Improving state estimation through projection post-processing for activity recognition with application to football

Ciszewski, Michał; Söhl, Jakob; Jongbloed, Geurt

DOI
10.1007/s10260-023-00696-z

Publication date
2023

Document Version
Final published version

Published in
Statistical Methods and Applications

Statistical Methods & Applications (2023) 32:1509–1538
https://doi.org/10.1007/s10260-023-00696-z

ORIGINAL PAPER

Improving state estimation through projection post-processing for activity recognition with application to football

Michał Ciszewski[1] · **Jakob Söhl**[1] · **Geurt Jongbloed**[1]

Accepted: 10 April 2023 / Published online: 26 April 2023

Abstract
The past decade has seen an increased interest in human activity recognition based on sensor data. Most often, the sensor data come unannotated, creating the need for fast labelling methods. For assessing the quality of the labelling, an appropriate performance measure has to be chosen. Our main contribution is a novel post-processing method for activity recognition. It improves the accuracy of the classification methods by correcting for unrealistic short activities in the estimate. We also propose a new performance measure, the Locally Time-Shifted Measure (LTS measure), which addresses uncertainty in the times of state changes. The effectiveness of the post-processing method is evaluated, using the novel LTS measure, on the basis of a simulated dataset and a real application on sensor data from football. The simulation study is also used to discuss the choice of the parameters of the post-processing method and the LTS measure.

Keywords Activity recognition · Wearable sensors · Post-processing · Performance measures

1 Introduction

In almost all areas of science and technology, sensors are becoming more prevalent. In recent years we have seen applications of sensor technology in fields as diverse as energy saving in smart home environments (Lima et al. 2015), performance assessment in archery (Eckelt et al. 2020), detection of mooring ships (Waterbolk et al. 2019), early detection of Alzheimer disease (Varatharajan et al. 2018) and recognition of emotional states (Kołakowska et al. 2020), to name just a few.

✉ Michał Ciszewski
M.G.Ciszewski@tudelft.nl

[1] Applied Mathematics, Delft University of Technology, Mekelweg 4, 2628 CD Delft, The Netherlands

Our main interest lies in the detection of human activities using sensors attached to the body. Sensors generate unannotated raw data, suggesting the use of unsupervised learning methods. If an activity specified in advance is of interest, then supervised learning and labelled data are required. However, the task of labelling activities manually from sensor data is labour-intensive and prone to errors, which creates the need for fast and accurate automated methods.

Human activity recognition (HAR) attracted much attention since its inception in the '90s. A plethora of methods are currently being used to detect human activities (Lara and Labrador 2013), with various deep learning techniques leading the charge (Minh Dang et al. 2020; Wang et al. 2019). In many studies (Ann Ronao and Cho 2017; Capela et al. 2015; Aviles-Cruz et al. 2019) only sensors embedded in a smartphone are used to classify user activities. Physical sensors, such as accelerometers or gyroscopes attached directly to a body or video recordings (from a camera), are the most popular sources of data for activity recognition (Rednic et al. 2012; Zhu and Sheng 2011; Cornacchia et al. 2016). Similarly, cameras can be either placed on the subject (Li et al. 2011; Ryoo and Matthies 2013; Watanabe et al. 2011) or they can observe the subject (Song and Chen 2011; Laptev et al. 2008; Ke et al. 2005). Rarely, both camera and inertial sensor data are captured at the same time (Chen et al. 2015).

The temporal structure of the time series should be taken into account when choosing a method for activity recognition. Simple classification techniques (such as logistic regression or decision trees) ignore time dependencies and will need to be improved after the procedure. Alternatively, methods which are more complicated and more difficult to train have to be deployed. Another challenge lies in the reliability of manual labelling (in case of supervised learning). Quite often it is unreasonable to assume that labels annotating the observed data are exact with regards to timings of transitions from one activity to another (Ward et al. 2006). Timing uncertainty can be caused by a deficiency of the manual labelling or the inability to objectively detect boundaries between different activities. This issue is well-known in the literature, for instance, Yeh et al. (2017) introduced a scalable, parameter-free and domain-agnostic algorithm that deals with this problem in the case of one-dimensional time series.

The main contribution of this paper is the introduction of a post-processing procedure, which improves a result of activity classification by eliminating too short activities. The method requires a single parameter which can be interpreted as the minimum duration of the activites (hence the choice of this parameter is driven by domain knowledge). It allows us to mitigate the problem of activites being fragmented in cases where some domain-specific information about state durations is available. In the current literature, ad hoc techniques are employed for post-processing of human activities and they are particularly suitable when the initial classifier is already performing satisfactory. A method that exemplifies this approach, utilizing majority voting, can be found in the article by Shakerian et al. (2022). Some more advanced approaches have also been devised in special cases, e.g. the approach proposed by Gil-Martín et al. (2020), which is limited to neural network classifiers. In comparison to any existing methods, our post-processing procedure ensures removal of all too short events and allows to specify the minimum length of activities

accepted in the post-processed result. Based on empirical evidence, the performance of classical machine learning classifiers improves significantly by our method. This enables simple and fast but less accurate classification methods to be upgraded to accurate and fast classifiers.

In order to compare the quality of competing activity recognition methods, an appropriate criterion for evaluating the performance is needed (also to demonstrate the performance of the post-processing procedure we introduce). Below are some commonly used performance measures:

- accuracy, precision, the F-measure (Lara and Labrador 2013; Lima et al. 2019),
- similarity measures for time series classification (Serrà and Arcos 2014), such as Dynamic Time Warping or Minimum Jump Costs Dissimilarity,
- custom vector-valued performance metric (Ward et al. 2011).

Our objective is to design a performance measure that satisfies problem-specific conditions, which will be specified later.

The outline of the paper is as follows. Section 2 provides a method for improving classification with a post-processing scheme that uses background knowledge on the specific context. In particular, it validates the state durations and provides an improved classification that satisfies the physical constraints on the state durations imposed by the context. Section 3 introduces specialized performance measures for assessing the quality of classification in general and in activity recognition in particular. The new performance measure also serves the purpose of showing the advantages of the post-processing fairly. Section 4 presents an application of the techniques in a simulated setting. The post-processing method was able to improve the estimates significantly. The method achieves similar results in an application to football data.

2 Improving classification by imposing physical restrictions

2.1 Post-processing by projection

When recognizing human activities, it is often the case that the result of the classification contains *events* (time intervals in which a classification result is constant) that are too short.[1] Usually ad hoc methods are used in order to discard those events, e.g. removal of any short events and replacing them with the next state in the classification, whose length is above a fixed threshold. There are also more advanced approaches, such as the one proposed by Gil-Martín et al. (2020). However, this particular method is suitable only when using a neural network as the classifier of choice, it does not ensure that too short events will always be eliminated (no matter what is exactly meant by 'too short') and lastly does not provide an intuitive understanding of the choice of its tuning parameter. Hence, our interest in a more formal

[1] Depending on the application 'too short' might be specified differently.

method that could be used in combination with any activity classifier. The goal of this section is to introduce a formalized approach to correcting for the classifier's mistakes regarding the activity durations by introducing a novel post-processing procedure.

Consider the set of states $\mathcal{S} = \{1, \ldots, M\}$ and a metric d on $\mathcal{S}$. Let ρ denote the *discrete metric*[2] on $\mathcal{S}$. Any state-valued function of time will be called a *state sequence*. In reality we are only able to obtain a discrete-time signal, however, the relevant information contained in such a signal is a list of all the state transitions, which can more easily be encoded in a function with continuous argument. Hence, we define $\mathcal{T}$, the set of all càdlàg[3] functions $f : \mathbb{R} \to \mathcal{S}$ with a finite number of discontinuities. We define the *standard distance* induced by a metric d between two state sequences as

$$\text{dist} : \mathcal{T} \times \mathcal{T} \ni (f, g) \to \text{dist}(f, g) = \int_{\mathbb{R}} d(f(t), g(t))dt. \tag{1}$$

If d is a metric on $\mathcal{S}$, then dist is a metric on $\mathcal{T}$. The standard distance induced by the discrete metric is the time spent by f in a state different from g.

Now, we define a measure of closeness between functions in $\mathcal{T}$, as our goal is to find a function close enough to a given function in $\mathcal{T}$, while reducing the number of jumps it has (which in turn will eliminate short events in the state sequence). Let $f, g \in \mathcal{T}$. Then we introduce the notation:

$$E_\gamma(f, g) = \text{dist}(f, g) + \gamma \cdot |J(g)|, \tag{2}$$

where $J(g)$ is the set of all discontinuities of g, $|J(g)|$ is the number of all discontinuities of g and γ is a penalty for a single jump of g.

Given $f \in \mathcal{T}$, our goal is to find any solution $\hat{f} \in \mathcal{T}$ of the minimization problem

$$\hat{f} \in \underset{g \in \mathcal{T}}{\arg\min}\ E_\gamma(f, g). \tag{3}$$

As a default, we will use the standard distance induced by the discrete metric.

In order to characterize the solution $\hat{f}$ of problem (3) we present the following lemma.

Lemma 2.1 *Let $\gamma > 0$ and $f \in \mathcal{T}$. Let J denote the set of all discontinuities of the function f. There exists a solution $\hat{f}$ of the problem (3) such that it does not contain jumps outside of J.*

Lemma 2.1 leads to the conclusion that in search for the solution of the minimization problem we can limit ourselves to a finite set of functions, namely a

[2] Distance between two different states is equal 1 and distance from a state to itself is equal 0.

[3] right continuous, left limits exist

subset of $\mathcal{T}$ with jumps only allowed at the same locations as the function f. The proof of lemma 2.1 can be found in the appendix.

In this minimization problem the choice of the parameter γ plays a crucial role. We will now show an interpretation of the penalty parameter that will ease the process of choosing it. It will also allow us to reformulate problem (3). First, we define a new set of functions.

Definition 2.1 (*Function with bounded minimum duration of states*) Given a parameter $\gamma > 0$ we define $\mathcal{G}_\gamma \subset \mathcal{T}$, the set of functions with *bounded minimum duration of states*, such that for $g \in \mathcal{G}_\gamma$ we have

- $g = \sum_{i=1}^{n-1} s_i \mathbb{1}_{[t_i, t_{i+1})}$ for some constant $n \in \mathbb{N}$, a sequence of states $\{s_1, \ldots, s_{n-1}\}$, such that $s_i \neq s_{i+1}$ for $i = 1, \ldots, n-2$, and an increasing sequence $t_1 < t_2 < \cdots < t_n$ (we allow $t_1 = -\infty$ and $t_n = \infty$),
- if $n \geq 2$, then $\forall_{i \geq 2} \quad t_i - t_{i-1} \geq \gamma$.

Lemma 2.2 below yields a connection between the penalty γ and the minimum duration of states that we impose on the solution of our minimization problem.

Lemma 2.2 *Let $\gamma > 0$ and $f \in \mathcal{T}$. Any solution $\hat{f}$ of problem (3) is an element of $\mathcal{G}_\gamma$.*

This lemma can be used in practice to select the size of the penalty. The Proof of Lemma 2.2 can be found in the appendix.

Given $f \in \mathcal{T}$, by Lemma 2.2 the minimization problem (3) is equivalent to the minimization problem

$$\hat{f} \in \operatorname*{arg\,min}_{g \in \mathcal{G}_\gamma} E_\gamma(f, g). \tag{4}$$

$\hat{f}$ will be called a projection of f onto $\mathcal{G}_\gamma$.

As mentioned before, the regularization by penalizing high numbers of jumps narrows down the set of possible solutions to a finite nonempty subset of $\mathcal{G}_\gamma$ (thanks to lemma 2.1), which leads to the existence of $\hat{f}$. However, the solution might not be unique, as illustrated by the following example.

Consider $\mathcal{S} = \{0, 1\}$, $f = \mathbb{1}_{[0.35, 0.45)} + \mathbb{1}_{[0.55, +\infty)}$ and $\gamma = 0.2$. Both $\hat{f}_1 = \mathbb{1}_{[0.35, +\infty)}$ as well as $\hat{f}_2 = \mathbb{1}_{[0.55, +\infty)}$ are projections of f. One could think of it as an issue, however, it reflects well our understanding of the original problem. The assumption is that f has impossibly short windows, because it is uncertain which activity is actually performed in the interval [0.35, 0.55). Looking only at f we are unable to decide which solution is more suitable, hence it is only natural that the method also returns two possible options.

We close with a remark regarding influence of the extreme values of γ on projection $\hat{f}$.

Remark 2.1 Let $f \in \mathcal{T}$. If $\gamma = 0$, then $\hat{f} = f$ is the only projection of f. If $\gamma = \infty$ and $E_\gamma(f, g) < \infty$ for some function $g \in \mathcal{T}$,[4] then g is constant and equal everywhere to the most common state of f and $\hat{f} = f$.[5]

2.2 Connection with the shortest path problem

In this section we devise a method for finding a projection in an efficient manner. It will be shown that the problem of finding the shortest path in a particular graph is equivalent to the minimization problem (4). This is possible thanks to the lemmas 2.1 and 2.2, which narrowed down the set of possible solutions to a finite set.

First, we present a lemma which further characterizes a projection of f.

Lemma 2.3 *Let $f \in \mathcal{T}$. Suppose $f \equiv c$ on an interval $[a, b]$ for some constant $c \in \mathbb{R}$. If $b - a > 2\gamma$, then $\hat{f} \equiv c$ on $[a, b]$. If $b - a = 2\gamma$, then there exists a projection such that $\hat{f} \equiv c$ on $[a, b]$.*

The Proof of Lemma 2.3 can be found in the appendix.

Remark 2.2 If $n > 2$, then there exists a projection such that the second and the second-to-last jump locations of the original function are not the first and the last (resp.) jump locations of this projection.

Remark 2.2 will be used when defining a particular graph and the proof can be found in the appendix.

We will assume that f has $n \geq 2$ jumps[6] at time points t_i for $i = 1, \ldots, n$:

$$f = \sum_{i=0}^{n} s_i \mathbb{1}_{[t_i, t_{i+1})}, \tag{5}$$

where $s_i \in \mathcal{S}$ for $i = 0, \ldots, n$ and $s_i \neq s_{i+1}$ for $i = 0, \ldots, n-1$. We use the following notation: $t_0 = -\infty, t_{n+1} = \infty$. In light of Lemma 2.3 we assume that

$$t_{i+1} - t_i < 2\gamma \tag{6}$$

for $i = 1, \ldots, n-1$. If this is not the case, then consider the coarsest partition of the set J of jumps of f:

$$J = \bigcup_{i=1}^{r} J_i$$

[4] Note that this is not always true. If the first and the last states of f are different, then any function can be a projection of f.

[5] Note that if $E_\gamma(f, g) < \infty$, then the first and the last states of f are the same and the constant function equal to that state is the only projection.

[6] If $n = 0$ or $n = 1$, then $f \in \mathcal{G}_\gamma$ and $\hat{f} = f$.

such that for jumps in J_i for $i = 1, \dots, r$ Eq. (6) holds and $\min J_i - \max J_{i-1} \geq 2\gamma$ for $i = 2, \dots, r$. For each J_i for $i = 1, \dots, r$ consider a function $f_i : \mathbb{R} \to \mathcal{S}$, such that $f_i \equiv f$ on $[\min J_i - 2\gamma, \max J_i + 2\gamma]$ and the only jumps of f_i lie in J_i. Once a projection $\hat{f}_i$ is found for f_i for all $i = 1, \dots, r$, we can then consider a function $\hat{f}$, defined as follows

$$\hat{f}(x) = \hat{f}_i(x) \tag{7}$$

given $x \in [\min J_i - 2\gamma, \max J_i + 2\gamma]$ for some $i = 1, \dots, r$. By Lemma 2.3, there exists a projection which does not change the states longer than or equal 2γ, hence $\hat{f}$ defined as in (7) is a projection of f. Given this remark, we can now assume that f is of the form (5) and satisfies (6).

We will now define a graph for the purpose of showing the connection between the problem of finding a projection $\hat{f}$ and the problem of finding a shortest path in a directed graph. Let $G = (V, A)$ be a directed graph such that the set of vertices V is given by

$$V = \{t_0, t_1, \dots, t_n, t_{n+1}\} \tag{8}$$

and the set of directed arcs is given by[7]

$$A = \{(t_k, t_l) \in V^2 : t_l - t_k \geq \gamma\} \backslash \{(t_0, t_2), (t_{n-1}, t_{n+1})\}. \tag{9}$$

There is a correspondence between each path from t_0 to t_{n+1} and a sequence of jumps in the interval $(t_1 - \gamma, t_n + \gamma)$. A path $(t_0, t_{l_1}, \dots, t_{l_m}, t_{n+1})$ can be associated with a function g with jumps at t_{l_1}, ..., t_{l_m}, such that $g(t_{l_k})$ is the most common value of f in interval $[t_{l_k}, t_{l_{k+1}})$. The definition (9) of the set of directed arcs ensures that all paths in the graph G correspond to at least one function in $\mathcal{G}_\gamma$.

We now introduce a weight function $W : A \to \mathbb{R}_+$ ensuring that the cost of the path coincides with the error $E(f, \cdot)$ of the corresponding function in the interval $(t_1 - \gamma, t_n + \gamma)$. Let $I_k = t_{k+1} - t_k$ for $k = 0, \dots, n$. It is noteworthy that $I_0, I_n = \infty$, while $I_k < 2\gamma$ for $k = 1, \dots, n-1$. We introduce the penalty for a jump $\phi_k = \gamma$ for $k = 1, \dots, n$ and $\phi_{n+1} = 0$. Now we define the weight function W:

$$W((t_k, t_l)) = \sum_{m=k}^{l-1} I_m d(s_{kl}, s_m) + \phi_l, \tag{10}$$

for $(t_k, t_l) \in A$, where s_{kl} represents the most common state in the interval $[t_k, t_l)$ of the original function f. The first term equals the $\text{dist}(f, g)$ in $[t_k, t_l]$. The second term adds a penalty for jump at t_l if t_l is finite (the penalty for jump at t_k was added on a previous arc in the path, if $k > 0$).

Theorem 2.1 (Problem equivalence) *Let $\gamma > 0$ and $(t_1, \dots, t_n)$ be the only discontinuities of a function $f \in \mathcal{T}$. Let $G = (V, A, W)$ be a weighted, directed graph as defined*

[7] In case of $n = 2$, both arcs have to be included in set A.

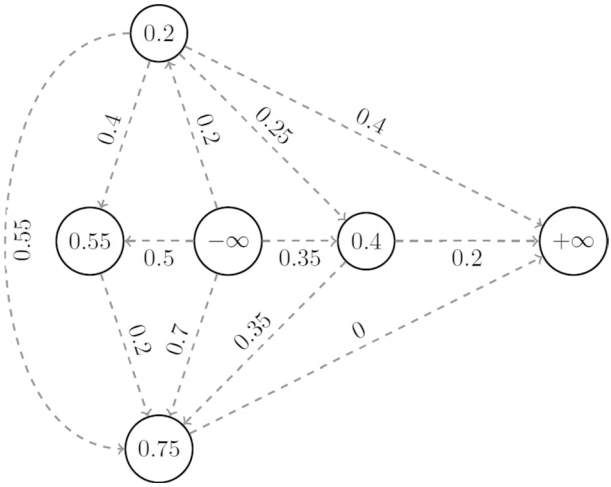

Fig. 1 Graph G constructed for the function f

in (8), (9), (10) *above. The task of finding a projection of f onto* $\mathcal{G}_\gamma$*, as defined in* (4)*, is equivalent to finding the shortest path from* t_0 *to* t_{n+1} *in the graph G.*

The proof of the theorem can be found in the appendix. Now, we will illustrate the method by an example.

Given $\gamma = 0.2$ and $\mathcal{S} = \{0, 1, 2, 3\}$, consider the function $f = \mathbb{1}_{[0.2,0.35)} + 2 \cdot \mathbb{1}_{[0.4,0.55)} + 3 \cdot \mathbb{1}_{[0.55,0.75)} + 2 \cdot \mathbb{1}_{[0.75,+\infty)}$. The graph G, as defined in (8), (9), (10), for f, is shown in Fig. 1. Note that the vertex corresponding to 0.35 is omitted in the graph, since there is no path from the vertex corresponding to $-\infty$ to it (according to the definition (9), the arc (0.2, 0.35) is not included).

There are nine possible paths from $-\infty$ to $+\infty$. The path $\hat{P} = (-\infty, 0.4, \infty)$ has the cost equal to 0.55 and is the shortest path from $-\infty$ to $+\infty$. Hence we conclude that $\hat{f} = 2 \cdot \mathbb{1}_{[0.4,\infty)}$ is the projection of f onto $\mathcal{G}_{0.2}$ (in this case, it can be shown $\hat{f}$ is the only projection of f).

2.3 Binary case

In case the set of states $\mathcal{S}$ consists of only two elements, a stronger result than Lemma 2.2 can be achieved. The main advantage of the binary case comes from the fact that we do not need to specify the sequence of states since knowing the starting state, each jump signifies a move to the only other available state. First, we present a supporting remark which further strengthens the relation between jumps of a function from $\mathcal{T}$ and its projection.

For the remainder of the section, we will always assume that $\mathcal{S} = \{0, 1\}$.[8]

Lemma 2.4 *Let $\gamma > 0$ and $f \in \mathcal{T}$. Let J denote the set of all discontinuities of the function f. If a function $g \in \mathcal{G}_\gamma$ contains a jump $j \in J(f)$, but in an opposite direction than in f, then g cannot be a projection of f onto $\mathcal{G}_\gamma$.*

Lemma 2.5 *Let $\gamma > 0$ and $f \in \mathcal{T}$. Any solution $\hat{f}$ of the problem (3) is an element of $\mathcal{G}_{2\gamma}$.*

The Proofs of Lemma 2.4 and lemma 2.5 can be found in the appendix. Lemma 2.5 leads to the equivalence of the problem (4) with the minimization problem:

$$\hat{f} \in \underset{g \in \mathcal{G}_{2\gamma}}{\arg\min}\, E_\gamma(f, g). \tag{11}$$

The strengthening of Lemma 2.1 by restricting not only the locations of the jumps but also their directions is a favorable change as it narrows the set of possible solutions.

Lemma 2.6 *Let $f \in \mathcal{T}$. Suppose $f \equiv c$ on an interval $[a, b]$ for some constant $c \in \mathbb{R}$. If $b - a > \gamma$, then $\hat{f} \equiv c$ on $[a, b]$. If $b - a = \gamma$, then there exists a projection such that $\hat{f} \equiv c$ on $[a, b]$.*

Proofs of Lemma 2.6 can be found in the appendix.

Lemma 2.6 potentially reduces the number of jumps that have to be considered in the post-processing. Moreover, lemma 2.4 reduces the number of arcs when building the graph making the process of finding the shortest path more effective.

Additionally, Remark 2.2 can also be strengthened.

Remark 2.3 If $n > 2$ and all states are shorter than γ (except for the first and the last state), there exists a projection such that the second and the second-to-last jump of the original function are not present in it.

Remark 2.3 allows us to ignore the second and the penultimate jump of the projected function when searching for jump locations in the projection. The proof of this remark can be found in the appendix.

The directed graph G has a different set of vertices compared to (8):[9]

$$V = \{t_0, t_1, \ldots, t_n, t_{n+1}\} \backslash \{t_2, t_{n-1}\}, \tag{12}$$

and of directed arcs compared to (9):

[8] This convention deviates from the notation established in Sect. 2.1 as it is more natural to use 0 and 1 as states in the binary case.

[9] If $n = 2$, then both of those jumps are present in V.

$$A = \{(t_k, t_l) \in V^2 : t_l - t_k \geq 2\gamma \text{ and } l - k \bmod 2 \equiv 1\}. \tag{13}$$

Theorem 2.2 (Problem equivalence-binary version) *Let* $\gamma > 0$ *and* $(t_1, \ldots, t_n)$ *be the only discontinuities of a function* $f \in \mathcal{T}$. *Let* $G = (V, A, W)$ *be a weighted, directed graph as defined in* (10), (12), (13). *The task of finding a projection of* f *onto* $\mathcal{G}_{2\gamma}$, *as defined in* (11), *is equivalent to finding a shortest path from* t_0 *to* t_{n+1} *in the graph* G.

Proof of the theorem 2.2 can be found in the appendix.

3 Incorporating domain knowledge into the performance measure of classification

3.1 Problem-specific requirements on the performance measure

In order to choose an appropriate performance measure for a given classification task, it is important to understand the problem-specific demands on the result. The standard distance (1), which can be understood as a continuous analogue of the most common performance metric, namely the misclassification rate, can often be inadequate to compare the classification results as it is a one-fits-all type of metric and if more is known about the problem, it might not represent the idea of accuracy that users have in mind. On the other hand, there have been other approaches to performance metrics, e.g. (Ward et al. 2011). Their approach focuses on characterizing the error in terms of the number of inserted, deleted, merged and fragmented events. Event fragmentation occurs when an event in the true labels[10] is represented by more than one event in the estimated labels,[11] whereas merging refers to several events in true labels being represented by a single event in the estimated labels. Ward et al. (2011) provide an overview of different performance metrics used in activity recognition proposing a solution to the problem of timing uncertainty as well as event fragmentation and merging. Their solution is based on segments, which are intervals in which neither the true labels nor the estimated labels change the state. If the state in the estimate and the state in the true labels agree in a given segment, they denote it as correctly classified. If that is not true, the segment is classified accordingly as fragmenting segment, inserted segment, deleted segment or merged segment. This provides a deeper level of error characterization, which is then used in different metrics of classifier performance. Their vector-valued performance metric is preferable when in-depth overview of the types of mistakes made by the classifier is needed. We will introduce a novel scalar-valued performance metric, which can be easily compared and includes problem-specific information such as timing uncertainty in the labels.

[10] If a state sequence corresponds to the true underlying sequence of activities in a time series, then it will be called the *true labels*.

[11] An estimate of the true labels will be called the *estimated labels*.

Table 1 The results of the labelling experiment; all times are in seconds

Partic.	Running		Jumping		Kick	
	Start	End	Start	End	Start	End
P1	2	7.3	2.7	5.2	2.5	3.5
P2	2	7.5	2.7	5.2	2.5	3.9
P3	2.3	6.6	2.7	5.1	2.7	3.6
P4	2.3	7.2	2.7	5.3	2.5	4.3
P5	2.2	7.2	2.9	5.4	2.5	4.1
Avg.	2.16	7.16	2.74	5.24	2.54	3.88
Std	0.15	0.34	0.09	0.11	0.09	0.33

The two last rows show the average and the sample standard deviation for each boundary

In this section, we aim at highlighting the main characteristics of the classification of movements based on wearable sensors and at translating them into specific requirements on the performance measure. Our first requirement comes from physical restrictions. The states considered in our application represent human activities, but also in more general contexts they often cannot be arbitrarily short; there is a lower bound on the length of the events in a state sequence. Hence, estimated labels that violate this lower bound indicate a bad performance. The lower bound condition requires two parameters: the lower bound and the penalty for each violation. The lower bound can either be estimated or determined from domain knowledge, while the penalty can be chosen more freely. Through physical restrictions we can see a deeper connection with the method introduced in Sect. 2. It is clear that the standard classification methods cannot ensure that the state sequence contains only events longer than a certain level. The post-processing method addresses this issue directly and as a consequence we can expect classifiers to benefit from it in the context of the new performance measure.

The issue of timing uncertainty should also be addressed when designing the performance measure. To illustrate its importance more clearly, we present an example. Five people were asked to detect boundaries between activities in different time series using a visualization tool. The tool outputs an animated stick figure model[12] given sensor data.

Three time series were selected, each with one of the following activities: running, jumping and ball kick. The start and the end of each activity were recorded by participants. Table 1 presents the results of the experiment.

The experiment indicates there is indeed uncertainty regarding the state transitions. Granted that the sample size is very small, we notice more variation in results referring to the end of activities rather than the beginnings. Additionally, we see more variation in the results for the kick than the jumping. So the boundaries of some activities seem to be more difficult to identify than of others.

[12] A symbolic representation of the human body using only lines.

3.2 Globally time-shifted distance

The standard distance (1) is an unsatisfying measure to compare two state sequences, since it does not incorporate the requirements posed in the previous section. In order to improve it, we start by modelling the timing uncertainty. Let $f \in \mathcal{T}$ be the true labels process and let f have n discontinuities $t_1, \ldots, t_n$. The locations of the discontinuities are corrupted by additive noise:

$$t_i = T_i + X_i,$$

for all $i = 1, \ldots, n$, where T_i is the true and unknown location of the i-th jump. In this section we will assume that $X_1 = X_2 = \cdots = X_n$ (all jumps are moved by the same value; the global time shift), although in general, it is more realistic to assume that $X_1, \ldots, X_n$ are independent random variables. We will relax this condition later.

We define a class of Globally Time-Shifted distances (GTS distances), loosely inspired by the Skorokhod distance on the space of càdlàg functions (Billingsley 1999, pp. 121). The GTS distances are parametrized by two parameters. The parameter w controls the weight of misclassification occurring from the uncertainty of the true labels, while the parameter σ controls the magnitude of the shift of activities.

Definition 3.1 (*Globally Time-Shifted distance*) Let $f, g \in \mathcal{T}$. Given $w \geq 0, \sigma > 0$ and a metric d on $\mathcal{S}$ we define a Globally Time-Shifted distance as:

$$GTS_{w,\sigma}(f, g) = \inf_{\epsilon \in [-\sigma, \sigma]} \{\text{dist}(f \circ \tau_\epsilon, g) + w|\epsilon|\},$$

where for $\epsilon > 0$ $\tau_\epsilon : \mathbb{R} \to \mathbb{R}$ is a time shift defined as follows:

$$\tau_\epsilon(t) = t - \epsilon.$$

Depending on the choice of parameters the GTS distance possesses certain properties. For $w > 0$ and $\sigma = \infty$, the GTS distance is an extended metric[13] and a proof of this fact is given in the appendix. If $w > 0$ and $\sigma > 0$, then it is a semimetric meaning that it has all properties required for a metric, except for the triangle inequality.

The main downside of using the GTS distance is the unrealistic assumption on timing uncertainty. However, if we know that the true labels preserve the true state durations then it is a good choice. Consider a function $f \in \mathcal{T}$ with two state transitions located at t_1 and t_2. Let $g \in \mathcal{T}$ also feature two state transitions located at $t_1 - \tau_1$ and $t_2 - \tau_2$. If $\tau_1 \neq \tau_2$, then there is no global time shift that can align the functions f and g. This implies that the true state durations need to be preserved in the estimate in order to align functions using the global time shift.

[13] It may attain the value ∞.

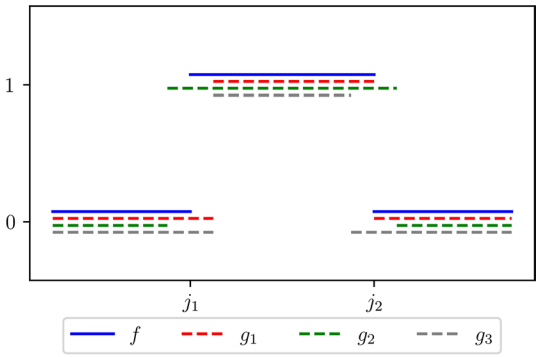

Fig. 2 The function f represents the true labels with an uncertainty around state boundaries, g_i are the approximations of f

3.3 Locally time-shifted distance and the duration penalty term

The global time shift stresses the state durations, which is not always desirable. For instance, if the true labels do not preserve the real state durations, or e.g. if the additive noise terms in the locations of the jumps are independent. Here is an example: Fig. 2 shows f and its approximations g_i for $i = 1, 2, 3$. It is impossible to align f with any of the g_i with a single time shift, however, it would be possible if each state transition could be shifted 'locally'.

Naturally, to accommodate for this issue, a suitable modification would be to replace one global time shift with multiple local time shifts. We now introduce a measure of closeness between state sequences which conceptually can be seen as derived from the GTS measure. We will be working with sequences of jumps, but more specifically given two sequences of state boundaries we will combine them together and sort the resulting joint sequence in an increasing order. Subsequent pairs of values in this sequence are determining segments understood as in Ward et al. (2011). We weigh different types of segments and the result is a weighted average of segment lengths, which is supposed to reflect well the error magnitude of the classifier.

We define segments formally and introduce a new distance on $\mathcal{T}$.

Definition 3.2 (*Segments*) Let $f, g \in \mathcal{T}$. The elements of the smallest partition[14] of $\mathbb{R}$ such that in each element of the partition neither f nor g changes state will be called segments.

Since functions from $\mathcal{T}$ are piece-wise constant and have a finite number of discontinuities, there is always a finite number of segments. The general form of segments that we will use is as follows:

[14] A partition that cannot be made coarser.

$$(-\infty, a_1) \cup \bigcup_{i=1}^{l-1} [a_i, a_{i+1}) \cup [a_l, \infty), \tag{14}$$

where $a_1 < a_2 \cdots < a_l$ if f and g are not both constant on the real line. Otherwise there is only one segment, consisting of the whole real line. By convention, $a_0 = -\infty$ and $a_{l+1} = \infty$, and

$$f(a_0) = f(a_1^-) = \lim_{x \to -\infty} f(x),\ f(a_{l+1}) = f(a_l).$$

Definition 3.3 (*Locally Time-Shifted distance*) Let $w \geq 0$, $\sigma > 0$ and d be a metric on $\mathcal{S}$. Let $f, g \in \mathcal{T}$ and their set of segments to be denoted as in (14). We define the Locally Time-Shifted distance (LTS distance) as

$$LTS_{w,\sigma}(f, g) = \sum_{i=1}^{l-1} \delta_i (a_{i+1} - a_i) d(f(a_i), g(a_i)),$$

where

$$\delta_i = \begin{cases} w, & a_{i+1} - a_i \leq \sigma,\ f(a_{i-1}) = g(a_{i-1}), f(a_{i+1}) = g(a_{i+1}) \\ 1, & \text{otherwise.} \end{cases}$$

Similarly to the GTS distance, the parameter w controls the weight of misclassification occurring from the uncertainty of the true labels. The case when $w < 1$ is more interesting to us, since it corresponds to timing uncertainty of the labels. If $w \geq 1$, then we put more importance on the timings of the jumps (opposite to timing uncertainty). The LTS distance is an extended semimetric for $w > 0$ (for a proof, see the appendix). The triangle inequality does not hold in general.

The LTS distance addresses the issue of timing uncertainty in the true labels. Let $\zeta > 0$[15] be the lower bound on the lengths of the events as determined by the domain knowledge (or through estimation if possible). Let $\lambda > 0$ be the penalty for each violation of the lower bound condition. For $f \in \mathcal{T}$ with its discontinuities $t_1, \ldots, t_n$, we introduce a *duration penalty term*:

$$DP_{\lambda,\zeta}(f) = \lambda \sum_{k=1}^{n-1} \mathbb{1}_{[0,\zeta)}(t_{k+1} - t_k).$$

This term will allow to lower the performance of classifications with unrealistically short events.

In practice, we will need to extend the functions to the real line in order to use the LTS distance as it is defined for functions with domain equal to the whole of $\mathbb{R}$. One natural extension could be to extend the first and the last state of each function indefinitely. However, this solution leads to a problem. Let $M > 0$. Consider

[15] Note that ζ is related in its interpretation to the γ parameter introduced in Sect. 2.

two functions $f : [0, M] \to \mathcal{S}$ and $g : [0, M] \to \mathcal{S}$ such that for some $0 < a < M$, $f(t) \neq g(t)$ on $[0, a)$. No matter how small a is, the distance between extended f and g will always be infinite when using this extension, since in this case extended f and g are in different states on the whole half line $(-\infty, a)$. Both functions need to be extended by the same state for the distance to be finite. We extend any function f defined on interval $[0, M]$ to the real line, setting its value to an arbitrary state outside of $[0, M)$. The distance is independent of the chosen state, as on the infinite segments that it introduces f and g are both equal. Without loss of generality, we choose state 1.

$$f^*(t) = \begin{cases} f(t), & t \in [0, M) \\ 1, & t \notin [0, M). \end{cases} \tag{15}$$

Notice that this extension does not have the problem stated above as f^* and g^* are equal on the segments that it introduces and does not change the value on the original segments regardless of the choice of the state outside of $[0, M]$.

We combine the LTS distance and the duration penalty term to define the LTS measure of closeness of two state sequences.

Definition 3.4 Let f be a function of true labels and g its estimate, both defined on $[0, M]$. The *LTS measure* is defined as:

$$LTS_{w,\sigma,\lambda,\zeta}(f, g) = \exp(-LTS_{w,\sigma}(f^*, g^*)/M - DP_{\lambda,\zeta}(g)).$$

The scaling through the division by M normalizes the LTS distance to the interval $[0, 1]$. The transformation $[0, +\infty) \ni x \to \exp(-x) \in (0, 1]$ maps the sum of the LTS distance and the duration penalty term to the interval $(0, 1]$, while reversing the order as well: g is closer to f if the LTS measure is closer to 1.

4 Application to activity recognition

4.1 Simulation study

We consider a dataset created using a random procedure, which mimics the behavior of activity recognition classifiers with varying accuracy (depending on the parameters). Let $\mathcal{S} = \{1, 2, 3\}$. Consider a function f representing a 60 second long state sequence:

$$f = \mathbb{1}_{[0,5)} + 2 \cdot \mathbb{1}_{[5,15)} + 3 \cdot \mathbb{1}_{[15,30)} + 2 \cdot \mathbb{1}_{[30,40)} + 3 \cdot \mathbb{1}_{[40,55)} + \mathbb{1}_{[55,60]}.$$

f will be referred to as the correct labels. We introduce noise into f in the following manner:

- two sequences of i.i.d. random variables are considered $\{Y_k\}$ and $\{Z_k\}$, with $Y_k \sim Exp(\mu_1)$ and $Z_k \sim Exp(\mu_2)$ for some parameters $\mu_1, \mu_2 > 0$,

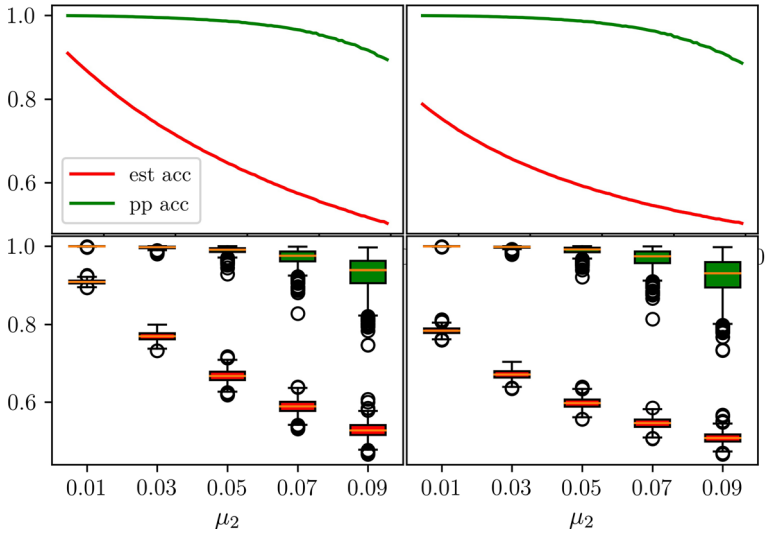

Fig. 3 The top left plot shows the mean accuracy of the noisy labels (red) and the mean accuracy of the post-processed labels (green) as calculated for different values of μ_2. The top right plot shows the LTS measure of the noisy labels (red) and the post-processed labels (green) as calculated for different values of μ_2. All lines drawn for 100 different values of μ_2. The bottom left boxplot shows the variability of the accuracy amongst the estimates (red) and the post-processed estimates (green). The bottom right boxplot shows the variability of the LTS measure amongst the estimates (red) and the post-processed estimates (green). Boxplots have been constructed for 5 different values of μ_2

- $\{Y_k\}$ represents the time spent in the correct state, while $\{Z_k\}$ represents the time spent in the incorrect state,
- we use the sequence $Y_1, Z_1, Y_2, Z_2, \ldots$ to generate noisy labels, where the sequence ends when the sum of all drawn numbers is exceeding 60 seconds,
- for each variable Z_i an incorrect state is chosen randomly out of the remaining two and f is changed to that state on interval $[\sum_{k=1}^{i-1}(Y_k + Z_k) + Y_i, \sum_{k=1}^{i}(Y_k + Z_k))$,
- μ_1 and μ_2 control the duration of the states.

As our performance measure we choose the LTS measure with parameters: $w = 0.6$, $\sigma = 0.35$, $\lambda = 0.0001$, $\zeta = 0.5$, $d = \rho$. The post-processing is performed for the noisy labels with parameter $\gamma = 0.5s$. To demonstrate the utility of the post-processing procedure, we draw the noisy function 1000 times for a given set of parameters (μ_1, μ_2) and compare the accuracy of the noisy labels, the accuracy of the post-processed labels, the LTS measure of the noisy labels and the LTS measure of the post-processed labels.

In the first setting, we fix $\mu_1 = 0.1s$. The procedure is repeated for $\mu_2 \in [0.01, 0.1]$ (100 sample points from the interval are chosen). Figure 3 compares the mean accuracy of the noisy labels and the post-processed labels as well as the mean LTS measure of the noisy labels and the post-processed labels.

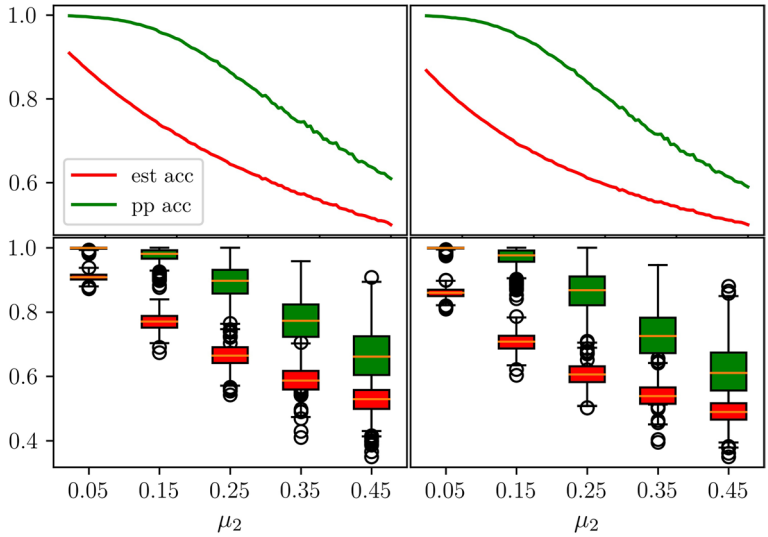

Fig. 4 The top left plot shows the mean accuracy of the noisy labels (red) and the mean accuracy of the post-processed labels (green) as calculated for different values of μ_2. The top right plot shows the LTS measure of the noisy labels (red) and the post-processed labels (green) as calculated for different values of μ_2. All lines drawn for 100 different values of μ_2. The bottom left boxplot shows the variability of the accuracy amongst the estimates (red) and the post-processed estimates (green). The bottom right boxplot shows the variability of the LTS measure amongst the estimates (red) and the post-processed estimates (green). Boxplots have been constructed for 5 different values of μ_2

In the second setting, we fix $\mu_1 = 0.5s$. The procedure is repeated for $\mu_2 \in [0.05, 0.5]$ (100 sample points from the interval are chosen). Figure 4 shows the mean accuracy of the noisy labels and the post-processed labels as well as the mean LTS measure of both the noisy labels and the post-processed labels.

In the third setting, we fix $\mu_1 = 1s$. The procedure is repeated for $\mu_2 \in [0.1, 1]$ (100 sample points from the interval are chosen). Figure 5 shows the mean accuracy of the noisy labels and the post-processed labels as well as the mean LTS measure of both the noisy labels and the post-processed labels.

All three experiments show the improvement in the accuracy as well as the LTS measure thanks to the use of post-processing. Additionally, we conclude that the post-processing method behaves better when dealing with multiple shorter intervals rather than fewer longer intervals. Moreover, the boxplots show more variability in the performance of the post-processed estimates when dealing with initial estimates with fewer but longer intervals of misclassification. This can be due to the fact that at the level of around 0.5 in accuracy and in the LTS measure, the post-processing starts behaving much worse and is not able to recover the original signal as reliably. It shows the limits of the method and the fact that there is a point at which the method starts to behave worse.

We also investigate the importance of the parameter γ on the results. We fix $\mu_1 = 0.1$, $\mu_2 = 0.08$. The procedure is repeated for $\gamma \in [0.01, 2.5]$ (100 sample

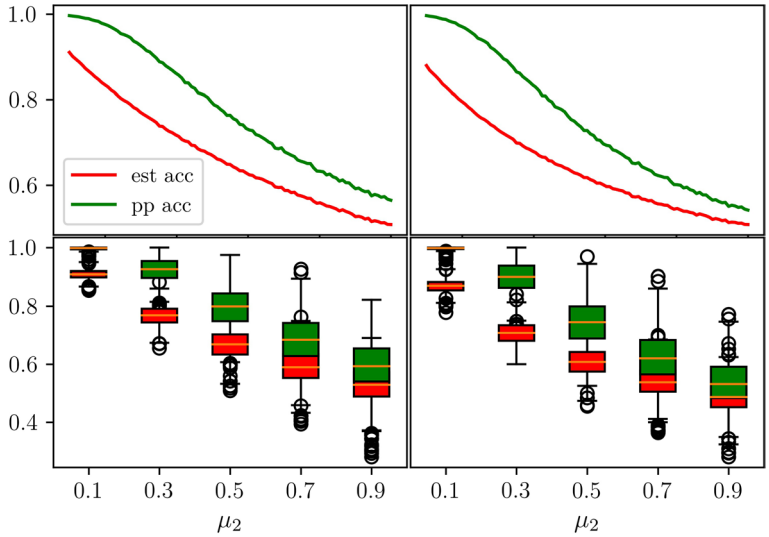

Fig. 5 The top left plot shows the mean accuracy of the noisy labels (red) and the mean accuracy of the post-processed labels (green) as calculated for different values of μ_2. The top right plot shows the LTS measure of the noisy labels (red) and the post-processed labels (green) as calculated for different values of μ_2. All lines drawn for 100 different values of μ_2. The bottom left boxplot shows the variability of the accuracy amongst the estimates (red) and the post-processed estimates (green). The bottom right boxplot shows the variability of the LTS measure amongst the estimates (red) and the post-processed estimates (green). Boxplots have been constructed for 5 different values of μ_2

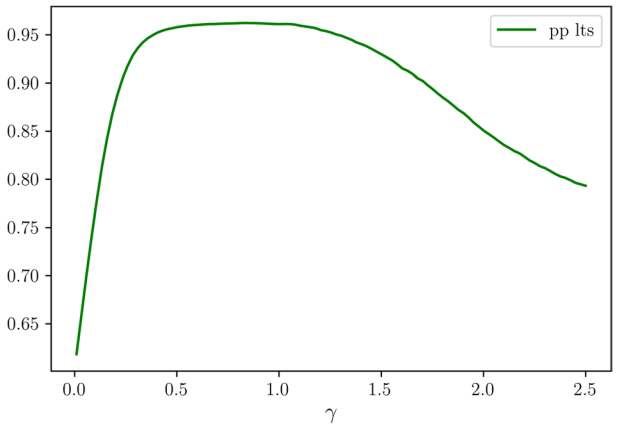

Fig. 6 The line shows the LTS measure of the post-processed labels drawn for 100 different values of γ. The mean accuracy of noisy labels was equal to 0.556 and the mean LTS measure of noisy labels was equal to 0.602

points from the interval are chosen). Figure 6 shows the mean LTS measure of the post-processed labels.

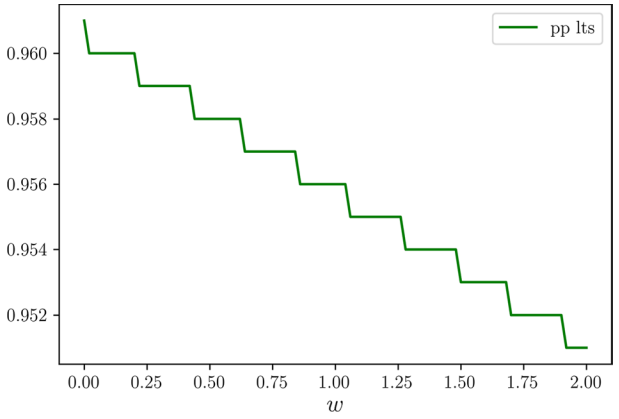

Fig. 7 The line shows the LTS measure of the post-processed labels drawn for 100 different values of w. The mean accuracy of noisy labels was equal to 0.555

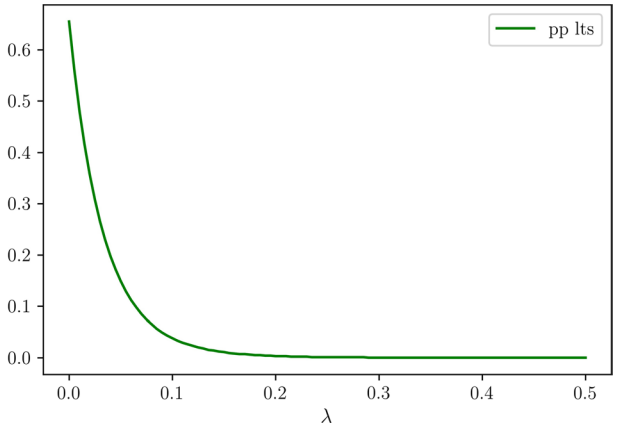

Fig. 8 The line shows the LTS measure of the post-processed labels drawn for different values of λ. The mean accuracy of noisy labels was equal to 0.56

We conclude that the parameter γ can influence the LTS measure of the post-processed functions $\hat{g}_i$. It needs to be chosen carefully since too low values will lead to accepting unrealistically short events while too high values will eliminate true events. In our case the values of γ between 0.5 and 1 are the most favourable. In practice the minimal length of the events in the true labels can inform on the choice of γ.

We finish the simulation study with a look at the parameters of the LTS measure. We will investigate the weight w first. Let all the other parameters of the LTS measure be set to $\sigma = 0.35$, $\lambda = 0.0001$, $\zeta = 0.5$. We fix $\mu_1 = 0.1, \mu_2 = 0.08, \gamma = 0.5$. The procedure is repeated for 100 different values of w in the interval [0, 2]. Figure 7 shows the mean LTS measure of the post-processed labels.

Figure 7 shows the effect of the parameter w on the LTS measure. As we can see on the y-axis, the values of the LTS measure are quite close together. Hence, we conclude that the choice of w is less important as its effect on the LTS measure is minimal. The main reason for this behaviour stems from the fact that σ restricts many of the erroneous intervals and the remaining ones for which w takes effect are quite small. Hence, the effect of w on the LTS measure is not large.

The parameter σ determines the length of the misclassified events up to which they are caused by timing uncertainty of the labels. σ can be chosen based on the domain knowledge, based on the experiment described in Sect. 3.1. The parameter ζ is a lower bound on the lengths of the events, hence can be determined by the domain knowledge. Given their clear interpretation, the parameters σ and ζ will not be subjected to the same procedure as the parameter w. Hence, the only parameter left to investigate is λ. As before, we fix $\mu_1 = 1, \mu_2 = 0.8, \gamma = 0.5$. We choose $w = 0.6$. The procedure is repeated for 100 values of λ between 0 and 0.5. Figure 8 shows the mean LTS measure of the post-processed labels. We can see that high values of λ can influence the LTS measure significantly, hence choices lower than 0.01 are preferable. We want to avoid that the penalty term is overshadowing the LTS distance.

4.2 Application to a football dataset

We will now demonstrate the benefits of the post-processing by projection in a real-life setting, utilizing the LTS measure to compare different methods of classification. Wilmes et al. (2020) give an extensive description of the football dataset of which we give a short summary below.

Eleven amateur football players participated in a coordinated experiment at a training facility of the Royal Dutch Football Association of The Netherlands. Five Inertial Measurement Units (IMUs) were attached to 5 different body parts: left shank (LS), right shank (RS), left thigh (LT), right thigh (RT) and pelvis (P). Each IMU sensor contains a 3-axis accelerometer (Acc) and a 3-axis gyroscope (Gyro). Athletes were asked to perform exercises on command, e.g. ‘jog for 10 meters’ or ‘long pass’. For each athlete and exercise this resulted in a 30-dimensional time series (5 body parts times 6 features per IMU) of length varying from 4 to 14 seconds. Each athlete performed 70–100 exercises which amounts to nearly 900 time series (each with a sampling frequency of 500 Hz). Time series are labelled with the command given to an athlete, but there are still other activities performed in each of the time series, for example standing still. This causes a problem; ignoring standing periods and treating them as part of the main signal pollutes the data and lowers the quality of the classification. To show the advantages of post-processing by projection, we select only two states: ‘standing’ and ‘other activity’ encoded as 0 and 1, respectively. 15 time series (representative of all possible actions performed by athletes) were manually labelled time point by time point in order to be able to train classifiers, and these will form our sample. All 15 time series were chosen from the single athlete.

In pre-processing we are using the sliding window technique on the sensors (Dietterich 2002). This method transforms the original raw data using windows of fixed length d and a statistic of choice T: given a time point t, its neighbourhood of size d is fed to the statistic T for each variable separately. Performing the procedure for each time point results in a time series of the same dimension as the original one, but every observation is equipped with some knowledge about the past and the future through the statistic T and through forming the neighbourhoods of size d. Regarding the choice of the statistic T one needs to be careful, since the sensors are highly correlated with each other. The information about standing contained in one variable is comparable to the one in another, namely the variance of the signal is low when the person is standing (differences can occur when considering different legs; a low variance on one leg might be misleading since the other leg might already be transitioning into another position).

Leave-5-out cross-validation will be performed in order to select the best performing classification method out of the 7 standard machine learning methods, which will be listed below. A typical approach to k-fold cross-validation with a training sample of size $k-1$ cannot be applied here, since a single time series is not a representative sample of different types of events. 15 time series will be used. In each iteration 10 time series will be randomly chosen for training and 5 for testing. The results are going to be shown for post-processed classifiers, unless specified otherwise. Before cross-validation can be performed, we need to fix the parameters of the performance measure we introduced in Sect. 2. The parameters of the LTS measure are chosen as follows:

- We have limited information regarding how uncertain locations of state transitions are, but based on the small experiment described in Sect. 3.1 we select $\sigma = 0.35$ (the largest deviation between different true labels).
- The parameter w is chosen as 0.6, but as shown in Sect. 4.1 its choice is not that important.
- The lower bound γ on the duration of activities is selected as the length of the shortest activity in the learning dataset, which is equal to 0.8s in our case.
- A penalty λ represents the cost of additional or missing jumps in a state sequence compared to the true labels. We decide for the penalty $\lambda = 0.01$ in order not to overshadow the LTS distance with too much importance placed on the penalty term (more details on that were given in Sect. 4.1, specifically in Fig. 8).

Before assessing classifiers on the training set, one needs to consider an appropriate feature set. Our variables are highly dependent on one another, so we start with feature selection. We perform feature ranking using the Relieff algorithm and select the 6 most relevant features based on the Relieff weights (more details on the method in Kononenko et al. 1997). Then we test all possible combinations of these features, which is now computationally feasible, in order to find the best set for each of the classifiers. The features selected by the Relieff algorithm are RTGyroX, RTGyroY, RTAccX, RTAccZ, LTAccY, PAccY, where the naming convention is as follows: RTGyroX refers to the x-axis of the gyroscope located on the right thigh and so forth.

Table 2 Average of the leave-5-out cross-validation scores for all classifiers using the best sensor set for each of them

Classifier	OG test	PP test
MLP	0.916+/-0.031	0.972+/-0.008
LR	0.898+/-0.034	0.968+/-0.015
kNN	0.59+/-0.05	0.967+/-0.020
RF	0.83+/-0.07	0.966+/-0.017
SVC	0.894+/-0.034	0.966+/-0.017
DT	0.83+/-0.07	0.965+/-0.008
NB	0.88+/-0.04	0.944+/-0.023

The pre-processing consisted of the sliding window technique in combination with summarizing by the standard deviation. The OG Test averages the LTS measure on the test set for the original classifier, while the PP Test is the same value for the post-processed classifier

Proceeding with the cross-validation we select the following classifiers (with their abbreviations) to be assessed: DT—Decision Tree, kNN—k-Nearest Neighbors, LR—Logistic Regression, MLP—Multi-layer Perceptron, NB—Naive Bayes, RF—Random Forest, SVM—Support Vector Machine. The results of leave-5-out cross-validation are shown in table 2. It is striking that the test scores of the post-processed classifiers are at most 0.028 apart. This is due to post-processing by projection. The correction it provides brings all classifiers closer together. This result can be extended even further. The test score of a decision tree ranges from 59–86% for different sensor sets before post-processing, while using the post-processing results in a range of test scores from 93–96.5% and this is not specific to decision trees only.

The example shows that the post-processing is crucial. Firstly, it increases the accuracy of a given estimator on a given feature set by 35%. Secondly, it diminishes the impact of feature selection as the difference in accuracy between different feature subsets decreases substantially. Feature selection is of course still important as it decreases the computational complexity of the problem and allows to get rid of redundancy in the feature set. However, with methods that only rank features such as Relieff the choice of the threshold we choose to classify a feature as significant or not is less important. Finally and most importantly, the post-processing by projection allows to select a method according to criteria other than the performance, namely the computational speed.

5 Conclusion

In this paper we have introduced a post-processing scheme that allows to improve estimates. It finds estimated activities that are too short and eliminates them in an optimal way by finding the shortest path in a directed acyclic graph.

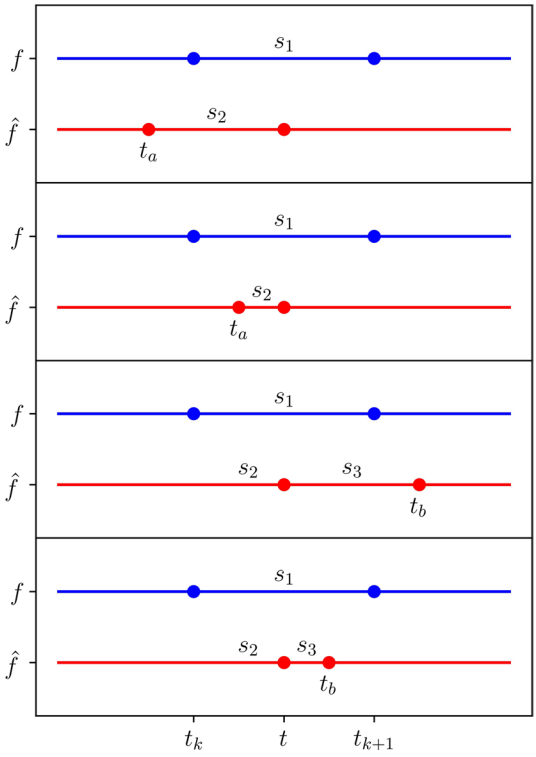

Fig. 9 Illustration supporting the Proof of Lemma 2.1. Plots correspond (from top to bottom) to cases 1a, 1b, 2a, 2b respectively

A simulation study is conducted to assess the benefits brought by the post-processing method. Generated noisy labels are improved with the use of the post-processing. The positive effects on the LTS measure are more significant when the noisy sequence contains more short intervals of misclassification.

Real-life football sensor data were used to assess the adequacy of the post-processing scheme in the more realistic setting. It significantly improved the performance of the classifiers. At the same time, the post-processed classifiers are closer to each other in performance than the original ones. This allows placing more importance on other criteria, such as the computational speed of the method. It should be noted that post-processing cannot correct for uncertainty in the classification result of the estimators. It can be seen in Figs. 3, 4 and 5 that the worse the original estimate the worse the post-processed one (at least as a rule of thumb as there can be cases when it is reversed). However, most importantly, the results of the application to the football dataset are promising. The post-processing by projection was able to improve the estimators of accuracy ranging from 59% to 86% up to a score of 93% to 96.5%. We note that the lowest score of the post-processed estimates for any given classification method is still higher than the highest score of the original estimates. An alternative to our method could be to integrate the penalization of too short windows into the classifier.

This is not an easy idea to realize, since classifiers usually do not consider the duration of activities themselves and they classify in a time linear manner. Nonetheless, if an appropriate scheme were to be defined, it would expand on the theory developed in this paper.

Another contribution are novel measures of classifier performance in the task of activity recognition using wearable sensors. They address the issue of timing offsets as well as unrealistic classifications, while retaining a typical scalar output of a performance measure allowing for easy comparisons between classifiers.

A Proofs

Proof of Lemma 2.1 Let $\hat{f}$ be a solution of problem (3) for a given function f. Assume that $\hat{f}$ contains a jump t outside of the set $J(f)$. Denote the jump or one of the jumps closest to t in the original function f by t_k. Without loss of generality we assume t_k is located left of t (t_k exists otherwise f is constant and $\hat{f} = f$). Let t_a and t_b denote the jump preceding and resp. following t in the projection $\hat{f}$. Let t_{k+1} denote the jump following t_k in the original function f. Let s_1 be the state in which the original function stays in the interval $[t_k, t_{k+1})$ and let s_2 be the state from which the projection $\hat{f}$ jumps at t and let s_3 be the state to which the projection $\hat{f}$ jumps at t.

We will consider multiple cases and in each of them we will present a modification to $\hat{f}$ that either shows that $\hat{f}$ cannot be a projection or that there exists a function which is not worse than $\hat{f}$ and does not contain a jump at t. The configurations of the cases are depicted in Fig. 9.

1. $s_1 \neq s_2$

   (a) $t_a < t_k$. Moving the jump t to t_k does not increase the error (and potentially lowers it, if $s_3 = s_1$).
   (b) $t_a \geq t_k$. We move the jump t_a to t, which results in lowering the error by at least γ. Then we go back to the beginning of the proof with redefined state s_2 and jump t_a.

2. $s_1 = s_2$

   (a) $t_b \geq t_{k+1}$. Moving the jump t to t_{k+1} lowers the error by at least $t_{k+1} - t$ since $s_3 \neq s_1$.
   (b) $t_b < t_{k+1}$. We move the jump t to t_b, which results in lowering the error by penalty term γ and $t_b - t$, since $s_1 = s_2$. We go back to the case $s_1 = s_2$ with t moved to t_b and t_b moved to the next jump in $\hat{f}$ (if it does not exist, then $t_b = \infty$). Eventually the jump t can be moved to t_{k+1} (when case 2(a) is reached).

Loops occurring in cases 1(b) and 2(b) are not problematic, since with each iteration the number of jumps of the solution is reduced, eventually cases 1(a) or 2(a) are reached. □

Proof of Lemma 2.2 Let $\hat{f}$ be a solution of the problem 3 for a given function f. Assume that for certain $\tilde{\gamma} < \gamma$, $\hat{f} \in \mathcal{G}_{\tilde{\gamma}}$ and $\hat{f} \notin \mathcal{G}_{\gamma}$. Hence there exist two jumps t_k and t_l of f and $\hat{f}$ (which follows from lemma 2.1), such that $\tilde{\gamma} < t_l - t_k < \gamma$. Since the state lasts less than γ, it can be removed (in the sense that one of the jumps is removed and either the previous state or following state is longer by $t_l - t_k$) with a gain in error of less than γ and decrease in error of exactly γ, which means we found a function with lower error than $\hat{f}$. This contradiction ends the proof. □

Proof of Lemma 2.3 Let f be a function with two neighboring jumps t_1, t_2 and the state s_1 between them. Assume $t_2 - t_1 \geq 2\gamma$. Since the interval is longer than or equal to 2γ it satisfies the condition of the class $\mathcal{G}_{\gamma}$. Let us assume that the projection $\hat{f}$ of f contains two neighbouring jumps t_a and t_b such that $t_a \leq t_1 < t_2 \leq t_b$ and the state in the interval $[t_a, t_b)$ is $s_2 \neq s_1$. We introduce notation $\alpha := t_1 - t_a$ and $\beta := t_b - t_2$. If $\alpha, \beta \geq \gamma$, then introducing the jumps at t_1 and t_2 with the state s_1 between them is possible, because the condition of the class $\mathcal{G}_{\gamma}$ is satisfied. Moreover, the error is decreased if $t_2 - t_1 > 2\gamma$ and is not increased if $t_2 - t_1 = 2\gamma$. If $\alpha \geq \gamma$ and $\beta < \gamma$, then introducing a jump at t_1 such that the state following it is s_1 is possible. Moreover, the error is decreased. Analogously when $\alpha < \gamma$ and $\beta \geq \gamma$. If $\alpha, \beta < \gamma$, then changing state s_2 to s_1 reduces the error.

In all cases, we have shown that there exists a projection that does not change the state longer than 2γ. □

Proof of remark 2.2 Let $\hat{f}$ be a projection of f onto $\mathcal{G}_{\gamma}$. Let t_1 and t_2 be the first two jumps in the original function f. Let s_1 and s_2 be the first two states in the original function f. If $\hat{f}$ had the first jump at t_2 from the state s_1, then a function g equal to $\hat{f}$ outside of interval $[t_1, t_2)$, but such that the jump from state s_1 is moved to the location of the jump t_1 has an error lower than or equal that of $\hat{f}$. If $\hat{f}$ had the first jump at t_2 from a state $s_i \neq s_1$, then the error is infinite (since the value of $\hat{f}$ differs from f on the interval $(-\infty, t_1)$) and $\hat{f}$ cannot be a projection.

The argument is analogous for the penultimate jump. □

Proof of theorem 2.1 We use Lemma 2.1 to prove that a projection of a function from $\mathcal{T}$ onto $\mathcal{G}_{\gamma}$ can only have jumps at the same positions as the jumps in the original function. This leads to the fact that finding the shortest path in the graph is equivalent to finding $\hat{f}$. □

Proof of Lemma 2.4 Let $\hat{f}$ be a projection of f onto $\mathcal{G}_{\gamma}$. Let t_k and t_{k+1} be two consecutive jumps of f. Assume that $\hat{f}$ contains a jump t_k, but in opposite direction than in f. From Lemma 2.1 we know that the next jump of $\hat{f}$ can occur at the earliest at t_{k+1}. This means that in the interval $[t_k, t_{k+1})$ the projection $\hat{f}$ is equal to $1 - f$. In this case, moving the jump at t_k to t_{k+1} (or in the case of $t_{k+1} \in J(\hat{f})$ removing both jumps)

reduces the error by $t_{k+1} - t_k$. Hence, we conclude, a jump from f can only be present in its projection if it is in the same direction as in f. □

Proof of Lemma 2.5 The proof of this lemma is analogous to the Proof of Lemma 2.2. The possibility of strengthening the previous result comes from the fact that we can remove two jumps at once, in effect reducing the error by 2γ. □

Proof of Lemma 2.6 The proof of this lemma is analogous to the Proof of Lemma 2.3 □

Proof of remark 2.3 Let $\hat{f}$ be a projection of f onto $\mathcal{G}_{2\gamma}$. Let t_1 and t_2 be the first two jumps in the original function f. Let 0 and 1 be the first two states in the original function f without loss of generality. By assumption $t_2 - t_1 < 2\gamma$ (note that without this assumption both jumps could be included in a projection). Since $[t_1, t_2)$ is not a valid activity (shorter than γ), if $\hat{f}$ has a jump at t_2, it does not have a jump at t_1. If $\hat{f}$ had a jump at t_2 from the state 0, then a function g equal to $\hat{f}$ outside of interval $[t_1, t_2)$, but such that the jump from state 0 is moved to the location of the jump t_1 has lower error than $\hat{f}$. If $\hat{f}$ had a jump at t_2 from the state 1, then the error is infinite (since the value of $\hat{f}$ differs from f on the interval $(-\infty, t_1)$) and $\hat{f}$ cannot be a projection.

The argument is analogous for the penultimate jump. □

Proof of theorem 2.2 We use Lemmas 2.1 and 2.4 to prove that a projection of a function from $\mathcal{T}$ onto $\mathcal{G}_\gamma$ can only have jumps at the same positions and in the same directions as the jumps in the original function. This leads to the fact that finding the shortest path in the graph is equivalent to finding $\hat{f}$. □

GTS distance with $w > 0$ ***and*** $\sigma = \infty$ ***is an extended metric*** We will show that:

$$GTS_w(f, g) = \inf_{\epsilon \in \mathbb{R}} \{\text{dist}(f \circ \tau_\epsilon, g) + w|\epsilon|\}$$

is an extended metric on $\mathcal{T}$.

0. Since for any ϵ, $\text{dist}(f \circ \tau_\epsilon, g) \geq 0$ and $w|\epsilon| \geq 0$ we conclude that the GTS_w is non-negative.
1. It is obvious to see that $GTS_w(f, f) = 0$ for any $f \in \mathcal{T}$. Now let us assume that for some $f, g \in \mathcal{T}$ we have $GTS_w(f, g) = 0$. This implies that

$$\exists_{(\epsilon_n)} \ \text{dist}(f \circ \tau_{\epsilon_n}, g) + w|\epsilon_n| \xrightarrow{n \to \infty} 0.$$

Since $\text{dist}(f \circ \tau_{\epsilon_n}, g) + w|\epsilon_n|$ is an upper bound of $\text{dist}(f \circ \tau_{\epsilon_n}, g)$ and $w|\epsilon_n|$, we have

$$|\epsilon_n| \xrightarrow{n \to \infty} 0,$$

$$\int_{\mathbb{R}} d(f \circ \tau_{\epsilon_n}(t), g(t)) d\lambda(t) \xrightarrow{n \to \infty} 0.$$

From Fatou's lemma we have

$$\int_{\mathbb{R}} \liminf_{n\to\infty} d(f(t-\epsilon_n), g(t))d\lambda(t) = 0,$$

where λ is the Lebesgue measure on $\mathbb{R}$. Because f and g are càdlàg, this implies that for almost all t we have $f(t-) = g(t)$ or $f(t) = g(t)$ and so we conclude that $f = g$.

2. Let $f, g \in \mathcal{T}$, we have

$$\begin{aligned} GTS_w(f,g) &= \inf_{\epsilon}\{\mathrm{dist}(f\circ\tau_\epsilon, g) + w|\epsilon|\} = \inf_{\epsilon}\{\mathrm{dist}(g\circ\tau_{-\epsilon}, f) + w|-\epsilon|\} \\ &= \inf_{-\epsilon}\{\mathrm{dist}(g\circ\tau_\epsilon, f) + w|\epsilon|\} = \inf_{\epsilon}\{\mathrm{dist}(g\circ\tau_\epsilon, f) + w|\epsilon|\} \\ &= GTS_w(g,f), \end{aligned}$$

hence we conclude that GTS_w is symmetric.

3. Letting $f, g, h \in \mathcal{T}$, we have

$$\begin{aligned} GTS_w(f,g) &= \inf_{\epsilon}\{\mathrm{dist}(f\circ\tau_\epsilon, g) + w|\epsilon|\} \\ &= \inf_{\epsilon_1,\epsilon_2}\{\mathrm{dist}(f\circ\tau_{\epsilon_1}\circ\tau_{\epsilon_2}, g) + w|\epsilon_1+\epsilon_2|\} \\ &\le \inf_{\epsilon_1,\epsilon_2}\{\mathrm{dist}(f\circ\tau_{\epsilon_1}\circ\tau_{\epsilon_2}, h\circ\tau_{\epsilon_2}) + \mathrm{dist}(h\circ\tau_{\epsilon_2}, g) + \\ &\quad + w|\epsilon_1| + w|\epsilon_2|\} \\ &= \inf_{\epsilon_1,\epsilon_2}\{\mathrm{dist}(f\circ\tau_{\epsilon_1}, h) + w|\epsilon_1| + \mathrm{dist}(h\circ\tau_{\epsilon_2}, g) + w|\epsilon_2|\} \\ &= \inf_{\epsilon_1}\{\mathrm{dist}(f\circ\tau_{\epsilon_1}, h) + w|\epsilon_1|\} + \inf_{\epsilon_2}\{\mathrm{dist}(h\circ\tau_{\epsilon_2}, g) + w|\epsilon_2|\} \\ &= GTS_w(f,h) + GTS_w(h,g), \end{aligned}$$

which shows that GTS_w satisfies the triangle inequality and that concludes the proof. □

The LTS distance with $w > 0$ ***is a semimetric*** Let $w > 0, \sigma > 0$ and a metric d on $\mathcal{S}$ be fixed. We observe that $LTS_{w,\sigma}$ is nonnegative. Symmetry of $LTS_{w,\sigma}$ follows directly from the definition. It only remains to show that $LTS_{w,\sigma}(f,g) = 0$ if and only if $f = g$ for $f, g \in \mathcal{T}$.

We have

$$LTS_{w,\sigma}(f,f) = 0,$$

because there is only one segment (as defined in 3.2). Assume now that $LTS_{w,\sigma}(f,g) = 0$ and $f \neq g$. In that case, there exists more than one segment.

$$LTS_{w,\sigma}(f,g) = \sum_{i=1}^{l-1} \delta_i (a_{i+1} - a_i) d(f(a_i), g(a_i)) = 0$$
$$\Rightarrow \forall_{i=1,2,3,\ldots,l-1} \quad f(a_i) = g(a_i),$$

which implies that $f = g$, which contradicts the assumption. We conclude that $LTS_{w,\sigma}(f,g) = 0$ iff $f = g$, which completes the proof. □

Acknowledgements We thank Erik Wilmes for providing football data of high quality and the stick-model animation tool. It was the basis for the analysis of our methods in section 4. We also thank Bart van Ginkel for the idea of how to generalize the performance measure from the binary to the multiclass case.

Author contributions All authors contributed to the study conception and design. Data analysis were performed by the first author. The first draft of the manuscript was written by the first author and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding This work is part of the research programme CAS with project number P16-28 project 2, which is (partly) financed by the Dutch Research Council (NWO).

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and material The dataset analysed during the current study are available under the link https://zenodo.org/record/3732988#.YMcXOqgzZEZ.

Code availability Custom code for post-processing and performance measures can be found at: https://github.com/mgciszewski/improving_state_estimation_2022.

References

Aviles-Cruz C, Rodriguez-Martinez E, Villegas-Cortez J, Ferreyra-Ramirez A (2019) Granger-causality: an efficient single user movement recognition using a smartphone accelerometer sensor. Pattern Recognit Lett 125:576–583. https://doi.org/10.1016/j.patrec.2019.06.029

Billingsley P (1999) Convergence of probability measures, 2nd edn. Wiley, Hoboken

Capela Nicole A, Lemaire Edward D, Natalie B (2015) Feature selection for wearable smartphone-based human activity recognition with able bodied, elderly, and stroke patients. PLoS One 10:e0124414. https://doi.org/10.1371/journal.pone.0124414

Chen C, Jafari R, Kehtarnavaz N (2015) Utd-mhad: a multimodal dataset for human action recognition utilizing a depth camera and a wearable inertial sensor. In: Proceedings of the 2015 IEEE international conference on image process. (ICIP). IEEE, New York, pp 168–172

Cornacchia M, Koray OY, Velipasalar S (2016) A survey on activity detection and classification using wearable sensors. IEEE Sens J 17:386–403. https://doi.org/10.1109/JSEN.2016.2628346

Dietterich TG (2002) Machine learning for sequential data: a review. In: Caelli T, Amin A, Duin RPW, Ridder DD, Kamel M (eds) Structural, Syntactic, and statistical pattern recognition, vol 2396 of Lecture notes in computer science. Springer, Berlin, Heidelberg, pp 15–30

Eckelt M, Mally F, Brunner A (2020) Use of acceleration sensors in archery. Proceedings 49:98. https://doi.org/10.3390/proceedings2020049098

Gil-Martín M, San-Segundo R, Fernández-Martínez F, Ferreiros-López J (2020) Improving physical activity recognition using a new deep learning architecture and post-processing techniques. Eng Appl Artif Intell 92:103679. https://doi.org/10.1016/j.engappai.2020.103679

Ke Y, Sukthankar R, Hebert M (2005) Efficient visual event detection using volumetric features. In: Proceedings of the tenth IEEE international conference on computer vision (ICCV'05), vol 1. IEEE, New York, pp 166–173

Kołakowska A, Szwoch W, Szwoch M (2020) A review of emotion recognition methods based on data acquired via smartphone sensors. Sensors 20:6367. https://doi.org/10.3390/s20216367

Kononenko I, Šimec E, Robnik-Šikonja M (1997) Overcoming the myopia of inductive learning algorithms with RELIEFF. Appl Intell 7:39–55. https://doi.org/10.1023/A:1008280620621

Laptev I, Marszałek M, Schmid C, Rozenfeld B (2008) Learning realistic human actions from movies. In: Proceedings of the 2008 IEEE Conference on Computer Vision and Pattern Recognition. IEEE, New York, pp 1–8

Lara OD, Labrador MA (2013) A survey on human activity recognition using wearable sensors. IEEE Commun Surv Tutor 15:1192–1209. https://doi.org/10.1109/SURV.2012.110112.00192

Li L, Zhang H, Jia W, Mao Z-H, You Y, Sun M (2011) Indirect activity recognition using a target-mounted camera. In: Qiu P, Xiang Y, Ding Y, Li D, Wang L (eds) Proceedings of the 2011 4th international congress on image and signal processing. IEEE, New York, pp 487–491

Lima WS, Souto E, Rocha T, Pazzi RW, Pramudianto F (2015) User activity recognition for energy saving in smart home environment. In: Proceedings of the 2015 IEEE symposium on computer and communication (ISCC). IEEE, New York, pp 751–757

Lima WS, Souto E, El-Khatib K, Jalali R, Gama J (2019) Human activity recognition using inertial sensors in a smartphone: an overview. Sensors 19:3213. https://doi.org/10.3390/s19143213

Minh Dang L, Min K, Wang H, Piran MJ, Lee CH, Moon H (2020) Sensor-based and vision-based human activity recognition: a comprehensive survey. Pattern Recognit 108:107561. https://doi.org/10.1016/j.patcog.2020.107561

Rednic R, Gaura E, Brusey J, Kemp J (2012) Wearable posture recognition systems: factors affecting performance. In: Proceedings of the 2012 IEEE-EMBS international conference on biomedical and health information. IEEE, New York, pp 200–203

Ronao CA, Cho S-B (2017) Recognizing human activities from smartphone sensors using hierarchical continuous hidden Markov models. Int J Distrib Sens Netw 13:1550147716683687. https://doi.org/10.1177/1550147716683687

Ryoo MS, Matthies L (2013) First-person activity recognition: What are they doing to me? In: Proceedings of the 2013 IEEE conference on computer vision and pattern recognition. IEEE, New York, pp 2730–2737

Serrà J, Arcos LJ (2014) An empirical evaluation of similarity measures for time series classification. Knowl Based Syst 67:305–314. https://doi.org/10.1016/j.knosys.2014.04.035

Shakerian R, Yadollahzadeh-Tabari M, Rad SYB (2022) Proposing a Fuzzy Soft-max-based classifier in a hybrid deep learning architecture for human activity recognition. IET Biomet 11:171–186. https://doi.org/10.1049/bme2.12066

Song K-T, Chen W-J (2011) Human activity recognition using a mobile camera. In: Proceedings of the 2011 8th international conference on ubiquitous robotics and ambient intelligent (URAI). IEEE, New York, pp 3–8

Varatharajan R, Manogaran G, Priyan MK, Sundarasekar R (2018) Wearable sensor devices for early detection of Alzheimer disease using dynamic time warping algorithm. Clust Comput 21:681–690. https://doi.org/10.1007/s10586-017-0977-2

Wang J, Chen Y, Hao S, Peng X, Lisha H (2019) Deep learning for sensor-based activity recognition: a survey. Pattern Recognit Lett 119:3–11. https://doi.org/10.1016/j.patrec.2018.02.010

Ward JA, Lukowicz P, Tröster G (2006) Evaluating performance in continuous context recognition using event-driven error characterisation. In: Hazas M, Krumm J, Strang T (eds) Location- and context-awareness. Springer, Berlin, Heidelberg, pp 239–255

Ward JA, Lukowicz P, Gellersen HW (2011) Performance metrics for activity recognition. ACM Trans Intell Syst Technol 2:6. https://doi.org/10.1145/1889681.1889687

Watanabe Y, Hatanaka T, Komuro T, Ishikawa M (2011) Human gait estimation using a wearable camera. In: Proceedings of the 2011 IEEE workshop on applied of computing vision. IEEE, New York, pp 276–281

Waterbolk M, Tump J, Klaver R, van der Woude R, Velleman D, Zuidema J, Koch T, Dugundji E (2019) Detection of ships at mooring dolphins with Hidden Markov Models. Transp Res Rec 2673:0361198119837495. https://doi.org/10.1177/0361198119837495

Wilmes E, de Ruiter CJ, Bastiaansen BJC, van Zon JFJA, Vegter RJK, Brink MS, Goedhart EA, Lemmink KAPM, Savelsbergh GJP (2020) Inertial sensor-based motion tracking in football with movement intensity quantification. Sensors 20:2527. https://doi.org/10.3390/s20092527

Yeh C-CM, Kavantzas N, Keogh E (2017) Matrix profile IV: Using weakly labeled time series to predict outcomes. In: Boncz P, Salem K (eds) Proceedings of the VLDB endow, vol 10. VLDB Endowment, pp 1802–1812

Zhu C, Sheng W (2011) Motion- and location-based online human daily activity recognition. Pervasive Mob Comput 7:256–269. https://doi.org/10.1016/j.pmcj.2010.11.004