Machine learning for recognition of individuals from motion capture time series: performance and explainability

Elena Mariolina Galdi[1,*], Marco Alberti[3], Alessandro D'Ausilio[4] and Alice Tomassini[5]

[1]Dipartimento di Ingegneria, Università di Ferrara, Ferrara, 44122, Italy
[3] Dipartimento di Matematica e Informatica, Università di Ferrara, Ferrara, 44122, Italy
[4]Dipartimento di Neuroscienze e Riabilitazione, Università di Ferrara, Ferrara, 44121, Italy
[5]Istituto Italiano di Tecnologia, CTNSC@Unife, Ferrara, 44121, Italy

Abstract

In this paper we describe an ongoing research project in which we investigate the capability of AI-systems to recognize individuals from motion capture data, e.g. using a neural network. In our previous work [1] we also showed which motion features more strongly characterize each individual.

In addition, we report on the application of some techniques suggested by Explainable AI's literature. In particular we have analyzed the parsimonious linear fingerprinting (PLiF) [2] and a specific learning shapelets method suggested by Tavenard [3].

Keywords

Explainable AI, Convolutional Neural Networks, Motion Capture, Movement Analysis, Parsimonious linear fingerprinting, Shapelets, Individual Motor Signature.

1. Introduction

The possibility of using AI models in order to recognize an individual on the basis of his/her movements or gestures has been studied in depth in the past years due to its significant applications in the security and medical areas. At the same time, it has become more and more important to provide an explanation for the decisions made by AI models and the European GDPR makes this point very clear (art. 13-15,22).

In [1], starting from a dataset derived from a recent neuroscience project on interpersonal behavioral coordination across multiple temporal scales [4], we demonstrated that it is possible to identify a subject from index finger extension and flexion using a convolutional neural network (CNN). In addition, we carried out an in-depth post-hoc interpretation [5] of our results to understand the movement characteristics that are more relevant for identification.

In particular we have investigated whether the recurrent discontinuities that characterize the microstructure of human movement composition (tiny recorrective speed-bumps in the range of 2-3 Hz which are often called sub-movements) play a crucial role in this recognition process. To investigate which movement features (i.e. temporal scales) are more relevant for the neural network, we decided to decompose the time series on the basis of their spectral content and we evaluated their impact of this and other key preprocessing choices on recognition accuracy. In particular, we showed that the frequency produces the largest impact on the ability of a CNN to recognize individuals from the movement of their finger is the fundamental harmonica. However, we also noted that this frequency is not sufficient to reach significant accuracy, and higher frequencies are needed to achieve this goal.

In addition to our previously published work, we here want to share our first results in applying some more structured XAI techniques. In particular we have analyzed the parsimonious linear fingerprinting (PLiF) [2] and a specific learning shapelets method suggested by Tavenard[3]. The remainder of this paper is organized as follows. After a brief overview on explainable AI for time series in Section 2, the experimental settings are described in Section 3. Than, we show the most significant experimental results in Section 4. We conclude the paper (Section 5) with a discussion of the results and possible directions for future research.

Ital-IA 2023: 3rd National Conference on Artificial Intelligence, organized by CINI, May 29–31, 2023, Pisa, Italy
elenamarioli.galdi@edu.unife.it (E. M. Galdi); marco.alberti@unife.it (M. Alberti); alessandro.dausilio@unife.it (A. D'Ausilio); alice.tomassini@iit.it (A. Tomassini)
0009-0009-2926-5161 (E. M. Galdi); 0000-0003-4712-3721 (M. Alberti); 0000-0003-1472-6200 (A. D'Ausilio); 0000-0003-2986-020X (A. Tomassini)

2. Explainable AI for Time Series

Let us begin by providing a definition of the time series [6]. A *Time Series* $x = \{t_1, t_2, ...t_m\}, \in \mathbb{R}^{m \times d}$ is a sequence of m d-dimensional real valued observations or time steps t_i. We talk about *univariate time series* if $d = 1$, and *multivariate time series* when $d > 1$. Furthermore, a *time-series classification dataset* $D = (X, Y)$ is a set of n time series, $X = x_1, x_2, \ldots, x_n \in \mathbb{R}^{n \times m \times d}$, with a vector of assigned labels (or classes) $Y = y_1, y_2, \ldots, y_n \in \mathbb{N}^n$. For dataset D containing l classes, y_i can take l different values. When $l = 2$, D is a binary classification dataset, whereas when $l > 2$, D is a multiclass classification dataset. In our case we have 60 classes that correspond to the 60 participants to the experiments, and so $l = 60$.

We can now define the problem of **time-series classification** , TSC, as follows: given a TSC dataset D, TSC is the task of training a function or mapping f from the space of possible inputs X to a probability distribution over the values of classes Y.

Rojat et al. in their work [7], present XAI methods specific for time series classification when CNNs is used. In particular, they introduced a *perturbation-based methods* that directly compute the contribution of the input features by removing, masking, or altering them, running a forward pass on the new input, and measuring the difference with the original input.

Our XAI approach was primarily the application of an agnostic, post-hoc model in which we modified some preprocessing parameters, as presented in Section 3.3, and then evaluated how these modifications affected the accuracy of the neural network. Therefore, we apply what Rojat defines a perturbation-based methods.

Since NN is not the only way to detect possible correlations between time series, we decided to explore other methods with the main purpose of extracting the most significant features within the time series that make classification possible.

A particular explanation method consists of extracting from the time series the sub-sequences of values that are most representative of class membership. These type of sub-sequences are called *shapelets* [6]. Shapelets were first introduced in [8] and are calculated by finding the subsequences that maximize the information gain when dividing the set of all subsequences into two classes based on their distance from the candidate.

Tavenard [3] proposed a "Learning Shapelets" method in order to learn a collection of shapelets that linearly separates the timeseries. The objective of this library, named *tslearn*, is to extract K shapelets which are then used to transform the input time series in a K-dimensional space, which is called the shapelet-transform space in the related literature.

On other method we try to apply in our research was the *parsimonious linear fingerprinting*(PLiF) for time series. PLiF, is a method to discover essential characteristics ("fingerprints"), by exploiting the joint dynamics in numerical sequences [2]. The main idea is to extract the essential numerical representation that characterizes the evolving dynamics of the sequences, trying to find clusters and to group similar motions together. At the high level, PLiF uses a modified, faster way of learning a Standard Linear Dynamical Systems (LDS) or Kalman filters, normalizes the resulting transition matrix, which reveals the natural frequencies and groups some of those harmonics/hidden variables, after ignoring the phase, thus accounting for lag-correlations.

The discovered groups of such frequencies are exactly the "fingerprints" (features) that PLiF is using for clustering, visualization, compression, etc.

In both cases, PLiF and Learning Shapelets, the main objective is to group the time series based on certain characteristics: the shape of the subseries for shapelets, the frequencies in the case of PLiF.

3. Experimental Settings

3.1. Dataset

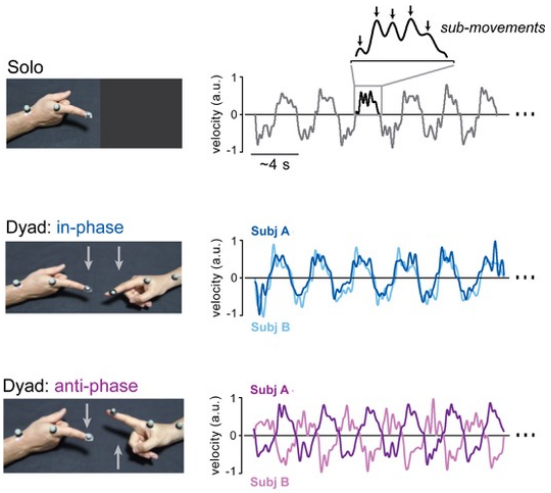

Figure 1: On the left, the experimental setup for data collection. From top to bottom, there are three settings for the solo, in-phase, and anti-phase tasks. The right panel shows the speed profile for the three different cases. Figure granted by the research of Tomassini et al.[4]

The dataset we have been working on comes from previous research; all experimental instrumentation is described in depth in [4]. In total, 60 participants, forming 30 pairs, performed a movement synchronization task. Participants were instructed to maintain a slow movement rhythm (full movement cycle: single flex-

ion/extension movements) by having them practice in a preliminary phase with a reference metronome set at 0.25 Hz. As shown in Figure 1, participants performed the experiment under three different conditions. In the first case, they performed alone (solo condition) with the only requirement being that they adhere to the instructed rhythm. The other two experiments were conducted in pairs, where they were asked to keep their right index fingers pointed toward each other (without touching) and to perform rhythmic flexion-extension movements as synchronous as possible, either toward the same direction (in-phase) or toward opposite directions (antiphase). Each trial had a duration of 2.5 minutes.

3.2. Application Architecture

We decided to develop our AI program with Python. The software was essentially split in two main components. The first module assigned to the preprocessing that is common to all the different XAI techniques applied later on. The second executed module it depends on the XAI technique we decided to used.

The first module, described in chapter §3.1, has a composite structure with different possible choices to preprocess the data and different parameters to set.

Once the data preprocessing has been completed, the segmented series are sent to the second module. In our previous work we have analyzed the response of a neural network. The TensorFlow Keras library was used to generate the neural network model. A Convolutional Neural Network (CNN) as been built for multiclass classification (60 classes, one for each partecipant). In this research we wanted to investigate two other techniques PLiF and learning shapelets, that are specific techniques for XAI in case of time series and already introduced in section 2.

3.3. Preprocessing Techniques

In the preprocessing phase, it is possible to independently choose the series type, filter method, type of segmentation, and whether the series must be normalized. Depending on the choice made, it is necessary to specify different input parameters. Table 1 lists the different parameters required for each pre-processing choice.

Choice	Parameter
MAW	Window Dimension
Band Pass	Low and High frequency cut
Extension-Flexion	Resized Subseries Dimension
Sliding Window	Subseries Dimesion and gap

Table 1
Parameters needed in function of different choices

3.3.1. Series Cut

We considered two different methods of cutting the series. As a first approach, we cut the time series corresponding to the **maximum finger positions** on the x-axis. In this way, each subseries represents a complete movement, extension and flexion of the index finger. However, this type of cutting creates a subset of different lengths that cannot be used directly as input to a CNN. This means that we had to resize the subsets to a fixed default size that had to be the same for all of them.

The second option for cutting the time series is called **sliding windows** and decides a priori the size of the subseries to be obtained and the gap between the next two subseries. We applied this method to check whether there was hidden information in the time series that was not related to the whole motion cycle (extension-flexion).

3.3.2. Series Filtering

We also studied the influence of time series filtering. Therefore, we applied two different types of filters to our data.

Moving average window (MAW) is a basic tool commonly used in time series to smooth signals. For each point in the series, a fixed number of successive points are averaged, and the result replaces the starting point. Clearly, the number of points involved in averaging corresponds to the size of the window: the larger the window, the smoother the signal will be.

We also applied to the signal standard **frequency filters**. Basically, we created a bandpass filter in which low and high cutoff frequencies can be set. If the low frequency cutoff was set to 0, it was applied as a low-pass filter. For this purpose, we created a Butterworth filter using the default function of the scipy. signal library in Python.

3.4. Neural Network Architecture

We decided to apply a CNN, as suggested in the literature for multi-class classification of time series data [9] [10] [11].

The structure of the neural network is described in the table 2. We used RMSprop as the optimizer and performed early stopping to avoid overfitting. We compared the results obtained with this neural network with a similar network made with a pytorch instead of tensorflow.keras. The results of these two networks are very similar.

4. Results

Hereafter we summarize the results obtained from our previous research, where, to evaluate the impact of dif-

1D Convolution	
Kernel: 3@64	ActFunct: ReLU
1D Convolution	
Kernel: 3@64	ActFunct: ReLU
1D Max Pooling	
Pool Size: 2	
1D Convolution	
Kernel: 3@64	ActFunct: ReLU
1D Convolution	
Kernel: 3@64	ActFunct: ReLU
1D Max Pooling	
Pool Size: 2	
Dense	
Dense	
Softmax	

Table 2
Structure of the Convolutional Neural Network

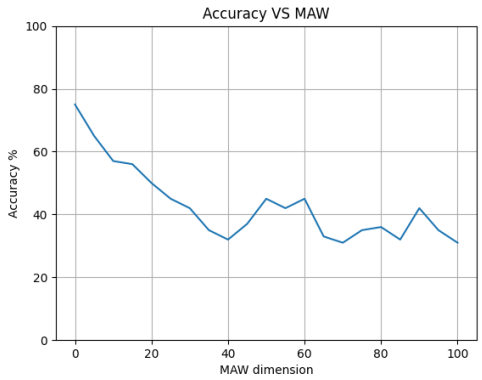

Figure 2: The figure shows how the accuracy changed as a function of the number of points included in the moving average.

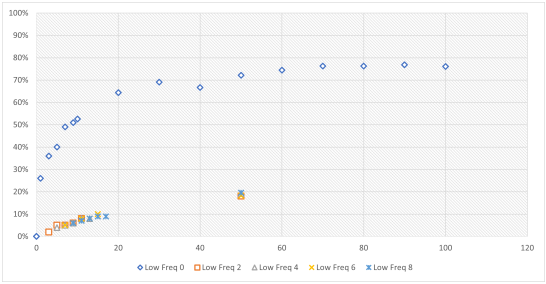

Figure 3: In the figure, it is reported how the accuracy changes for different band pass filters: each curve corresponds to a filter with a specific low-frequency cut (0 Hz, 2 Hz, 4 Hz, 6 Hz, 8 Hz), while in the x-axis, the high-frequency cut is reported.

ferent parameters on recognition performance, we examined the accuracy of our CNN. First, we found that by using a low-pass filter set at 50 Hz and cut based on the maximum finger position, the accuracy of CNN was higher considering the normalized speed profile as input, than the one obtained with the position or acceleration profiles.

The experiment comparing the two types of series segmentation doesn't show significant differences in terms of accuracy, while the computational time was drastically longer when the data was cut with **sliding windows**. This convinced us to choose the **cut at the maximum finger position** as the standard segmentation method for our series.

As shown in Figure 2, the maximum accuracy obtained by applying **MAW**, occurs with a window size of zero, that is, when no filter has been applied. As the window size increases, the accuracy decreases until it reaches 30% with a window size of 100 points. Remember that when MAW windows include 100 points, only the main shape of the motion is visible. The change in accuracy as a function of different **frequency filters** is shown in Figure 3. As can be clearly seen, the fundamental frequency (0.25 Hz or the instructed rhythm of finger flexion-extension) is the most significant frequency, and if it is removed from the signal, no recognition can be made. The fact that each individual is characterized by its own preferred tapping rhythm is well known in the neurophysiological literature [12]. Moreover, our experiments show that this frequency alone is not sufficient and accuracy increases with the addition of other frequencies. Interestingly, it is known in neurophysiology that in motion, even with the differences that may exist between the movement of a finger or a leg, frequencies above 15 Hz begin to be attenuated. From 20-30 onward, there is no longer any physiological relevance. Instead, in our experiments, we still see further increases when frequencies above 30 Hz are added, meaning that the network is still learning something. Future in-depth analyses should investigate what our neural network is learning in the range between 30 Hz and 70 Hz. In any case, at 20 Hz, the accuracy is already 65 percent, a very good performance for classification among the 60 classes.

Since one of our initial goals was to investigate the role of sub-movements in defining individual motor signatures, we focused our attention on the frequencies of 2-4 Hz. Although we did not notice any significant change in the accuracy of the bandpass filter with a low cutoff at 2 Hz and a high cutoff at 4 Hz, we could clearly observe a significant increase in the slope of the low-pass filter around these frequencies (figure 4). As explained earlier, the fundamental frequency (0.25 Hz) probably contained most of the information (less than 30% accuracy). However, the performance is far from their plateau;

indeed, the model improves vastly with the addition of the secondary motion interval.

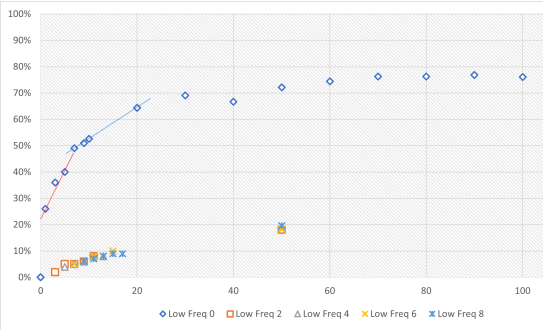

Figure 4: The figure shows that the accuracy increases as the frequencies change. In particular, the slope in the accuracy from 0 to 4 Hz is shown in red, and that from 6 to 20 Hz is shown in blue.

As we said at the beginning of this paper, CNN is just one of the possible path that can be follow when we want to classify time series. In our current work we make some first experiment using PLiF and tslearn trying to find if it's possible to group in cluster the time series due to some intrinsic characteristics and than verify if these clusters may identify the correct class of the TS that means the owner of the movement. As a starting experiment, we use a low-pass filter set at 50 Hz and cut based on the maximum finger position and the resulting time series were use as input for both tslearn and PLif.

For PLiF we used the mathlab tool proposed by Li [13]. We tried applying the tool to a smaller csv file than the one obtained from the preprocessing python program described above. The initial goal was just to verify that the tool worked correctly with our inputs. For this purpose we created inputs with an increasing number of series, at first only 5 series belonging to 2 different classes and then gradually increasing the number of series and classes involved. The figure 5 shows the result of applying PLiF to 5 time series belonging to 2 different classes. Unfortunately, it is not possible to see a clear distinction between the series belonging to the 2 classes.

Similar results we obtained with tslearn. In this case we gave as inputs to the model four series that belong to four different classes, and asked the model to use four shapelets to identify peculiar sub-sequences in the series. The results are proposed in the figure 6. It's clear that none of the shapelets can make a clear distinction among the series.

5. Discussion

Our previous work demonstrated that it is possible to recognize subjects by starting from their index finger

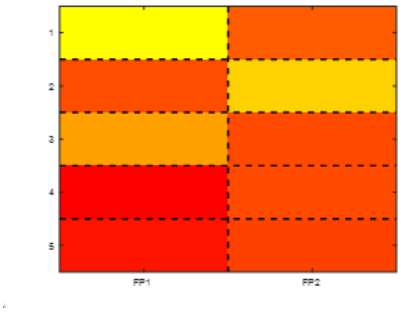

Figure 5: The figure shows the output of Li's Matlab software to extrapolate the fingerprint from the series. The inputs were 5 series belonging to two different classes: starting from the top, the first three rows belong to the first class, while the others belong to the second class. The similar colors indicate that the series are considered similar by the classifier.

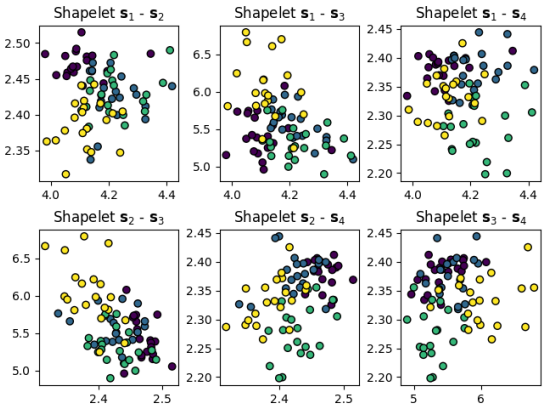

Figure 6: Each of the six graphs represent a space made by two different shapelets at a time. In each plot it's shown the capability of the two shapelets of distinguish series that belongs to different classes

movements.

In addition, we found that the fundamental frequency is critical in subject recognition, but not by itself. Higher frequencies contribute significantly to increasing accuracy, but only in the presence of the fundamental frequency. It would be interesting to be able to explain the gap between the 30 percent accuracy obtained with the fundamental frequency alone and the 75 percent accuracy obtained with a low-pass filter with a bandwidth of 60-70 Hz.

Our current goal was to understand whether secondary movements play a central role in the recognition process, and although it has not been fully demonstrated, we have some clues. The slope of the accuracy curve has provided us with a cue for further investigation in this direction.

In this work we tried to answer this question by applying more structured clustering method such as tslearn or PLif, section 2, to determine what other signal features most influence CNN classification. The experiments done so far to extract meaningful shapelets by applying a Python library (*tslearn*) implemented by Tavenard [3], did not get any meaningful results. Infact, it was not able to find shapelets that clearly distinguish series that belongs to different classes.

Also with PLif, the obtained results were not significant. This is easier to explain because PLif build up the clustering based on the spectral analysis of the signals. In our case, the fundamental harmonic was established by the person conducting the experiment and a more sophisticated analysis is needed to extract meaningful peculiarities.

This result is also consistent with an attempt we made to classify the spectra of the time series with the neural network described in Section 3.4, which was unsuccessful as well.

A possible explanation for the unsatisfactory results obtained by the two methods we applied is that they were created with macro classifications in mind (distinguishing between walking and running, walking on a hard (concrete) or soft (carpeted) floor), while in our case we wish to classify among 60 different classes.

At the same time, we do not want to abandon these two interesting methods of XAI for future applications in neurophysiology. In fact, one possible development of our research is to investigate early development of neurodegenerative diseases such as Parkinson's.

Acknowledgments

This work was partly supported by the University of Ferrara FIRD 2022 project "Analisi di serie temporali da motion capture con tecniche di machine learning".

References

[1] E. M. Galdi, M. Alberti, A. D'Ausilio, A. Tomassini, Why can neural networks recognize us by our finger movements?, in: A. Dovier, A. Montanari, A. Orlandini (Eds.), AIxIA 2022 – Advances in Artificial Intelligence, Springer International Publishing, Cham, 2023, pp. 327–341.

[2] L. Li, B. A. Prakash, C. Faloutsos, Parsimonious linear fingerprinting for time series, Proceedings of the VLDB Endowment 3 (2010) 385–396. doi:10.14778/1920841.1920893.

[3] R. Tavenard, J. Faouzi, G. Vandewiele, F. Divo, G. Androz, C. Holtz, M. Payne, R. Yurchak, M. Rußwurm, K. Kolar, E. Woods, Tslearn, a machine learning toolkit for time series data, Journal of Machine Learning Research 21 (2020) 1–6. URL: http://jmlr.org/papers/v21/20-091.html.

[4] A. Tomassini, J. Laroche, M. Emanuele, G. Nazzaro, N. Petrone, L. Fadiga, A. D'Ausilio, Interpersonal synchronization of movement intermittency, iScience 25 (2022) 104096. doi:10.1016/j.isci.2022.104096.

[5] A. Preece, Asking 'Why' in AI: Explainability of intelligent systems – perspectives and challenges, Intelligent Systems in Accounting, Finance and Management 25 (2018) 63–72. doi:10.1002/isaf.1422.

[6] A. Theissler, F. Spinnato, U. Schlegel, R. Guidotti, Explainable AI for Time Series Classification: A Review, Taxonomy and Research Directions, IEEE Access 10 (2022) 100700–100724. doi:10.1109/ACCESS.2022.3207765.

[7] T. Rojat, R. Puget, D. Filliat, J. Del Ser, R. Gelin, N. Díaz-Rodríguez, Explainable Artificial Intelligence (XAI) on TimeSeries Data: A Survey, 2021. arXiv:2104.00950.

[8] L. Ye, E. Keogh, Time series shapelets: A new primitive for data mining, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '09, ACM Press, Paris, France, 2009, p. 947. doi:10.1145/1557019.1557122.

[9] Z. Cui, W. Chen, Y. Chen, Multi-Scale Convolutional Neural Networks for Time Series Classification, 2016. arXiv:1603.06995.

[10] Y. Zheng, Q. Liu, E. Chen, Y. Ge, J. L. Zhao, Time series classification using multi-channels deep convolutional neural networks, in: F. Li, G. Li, S.-w. Hwang, B. Yao, Z. Zhang (Eds.), Web-Age Information Management, Springer International Publishing, Cham, 2014, pp. 298–310.

[11] H. I. Fawaz, G. Forestier, J. Weber, L. Idoumghar, P.-A. Muller, Deep learning for time series classification: A review, Data Mining and Knowledge Discovery 33 (2019) 917–963. doi:10.1007/s10618-019-00619-1. arXiv:1809.04356.

[12] B. H. Repp, Y.-H. Su, Sensorimotor synchronization: A review of recent research (2006–2012), Psychonomic Bulletin & Review 20 (2013) 403–452. doi:10.3758/s13423-012-0371-2.

[13] L. Li, PLiF, 2010. URL: https://github.com/lileicc/dynammo.