The Importance of Multiple Temporal Scales in Motion Recognition: when Shallow Model can Support Deep Multi Scale Models

Vincenzo D'Amato, Luca Oneto, Antonio Camurri, Davide Anguita
University of Genoa
Via Opera Pia 11a, 16145, Genova, Italy
{vincenzo.damato,luca.oneto,antonio.camurri,davide.anguita}@unige.it

2022 International Joint Conference on Neural Networks (IJCNN) | 978-1-7281-8671-9/22/$31.00 ©2022 IEEE | DOI: 10.1109/IJCNN55064.2022.9892372

***Abstract*—The execution of a human movement involves different muscles that are activated and coordinated by the brain at different temporal scales in a complex cognitive process. For this reason, studying human motion requires to properly model multiple temporal scales that fully describe its complexity. Current approaches are not able to address this requirement properly or are based on oversimplified models with obvious limitations. Data-driven models represent research frontiers able to provide new insights.**

In this work we will investigate different data-driven approaches. The first one is based on shallow models that, while achieving reasonably good recognition performance, require to handcraft features according to the domain knowledge The second one is based on deep models that can be extended to manage multiple temporal scales but they are hard to exploit as too many architecture configurations exist. For this reason, we will propose a new deep multiple temporal scale data-driven model, based on Temporal Convolutional Network, capable of learning features from the data at different temporal scales, of outperforming state of the art deep and shallow models, and of exploiting shallow models to tune the architecture configuration.

We designed, collected data and tested our proposal in a specially devised experiment, to prove the validity of our approach. In particular, we collected motion capture data about dyad actions where two people exchange a ball. As the weight of the ball and the throwing intentions change, we will show how it is possible to automatically detect either the weight of the ball or the intention behind the throw just based on motion data. Data regarding our experiment and code of the methods proposed in this work are also made freely available to the research community.

Results support both the proposal and the need for the use of deep multi scale models as a tool to better understand human movement and its multiple time scale nature.

***Index Terms*—Motion Recognition, Multiple Temporal Scales, Shallow Learning, Deep Learning, Feature Engineering, Feature Learning, Open Data, Open Implementation**

I. Introduction

Social Signal Processing (SSP) aims at bridging the social intelligence gap between humans and machines [1]. However, different definitions of social signals exist [2]–[4]. The one we opt in this work is the one of Vinciarelli et al. [2] which describe them as observable behaviours that produce tangible changes in others, whether this means modifying their inner state, modifying their observable behaviour, or changing their beliefs about the social setting. Vinciarelli et al. [1], also distinguish three major components in SSP, i.e., modelling, analysis, and synthesis of social behaviour. In particular, the modelling phase focuses on laws and principles of social interaction and how non-verbal behaviour influences them. Secondly, the analysis phase focuses on the development of automatic techniques for extracting and interpreting non-verbal behavioural cues in data. Finally, the synthesis phase focuses the automatic generation of appropriate non-verbal behaviour. These three aspects constitute the foundation of SSP.

In social signals, non-verbal behaviour surely conveys a great amount of information [5], [6]. Non-verbal behaviour are primary related to full-body movements (e.g., gesture [7] and posture [8]) and secondary to cues for the perception and interpretation of social signals (e.g., facial expression [9] and mutual gaze [10]). Consequently, understanding and interpreting them represent a fundamental phase in order to understand social interactions. In this work, we focus on non verbal behaviour conveyed through full-body movement. These movements can be effectively and efficiently acquired with Motion Capture (MOCAP), a technology that allows to capture both fine and gross movement features, while humans do it (unconsciously) with much lower level of details and accuracy [11]. Their interpretation, instead, can be easily performed by humans while for machines it can be a challenging task [11], [12]. For example, Caramiaux et al. [13] discussed how difficult it is to understand what makes a gesture expressive because this operation implies the consideration of different aspects such as dynamics, the mechanism that enacts it and spatial location.

Recent studies [14]–[16] highlighted how human movements are hierarchically nested. In other words, a layered movement structure involves muscles of different size, responsiveness and precision with a complex non-linearly stratified temporal dimension where every muscle has its own temporal scale. Even the simplest actions are composed by a set of sub-actions involving different parts of the body which cooperate to create a resulting smooth, effective, and expressive movement in a complex multiple temporal scale cognitive task. For instance, the action of grasping a glass of water involves different parts of the body: e.g., starting from the movement of the entire body, followed by the upper body part, then the shoulder, the arm, the fingers and so on. Following the example, it is easy to understand how the human body can be described as a kinematic tree, consisting of joints connected to each other. As a first raw approximation, we could state

that larger muscles are always slower and characterised by a slower perceptual response over time than smaller muscles. This consideration is not often true since some movements of larger muscles can be fast: for instance, the necessary postural corrections that enable us to keep our balance when we are at risk of falling involve large muscles that are quickly activated in order to safeguard us from the risk of a fall. Moreover, the multiple temporal scales nature of the human movements characterises also how humans perceive other people's movements [17]. For example, Johansson et al. [18] discussed how human beings are capable of understanding and predicting the movements of other humans even from a limited number of moving points. More in detail, this skill depends on the human ability to create relations between different temporal and spatial layers using forward/feedback connections. This process is driven by the brain and the body as time keepers that coordinate different internal, mental, and physiological clocks. However, it is worth noting that recent studies [19] show that the information contained in this limited number of moving points depends not only on the activity performed but also on more complex cognitive and affective phenomena. For example, Meeren et al. [20] consider the stimulus relation as a temporal dynamic of feedback, and affective qualities.

Although it is nowadays well known that human movement and perception of human movement are characterised by multiple temporal scales [14]–[17], [20], few works in the literature are focused on studying this particular property. For example, Ihlen et al. [21] provided quantitative support for studying multiple time scales in human action and perception using wavelet-based multifractal analysis in the response of four cognitive tasks (i.e., simple response, word naming, choice decision, and interval estimation). Camurri et al. [22] demonstrated that computational models of expressive qualities should operate at different time scales starting from previous research on human perception and dance theories [23]. Later, still Camurri et al. [22] proposed a framework in which features are computed at different levels, i.e., low-level features (e.g., speed) are computed instantaneously while higher-level features (e.g., impulsivity) are computed on a larger time scale. Recent research, conducted within the European FET PROACTIVE project EnTimeMent[1], focuses on the importance of multiple time scales in motion analysis and prediction. And also recent research works focus on these issues. For example, Beyan et al. [24] proposed an approach that is able to model the dynamics of whole-body motion data represented on multiple time scales where features are processed by two independent, parallel shallow Temporal Convolutional Networks. Yao et al. [25] showed how a 3D Convolutional Network based on multiple temporal scales outperformed standard deep learning model with a single temporal scale in action recognition tasks. Stergiou et al. [26] proposed a novel convolutional block (MTConv) useful to extract spatio-temporal patterns in action recognition problems. Lin et al. [27] proposed a novel multi-scale temporal information extractor able to aggregate temporal information from different temporal scales in gait recognition tasks.

In the last few years, data-driven models (based on Machine Learning and Data Mining) played a key role in the advancement of SSP [28]. This is due to the availability of large amounts of data required by these models to work properly and effectively [29], [30]. In fact, data-driven models have the ability to extract useful and actionable information from these large amounts of data to provide insights on the complex processes in social signals as core tools able to empower and supplement expert-or-physics-based models [31]. This trend has been lately exacerbated by the unexpected success of these tool in solving real world problem with super-human performance [32]–[35] or the expectation to do so in the near future [36]. Following this trend, in this work we will investigate how data-driven methods can push forward the research in SSP, with a particular reference to the non-verbal full body movement understanding focusing on the importance of multiple temporal scales. In particular, we will first show how shallow data-driven models [37], i.e., models that require handcrafted features based on domain-specific knowledge, can achieve good recognition performance and their limitations in handling the multiple temporal scales that characterise human movement. Secondly, we will investigate how deep models [29], which are able to automatically learn features from the data at different temporal scales, cannot be naively applied to address the problem because of the limited of sample available in most application and because of the huge number of architectural choices that need to be explored in order to achieve optimal results. In fact, we will compare different deep models, i.e., Long-Short Term Memory network (LSTM) [15], [38], [39] and Temporal Convolutional Network (TCN) [40]–[43] showing that their naive application results in recognition performance below the ones of shallow models. Then we will propose to blend shallow and deep models taking the best of the two worlds to achieve optimal performance. In particular, we will use shallow models to reduce the number of possible architectural choices (i.e., reducing the number of MOCAP input time series) and an empowerment of current TCN to capture and learn the multiple temporal scale nature of the phenomena under investigation.

In order to prove the validity of our study, we designed, collected data and tested our hypothesis in a specially devised experiment. Specifically, we analysed human movement in dyad actions where two people exchange a ball. As the weight of the ball and the throwing intentions change, we will show how it is possible to automatically detect either the weight of the ball or the intention behind the throw. A publicly available dataset of our experiment is available for the scientific community as well the code developed to generate all the results presented in this paper[2].

The rest of the paper is organised as follows. Section II describes our experiment on human movement and details the related collected data. Section III presents state-of-the-art methods and the newly proposed ones that we employed in this work. Section IV reports the results in applying the methods described in Section III on the data collected in Section II. Section V concludes the paper.

[1]https://entimement.dibris.unige.it

[2]https://github.com/lucaoneto/IJCNN2022_Balls

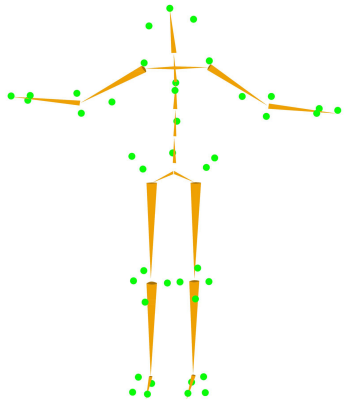


Fig. 1: Physical markers (in green) and the resulting skeleton of 24 main joints (in orange) of the body.

Group	Ball: Light: Intention: Fair	Light: Aggressive	Light: Deceptive	Heavy Ball: Fair	Heavy Ball: Aggressive	Heavy Ball: Deceptive	Tot
1	17	13	11	13	13	9	76
2	9	13	11	9	11	13	66
3	11	12	14	19	18	13	87
4	9	11	13	19	13	15	80
5	9	13	9	9	11	9	60
6	21	26	21	23	19	10	120
7	11	17	9	16	15	11	79
8	19	19	30	29	29	25	151
9	9	15	15	15	11	11	76
10	9	9	15	15	15	9	72
11	9	9	13	15	15	15	76
12	9	13	15	15	11	11	74
13	9	15	13	15	15	15	82
Tot.	151	185	189	212	196	166	1099

TABLE I: Raw Dataset

II. Problem Description and Data Availability

In order to study non-verbal full body movement understanding focusing on the importance of multiple temporal scales, we designed, collected data and tested our hypothesis in a specially devised experiment. Specifically, we analysed human movement in dyad actions where two people exchange a ball of different weights (light and heavy) with different intentions (fair, aggressive, and deceptive). The scope is to automatically detect, just based on MOCAP data, what is

- the weight of the ball, i.e., light or heavy;
- the intention of the ball exchange, i.e., fair, aggressive, or deceptive.

The data employed in the current work were collected during the EnTimeMent FET PROACTIVE project[3]. The project studies how multi-temporal scales, which is an intrinsic property of the human brain, can be used to effectively detect movement behaviours in individual, group and dyad scenarios.

More in details, the full-body MOCAP configuration (i.e., the Sports Marker Set by Qualysis[4]) has been used to collect data that allows to detect the location of 24 main joints of the body (skeleton) based on 42 markers (see Figure 1) keeping track of the launcher and the receiver. Recordings were performed at Casa Paganini - InfoMus Research Centre of Genoa[5]. The movements of 26 participants were acquired, who are randomly assigned into 13 groups of two people (i.e., a person can belong to just one group). The two people in the group exchange the different balls (light or heavy) with different intentions (fair, aggressive, or deceptive) at a distance of approximately 3 metres. In fact, they are free to move in a fixed rectangular space of 1×3 metres (i.e., an island) identified by a visible tape on the floor. For what concerns the balls weight

- light ball weight was 0.1 kilograms;
- heavy ball weight was 2 kilograms.

For what concerns the launch intentions

- fair means that the two participants launch the ball to each other trying to facilitate the reception of the ball;
- aggressive means that the two participants launch the ball to each other trying to hit each others;
- deceptive means that the two participants launch the ball to each other trying to hinder the reception of the ball.

An example of the launches made with different ball weight and different intentions are shown in Figure 2.

During the experiment, participants started with the light ball. They are asked to exchange the ball a random number of times (from 10 to 30) first with fair, then with aggressive, and finally with deceptive intention. Then the experiment continues with the heavy ball using the same protocol. Some lunches have been discarded when something went wrong (e.g., people outside the island, problem of balance, etc.).

Participants were asked to throw the ball using two hands: this choice facilitates the involvement of the full-body in both the launch and the reception phases avoiding both too complex movement and too high speed in the launch, as typically happens with single hand launches.

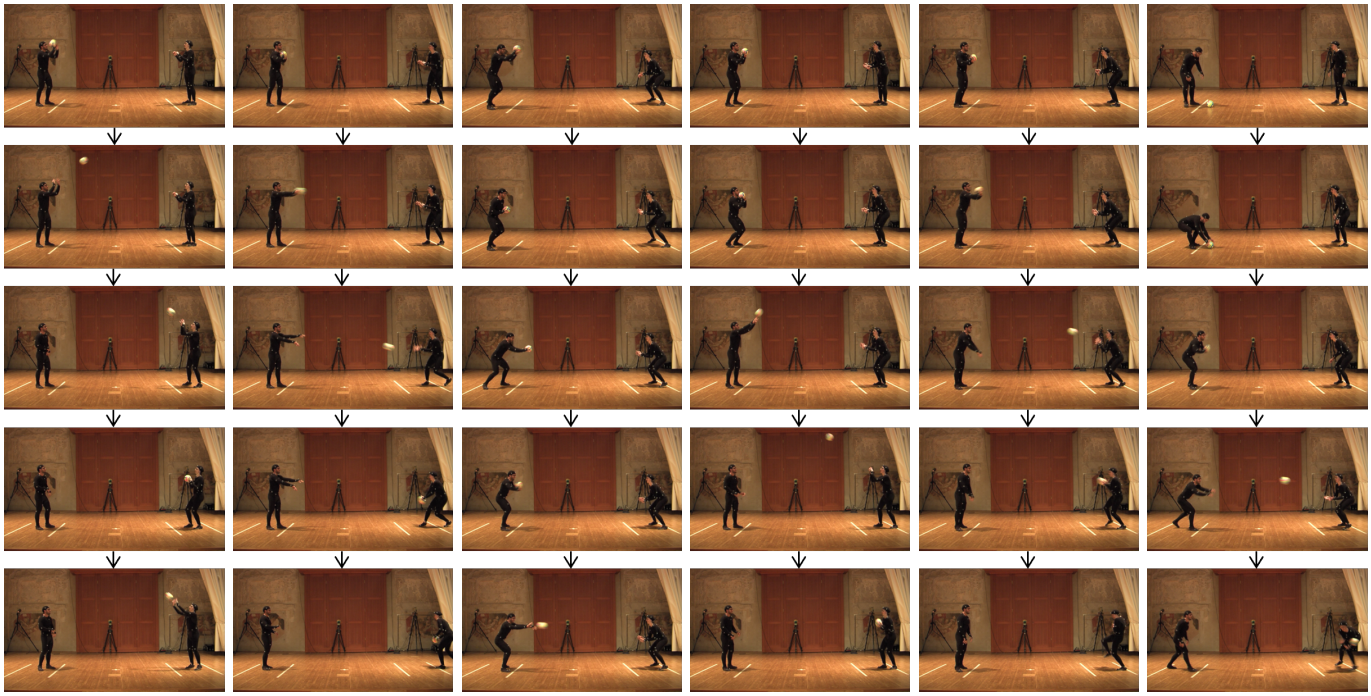
Note that, launching and receiving a ball contain both symmetric (launching requires an expansion while receiving a compression) and asymmetric (to launch a ball one foot is usually behind the other) actions which easily enable natural movements avoiding static postures.

For each launch we collected who is the launcher and who is the receiver and the position of the 24 joints of the skeleton (each joint gives x, y, and z position) with sampling rate of 60 Hertz for both launcher and receiver from the moment the launch started (from still position) until the receiver concludes the reception (to still position) for a total o 48×3 time series plus a boolean variable indicating who is launching.

The resulting raw dataset is described in Table I.

Then, the raw datasets have been normalised as follows. For the launcher, the $24x3$ time series has been translated into 24 time series corresponding to distance, in time, between each of the 24 joints of the skeleton and the barycentre of the 24 joints of the skeleton. The same has been done for the receiver. Then the distance between the barycentres, in time, has been added. The resulting dataset consists then in 49 time series (first the 24 of the launcher then the 24 of the receiver and then the distance between their barycentres) for each of the launches organised as in Table I.

[3]https://entimement.dibris.unige.it/
[4]https://www.qualisys.com/
[5]www.casapaganini.org



(a) Light&Fair (b) Light&Aggressive (c) Light&Deceptive (d) Heavy&Fair (e) Heavy&Aggressive (f) Heavy&Deceptive

Fig. 2: Example of launches for different Ball Weights (Light or Heavy) and different Launch Intentions (Fair, Aggressive, or Deceptive).

III. Methods

In this section, we will present to readers the methodology we followed in our analysis. We will start describing some preliminaries in Section III-A. Then we will continue with the presentation of the Shallow Models in Section III-B. Following Shallow Models we will present the Deep Models in Section III-C: noticing then that the architectural choices in Deep Models are too many to be reasonably explored and that the number of weights is too big for the size of our dataset, we will show how to leverage on Shallow Models to reduce this both the architectural choices and the weights to be tuned in the deep architectures. Finally, in Section III-D we will illustrate the different extrapolating scenarios (of increasing difficulty), namely we will try to understand if the models are able to extrapolate over different ball weights or different intentions or over different groups.

A. Preliminaries

The problem described in Section II, namely predicting the weight of the ball (light or heavy) or the intention of the launch (fair, aggressive, or deceptive) based on the 49 time series deriving from MOCAP, can be easily mapped into a now classical classification problem [37]. In particular, given an input space $\mathcal{X}$ (in our case the 49 time series deriving from MOCAP), an output space $\mathcal{Y} = \{1, \cdots, c\}$ (in our case $c = 2$ when predicting the ball weight and $c = 3$ when predicting launch intention), a series of example of the input/output relation, namely a dataset $\mathcal{D}_n = \{(X_1, Y_1), \cdots, (X_n, Y_n)\}$ with $X \in \mathcal{X}$ and $Y \in \mathcal{Y}$, an algorithm $\mathscr{A}_{\mathcal{H}}$, characterised by its hyperparameters $\mathcal{H}$, selects a model f inside a set of possible ones $\mathcal{F}$ based on $\mathcal{D}_n$. The quality of f in approximating the unknown input/output relation is measured by one or more metrics M. Many different metrics are available in literature [44] and, in this work, we will exploit: the percentage of accuracy (ACC), the precision (PRE), the recall (REC), and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC). Finally, note that, in order to tune the performance of the $\mathscr{A}_{\mathcal{H}}$ we need to optimise the choice of $\mathcal{H}$ which is not a trivial task, especially for deep models [45], [46].

B. Shallow Models

Shallow models rely on a simple idea. First $\mathcal{X}$ is mapped in a new space $\mathcal{X}'$ based on previous knowledge about the problem or based on classical signal processing techniques [47], [48]. Then, on this new space, a linear (e.g., Support Vector Machines [49]) or nonlinear (e.g. Kernel Methods [49] or Ensemble Methods [50], [51]) shallow modes are applied. Note that the first step is not always present, e.g., if the original data are already expressive enough or formatted in such a way it allows the direct application of shallow models.

In our case this step is fundamental since we cannot feed directly the 49 series into the classical shallow models for two reasons [52]. The first one is that the series are of different lengths. The second one is that simply feeding the 49 raw features as input to the problem will never work. For this reason, in our case we performed a feature engineering phase following state-of-the-art approaches proposed in [53]–[56] where classical signal processing techniques have been applied in order to extract from the time series a vector of

Function	Description
mean	Mean value
var	Variance
mad	Median absolute value
max	Largest value in array
min	Smallest value in array
sma	Signal magnitude area
energy	Average sum of squares
iqr	Interquartile range
entropy	Signal Entropy
correlation	Correlation coefficient between series
kurtosis	Signal Kurtosis
skewness	Signal Skewness
maxFreqInd	Largest frequency component
argMaxFreqInd	Index largest frequency component
meanFreq	Frequency signal weighted average
skewnessFreq	Frequency signal Skewness
kurtosisFreq	Frequency signal Kurtosis
ampSprec	Amplitude Spectrum of the frequency signal
angle	Phase angle of the frequency signal

TABLE II: List of measures for computing feature vectors.

representation composed by variables such as the mean, the median, and the signal magnitude area for both the time and frequency domains (see Table II) for a total of $d = 2009$ ($\mathcal{X}' \subseteq \mathbb{R}^d$). The Fast Fourier Transform was used to obtain frequency components for each split considered. Table II shows the list of measures applied in both time and frequency domain signals. The complete list of features is also present in the repository associated with this work[2].

At the end of this feature engineering step, we applied a series of state-of-the-art top performing classification algorithms [6] [57], [58]: Linear and Nonlinear (Gaussian Kernel) Support Vector Machines [49] (respectively LSVM and KSVM), Random Forest (RF) [50], and XGBoost [51]. Each one of these models have a series of hyperparameters that need to be tuned. LSVM has the regularisation hyperparameter C while KSVM has both C and the kernel coefficient γ to be tuned. In RF we need to tune the number of features to randomly sample from the whole features during each node of each tree creation n_f, the maximum number of elements in each leaf of each tree n_l, and the maximum depth of each tree n_d. As RF performance improves increasing the number of trees n_t we set it to 1000 as a reasonably large number but yet computationally tractable. Finally, in XGBoost we need to tune the learning rate of the gradient l_r, the max dept of arch tree n_d, the minimum loss reduction m_l, number of training to randomly sample from the whole training set for each tree creation n_b, and the number of feature to randomly sample from the whole featured during each node of each tree creation n_f. The summary of these hyperparameters with the associated search space is reported in Table III.

In order to further improve the performance of the shallow models we decided to add a dimensionality reduction step to remove all the uninformative variables which may be numerous since the feature engineering phase is quite comprehensive [59]. In particular, for each training phase of each model we applied the permutation feature importance method [50], [60], [61], using the mean decrease of accuracy as a metric, and removed all the features with no positive impact according to this metric. Then we retrained the model on this reduced feature set.

The pipeline we proposed for shallow models is depicted in Figure 3.

[6]Results in Kaggle www.kaggle.com, the most popular Machine Learning competition website, shows how SVM, RF, and XGBoost algorithms are the top winner algorithms.

Algorithm	Hyperparameters
LSVM	$C : \{0.001, 0.01, 0.1, 1, 10, 100\}$
KSVM	$C : \{0.001, 0.01, 0.1, 1, 10, 100\}$ γ: $\{0.1, 0.01, 0.001, 0.0001\}$
RF	$n_f : \{d^{1/3}, d^{1/2}, d^{3/4}\}$ $n_l : \{1, 3, 5, 10\}$ $n_d : \{5, 7, 10\}$ $n_t : \{1000\}$
XGBoost	$l_r : \{0.01, 0.02, 0.03, 0.04, 0.05\}$ $n_d : \{3, 5, 10\}$ $m_l : \{0, 0.1, 0.2\}$ $n_b : \{0.6n, 0.8n, 1n\}$ $n_f : \{0.5d, 0.8d, 1d\}$
LSTM	l_r: $\{0.0001, 0.0005, 0.001, 0.005, 0.01\}$ $d_{r,0} : \{0.1, 0.15, \ldots, 0.5\}$ $d_{r,i} : \{0.1, 0.15, \ldots, 0.5\}$ $C : \{0.00001, 0.00005, 0.000001\}$ $n_i : \{16, 32, 64, 128, 256\}$ $h_l : \{1, 2, 3, 4\}$
TCN	l_r: $\{0.0001, 0.0005, 0.001, 0.005, 0.01\}$ $d_{r,0} : \{0.1, 0.15, \ldots, 0.5\}$ $C : \{0.00001, 0.00005, 0.000001\}$ $h_l : \{1, 2, 3, 4\}$ $n_i : \{16, 32, 64, 128, 256\}$ $k_{s,i} : \{3, 5, 7, 9, 11\}$

TABLE III: Hyperparameters and Hyperparameters search space for all algorithms tested in this work.

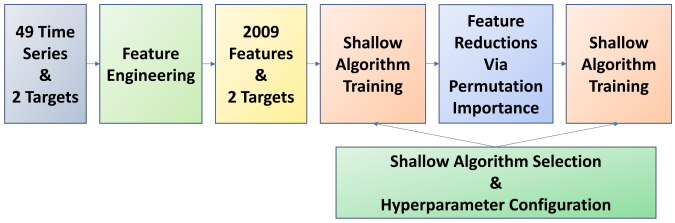


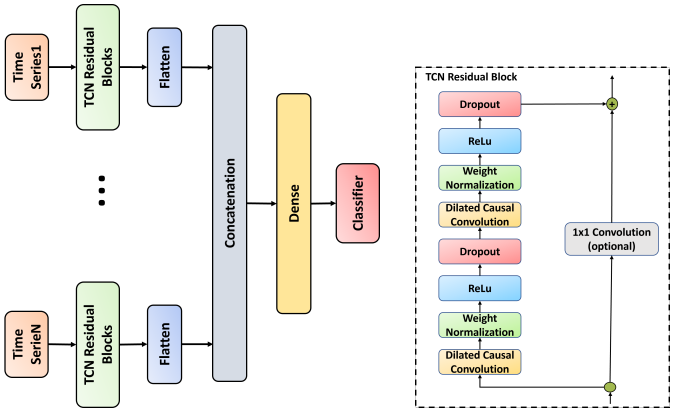
Fig. 3: Pipeline for the Shallow Models.

.

For the problem of tuning the hyperparameters and assessing the performance of the final model please refer to Section III-D.

C. Deep Models

For the scope of this work, shallow models have two main limitations. The first one is the dependency on handcrafted and experience-based features identified through a feature engineering step which may include too many irrelevant features or leave out important features. The second and most important one, is the loss of description of the different temporal behaviours intra and inter series. In other words, when we extract significant features, we are not able to fully capture different time scales. This constraint may produce a loss of information, since different time series may have a different temporal information response that we flatten with our feature map. As we will describe in the rest of this section, deep models allow us to overcome both limitations. Therefore, it is necessary first to rely on state-of-the-art architectures able to address temporal analysis such as the classical and the bidirectional Long-Short-Term Memory network (LSTM) [62]. These architectures can learn the features that can automatically address the problem and allow the capture of the two main temporal scales of the problem under investigation, i.e., long and short term. However, although these architectures can handle the first limitation of shallow models, they are not able to fully address the second one because by focusing mainly on long and short-term dependencies, they are not able to deal with multiple temporal scales.



(a) High level representation of the architecture.

(b) TCN-based layer architecture details.

Fig. 4: The proposed Deep Multi Scale Models architecture based on TCN.

As classical (LSTM) and bidirectional (BLSTM) LSTM networks, we rely on a standard architectures [29], [38], [39] where each input time series are fed to an LSTM layer that returns as output a vector with the same input dimension. The LSTM layer deals with extracting the representation vector that is fed directly to a dense layer which will produce the prediction. We trained the network using an ADAM optimiser [63] empowered with one-cycle learning rate [64] to improve convergence. These two architectures have a series of hyperparameters to be tuned: the learning rate l_r, the dropout rate $d_{r,0}$ on the last LSTM layer and the final dense fully connected layer, the number LSTM layers h_l, the dropout rate $d_{r,i}$ in each LSTM layer ($i \in \{1, \cdots, h_l\}$), the number of LSTM cells in each LSTM layer n_i ($i \in \{1, \cdots, h_l\}$), the L2 regularisation on the weight of the entire network C (see Table III). Note that in this case the hyperparameters configuration space is much larger with respect to shallow models.

Unfortunately, as we will discuss in the Section IV, these two architectures are not able to outperform the shallow models. The main reason behind this result is the limitations of the LSTM architectures able to handle only a very limited number of temporal scales.

To overcome LSTM limitations, we decided to substitute the LSTM blocks with Temporal Convolutional Network (TCN) residual blocks [40], [41] which are able to focus on multiple temporal scales for each raw time series independently. The proposed architecture is reported in Figure 4. The peculiarities of the proposed deep multiple temporal scale architecture based on TCN are mainly three: (i) the convolutions in the architecture are causal, namely there is not information leakage from future to past, (ii) the architecture can handle different sequence lengths and map it to an output sequence of the same length like the LSTMs, and (iii) is able to handle long effective history. For what concerns (i) the TCN uses causal convolutions. For what concerns (ii), it is due to the use of 1D fully-convolutional network model where each hidden layer has the same length of the input layer; zero padding of length $(\text{kernel size} - 1)$ is added to preserve the previous length.

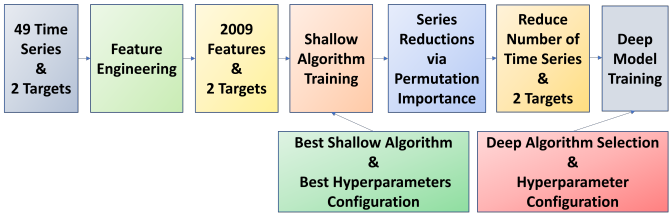


Fig. 5: Pipeline for the Deep Models.

.

As for the (iii) we employed dilated convolution that enables a large receptive field [65] without employing too deep TCN residual blocks. The network has been trained, like the LSTM-based architectures, with the ADAM optimiser empowered with one-cycle learning rate. This architecture has a series of hyperparameters that need to be carefully tuned: the learning rate l_r, the dropout rate $d_{r,0}$ on the last TCN layer and the final dense fully connected layer, the number of TCN blocks h_l for each time series, the number of filters in each block n_i ($i \in \{1, \cdots, h_l\}$), the kernel dimension $k_{s,i}$ for each series s and each block i, and the L2 regularisation on the weight of the entire network C (see Table III). Note that in this case the hyperparameters configuration space further explodes with respect to LSTM since basically we have, possibly, different configurations of the kernel dimension for each time series.

For this reason, in order to reduce this hyperparameters configuration space, and contemporary reduce the number of weight of the final network (both for LSTM and TCN based deep models), we decided to reduce the number of input series from the original 49. In fact, it is reasonable to assume that many of these series contain redundant information. In order to address this issue we rely on shallow models. In particular, similarly to the feature reduction phase implemented in shallow models, we implemented a time series reduction phase. Using again the permutation importance with mean decrease of accuracy as a metric, we permuted not the engineered features but the original time series discarding all the time series that non positively contribute according to the mean decrease of accuracy. Thanks to this reduction in the number of input time series, the LSTM and TCN based architectures strongly reduce the number of weights to tune and the hyperparameters configuration search space.

The pipeline we proposed for deep models is depicted in Figure 5.

For the problem of tuning the hyperparameters and assessing the performance of the final model please refer to Section III-D.

D. Extrapolating Scenarios

In our experiment, we will study three different extrapolating scenarios based on the intrinsic hierarchy of the dataset. This will allow us to understand the extrapolation ability and the robustness of the different models described in Sections III-B and III-C:

- Leave One Intention Out (LOIO): in this scenario the models have been trained with all data except the one referring to one launch intention for one ball weight of one group which has been kept apart for testing purposes;

- Leave One Ball Out (LOBO): in this scenario the models have been trained with all data except the one referring to a ball weight of one group which has been kept apart for testing purposes;
- Leave One Group Out (LOGO): in this scenario the models have been trained with all data except the one of one group which has been kept apart for testing purposes.

We report in Figure 6 a visual representation of these three scenarios. In particular, in the figure, we highlighted data hidden from the training phase and exploited just for testing purposes.

What remains to be addressed is how to tune the hyperparameters (architecture) of the shallow and deep models and how to assess the final performance.

For what concerns the last point the answer is easy. Based on the different scenarios (LOIO, LOBO, and LOGO) we have to split the data in Training $\mathcal{D}_n$ and Test $\mathcal{T}_t$ sets using the principle described above. Then we can use $\mathcal{D}_n$ to both train the model and select the best hyperparameters (architecture) of the shallow and deep models and used $\mathcal{T}_t$ to assess the performance of the final model using the metrics we defined in Section III-A. Repeating multiple times this procedure will give us the average performance in the different scenarios.

Instead, for tuning the hyperparameters (architecture) of the shallow and deep models we proceeded as follows. We took $\mathcal{D}_n$ and split it into Learning $\mathcal{L}_l$ and Validation $\mathcal{V}_v$ sets using LOIO, LOBO, or LOGO. Then for each model (shallow or deep) we train it with $\mathcal{L}_l$ with many different hyperparameters configurations and measure its performance on $\mathcal{V}_v$ with ACC. Then we repeated the experiment multiple times and selected the hyperparameters configuration which gives the best average ACC. Finally we retrained the model with the selected best configuration of the hyperparameters on the whole $\mathcal{D}_n$ which is the model that will be used for testing purposes (see previous paragraph). For shallow models (SVM, RF, and XGBoost) we performed a grid search (all possible hyperparameters configurations have been tested) [45]. For deep models (LSTM and TCN) a grid search was not computationally feasible so we performed a random search over 500 random configurations [66].

IV. Experimental Results

In this section, we report the results of applying the methodology presented in Section III to the data described in Section II. The hyperparameter selection and the performance assessment strategies (in the different extrapolating scenarios) are reported in the previous section while the complete list of hyperparameter configuration for all tested algorithms is reported in Table III.

As observed in Section II, we want to address two different scopes, namely the automatic detection of the ball weight or the launch intention in a ball exchange scenario.

Let us start by presenting the results obtained when the target is the ball weight. Tables IV and V report the recognition performance (measured with different metrics, i.e., ACC, PRE, REC, and ROC-AUC) in all the proposed scenarios (LOIO, LOBO, and LOGO) for the different 13 groups together with the average over the groups of the different learning algorithms (LSVM, KSVM, RF, XGBoost, LSTM, BLSTM, and TCN).

Alg. / Group	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
1	80.7±2.9	65.5±7.7	61.2±3.1	61.3±2.0	68.7±12.2	61.9±8.5	**94.1±6.6**
2	**94.7±0.9**	76.9±12.6	62.7±3.4	67.3±1.8	68.1±12.3	59.1±8.9	89.0±4.8
3	92.7±1.8	80.4±10.6	82.2±2.5	88.1±1.1	66.4±11.7	63.4±9.3	**97.0±4.4**
4	**93.1±0.7**	75.9±12.3	86.8±1.9	91.8±2.4	72.2±11.9	68.7±12.9	91.6±4.2
5	79.4±3.4	66.6±9.4	61.9±2.0	61.9±2.8	63.5±12.1	60.8±10.8	**93.0±2.6**
6	93.6±0.5	79.4±12.4	91.5±0.9	92.3±0.8	76.4±12.9	73.1±13.5	**99.8±4.2**
7	**91.8±2.6**	78.8±11.5	90.6±1.6	91.6±1.5	70.1±10.7	66.1±11.9	**91.8±4.4**
8	89.9±1.0	76.3±10.7	76.3±1.8	79.2±2.4	74.3±14.4	62.1±8.9	**96.9±5.2**
9	94.0±1.2	80.3±13.4	93.5±1.7	**94.3±3.2**	69.5±12.8	66.6±11.8	92.7±2.5
10	**93.5±0.6**	81.5±12.1	87.2±2.7	85.2±0.6	74.9±13.3	62.0±8.7	88.9±2.9
11	**88.6±2.0**	73.0±14.8	69.2±3.2	74.5±2.0	67.6±12.7	67.6±8.6	84.9±6.7
12	**92.6±1.0**	79.6±10.6	84.5±3.3	91.1±0.7	73.2±14.9	71.4±13.5	90.8±4.9
13	87.6±1.4	75.5±10.3	83.1±2.3	80.7±1.1	73.0±11.8	68.6±12.7	**94.5±3.8**
Avg.	90.2±1.5	76.1±11.4	79.3±2.3	81.5±1.7	70.6±12.6	65.5±10.8	**92.7±4.4**

(a) LOIO

Alg. / Group	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
1	83.9±3.6	55.1±14.6	63.3±2.7	64.0±2.4	63.4±12.8	61.2±8.6	**86.3±14.1**
2	**95.7±1.7**	81.0±17.6	54.3±4.3	62.0±3.5	61.7±8.6	59.9±9.8	82.5±10.7
3	**91.9±2.3**	73.7±20.0	77.6±4.2	85.8±1.3	72.4±14.5	62.1±11.2	89.4±11.2
4	92.7±0.0	77.3±20.8	85.8±2.2	90.5±1.8	69.5±12.7	69.9±13.4	**93.1±4.1**
5	80.4±4.1	63.2±15.6	57.3±2.3	56.4±3.0	76.5±14.0	72.5±14.5	**93.3±3.9**
6	91.6±0.6	79.5±19.8	90.6±1.0	90.3±1.3	57.6±7.2	64.9±9.7	**95.0±7.9**
7	92.1±2.3	79.0±21.4	91.4±1.3	**93.3±1.5**	75.3±7.1	69.5±12.7	93.2±4.2
8	88.5±0.7	74.0±19.6	70.3±1.9	73.8±1.3	65.3±12.5	58.7±7.1	**96.2±4.8**
9	93.3±1.4	83.8±19.3	91.0±3.3	91.8±1.7	63.2±9.7	67.8±12.0	**95.1±2.6**
10	**90.5±0.7**	83.5±17.1	83.7±2.3	84.8±0.9	65.7±11.8	58.2±7.8	83.7±13.9
11	**87.5±2.1**	81.1±14.2	66.4±3.5	72.7±2.1	86.0±7.3	66.7±13.9	86.6±7.8
12	91.3±1.7	76.4±18.9	83.8±3.1	**91.4±3.0**	72.5±13.0	55.2±14.4	88.9± 9.8
13	**86.7±1.5**	70.4±20.6	80.2±2.6	79.6±2.0	73.9±13.1	74.7±13.1	85.4±14.1
Avg.	89.7±1.7	75.2±18.4	76.6±2.7	79.7±2.0	69.5±11.1	64.7±10.6	**89.9±8.4**

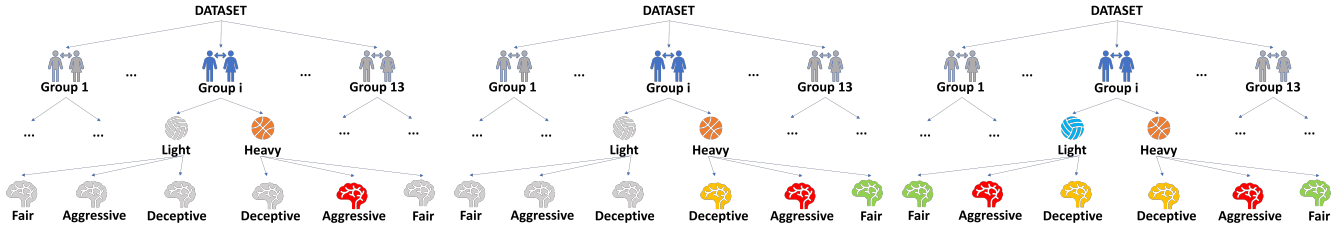
(b) LOBO

Alg. / Group	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
1	38.3±0.0	49.5±4.0	64.1±2.8	67.6±1.9	59.2±6.6	54.7±4.3	**84.6±9.1**
2	58.7±1.1	60.1±2.2	54.2±4.7	57.0±3.9	55.7±4.0	51.6±1.3	**80.3±12.0**
3	56.0±0.0	62.5±4.0	74.7±5.2	77.9±2.6	59.8±6.1	56.0±4.1	**88.7±7.4**
4	88.2±0.0	82.2±0.3	85.8±2.5	**90.3±1.4**	73.2±4.3	51.9±1.8	84.2±12.4
5	65.0±1.1	61.4±1.3	59.4±2.8	54.8±2.4	58.7±3.0	52.9±2.9	**87.6±3.9**
6	90.0±0.0	91.5±2.4	90.6±1.4	91.2±1.3	73.2±5.3	54.5±2.0	**94.8±2.5**
7	88.8±0.0	90.9±2.4	92.5±1.5	**92.9±2.2**	71.1±2.7	58.5±3.5	92.7±2.5
8	74.1±2.2	75.7±2.2	75.2±1.8	77.3±1.6	61.8±6.5	55.3±3.3	**93.2±4.9**
9	88.2±3.0	85.4±3.9	91.7±3.6	**92.2±1.2**	59.9±7.2	54.0±2.6	81.4±8.1
10	81.6±0.2	88.4±1.3	86.0±1.6	85.1±1.4	62.4±3.2	61.7±6.5	**89.4±3.7**
11	73.0±0.6	70.2±3.8	69.7±3.8	74.4±1.5	59.8±5.1	55.5±3.5	**91.6±2.7**
12	79.9±0.5	83.4±2.6	84.3±3.2	**91.4±1.6**	77.1±3.8	53.6±2.9	85.7±6.6
13	75.4±0.0	78.4±3.1	81.5±2.1	78.8±2.4	65.8±5.4	57.4±5.2	**83.5±9.5**
Avg.	73.6±0.7	75.3±2.6	77.7±2.9	79.3±2.0	64.4±4.9	55.2±3.4	**87.5±6.6**

(c) LOGO

TABLE IV: Predicting the Ball Weight: ACC of the different algorithms (LSVM, KSVM, RF, XGBoost, LSTM, BLSTM, and TCN) for each one of the 13 groups together with the average across the groups.

Observing Tables IV and V we can easily see how TCN is able to outperform all the other algorithms in all the scenarios no matter the considered metric. Note that only the PRE of TCN in the LOGO scenario is slightly lower than that of LSVM. LSVM is the only algorithm able to obtain results close to TCN's in two of the three proposed extrapolation scenarios, i.e., the simplest one (LOIO and LOBO). Note that,



(a) LOIO (b) LOBO (c) LOGO

Fig. 6: Visual representation of the three extrapolating scenarios. In particular we highlighted data hidden from the training phase and exploited just for testing purposes.

Alg. / Metric	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
ACC	90.2	76.1	79.3	81.5	70.6	65.5	**92.7**
PRE	89.7	71.8	78.8	81.1	62.9	61.4	**92.0**
REC	90.9	90.5	84.5	85.7	73.3	67.1	**94.8**
ROC-AUC	0.96	0.85	0.89	0.89	0.77	0.72	**0.99**

(a) LOIO

Alg. / Metric	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
ACC	89.7	75.2	76.6	79.7	69.5	64.7	**89.9**
PRE	**89.4**	71.9	76.5	79.7	62.4	62.5	89.3
REC	90.6	87.2	81.6	83.4	72.8	66.7	**93.6**
ROC-AUC	0.95	0.91	0.86	0.88	0.75	0.69	**0.97**

(b) LOBO

Alg. / Metric	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
ACC	73.6	75.3	77.7	79.3	64.4	55.2	**87.5**
PRE	75.1	76.5	77.7	79.6	62.1	54.6	**86.4**
REC	75.9	78.6	82.7	83.5	66.7	56.6	**89.2**
ROC-AUC	0.82	0.85	0.87	0.87	0.73	0.65	**0.93**

(c) LOGO

TABLE V: Predicting the Ball Weight: ACC, PRE, REC and ROC-AUC of the different algorithms (LSVM, KSVM, RF, XGBoost, LSTM, BLSTM, and TCN) averaged over the 13 groups.

the shallow models (LSVM, KSVM, RF, and XGBoost) are often competitive with (or better than) the classical deep one (LSTM and BLSTM) while TCN is always the top performing methods.

The same behaviour can be observed in Tables VI and VII, which are the counterpart of Tables IV and V when the target is the launch intention.

In conclusion, based on the experimental results, we can make a series of observations. TCN is able to outperform all other algorithms no matter the target (ball weight or launch intention), scenarios (LOIO, LOBO, and LOGO), and metric exploited. Shallow models generally outperform classical deep models but they fail in achieving the performance of TCN. In some simple extrapolation scenarios, LSVM is able to compete with TCN. This confirms what we discussed in the paper, namely it is not simple with deep models to outperform well calibrated shallow models but when deep models are empowered with the knowledge that can be extracted by shallow models they can reach even remarkable improvements in terms of recognition performance in very complex extrapolating scenarios.

V. Conclusions

In this work we argued that in order to study human motion it is required to properly model multiple temporal scales that fully describe its complexity. In fact, human movement involves different muscles that are activated and coordinated by the brain at different temporal scales in a complex cognitive process. In this context, data-driven models represent research frontiers able to provide new insights but current approaches are not able to properly address the necessity of modelling so many time scales.

For this reason, in this work we investigated different data-driven approaches. The first one is based on shallow models that, while achieving reasonably good recognition performance, require to handcraft features according to the domain knowledge. The second one is based on deep models that can be extended to manage multiple temporal scales but they are hard to exploit as too many architecture configurations exist. For this reason, we will propose a new deep multiple temporal scale data-driven model, based on Temporal Convolutional Network, capable of learning features from the data at different temporal scales, of outperforming state of the art deep and shallow models, and of exploiting shallow models to tune the architecture configuration. Then, we designed, collected data, and tested our proposal in a specially devised experiment, to prove the validity of our approach. In particular, we collected motion capture data about dyad actions where two people exchange a ball. As the weight of the ball and the throwing intentions change, we showed how it is possible to automatically detect either the weight of the ball or the intention behind the throw just based on motion data. Results support both the proposal and the need for the use of deep multi scale models as a tool to better understand human movement and its multiple temporal scale nature.

Data regarding our experiment, pipelines, and code of the methods proposed in this work are also made freely available to the research community.

Acknowledgements

This research has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 824160 (FET PROACTIVE EnTimeMent Project).

Alg. / Group	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
1	**94.1±1.7**	90.3±5.5	73.7±4.2	70.7±1.0	63.3±12.9	58.5±6.8	92.9±3.7
2	86.0±1.0	81.9±8.8	82.2±3.1	82.1±2.0	60.1±7.4	60.1±7.5	**90.0±2.6**
3	84.3±0.8	82.5±8.1	74.8±3.7	71.4±2.7	67.8±13.9	75.0±13.6	**90.8±3.6**
4	79.2±1.0	76.4±6.7	82.8±2.6	86.4±1.5	65.5±10.0	70.9±13.4	**90.5±3.0**
5	82.2±1.0	72.3±9.4	72.5±3.5	77.3±3.9	75.7±13.9	71.5±14.0	**90.5±1.9**
6	**95.8±0.0**	86.2±12.4	**95.8±1.4**	95.0±1.2	63.3±11.3	66.4±11.1	92.6±3.3
7	90.3±0.6	80.4±8.4	77.3±2.3	76.2±2.3	71.2±13.0	68.2±12.4	**93.5±3.4**
8	75.9±0.7	70.5±9.4	73.6±1.5	76.9±3.3	77.8±14.2	75.4±15.2	**92.6±3.9**
9	74.3±1.4	72.9±5.4	81.7±2.8	84.9±1.2	74.6±11.9	68.5±11.4	**91.7±3.3**
10	**96.2±1.7**	86.1±7.8	89.7±1.9	93.3±2.1	74.4±10.6	74.4±11.1	93.6±3.3
11	88.9±1.1	88.2±9.6	92.0±2.8	89.8±1.7	61.8± 9.9	63.8±10.0	**92.3±3.8**
12	90.7±0.8	77.3±11.5	80.9±3.5	81.6±1.9	67.6±11.2	64.9±9.5	**93.7±3.6**
13	91.4±1.2	83.8±17.8	87.5±3.5	88.1±2.4	69.5±9.7	65.5±12.2	**91.7±4.0**
Avg.	86.9±1.0	80.7±8.5	81.9±2.8	82.6±2.1	68.7±11.5	67.9±11.4	**92.0±3.3**

(a) LOIO

Alg. / Group	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
1	84.8±1.7	**88.2±2.5**	73.6±3.8	72.1±1.2	62.7±9.5	63.3±10.6	85.1±4.5
2	83.3±2.9	85.9±4.2	83.5±2.8	81.6±3.2	62.9±10.4	58.8±8.2	**89.6±5.0**
3	82.3±0.5	83.0±2.0	75.3±3.8	73.1±1.8	67.9±13.6	67.9±11.2	**90.0±6.4**
4	77.2±0.7	78.8±1.3	83.5±2.9	87.7±1.6	68.2±14.2	68.8±14.0	**90.9±5.7**
5	80.6±0.6	80.8±2.1	72.6±2.9	76.8±3.0	69.0±13.3	70.0±13.6	**86.1±3.8**
6	95.1±0.3	94.5±0.9	**96.0±1.4**	**96.0±0.9**	73.3±14.5	76.2±15.5	92.4±4.9
7	88.5±1.0	87.8±0.9	77.4±2.3	77.3±1.1	82.6±9.6	84.0±10.1	**90.4±5.3**
8	78.7±0.4	78.9±0.9	74.7±2.3	75.7±1.5	62.3±9.6	59.7±8.4	**92.0±5.9**
9	72.7±0.0	73.0±3.6	79.8±2.8	82.8±2.8	72.5±14.0	65.4±12.2	**88.1±7.6**
10	**95.6±1.1**	92.2±2.7	90.7±1.7	92.4±1.6	74.5±12.6	72.8±14.2	88.4±6.0
11	86.3±2.4	93.1±2.7	90.8±2.2	89.6±2.7	68.8±15.3	67.3±12.7	**95.3±5.4**
12	**90.6±0.8**	87.4±2.9	83.7±2.5	84.6±2.3	70.1±11.7	61.4±8.4	89.3±6.4
13	91.5±0.0	90.1±0.9	87.7±2.9	87.0±2.2	73.8±12.8	77.0±13.9	**93.1±3.4**
Avg.	85.2±1.0	85.7±2.1	82.2±2.6	82.8±2.0	69.9±12.4	68.7±11.8	**90.1±5.4**

(b) LOBO

Alg. / Group	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
1	79.5±1.8	82.8±4.9	67.4±4.6	65.8±2.1	54.5±2.7	54.6±3.0	**85.7±6.3**
2	82.2±2.6	**90.0±4.1**	81.9±3.1	83.5±3.9	53.4±2.6	53.1±2.5	80.4±9.3
3	80.0±0.4	81.2±1.1	71.2±4.4	67.3±2.5	56.9±3.0	56.2±3.5	**86.1±2.2**
4	73.7±1.1	75.9±0.7	82.0±2.6	**87.5±1.7**	75.7±4.9	75.7±4.6	83.5±5.4
5	79.2±0.0	79.2±0.0	71.5±3.9	74.3±2.9	58.3±4.2	57.3±5.9	**89.4±4.2**
6	94.4±0.0	93.4±1.1	**94.6±2.7**	94.2±1.2	78.4±7.0	76.1±5.1	91.2±5.5
7	**87.8±1.1**	87.3±2.2	75.0±2.5	76.3±2.8	75.0±4.1	74.7±5.2	84.7±5.7
8	76.7±1.6	75.8±0.6	73.1±2.3	74.8±1.1	71.7±5.0	72.1±6.8	**88.6±4.9**
9	68.0±2.3	72.0±3.0	81.6±3.3	**85.0±1.6**	55.0±1.4	54.7±1.2	81.8±5.2
10	90.2±0.3	91.3±1.4	89.2±2.6	**93.3±0.7**	61.6±7.7	62.2±8.3	89.4±5.4
11	84.1±0.0	**95.2±3.3**	90.7±3.2	90.7±1.7	61.3±6.2	61.7±5.4	74.8±6.9
12	**90.6±0.0**	85.0±2.5	81.5±2.2	84.6±1.9	82.5±2.3	82.1±2.8	90.1±4.0
13	**90.5±0.0**	89.1±0.6	89.8±2.6	89.2±3.6	70.2±2.4	69.8±2.9	89.0±4.4
Avg.	82.8±0.8	84.5±2.0	80.7±3.1	82.0±2.1	65.7±4.1	65.4±4.4	**85.7±5.3**

(c) LOGO

TABLE VI: Predicting the Launch Intention: ACC of the different algorithms (LSVM, KSVM, RF, XGBoost, LSTM, BLSTM, and TCN) for each one of the 13 groups together with the average across the groups.

Alg. / Metric	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
ACC	86.9	80.7	81.9	82.6	68.7	67.9	**92.0**
PRE	86.9	82.6	82.5	83.1	69.3	68.5	**92.6**
REC	86.8	80.6	82.0	82.9	68.9	67.9	**92.0**
ROC-AUC	0.96	0.95	0.93	0.94	0.79	0.77	**0.99**

(a) LOIO

Alg. / Metric	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
ACC	85.2	85.7	82.2	82.8	69.9	68.7	**90.1**
PRE	85.4	86.0	82.7	83.1	70.5	69.9	**91.3**
REC	85.3	85.8	82.3	83.0	68.9	69.5	**89.2**
ROC-AUC	0.95	0.95	0.93	0.94	0.80	0.78	**0.97**

(b) LOBO

Alg. / Metric	LSVM	KSVM	RF	XGBoost	LSTM	BLSTM	TCN
ACC	82.8	84.5	80.7	82.0	65.7	65.4	**85.7**
PRE	82.6	84.3	80.9	81.9	65.9	65.7	**86.4**
REC	82.4	83.7	80.3	81.6	65.3	65.3	**85.5**
ROC-AUC	0.94	0.94	0.92	0.93	0.76	0.76	0.94

(c) LOGO

TABLE VII: Predicting the Launch Intention: ACC, PRE, REC and ROC-AUC of the different algorithms (LSVM, KSVM, RF, XGBoost, LSTM, BLSTM, and TCN) averaged over the 13 groups.

REFERENCES

[1] A. Vinciarelli, M. Pantic, D. Heylen, C. Pelachaud, I. Poggi, F. D'Errico, and M. Schroeder, "Bridging the gap between social animal and unsocial machine: A survey of social signal processing," *IEEE Transactions on Affective Computing*, vol. 3, pp. 69–87, 2011.

[2] A. Vinciarelli and A. S. Pentland, "New social signals in a new interaction world: the next frontier for social signal processing," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 1, pp. 10–17, 2015.

[3] M. Mehu and K. R. Scherer, "A psycho-ethological approach to social signal processing," *Cognitive processing*, vol. 13, pp. 397–414, 2012.

[4] I. Poggi and F. D'Errico, "Social signals: a framework in terms of goals and beliefs," *Cognitive Processing*, vol. 13, pp. 427–445, 2012.

[5] A. Vinciarelli, M. Pantic, and H. Bourlard, "Social signal processing: Survey of an emerging domain," *Image and vision computing*, vol. 27, pp. 1743–1759, 2009.

[6] N. Ambady, F. J. Bernieri, and J. A. Richeson, "Toward a histology of social behavior: Judgmental accuracy from thin slices of the behavioral stream," in *Advances in experimental social psychology*, 2000.

[7] H. Lausberg and H. Sloetjes, "The revised neuroges-elan system: An objective and reliable interdisciplinary analysis tool for nonverbal behavior and gesture," *Behavior research methods*, vol. 48, pp. 973–993, 2016.

[8] W. S. Condon and W. D. Ogston, "A segmentation of behavior." *Journal of psychiatric research*, 1967.

[9] D. S. Wickramasuriya, M. K. Tessmer, and R. T. Faghih, "Facial expression-based emotion classification using electrocardiogram and respiration signals," in *IEEE Healthcare Innovations and Point of Care Technologies*, 2019.

[10] L. Fan, W. Wang, S. Huang, X. Tang, and S. C. Zhu, "Understanding human gaze communication by spatio-temporal graph reasoning," in *IEEE/CVF International Conference on Computer Vision*, 2019.

[11] M. W. Eysenck and M. T. Keane, *Cognitive psychology: A student's handbook*. Psychology press, 2020.

[12] C. M. Funke, J. Borowski, K. Stosio, W. Brendel, T. S. Wallis, and M. Bethge, "Five points to check when comparing visual perception in humans and machines," *Journal of Vision*, vol. 21, pp. 16–16, 2021.

[13] B. Caramiaux, M. Donnarumma, and A. Tanaka, "Understanding gesture expressivity through muscle sensing," *ACM Transactions on Computer-Human Interaction (TOCHI)*, vol. 21, pp. 1–26, 2015.

[14] M. Wijnants, R. Cox, F. Hasselman, A. Bosman, and G. Van Orden, "A trade-off study revealing nested timescales of constraint," *Frontiers in physiology*, vol. 3, p. 116, 2012.

[15] J. Butepage, M. J. Black, D. Kragic, and H. Kjellstrom, "Deep representation learning for human motion prediction and classification," in *IEEE conference on computer vision and pattern recognition*, 2017.

[16] F. Zhou, F. De la Torre, and J. K. Hodgins, "Hierarchical aligned cluster analysis for temporal clustering of human motion," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 3, pp. 582–596, 2012.

[17] R. L. Goldstone, J. R. de Leeuw, and D. H. Landy, "Fitting perception in and to cognition," *Cognition*, vol. 135, pp. 24–29, 2015.

[18] G. Johansson, "Visual perception of biological motion and a model for its analysis," *Perception & psychophysics*, vol. 14, no. 2, pp. 201–211, 1973.

[19] F. E. Pollick, H. M. Paterson, A. Bruderlin, and A. J. Sanford, "Perceiving affect from arm movement," *Cognition*, vol. 82, no. 2, pp. B51–B61, 2001.

[20] H. K. M. Meeren, N. Hadjikhani, S. P. Ahlfors, M. S. Hämäläinen, and B. De Gelder, "Early preferential responses to fear stimuli in human right dorsal visual stream-a meg study," *Scientific reports*, vol. 6, p. 24831, 2016.

[21] E. A. F. Ihlen and B. Vereijken, "Interaction-dominant dynamics in human cognition: Beyond $1/f\alpha$ fluctuation." *Journal of Experimental Psychology: General*, vol. 139, no. 3, p. 436, 2010.

[22] A. Camurri, G. Volpe, S. Piana, M. Mancini, R. Niewiadomski, N. Ferrari, and C. Canepa, "The dancer in the eye: towards a multi-layered computational framework of qualities in movement," in *International Symposium on Movement and Computing*, 2016.

[23] J. Newlove and J. Dalby, *Laban for all*. Routledge, 2019.

[24] C. Beyan, S. Karumuri, G. Volpe, A. Camurri, and R. Niewiadomski, "Modeling multiple temporal scales of full-body movements for emotion classification," *IEEE Transactions on Affective Computing*, 2021.

[25] G. Yao, T. Lei, J. Zhong, and P. Jiang, "Learning multi-temporal-scale deep information for action recognition," *Applied Intelligence*, vol. 49, pp. 2017–2029, 2019.

[26] A. Stergiou and R. Poppe, "Multi-temporal convolutions for human action recognition in videos," in *International Joint Conference on Neural Networks*, 2021.

[27] B. Lin, S. Zhang, Y. Liu, and S. Qin, "Multi-scale temporal information extractor for gait recognition," in *IEEE International Conference on Image Processing*, 2021.

[28] S. Järvelä, D. Gašević, T. Seppänen, M. Pechenizkiy, and P. A. Kirschner, "Bridging learning sciences, machine learning and affective computing for understanding cognition and affect in collaborative learning," *British Journal of Educational Technology*, vol. 51, pp. 2391–2406, 2020.

[29] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.

[30] P. J. Bota, C. Wang, A. L. Fred, and H. P. Da Silva, "A review, current challenges, and future possibilities on emotion recognition using machine learning and physiological signals," *IEEE Access*, vol. 7, pp. 140 990–141 020, 2019.

[31] B. Kratzwald, S. Ilić, M. Kraus, S. Feuerriegel, and H. Prendinger, "Deep learning for affective computing: Text-based emotion recognition in decision support," *Decision Support Systems*, vol. 115, pp. 24–35, 2018.

[32] D. Cireşan, U. Meier, J. Masci, and J. Schmidhuber, "A committee of neural networks for traffic sign classification," in *International joint conference on neural networks*, 2011.

[33] A. Hekler, J. S. Utikal, A. H. Enk, W. Solass, and Others, "Deep learning outperformed 11 pathologists in the classification of histopathological melanoma images," *European Journal of Cancer*, vol. 118, pp. 91–96, 2019.

[34] D. Silver, J. Schrittwieser, K. Simonyan *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, pp. 354–359, 2017.

[35] J. Jumper, R. Evans, A. Pritzel, T. Green, and Others, "Highly accurate protein structure prediction with alphafold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[36] K. Grace, J. Salvatier, A. Dafoe, B. Zhang, and O. Evans, "Viewpoint: When will AI exceed human performance? evidence from AI experts," *Journal of Artificial Intelligence Research*, vol. 62, pp. 729–754, 2018.

[37] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory To Algorithms*. Cambridge University Press, 2014.

[38] D. Liciotti, M. Bernardini, L. Romeo, and E. Frontoni, "A sequential deep learning application for recognising human activities in smart homes," *Neurocomputing*, vol. 396, pp. 501–513, 2020.

[39] D. Jirak, S. Tietz, H. Ali, and S. Wermter, "Echo state networks and long short-term memory for continuous gesture recognition: A comparative study," *Cognitive Computation*, 2020.

[40] S. M. Lee, S. M. Yoon, and H. Cho, "Human activity recognition from accelerometer data using convolutional neural network," in *IEEE international conference on big data and smart computing*, 2017.

[41] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803. 01271*, 2018.

[42] L. C. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A. L. Yuille, "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs," *IEEE transactions on pattern analysis and machine intelligence*, vol. 40, pp. 834–848, 2017.

[43] X. Dai, B. Singh, J. Y. H. Ng, and L. Davis, "Tan: Temporal aggregation network for dense multi-label action recognition," in *IEEE Winter Conference on Applications of Computer Vision*, 2019.

[44] M. Sokolova and G. Lapalme, "A systematic analysis of performance measures for classification tasks," *Information processing & management*, vol. 45, no. 4, pp. 427–437, 2009.

[45] L. Oneto, *Model Selection and Error Estimation in a Nutshell*. Springer, 2019.

[46] T. Elsken, J. H. Metzen, and F. Hutter, "Neural architecture search: A survey," *The Journal of Machine Learning Research*, vol. 20, no. 1, pp. 1997–2017, 2019.

[47] A. Zheng and A. Casari, *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc., 2018.

[48] P. Duboue, *The Art of Feature Engineering: Essentials for Machine Learning*. Cambridge University Press, 2020.

[49] J. Shawe-Taylor and N. Cristianini, *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[50] L. Breiman, "Random forests," *Machine Learning*, vol. 45, pp. 5–32, 2001.

[51] T. Chen and C. Guestrin, "Xgboost: A scalable tree boosting system," in *SCM SIGKDD international conference on knowledge discovery and data mining*, 2016.

[52] G. Dong and J. Pei, *Sequence data mining*. Springer Science & Business Media, 2007.

[53] J. L. Reyes-Ortiz, L. Oneto, A. Sama, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[54] N. D. Cilia, C. De Stefano, F. Fontanella, and A. Scotto, "A ranking-based feature selection approach for handwritten character recognition," *Pattern Recognition Letters*, vol. 121, pp. 77–86, 2019.

[55] V. D'Amato, E. Volta, L. Oneto, G. Volpe, A. Camurri, and D. Anguita, "Understanding violin players' skill level based on motion capture: a data-driven perspective," *Cognitive Computation*, vol. 12, no. 6, pp. 1356–1369, 2020.

[56] A. Roy, B. Banerjee, A. Hussain, and S. Poria, "Discriminative dictionary design for action classification in still images and videos," *Cognitive Computation*, vol. 13, pp. 698–708, 2021.

[57] M. Fernández-Delgado, E. Cernadas, S. Barro, and D. Amorim, "Do we need hundreds of classifiers to solve real world classification problems?" *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 3133–3181, 2014.

[58] M. Wainberg, B. Alipanahi, and B. J. Frey, "Are random forests truly the best classifiers?" *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 3837–3841, 2016.

[59] L. Van Der Maaten, E. Postma, and J. Van den Herik, "Dimensionality reduction: a comparative," *Journal of Machine Learning Research*, vol. 10, no. 66-71, p. 13, 2009.

[60] A. Fisher, C. Rudin, and F. Dominici, "All models are wrong, but many are useful: Learning a variable's importance by studying an entire class of prediction models simultaneously." *Journal of Machine Learning Research*, vol. 20, no. 177, pp. 1–81, 2019.

[61] C. Molnar, *Interpretable machine learning*. Lulu.com, 2020.

[62] W. Zhu, C. Lan, J. Xing, W. Zeng, Y. Li, L. Shen, and X. Xie, "Co-occurrence feature learning for skeleton based action recognition using regularized deep lstm networks," in *AAAI conference on artificial intelligence*, 2016.

[63] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412. 6980*, 2014.

[64] M. A. Hannan, D. N. T. How, M. B. Mansor, M. S. H. Lipu, P. J. Ker, and K. M. Muttaqi, "State-of-charge estimation of li-ion battery using gated recurrent unit with one-cycle learning rate policy," *IEEE Transactions on Industry Applications*, vol. 57, pp. 2964–2971, 2021.

[65] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *arXiv preprint arXiv:1511. 07122*, 2015.

[66] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization." *Journal of machine learning research*, vol. 13, no. 2, 2012.