Contents lists available at ScienceDirect

Heliyon

journal homepage: www.cell.com/heliyon

Research article

Unsupervised and scalable low train pathology detection system based on neural networks

Jorge Sanchez-Casanova [*], Judith Liu-Jimenez, Paloma Tirado-Martin, Raul Sanchez-Reillo

University Group for ID technologies (GUTI), University Carlos III of Madrid, Spain

ARTICLE INFO

Keywords:
Pathology detection
Recurrent neural network
Gait analysis
Biomechanics
Signal processing
Pattern recognition

ABSTRACT

Currently, there exist different technologies applied in the world of medicine dedicated to the detection of health problems such as cancer, heart diseases, etc. However, these technologies are not applied to the detection of lower body pathologies. In this article, a Neural Network (NN)-based system capable of classifying pathologies of the lower train by the way of walking in a non-controlled scenario, with the ability to add new users without retraining the system is presented. All the signals are filtered and processed in order to extract the Gait Cycles (GCs), and those cycles are used as input for the NN. To optimize the network a random search optimization process has been performed. To test the system a database with 51 users and 3 visits per user has been collected. After some improvements, the algorithm can correctly classify the 92% of the cases with 60% of training data. This algorithm is a first approach of creating a system to make a first stage pathology detection without the requirement to move to a specific place.

1. Introduction

Gait analysis consists of the study of the movement of the human body. The obtained information can be used to identify people [1], for medical [2] or for sports purposes [3]. Healthy walk pattern has been already studied by Watelain et al. [4], any change in it can be used to identify both physical [5] and neurological problems [6]. There are also studies that use the gait analysis for rehabilitation purposes [7, 8]. In all previous cases, gait analysis is performed using devices that register the body movements. Traditionally the analysis was conducted by a specialist in a controlled laboratory with an optical system based on cameras and infrared (IR) markers that are placed on the patient's body. The main problem of those systems is that are quite expensive and force the patient to move to the place where the data is going to be acquired.

Currently, new technologies have made possible to create new low-cost systems with a suitable level of precision [9]. These new systems are based in different technologies such as cameras [10, 11, 12], pressure treadmills [13], insoles [14] and kinematic systems [15]. Those technologies have allowed the human gait analysis using wearable sensors [16] and the possibility of carrying out the gait analysis to non-hospital environments [17]. These improvements are really significant because the way of walking can be affected by the environment [18]. So, for this paper, a fully portable system has been used. This system is based in 3 dimensions (3D) kinematic sensors placed in the lower train.

Following this path, the goal of this paper is to present an algorithm capable of distinguishing pathologies of the lower train with the requirement of not needing to register previously to be able to use the system in non-controlled environments. With this objective in mind a database of healthy and pathological walks has been created. In this case a clubfoot has been simulated as pathological walk, however due to the results presented here we expect the system can be used for any pathology. The signals have been processed in order to remove the unnecessary information and to extract the GCs. Once the cycles are prepared, a Recurrent Neural Network (RNN) has been used to classify the walks.

This article presents first the evaluation database, the used device and the proceeding and the final dataset. All this information can be found in section 2. In section 3 the description of the proposed algorithm is exposed. The explanation of the NN and the optimization process are presented in section 4 and 5. The last sections correspond with the performed experiments and the conclusions.

2. Evaluation database

A proper database is an essential requirement to evaluate an algorithm as the quality of the signals in the database influences the results.

[*] Corresponding author.
E-mail address: jorgsanc@uc3m.es (J. Sanchez-Casanova).

https://doi.org/10.1016/j.heliyon.2021.e06270
Received 9 September 2020; Received in revised form 14 December 2020; Accepted 9 February 2021

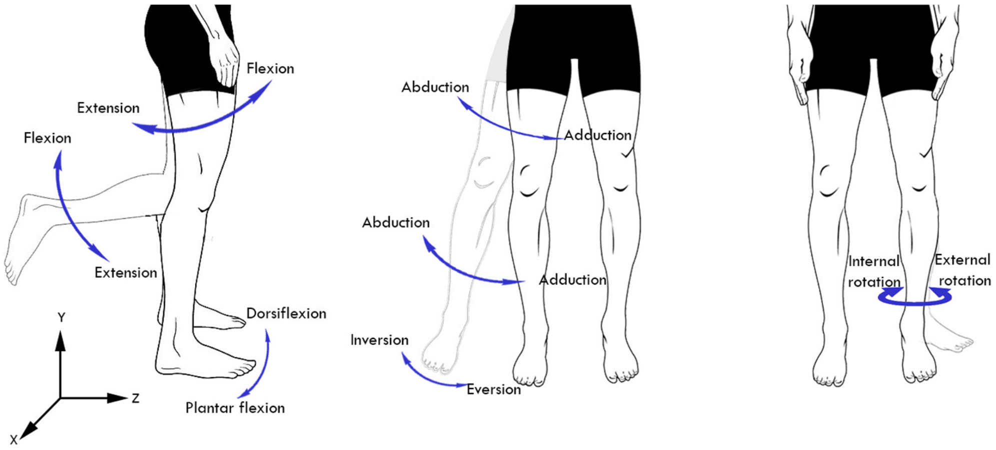

Fig. 1. Different angle movements.

Table 1. Technical specification of the sensors.

	Dynamic Range	Sensibility
Accelerometer	±34.9 m/s^2	0.06 mV/o/S
Gyroscope	±39.22 – 156.88 rad/s	0.122 mg
Magnetometer	±810 μT	0.092 V/gauss

Due to the absence of databases with healthy and pathological walk signals, for this study we collected our own database. Although any pathology could be used, to evaluate the system this study uses clubfoot as simulated pathology. In this section, the used device to acquire the signals, the protocol and the final dataset are explained.

2.1. Capture system

The device used to acquire the database is the Technaid Tech-MCS v3 [19], a professional tool to register the movement and the orientation of the human body. The Tech-MCS can be divided into the Tech-IMUs and the Tech-HUB. The IMUs (Inertial Measurement Unit) are small electronic devices based on MEMS (Micro Electro-Mechanical Systems) technology. Inside of each IMU, there is an accelerometer, a gyroscope and a magnetometer, all working in 3D. The Tech-HUB is the device that collects and stores all the data from the IMUs. This system can capture the acceleration (m/s^2), angular acceleration (rad/s) and the magnetic field (μT) for each IMU. All this information is merged using a Kalman Extended Filter (KEF) [20] to obtain the orientation of each IMU. With these orientations, the joint angles can be obtained. In Table 1 the dynamic range and the sensibility of each sensor are presented.

2.2. Evaluation protocol

Since the lower train movements are those we are interested in, the IMUs have been placed in both legs in the following way: 2 at each foot, 2 at the middle of each shin, 2 at the middle of each thigh and one at the middle of the lumbar. Before each walk the IMUs are calibrated by the Tech-HUB, obtaining the relative position between them, thereby, removing the differences in IMU positions on different visits.

In each walk 81 different signals are acquired with a sampling frequency of 250 Hz. Sixty-three of these signals correspond to the information from the sensors (acc, gyro, mag), the others 18 are the angle values from the joints and correspond to different movements of the leg. In Fig. 1 the different movements per plane that are obtained by the system are presented. Fig. 3 shows an angle signal from the knee. Analysing the angle signal, it can be clearly appreciate the GCs, furthermore it can be find out that the highest value is where the leg is totally extended.

Because of the difficulty to have access to people with some pathology and the reduced number of people with a common pathology, for this study a simulated pathology is used. Due to comfortability for the users and the easiness to replicate, clubfoot walk is used as a fake pathology and a sole padding is used to simulate it.

To avoid spurious data and to maximize the amount of data each user must do three visits within an established period (Table 2). Each visit consists of sixteen walks: eight healthy walks, four left pathology walks and four right pathology walks. There must be at least fifteen days between the first and second visit, and two months between the second and the third. The walks are performed in a twenty meters flat surface, with the only restriction of wearing neither heels nor slippers.

3. Proposed algorithm

The objective of the system is to create an algorithm capable of distinguishing between healthy walks, right pathology walks and left pathology walks. As it is shown in Fig. 2 the algorithm is divided into three different sections: pre-processing, cycle extraction and classification.

Walking is a repetitive movement, right-step left-step and repeat. These repetitions are called gait cycles. The frequency of these GCs depends on the person, however Fernandez-Lopez et al. [1] have demonstrated that the frequency is approximately 1 Hz. As a result of the low frequency of the movements, the signals can be filtered in order to remove unnecessary information. After the filtering process the signal is trimmed in order to obtain the GCs. Using those CGs the feature matrix is created. In the following subsections, all the process is explained in detail.

3.1. Pre-processing

As mentioned above, gait is a low-frequency signal, thus there is information that can be filtered without losing discriminative information. Due to the various positions and sensors, the main frequency can vary from one to another, so to prevent losing valuable information instead of setting a common cut-off frequency all the signals are evaluated in the frequency domain. After running some tests, we set heuristically the cut-off frequencies at the point where the power spectrum falls under −20 dB. A third-order Butterworth filter is used due to its simplicity and its flat response. In Table 3 the cut-off frequencies that accomplish the mentioned rule are presented. As we can see the values from shin and feet are higher than those from the lumbar and thigh. This is because the shin and the feet make small movements with a higher frequency.

In Fig. 3 the frequency and time domain of the left knee signals before (left) and after (right) filtering are showed. As it can be appreciated thought this process the frequency information is reduced but in the time domain the signal looks the same. So, filtering the signals

Table 2. Dataset samples.

Visit	Users	Pathological walks	Healthy walks	Acc signals (3D)	Mag signals (3D)	Gyr signals (3D)	Angle signals
1	51	408	408	5712	5712	5712	14688
2	32	256	256	3584	3584	3584	9216
3	21	168	168	2352	2352	2352	6048
Total		832	832	11648	11648	11648	29952

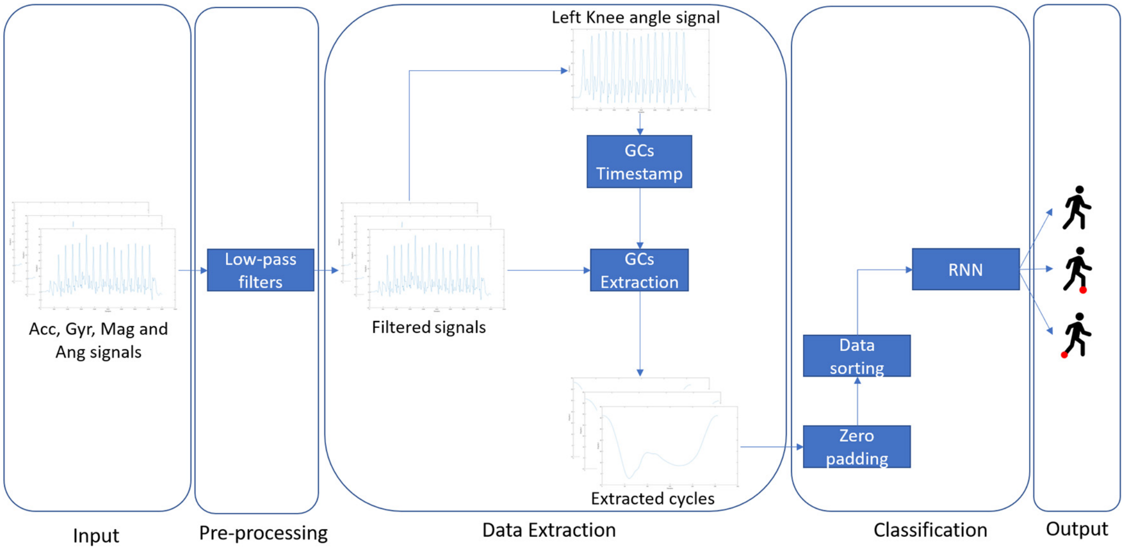

Fig. 2. Algorithm schema.

Table 3. Cut-off frequencies.

Position	Signal	frequency (Hz)
All	Angle	10
Lumbar and thigh	Accelerometer	10
	Gyroscope	5
	Magnetometer	5
Shin and feet	Accelerometer	20
	Gyroscope	10
	Magnetometer	10

have removed the unnecessary information which is located in high frequency.

3.2. Data extraction

Each walk is formed by several GCs, so extracting and using those GCs instead of the whole signal reduces the computational cost of the algorithm, and increases the data significantly. The process of finding the starting points of the cycles is performed once for each group as all the signals from the same walk have the same timestamps. Because of the clearness of the GCs, the flexion/extension movement from the left knee (x_{LK}) is used as a guide for the data extraction process. Another feature to bear in mind is the GC duration, so a minimum and maximum length are set. Thanks to those characteristics the process of obtaining the starting points of the CGs can be done in four steps:

1. The first step is to obtain the peaks that accomplish the next relationship (V_{pk}):

$$V_{pk} > max(x_{LK}) - 2\frac{max(x_{LK}) + |min(x_{LK})|}{5} \tag{1}$$

   This means that only the values placed in the upper part of the signal are taken into consideration. These points are where the leg is totally extended forward.

2. The next step is to guarantee that the distance between the peaks is bigger than 0.9 seconds between the points [21] (f_s = sampling rate, P_{pk_n} = position of the n^{th} peak).

$$P_{pk_{n+1}} - P_{pk_n} > 0.9 f_s \tag{2}$$

3. The following step consists in discarding all the cycles that are longer than a threshold empirically found, which corresponds to:

$$P_{pk_{n+1}} - P_{pk_n} \leq 8\frac{\sum_{i=1}^{n} P_{pk_i}}{5n} \tag{3}$$

   The purpose of this is to avoid those cycles that due to an error were not correctly acquired, an example of one of those cycles can be seen in Fig. 4d.

4. The last step is to discard all the points that are in the first and last 3 seconds to avoid GCs with an uncommon waveform.

$$P_{pk_1} < 3 f_s \tag{4}$$

$$P_{pk_n} > length(x_{LK}) - 3 f_s \tag{5}$$

Fig. 4 shows the process of extracting the GCs. In Fig. 4a, the result of finding the peaks with any restriction can be observed. Fig. 4b shows the process of looking for the peaks considering the equation (1). It is hard to appreciate with naked eye but at each maximum point, there are two peaks detected. To solve the problem the minimal distance between peaks (equation (2)) is also applied in Fig. 4c. In Fig. 4c the peaks within the three first and last seconds (equations (4) and (5)) are also removed.

As all the walk signals were acquired at the same time all of them have the same timestamps, we can divide all the signals using the points obtained in the process above explained. It should be noted that not all the GCs have the same length, even between cycles in the same walk. In order to solve the problem and avoid losing information, the length of all the cycles is increased up to the length of the longer one by zero padding. Finally, the last step is to sort all the data in a matrix, where the rows correspond with the samples of the cycles and the columns with the different sensors.

$$M_n = \begin{bmatrix} An_0 & Ac_0 & Gy_0 & Mg_0 \\ An_1 & Ac_1 & Gy_1 & Mg_1 \\ \vdots & \vdots & \vdots & \vdots \\ An_L & Ac_L & Gy_L & Mg_L \end{bmatrix}$$

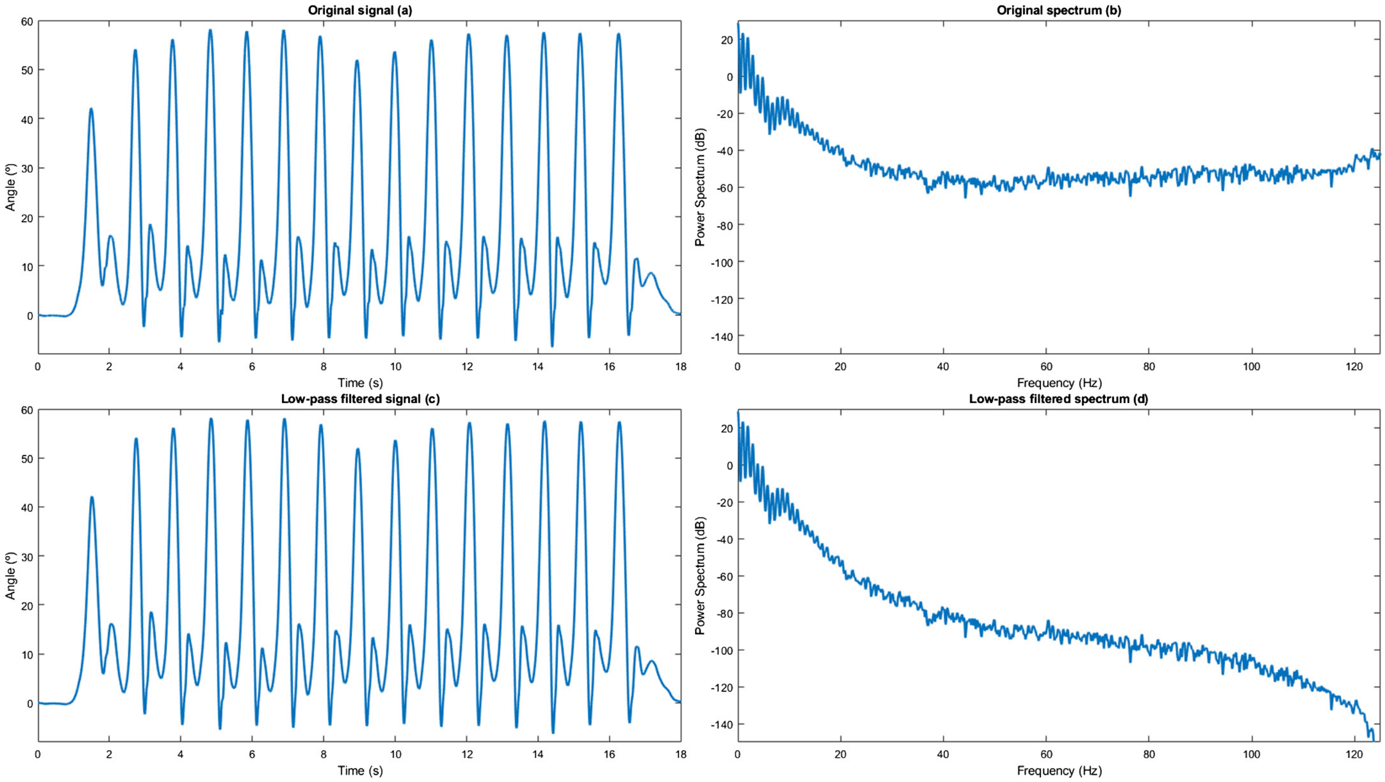

Fig. 3. Knee signal before (left) and after (right) filtering.

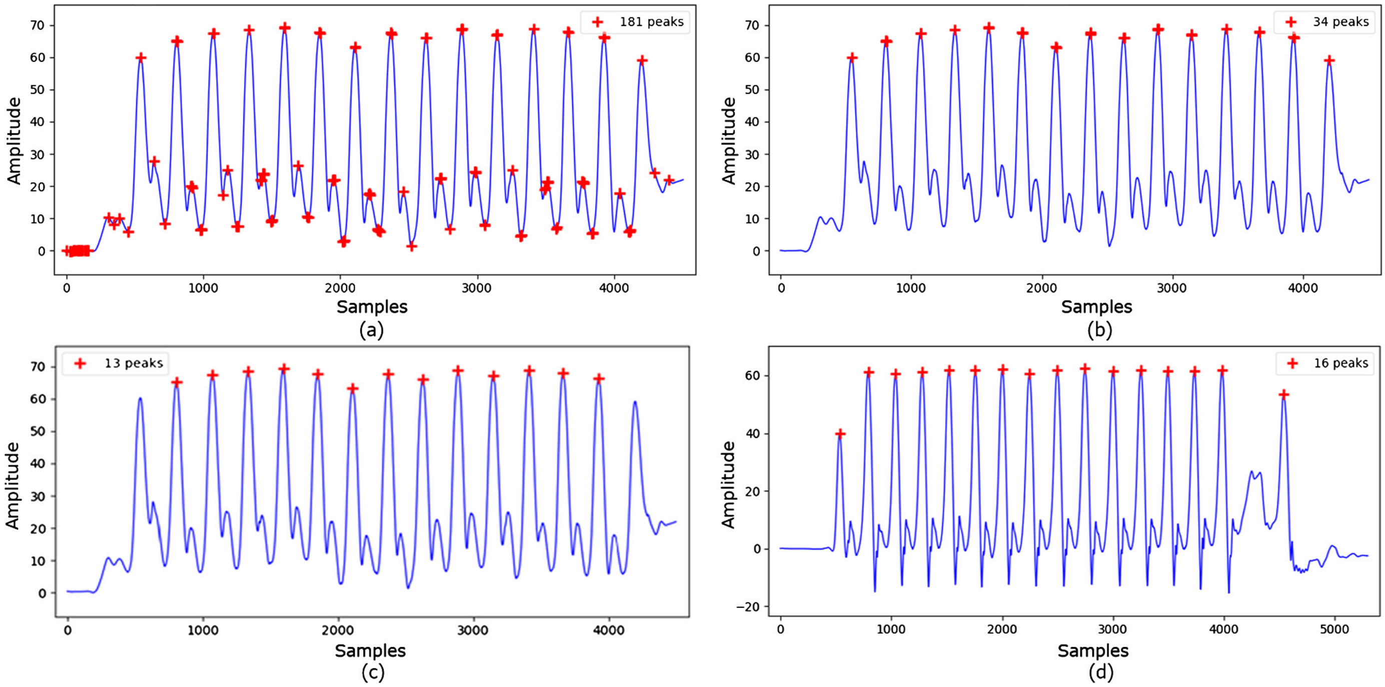

Fig. 4. Data extraction process: a) peaks obtained without restriction. b) peaks obtained considering equation (1). c) peaks obtained considering equations (1)-(5). d) example of a signal with an incorrect cycle.

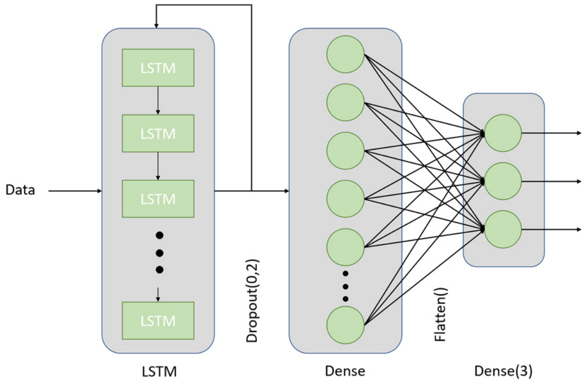

Fig. 5. Neural network schema.

Where:

$$An_n = \begin{bmatrix} LHip_n & RHip_n & LKnee_n & RKnee_n & LAnkle_n & RAnkle_n \end{bmatrix}$$

$$Ac_n = \begin{bmatrix} Lumb_n & LThigh_n & RThigh_n & LShin_n & RShin_n & LFoot_n & RFoot_n \end{bmatrix}$$

L is the maximum number of samples in the longest cycle.

The structure of the matrix is as follows. An_n is a vector formed by information from the left hip ($LHip_n$), right hip ($RHip_n$), left knee ($LKnee_n$), right knee ($RKnee_n$), left ankle ($LAnkle_n$) and right ankle ($RAnkle_n$) from the n^{th} GC. This information corresponds to the joint angles shown in Fig. 1. Similarly, Ac_n corresponds to the accelerometer information, and it is formed by information from the Lumbar ($Lumb_n$), left thigh ($LThigh_n$), right thigh ($RThigh_n$), left shin ($LShin_n$), right shin ($RShin_n$), left foot ($LFoot_n$) and right foot ($RFoot_n$). The structure of Gy_n and Mg_n is the same as that of Ac_n, but it contains the information from the gyroscope and magnetometer.

3.3. Classifier: neural network

Neural networks are a set of algorithms that mimics the way the human brain operates with the purpose of recognising patterns [22]. These algorithms have different attributes such as adaptive learning, real-time operation and prognosis, which make NNs a powerful tool to solve different problems. The capability of NN algorithms detecting health problems has been widely demonstrated [23, 24].

As we are working with temporal signals, RNN are used, and in special a Long-Short Term Memory (LSTM) layer [25]. RNNs have a high configuration capability, the problem is that for a new approximation there is no reference of the number of layers, number of neurons per layers, activation algorithm, filters per layer, etc. Finding the best configuration of these hyperparameters to solve the problem can be an arduous task, but some techniques make the process easier such as Random Search (RS).

In Fig. 5 a scheme of the NN is presented. An LSTM layer is used as an input layer and dense layers are used as hidden layers. The number of hidden layers is established in the hyperparameter optimization process. The output layer is also a dense layer with three neurons, coinciding with the number of classes (healthy, right pathology and left pathology). In order to prevent overfitting and to increase the accuracy a dropout of 0.2 is performed before every dense layer excluding the output layer.

Once we have the procedures next step is to find the optimal NN configuration. We can divide the process into two: split up the dataset and optimise the hyperparameter using RS.

4. RNN hyperparameter optimization

The first step is to divide the original dataset into two, one for training and other for testing. To create the sub-datasets all the walks are randomly divided. In the hyperparameter optimization process, the NN is trained using 60% of data and tested with the remaining 40%.

Table 4. Hyperparameters range.

Variable	Values
Number of hidden layers (HL)	[1,5]
Number of filters (NF)	$2^{[4,10]}$
Learning rate (LR)	$10^{[-1,-4]}$
Relationship between layers (RL)	[1,4]
First layer activation	tanh
Hidden layers activation	relu
Output layer activation	softmax
Optimizer	adam

Table 5. Different relationships used. n is the value under study. m is the number of the layer.

Number	Relationship
1	2^{n}
2	2^{n-m}
3	2^{n+m}
4	$2^{n-(m+1)}$

The process of creating the two datasets is different between train and test. In the training dataset, the cycles from the walk are extracted and used individually as an input to train the network. However, for testing the cycles are extracted and grouped depending on the walk. In the classification process, the cycles of the same walk are classified individually, and the mean of the result of all the cycles is used as a result of the walk.

4.1. Random search optimization

RS is a method to find the optimal hyperparameters combination. Upper and lower limits are set for each hyperparameter, and in each execution a value within the established range is chosen. When the process has been executed enough times, hyperparameters limits can be reduced by discarding those values which provides lower accuracy and the process repeated. Once the resulting grid is small enough instead of repeating the process, all the possible values are tested.

In the first place, to obtain a general idea about the hyperparameter optimization, the algorithm is run 5% of the total cases. Once the grid is reduced the number of executions grows up to 20% of the new possibilities. After the second execution of RS, the hyperparameters are fine tuned to find the optimal configuration. In order to prevent that the hyperparameter optimization gets stuck in specific group of data, the groups are randomly chosen for each iteration.

In each RS loop, one random value of each hyperparameter is chosen, and the process of training/testing is performed 5 times. The taken accuracy for that configuration is the mean value of those 5 executions.

In Table 4 the ranges of values of the hyperparameters are showed. As both the number of layers and the number of neurons/filters per layers are under study, the number of their combinations grows exponentially. To reduce the complexity of the study, instead of trying all the possibilities four different relationships between the number of filters of the first layer and the number of neurons of the hidden layers are studied. Table 5 shows the different relationships studied.

After executing the RS algorithm few times we can have an idea about which configurations work better. The accuracy results of these executions are presented in Table 6. It can be seen that the configurations with one and five Hidden Layers(HL) do not show the best accuracy, furthermore, a low number of neurons per layer do not give a proper result, neither the higher one does it. So, we can see that the central configurations are the most optimal. The Learning Rate (LR) and Relation between Layers (RL) parameters do not show a clear relation about which is better. Because of the results, we can set the limit of HL between 2 and 4 and Number of Filters (NF) from 128 to 512, in addition the cases from HL=1 and RL=4 are also included. In the case of the LR and the RL, we maintain the same limits. Once the new limitations

Table 6. Accuracy results of the algorithm in the first random search execution. The red rectangle demarcate the new borders for the second execution of RS algorithm. HL=hidden layers. RL=relationship between. NF=number of filters of the first layer. LR=Learning rate.

	HL	1	1	1	1	2	2	2	2	3	3	3	3	4	4	4	4	5	5	5	5
LR	NF/RI	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4	1	2	3	4
10^{-1}	16															62.5					
10^{-2}	16			51.6							44.7			69.7							
10^{-3}	16																		56.5		
10^{-4}	16					61.2															
10^{-1}	32														55.7						
10^{-2}	32									60.3											
10^{-3}	32																		63.3		
10^{-4}	32						65.3						71.6								
10^{-1}	64			46.9																	
10^{-2}	64																				
10^{-3}	64									68.5					78.5					72.5	
10^{-4}	64																				
10^{-1}	128												85.8				82.6				
10^{-2}	128	72.5																			
10^{-3}	128								81.6												
10^{-4}	128			67.8																	
10^{-1}	256														72.2						
10^{-2}	256				85.6																
10^{-3}	256																				66.3
10^{-4}	256							84.5									72.6				
10^{-1}	512											79.6			81.5						
10^{-2}	512					78.3															
10^{-3}	512										75.8										
10^{-4}	512		65.7																		
10^{-1}	1024																71.8				
10^{-2}	1024			46.8															68.4		
10^{-3}	1024													65.2							
10^{-4}	1024								41.6												

are established, the RS algorithm is run to obtain the best configuration. After the RS optimization process, we look for the configuration with higher accuracy. The best configuration is the one with NF=128, HL=3, RL=3, and LR = 10^{-2} which gives an accuracy of 86.2%.

4.2. Fine-tuning optimization

The next step is to fine-tune the hyperparameters using the results of the last RS execution as reference. Due to the changes in LR does not have an impact on the accuracy the value is fixed to the one in the configuration above. In the case of the NF, the accuracy is higher with NF=128, so in the fine-tuning process, it remains unchanged. The different relationships give quite different results, so only the RL=3 and RL=4 are used, which are the two configurations with the highest accuracy. As the best value for HL cannot be discerned the values of HL goes from 2 to 4, in the last RS execution. Finally, the training/testing ratio is added as a parameter under study to the fine-tuning process, in that way, a better behaviour of the algorithm can be observed. Moreover, the algorithm is executed ten times and the mean value is the one taken.

In Fig. 6 the accuracy results of the fine-tuning configurations are shown. The first observation we make is that the accuracy grows with the percentage of training data, this is a common behaviour in RNN algorithms, however overtraining increases the risk of overfitting phenomena. The configurations with worst results are the RNNs with HL4_RL3 and HL4_RL4, i.e. the configurations with more layers. It can also be seen that the configuration with RL = 3 gives a bit better results than the one with RS = 4. The next configurations offering better results are HL3_RL4 and HL2_RL4, and the results of both are so close. Finally, we have the configurations HL3_RL3 and HL2_RL3, if we compare them, we can see than with a low percentage of training data the configurations give similar results, but beyond the 40% of training data HL2_RL2 is the best configuration. At the light of the outcomes, we can say that the RNN offering the best results is the one with NF128_HL2_RL3_LR10^{-2}.

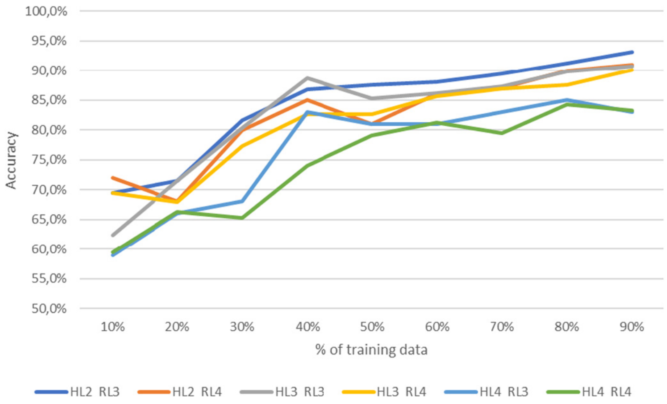

Fig. 6. Accuracy results of the fine-tuning configurations.

5. Experiments

At this point we can say the optimal configuration is the one with NF128_HL2_RL3_LR10^{-2}, so all the experiments are performed using that configuration. Moreover, to have a more detailed behaviour of the RNN all the experiments are performed using different ratios of training/testing data. The aim of the experiments is to evaluate the RNN behaviour in different scenarios and to improve it if possible. If one of the experiments improves the results, the changes are incorporated in the main RNN and used in the following experiments. The experiments that have been performed are the following:

1. To divide the dataset to train with some users and test with others.
2. Test whether adding the first and last 3 seconds improve the algorithm and adding the physiological information of the users (age, height, etc.).
3. Test whether it is possible reduce the quantity of signals without worsen the system.

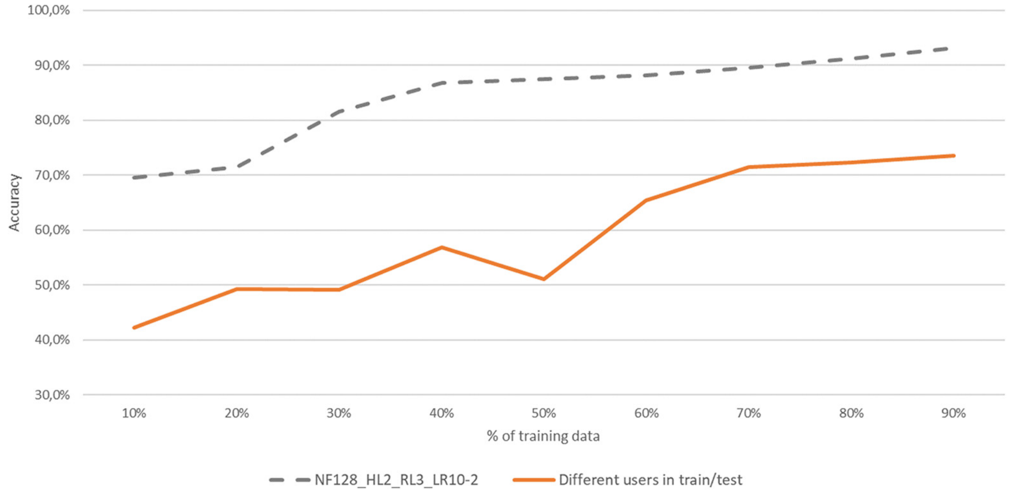

Fig. 7. Accuracy results for experiment 1.

4. Test whether all the cycles in a walk have the same importance.

5.1. Experiment 1

The most common problem in pattern recognition systems based on NN is when a new user is registered, the whole system must be retrained. In other fields, as biometric recognition, it is the normal behaviour, as you cannot be identified if you are not inside the system. However for medical purposes it is unviable to have samples of the users before the pathology identification. To solve this problem, the main objective of this paper is to create a valid system that uses different users in the training and testing dataset. So, from this experiment, all the datasets do not share users in the training and testing datasets.

The first step is to create the two datasets, to avoid the system always training with the same users, the users are randomly chosen in each iteration. Furthermore, as all the users do not have the same number of walks, for each configuration of training/testing ratio the algorithm is run 20 times, instead of the 10 as it was done in the optimization process.

In Fig. 7 we can see that the accuracy of the new approach falls sharply. This behaviour is due to the training and testing datasets no longer share users.

Removing the user variable, the process of obtaining the pattern that identifies the pathology becomes more complex. To solve this problem two new approaches are going to be evaluated: removing the restriction of the first and last 3 seconds and adding the physiological information of the users.

The results of the experiment do not present an improvement with respect to the initial configuration, however, the configuration is maintained in future experiments as this reflects the scalability system requirement.

5.2. Experiment 2

The aim of this experiment is to improve the accuracy of the algorithm presented in the last experiment by increasing the number of GCs and adding extra information about the user. With the purpose of adding more GCs, the data extraction process is modified: the limitation of removing the first and last three seconds is deleted (equations (4) and (5)), furthermore the physiological information is added.

Due to the cycles of the first and last three seconds are used, the data in the dataset grows, however the length of the cycles still the same. In this experiment, Matrix Mn is modified to include the following information: height (Ht), weight (Wt), foot size (Fs), gender (Gd), age (Ag) and sport activity (Sa). Each of these fields is introduced in each row of the previous Mn matrix.

$$M_n = \begin{bmatrix} An_0 & Ac_0 & Gy_0 & Mg_0 & Ht & Wt & Fs & Gd & Ag & Sa \\ An_1 & Ac_1 & Gy_1 & Mg_1 & Ht & Wt & Fs & Gd & Ag & Sa \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ An_L & Ac_L & Gy_L & Mg_L & Ht & Wt & Fs & Gd & Ag & Sa \end{bmatrix}$$

Where:

$Sa = $ Sport activity,

$$where \begin{cases} N = \text{None} \\ M = \text{once a month or less} \\ W = \text{once a week} \\ T = \text{twice a week or more} \\ D = \text{daily} \end{cases}$$

Fig. 8 shows the result of executing the algorithm with the new approaches. Blue line shows the results of the system when adding extra cycles from the beginning and the end of the walk. These results shows that the algorithm accuracy increases and that the response of the system is flatter. Yellow line shows the result of including the physiological information in addition to those extra seconds. As we can see the accuracy of the system has increased significatively, this can be explained due to physiological differences provide useful information to the system.

5.3. Experiment 3

In each walk, information of three different sensors and the joint angles are captured. Some information can be not useful or lead to the misidentification of the pathologies. Fig. 9 shows an example of all the signals captured by the system. As we can see the signals from the magnetometer does not have a clear pattern, such as the gyroscope or the accelerometer. In this experiment, the validity of different signals combinations of the data set are evaluated. Due to in the last experiment, the accuracy of the system has been improved the system configuration with the extra cycles and the physiological information is used.

In order to reduce the number of signals the module of the three different axes is performed for each of the signals i.e. accelerometer, gyroscope and magnetometer:

$$module = \sqrt{x^2 + y^2 + z^2} \tag{6}$$

To perform this experiment all the different combinations of the signals with and without using the module are tested. A total of 29 different configurations have been tested. For clarity of the results only the 6 best configurations are presented in Fig. 10.

In Fig. 10, the nomenclature used is the following. The different signals are accelerometer (Ac), gyroscope (Gy), magnetometer (Mg) and

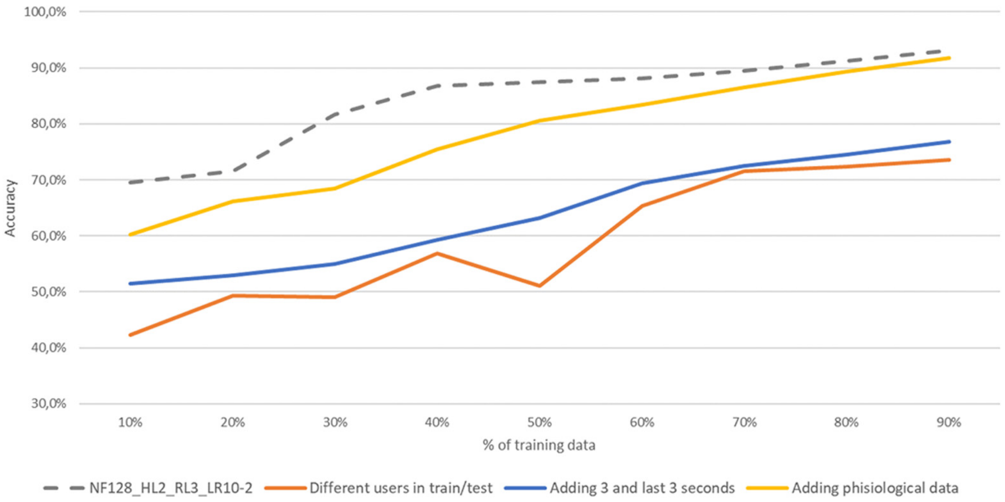

Fig. 8. Accuracy results for experiment 2.

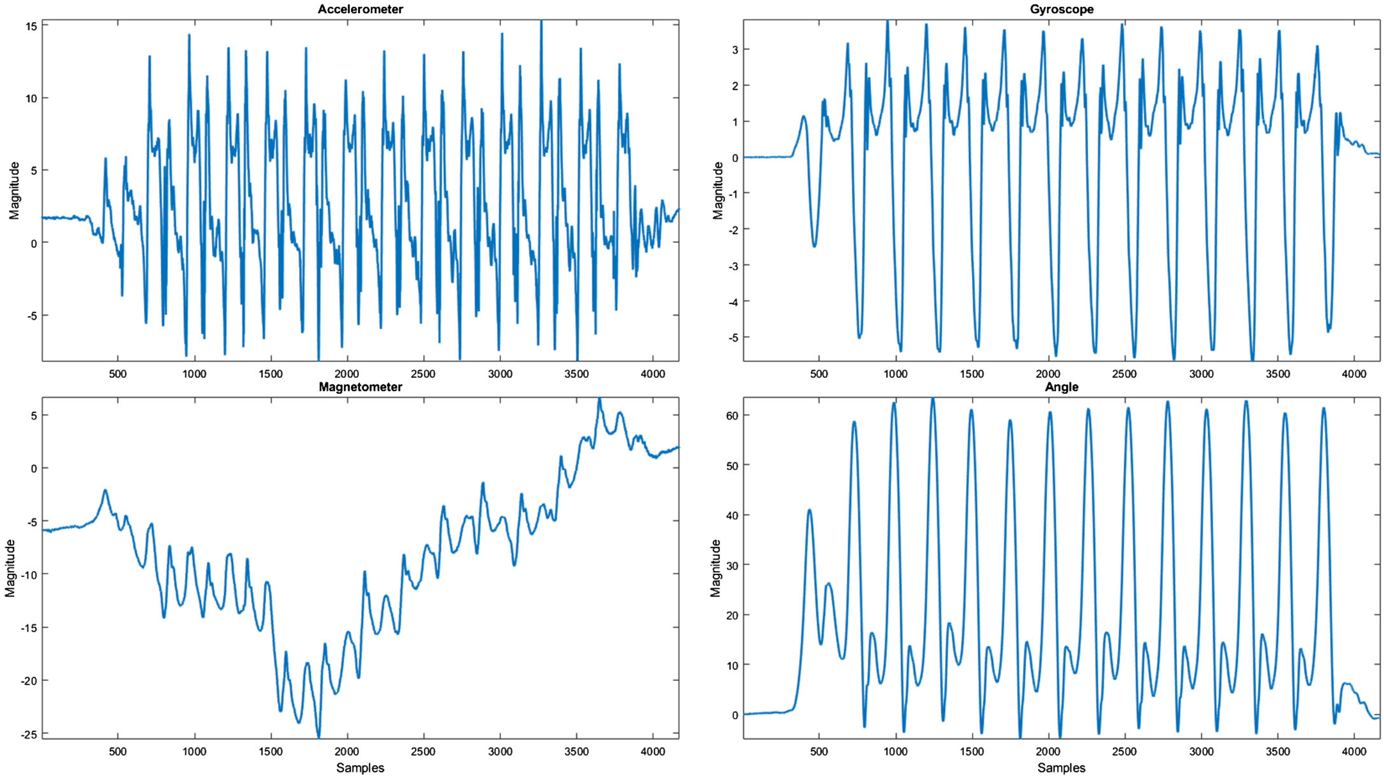

Fig. 9. Example of all the signals captured. Top-left accelerometer. Top-right gyroscope. Bottom-left magnetometer. Bottom-right angle.

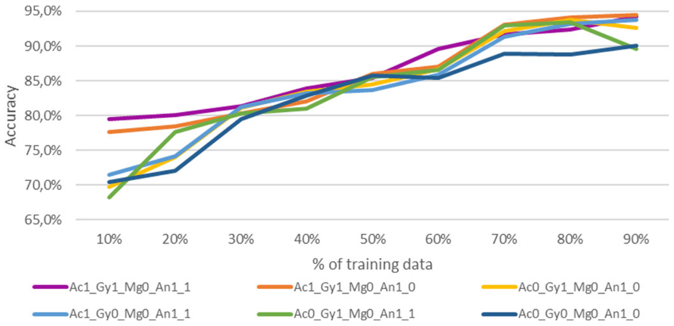

Fig. 10. Accuracy results for experiment 3.

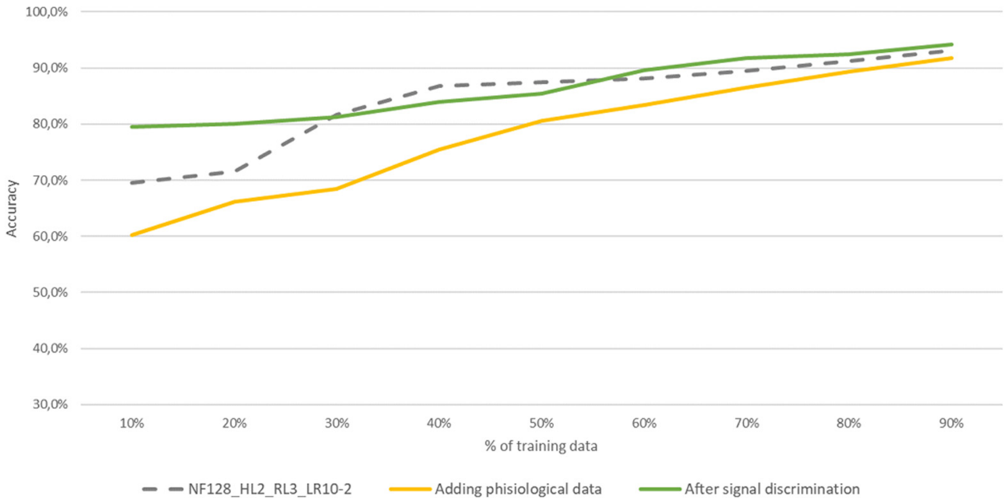

Fig. 11. Comparison of accuracy results.

angles (An). The “1” after the name indicates that the signal is used and “0” indicates the opposite. The last parameter specifies if the magnitude is used (1) or not (0). It can be seen accuracies of the different configurations are quite similar, mostly in the range from 30% to 70% of training data. If we observe the configurations, it can be appreciated that none of them uses the magnetometer signals, so the magnetometer signals do not provide relevant information. On the other side, the angle signals are in all the configurations. Concerning the accelerometer and gyroscope, the first one is in 50% of the configurations, and the other is in 67% of them. To the light of these results, we can say that the more important signals are the angles of the joints followed by the gyroscope signals. The accelerometer signals are the less relevant, however, they improve slightly the results. Regarding the signal module, there is no clear pattern. One configuration to note is Ac0_Gy0_Mg0_An1_0, which only using the angle signals is one of the best configurations. The two best configurations are Ac1_Gy1_Mg0_An1_1 and Ac1_Gy1_Mg0_An1_0, the difference between both of them is that one uses the module and the other one no. The Ac1_Gy1_Mg0_An1_1 configuration has higher accuracy in almost every ratio, and a flatter behaviour, so we take it as the best configuration. Once we have the best configuration, we compare it with the results of the previous experiment, Fig. 11. We can see that the accuracy of the new configuration has increased from the previous experiment, mostly with a low ratio of training data, but the most important fact is that the configuration gives better results than the original one. As the algorithm has been improved, the changes are maintained for the following experiments.

5.4. Experiment 4

When classifying the walks, the GCs of a walk are individually classified, and the mean value is used as a result of the walk. Doing this, all the cycles have the same importance in the final result, but it could be that some CGs are more significant than others. The aim of this experiment is to check if there are some CGs that provide better results than others, and if necessary, to create a weighing schema better than the mean value.

To do this the GCs of the walks are classified but instead of getting the mean value of the cycles of the same walk, the accuracy result of each GC is saved. Once all the walks are classified, the next step is to obtain the mean value of all the cycles of the same time position. In this experiment, all the training/testing ratios have been used and the final result for each GC is the average of all the ratios. Fig. 12 shows the mean of accuracy results for the different GCs positions.

As can be appreciated, the GCs from the beginning of the signal looks like they are more representative than others. To the light of outcomes, a weighted mean can be used. To obtain the weights the next formula is used:

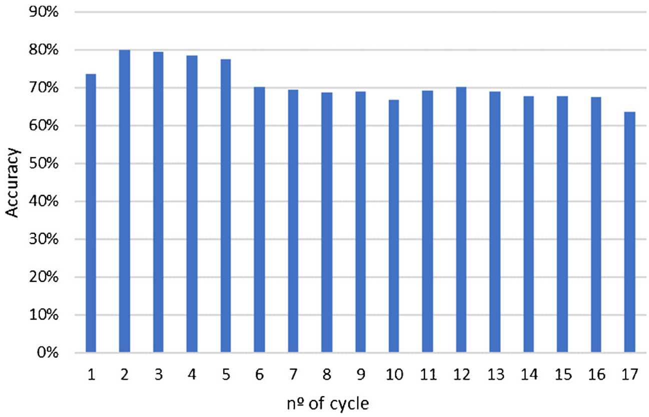

Fig. 12. Accuracy results of the different GCs positions.

$$w_i = 1 - (max(a) - a_i) \tag{7}$$

Where:

a_i is the accuracy from i^{th} cycle in the Fig. 12.
a are the accuracies of the different GCs positions.
i is the sub-index for the number of cycle.

Once all the weights are obtained the weighted mean is performed. In Fig. 13, the results of the execution of the algorithm with the weights are presented.

5.5. Final algorithm results

Once all the experiments are done the new system can be established. This new system consists of a RNN, with the same structure presented above. The hyperparameter configuration still the same, NF128_HL2_RL3_LR10^{-2}. Regarding to the experiments extra information the final configuration is the next:

- The users in the training and testing dataset are different.
- Removed the rule of discarding the first and last three seconds.
- Physiological data is included
- Magnetometer signals are discarded.
- To obtain the final result a weighted mean is used.

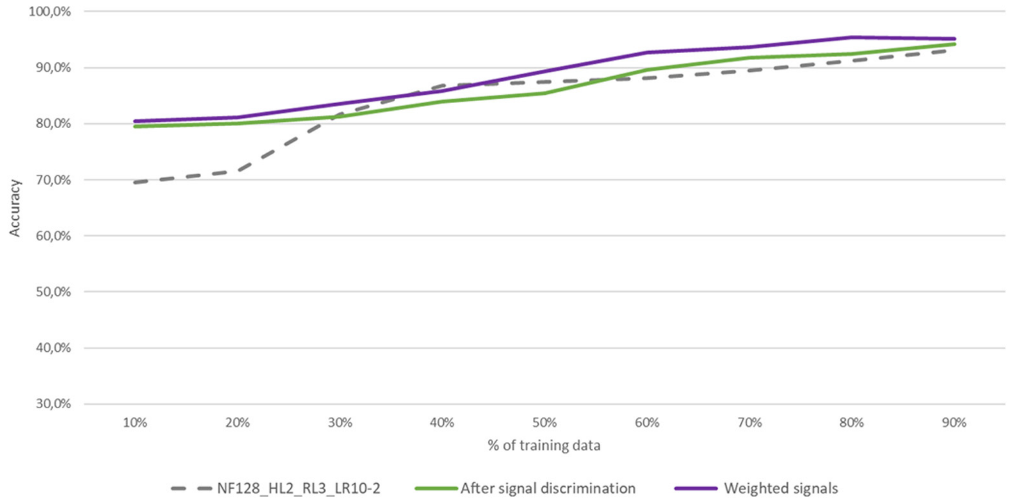

Fig. 13. Accuracy results of experiment 4.

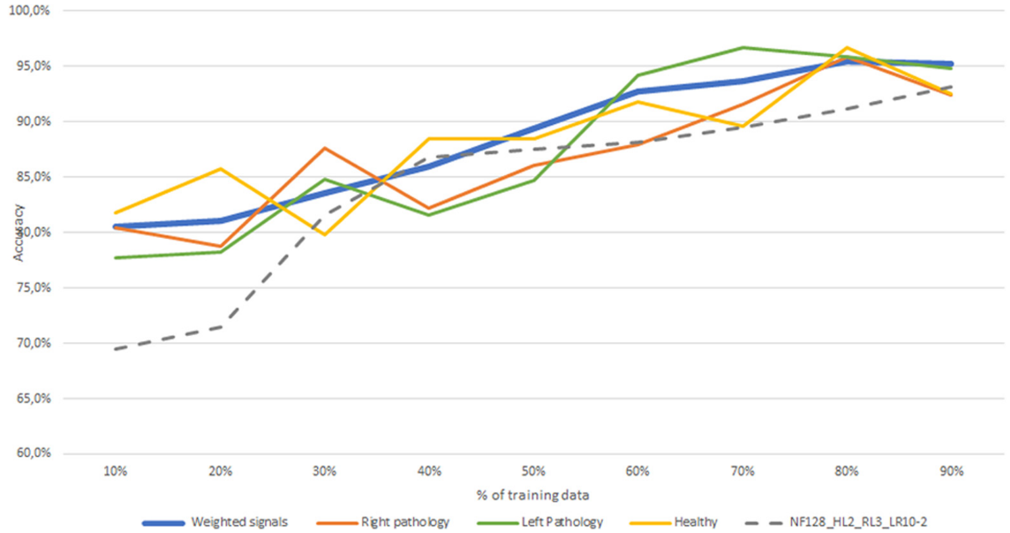

Fig. 14. Accuracy results of the classification with only one class.

Once the final algorithm is created it is necessary to evaluate if all the cases are equally classified. In order to perform this evaluation, the system is trained using different ratios of training and testing data. Regarding to the testing, the system is tested three times for each configuration, once using only right pathology walks, another one using only left pathology walks and the last using only healthy walks. By doing this, whether the system classifies equally all the walks can be observed.

As it can be appreciated in Fig. 14 there are not big differences classifying the cases, the maximum difference is around 4%. With these results, we can assume that the algorithm classifies all the cases equally.

As it is shown in Fig. 14, the new configuration of the algorithm (blue) is better than the original one (dotted grey). It can be observed that the bigger change happens with lower training ratio, being the improvement close to 10%. Beyond 30% of training ratio, the improvement is reduced up to 2.5%.

6. Conclusions

This article presents a RNN-based algorithm capable of classifying lower train pathologies with the restriction of not sharing users between the training and testing datasets. Thereby, it is not necessary to re-train the system when a new user arrives, giving scalability to the algorithm. The system works with the kinematic signals as well as the joint angles which are processed in order to extract the GCs. The information of the different sensors that corresponds with one gait cycle are sorted in a matrix, and those matrices are used as a basic unit to feed the RNN. In this paper a database with a simulated clubfoot pathology has been used, but the algorithm could be used for any pathology.

Due to the complexity of the problem, in the first approach, a NN is created to identify the clubfoot pathology without the restriction of not sharing users in the datasets. To optimize the network a random search method is performed. With this purpose, a range of values for some hyperparameters is established and the optimization is executed. After the first execution, the range of values is reduced, and the algorithm is run again. Finally, the best configuration is obtained using fine-tuning.

After the optimization progress, we found that the best configuration is the one with 128 filters in the first layer, 2 hidden layers, the third relationship between neurons, and a learning ratio of 10^{-2}. This configuration gives 89.5% accuracy with a ratio of 70%-30% of training/testing data. It is a proper result, but the system is not real-world appropriate, because for classifying a new user the whole system must be retrained.

In order to solve this problem, a new system is created with the restriction of not testing with the users used in the training. Due to this change, the accuracy of the system falls by approximately 20%. To improve the system some experiments have been performed.

The first experiment performed is to include the first and last 3 seconds of the signals, that in the original experiment were excluded. Result of this experiment the accuracy grows from 71.5% to 72.5%. This improvement is due to the amount of data have been increased. The next experiment is to add the physiological information of the users (height, weight, foot size . . .) to the system. This change improves the accuracy of the system by up to 86.5%. This improvement is due to the fact that the physiological differences affect the way of walking and adding that information the system can identify those differences and focus on the pathological patterns.

The following experiment evaluates whether all the used signals have a positive impact on the system or not. From the observations done in this work, magnetometer signal seems to worse the system results, that it is the reason to remove it from the data matrix, but maintain other signals such as accelerometer, gyroscope. Another interesting fact is that doing the module of the x, y, and z signals of the accelerometer and gyroscope improves a bit the algorithm. Applying these changes the accuracy of the system is improved up to 91.7%

The last experiment performed consists on evaluation the cycle influence. The cycles from the beginning of a walk have more significance than the rest. This may be due to the fact that at the beginning of the walk the pathology is more pronounced, and when we walk a bit, we get used to it. By entering this information into the system, the accuracy increases by up to 93.7%.

After all the experiments performed, we got an algorithm capable of classifying pathologies, without the limitation of previous registration of the user and the retraining of the whole system. Although the system is able to identify the pathologies, in the used database there only are one faked pathology. So, it would be interesting to evaluate the system with a real database with more than one real pathologies. Other facts to have in mind to make the system closer to a real-world one are the influence of all the sensors positions used, the importance of the surface, and the minimum number of cycles needed to identify pathologies.

Although there still some improvements to be made to the system so that it can be used in a real medical environment, with the contributions presented in this article it is possible to have a scalable system to make a first stage pathologies identification in an uncontrolled environment.

Author Contribution Statement

Jorge Sanchez-Casanova: Conceived and designed the experiments; Performed the experiments; Analyzed and interpreted the data; Contributed reagents, materials, analysis tools or data; Wrote the paper.

Judith Liu-Jimenez: Conceived and designed the experiments; Analyzed and interpreted the data; Wrote the paper.

Paloma Tirado-Martin: Conceived and designed the experiments; Wrote the paper.

Raul Sanchez-Reillo: Contributed reagents, materials, analysis tools or data.

Funding Statement

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

Data Availability Statement

The authors do not have permission to share data.

Declaration of Interests Statement

The authors declare no conflict of interest.

Additional Information

No additional information is available for this paper.

References

[1] P. Fernandez-Lopez, J. Sanchez-Casanova, P. Tirado-Martin, J. Liu-Jimenez, Optimizing resources on smartphone gait recognition, in: IEEE International Joint Conference on Biometrics, IJCB 2017 2018-Janua, 2018.

[2] H. Gök, S. Ergin, G. Yavuzer, Kinetic and kinematic characteristics of gait in patients with medial knee arthrosis, Acta Orthop. Scand. 73 (6) (2002) 647–652.

[3] H. Lee, S.J. Sullivan, A.G. Schneiders, The use of the dual-task paradigm in detecting gait performance deficits following a sports-related concussion: a systematic review and meta-analysis, Scand. J. Med. Sci. Sports 16 (1) (2013) 2–7.

[4] E. Watelain, F. Barbier, P. Allard, Gait pattern classification of healthy elderly men based on biomechanical data, Arch. Phys. Med. Rehabil. 81 (5) (2000) 579–586.

[5] M. Khokhlova, C. Migniot, A. Morozov, O. Sushkova, A. Dipanda, Normal and pathological gait classification LSTM model, Artif. Intell. Med. 94 (December 2018) (2019) 54–66.

[6] Y. Zhang, P.O. Ogunbona, W. Li, B. Munro, G.G. Wallace, Pathological gait detection of Parkinson's disease using sparse representation, in: 2013 International Conference on Digital Image Computing: Techniques and Applications, DICTA 2013, 2013.

[7] M. Go, C. Iacovelli, E. Russo, S. Pournajaf, C.D. Blasi, M. Franceschini, Stroke gait rehabilitation: a comparison of end-effector, overground exoskeleton, and conventional gait training, Applied Sciences 9 (13) (2019) 2627.

[8] H. Sritart, S. Taertulakarn, A Review of Wearable Sensor for Stroke Patients, 2016, pp. 27–32.

[9] A. Muro-de-la Herran, B. García-Zapirain, A. Méndez-Zorrilla, Gait analysis methods: an overview of wearable and non-wearable systems, highlighting clinical applications, Sensors (Switzerland) 14 (2) (2014) 3362–3394.

[10] E. Auvinet, J. Meunier, F. Multon, Multiple Depth Cameras Calibration and Body Volume Reconstruction for Gait Analysis, M2S - Universite de Rennes 2/Universite de Montreal, Montreal, Canada, 2020, pp. 478–483.

[11] A. Dubois, F. Charpillet, A Gait Analysis Method Based on a Depth Camera for Fall Prevention, 2014, pp. 4515–4518.

[12] B.F. Mentiplay, L.G. Perraton, K.J. Bower, Y.-h. Pua, R. Mcgaw, S. Heywood, R.A. Clark, Gait assessment using the Microsoft Xbox One Kinect: concurrent validity and inter-day reliability of spatiotemporal and kinematic variables, J. Biomech. 48 (10) (2015) 2166–2170.

[13] F. Dierick, M. Penta, D. Renaut, C. Detrembleur, A force measuring treadmill in clinical gait analysis, Gait Posture 20 (3) (2005) 299–303.

[14] S. Crea, M. Donati, S. Marco, M.D. Rossi, C.M. Oddo, A Wireless Flexible Sensorized Insole for Gait Analysis, 2014, pp. 1073–1093.

[15] F.D. Groote, T.D. Laet, I. Jonkers, J.D. Schutter, Kalman smoothing improves the estimation of joint kinematics and kinetics in marker-based human gait analysis, J. Biomech. 41 (16) (2008) 3390–3398.

[16] D. Tarniţă, Wearable sensors used for human gait analysis, Rom. J. Morphol. Embryol. 57 (2) (2016) 373–382.

[17] C. Tunca, N. Pehlivan, N. Ak, B. Arnrich, G. Salur, C. Ersoy, Inertial sensor-based robust gait analysis in non-hospital settings for neurological disorders, Sensors (Switzerland) 17 (4) (2017) 1–29.

[18] S.D. Din, M. Elshehabi, B. Galna, M.A. Hobert, E. Warmerdam, U. Suenkel, K. Brockmann, F. Metzger, C. Hansen, D. Berg, L. Rochester, W. Maetzler, Gait Analysis with Wearables Predicts Conversion to Parkinson Disease, 2016, pp. 357–367.

[19] Technaid, [link], https://www.technaid.com/.

[20] R.E. Kalman, A new approach to linear filtering and prediction problems, J. Fluids Eng., Trans. ASME 82 (1) (1960) 35–45.

[21] J. Sanchez-Casanova, J. Liu-Jimenez, P. Fernandez-Lopez, R. Sanchez-Reillo, Recurrent neural network for gait pathology detection, in: BIOSIGNALS 2020 - 13th International Conference on Bio-Inspired Systems and Signal Processing, Proceedings; Part of 13th International Joint Conference on Biomedical Engineering Systems and Technologies, BIOSTEC 2020 (2020) 60–67.

[22] D.-s. Management, S. Homes, Design and implementation of cloud analytics-assisted smart power meters considering advanced artificial intelligence as edge analytics in demand-side management for smart homes, Sensors (Basel) 19 (9) (2019) 2047.

[23] O. Karan, C. Bayraktar, H. Gümüşkaya, B. Karlik, Diagnosing diabetes using neural networks on small mobile devices, Expert Syst. Appl. 39 (1) (2012) 54–60.

[24] Z.H. Zhou, Y. Jiang, Y.B. Yang, S.F. Chen, Lung cancer cell identification based on artificial neural network ensembles, Artif. Intell. Med. 24 (1) (2002) 25–36.

[25] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.