Using Ground Reaction Forces To Predict Fatigue In Treadmill Running: A Machine Learning Approach

Sérgio Baldo Junior
University of Sao Paulo
Thiago Faria dos Santos
University of Sao Paulo
Renato Tinós
University of Sao Paulo
Paulo Roberto Pereira Santiago (paulosantiago@usp.br)
University of Sao Paulo

Research Article

Keywords: running, Machine, patterns, fatigue

Posted Date: December 28th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-1197224/v1

Using Ground Reaction Forces to Predict Fatigue in Treadmill Running: a Machine Learning Approach

Sérgio Baldo Junior[1], **Thiago Faria dos Santos**[2], **Renato Tinós**[1,+], **and Paulo Roberto Pereira Santiago**[2,+,*]

[1]Department of Computing and Mathematics, FFCLRP, University of São Paulo, Ribeirão Preto, 14040-901, Brazil
[2]School of Physical Education and Sport of Ribeirão Preto, University of São Paulo, Ribeirão Preto, 14040-907, Brazil
[*]paulosantiago@usp.br
[+]these authors contributed equally to this work

ABSTRACT

The analysis of running patterns, especially those associated with fatigue, can help specialists in designing more efficient workouts and preventing injuries in high-performance sports. However, classifying running patterns is not trivial for humans. An interesting alternative is to use Machine Learning methods, such as Artificial Neural Networks (ANNs), to classify running patterns. In this work, ground reaction forces are measured by sensors coupled to the base of a low-cost open-source treadmill. ANNs are used to classify the force signals and to indicate the occurrence of fatigue. Different features, extracted from the force signals, are proposed and investigated. A Genetic Algorithm (GA) is used to select the best features. The experimental results indicate that the ANN is able to classify the running patterns with good accuracy. In addition, some features selected by the GA provide important information regarding the identification of fatigue in treadmill running.

1 Introduction

The number of people who adopt running as a form of sports practice has grown in recent years. The increase in amateur runners is primarily related to the physical, mental, financial, and ease-of-access benefits of running[1]. Many professional athletes, from the most diverse sports, also practice running because of its many benefits. However, in contrast to its many benefits, running can often cause musculoskeletal injuries[2]. There is scientific evidence indicating that the impact of the foot on the floor is directly related to the chances of injury in joggers[3]. This observation denotes the importance of a careful analysis of the ground reaction forces, mainly as a possible variable of control of the intensity of the run during training sessions of amateur and professional athletes[4].

In addition, the impact of the foot on the floor is also related to energy savings and performance[5,6]. Researchers[7,8] suggested that the pattern of running with the foot on the "forefoot" and bare feet, i.e., without the use of specific running shoes, is more efficient compared to the "hindfoot" pattern with footwear. These studies analyzed the running patterns efficiency based on energy savings, propulsion performance, and lower risk of injury.

In their most recent study, Lieberman et al.[8] supported their statements by investigating the "transient impact" measure when the foot touches the floor. The authors found that the values of transient impact were significantly lower when adopting a "forefoot" step pattern, even without shoes with cushioning technology. Following this trend, sports shoe industry giants created minimalist models for forefoot runners based on the barefoot running concept popularized by Lieberman and collaborators.

The previous studies make evident the importance of studying patterns of running through innovative methods and based on different measures. Progress in this direction is increasingly present in the leading research laboratories and the main training centers in the world. The ability to quickly identify changes in running patterns caused by fatigue is of great importance. Such identification can assist in the development of more efficient training and the prevention of injuries. However, detecting changes in the running movement is usually a complex task, as it requires a high level of perception from the observer/trainer. Moreover, there is no decisive marker to define fatigue.

Given the complexity of running patterns associated with fatigue, some solutions that use computational models for the automatic classification of running patterns have recently emerged. Pereira et al.[9] proposed a computational model based on Complex Networks to verify changes in body patterns during treadmill running. The proposed model used many variables related to running and was employed in the analysis of the occurrence of fatigue. It is worth mentioning that the study did not aim to identify changes that preceded the occurrence of fatigue. Furthermore, the variables used in the study were not processed

in real-time during running training. The application of the method during runner training sessions is therefore limited, unlike methods that provided auditory[10] or visual feedback[11] to afford improvements in running biomechanics.

Machine Learning techniques operating in real-time are an interesting approach for verifying pattern changes during running. This works proposes the use of an Artificial Neural Network (ANN) to classify running patterns online. ANNs have been widely used in numerous applications, with emphasis on those that involve approximation of functions, prediction of time series, classification, and processing of natural language[12]. Here, the inputs of the ANN are features extracted from ground reaction forces captured by sensors (load cells) coupled to the base of a low-cost open-source treadmill. The outputs of the ANN indicate the occurrence of fatigue. The ANN used in this work is an Multilayer Perceptron (MLP). Different features, extracted from the force signals, are proposed and investigated.

Different features, extracted from the force signals, are proposed and analyzed. A Genetic Algorithm (GA) is used as a search method to select the most significant features for the classification done by the ANN. GAs are population meta-heuristics that have been successfully employed in feature selection in many Machine Learning problems[13]. Here, the analysis of relevant features can also be important for understanding the main characteristics for detecting fatigue in treadmill running.

The remainder of the paper is organized as follows. Section 2 presents the proposed methodology. Section 3 presents the experimental results. Finally, Section 4 presents the discussions of results and the conclusions.

2 Methods

The objective of this study is to classify running patterns from force signals emitted by load cells attached to an open-source treadmill. The classification of fatigue is done according to the runner's perceived exertion. Figure 1 shows the steps of the proposed methodology; these steps are presented in the following.

2.1 Open-Source Treadmill

A low-cost and open-source treadmill was designed and built at the Biomechanics and Motor Control Laboratory (LaBioCoM), of School of Physical Education and Sport of Ribeirão Preto, in a previous work supervised by one of the authors (Paulo R. P. Santiago). The treadmill is composed of: i) upper metallic structure (where are fixed the engine, rollers, and canvas); ii) lower metallic structure (where are fixed the force and tilt control modules); iii) rollers; iv) canvas; v) engine support.

In order to collect the ground reaction forces produced by the runner, four CTR load cells (*MK Controle e Instrumentação LTDA*, São Paulo, Brazil) were attached to the base of the open-source treadmill. The capacity of each load cell is 1-ton force. The force signals captured by each load cell are amplified by an MKTC-05 amplifier (*MK Controle e Instrumentação LTDA*, São Paulo, Brazil). Then, the signals are sent by different channels to an analog/digital data acquisition board (National Instruments, Texas, USA) that records data at the 1 GHz frequency. Finally, MATLAB (MathWorks, Massachusetts, USA) with the Data Acquisition (DAQ) package is used for pre-processing.

2.2 Dataset and Pre-Processing

The dataset used for training and testing the MLP is obtained in running sessions performed in 5 days. A volunteer performed 60-second training sessions on the treadmill, maintaining a constant speed of 10 km/h. The experiments were performed within the ethical standards set out in Resolution 466/12, of the National Health Council of 12/12/2012 (BRASIL, 2012) and the Resolution of Helsinki (2001). The research was approved by the Research Ethics Committee of the School of Physical Education and Sport of Ribeirão Preto, the University of São Paulo (opinion id: 2.311.522; CAAE: 70232317.2.0000.5659). A code written in Python3.8 was developed for pre-processing the ground reaction forces signals.

To smooth out the noise present in the signals, a 4th-order digital Butterworth filter with a low-pass cut-off frequency of 59 Hz, available at the Scipy Library, was used. After applying the filter, the signals were normalized regarding the runner's body weight (BW). Then, the signals from the four sensors were added together, resulting in a single time series for each running session. The next step was to selected the last 45 seconds of each session in order to remove artifacts. Finally, we automatically split the signals in two: the signal generated by the right foot and the signal generated by the left foot. This is necessary because the contact with the right foot on the treadmill generates a higher force signal on sensors 1 and 2, while the contact with the left foot generates a higher force signal on sensors 3 and 4.

The MLP is trained by backpropagation[12]. The desired outputs (labels) are generated according to the Borg Rating of perceived exertion[14] during the running sessions. At the end of each session, the volunteer informed his perception of effort during the exercise (label). The scale used for the perceived exertion ranges from 0 to 10. Two classes were defined based on the scale values: i) "rested" for values from 0 to 7; ii) "tired" for values from 8 to 10.

The input of the MLP is a features vector extracted from the pre-processed force signals. Each time series generated in each running session is split according to the steps of the runner. This is automatically done by the code developed by us. Every step of the runner on the treadmill generates a time series window. Features are extracted from each time series window, i.e., each step of the runner generates a subset of features. The MLP uses as inputs the features extracted by the two feet. In other words,

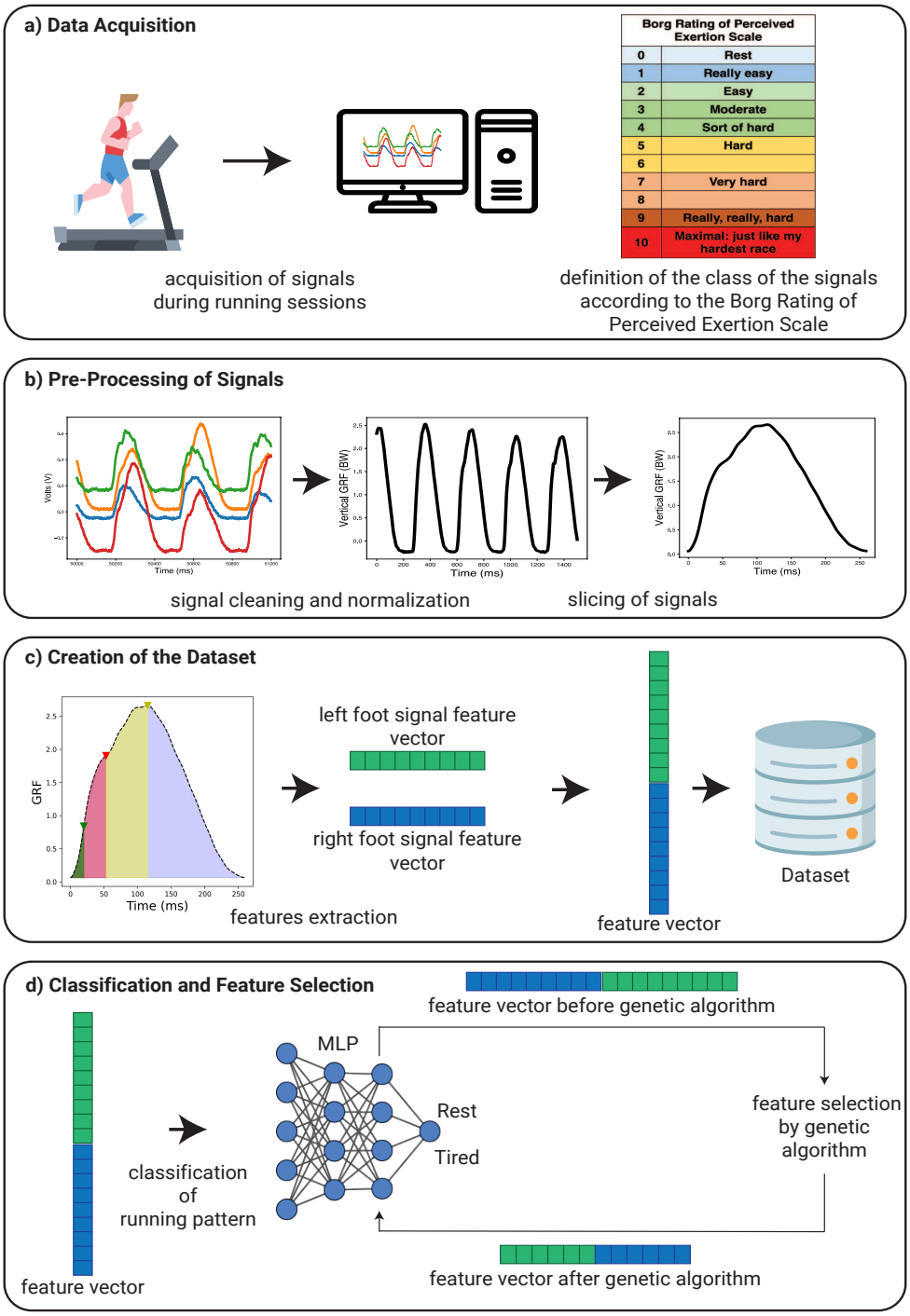


Figure 1. Steps of the methodology proposed. **a) Data acquisition:** acquisition of signals emitted by the load cells attached to the treadmill during 60 s running sessions; at the end of each session, the runner indicates the perceived exertion in order to label the examples used for training and testing the ANN. **b) Pre-processing of signals:** noise is removed from the signals, that are then normalized, and sliced one by one. **c) Creation of the dataset:** Extraction of the features of two consecutive signals: a force signal produced by the right foot and a force signal produced by the left foot. A single example of the dataset is composed of features extracted from two consecutive signals. **d) Classification:** the input of the MLP is a feature vector. Feature selection is performed by the GA in order to improve the classifier performance. The selected features are then used to train and test the MLP.

an example of the dataset is composed of features extracted from a time series window generated by a step of the right foot and features extracted from a time series window generated by a (consecutive) step of the left foot.

The MLP generates a classification ("rested" or "tired") for every two steps of the runner on the treadmill, i.e., the features extracted on two consecutive steps are averaged and presented to the MLP. Each running session generated an average of 62 examples. Different running sessions were performed in each of the five days. A total of 3.870 examples were obtained, with 2.558 labeled as "rested" and 1.312 labeled as "tired". Table 1 presents the number of running sessions performed each day for each class (label) .

Day	Number of Running Sessions	Class (label)		Number of Examples	
		Rested	Tired	Rested	Tired
1	12	8 sessions	4 sessions	513	251
2	12	7 sessions	5 sessions	436	316
3	13	9 sessions	4 sessions	558	250
4	12	8 sessions	4 sessions	495	246
5	13	9 sessions	4 sessions	556	249

Table 1. The number of running sessions performed each day and the label and number of examples from each running session.

2.3 Features Extraction

We use two strategies for defining the features extracted from the force signals: i) interview with a researcher, specialist in the field of sports, and study of the literature related to the analysis of running patterns[8,15]; ii) study of the literature related to the classification of similar time series generated by human bodies, more specifically, the extraction of characteristics of electrocardiogram (ECG) signals for the classification of cardiac arrhythmia[16]. We extracted two types of features: i) features related to the shape of the signals, and ii) features extracted using Fast Fourier Analysis.

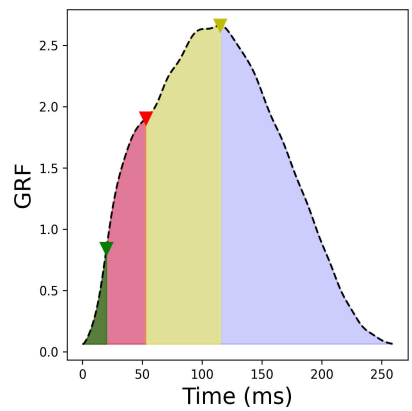
In their study, Lieberman et al.[8] used the "transient impact" measures to investigate the relationship between foot touching the floor in habitually barefoot versus shoe runners. In[15], researchers analyzed the value of impact transient and other features to determine whether differences in running biomechanical patterns exist between female runners with a history of prior Tibial Stress Fracture and those who have never suffered this injury. The studies show that the variables related to "transient impact" are helpful to analyze the biomechanical patterns in runners. Therefore, we extracted features related to the "transient impact" in order to classify the running patterns and predict the fatigue in runners.

Here, we propose 15 shape features (for each foot). These features are computed based on peaks in the ground reaction force (GRF) curves. These peaks are related to points of the foot that touch the floor during the running. The triangles, distances, and areas in the example of Figure 2.a shows how the shape features are computed. They are:

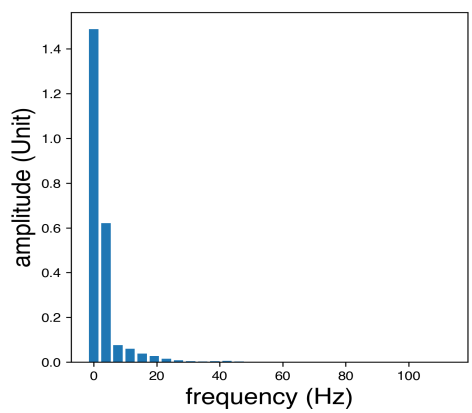
- Value (magnitude) of the maximum peak (yellow triangle, vertical axis);
- Time to the maximum peak (yellow triangle, horizontal axis);
- Value of the first transient impact (red triangle, vertical axis);
- Time to the first transient impact (red triangle, horizontal axis);
- Value of the most inclined point (green triangle, vertical axis);
- Time to the most inclined point (green triangle, horizontal axis);
- Number of peaks;
- Duration of the foot contact (distance from the beginning to the end of the foot contact, horizontal axis);
- Integral of the curve of the foot contact (areas: green + red + yellow + blue);
- Integral of the curve up to the maximum peak point (areas: green + red + yellow);
- Integral of the curve up to the first transient impact (areas: green + red);
- Integral of the curve up to the greatest inclination over time (green area);

- Integral of the curve from the first transient impact to the the maximum peak (yellow area);
- Integral of the curve from the most inclined point to the first transient impact (red area);
- Integral of the curve from the maximum peak to the end of the foot contact (blue area)

Ten features are extracted by using the Fast Fourier Transform. They correspond to the first ten amplitudes (for the first 10 frequencies) of the Fourier Transform (Figure 2.b). A code written in Python3.8 is used to extracted all 50 features (25 for each foot).



(a) Shape features for the ground reaction force (GRF) signal.



(b) Example of the application of the Fast Fourier Transform.

Figure 2. Features extracted from the signal. **a)** Shape features. **b)** Fourier transform features.

2.4 Classification and Feature Selection

2.4.1 Performance Metrics and Cross-Validation

Different performance metrics were used to test the MLP. As an alternative to using MLP to classify the running patterns, we used a Decision Tree as a classifier. Thus, the performance metrics were also used to compare the MLP and the Decision Tree.

The performance metrics are based on: i) the number of correctly classified examples of the class "tired" (true positives, tp); ii) the number of correctly classified examples of the class "rested" (true negatives, tn); iii) the number of "tired" examples that were incorrectly classified as "rested" (false positive, fp); iv) the number of "rested" examples that were incorrectly classified as "tired" (false negative, fn). Based on them, we computed the accuracy (ACC), the true positive rate or sensitivity (TRP), the specificity or true negative rate (SPC), and the F1-score ($F1$)[17]:

$$ACC = \frac{tp+tn}{tp+fn+fp+tn} \tag{1}$$

$$TPR = \frac{tp}{tp + fn} \tag{2}$$

$$SPC = \frac{tn}{fp + tn} \tag{3}$$

$$F_1 - score = \frac{tp}{tp + \frac{fn+fp}{2}} \tag{4}$$

In order to evaluate the results of the classifiers, k-fold Cross-Validation was used. In k-fold Cross-Validation, the dataset is randomly divided into k folds. Then, the Machine Learning model is trained with k-1 folds and tested with one fold. This process is repeated until all folds are tested. In order to reproduce a real-world situation, we consider Cross-Validation with 5 folds, where the data obtained in all the sessions of each the day make up each fold. Figure 3 shows how the Cross-Validation was applied. The performance metrics were computed based on the confusion matrix obtained in 5-fold Cross-Validation. To handle the unbalanced dataset, performance metrics incorporated weights.

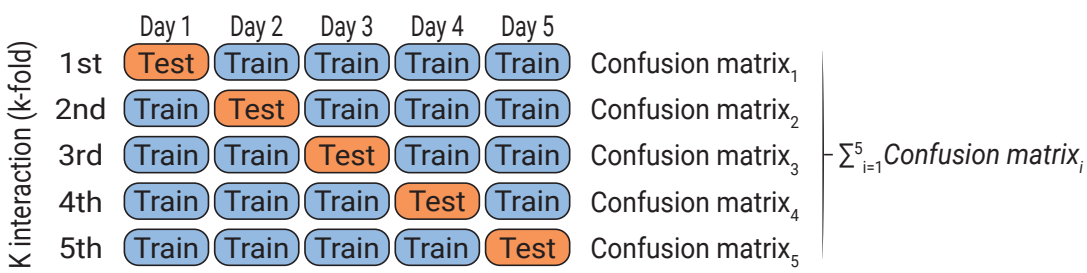


Figure 3. Each fold in the cross validation is made up of data from running sessions on a different day. Thus, the model is trained and tested 5 times. A confusion matrix is generated for each training. The final confusion matrix, used to compute the performance metrics, is obtained by summing the individual confusion matrices.

2.4.2 Multilayer Perceptron

The MLP, used for classifying the running patterns, was developed in Python3.8 by using TensorFlow. Initial experiments (not shown here) were run in order to set the initial hyper-parameters of the MLP. The MLP hyper-parametes, used in the experiments presented in Section 3, are shown in Table 2.

Hyper-Parameter	Values
Optimizer	Adam
Activation Function	Relu
Number of Hidden Layers	1
Number of Neurons in the Hidden Layer (n_h)	$n_h = n_a + 2$, where n_a is the number of inputs (features)
Loss Function	Binary cross entropy
Learning Rate	0.001
Batch Size	64
Epochs	500

Table 2. Hyper-parametes of the MLP.

2.4.3 Genetic Algorithm

The wrapper approach is used for selecting features for the classifier (MLP). In this approach, an optimization algorithm finds subsets of features that, when used by the classifier, minimize the classification error. In this work, we use a GA, implemented in Python3.8, as the optimization algorithm for the feature selection problem. The characteristics of the GA used in this work are:

- **Codification:** The chromosome of each individual of the population is a binary vector indicating the selected features. The i-th element of the chromosome indicates if the i-th feature (of the set of features) is selected (1) or not (0). .

- **Evaluation:** In order to evaluate the subset of features codified in the chromosome of the individual, we first train the MLP by using the examples for the first two days. Then, the MLP is tested with the examples of the third day. The fitness of the individual with chromosome **c** is composed of two terms:

$$f(\mathbf{c}) = ACC(\mathbf{c}) \times \alpha + (1 - \frac{n}{N}) \tag{5}$$

  where $ACC(\mathbf{c})$ is the accuracy of the MLP with features given by **c**, α is parameter that controls the relative importance of each of the two terms, n is number of selected features (number of 1´s in **c**), and N is the total number of features in the original set of features. The second term of Eq. 5 is higher for subsets with fewer features. The fitness function should be maximized by the GA; thus, we want to find subsets with few features that result in high accuracy.

- **Reproduction and Selection**: Individuals are reproduced by two-point crossover and flip mutation. Tournament selection, where the best of k_t randomly chosen individuals is selected, is applied. Elitism is also applied to select the best individual in the population.

Based on preliminary experiments, the following parameters were chosen: number of generations equal to 100, population size equal to 100, $\alpha = 4.5$, probability of crossover equal to 0.6, probability of mutation equal to $\frac{1}{n}$, where n is number of selected features (number of 1´s in **c**) , and $k_t = 2$.

2.5 Sliding Window Technique

The proposed classification system presented in the previous sections returns an output ("tired" or "rested") for each two steps of the runner. The classification can be improved when two or more consecutive outputs are taken in account. The data collected in a running session is a time series, where we expect that the Borg Rating of Perceived Exertion does not change very much in consecutive steps. Thus, a sliding window technique with overlap, where the classification is performed considering windows of outputs, is tested in the experiments. The application of this technique is illustrated in Figure 4 where X represents the sliding window size and X_{inc} represents the increment for the next sliding window.

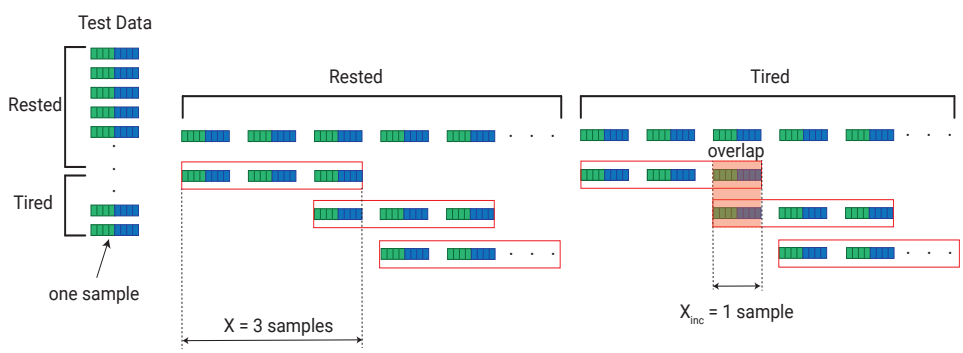


Figure 4. Example of the sliding window technique.

3 Results

In this section, we present results of three experiments. In the first one, the MLP with all features presented in Section 2.3 is tested. Then, results of the MLP with features selected by the GA are presented. Finally, the MLP is compared to another Machine Learning model.

3.1 MLP with all features

The performance of MLP using all features with and without the sliding window technique is presented in Table 3. The small true positive rate (TRP) indicates that the MLP (without the sliding window) does not classify well the examples with the label

Without Sliding Window Technique				With Sliding Window Technique			
ACC	TRP	SPC	F1-score	ACC	TRP	SPC	F1-score
0.68	0.52	0.80	0.58	0.76	0.61	0.87	0.68

Table 3. Performance of the MLP with all features.

"tired". However, the sliding window technique improve the performance of the classifier and decrease errors that occur in some specific positions. As a result, all performance metrics are substantially improved.

Figure 5 shows the percentage of errors of the MLP according to the values of the Borg Rating of Perceived Exertion Scale (RPE) for each example. One can remember that the value of RPE at the end of each running session is used to create the label of the training and testing examples; examples are labeled as “rested” for RPE from 0 to 7 and “tired” for RPE from 8 to 10. The percentage of errors is obtained by computing the number of errors that occur in the classification of examples matching a certain value of RPE. The figure shows that the number of errors for the class "rested" increases when the RPE increases. The maximum percentage of errors occurs when RPE is equal to 7, that is the maximum value for the class "rested".

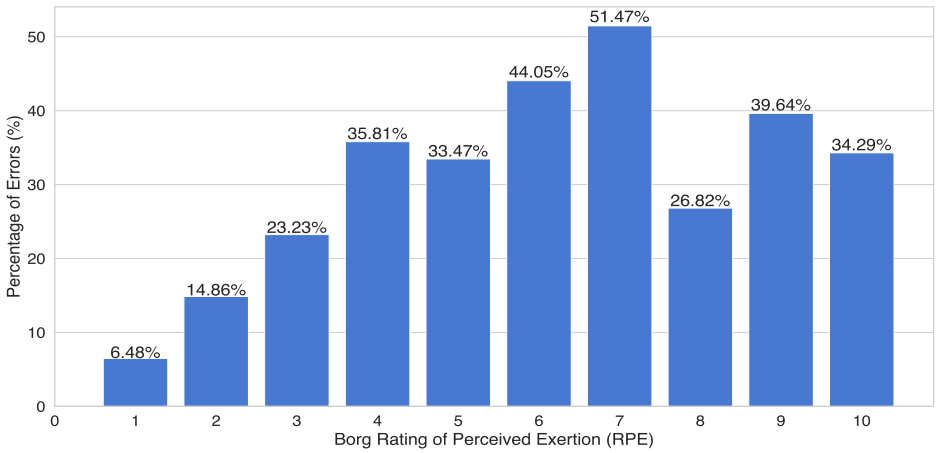


Figure 5. Percentage of errors regarding the Borg Rating of Perceived Exertion Scale (RPE).

3.2 MLP with features selected by the GA

The GA was run ten times with different random seeds. At the end of each run, the feature vector for the best individual was selected. Feature vectors obtained in different runs of GA are generally different local optima; so we analyzed the frequency that each feature appeared in the best individuals obtained in the different runs. Then, the features that appeared more often in the best individuals for the different run were selected. This new feature vector are then used in the MLP.

We tested the MLP with "10%" to "90%" of the most frequent features. For example, when the frequency was "50%", 4 features appeared in the best individuals of at least 5 out of 10 runs of the GA. Table 4 shows the performance of the MLP using feature vectors considering different frequencies. The best performance was obtained for frequency equal to 40%, when 5 features were used by the MLP. They are: i) Value of the first transient impact for the right foot; ii) Third amplitude of the Fourier Transform for the right foot; iii) Number of peaks for the left foot; iv) Integral of the curve from the first transient impact to the the maximum peak for the left foot; v) Eighth amplitude of the Fourier Transform for the left foot.

3.3 Comparison to other Machine Learning model

The performance of a Decision Tree, implemented in SciKit-Learn with the standard hyper-parameters, was compared to the performance of MLP. The Decision Tree is a symbolic model that is one of the most popular Machine Learning models. Table 5 shows the performance of the Decision Tree and the MLP with all features. Moreover, one can observe that the MLP presents a better performance than the Decision Tree.

Number of features	Frequency	Without Sliding Window Technique				With Sliding Window Technique			
		ACC	TRP	SPC	F1-score	ACC	TRP	SPC	F1-score
19	>= 10%	0.62	0.46	0.76	0.53	0.69	0.53	0.82	0.60
14	>= 20%	0.63	0.47	0.76	0.53	0.69	0.52	0.81	0.59
9	>= 30%	0.61	0.45	0.75	0.51	0.68	0.51	0.80	0.57
5	>= 40%	0.65	0.49	0.77	0.54	0.72	0.56	0.81	0.61
4	>= 50%	0.60	0.43	0.73	0.48	0.64	0.46	0.76	0.51
4	>= 60%	0.60	0.43	0.73	0.48	0.64	0.46	0.76	0.51
2	>= 70%	0.62	0.45	0.74	0.50	0.69	0.52	0.78	0.55
2	>= 80%	0.62	0.45	0.74	0.50	0.69	0.52	0.78	0.55
1	>= 90%	0.66	0.49	0.69	0.34	0.68	0.59	0.68	0.13

Table 4. Performance of the MLP with features selected by the GA.

Model	Without Sliding Window Technique				With Sliding Window Technique			
	ACC	TRP	SPC	F1-score	ACC	TRP	SPC	F1-score
Multilayer Perceptron	0.68	0.52	0.80	0.58	0.76	0.61	0.87	0.68
Decision Tree	0.60	0.42	0.71	0.44	0.67	0.51	0.73	0.46

Table 5. Performance of the Decision Tree and MLP with all features.

4 Discussion and Conclusions

In the present study, we proposed to use an MLP to classify running patterns associated with fatigue on a low-cost open-source treadmill. Features extracted from force signals emitted by load cells are used as inputs by the MLP. We proposed the use of different features for this task. In order to improve the performance of the classifier and reduce the number of inputs of MLP, we used a GA for wrapper feature selection.

The MLP presented better results that the Decision Tree regarding all performance metrics (Table 5). According to Table 3 the MLP with all features obtained 0.68 of accuracy and 0.80 of specificity. However, the sensitivity of the MLP was just 0.52 and the F1-score was 0.58. This result shows that the MLP (without the sliding window technique) was not efficient in classifying samples of the "tired" class. In order to improve the metrics obtained by the classifier, we used a sliding window technique to minimize the errors that occur in some specific examples that have similar consecutive outputs. When sliding window technique was used, the MLP improved all performance metrics (ACC=0.76, TRP=0.61, SPC=0.87, and F1=0.68).

When the errors where analyzed according to the Borg Rating of Perceived Exertion Scale - RPE (Figure 5), it is possible to observe that the number of errors increased when RPE increased for the class "rested". The highest number of errors occurred for RPE=7, that was the highest value of RPE for the class "rested". These results indicate that the subjective rating for the perceived exertion performed by the volunteer caused some errors for examples of the class "rested" obtained close to the transition to the class "tired". This is an interesting finding, indicating that a more robust labeling technique can be used in the future to improve the proposed method. Another possible future work in this direction is to consider the problem as a regression instead of classification problem; in this case, the MLP will be used to predict the RPE instead of classifying between two classes ("rested" and "tired").

Regarding the feature selection experiments, the best performance was obtained by an MLP with only 5 features (Table 3). The performance of the MLP with feature selection, when compared to the MLP with all features, were worse (ACC=0.72, TRP=0.56, SPC=0.81, and F1=0.61). However, it is surprising that with just 5 out of 50 features, the MLP was able to obtain an accuracy almost as good as when all features were used. The selected features also helps understanding the identification of fatigue in treadmill running. The best feature vector (Section 3.2) had two features that represent the slope of the curve generated by the contact of the feet on the floor: i) Value of first transient impact for the right foot; ii) Integral of the curve from the first transient impact to the the maximum peak for the left foot. These findings suggest that there is a significant difference between the slope of the curves when the runner is rested and tired. Moreover, the value of the first transient impact for the right foot corresponds to the feature used by Lieberman et al. (2010)[8] to study the foot strike pattern in runners. Finally, the number

of peaks is related to the different points of the foot that touch the ground; when the runner is tired, this pattern changes, i.e., the number of peaks is an important marker for differentiating the running patterns.

The results show that the MLP is able to classify the running patterns by using the ground reaction forces signals. This work suggests new possibilities to assist in the development of more efficient training and the prevention of athlete's injuries. In the proposed method, the dataset is composed of examples generated in running sessions of just one user. However, the same methodology can be used in the future to classify running patterns of different users. The impact of classifying running patterns of different users should be investigated in the future. Also, we can consider different labels for the running patterns in the future. Finally, other Machine Learning models and features can be considered.

References

1. Lakomy, H. K., Campbell, I. & Williams, C. Treadmill performance and selected physiological characteristics of wheelchair athletes. *Br. J. Sports Medicine* **21**, 130–133, DOI: https://doi.org/10.1136/bjsm.21.3.130 (1987).
2. Pons-Villanueva, J., Seguí-Gómez, M. & Martínez-González, M. A. Risk of injury according to participation in specific physical activities: a 6-year follow-up of 14 356 participants of the sun cohort. *Int. J. Epidemiol.* **39**, 580–587, DOI: https://doi.org/10.1093/ije/dyp319 (2010).
3. McCormick, F., Nwachukwu, B. U. & Provencher, M. T. Stress fractures in runners. *Clin. Sports Medicine* **31**, 291–306, DOI: https://doi.org/10.1016/j.csm.2011.09.012 (2012).
4. Heise, G. D. & Martin, P. E. Are variations in running economy in humans associated with ground reaction force characteristics? *Eur. J. Appl. Physiol.* **84**, 438–442, DOI: https://doi.org/10.1007/s004210100394 (2001).
5. Gruber, A. H., Umberger, B. R., Braun, B. & Hamill, J. Economy and rate of carbohydrate oxidation during running with rearfoot and forefoot strike patterns. *J. Appl. Physiol.* **115**, 194–201, DOI: https://doi.org/10.1152/japplphysiol.01437.2012 (2013).
6. Ogueta-Alday, A., Rodríguez-Marroyo, J. A. & García-López, J. Rearfoot striking runners are more economical than midfoot strikers. *Medicine Sci. Sports Exerc.* **46**, 580–585, DOI: https://doi.org/10.1249/MSS.0000000000000139 (2014).
7. Bramble, D. M. & Lieberman, D. E. Endurance running and the evolution of homo. *Nature* **432**, 345–352, DOI: https://doi.org/10.1038/nature03052 (2004).
8. Lieberman, D. E. *et al.* Foot strike patterns and collision forces in habitually barefoot versus shod runners. *Nature* **463**, 531–535, DOI: https://doi.org/10.1038/nature08723 (2010).
9. Pereira, V. *et al.* Complex network models reveal correlations among network metrics, exercise intensity and role of body changes in the fatigue process. *Sci Rep* **5**, DOI: https://doi.org/10.1038/srep10489 (2015).
10. Tate, J. J. & Milner, C. E. Sound-intensity feedback during running reduces loading rates and impact peak. *The J. Orthop. Sports Phys. Ther.* **47**, 565–569, DOI: https://doi.org/10.2519/jospt.2017.7275 (2017).
11. Agresta, C. & Brown, A. Gait retraining for injured and healthy runners using augmented feedback: A systematic literature review. *The J. Orthop. Sports Phys. Ther.* **45**, 576–584, DOI: https://doi.org/10.2519/jospt.2015.5823 (2015).
12. Haykin, S. *Neural Networks and Learning Machines* (Pearson, 2009), 3 edn.
13. Xue, B., Zhang, M., Browne, W. N. & Yao, X. A survey on evolutionary computation approaches to feature selection. *IEEE Transactions on Evol. Comput.* **20**, 606–626, DOI: https://doi.org/10.1109/TEVC.2015.2504420 (2015).
14. Borg, G. A. Psychophysical bases of perceived exertion. *Medicine Sci. Sports Exerc.* **14**, 377–381 (1982).
15. Milner, C. E., Ferber, R., Pollard, C. D., Hamill, J. & Davis, I. S. Biomechanical factors associated with tibial stress fracture in female runners. *Medicine science sports exercise* **38**, 323–328, DOI: https://doi.org/10.1249/01.mss.0000183477.75808.92 (2006).
16. Anwar, S. M., Gul, M., Majid, M. & Alnowami, M. Arrhythmia classification of ecg signals using hybrid features. *Comput. Math. Methods Medicine* **2018**, DOI: https://doi.org/10.1155/2018/1380348 (2018).
17. Sokolova, M. & Lapalme, G. A systematic analysis of performance measures for classification tasks. *Inf. Process. & Manag.* **45**, 427–437, DOI: https://doi.org/10.1016/j.ipm.2009.03.002 (2009).

Acknowledgements

This study was partially funded by: the São Paulo Research Foundation (FAPESP) under grants #2019/07665-4, #2019/17729-0, #2020/01022-1, and #2021/09720-2; Pro-Rectory of Research of University of São Paulo – USP ("Support Program for Projects

that Make Use of Intelligent Digital Systems"); Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001; and grant #305755/2018-8, National Council for Scientific and Technological Development - CNPq.

Author contributions statement

S.B.J., R.T, and P.R.P.S. conceived the methodology, planned the experiments, contributed to the data analysis and to the interpretation of the results, and wrote the manuscript. In addition, S.B.J. developed the codes and carried out the experiments. T.F.S. built the hardware (open-source treadmill) and contributed to the data acquisition. All authors provided critical feedback and revised the manuscript.