

Predicting gait events from tibial acceleration in rearfoot running: a structured machine learning approach

Pieter Robberechts^{1*}, Rud Derie^{2*}, Pieter Van den Berghe², Joeri Gerlo², Dirk De Clercq², Veerle Segers², and Jesse Davis¹

¹ Department of Computer Science, KU Leuven
Leuven, Belgium
`first.lastname@cs.kuleuven.be`

² Department of Movement and Sport Sciences, Ghent University
Ghent, Belgium
`first.lastname@ugent.be`

Abstract. Gait event detection of the initial contact and toe off is essential for running gait analysis. Heuristic-based methods exist to estimate these key gait events from tibial accelerometry. These heuristic-based methods are unfortunately tailored to very specific acceleration profiles, which may offer complications when dealing with larger data sets and inherent biological variability. Therefore, the purpose of this study was to compare a previously utilised heuristic method of gait event detection to an original proposed method using a structured machine learning approach. Force-based event detection acted as the criterion measure in order to assess the accuracy of the predicted gait events. 3D tibial acceleration and ground reaction force data from 93 rearfoot runners were captured. A heuristic method and two machine learning methods were employed to derive initial contact, toe off and stance time from tibial acceleration signals. Both machine learning methods significantly outperformed existing heuristic approaches. Furthermore, results indicate that a structured recurrent neural network machine learning model offers the most accurate and consistent estimation of the gait events and its derived stance time during level overground running. The machine learning methods seem less affected by intra- and inter-subject variation within the data, allowing for accurate and efficient automated data output with possibilities for real-time monitoring and biofeedback during prolonged measurements.

1 Introduction

The running gait comprises of the stance and swing phases, separated by two key events: initial contact (IC) and toe off (TO) (Figure 1). Determining the timing of these events allows performing a detailed stride-by-stride analysis of a runner's gait. Moreover, many variables relevant for gait analysis are defined

* PR and RD contributed equally to this work.

with respect to either one or both of these gait events. For example, stance time (ST) is defined as the time span between both events. Therefore, accurate and consistent detection of these gait events is crucial for any gait analysis.

The criterion instrument for gait event detection is a force platform [15]. This expensive device is part of an instrumented runway or treadmill [13,18], hence restricted by its measurement zone. Therefore, body-worn accelerometers and matching detection algorithms have been proposed as an ambulatory method for gait event detection [6].

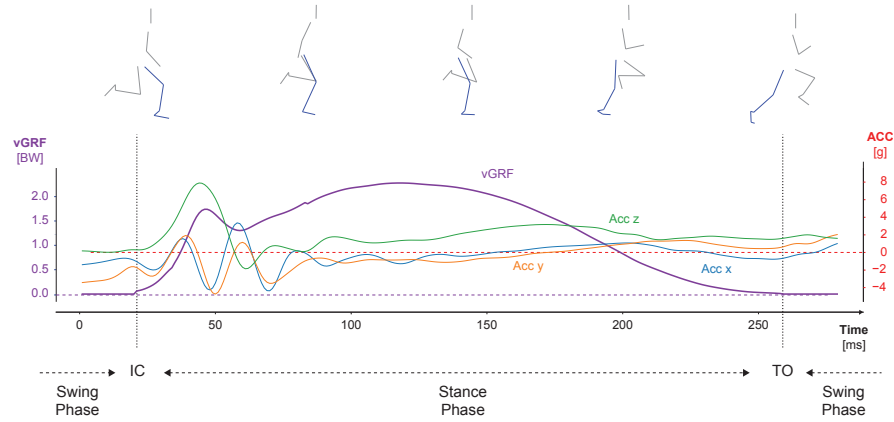


Fig. 1. The running gait cycle can be split in a stance phase and a swing phase, separated by the initial contact (IC) and toe-off (TO) events. The timing of these gait events are estimated from 3D tibial acceleration signals (ACC). Ground truth labels are based on the vertical ground reaction force (vGRF).

However, these accelerometry-based event detection methods are all heuristic-based [10,11,14]. For example, Mercer et al. [11] used an accelerometer affixed to the shin, identifying IC as a local minima before the axial peak tibial acceleration and TO as the second local maxima after that axial peak. These heuristic-based methods assume that both gait events are associated with typical acceleration features, neglecting inter-subject variation. Recent advances in the field of machine learning [9,13], specifically the success of neural networks [5], suggest that a data-driven approach for gait event detection using machine learning may lead to better accuracy and consistency.

This study evaluated gait event detection (IC-TO) from 3D tibial acceleration signals using a heuristic-based method and two machine learning methods. Criterion validation happened by comparing the estimated timings to those determined using a force platform. We evaluated the success of event detection, the absolute error of prediction and its variability to propose an accurate estimation method.

2 Methods

2.1 Subjects, instrumentation and experimental procedure

This study recruited 93 rearfoot runners from the local running community (Table 1). These runners were free of running-related injuries during the last six months, ran at least 15 km per week and signed an informed consent. Approval for the study was obtained from the local ethical committee (bimetra 2015/0864).

Table 1. Characteristics of subjects, expressed as mean \pm standard deviation.

	Men (<i>n</i> = 55)	Women (<i>n</i> = 38)
Age (Yrs)	35.9 \pm 9.2	34.6 \pm 10.8
Body height (m)	1.79 \pm 0.07	1.67 \pm 0.06
Body weight (kg)	76.5 \pm 10.2	60.6 \pm 7.3
Training volume (km/week)	36.4 \pm 16.9	27.9 \pm 11.0

Data collection took place during two different projects, but with the same measurement setup. The first cohort consisted of 13 subjects, wearing a standardized neutral running shoe (Li Ning Magne, ARHF041), who were asked to run on a 30-m instrumented running track at multiple speeds (2.55 m.s⁻¹, 3.20 m.s⁻¹, 5.10 m.s⁻¹ and preferred running speed) [18]. The second cohort consisted of 80 runners, wearing their own regular training shoes, running at 3.20 m.s⁻¹ only. The running speed was controlled with timing gates and runners received feedback if they did not run within \pm 0.2 m.s⁻¹ of the target speed.

All runners wore a backpack/tablet system to measure the tibial acceleration. Two tri-axial accelerometers (LIS331, Sparfukun, Colorado, USA;1000 Hz/axis), were tightly strapped with medical tape on the antero-medial side of both shins, eight centimeters above the medial malleolus [18]. Accelerometers were orientated along the longitudinal axis of the tibia. The skin around the lower leg was pre-stretched with sport tape to improve the rigid coupling between the accelerometers and the tibia. Simultaneously, ground reaction forces were measured at 1000 Hz by two built-in force platforms (2 m and 1.2 m, AMTI, Watertown, MA). Tibial acceleration and force data were synchronized in time by means of an infrared impulse sent from a motion capture system and captured by an infrared sensor at the backpack system [18].

2.2 Data preprocessing

The vertical ground reaction force was filtered using a Butterworth second-order, zero-lag, low-pass filter with a 60 Hz cutoff frequency. For each trial containing at least three steps, the second step in the sequence was extracted using the force data. For each second step, a period ranging from 200 ms before IC to 200 ms

after TO was extracted. The data from the left and right leg were mirrored. Consequently, each of these steps starts with a right foot making ground contact. This procedure resulted in 1003 examples. Tibial acceleration signals were filtered using a second-order band-pass filter with cutoff frequencies of 0.8 Hz and 45 Hz. Using the filter configuration as a hyper-parameter during the learning phase (Section 2.5), this configuration gave the best results.

2.3 Feature construction

For each sampled value of each example, a feature vector was constructed from the x, y and z components of the bi-lateral acceleration profiles. Below we describe the features used in the final models. For the full list of considered features, we refer to the supplementary materials.

Filtered Acc {Left, Right} {x, y, z} The raw values in the filter acceleration signals at each time step.

Filtered Acc {Left, Right} Total The magnitude of the resultant acceleration.

Jerk {Left, Right} {x, y, z} The first derivative of the bandpass-filtered acceleration signals.

Jerk {Left, Right} Total The magnitude of the resultant jerk.

Roll {Left, Right} The roll extracted from the acceleration signals. Here a custom second-order Butterworth low-pass filter at 60 Hz was applied.

Pitch {Left, Right} The pitch extracted from the same low-pass filtered acceleration signals.

Acc Right x Peak Min A moving average filtered labeling of local minima in the anterior-posterior x-component of the foot making ground contact. This marks the neighborhood of a clear peak value for the underlying acceleration signal.

All features were standardized by removing the mean and scaling to unit variance. This scaling happened independently on each feature and independently for each example.

2.4 Gait event detection

Formally, the problem of gait event detection can be specified as:

Given: A 3D tibial acceleration signal $x(t)$ of length l , described by a sequence $\bar{x} = (x_1, \dots, x_l)$ of D-dimensional feature vectors $x_t \in \mathbb{R}^D$ with $t = 1, \dots, l$.

Find: The gait event or phase $y_t \in \{\text{Swing, IC, Stance, TO}\}$ for each corresponding x_t , such that $\bar{y}^* = (y_1, \dots, y_l)$ is the correct sequence of gait events and phases.

In machine learning, this type of problem is traditionally solved by multiclass classification algorithms. In this setting, the task is to find the most likely output label $y \in \{1, 2, \dots, K\}$ for each input $x \in \mathbb{R}^D$. Therefore, the algorithm learns a scoring function f such that $f(x, y^*) > f(x, y)$ for all $y \neq y^*$, where y^* is the true label and y is an imposter label. This scoring function is then evaluated for each possible output label y and the sample is finally labelled with the highest scored output label y^* :

$$y^* = \operatorname{argmax}_{y \in [1, K]} f(x, y). \quad (1)$$

For computational feasibility, the scoring functions generally are a linear form of a joint feature vector $\Phi(x, y)$:

$$f(x, y) = w^T \cdot \Phi(x, y).$$

Here, the features $\Phi(x, y)$ should quantify how “compatible” the input x is with the output y . The vector w are parameters learned from the data that correspond to the weight given to each feature in the computation of the final score. A natural way to represent Φ is an outer product between x and the label space. This yields the following representation:

$$\Phi(x, k) = \left\langle \underbrace{0, 0, \dots, 0}_{D(k-1) \text{ zeros}}, \overbrace{x}^{\in \mathbb{R}^D}, \underbrace{0, 0, \dots, 0}_{D(K-k) \text{ zeros}} \right\rangle \in \mathbb{R}^{DK}, \quad (2)$$

with D the number of features and $k \in y$. In this representation, w effectively encodes a separate weight for every feature/label pair.

In the case of gait event detection, however, the output has a natural structure: IC and TO events alternate each other and the time difference between both events is similar from stride to stride. We can benefit from this inherent structure of the output to train a more accurate predictor [2].

In this “structured prediction” setting a score is similarly assigned to each possible output, given the input. However, both input and output are now sequences instead of individual samples. Specifically, given an input sequence \bar{x} of a tibial acceleration signal and a corresponding possible segmentation $\bar{y} = (\text{Swing, IC, Stance, } \dots)$, the task is to find the element \bar{y}^* of all possible output sequences γ that maximizes a scoring function $w^T \cdot \Phi(\bar{x}, \bar{y})$:

$$\bar{y}^* = \operatorname{argmax}_{\bar{y} \in \gamma} w^T \cdot \Phi(\bar{x}, \bar{y}). \quad (3)$$

However, in the structured setting, every input \bar{x} has many possible segmentations (Figure 2). Therefore, the main challenge is how to efficiently search for the optimal input-output combination. Below, we introduce two machine learning algorithms to solve this problem.

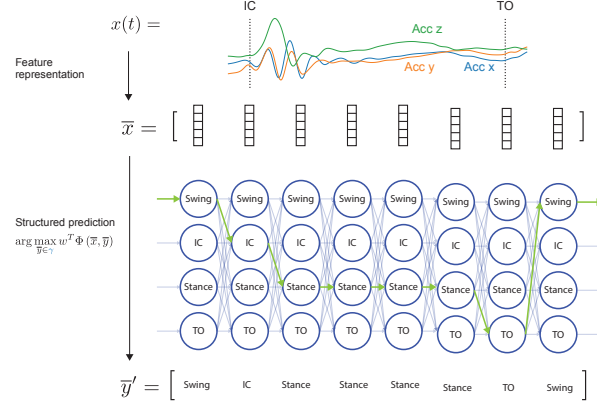


Fig. 2. A graphical illustration of the structured prediction task. First, the acceleration signals of a step are transformed into a feature vector representation. Next, the structured prediction algorithm maps the sequence of input vectors to the most likely gait segmentation sequence. At each time step t the corresponding feature vector x_t can have any of the four possible labels. Any path through this trellis corresponds to a unique labeling of this stride. The gold standard path is drawn with bold green arrows.

2.4.1 Reference and baseline methods

The ground truth of the IC and TO timings was determined per vertical ground reaction force. The threshold for detection was set at 20 N. Additionally, we used the method defined by Mercer et al. [11] (hereafter referred to as the M-method [12]) to set a baseline for the machine learning models. The IC and TO timings were herein defined as the minimum point before positive axial peak tibial acceleration and the minimum acceleration after a second local maximum after positive axial peak tibial acceleration, respectively.

2.4.2 Structured perceptron model

As a simple structured learning algorithm, we used the averaged structured perceptron algorithm [3,4] from the SeqLearn³ package. This is a generalization of the standard perceptron algorithm and the basic building block of the deep learning algorithm introduced in the next section. The algorithm makes predictions by choosing the output sequence \bar{y} that maximizes a score given by $w^T \Phi(\bar{x}, \bar{y})$. And if this output sequence is incorrect, it will adjust the weights w toward the correct output sequence \bar{y}^* .

The key insight is that $\Phi(\bar{x}, \bar{y})$ can be decomposed as $\sum_{i=1}^p \Phi'_i(\bar{x}, y_i)$. This makes it possible to search efficiently over all possible output sequences \bar{y} using a variant of the Viterbi algorithm [7]. The joint feature functions Φ'_i are a

³ <http://larsmans.github.io/seqlearn/>

combination of the *unary features* given by equation 2 and *Markov features* that quantify the likelihood of transitioning from one state to another in the next sample. These are learned from the data and represent for example, that an IC event is always followed by the stance phase.

2.4.3 Structured RNN model

The joint feature vector $\Phi(\bar{x}, y_i)$ can be extracted using different techniques. Usually, this is done by handcrafting features as in the structured perceptron model. An interesting alternative for the gait event detection problem is the usage of a Recurrent Neural Network (RNN) model [1]. RNNs are a deep network architecture that can model the behavior of dynamic temporal sequences using an internal state which can be thought of as memory [8]. RNNs provide the ability to predict the current frame based on the previous and/or next frames. As such, the model can learn which long-term patterns in the acceleration profiles are relevant for determining the timings of the gait events.

We optimized a RNN model and a structured prediction model in an end-to-end fashion. Therefore, we use the structural hinge loss [17]

$$\bar{l}(w, \bar{x}, \bar{y}) = \max_{\bar{y}^* \in \gamma} [0, l(\bar{y}, \bar{y}^*) - w^T \Phi(\bar{x}, \bar{y}) + w^T \Phi(\bar{x}, \bar{y}^*)] \quad (4)$$

with

$$l(\bar{y}, \bar{y}^*) = |y_{IC} - y_{IC}^*| + |y_{TO} - y_{TO}^*|. \quad (5)$$

Since both the loss function and the RNN are differentiable, we can optimize them using stochastic gradient descent.

Specifically, we use the RNN outputs as feature functions for a structured prediction model (Figure 3). First, an RNN encodes the filtered acceleration signals of an entire stride and outputs a new representation for each of the samples. This new representation corresponds to an approximate likelihood of an event. Then an efficient search is executed over all possible sample values so that the most probable one can be selected. Therefore, we use a constrained peak detection algorithm. This algorithm selects the local maxima that satisfy the following constraints:

- A IC and TO event of opposing foot are separated by at least 35 ms and at most 200 ms
- A TO and IC event of the same feet are separated by at least 160 ms and at most 350 ms

2.5 Model learning and performance evaluation

The machine learning models were trained and assessed in a two-step procedure. First, 5-fold cross-validation was used to obtain a good set of features and hyper-parameters for both models. For the perceptron model, the dataset was split into training (83 subjects) and test (10 subjects) sets. We found the 21 features

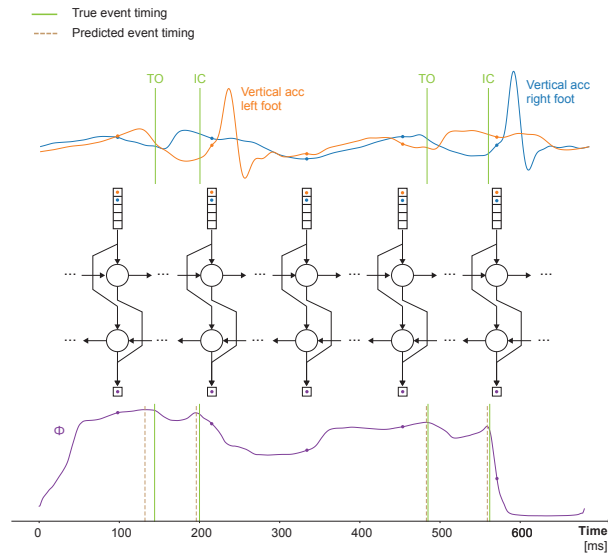


Fig. 3. Visual representation of a two-layer bidirectional recurrent neural network with long short-term memory cells to map the multivariate time series of tibial acceleration profiles (top plot) to the likelihood of an event. Subsequently, a constrained peak detection algorithm is used to determine the most likely timing of IC and TO events. For simplicity, we only plot the axial acceleration profiles.

described in section 2.3 to give the best result. The learning rate was set to 0.1, which is the only parameter of this model. For the RNN model, the dataset was randomly split into training (73 runners), validation (10 runners), and test (10 runners). The validation set was used for early stopping. The same features as in the perceptron model were used, excluding the *Acc Right x Peak Min* feature. Furthermore, we achieved the best results with two bidirectional long short-term memory layers with dropout 0.2 after each recurrent layer and 50 hidden units.

Using these hyper-parameters and features, the models were retrained in a leave-one-out cross-validation analysis to evaluate the accuracy. Each model was iteratively trained on 92 of the 93 test subjects and then the accuracy of the model was tested on the 93th subject. This procedure was repeated 93 times and each time the data of a different subject was left out, obtaining an out-of-sample prediction for each subject’s steps. Doing hyper-parameter tuning and feature selection for each of these 93 folds separately is not computationally feasible, hence the two-step procedure.

For each step, the relative error and absolute error were determined for the estimated IC and TO event timings [12]. Relative errors were calculated as the arithmetic difference (ms) between the predicted event timings (t_{acc}) obtained through the acceleration profiles and reference timings (t_{vgf}) obtained through the force-platform method: Relative error = $t_{acc} - t_{vgf}$. A positive value indicates that the detected event occurred after the reference (time lag). Absolute errors, indicating the error magnitude regardless of direction, were calculated as the absolute value of relative errors: Absolute error = $|t_{acc} - t_{vgf}|$.

ST was determined from the estimated gait events. As for the event timings, relative and absolute errors on the estimated ST were calculated as the arithmetic difference between the estimated ST using the accelerometer-based method and reference. Here, a positive relative error corresponds to an overestimation of the ST.

The number of trials completed by each runner varied. In order to avoid that one runner would excessively impact the accuracy of our models, we computed the global median relative error and median absolute error in a two-step procedure. First, for each runner, the average median absolute error and median relative error were computed over all strides of that runner. Thereafter, the global metrics were calculated as the median values of these metrics over all runners.

2.6 Statistical analysis

Statistical analysis was executed in Python using the *Statsmodels* and *Scipy* libraries, with the significance level set at $p = 0.05$. A Shapiro-Wilk test for normality was first performed on the relative difference of ST. Subsequently, a Friedman test (and Wilcoxon signed-rank tests for comparing pairs of models) and Levene’s test for non-normal distribution were used to examine whether the various prediction methods have significantly different accuracies and standard deviations. Post-hoc testing was conducted using Bonferroni correction. Failed

predictions were imputed with the subject’s average estimated ST at the corresponding running speed. Statistical analysis on the IC and TO estimates was not possible since there exists no logical imputation.

3 Results

The relative differences of ST showed a non-normal distribution (all $p < 0.001$). Both machine learning models outperformed the heuristic M-method (both $p < 0.001$) and significant differences were displayed between the perceptron and RNN methods ($p < 0.001$) (Table 2). Regarding the success of gait detection, our models rarely failed to identify a valid combination of IC and TO event timings. The M-method, on the other hand, often failed to identify these events because the acceleration profile of the failed cases lacked clear local maxima and minima. In the cases with successful identification, the M-method estimated both gait events considerably earlier than the criterion standard (Table 2). The deviations in median relative error of the structured perceptron model and the structured RNN model are close to zero (Table 2). These predicted event timings thus do not consistently lead or lag behind the true event timings.

Table 2. Median relative error (MRE) and median absolute error (MAE) between each estimation method and the reference for initial contact and toe-off detection (–: time lead; +: time lag) and stance time estimation (–: shorter time; +: longer time). Stance time was calculated as the timespan between the initial contact and toe-off events. Median \pm standard deviation.

Variable	Method	Initial Contact	Toe-off	Stance time
MRE (ms)	M-method	–9.0 \pm 9.8	–34.0 \pm 76.3	–26.5 \pm 76.8
	Structured perceptron	0.8 \pm 4.0	0.0 \pm 9.8	–2.8 \pm 10.7
	Structured RNN	–1.0 \pm 2.0	0.5 \pm 4.1	2.0 \pm 4.7
MAE (ms)	M-method	11.5 \pm 6.6	71.8 \pm 63.4	60.0 \pm 58.8
	Structured perceptron	3.5 \pm 3.0	12.8 \pm 7.9	14.5 \pm 9.3
	Structured RNN	2.0 \pm 1.3	3.2 \pm 3.1	4.2 \pm 3.9
Failed predictions (%)	M-method	00.60	21.93	22.53
	Structured perceptron	-	-	2.99
	Structured RNN	-	-	1.69

Bold: minimum MAE for detected IC and TO and estimated stance time.

Regarding the absolute error of estimation, our machine learning models better estimated the IC than the M-method (Table 2). The TO event was harder to estimate. Still, the RNN model clearly outperformed the other methods (Table 2). This RNN model also significantly outperformed the other methods in terms of error variability ($p < 0.001$). Figure 4 shows the risk of an error greater

than a given threshold for the machine learning models and the heuristic M-method. In 83% of the examples, the RNN model was off by at most 10 ms whereas the perceptron model only attained this level of accuracy in 40% of the examples.

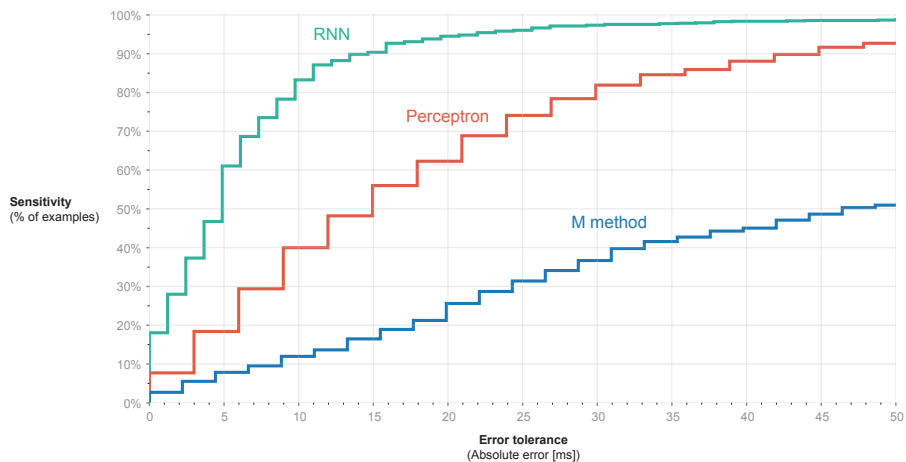


Fig. 4. Sensitivity (true positive ratio) of the three methods as a function of error tolerance. The absolute error (ms) is the deviation from the stance time, as determined by means of a force plate. Both of our machine learning models outperform the heuristic M-method.

4 Discussion

This study examined gait event detection (IC-TO) in 3D tibial acceleration signals of rearfoot runners. Three approaches for event detection were examined: a heuristic-based method [11], a structured perceptron model with hand-crafted features and a deep learning structured RNN model. While the M-method detected the gait events consistently earlier with underestimation of the ST [12], our machine learning models better handled the estimation of IC and TO. The M-method cannot deal with the variation observed in the acceleration signals between subjects and even between strides of a subject. Heuristic-based methods to determine gait events relying on a universal pattern in the acceleration profiles results in a consistent lead (too soon) or lag (too late) in the estimates.

Using a structured learning approach, our machine learning algorithms could deal with some variability in 3D tibial acceleration. Despite accurate predictions for most examples, the errors for an individual step can be quite large. For 3% of the examples, the RNN model predicted a ST that deviates at least 50 ms from

the criterion reference (Figure 4). Most acceleration patterns of these examples belong to the same four subjects and were manually investigated. Unfortunately, no clear patterns were distinguished. One may further improve the models by adding data of a large number of runners with different, unique patterns. Alternatively, a model could be specifically trained for each pattern. Using principles of transfer learning, one can learn such specific models from a limited dataset [16].

The automated methods enable accurate and real-time detection of key events whilst running overground. The mean computation time of the perceptron and RNN models needed to go from the raw signals to estimated gait event timings were respectively 4 ms and 142 ms (2.3 GHz Intel Core i5, 16 GB LPDDR3 RAM), which is below a typical ST in overground rearfoot running at submaximal speed. Given that the prediction models require data of a complete step to make a prediction, estimates can be provided before the end of the next step. This ability to output running gait parameters accurately and promptly permits the development of an automated feedback system based on the consistency or fluctuation of spatio-temporal parameters. Further research should investigate our proposed method when applied outdoors on different terrains. Altogether, this study presents a structured RNN learning approach which accurately detects IC and TO events in 3D tibial acceleration profiles for rearfoot runners when running indoor on a standard sports floor. This algorithm offers possibilities towards implementation in overground gait analysis or gait-retraining of rearfoot runners without the need of an embedded force plate.

Acknowledgements

This work was supported by the H2020 Interreg EU (Nano4Sports project), the Research Foundation Flanders (FWO.3F0.2015.0048.01), the KU Leuven Research Fund (C32/17/036) and the International Society of Biomechanics (matching dissertation grant program 2019).

References

1. Adi, Y., Keshet, J., Cibelli, E., Goldrick, M.: Sequence segmentation using joint rnn and structured prediction models. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 2422–2426. IEEE (2017)
2. Chang, K.W., Kundu, G., Roth, D., Srikumar, V.: Learning and inference in structured prediction models. In: AAAI-16 Tutorial Forum (February 2016)
3. Collins, M.: Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In: Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10. pp. 1–8. EMNLP '02, Association for Computational Linguistics, Stroudsburg, PA, USA (2002). <https://doi.org/10.3115/1118693.1118694>
4. Daume, III, H.C.: Practical Structured Learning Techniques for Natural Language Processing. Ph.D. thesis, University of Southern California, Los Angeles, CA, USA (2006)
5. Deng, L., Yu, D., et al.: Deep learning: methods and applications. *Foundations and Trends in Signal Processing* **7**(3–4), 197–387 (2014)

6. Falbriard, M., Meyer, F., Mariani, B., Millet, G.P., Aminian, K.: Accurate estimation of running temporal parameters using foot-worn inertial sensors. *Frontiers in Physiology* **9**, 610 (Jun 2018). <https://doi.org/10.3389/fphys.2018.00610>
7. Forney, G.D.: The viterbi algorithm. *Proceedings of the IEEE* **61**(3), 268–278 (1973)
8. Graves, A., Fernandez, S., Gomez, F., Schmidhuber, J.: Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In: *Proceedings of the 23rd International Conference on Machine Learning*. pp. 369–376. ACM (2006)
9. Halilaj, E., Rajagopal, A., Fiterau, M., Hicks, J.L., Hastie, T.J., Delp, S.L.: Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities. *Journal of Biomechanics* **81**, 1–11 (Nov 2018). <https://doi.org/10.1016/j.jbiomech.2018.09.009>
10. Lee, J.B., Mellifont, R.B., Burkett, B.J.: The use of a single inertial sensor to identify stride, step, and stance durations of running gait. *Journal of Science and Medicine in Sport* **13**(2), 270–273 (Mar 2010). <https://doi.org/10.1016/j.jsams.2009.01.005>
11. Mercer, J.A., Bates, B.T., Dufek, J.S., Hreljac, A.: Characteristics of shock attenuation during fatigued running. *Journal of Sports Sciences* **21**(11), 911–919 (Nov 2003). <https://doi.org/10.1080/0264041031000140383>
12. Mo, S., Chow, D.H.K.: Accuracy of three methods in gait event detection during overground running. *Gait & Posture* **59**(Supplement C), 93–98 (Jan 2018). <https://doi.org/10.1016/j.gaitpost.2017.10.009>
13. Ngho, K.J.H., Gouwanda, D., Gopalai, A.A., Chong, Y.Z.: Estimation of vertical ground reaction force during running using neural network model and uniaxial accelerometer. *Journal of Biomechanics* **76**, 269–273 (Jul 2018). <https://doi.org/10.1016/j.jbiomech.2018.06.006>
14. Norris, M., Kenny, I.C., Anderson, R.: Comparison of accelerometry stride time calculation methods. *Journal of Biomechanics* **49**(13), 3031–3034 (Sep 2016). <https://doi.org/10.1016/j.jbiomech.2016.05.029>
15. Novacheck, T.F.: The biomechanics of running. *Gait & posture* **7**(1), 77–95 (1998)
16. Pan, S.J., Yang, Q.: A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering* **22**(10), 1345–1359 (Oct 2010). <https://doi.org/10.1109/TKDE.2009.191>
17. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research* **6**(Sep), 1453–1484 (2005)
18. Van den Berghe, P., Six, J., Gerlo, J., Leman, M., De Clercq, D.: Validity and reliability of peak tibial accelerations as real-time measure of impact loading during over-ground rearfoot running at different speeds. *Journal of Biomechanics* **86**, 238–242 (mar 2019). <https://doi.org/10.1016/j.jbiomech.2019.01.039>