, pp. 326–327, Sep. 1995.
[29] H. K. Jabbar and R. Z. Khan, "Methods to avoid over-fitting and under-fitting in supervised machine learning (comparative study)," in *Proc. Comput. Sci., Commun. Instrum. Devices*, 2014, pp. 163–172.
[30] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Proc. Int. Joint Conf. AI*, Aug. 1995, vol. 14, no. 2, pp. 1137–1145.
[31] M. N. Murty and V. S. Devi, *Introduction to Pattern Recognition and Machine Learning*, vol. 5. Singapore: IISc Press/World Scientific, 2015.
[32] A. Guryanov, "Histogram-based algorithm for building gradient boosting ensembles of piecewise linear decision trees," in *Analysis of Images, Social Networks and Texts*. Cham, Switzerland: Springer, 2019, pp. 39–50.
[33] N. Buduma and N. Locascio, *Fundamentals of Deep Learning*, 1st ed. Newton, MA, USA: O'Reilly Media, Jun. 2017.
[34] J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl, "Algorithms for Hyper-Parameter Optimization," in *Advances in Neural Information Processing Systems*, vol. 24, J. Shawe-Taylor, R. S. Zemel, P. L. Bartlett, F. Pereira, and K. Q. Weinberger, Eds. Red Hook, NY, USA: Curran Associates, 2011, pp. 2546–2554.
[35] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.
[36] C. Olah. (2015). *Understanding LSTM Networks*. [Online]. Available: http://colah.github.io/posts/2015-08-Understanding-LSTMs/
[37] J. H. Friedman, "Stochastic gradient boosting," *Comput. Statist. Data Anal.*, vol. 38, no. 4, pp. 367–378, Feb. 2002.
[38] G. Ke *et al.*, "Lightgbm: A highly efficient gradient boosting decision tree," in *Advances in Neural Information Processing Systems*, vol. 30. Red Hook, NY, USA: Curran Associates, 2017, pp. 3146–3154.

Yacouba Kone received the M.Sc. degree in data science statistical modeling from University Bretagne Sud in 2019. He worked as a Consulting Data Analyst/Manager at World Bank, Burkina Faso in 2017, on the project result-based financing (RBF). He is a Research Engineer with the GEOLOC Laboratory, University Gustave Eiffel.

Ni Zhu received the Engineering degree in aeronautic telecommunications from the École Nationale de l'Aviation Civile (ENAC) in 2015 and the Ph.D. degree in science of information and communication from the University of Lille in 2018. She is a Permanent Researcher with the GEOLOC Laboratory, University Gustave Eiffel. Her current research interests include GNSS channel propagation modeling in urban environments, integrity monitoring for terrestrial vehicles, and multisensory fusion techniques for indoor/outdoor pedestrian positioning.

Valerie Renaudin (Member, IEEE) received the M.Sc. degree in geomatics engineering from ESGT in 1999 and the Ph.D. degree in computer, communication, and information sciences from EPFL in 2009. She was a Technical Director at Swissat Company, Switzerland, developing real-time geopositioning solutions based on a permanent global navigation satellite system (GNSS) network, and a Senior Research Associate with the PLAN Group, University of Calgary, Canada. She is currently leading the GEOLOC Laboratory and the team dedicated to positioning and navigation for travelers in multimodal transport. She is a Research Director with University Gustave Eiffel. Her research interests include outdoor/indoor navigation using GNSS, and inertial and magnetic data, particularly for pedestrians. She has been a member of the IEEE Society since 2013 and the Steering Committee of the Indoor Positioning and Indoor Navigation International Conference. She received the European Marie Curie Career Integration Grant.

Miguel Ortiz received the M.Sc. degree in mechanics, automation, and engineering from École Nationale Supérieure d'Arts et Métier in 2001. He is currently a Research Engineer with the GEOLOC Laboratory, University Gustave Eiffel. He joined the laboratory after six years spent in a company, where he managed systems architecture for automotive applications. He was an expert in embedded electronic systems, his scientific interests focus on software and hardware developments for both intelligent transport systems (ITS) and pedestrian navigation research field. Since 2017, he has been the Convener of CEN/CENELEC TC5-WG1 named navigation and positioning receivers for road applications.