

# BoostSole: Design and Realization of a Smart Insole for Automatic Human Gait Classification

Badis Djamaa  
Computer Science Department  
Ecole Militaire Polytechnique  
Algiers, Algeria  
badis.djamaa@gmail.com

Med Messaoud Bessa  
Computer Science Department  
Ecole Militaire Polytechnique  
Algiers, Algeria  
messaoubessa@gmail.com

Badreddine Diaf  
Computer Science Department  
Ecole Militaire Polytechnique  
Algiers, Algeria  
diafbadreddine1@gmail.com

Abdenebi Rouigeb  
Computer Science Department  
Ecole Militaire Polytechnique  
Algiers, Algeria  
rouigeb.abdenebi@gmail.com

Ali Yachir  
Computer Science Department  
Ecole Militaire Polytechnique  
Algiers, Algeria  
ali.yachir@gmail.com

**Abstract**—This paper presents BoostSole; a smart insole based system for automatic human gait recognition. It consists of a smart instrumented insole connected to the cloud via the patient’s smartphone using low-power wireless communication. First, the design of BoostSole is introduced with discussions of sensors choice, placement, calibration, and data communication. Next, an adaptive multi-boost classification algorithm is deployed to accurately identify different gait patterns. The algorithm is fast and lightweight and can be implemented in ordinary smartphones with a small footprint in terms of computational requirements, energy consumption, and communication usage. Raw and on-device classified data can be securely uploaded to a distant cloud server for continuous monitoring and analysis. Indeed, they can be visualized and exploited by doctors to identify/correct walking habits and assess the risks of chronic pain associated with an abnormal walk. The system has been evaluated on a dataset containing three gait patterns, namely: shuffle walk; toe walking; and normal gait. Obtained results are promising with more than 97% classification accuracy accompanied by low response time and computational demands.

**Index Terms**—Smart insole, Human Gait Analysis, Force Sensing Resistors, MultiBoost Classification, Internet of Things.

## I. INTRODUCTION

**W**ALKING is a fundamental movement of the human body, which has a direct impact on its health. Indeed, a simple abnormality in walking can cause serious health problems that can range from simple pain to the loss of the walking ability. This is why gait analysis is very important for assessing human health. Indeed, such an analysis allows the evaluation and diagnosis of walk abnormalities before medical interventions. It also makes it possible to monitor surgical procedures and rehabilitation of patients from interventions that can affect their ability to stand or walk.

In the past, gait analysis was conducted using subjective methods, which are essentially based on the observations of specialists under clinical conditions. Indeed, the various parameters related to a patient’s gait are observed, noted, and evaluated by the specialist while he is walking on a prede-

termined circuit. Now, advances in new technologies have given rise to devices and techniques allowing an objective, automatic, and fast assessment of different gait parameters. Thus, allowing more effective measurement and providing specialists with a large amount of reliable information on patients’ gaits. This reduces the cost and the margin of error caused by subjective techniques.

Such technological devices can be classified into two different approaches: those based on Non-Wearable Sensors (NWS); and those relying on Wearable Sensors (WS). NWS systems, generally based on image processing and ground sensors, require the use of controlled stations where the sensors are located and capture walking data while the subject is moving on a marked walkway. Their main advantage is in liberating the subject from any constraints, but they are too expensive and might not capture real-world gait characteristics. On the other hand, WS systems make it possible to analyze data outside the laboratory and to capture information on human walks during their daily activities. Indeed, such sensors can be placed on different parts of the patient’s body, such as the feet, knees, or hips, to measure relevant gait characteristics. However, WS may be constraining as they must be worn by the subject.

From the WS class, footwear systems stand as a non-obstructive method that addresses most of the issues related to wearable systems while preserving their advantages. For instance, footwear systems only require to instrument the insole/shoe with non-obstructive invisible sensors such as flexion sensitive sensors, force-sensitive sensors, and inertial measurement units [1]. The patient will wear the instrumented shoe/insole similarly to an ordinary one. Moreover, footwear systems are generally more accurate and lower cost when compared with other WS and NWS systems. Furthermore, they can serve other needs such as preventing foot ulcers in diabetics and detecting falls in the elderly.

In this work, a prototype of a smart insole based system for human gait recognition and classification, dubbed

BoostSole, is developed. BoostSole comprises a low-cost, low-power instrumented insole that continuously acquires gait data and transfers it, via low-power wireless communication, to the patient's smartphone for on-device analysis before being reported to the physician for decision making. The aim is to provide a low-cost, reliable, and time-efficient decision support system for physicians to identify and classify human gait in real-world scenarios in a non-obstructive way. More particularly, this paper provides the following contributions:

- Design, conception, and realization of BoostSole along with sensors choice, placement, and calibration.
- Development of BoostSole smartphone and desktop software applications as well as data acquisition, processing, and decision support processes.
- Extensive performance evaluations of BoostSole with a multitude of machine learning algorithms to classify three gait types under different performance metrics including accuracy, time efficiency, and lightweight aspects.

The remainder of this paper is organized as follows. Section II presents and discusses related work. The architecture of the BoostSole system is presented in Section III, while the design, choice, placement, and calibration of sensors are the object of Section IV. Section V is devoted to detailing the communication, feature extraction, and software components of the BoostSole system. This is followed by extensive performance evaluations of BoostSole for classifying three gait types (shuffle, toe, and normal) using a multitude of machine learning algorithms in Section VI. The paper ends in Section VII with conclusions and ideas for future directions.

## II. RELATED WORK

In the last few years, many research works have used footwear sensors for human gait analysis. [2]–[7] are examples of such research. Overall, there is a big similarity in the type of sensors used in these works, with some exceptions in the number and the placement of the sensors. Differences, mainly, reside in the artificial intelligence algorithms used to classify human gait for identification, activity recognition, and/or injury/fall detection and prevention.

For instance, [8] uses hidden Markov chains to detect the phases of the human's gait, [9] used Support Vector Machine (SVM) techniques to classify three types of walks, and [10] analyzed their data using Principal Component Analysis (PCA). Besides, [2] also used PCA to analyze their data and classify three types of walking: normal, toe, and dragging foot walking. PCA results showed a similarity between dragging foot walking and normal walking.

Recently, the authors of [11] used the AdaBoost tree classifier for gait asymmetry detection with smart insole attaining an accuracy of 89.9%. On the other hand, [12] used Deep Convolution Neural Network (DCNN) to classify seven (07) types of gait: walking, fast walking, running, stair climbing, stair descending, hill climbing, and hill descending with an accuracy of more than 90%. Finally, in [13], the authors used a commercial "FootLogger" smart insole to classify seven (07) types of gait with Null-Space Linear Discriminant Analysis

(NLDA), and they found that the larger the number of steps of a sample, the higher the classification performance becomes. Table I summarizes some of those work with a focus on the classification method used and walking phases detected. Besides, the table also presents the type of used sensors along with the hardware and software costs of the considered works.

Different from the above, the presented system is designed to be low-cost, lightweight, resource-lean, and time-efficient while providing reasonable accuracy in abnormal walk identification and classification. In the sequel, we will discuss the system architecture, the sensors used, and their placement.

## III. SYSTEM ARCHITECTURE

The architecture of our system is depicted in Fig. 1, which is made of five main components, namely: (1) the low-cost instrumented sole; (2) patient's smartphone; (3) physician's working station; along with (4) local and (5) remote communication bridges.



Fig. 1: System architecture

The first component is the main element of the BoostSole system. It consists of a smart insole equipped with a multitude of miniaturized sensors of force, flexion, and IMUs continuously collecting gait characteristics. Indeed, such sensors can measure many parameters that characterize walking, such as the timing of the heel strike and detachment of the foot, dorsi/plantar flexion, step length, and walking speed.

This data will be transmitted to patients' smartphones via low-power wireless communications where it will be processed and analyzed with lightweight on-device machine learning and visualized via mobile applications which can be used anywhere and at any time. Raw and on-device processed data can be then transmitted to treating physicians via secure remote communications as can be seen from Fig. 1. The remote application (component 3 in Fig. 1) can visualize and analyze the gait further so to help the physician decide by comparing the obtained results with a reference.

TABLE I: Related work summary and paper contributions

Ref.	Walking phases/types	Classification method	FSR	IMU	Flex	Hardware cost	Software cost	Observation
[12]	- Walking - Fast walking. - Running. - Stairs Ascending/Descending. - Hill Climbing/Descending	DCNN	Yes	Yes	No	High	High	Used "FootLogger" smart insole with a classification accuracy of more than 90%.
[14]	- Walking. - Sidestepping. - Jumping. - Kicking. - Squatting.	DNN	Yes	Yes	No	High	Medium	A system able to predict the movement of the lower body.
[15]	- Walking. - Running. - Stairs Ascending/Descending.	SVM	Yes	Yes	No	High	High	An accuracy of 99.8% in the recognition of daily living activities.
[13]	- Heel strike. - Foot flat. - Mid stance. - Heel off - Toe off. - Mid swing. - Late swing.	NLDA	Yes	Yes	No	High	High	Used "Footlogger" smart insole to classify seven types of the gait cycle.
[16]	- Heel strike. - Stance. - Heel off. - Swing.	None	Yes	Yes	No	High	Low	"FootMov" system detects gait phases using a developed algorithm.
[2]	- Normal walking. - Tip-toe walking. - Dragging foot walking.	PCA	Yes	No	Yes	High	Low	Can classify three types of walks using ZigBee for communication and PCA for recognition.
This work	- Normal walking. - Tip-toe walking. - Shuffle walk.	MultiBoostAB with Random Forest	Yes	Yes	Yes	Low	Low	Can classify three types of walks with a lightweight algorithm that can be implemented in a smartphone.

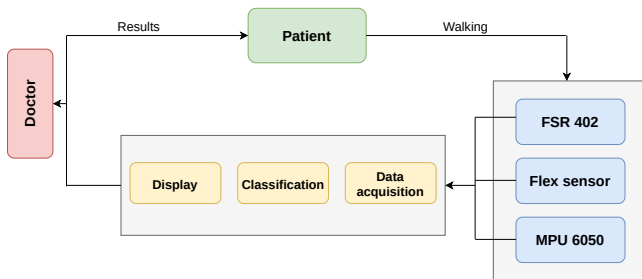


Fig. 2: System's components and processes.

The processes and functionalities realized by the main components of our architecture are depicted in Fig. 2. Thus, the smart insole acquires and transmits data to local patient's gadgets, while the smartphone and/or the desktop application is/are responsible for pre-processing, classification, and visualization by the patient and/or the physician.

#### IV. DESIGN AND REALISATION OF BOOSTSOLE

The first step in designing BoostSole was to choose the appropriate sensors, to create a highly instrumented low-cost sole capable of reliability detecting gait parameters. The following subsections detail the choice, role, and number of sensors along with their locations and calibrations.

##### A. Choice, role, and number of sensors

BoostSole relies on several sensors to capture fundamental human gait parameters. In our design, the focus is on using the minimum number of sensors that allows identifying human gait. The main used sensors are described below.

1) *Force sensor*: Force sensor or Force Sensing Resistor (FSR), is a robust device made of thick polymer film, which exhibits a decrease in resistance with the increase in the force applied to the sensor surface. This force sensitivity is optimized for use in human touch control of electronic devices such as automotive electronics, medical systems, and in industrial and robotic applications [4]–[7], [17].

We chose to use FSR 402, shown in Fig. 3a), because of its very miniature size (0.45mm), its simplicity, and ease of integration. Its robustness is up to 10 million actuation with a low activation force of 0.1N and a sensitivity of up to 10N. Besides, it is low-cost, ergonomic, and fits well to measure the pressure applied to the bottom of the foot. Furthermore, the combination of several FSRs can be applied to find the center of force beneath the foot. The number of FSRs depends on the accuracy required by the application and the cost of the developed prototype.

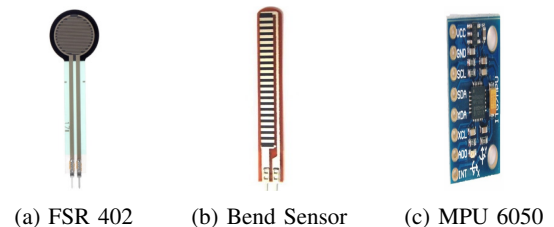


Fig. 3: Used sensors

2) *Bend sensor*: A bend or flex sensor, shown in Fig. 3b), is a sensor that measures the bending angles. The resistance of the elements of the sensor increases by bending. The more the sensor is bent, the more it tends towards an infinite resistance (open circuit). Since the resistance is directly proportional

to the curvature, it is used as a goniometer and is often called a flexible potentiometer. It is also used in a multitude of other domains including, rehabilitation, physical activities, machines, measuring tools.

The substrate of the bending sensor, which is produced from ink, carbon, or graphite [18] plays an important role in its performance. In our system, the bend sensor is used to measure the flexion angles below the foot depending on gait type and phases. We have used a flex sensor of 2.2" height, which can give values between 45 and 15 KOhms depending on the curvature radius, which is enough for BoostSole.

3) *Inertial measurement unit*: Inertial measurement units, generally, comprise an accelerometer and a gyroscope and can be attached to a mobile or any other object. The accelerometer can measure the linear acceleration along one or 3 orthogonal axes. On the other hand, when one seeks to detect a rotation or angular speed, the gyroscope is used. These sensors are pervasively used in a multitude of applications including games, gesture recognition, location-based services, movement-based game controllers, 3D remote controls for digital TVs, and portable sensors for health, fitness, and sports.

In this project, we used an MPU 6050 module (Fig. 3c), which combines a 3-axis gyroscope and a high-precision 3-axis accelerometer to form an inertial unit calculating acceleration and angular speeds of a human gait. Table II summarises the main characteristics of each sensor, their number, and their unit prices in the market. It can be observed from this table that the realization of a gait analysis support system can be low-cost compared to its usefulness and its reliability.

### B. Sensors' placements and BoostSole prototype

This section details the sensors' placements and presents the realized prototype.

1) *Sensors' placements*: Once the choice of sensors is made, the emphasis is on choosing the right locations to place them beneath the foot. In this prototype, presented in Fig. 4, the 03 force sensors are placed under the toe, between the toe and the middle of the foot, and under the heel to capture the movements made by the patient. This is justified by the fact that a human being when walks, his weight is generally distributed on three essential points on the foot. These points are the toe, the heel, and the place between the middle and the toe [19]. The flex sensor is placed in the middle of the foot to calculate angles. Remains, the last sensor, which is MPU 6050. This latter is fixed behind the foot to capture the translations with the angular velocities during feet movements. It is put in the microcontroller unit detailed below.

2) *Wiring BoostSole*: Based on these locations and the sensors seen in the previous sections, we created the first prototype of BoostSole, illustrated in Fig. 5, by wiring them to a microcontroller unit. The microcontroller brings together the essential elements necessary for wiring and reading sensor data such as micro-controller, memory, peripheral units, and input interfaces. The realization of the prototype was made using Arduino UNO; a well-known low-cost system-on-chip.

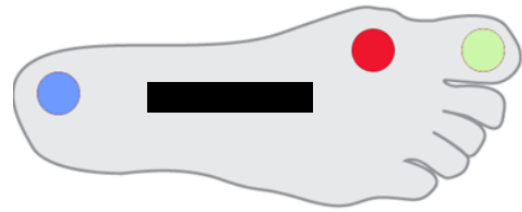


Fig. 4: Sensors' locations

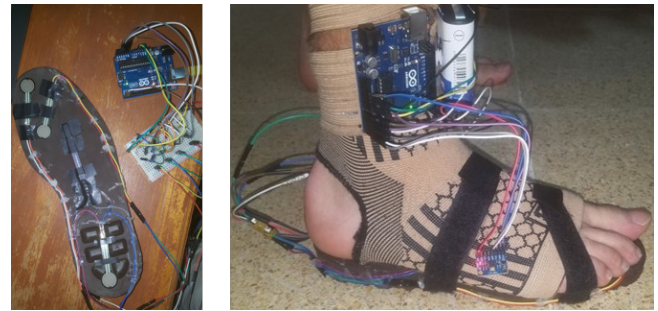


Fig. 5: The first prototype.

### C. Sensors' calibrations

To be correctly used, the sensors must be calibrated. This section explains how we calibrated the flex sensor, FSRs, and MPU 6050 to obtain correct values.

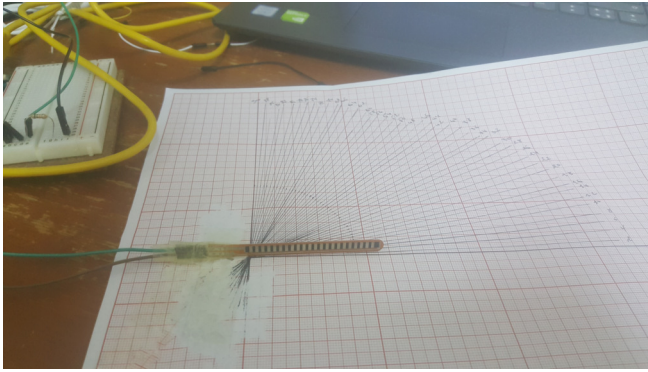
For the flex sensor, we proceed as follows. First, we draw a semi-circle on a paper and draw angles from  $0^\circ$  to  $180^\circ$  by a step of  $2^\circ$  as can be seen from Fig. 6a. Next, we have interfaced the flex sensor with Arduino and fixed it in the prepared paper. Then, we bend the sensor at each angle and note the value given in the Arduino IDE. This experiment has been repeated multiple times. Finally, we draw a graph representing the table containing values obtained from Arduino and the values of the real angles. Fig. 6a presents the flex sensor calibration process, and Fig. 6b presents a portion of the calibration data plotted in a graph. The values represented in this figure are for angles from  $0^\circ$  to  $20^\circ$ . As can be seen from this figure, flex values show a linear relationship with measured angles that can reliably be exploited for gait analysis.

For FSR calibration, we proceeded similarly to the flex sensor, but in this case, we applied different weights and read the FSR values on Arduino IDE. After that, we drew a table similar to that of the flex. The details of this calibration process are alike those of [20]. Obtained results are in a concordance with the conclusions of [20] and show a linear relationship between FSR data and weights up to a value of 80Kgs.

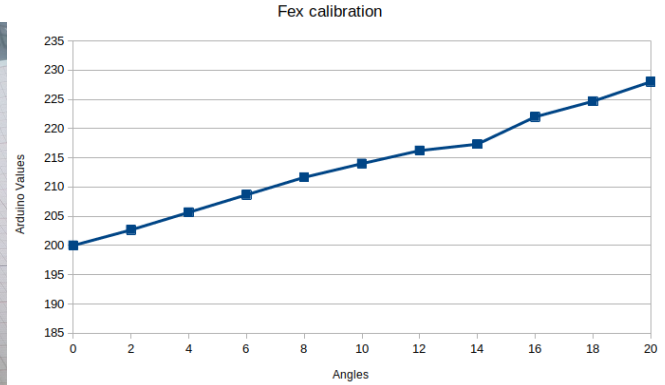
Finally, for the calibration of the MPU 6050, we calculated the average of the first 1000 values and then subtract this value from the values read by Arduino. these relative values were used. It should be noted, however, that the values of the MPU 6050 diverge quickly, so we need to re-calibrate the sensor periodically to get correct measurements.

TABLE II: Sensors' characteristics

	IMU	FSR	Flex sensor
Trade name	MPU 6050	FSR 402	FS
Number	01	03	01
Price	\$ 5.98	\$ 8.67	\$ 12.30
Uses	- Angles between the two legs.	- To calculate pressure under the insole.	- To calculate angle between the insole and the ground.
Dimensions (mm)	15.6*20.3*2.5	18.28*56.33*1.25	6.35*112.24*0.43
Life cycle	12 months	10 millions values	>1 million values
Temp range	-40°C to +105°C	-30 - +70 °C	-35°C to +80°C



(a) Process



(b) Results

Fig. 6: Flex sensor calibration.

## V. DATA COMMUNICATION, CLASSIFICATION AND VISUALIZATION

Once the prototype is completed, we focused on the other parts of the architecture, namely: sending data to a processing station, classifying movements, and displaying the results.

### A. Data communication

Communication plays an important role in the design of a smart insole. Indeed, besides being the key component in ensuring reliable transmissions of gait data from the embedded microcontroller to the processing station, it is crucial to the system's energy consumption and hence on its lifetime.

Today, a multitude of wireless communication technologies exist in the market. Each has its applications, advantages, and drawbacks as can be seen from Table III. Thus, while WiFi-based solutions are very pervasive, they consume much energy making their lifetime in hours, which does not fulfill the requirements of boostSole. On the other hand, IEEE 802.15.4 solutions provide better energy consumption that fulfills the requirements of BoostSole, but they are not pervasive and are not available in ordinary smartphones/PCs, which limit their applicability. Bluetooth Low-Energy (BLE) has the advantages of both, making it an important candidate for the BoostSole prototype. To do so, we have chosen the HM-10 BLE module, which implements Bluetooth 4.1 specification. Indeed, it provides reliable communication by channel hopping to avoid interference with co-existing networks, along with high throughput for capturing sensor data. Furthermore, the module goes into sleep automatically when no data activity is detected. Besides, it can be integrated into Arduino via a

serial link. Finally, it should be noted that the pairing between the processing station and BoostSole is initiated by the station allowing the insole to start sending gait data just after pairing.

TABLE III: Low-power wireless communication technologies

	Bluetooth	WiFi	ZigBee	BLE	Z-Wave
Standard	802.15.1	802.11n	802.15.4	802.15.1	G.9959
Frequency	2.4 GHz	2.4/5 GHz	2.4 GHz	2.4 GHz	868 MHz
Topology	star	star	star, mesh	star, mesh	mesh
Data rate	2 mbps	100 mbps	250 kbps	1 mbps	40 Kbps
Range	15-30 m	10-100 m	10-100 m	15-30 m	30-100 m
Battery	Months	Days	Years	Years	years
Pervasive.	Yes	Yes	No	Yes	No

### B. Feature extraction and classification

The raw data captured by the sole are transmitted via BLE to the patient's Smartphone for on-device gait analysis. Before being processed, the acquired data will be segmented. Fig. 7 details the feature extraction process. As can be seen from this figure, the input signals  $S_1, \dots, S_n$  are respectively discretized into  $Y_1, \dots, Y_n$  sequences, where each  $Y_i(j)$  represents the mean of the  $j^{th}$  interval of  $S_i$  (Fig. 7). Such sequences are, then, segmented with a sliding window procedure, where a fixed-length window  $W$  is shifted along the signal sequence for frame extraction. Consecutive frames usually overlap to some degree (less than 50%). In the end, a set of vectors of size  $n * |W|$  are generated.

For classification, we deploy a supervised learning approach. Thus, the generated vectors along with the labels provided by experts are fed to a supervised machine learning algorithm for training as can be seen from Fig. 8. In order to

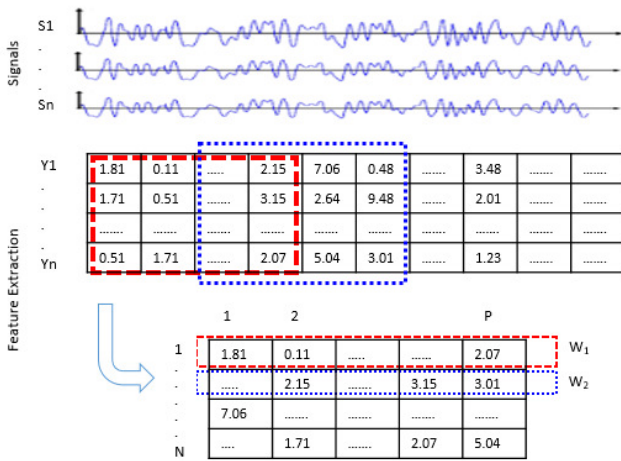


Fig. 7: Feature extraction

select the best classifiers, a number of well-known algorithms including SVM, kNN, decision trees, and ensemble classifiers will be evaluated in terms of accuracy, time, and complexity.

The chosen classifiers are known to be powerful with a high capacity of generalization. For instance, SVM belongs to the kernel-based family which aims to fit an optimal hyper-plane to accurately classify both linearly separable and linearly inseparable data [21]. kNN is a non-parametric method used for classification and regression. Boosting classifiers such as Bagging, Boosting, AdaBoost, and MultiBoostAB are a type of meta-algorithms that use decision trees and discriminant analysis learners to improve the classification. Their main idea is to boost weak classifiers. Multi-boosting [22] is a representative sophisticated algorithm of this class. It is an extension to the AdaBoost with Wagging.

The trained models will be used to classify feature vectors extracted from a given test signal. Then, a majority vote can be performed to predict the gait class of the signal (Fig. 8).

### C. BoostSole software application

We developed both a desktop and an android application.

1) *Desktop application:* The desktop application is developed using JavaFX. Before it shows up, the application must first connect with the insole. After that, the user can see a graphical user interface that contains three (03) empty charts, one is for the three FSR sensors, the second is for the flex sensor, and the last one (at the bottom) is for MPU 6050. At the right, there is a start button to choose the walking period (30 seconds, 1, 2, 5, and 10 minutes). When the user clicks on that button, the signals acquired from each sensor are visualized in their corresponding places and stored in a specified path as can be seen in Fig. 9. At the end of the walking period, the application stops plotting the data and the classification results can be displayed.

2) *Android application:* The Android application is developed using Android Studio. It allows a user to connect with the BootSole using BLE, and visualize the pressure sensors data in

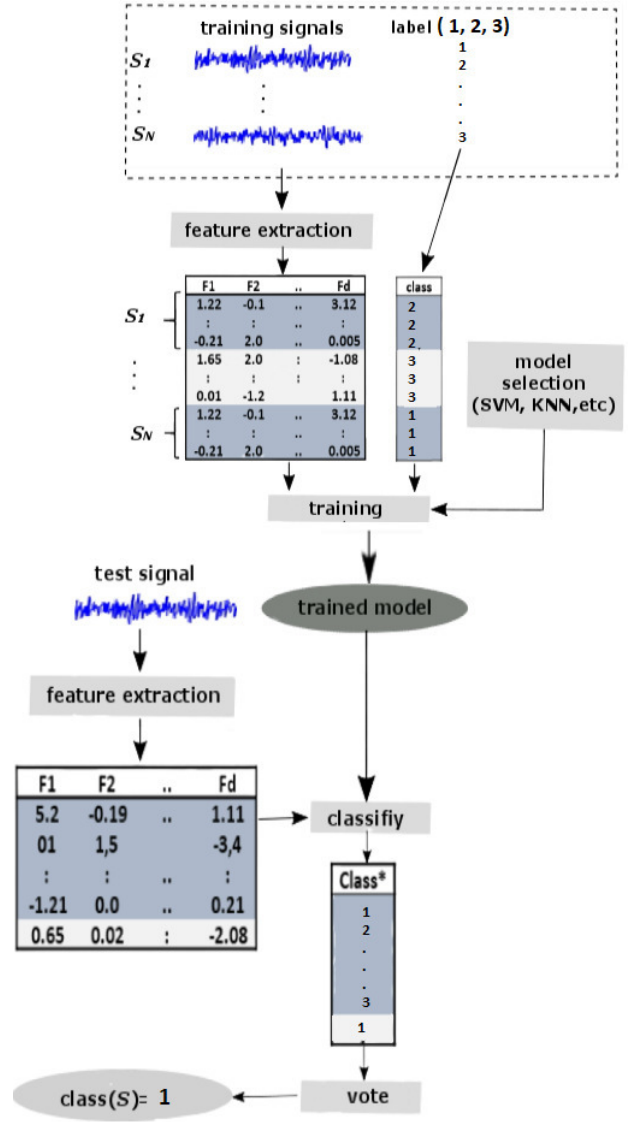


Fig. 8: Classification process

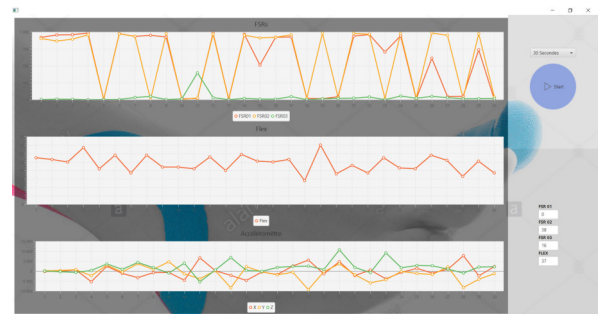


Fig. 9: The desktop application

a Heat-Map. We used three Android activities: the first starts the communication, the second visualizes the pressure map, and the third analyzes and displays gait recognition results.

## VI. PERFORMANCE EVALUATION

This section evaluates the performance of the proposed system. We start by describing the dataset, methodology, and metrics before discussing the obtained results.

### A. Experimental dataset

To access the system, we collected data from 5 healthy volunteers (age [y]:  $23.5 \pm 1.3$ ; height [m]:  $1.77 \pm 0.08$ , weight [kg]:  $78 \pm 5$ ). All walking sequences were tracked using BoostSole. A single sensing unit was captured with a  $1Hz$  sampling rate in order to save energy. All the recorded data was sent via BLE to a laptop placed in close proximity to the participant. The volunteers performed a continuous sequence of three walking types, namely: shuffle walking (class 1); normal walking (class 2); and toe walking (class 3). For each one, the volunteers walked for 30s, which make it a 90s total. Fig. 10 presents a screenshot for a representative gait data collected for each type by the desktop application.

### B. Evaluation methodology and metrics

In our evaluations, six (06) classifiers were considered, namely SVM, kNN ( $k = 5$ ), Stacked, Random Forest (RF), MultiBoostAB with RF (MB-RF), and MultiBoostAB with Logistic Model Tree (MB-LMT). The 5-fold cross-validation method is followed to evaluate the accuracy of the aforementioned classifiers under different window lengths (from 1s to 10s). Also, in classification, we have only used the data collected from FSRs and bend sensors to assess their ability to distinguish walk patterns. All main results were obtained using an i7-8750H @2.20GHz, 16 Go RAM, and a GTX-1050 4Go GPU. Average accuracy, precision, recall, F-measures, and Receiver Operating Characteristic Curve (ROC) Areas were used to evaluate the effectiveness of the involved classifiers. They are measured as follows:

- *Accuracy*: the ratio of number of correct predictions to the total number of input samples.

$$Accuracy = \sum_c \frac{TP_c + TN_c}{TP_c + TN_c + FP_c + FN_c}, c \in classes \quad (1)$$

- *Precision*: An average per-class agreement of the data class labels with those of a classifiers.

$$Precision = \sum_c \frac{TP_c}{TP_c + FP_c}, c \in classes \quad (2)$$

- *Recall*: Average per-class effectiveness of a classifier to identify class labels.

$$Recall = \sum_c \frac{TP_c}{TP_c + FN_c}, c \in classes \quad (3)$$

- *F-Measure*: The harmonic mean of the macro-average precision and recall.

$$F-Measure = \frac{2 * Precision * Recall}{Precision + Recall} \quad (4)$$

- *Receiver Operating Characteristic Curve Areas (ROC areas)*: It represents the area below the plot of the

true positive rate against the false positive rate. It shows the trade-off between sensitivity and specificity.

Furthermore, the best models were deployed on an android device (Samsung S7-Edge) in order to evaluate their average response time, memory, CPU, and battery usage.

### C. Results and discussions

1) *Comparison of classifiers*: This experiment aims to find a suitable classifier and window length for the BootSole system. Table IV shows the average accuracy for different classifiers under different window lengths. Overall, the accuracy of all algorithms increases with increasing window length up to a window of 8s. When it comes to individual classifiers, RF and MB-RF have achieved better accuracy compared to SVM, kNN, MB-LMT, and stacked classifiers. Indeed, average accuracies of above 95% were observed as soon as a window of 3s for both RF and MB-RF. Besides, MB-RF achieved an accuracy of about 97% in a 4s window length. Furthermore, this classifier reached almost 100% accuracy for 7s window length. For the sake of time, energy, and computational resources, a 4s window is used.

In addition to the accuracy results, MB-RF has shown the smallest test response time, which makes it very promising for classifying walking types using BootSole. Before embedding it in the Android application, we will get a closer look at MB-RF in the following section.

2) *A detailed evaluation of MB-RF*: Table V shows different evaluation metrics obtained for the MB-RF classifier. On average, a precision value of around 0.969 has been recorded, which allows MB-RF to predict correctly the positive observations to the total predicted positive observations. A similar value has also been registered for recall, allowing MB-RF to classify positive observations w.r.t. the observations in the actual class with 96.8%. Besides, the confusion matrix, given in Table VI, shows that the toe and shuffle classes are well discriminated, whereas, signals from the normal walk are slightly hard to be correctly classified. To confirm such results, we have drawn ROC curves and assessed the area under ROC. All the curves start on the left-hand border and then follow the top border of the ROC space, which justifies the results presented in Table V. Indeed, MB-RF recorded a 0.989 average ROC area (0.988 for the shuffle, 0.982 for normal, and 0.995 for toe walking) in a 4s window length.

3) *MB-RF resource consumption on Android*: By giving the best results, MB-RF is a promising classifier for BoostSole. However, before embedding it in handheld devices, its resource consumption needs to be examined. To do so, we have conducted a new battery of tests on a smartphone. To put results into context, we have compared MB-RF with the two following best classifiers that shown accuracy around 95% in the 4s window length (SVM and RF). The three models (trained with the Weka software) were deployed on an android device (Samsung S7-Edge). The average response time along with memory, CPU, and battery usages are reported.

Table VII presents the Average Response Time (ART) and the memory usage of the three algorithms. It is clear from this

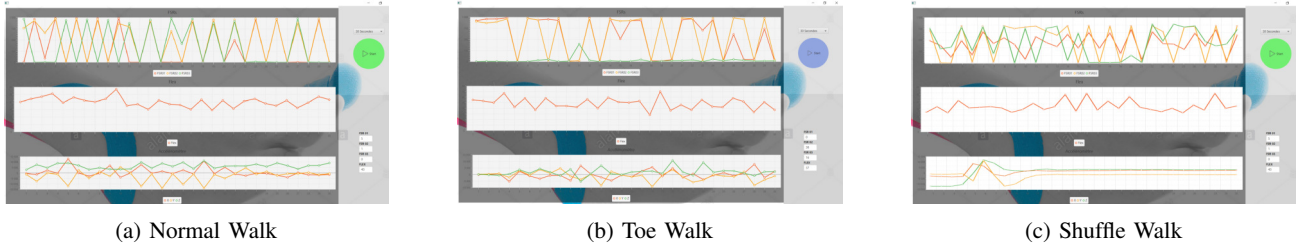


Fig. 10: Data visualisation

TABLE IV: Average accuracy of different classification algorithms using different window lengths

Win. (s)\classifiers	SVM	RF	MB-RF	MB-LMT	Stacked	kNN
<b>1</b>	72.16%	79.06%	78.62%	75.28%	64.14%	69.47%
<b>2</b>	90.40%	91.74%	91.74%	90.63%	83.89%	87.45%
<b>3</b>	93.30%	<b>95.09%</b>	<b>95.54%</b>	91.07%	91.51%	87.91%
<b>4</b>	94.62%	<b>96.41%</b>	<b>96.86%</b>	93.27%	89.26%	87.84%
<b>5</b>	93.96%	91.28%	91.28%	89.93%	89.91%	85.82%
<b>6</b>	93.92%	94.59%	95.27%	93.24%	88.39%	85.57%
<b>7</b>	<b>97.30%</b>	<b>100%</b>	<b>100%</b>	93.69%	91.21%	84.62%
<b>8</b>	<b>98.20%</b>	<b>100%</b>	99.10%	95.50%	89.67%	87.45%
<b>9</b>	97.75%	94.38%	98.88%	94.38%	87.39%	86.27%
<b>10</b>	97.73%	97.73%	96.59%	96.59%	67.71%	80.86%

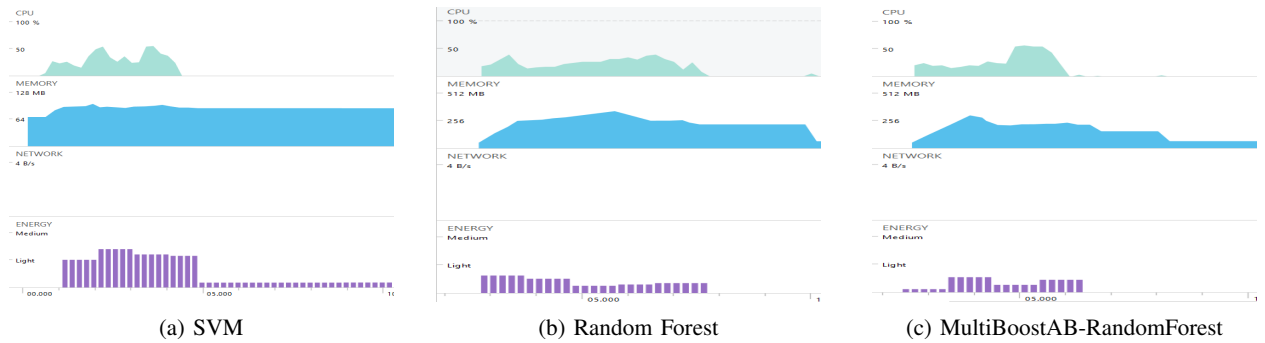


Fig. 11: Classifiers resource consumption in Android

TABLE V: Metrics of MultiBoostAB-RF for 4s window length

Class	Precision	Recall	F-Measure	ROC Area
<b>Shuffle Walking</b>	0.961	0.987	0.974	0.996
<b>Normal Walking</b>	0.986	0.920	0.952	0.988
<b>Toe Walking</b>	0.961	1.000	0.980	0.998
<b>Weighted Avg.</b>	0.969	0.969	0.968	0.994

TABLE VI: Confusion matrix

Class	Shuffle Walking	Normal Walking	Toe Walking
<b>Shuffle Walking</b>	0.9867	0.0133	0.0000
<b>Normal Walking</b>	0.0400	0.9200	0.0400
<b>Toe Walking</b>	0.0000	0.0000	1.0000

table that MB-RF has a better response time and less memory storage. With regards to the CPU, memory, and battery usage, Fig. 11 shows the results obtained with the android profiler tool in the onStart method. It is clear from this figure that SVM consumes a lot of resources (high battery and memory

consumption with almost 50% extra CPU usage). For RF, we can see a bit more memory and energy usage than MB-RF.

TABLE VII: Average response time and memory usage

	MultiBoostAB-RF	RandomForest	SVM
<b>ART (ms)</b>	0.458893563	1.611349833	0.727204667
<b>RAM (MB)</b>	1.3	1.8	1.9

## VII. CONCLUSION AND FUTURE WORK

In this paper, a smart insole based system for automatic human gait analysis, dubbed BoostSole, was proposed. The aim was to develop a low-cost, objective, and reliable system to help physicians in continuous analysis of walk patterns. Obtained results demonstrated the capacity of BoostSole to provide accurate rates while consuming fewer resources. Nevertheless, BoostSole can be enriched by adding more sensors to be able to distinguish a wider array of gait types across



heterogeneous populations. Furthermore, using flexible and miniaturized chips can make BoostSole more practical and ergonomic. Moreover, while the local communication might not require strong security, remote communications with the server need to be investigated for proper security and less bandwidth consumption. Finally, we are planning to expertise BoostSole with health specialists. Investigating other on-device learning algorithms such as deep federated learning is also planned.

#### REFERENCES

- [1] A. Muro-de-la Herran, B. Garcia-Zapirain, and A. Mendez-Zorrilla, "Gait Analysis Methods: An Overview of Wearable and Non-Wearable Systems, Highlighting Clinical Applications," *Sensors (Basel, Switzerland)*, vol. 14, pp. 3362–3394, Feb. 2014.
- [2] W. Donkrajang, N. Watthanawisuth, J. P. Mensing, and T. Kerdcharoen, "Development of a wireless electronic shoe for walking abnormalities detection," in *The 5th 2012 Biomedical Engineering International Conference*, pp. 1–5, Dec. 2012.
- [3] S. J. Morris, *A shoe-integrated sensor system for wireless gait analysis and real-time therapeutic feedback*. PhD thesis, Massachusetts Institute of Technology, 2004.
- [4] S. Bamberg, A. Benbasat, D. Scarborough, D. Krebs, and J. Paradiso, "Gait Analysis Using a Shoe-Integrated Wireless Sensor System," *IEEE Transactions on Information Technology in Biomedicine*, vol. 12, pp. 413–423, July 2008.
- [5] Hyejeong Nam, Jin-Hyun Kim, and Jee-In Kim, "Smart Belt : A wearable device for managing abdominal obesity," in *2016 International Conference on Big Data and Smart Computing (BigComp)*, (Hong Kong, China), pp. 430–434, IEEE, Jan. 2016.
- [6] A. De Santis, E. Gambi, L. Montanini, L. Raffaelli, S. Spinsante, and G. Rascioni, "A simple object for elderly vitality monitoring: The smart insole," in *2014 IEEE/ASME 10th International Conference on Mechatronic and Embedded Systems and Applications (MESA)*, (Senigallia, Italy), pp. 1–6, IEEE, Sept. 2014.
- [7] W. Donkrajang, N. Watthanawisuth, J. P. Mensing, and T. Kerdcharoen, "A wireless networked smart-shoe system for monitoring human locomotion," in *The 4th 2011 Biomedical Engineering International Conference*, (Chiang Mai, Thailand), pp. 54–58, IEEE, Jan. 2012.
- [8] J. Bae and M. Tomizuka, "Gait Phase Analysis based on a Hidden Markov Model," *IFAC Proceedings Volumes*, vol. 43, no. 18, pp. 746–751, 2010.
- [9] G.-M. Jeong, P. Truong, and S.-I. Choi, "Activity classification of three types of walking regarding stairs using plantar pressure sensors," *IEEE Sensors Journal*, vol. PP, pp. 1–1, 03 2017.
- [10] T. Nilpanapan and T. Kerdcharoen, "Social data shoes for gait monitoring of elderly people in smart home," in *2016 9th Biomedical Engineering International Conference (BMEiCON)*, (Laung Prabang, Laos), pp. 1–5, IEEE, Dec. 2016.
- [11] S. Marquez J, R. Atri, M. R. Siddiquee, C. Leung, and O. Bai, "A Mobile, Smart Gait Assessment System for Asymmetry Detection Using Machine Learning-Based Classification," *Journal of Biomedical Engineering and Medical Devices*, vol. 03, no. 02, 2018.
- [12] S.-S. Lee, S. T. Choi, and S.-I. Choi, "Classification of Gait Type Based on Deep Learning Using Various Sensors with Smart Insole," *Sensors*, vol. 19, p. 1757, Apr. 2019.
- [13] S.-I. Choi, S.-S. Lee, H.-C. Park, and H. Kim, "Gait Type Classification Using Smart Insole Sensors," in *TENCON 2018 - 2018 IEEE Region 10 Conference*, (Jeju, Korea (South)), pp. 1903–1906, IEEE, Oct. 2018.
- [14] W.-k. Tam, A. Wang, B. Wang, and Z. Yang, "Lower-body posture estimation with a wireless smart insole," in *2019 41st Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, (Berlin, Germany), pp. 3348–3351, IEEE, July 2019.
- [15] D. Chen, Y. Cai, X. Qian, R. Ansari, W. Xu, K.-C. Chu, and M.-C. Huang, "Bring Gait Lab to Everyday Life: Gait Analysis in Terms of Activities of Daily Living," *IEEE Internet of Things Journal*, vol. 7, pp. 1298–1312, Feb. 2020.
- [16] N. Carbonaro, F. Lorussi, and A. Tognetti, "Assessment of a Smart Sensing Shoe for Gait Phase Detection in Level Walking," *Electronics*, vol. 5, p. 78, Nov. 2016.
- [17] R. Das and N. Kumar, "Investigations on postural stability and spatiotemporal parameters of human gait using developed wearable smart insole," *Journal of Medical Engineering & Technology*, vol. 39, pp. 75–78, Jan. 2015.
- [18] G. Saggio, F. Riillo, L. Sbernini, and L. R. Quitadamo, "Resistive flex sensors: a survey," *Smart Materials and Structures*, vol. 25, no. 1, p. 013001, 2015.
- [19] J. Pineda-Gutierrez, L. Miro-Amarante, M. Hernandez-Velazquez, F. Sivianes-Castillo, and M. Dominguez-Morales, "Designing a Wearable Device for Step Analyzing," in *2019 IEEE 32nd International Symposium on Computer-Based Medical Systems (CBMS)*, (Cordoba, Spain), pp. 259–262, IEEE, June 2019.
- [20] A. Jor, S. Das, A. S. Bappy, and A. Rahman, "Foot Plantar Pressure Measurement Using Low Cost Force Sensitive Resistor (FSR): Feasibility Study," *Journal of Scientific Research*, vol. 11, pp. 311–319, Sept. 2019.
- [21] A. Jović, K. Brkić, and N. Bogunović, "Decision Tree Ensembles in Biomedical Time-Series Classification," in *Pattern Recognition*, vol. 7476, pp. 408–417, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. Series Title: Lecture Notes in Computer Science.
- [22] G. I. Webb, "Multiboosting: A technique for combining boosting and wagging," *Machine Learning*, vol. 40, no. 2, pp. 159–196, 2000.