Dipartimento di Informatica, Bioingegneria,
Robotica ed Ingegneria dei Sistemi

Deep Multi Temporal Scale Networks for Human Motion Analysis

by

Vincenzo Stefano D'Amato

Theses Series **DIBRIS-TH-2022-XX**

DIBRIS, Università di Genova
Via Opera Pia, 13 16145 Genova, Italy http://www.dibris.unige.it/

Università degli Studi di Genova

Dipartimento di Informatica, Bioingegneria,

Robotica ed Ingegneria dei Sistemi

Ph.D. Thesis in Computer Science and Systems Engineering
Systems Engineering Curriculum

Deep Multi Temporal Scale Networks for Human Motion Analysis

by

Vincenzo Stefano D'Amato

December, 2022

Dottorato di Ricerca in Informatica ed Ingegneria dei Sistemi
Indirizzo Ingegneria dei Sistemi
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università degli Studi di Genova

DIBRIS, Univ. di Genova
Via Opera Pia, 13
I-16145 Genova, Italy
http://www.dibris.unige.it/

Ph.D. Thesis in Computer Science and Systems Engineering
Systems Engineering Curriculum
(S.S.D. INF/01)

Submitted by Vincenzo Stefano D'Amato
DIBRIS, Univ. di Genova

. . . .

Date of submission: October, 2022

Title: Deep Multi Temporal Scale Networks for Human Motion Analysis

Advisor: Antonio Camurri, Luca Oneto
Dipartimento di Informatica, Bioingegneria, Robotica ed Ingegneria dei Sistemi
Università di Genova

. . .

Ext. Reviewers:

Abstract

The movement of human beings appears to respond to a complex motor system that contains signals at different hierarchical levels. For example, an action such as "grasping a glass on a table" represents a high-level action, but to perform this task, the body needs several motor inputs that include the activation of different joints of the body (shoulder, arm, hand, fingers, etc.). Each of these different joints/muscles have a different size, responsiveness, and precision with a complex non-linearly stratified temporal dimension where every muscle has its temporal scale. Parts such as the fingers responds much faster to brain input than more voluminous body parts such as the shoulder. The cooperation we have when we perform an action produces smooth, effective, and expressive movement in a complex multiple temporal scale cognitive task. Following this layered structure, the human body can be described as a kinematic tree, consisting of joints connected. Although it is nowadays well known that human movement and its perception are characterised by multiple temporal scales, very few works in the literature are focused on studying this particular property.

In this thesis, we will focus on the analysis of human movement using data-driven techniques. In particular, we will focus on the non-verbal aspects of human movement, with an emphasis on full-body movements. The data-driven methods can interpret the information in the data by searching for rules, associations or patterns that can represent the relationships between input (e.g. the human action acquired with sensors) and output (e.g. the type of action performed). Furthermore, these models may represent a new research frontier as they can analyse large masses of data and focus on aspects that even an expert user might miss. The literature on data-driven models proposes two families of methods that can process time series and human movement. The first family, called shallow models, extract features from the time series that can help the learning algorithm find associations in the data. These features are identified and designed by domain experts who can identify the best ones for the problem faced. On the other hand, the second family avoids this phase of extraction by the human expert since the models themselves can identify the best set of features to optimise the learning of the model.

In this thesis, we will provide a method that can apply the multi-temporal scales property of the human motion domain to deep learning models, the only data-driven models that can be extended to handle this property. We will ask ourselves two questions: what happens if we apply knowledge about how human movements are performed to deep learning models? Can this knowledge improve current automatic recognition standards?

In order to prove the validity of our study, we collected data and tested our hypothesis in specially designed experiments. 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.

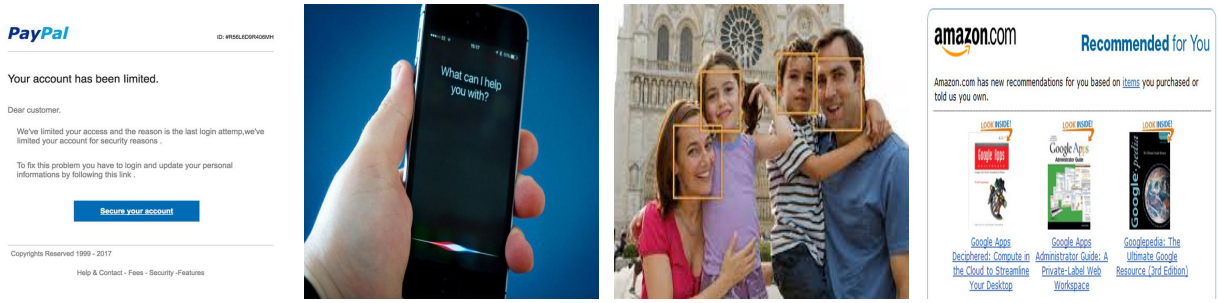
Table of Contents

Chapter 1

Introduction

1.1 Overview

The term Machine Learning (ML) refers to the automated detection of meaningful patterns in data [SSBD14]. In recent years, it has become a common tool able to be applied in several contexts. Indeed, we are surrounded by ML-based technology: anti-spam software learns to filter our email messages [FAO+21], credit card transactions are secured by software that learns how to detect frauds [AAO17], and large online retailers can automatically suggest products to us based on our past purchases [LJKP21]. Moreover, also recently smartphones learn to recognize voice commands and digital cameras learn to automatically detect faces when we take a photo. ML is also widely used in scientific applications such as bioinformatics, medicine, and astronomy. One reason for ML's success is the complexity of the patterns that need to be detected, improving the human limits that cannot provide an explicit, fine-detailed specification of how such tasks should be executed. Similarly to human beings, where many skills are acquired or refined through learning from experience, ML algorithms are concerned with endowing programs with the ability to learn and adapt their behaviour thanks to the input available to them. The input to a learning algorithm is represented by the training data, which represents experience, and the output is some expertise. More in detail, we can identify two main scenarios where ML can be applied: tasks that are too complex for human knowledge and tasks that require adaptability. For instance, there are numerous tasks that we human beings perform every day; yet our introspection concerning how we do them is not sufficiently elaborate to extract a well-defined program. Examples of such tasks include image understanding, speech recognition, and driving. The application of ML algorithms to such tasks achieves quite satisfactory results, once exposed to sufficiently many training examples. Another broad family of tasks that benefit from ML techniques are related to the analysis of very large and complex datasets: electronic commerce, weather prediction, analysis of genetic data, turning medical archives into medical knowledge, and the study of human actions. With the increasing availability of data, it becomes challenging for humans to



(a) name1. (b) name2. (c) name3. (d) name4.

Figure 1.1: Example of tangible ML applications in real life.

perform a deep analysis of such large and complex information. Finally, traditional programs are too rigid to be adapted to new data and often stay unchanged until a new release. On the other hand, ML tools are, by nature, adaptive to changes in the environment they interact with. Applications in such contexts include spam detection programs, which can automatically adapt to changes like spam emails.

Since learning is a very wide domain, it is possible to identify several subfields dealing with different types of learning tasks. Such learning tasks can be grouped into three different sets (i.e., supervised learning, unsupervised learning, and semi-supervised learning) according to output availability. In supervised learning tasks, we have the exact output value and we can easily map the association between input and output variables. On the other hand, in an unsupervised learning scenario, the output value does not exist at all and the aim is to find some common patterns that identify groups or anomalies in the data. Finally, in semi-supervised tasks, we have the exact output value, but some common patterns need to be detected to improve the recognition performance of an algorithm. Figure 1.2 shows an overview of these different learning tasks.

More formally, we can define unsupervised, supervised, and semi-supervised learning problems as follows:

Unsupervised Learning: there is no outcome measure present in the dataset. The goal is to describe the associations and patterns among a set of input measures. Examples of such problems are:

- *Data Clustering*: given a data matrix D, the goal is to partition its records into sets $C_1, \ldots, C_k$, such that the records are most similar to one another. Note that this is an informal definition because clustering allows for a wide variety of definitions of similarity, some of which are not clearly defined in closed form by a similarity function. An alternative definition of clustering is often related to optimization problems where the variables of the optimization problem represent cluster memberships of data points, and the objective function maximises a concrete mathematical quantification of intra-group similarity in terms of such variables.

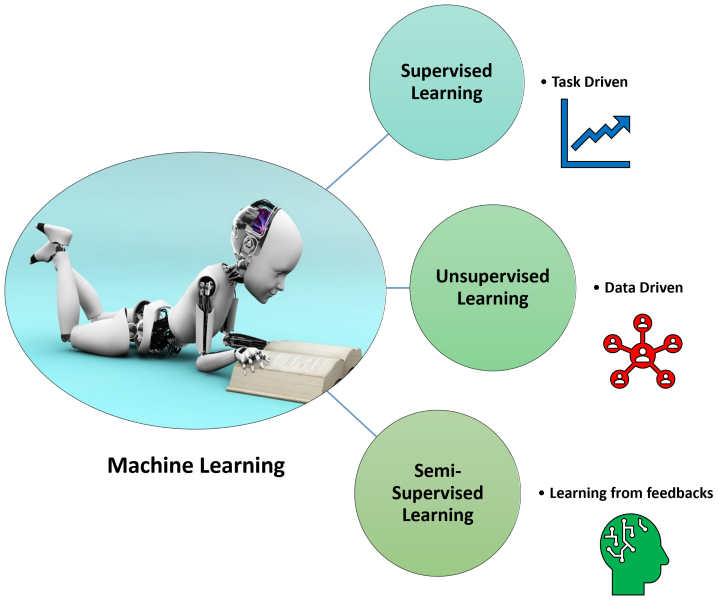


Figure 1.2: The different learning tasks in ML.

- *Outlier Detection*: given a data matrix D, the goal is to identify outliers, namely the records that are significantly different from the other ones. Hawkins formally defined an outlier as follows: "An outlier is an observation that deviates so much from the other observations as to arouse suspicions that it was generated by a different mechanism". Note that outliers are often referred to as discordant, deviants, abnormalities, or anomalies in the literature. The outlier detection problem is related to the clustering problem by complementary. This is a consequence that outliers correspond to dissimilar data points from the main groups in the data, whereas the main groups in the data define clusters.

Supervised Learning: there is an outcome measure in the dataset. The goal is to predict the value of an outcome measure based on several input measures. Examples of such problems are:

- *Data Classification*: given an $n \times d$ training data matrix D (database D), and a class label value in $\{1 \ldots k\}$ associated with each of the n rows in D (records in D), we want to create a training model M, which can be used to predict the class label of a d-dimensional record. Many ML problems are directed toward a specialised goal that is sometimes represented by the value of a particular feature in the data. This particular feature is referred to as the class label. Therefore, such problems are supervised, where the relationships of the remaining features in the data concerning this special feature are learned. The data used to learn these relationships are called training data. The learned model may then be used to determine the estimated class labels for records, where the label is missing. The record, whose class label is unknown, is referred to as the test record.

- *Regression*: given an $n \times d$ training data matrix D (database D), where i-th record is the d-dimensional input feature vector X_i, and the corresponding response variable y_i, the goal is to model the dependence of each response variable y_i on the corresponding independent variables X_i (e.g., as linear relationship, in linear regression problems).
- *Forecasting*: similar to regression in main aspects, but the concept of time is present in these types of problems. Moreover, forecasting can provide insights both in the short term and in the long term. Note that forecasting is one of the most common applications of time series analysis which finds applications when we want to investigate future trends (e.g., weather forecasting, stock markets, and economic indicators).

Semi-Supervised Learning: datasets present both labelled and unlabelled records and the goal is to improve the effectiveness of classifiers. Although unlabeled data does not contain any information about the label distribution, it does contain a significant amount of information about the manifold and clustering structure of the underlying data. Note that classification problems are the supervised version of clustering problems. For this reason, this connection can be leveraged to enhance classification recognition performance. The core idea is that in most real datasets, labels vary smoothly in overdense regions of the data. Determining dense regions in the data requires only unlabelled information. Examples of such problems are:

- *Active Learning*: since in real life it is often expensive to acquire labels, active learning can deal with this problem. Indeed, in these types of problems, the user is actively involved in determining the most informative records for which the labels need to be acquired. Typically, these provided records are the more informative ones to understand the distribution of the class label is unknown. An example of the application of this problem is Amazon's AI, which learns through a like or rank mechanism and can suggest recommended products based on the information you provide.
- *Reinforcement Learning*: this type of problem learns what to do, and how to map situations to actions, and also maximises a numerical reward signal. In this context, the learner does not know which actions to take, as happens in most forms of ML, but instead, it must discover which actions provide the most reward by trying all of them. The key difference between supervised learning is that it needs interactions with users to understand their tastes, whereas in supervised learning the available data is already provided to the learner by a competent external supervisor. Note that in interactive problems it is often impractical to obtain examples of the desired behaviour that are correct and representative of all the situations in which the learner has to act. Moreover, reinforcement learning problems, it is the management of the trade-off between exploration and exploitation, which does not occur in other learning problems. To obtain the maximum reward, the reinforcement learning agent has to try numerous combinations of actions and gradually choose the best one. To do this, the user's past experiences are preferred to have a prediction that is congruent with the subject but trying to predict new experiences that may satisfy the user.

Up to now, we have observed what ML is and what kind of tasks it can solve. But how can we solve these tasks? Many artificial intelligence tasks can be solved by designing the right knowledge (i.e., the set of features) to extract for that task, and then providing this knowledge to a simple ML algorithm. For example, a useful feature for speaker identification from sound is an estimate of the size of the speaker's vocal tract. This can provide a strong clue as to whether the speaker is a child, a woman, or a man. However, for many tasks, it is difficult to know what features should be extracted. For example, suppose that we would like to write a program able to detect cars in pictures. We know that cars have wheels, so we might use the presence of a wheel as a feature. Unfortunately, it is difficult to describe exactly what a wheel looks like in terms of pixel values. A wheel has a simple geometric shape but its image can vary, making it difficult to correctly identify it due to effects such as shadows falling on the wheel, the car wing, an object in the foreground obscuring part of the wheel, the sun dazzling the metal parts of the wheel, and so on. In the literature, the problem just described, namely the need for knowledge to guide the ML algorithm, has led to the distinction between two different families of ML models: shallow models [SSBD14], those that benefit from features extracted by an expert user of the domain [ZC18, Dub20], and deep models [GBC16], those that automatically identify the best set of features directly from the data. Which of the two families is the best? Unfortunately, there is no general rule, but the results depend on the problem and the amount of data available. Shallow models are simple, stable, efficient, reliable and often more intuitive. One of the disadvantages of shallow models is due to the feature engineering phase, where an experienced user may find better features than an ordinary user. Deep models, on the other hand, are potentially much more powerful in terms of performance than shallow models but require a lot of data to outperform them. Figure 1.3 shows that for a small amount of data, shallow models are preferable in terms of performance, a scenario that is reversed when there is a lot of data available. Moreover, deep models are very complex and require the high skills of the programmer who will have to implement them and manage their complexity, which often results in high instability of model performance.

Although research into deep models dates back as far as the 1950s, it is only in the last decade that these models have found their way into many AI tasks and found interest among stakeholders and researchers. Figure 1.4 shows the trend of Google searches for these terms from 2012 to the present. Deep learning models have been successfully used in commercial applications since the 1990s but were often regarded as being more of an art than technology, and something that only an expert could use, until recently. Some skill is indeed required to get good performance from a deep learning algorithm. Fortunately, the amount of skill required reduces as the amount of training data increases. Nowadays, learning algorithms achieve human performance on complex tasks that were unimaginable in the past, when researchers struggled to solve toy problems (the 1980s). This was made possible by algorithms that optimise the training phase of very deep architectures. One of the most important new developments is that today we can provide these algorithms with the resources they need to succeed. This trend is driven by the increasing digitization of society. Indeed, as more and more of our activities take place on computers,

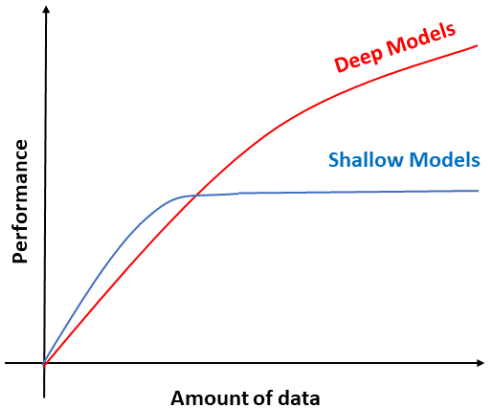


Figure 1.3: The performance of shallow and deep models when varying the amount of data.

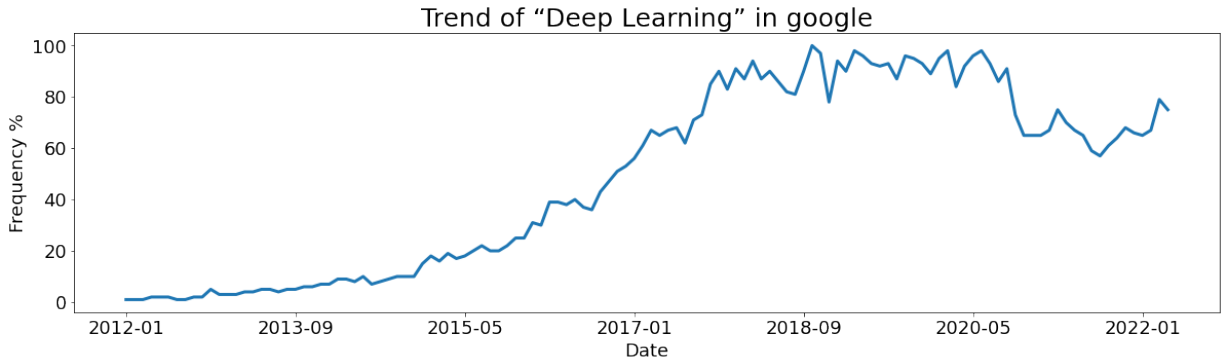


Figure 1.4: Trend of Deep Learning researches in google from 2012. Plot generated with `https://www.google.com/trends`.

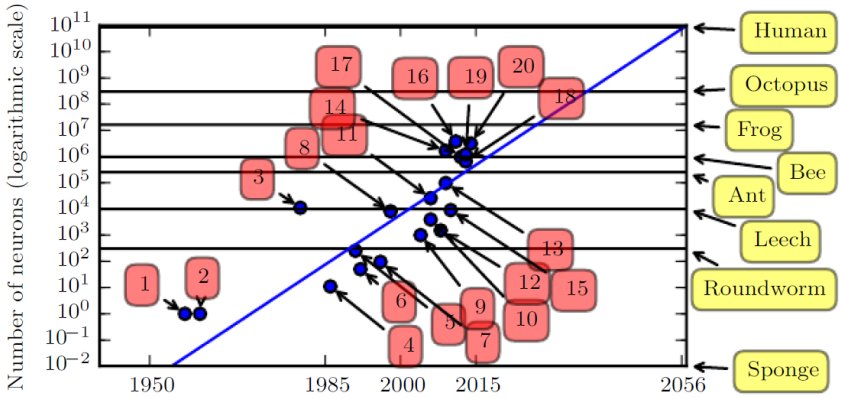


Figure 1.5: Since the introduction of hidden units, artificial neural networks have doubled in size roughly every 2.4 years. Biological neural network sizes from [GBC16].

smartphones and smartwatches, more and more of what we do are recorded. As our devices are increasingly networked together, it becomes easier to centralise these records and use them as a dataset for ML applications. The "Big Data" era has made ML much easier because the key burden of statistical estimation, namely generalising well to new data after observing only a small amount of data, has been considerably lightened. Another key reason why neural networks are so successful today, after enjoying relatively little success since 1980, is that we have the computational resources to run much larger models today. These models, strongly inspired by animal synapses, are inspired by one of the main insights of connectionism; namely that animals become intelligent when many of their neurons work together. A single neuron or a small collection of neurons is not particularly useful. In terms of the total number of neurons, neural networks have been surprisingly small until recently, as shown in Figure 1.5. Since the introduction of hidden units, artificial neural networks have doubled in size roughly every 2.4 years. This growth is driven by faster computers with more memory and the availability of larger datasets. Larger networks can achieve greater accuracy on more complex tasks. This trend looks set to continue for decades. Biological neurons can represent more complicated functions than today's artificial neurons, so biological neural networks could be even larger than this graph shows. In retrospect, it is not particularly surprising that neural networks with fewer neurons than a leech were unable to solve sophisticated artificial intelligence problems. Even today's networks, which we consider quite large from a computational systems point of view, are smaller than the nervous system or even relatively primitive vertebrate animals like frogs. The increase in model size over time, due to the availability of faster CPUs, the advent of generic GPUs, faster network connectivity and better software infrastructure for distributed computing are one of the most important trends in the history of deep learning. This trend is generally expected to continue.

1.2 Thesis Outline

In this thesis, we will focus on the analysis of human movement using data-driven techniques. In particular, we will focus on the non-verbal aspects of human movement, with an emphasis on full-body movements. These movements, to be analysed, can be acquired using non-invasive sensors (e.g., video camera, Kinect) or invasive sensors (e.g., Mocap, Inertial Measurement Unit). The difference between the two approaches is mainly related to three factors. The first factor is the discomfort of wearing a device (more or less obtrusive) on one's body during movement. In the most extreme cases, this can degenerate into wearing actual suits to be worn (e.g. Mocap). The risk of using this type of sensor is to cause stress to the participant, compromising their performance in the long term. On the other hand, invasive devices are much more accurate and robust in acquiring human movement data. Indeed, these sensors do not need to deal with problems such as light conditions or possible occlusions typical of video camera sensors. Unfortunately, however, the precision of detail of these sensors comes at a price. Their use is often in the research field as it is not an affordable expense for everyone.

In any case, these types of sensors can acquire information on human movement, often described as time series. At each instant of time when the action of interest is performed, these sensors provide instant-by-instant (timestamp) information either on the spatial position of the skeletal joint or on the muscular activation of a particular muscle.

In order to process this data, several methodologies in the literature exist related to the different disciplines that study human movement. As already mentioned, the focus of this thesis will be on data-driven methods. The data-driven methods can interpret the information in the data by searching for rules, associations or patterns that can represent the relationships between input (e.g. the human action acquired with sensors) and output (e.g. the type of action performed). Furthermore, these models may represent a new research frontier as they can analyse large masses of data and focus on aspects that even an expert user might miss.

The literature on data-driven models proposes two families of methods that can process time series and human movement. The first family, called shallow models, extract features from the time series that can help the learning algorithm find associations in the data. These features are identified and designed by domain experts who can identify the best ones for the problem faced. On the other hand, the second family avoids this phase of extraction by the human expert since the models themselves can identify the best set of features to optimise the learning of the model.

This thesis aims to understand how human actions can be modelled by a data-driven model. A better design of these actions can, in the future, lead to numerous advantages when applied to everyday life. In particular, an integration of this technology in the homes of people with physical and/or motor problems can produce numerous benefits: from simple monitoring of health conditions to a prediction of a degenerative health condition, from the prevention of injuries due to critical postures to simple support for everyday life. The idea of this thesis will be to understand

whether there is a correlation between how people perceive and interpret their own and others' movements. The movement of human beings appears to respond to a complex motor system that contains signals at different hierarchical levels [WCH+12, BBKK17, ZDlTH12, Aur12]. For example, an action such as "grasping a glass on a table" represents a high-level action, but to perform this task, the body needs several motor inputs that include the activation of different joints of the body (shoulder, arm, hand, fingers, etc.). Each of these different joints/muscles have a different size, responsiveness, and precision with a complex non-linearly stratified temporal dimension where every muscle has its temporal scale. Parts such as the fingers responds much faster to brain input than more voluminous body parts such as the shoulder. The cooperation we have when we perform an action produces smooth, effective, and expressive movement in a complex multiple temporal scale cognitive task. Following this layered structure, the human body can be described as a kinematic tree, consisting of joints connected. As a first approximation, we can state that larger muscles are slower and are characterised by a slower perceptual response over time for the smaller muscles. Nevertheless, some movements of larger muscles can be fast: for example, the small corrections to keep us in balance to compensate for a loss of balance, to avoid the risk of falling. Note also that the multiple temporal scales nature of the human movement, also characterises how humans perceive other people's movements [Hol09, GdLL15].

Although it is nowadays well known that human movement and its perception are characterised by multiple temporal scales [WCH+12,BBKK17,ZDlTH12,MHA+16,Hol09,GdLL15,SHTF+19, Aur12], very few works in the literature are focused on studying this particular property. For instance, Ihlen et al. [IV10] provided quantitative support for studying the multiple temporal scales in human action and perception using wavelet-based multifractal analysis in the response series of four cognitive tasks (simple response, word naming, choice decision and interval estimation). Camurri et al. [CVP+16] demonstrate that computational models of expressive qualities should operate at different temporal scales starting from previous research on human perception and dance theories [ND19]. Authors of [CVP+16] propose a framework where features are computed at different levels, i.e., low-level features (e.g., velocity) are computed instantaneously, while higher ones (e.g., impulsiveness) are computed on a larger temporal scale. In image recognition tasks like object detection, semantic segmentation, and action recognition, Temporal Convolutional Networks (TCNs) with dilated convolutions [RHGS15, CPK+17, DSND19] have been widely adopted to increase receptive field sizes without increasing model complexity. Indeed, by applying dilated convolutions with different filter sizes, multiple temporal scales can be efficiently captured and the use of this mathematical operation can handle larger temporal contexts efficiently. Recent research, carried out in the European FET PROACTIVE Project EnTimeMent[1], focuses its attention on addressing the importance of multiple temporal scales in movement analysis and prediction. Inside EnTimeMent, Beyan et al. [BKV+21] propose an approach that can model the dynamics of full-body movement data represented on multiple temporal scales where features are processed by two independent and parallel shallow TCNs.

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

Therefore, with this thesis, we will provide a method that can apply the multi-temporal scales property of the human motion domain to deep learning models, the only data-driven models that can be extended to handle this property. We will ask ourselves two questions: what happens if we apply knowledge about how human movements are performed to deep learning models? Can this knowledge improve current automatic recognition standards?

In this thesis, we will try to answer these questions. Note that in order to obtain effective, complete and robust answers, we will analyse both families of methods whenever possible. The results will demonstrate that although the majority of research follows the direction of deep models because, when there is a lot of data available these provide better results, shallow models remain a high standard to overcome to date. In the analysed datasets, these (shallow) models produce distinctly high recognition performance, often better than (deep) models specifically designed to handle time series problems. This assumption will prove particularly truthful in our experiments both for small and large datasets. The deep models that we will analyse in this thesis are of two types: the first, used as a baseline to compare the results of the shallow models and our proposal, based on a recursive architecture called LSTM that represents the state-of-the-art to date; the second, used to evaluate our proposal, based on the different intrinsic time scales of human motion. The results will show that LSTM architectures achieve far lower recognition performance than shallow models. On the other hand, our proposed architecture will be able to outperform both models (shallow and deep). In order to prove the validity of our study, we collected data and tested our hypothesis in a specially designed experiments.

Below are the papers written during my doctoral work that will be presented in the Chapters 3, 4, and 5, i.e., in the Methodology, Application & Data, and Experimental Result chapters:

1. *D'Amato, Vincenzo, et al. "Understanding violin players' skill level based on motion capture: a data-driven perspective." Cognitive Computation 12.6 (2020): 1356-1369.*;
2. *D'Amato, Vincenzo, et al. "Accuracy and intrusiveness in data-driven violin players skill levels prediction: Mocap against myo against kinect." International Work-Conference on Artificial Neural Networks. Springer, Cham, 2021.*;
3. *D'Amato, Vincenzo, et al. "Keep it Simple: Handcrafting Feature and Tuning Random Forests and XGBoost to face the Affective Movement Recognition Challenge 2021." 2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW). IEEE, 2021.*;
4. *D'Amato, Vincenzo, et al. "The Importance of Multiple Temporal Scales in Motion Recognition: when Shallow Model can Support Deep Multi Scale Models." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022.*;
5. *D'Amato, Vincenzo, et al. "The Importance of Multiple Temporal Scales in Motion Recognition: from Shallow to Deep Multi Scale Models." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022..*

The remainder of the thesis is organised as follows:

1. in Chapter 2, we will present a comprehensive overview of current work on the state-of-the-art of human movement, starting from the first naive hypotheses up to current methodologies.

2. in Chapter 3, we will present the complete methodology we followed in our analysis, also providing the main steps required to improve recognition performance.

3. in Chapter 4, we will present the analysed dataset on which we applied the methodology presented in the previous chapter.

4. in Chapter 5, we will present the recognition performance obtained on the data presented in Chapter 4 following the methodology presented in Chapter 3.

5. in Chapter 6, we will present the conclusions of the application of the methodology described above.

Chapter 2

Related Works

Human motion analysis is the systematic study of human motion to gather quantitative information about the mechanics of a motor task by careful observation, augmented by instrumentation for measuring body movements, body mechanics and the activity of the muscles [CDCLC05]. In literature, this represents one of the fanciest studies since exploring the reasons that cause or lead to performing movement can provide insights when investigated. The first scientific evidence of human movement is attributed to Aristotle, who described it in terms of the mechanical, mathematical and anatomical paradigms developed during Greek antiquity [LC12]. In the Renaissance, Leonardo da Vinci first investigated human anatomy by identifying the muscles and nerves of the human body and describing the body mechanics during different tasks (e.g. walking, jumping, and standing) [LC12]. Furthermore, Leonardo da Vinci's studies suggested naive techniques to demonstrate that muscles interacted with each other during movement in a progressive activation. Despite Leonardo da Vinci's descriptions of the human body, it was not until the mid-16th century that Vesalius published the first anatomy book, De Humani Corporis Fabrica, which earned him the credit of being the "father of modern anatomy" [LC12]. Galileo's discoveries in physics at this time were central to the establishment of a foundation for the study of the mechanics of human motion (biomechanics) [AKH+18]. William Harvey published De Motu Cordis introducing the proposition of blood circulating through the body, a cornerstone for modern physiology [AKH+18].

A century later, Borelli published De Motu Animalum (On the Movement of Animals), where he clarified muscular movement and body dynamics by estimating the centre of mass [LC12]. Borelli is considered the "father of biomechanics".

The late 17th century and early 18th century saw some of the ideals for movement and physical activity espoused earlier by the Greek philosophers re-emerging in the writings of philosophers. For instance, Locke's Some Thoughts Concerning Education [AKH+18] contained the famous dictum "mens sana in corpore sano" (a sound mind in a sound body), which represented a com-

plete reversal of the prevailing philosophy of the Middle Ages. This investigation implies that the study and care of one require the understanding and development of the other. Rousseau's Emile [AKH+18] advanced the view that movement, in the form of free play, was critical to cognitive, perceptual, and motor development, thus anticipating one of the predominant themes of modern research in motor control and pedagogy.

The 19th century was a period of great scientific discoveries, many of which laid the foundations for the modern discipline of human movement studies. For example, the understanding of the neural basis of movement was significantly advanced by Bell's discovery of the respective sensory and motor functions of the dorsal and ventral root ganglia in the spinal cord (articulated in his text The Nervous Systems of the Human Body) [AKH+18] and the studies of German psychologist Hermann von Helmholtz and others on nerve conduction velocity [AKH+18]. Similarly, the sub-discipline of biomechanics entered a new period of precision measurement with the release of Muybridge's monumental 11-volume Animal Locomotion [AKH+18], containing, for the first time, techniques for high-speed sequential photographic analysis of the human and animal gait. Hitchcock [AKH+18] collected extensive systematic anthropometric data, culminating in the publication of the Anthropometric Manual, which first appeared in 1887.

At the beginning of the 1960s, the foundations of the discipline of human movement studies were already well laid. Research activity, which throughout the first half of the century had been sporadic and had taken a back seat to the practical issues associated with the profession of physical education, had developed in the 1960s in terms of both quantity and quality and demanded greater recognition and identity. Henry [Hen64] and Rarick [Rar67] tried to define a single theme for the growing but diverse physical education research work addressing a contribution to the beginning of a discipline of human movement studies. In 1975, Whiting [AKH+18] created a new journal with the disciplinary title Journal of Human Movement Studies. The emergence of a worldwide discipline encouraged the search for academic credibility and worthy recognition from those undertaking fundamental research on human movement. With the initial attempts by Henry, Rarick, and Whiting to define a discipline, there was a transition in course offerings and department names at academic institutions that offered studies away from physical education towards alternatives in the field, such as human movement studies or kinesiology. Following initial efforts to define the scope and unifying nature of the discipline in the mid-1960s and beyond, the 1970s and subsequent decades were characterised by the emergence of specialised sub-disciplines (as illustrated in Figure 2.1), each with their professional bodies, meetings and research journals. Although the emergence of specialised sub-disciplines is a positive sign for the discipline of human movement studies in terms of adding depth to the human movement knowledge base, it creates the potential for fragmentation.

This is the consequence of the complexity of human movement. Indeed, movement potential and performance are known to be influenced by many things, including biological factors (e.g., maturation, ageing, training and lifestyle), health factors (e.g., disease, disuse, and injury), and social factors (e.g., motivation, opportunity and incentive). For these reasons, it is clear that a

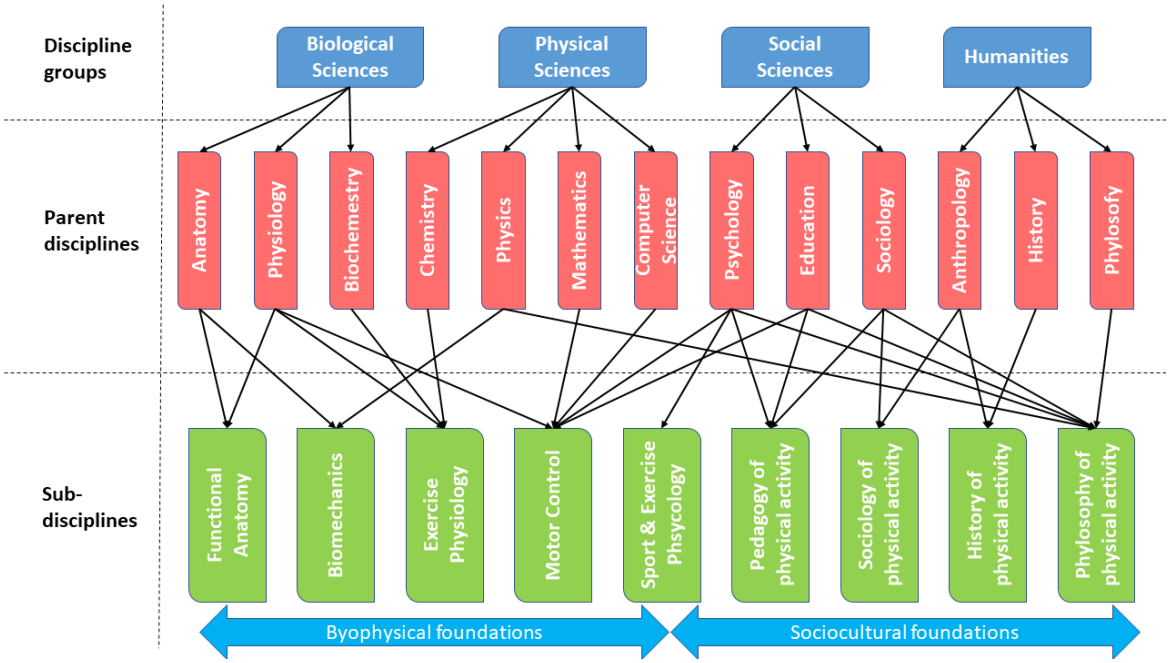


Figure 2.1: One possible conceptualisation of the structure of knowledge about human movement. The discipline of human movement studies is represented by the green boxes.

discipline of human movement studies must draw heavily, but not completely, on the theories, knowledge and methods of a wide range of other disciplines that provide an integrative focus on human movement. For instance, information relevant to the discipline of human movement studies can be derived from the biological disciplines (e.g., anatomical science, biochemistry, and physiology); disciplines in the humanities (e.g., history and philosophy); or physical science disciplines (e.g., chemistry, mathematics, and computer science). Figure 2.1 represents one possible way of conceptualising the organisation of knowledge in a discipline of human movement studies. Figure 2.1 shows that each sub-discipline draws theories, knowledge and methods from one or more related disciplines. However, these disciplines approach a specific study from their own perspective, not integrating theories, knowledge and methods of related disciplines. Figure 2.1 allows us to observe two main points concerning the conceptualisation of the discipline of human movement studies:

- The clustering of disciplines into groups of disciplines and the selection of sub-discipline groups is necessarily somewhat arbitrary.
- The disciplines and sub-disciplines are organised in Figure 2.1 to present a generally progressive shift from a focus on the micro-phenomena to the macro-phenomena of human movement.

In Figure 2.1, each sub-discipline is essentially a distinct component, occasionally with similar characteristics to others. To some extent, this representation accurately reflects the state-of-the-art in this research field, with increasing differentiation and specialisation of human movement studies that may produce fragmentation and an inevitable loss of integrity of the disciplinary base. However, the study of human movement should be represented as multidisciplinary, whereas the desired direction is to make it more interdisciplinary and eventually interdisciplinary or trans-disciplinary (Figure 1.2).

Nowadays, the study of human movement underwent this mutation receiving contributions from many research areas (or disciplines), such as cognitive neuroscience, experimental psychology, biomechanics, interaction design, artificial intelligence, and theories from the arts [Pie98, KBB12, PSOC16]. This mutation is the consequence of the complexity behind human movement that can be affected by many aspects, such as social interactions, behavioural situations, and physical impairments. Each of these aspects influences human activities, and a partial study on one of these is not able to predict the whole reasons that generate a movement. Moreover, each aspect can vary from person to person and by the emotion felt in that particular moment. It becomes difficult, therefore, to correctly distinguish a cause or emotion for all individuals, as everyone will have a different mode of execution in human motion tasks. Furthermore, the same action can be different, even if performed by the same person (e.g., joy and fear change our walking). Therefore, it is straightforward to understand that studying the full complexity related to the analysis of human movement becomes unfeasible for just one research field. For this reason, the different disciplines address this field, each with its perspective and knowledge [Hol09, SHTF+19, HRF+18, DVO+20, PSOC16, Abe13]. For instance, experimental psychology and cognitive neuroscience provide theoretical frameworks and cognitive models [Abe13, Eno08]. On the other hand, computational methods (e.g., data mining and machine learning) can leverage data collected by specific sensors (e.g., video, motion capture, and inertial sensors) to provide insights into movement qualities [DVO+20, PSOC16].

At a lower level, we can find a further fragmentation of the study of human movement concerning the contexts of use of the various disciplines. Indeed, according to [Sze10], we can identify three main research areas where human movement is often studied: surveillance (e.g., checking for critical events such as fall detection in frail elderly), control (e.g., improving the mobility of a patient), and analysis (e.g., understand the quality of full-body movement in sports or expressive emotional communication). Moreover, in these research areas, we can find a further fragmentation of studies due to different applications. Studies in human movement find many applications [KBB12], including physical rehabilitation, sports scoring and skill assessment [LDZ+19, DGPAC19] and applications involving the full-body expression of emotions and non-verbal social signals.

2.1 Social Signal Processing

In this context, the Social Signal Processing (SSP) research field aims at bridging the social intelligence gap between humans and machines [VPH+11]. Note that, different definitions of social signals exist in literature [VP15, MS12, PD12]. In this work, we opted for the one defined by Vinciarelli et al. [VP15] 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. [VPH+11], also distinguish three major components in SSP, i.e., modelling, analysis, and synthesis of social behaviour. In particular, the modelling phase focuses on the 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 on 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 [VPB09, ABR00].

For example, Ambady [ABR00] focus on how human beings can understand social signals, even if there exists a wide variety of non-verbal behaviours depending on the individual characteristics of the different people. The understanding and interpretation of body language are fundamental to studying the behaviour of human beings involved in social interactions. In particular, the authors of [ABR00] discuss the cognitive and affective mechanisms that influence the process of information from thin slices of the behavioural stream. Moreover, the authors of [VPB09] discuss how next-generation computing needs to include the essence of social intelligence, the ability to recognise human social signals and social behaviours like turn-taking, politeness, and disagreement, to become more effective and more efficient.

Figure 2.2 reports the situation involving one man and one woman. Observing only the behavioural cues from the two silhouettes, we can understand that they are arguing. Every detail of the human body says something about what is going on, even if we have nothing more than a simple visual input. Similarly, looking at Figure 2.3 is possible to understand which is the social context. The man's body posture, with extended hands and palms facing up, suggests positive vibes toward the child as if he wanted to help him. The behavioural cues perceived from this picture are very different from those in the previous figure but the social signal aspects are essentially the same. This phenomenon highlights how the body's motion can transmit multiple pieces of information to an external observer watching a social interaction happen.

2.2 Non-verbal Behavior

Non-verbal behaviour are primary related to full-body movements (e.g., gesture [LS16] and posture [CO67]) and secondary to cues for the perception and interpretation of social signals (e.g.,

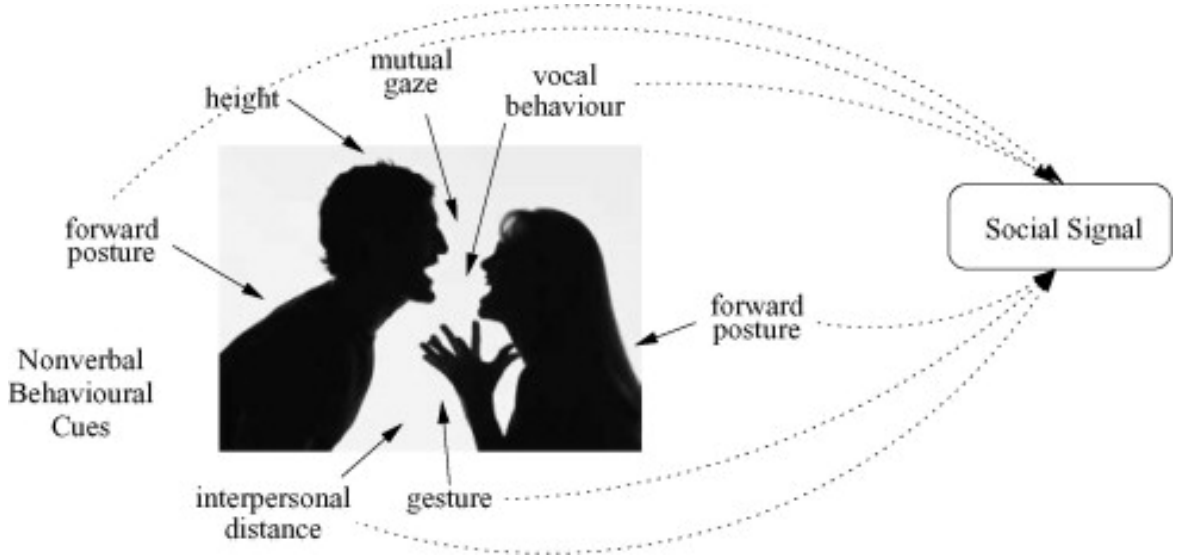


Figure 2.2: Social signals from non-verbal behaviour, first example [VPB09].

facial expression [WTF19] and mutual gaze [FWH+19]), as shown in Figures 2.2 and 2.3. Consequently, understanding and interpreting them is a fundamental step in deepening social interactions. In this thesis, we focus on non-verbal behaviour conveyed through full-body movement. To better understand the role of full-body movement, it is necessary to provide an overview of the historical evolution of this research field:

- *the Early 1950s*: flourishing of the investigation concerning non-verbal communication, to which full-body movements belong. This is a consequence of the increasing interest in semiotics [EF69], defined as the study of signs and symbols and their use or interpretation [Eco16], a field that goes beyond the commonly spoken language.
- *Up to 1970s*: the focus of the research considered mainly unimodal systems, namely, the system that can handle a unique channel of sensory input/output, like one of the aspects mentioned above (posture [CO67], gesture [LS16], facial expression [WTF19], etc.).
- *End of the 1980s*: the explosion of the need to integrate multiple modalities communication systems [PS13]. The possibility to consider at the same time a large number of inputs/outputs coming from different sensory channels adds significance to the analysis of human behaviour. Nevertheless, multimodality can still be considered an open research problem, as it is not completely known how all information from different sensory channels is perceived and merged by humans.

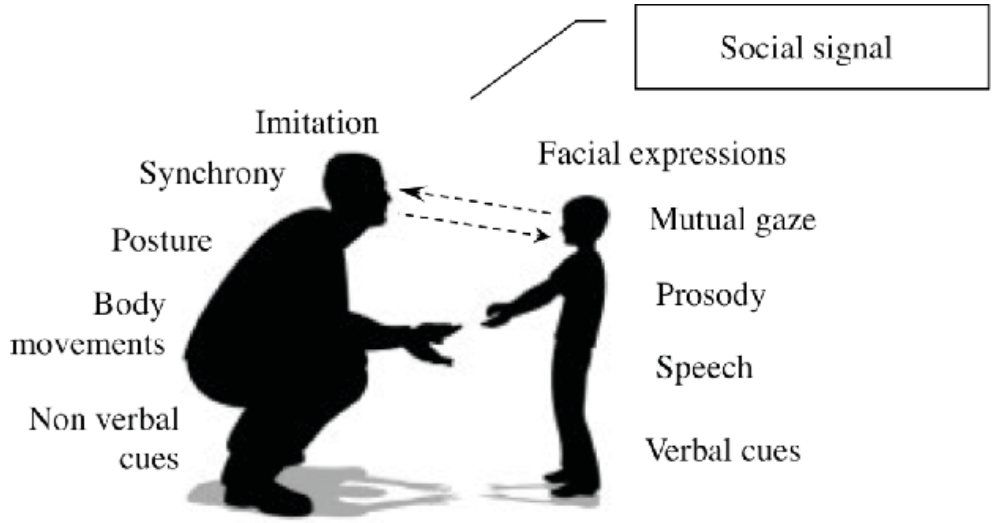


Figure 2.3: Social signals from non-verbal behavior, second example [CCPC12].

2.3 Full-body Movement

Full-body movement is a specific component of non-verbal behaviour. These movements can be effectively and efficiently acquired with Motion Capture (Mocap), namely, a technology that allows capturing both fine and gross movement features (posture, position of limbs, direction, and speed of movement) while humans do it (unconsciously) with a much lower level of details and accuracy [EK20]. Such characteristics can help to detect non-verbal social cues [KBB12, KSG+13]. Their interpretation, instead, can be easily performed by humans while for machines it can be a challenging task [EK20, FBS+21]. For example, Caramiaux et al. [CDT15] discussed how difficult it is to understand what makes a gesture expressive because this operation implies considering different aspects such as dynamics, the mechanism that enacts it and spatial location. In recent years, there has been a growing interest in the development of technology that can distinguish the emotion of people [FT05] as the role played by affect in human development and everyday functioning is now well recognised [IA02]. The research increases the focus on the possibility of using body expressions to construct affectively aware technologies. There are three possible reasons for this attention: scientific, technological and social [KBB12, KSG+13]. Firstly, more and more studies from various disciplines have shown that body expressions are as powerful as facial expressions in conveying emotions [Arg13, Bul16, EF74, VdSRDG07]. Secondly, with the increasing ubiquity of technologies used by the everyday person [FT05], they allow for multimodal interaction in which bodily expressions assume a richer role, even beyond that of gestures. A typical example is offered by whole-body computer games (e.g. Nintendo Wii and Microsoft Kinect), in which body movement is a way to capture and affect our emotional and cognitive performance, as well as a means to control the interaction between

us and the computer [NBW+05, CS09]. Thirdly, bodily expressions could be integrated into crucial applications for many areas of society, such as health care, education, security, law enforcement, games and entertainment. For instance, in chronic pain rehabilitation, the authors of [KLJ03, HHK+06] showed how specific movements and postural patterns (i.e., guarding behaviours) provide information on the emotional conflict felt by patients and their level of ability to relax. Clinical practitioners use this information to tailor support to patients during therapy. In teaching support instead, the teacher can react to the body language and actions of the students to improve the learning process and maintain motivation [DVO+20]. Indeed, students tend to lose motivation when a high level of affective states (e.g., fear of failure, frustration, anxiety) occurs. Affect expression occurs through verbal and nonverbal communication channels such as eye gaze, facial expressions, and bodily expressions [PR00]. However, the research mainly focused on non-verbal affect recognition, on facial expression in particular [DG09]. The study and the analysis of facial expression represent the basis for learning how human processes affect neurologically [Ado02]. On the other hand, the research on body movement and posture. As shown by the authors of [DG09], there is a wide gap between studies on facial expression stimuli, audio stimuli, and whole-body expressions. However, bodily expression represents a crucial point in non-verbal communication [MF69, Arg13]. The authors of [MF69, Wal98] showed how body posture changes an affective state of a person. Moreover, the authors of [MF69] found that the body posture change according to the attitude toward their interaction partner. The authors of [APM+16] analysed intrapersonal synchronisation in full-body movements to show how this influences the different expressive qualities. In their experiment, some professional dancers performed different movements from which a dataset was collected. Results showed that movements performed with different qualities display a significantly different amount of intrapersonal synchronization.

2.4 Multimodality

Humans have multiple sensory channels that allow them to perceive and relate to the environment [PS13]. Multiple sources of information concerning the external world and our bodies can provide us with rich, robust, and more precise information, ultimately allowing for more adaptive behaviour. When humans perceive an event (e.g., a dog barking), visual and auditory cues provide information about where the event is. The spatial and temporal properties of the environment (and of the events taking place within it) can be redundantly sensed, though with different levels of precision, via multiple sensory channels, and hence are typically considered as being amodal stimulus properties (i.e., non-modality-specific properties, a concept dating back to Aristotle's sensus communis). Therefore, redundant cues refer to sensory information, often perceived through different sensory channels (though not necessarily through all of them), that refers to the property of the physical world. An example of this behaviour is the size or shape of an object, which can be sensed redundantly by vision and touch. In

other words, multiple senses allow the acquisition of complementary (i.e., non-redundant) information. Returning to the example above, i.e., the barking dog, the colour can only be perceived visually, while the pitch auditorily. Therefore, as perceptual correlates of different stimulus properties, colour and height can be considered complementary cues [GA10]. Over the past decade, research has focused on how our brains integrate redundant cues from multiple sensory channels [MW11, Ste12], such as the seen and perceived dimensions of objects [EB02]. Moreover, empirical research [EB04,TKL11,KBM+07] demonstrated that the integration of redundant cues enables humans (and other animals) to generate more accurate and robust combined sensory estimates. On the other hand, the presence of several complementary clues tuned to different properties of the environment provides non-redundant information about the external environment, enriching part of the richness of our sensory experiences. For over a century, scientists have debated the existence of seemingly arbitrary compatibility effects between complementary intermodal signals even in non-synaesthetes [Spe11, PS13]. For example, most researchers [GS06] readily associate large objects with low sounds, whereas they consider it less natural to match them with jarring sounds. Such observations have led some researchers to assume that all humans are, at least to some extent, synaesthetic [MM01, MW06, WHT06]. Having said this, proponents of these claims argue that the strength of synaesthetic experiences can vary considerably between individuals. Note that complementary cues are often correlated in the real world. Therefore, it could be argued that, rather than constituting a weak form of synesthesia, such intermodal correspondences simply reflect learned associations between features of naturally occurring multisensory stimuli: going back to our initial example, if the barking dog behind the fence is a small dog like a Chihuahua, it is very unlikely that their growl(s) are deep (i.e., low-pitched)!

2.5 Multi-temporal Scales

Recent studies [WCH+12, BBKK17, ZDlTH12, Aur12] show how human movements are hierarchically nested: a movement structure involves muscles of different size, reactivity and precision, with a complex non-linear layered temporal dimension where each one has its own time scale. Each action, even the simplest ones, includes a set of sub-actions involving different body parts that cooperate to create a smooth, effective, and expressive movement in a complex multiple temporal scale cognitive task. For instance, pointing a finger toward an object starts from the movement of the entire body, followed by the upper part, the shoulder, the arm, the finger, etc. The human body can be described as a kinematic tree where joints are connected. As a first approximation, we can state that larger muscles are slower and characterised by a slower perceptual response over time concerning the smaller muscles. Nevertheless, some movements of larger muscles can be fast: for example, the corrections to keep us in balance to compensate for a loss of it, avoiding the risk of fall. Note also that the multiple temporal scales nature of the human movements also characterises how humans perceive other people's movements [Hol09,GdLL15]. Human beings can understand and predict the movements of other humans even from a limited

number of moving points [Joh73]. This skill depends on the ability of humans to create relations between different temporal and spatial layers using forward/feedback connections. This process is driven by the brain and the body as timekeepers coordinating different internal, mental, and physiological clocks. Nevertheless, it is worth noting that recent studies [PPBS01] demonstrate that the information contained in these limited number of moving points does not depend only on the activity performed but also on more complex cognitive and affective phenomena. For example, Meeren et al. [MHA+16] consider the relation stimulus as temporal dynamics of the feedback and affective qualities.

Although it is nowadays well known that human movement and perception have multiple temporal scales [WCH+12,BBKK17,ZDlTH12,MHA+16,GdLL15], few works in the literature are focused on studying this particular property. For example, Ihlen et al. [IV10] 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. [CVP+16] demonstrated that computational models of expressive qualities should operate at different time scales starting from previous research on human perception and dance theories [ND19]. Still, Camurri et al. [CVP+16] proposed a structure where features are calculated at different levels. For example, the extraction of low-level features (e.g. speed) happens instantaneously, while higher-level features (e.g. impulsivity) on a larger time scale. The recent *European FET PROACTIVE project EnTimeMent*[1] focuses on the importance of multiple time scales in motion analysis and prediction. As well as recent research works focus on these issues. For example, Beyan et al. [BKV+21] proposed an approach that can 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. [YLZJ19] showed how a 3D Convolutional Network based on multiple temporal scales outperformed a standard deep learning model with a single temporal scale in action recognition tasks. Stergiou et al. [SP21] proposed a novel convolutional block (MTConv) useful to extract spatio-temporal patterns in action recognition problems. Lin et al. [LZLQ21] proposed a novel multi-scale temporal information extractor able to aggregate temporal information from different temporal scales in gait recognition tasks.

2.6 Machine Learning for Human Movement Analysis

In the last few years, data-driven models (based on Machine Learning and Data Mining) played a crucial role in the advancement of SSP [JGS+20] thanks to the availability of large amounts of data required by these models to work properly and effectively [GBC16, BWFDS19]. Data-driven models can extract meaningful and actionable information from these large amounts of data to provide insights into the complex processes in social signals as core tools able to em-

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

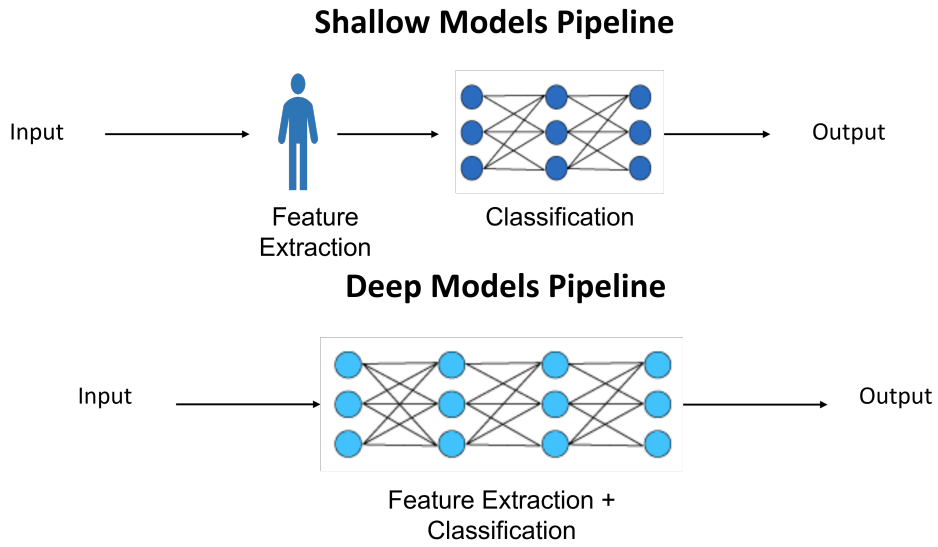


Figure 2.4: Pipelines for shallow and deep models.

power and supplement expert-or-physics-based models [KIK+18]. Recently, this trend has been accelerated by both the unexpected success of these tools in solving a real-world problem with super-human performance [CMMS11, HUE+19, SSS+17, JEP+21] or the expectation to do so soon [GSD+18]. Following this trend, in this thesis, 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 [SSBD14], models that require handcrafted features based on domain-specific knowledge (see Figure 2.4), can achieve good recognition performance and their limitations in handling the multiple temporal scales that characterise the human movement. Secondly, we will analyse how deep models [GBC16], models that can automatically learn features from data (see Figure 2.4), can be extended to handle multiple temporal scales but cannot be naively applied to address the problem due to real-world difficulties (limited data available in most applications and a huge number of architectural choices to explore to obtain optimal results).

In literature, human movement studies use both families of methods based on the cardinality of the sample size. In small cardinality datasets, Deep Learning-based methods cannot be employed since they require a huge amount of data to be reliable and outperform traditional ML models with context-specific experience-based engineered features. For this reason, in most studies, shallow ML models are employed. These models are applied successfully to the field of human movement studies and, in particular, in the cognitive computation [ARB18, OSDCI17, KDVdS17, WXWL18]: in sequential learning [ZWS+18], in sentiment analysis, in data management [WZL+17] and in classification problems [SU17, Hua14].

2.6.1 Shallow Models in Literature

Weiwei et al. [Wei22] used Random Forest (RF), a shallow model, for the automatic detection of three different sports actions (i.e. running, jumping and walking), demonstrating better accuracy and effectiveness than more complex deep models. Javeed [JJK21] uses Random Forest on multimodal data obtained through wearable sensors such as the Inertial Measurement Unit (IMU), Mechanomyography (MMG), and Electromyography (EMG) to identify health disorders such as asthenia. Hafeez et al. [HJK21] uses Random Forest for estimating the UTD-MHAD dataset activities [WLHL16], achieving higher accuracy than previously used deep models such as LSTM and CNN. In this study, the authors extract features from depth images (acquired via Kinect), 20 skeleton points, and two wearable inertial sensor accelerometers positioned on the right wrist and right thigh of the subject. Thakur et al. [TB22] use an RF-based algorithm to extract the most informative features in two Human Activity Recognition (HAR) datasets collected using smartphones. The authors compare this technique, where an SVM (Support Vector Machine) classifier is added, with a deep model like CNN, previously used to solve the same task. Radhika et al. [RPC22] also use RF to classify six actions (i.e., lying, sitting, standing, walking, walking downstairs, and walking upstairs) typical of HAR problems captured by participants' hand-held smartphones. The results show that RF achieves better classification metrics (i.e., F1-score, accuracy, precision and sensitivity) than shallow models such as KNN and SVM.

Pribadi et al. [PS22] train an SVM to analyse data acquired through an Inertial Measurement Unit (IMU) sensor applied on the hand of welders. By extracting features such as root mean square (RMS), correlation index, spectral peaks and spectral power, it was possible to distinguish skilled and unskilled welders simply by analysing the movements they perform in their work. Yin et al. [YYY22] train an SVM to identify the motion state of the human body through adaptive time window segmentation, using signals from foot force sensors and inertial measurement unit (IMU) sensors. In their research, the authors analysed six activity types (i.e., sitting, standing, going upstairs, going downstairs, walking and jogging) and obtained very high recognition performance. Shioiri et al. [SSFK21] compare an SVM and a CNN for the classification of fall and no fall risk, typical of gait analysis problems. In particular, the authors use data acquired with micro doppler radar, demonstrating how a shallow model trained on spectrogram images such as SVM achieves better results than a deep model such as CNN, with a 6% higher classification rate. Zhou et al. [ZY21] uses an SVM trained by HOG (Histogram of Oriented Gradient) and LTP (Local Trinary Patterns) features extracted from images to improve pedestrian detection efficiency. Specifically, the authors use the weighted fusion method to merge the colour map features with the depth map features and finally use the classifier to detect pedestrians.

Wang et al. [WZJ21] train a K-Nearest Neighbour (KNN) model to learn and automatically recognise six human actions, including walking, climbing stairs, walking downstairs, sitting, standing and lying down. Simply by using a three-axis acceleration sensor embedded in a smartphone, the authors obtain an average accuracy of 96.70% for optimal k values. Siddiqui et al. [SGK22] compared several machine learning algorithms for classifying human movements

using Mocap data. Specifically, data were recorded from a participant performing a stacking scenario comprising simple arm movements at three different speeds (slow, normal, fast). The models were trained on actions performed on slow and normal speed movement segments and then generalised to fast speed movements. The authors identify KNN as the best algorithm for solving their task, achieving an accuracy of approximately 99%. Eltanani et al. [ESD21] train KNN to automatically recognise the gait patterns of healthy individuals and patients suffering from irregular gait patterns caused by physically disturbing conditions, such as strapping muscles. The authors show that, for the classification of well-apparent (normal) and ill-apparent (strapped) gaits in the MOReS dataset, they achieve an accuracy of 67.7%. Rahman et al. [RRI$^+$22] used KNN for the automatic detection of hand gestures (i.e., closed hand, open hand, OK sign and downward indication) collected from two distinct views (i.e., lateral and frontal) of two continuous waves (CW) radars. The classification results indicate that both hand gesture signals from the two radars achieved high accuracy when exploring a Leave-One-Out (LOO) scenario.

Gao et al. [GMW$^+$22] set the goal of automatically recognising lower-limb movements. They compare data acquired through EMG, IMU and a combination of them to distinguish three posture patterns, including walking on the floor, squatting and leg extension while seated. The authors demonstrate how eXtreme Gradient Boosting (XGBOST) with Bayesian optimisation is better in terms of recognition performance (i.e., F1-score, accuracy, precision, and recall) than RF and a deep model such as MLP. Purnomo et al. [PLAH21] propose a classification system for breathing patterns using the XGBoost and Mel-frequency cepstral coefficient (MFCC) feature extraction. Breathing patterns are collected using FMCW radar technology useful for developing non-contact medical devices. The results of the respiratory pattern classification were presented on a dataset consisting of five breathing patterns, achieving an accuracy of 87.38%. Vong et al. [VTW$^+$21] demonstrate that an XGBoost classifier is the best algorithm to recognise human activity and falls. In particular, the authors starting from the UniMiB SHAR dataset [MMN17] that records 17 activities (9 basic daily activities and 8 fall activities), use four feature extraction methods (i.e., Chi-square, mutual information, ANOVA and Pearson's coefficient), obtaining a total of 336 features. The authors observe how XGBoost incorporated with the mutual information method offers the best performance, with accuracy, precision, recall and F1 scores of 91.22%, 87.27%, 86.29% and 86.40%, respectively. Lisca et al. [LPGA21] acquire data from a single motion sensor to evaluate the goalkeeper's movement in a football team and provide an easy-to-understand explanation of goalkeeper kinematics. A pair of special goalkeeper gloves, with an embedded IMU sensor, allows for collecting raw data and quaternion. The authors observe how XGBoost can achieve better recognition performance (i.e., accuracy, precision, recall, macro F1-score, and error rate) in the two classification tasks, even compared to deep models. The two classification tasks are: to evaluate dive types (i.e., dive or not dive) and movement types (i.e., dives, catches, dive stand, throws).

2.6.2 Deep Models in Literature

Cao et al. [C+22] train a Long Short Term Memory network (LSTM) for an automatic human motion recognition system, referring mainly to Facebook's theory of action awareness and expanding the type and range of recognised images. This system uses the bottom-up idea to recognise the human body in a set of images: i.e., once human joints are identified, they are combined as a node in the system. Using LSTM, this method can recognise the actions of different regions without human recognition and then combine them. The authors achieve a recognition rate of the 3D database of utKinect actions of 95.96%. Li et al. [LZQ+21] proposed an LSTM-based recognition information processing system to collect and recognise human movement data. LSTM integrates a complete three-layer human motion recognition processing system, which can simplify the entire data acquisition process and reduce the missing data. The authors achieve an accuracy of 98.30% on the open PAMAP2 dataset [RS12], using LSTM and the proposed system. Xu et al. [XZ22] propose Inception-LSTM, an attention mechanism that takes in input an inertial signal. In particular, they first extract the spatial features from multiple scales using the Inception parallel convolution structure [SLJ+15], then the Efficient Channel Attention (ECA) module to identify the relevant details from the extracted features, and finally, the LSTM network to extract the temporal features to achieve human motion posture recognition. The authors obtain a recognition accuracy of 95.04% on the public PAMAP2 dataset and 98.81% on the self-built dataset. Tang et al. [TTS+21] train an LSTM to prevent chronic spinal problems. To recognise seven postures in unsupported human sitting, the authors use raw data from four 9-axis IMUs evenly distributed between the thoracic and lumbar regions (T1-L5) and aligned in a sagittal plane to acquire kinematic information about the subjects' backs during alternating static-dynamic movements. Bian et al. [BSD22] propose a method based on biomechanical knowledge of movement to predict human movement in human activities (i.e., stand-to-stand and walking). Specifically, the authors study the plausibility of skeletal joint movements using a muscle-skeletal model within an LSTM.

Yi et al. [YZH+22] propose a bi-RNN architecture incorporating two modules: 1) the kinematics module: a neural kinematics estimator, which infers the human motion from the 6 IMUs (placed on the left/right forearms, left/right lower legs, head, and pelvis), followed by 2) the dynamics module: a physics-aware motion optimiser, which refines the human motion and outputs the physical properties. The combination of the two modules leads to greater accuracy and realism, as demonstrated by the experiments. Zheng et al. [ZMY+21] propose a method for reconstructing human posture based on deep learning and scattered inertial sensors. This method uses bi-RNN for human posture reconstruction Human posture reconstruction performance is evaluated with different training data and sensor placement selection methods, and experimental results show that the proposed method is advantageous for both posture reconstruction accuracy and model training time.

Tong et al. [TML+22] train a Bidirectional-Gated Recurrent Unit-Inception (Bi-GRU-I) model on inertial sensors data to evaluate human activities. The proposed model consists of 2 Bi-GRU

layers, 3 Inception layers, 1 GAP (Global Average Pooling) layer and 1 softmax layer. The authors compare their model with those in the literature on three datasets (i.e., the self-collected CATP dataset, Wireless Sensor Data Mining (WISDM) [KWM11] and the University of California, Irvine (UCI-HAR) dataset [AGO+13]) obtaining better recognition performance and robustness. Yu et al. [YLZ+21] use a combination of GRU with a 1D-Convolutional Neural Network (CNN) to reconstruct short- and long-term actions on the Human3.6M database [IPOS13], which contains 3.6 million 3D human postures and covers 15 kinds of motion such as walking, eating, smoking and so on. The authors sequentially use a GRU model to extract temporal features from the data and then apply a 1D-CNN to reduce the number of features. This subset of features is used to generate actions. Tonchev et al. [TMPP21] use a variant of the GRU model to predict human postures in a set of time instances. In particular, the authors optimise the GRU model by replacing the weighting of recurrent inputs and outputs with convolution, using the graph structure of the human skeleton.

He et al. [HFL22] train a Temporal Convolutional Network(TCN) capable of learning both global and local sub-sequence features for time series classification. The authors obtain better recognition performance in terms of average accuracy, average Macro-f1 and classification metrics on the UCR/UEA dataset [BLB+17], which includes 85 time series of eight different types (i.e., device, ECG, image, motion, sensor, simulated, sound and spectrum). Tong et al. [THP21] propose a TCN for the automatic detection of hand tremors to assist physicians in the diagnosis and treatment of Parkinson's disease. Pulse acceleration information of Parkinson's patients with hand tremors and healthy subjects was acquired by a wearable device with an inertial sensor. Through leave-one-out validation, the authors demonstrate how the TCN showed better recognition performance than models in the literature. Boner et al. [BVM22] train a TCN for human gesture recognition on a dataset of 12 users where each participant performs 6 different hand gestures (i.e., swipe left, swipe right, swipe down, swipe up, and pull, push). The proposed architecture is very efficient in handling this classification problem, achieving more than 95.00% accuracy using a small model with a RAM weight of less than 100KB. Tang et al. [TZY22] optimise a two-level TCN model (i.e., one analysing temporal information and one reinforcing spatial-temporal trajectory) to automatically reconstruct human motion. The authors obtain remarkable results on three benchmark datasets, including Human3.6M [IPOS13], the CMU Mocap dataset [2], and the 3D pose in the Wild dataset [VMHB+18] in both short-term and long-term prediction, confirming its effectiveness and efficiency. Sakagami et al. [SYM21] train a TCN on 4 different domains (i.e., time series obtained from radar, such as Time-Doppler, Time-Range, RangeDoppler and their combination) to automatically recognise 10 actions of human movement (i.e., sit in a chair, stand up from a chair, squat down, get up from a squatting position, raise one's hand, lower one's hand, pick up things, fall, walk towards the radar, and run towards the radar). Comparing the four different domains, the authors observe the recognition performances obtained with a minimum accuracy greater than 87.00%.

[2] `http://Mocap.cs.cmu.edu/`

Chapter 3

Methodology

In this chapter, we will present to the readers the methodology we followed in our analysis. We will start describing some preliminaries in Section 3.1. Then we will continue with the presentation of the Shallow Models in Section 3.2. Following Shallow Models, we will present the Deep Models in Section 3.3. Note that we will handle classification problems [SSBD14] where, when possible, we analysed both shallow and deep models. These models will be used to manage time information related to human movements in different applications, as we will observe in Chapter 4. Moreover, in this section, we will describe the best practices needed to exploit the recognition performance of such models. Finally, we will highlight the strategies needed to explain the problem under investigation.

Note that the methodology presented in this chapter has been followed for all the works done in my PhD. The list of articles that followed this methodology is given below:

1. *D'Amato, Vincenzo, et al. "Understanding violin players' skill level based on motion capture: a data-driven perspective." Cognitive Computation 12.6 (2020): 1356-1369*;

2. *D'Amato, Vincenzo, et al. "Accuracy and intrusiveness in data-driven violin players skill levels prediction: Mocap against myo against kinect." International Work-Conference on Artificial Neural Networks. Springer, Cham, 2021*;

3. *D'Amato, Vincenzo, et al. "Keep it Simple: Handcrafting Feature and Tuning Random Forests and XGBoost to face the Affective Movement Recognition Challenge 2021." 2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW). IEEE, 2021*;

4. *D'Amato, Vincenzo, et al. "The Importance of Multiple Temporal Scales in Motion Recognition: when Shallow Model can Support Deep Multi Scale Models." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022*;

5. *D'Amato, Vincenzo, et al. "The Importance of Multiple Temporal Scales in Motion Recognition: from Shallow to Deep Multi Scale Models." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022.*

3.1 Preliminaries

The problems we will observe in Chapter 4 can be easily mapped into a now classical classification problems [SSBD14]. In particular, we will handle time series information, acquired with different sensors in different datasets.

Let $\mathcal{X} \subseteq \mathbb{R}^d$ be the input space (in our cases the different time series available for each dataset), and let $\mathcal{Y} = \{0, 1, \ldots, c\}$ (where c can vary in according to the number of the target classes) be the output space. Let

$$\mathcal{D}_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}, \tag{3.1}$$

where $X_i \in \mathcal{X}$ and $Y_i \in \mathcal{Y}$ $\forall i \in \{1, \cdots, n\}$, be a sequence of $n \in \mathbb{N}^*$ samples drawn from $\mathcal{X} \times \mathcal{Y}$. Let us consider a model (function)

$$f : \mathcal{X} \to \mathcal{Y} \tag{3.2}$$

chosen from a set $\mathcal{F}$ of possible hypotheses. An algorithm

$$\mathscr{A}_{\mathcal{H}} : \mathcal{D}_n \times \mathcal{F} \to f \tag{3.3}$$

characterized by its hyperparameters $\mathcal{H}$ selects a model inside a set of possible ones based on the available dataset. The error of f in approximating $\mathbb{P}\{Y \mid X\}$ is measured by a prescribed metric

$$M : \mathcal{F} \to \mathbb{R}. \tag{3.4}$$

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 [SSBD14] 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), among others (see Section 3.5) . To tune the performance of the $\mathscr{A}_{\mathcal{H}}$, namely to select the best set of hyperparameters, and to estimate the performance of the final model according to the desired metrics, a Model Selection (MS) and Error Estimation (EE) phases need to be performed [One19] (see Sections 3.4). Moreover, to understand, from a cognitive point of view, how the algorithm exploits the derived features to make a prediction a Feature Ranking phase is also performed (see Section 3.2.5).

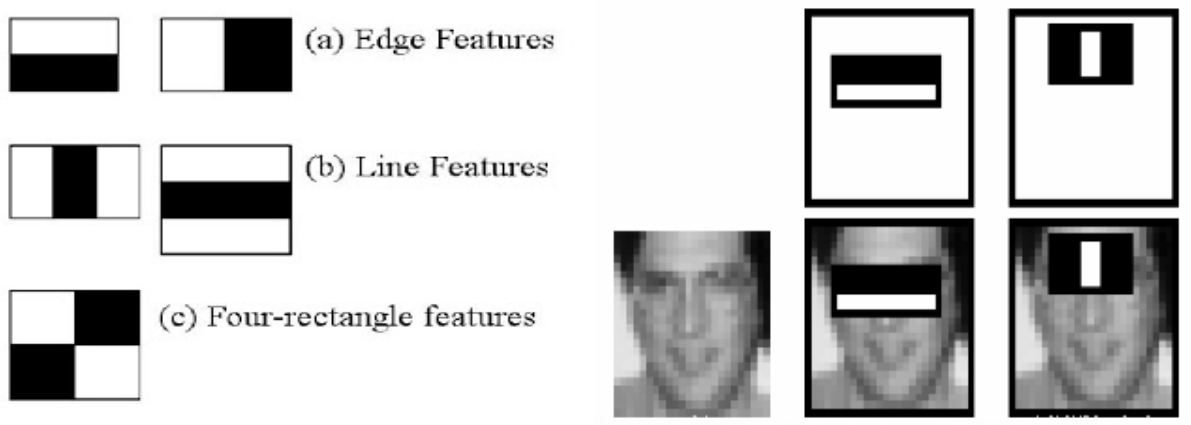


Figure 3.1: The Haar features.

3.2 Shallow Models

As we observed in the previous chapter, there are two very different families in ML: shallow and deep models. But what is the real difference between the two? Let us start with the shallow models. The basis for these models is simple: the more you know about the domain, the better. In other words, these models benefit from a user's domain knowledge to make predictions. Therefore, the human figure is essential as they must be able to identify features that can help the (shallow) models. Applying domain expert's knowledge occurs simply by mapping the raw data $\mathcal{X}$ into a new representation space $\mathcal{X}'$. But what does this mean exactly? Let us consider two examples:

- We want to detect the presence of people in images. A binary classification problem is defined with two target classes, people and non-person. Performing our classification task can lead to analysing the information pixel by pixel to address the problem. However, this will lead to a naive solution to the problem. An alternative approach, where we could provide some advantages to analyse each image with domain expert knowledge, is the use of Haar features [VJ04], i.e., those features designed for detecting faces in pictures.
- We want to detect pain in the elderly using motion sensors. As before, a binary classification problem is defined with two target classes, the presence and absence of pain. A naive ML user could analyse each sensor over time by observing when pain produces an irregular movement. On the other hand, we can provide informative insights by extracting areas of the body that are particularly prone to pain (e.g., by extracting features from the back that describe high-level information about that particular area of the body).

In general, the representation space $\mathcal{X}'$ is obtained with classical signal processing techniques [ZC18, Dub20]. Then, on this new space, we can apply different shallow model algorithms, such for instance: a linear (e.g., Support Vector Machines [STC04]) or nonlinear (e.g. Kernel Meth-

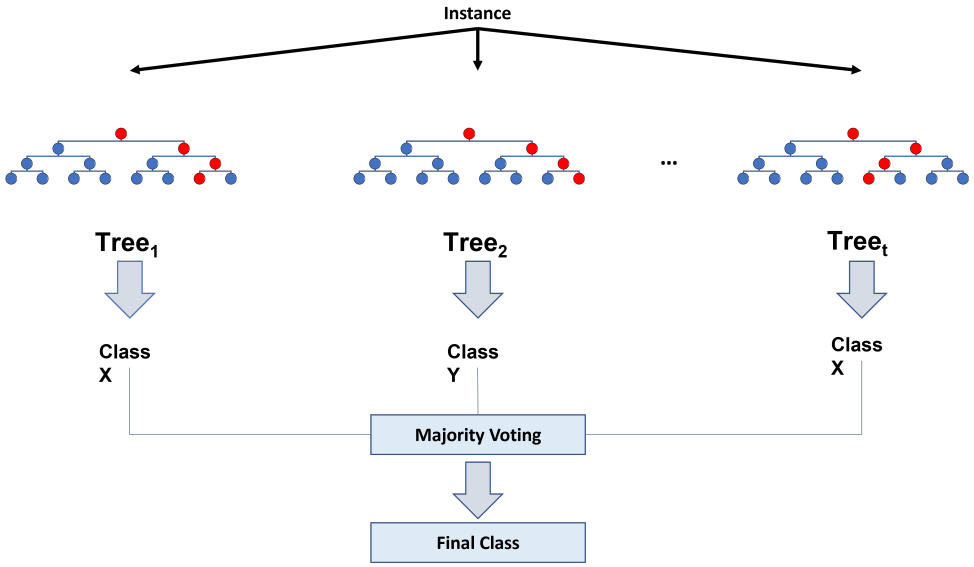


Figure 3.2: A simplified view of the Random Forest classifier.

ods [STC04] or Ensemble Methods [Bre01, CG16]). Note that this step, called feature engineering, is not always present, e.g., if the original data are already expressive enough or formatted to allow the direct application of shallow models. Despite this, the feature engineering step remains necessary for improving the performance of a shallow model. In many cases, the feature engineering step is fundamental since we cannot directly feed the raw data (e.g., the time series) into classical shallow models for two reasons [DP07]. The first one is that time series are of different lengths. The second is that simply feeding raw data as input to the faced problem will never work. Performing a feature engineering step allows overcoming these limitations following state-of-the-art approaches proposed in [ROOS$^+$16, CDSFS19, DVO$^+$20, RBHP21]. These studies adopt classical signal processing techniques to extract a vector of representation from the time series. This vector is composed of variables such as the mean, the median, and the signal magnitude area for both the time and frequency domains for a total of d features ($\mathcal{X}' \subseteq \mathbb{R}^d$).

Finally, at the end of this feature engineering step, it is possible to apply a series of state-of-the-art top-performing classification algorithms [1] [FDCBA14, WAF16]: Linear and Nonlinear (Gaussian Kernel) Support Vector Machines [STC04] (respectively LSVM and KSVM), Random Forest (RF) [Bre01], and XGBoost [CG16].

3.2.1 Random Forest

A powerful algorithm, both in terms of theoretical properties and practical effectiveness [FDCBA14, WAF16], for classification is RF developed in [Bre01] for the first time. For a complete understanding of RF, we need to recall how a binary Decision Tree (DT) [RM08] is defined and constructed. A binary DT for classification is a recursive binary tree structure in which a node represents a check on a particular feature; each branch defines the outcome of the check, and the leaf nodes represent the final classification. A particular path of exploration from the tree's root to one of its leaves represents a classification rule. Based on a recursive schema, a DT grows until it reaches a desired depth n_d. Each node of the DT (both root and nodes) is constructed by choosing the features and the check that most effectively separates the data satisfying the partial rule into two subsets based on the information gain (or other metrics like classification accuracy). Given this definition of DT, it is then possible to understand RF and the learning phase of each of the n_t DT which compose the forest. From $\mathcal{D}_n$, a bootstrap sample (sample with replacement) $\mathcal{D}'$ of n_b is extracted. Then a DT is learned based on $\mathcal{D}'$, but the best check/cut is selected among a subset of n_v features over the possible n_f features randomly chosen at each node. n_d is set to infinite; i.e., the DT is grown until every sample of $\mathcal{D}'$ is correctly classified. In the forward phase, i.e., when a previously unseen X needs to be labelled, each DT composing the RF is exploited to classify X; the final classification is taken with the majority vote. Note that n_b, n_v, n_d, and n_t are the hyperparameters of the RF. If $n_b = n$, $n_v = \sqrt{d}$, and $n_d = \infty$ we obtain the original RF formulation [Bre01], where n_t is usually chosen to trade-off accuracy and efficiency [OOA16] since the larger it is the better.

Therefore, 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 by increasing the number of trees n_t we set it to 1000 as a reasonably large number yet computationally tractable. For the problem of tuning the hyperparameters and assessing the performance of the final model, please refer to Section 3.4. Figure 3.2 shows a simplified view of Random Forest where the possible decision paths of the algorithm are shown in red.

3.2.2 SVM

One of the most influential approaches to supervised learning is the support vector machine [CV95, PS20, STC04]. This model is similar to logistic regression as it is driven by a linear function $w^T x + b$. The SVM predicts the positive class when $w^T x + b$ is positive and vice versa, the negative class. Figure 3.3 shows a simplified view of the algorithm. The major innovation associated with support vector machines is the kernel trick, i.e., many ML algorithms can be written

[1]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.

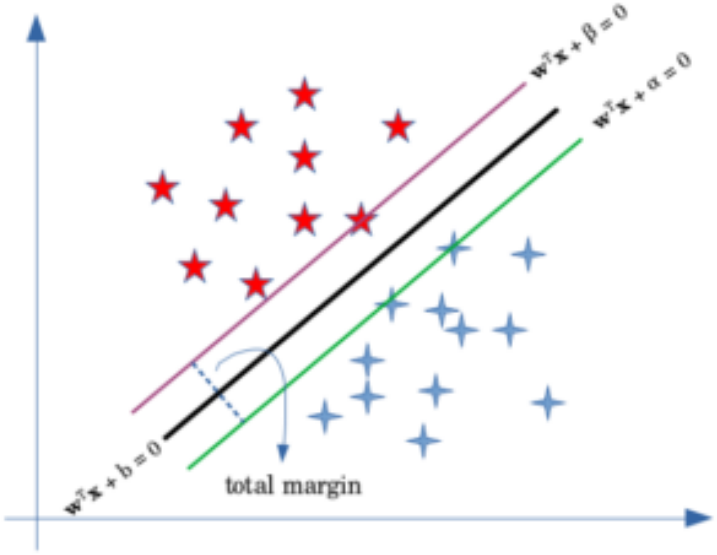


Figure 3.3: A simplified view of the Support Vector Machine classifier.

exclusively in terms of dot products between examples. For example, it can be shown that the linear function used by the support vector machine can be re-written as

$$w^T x + b = b + \sum_{i=1}^{m} \alpha_i x^T x^{(i)} \tag{3.5}$$

where $x^{(i)}$ is a training example and α is a vector of coefficients. Re-writing the learning algorithm this way allows us to replace x by the output of a given feature function $f(x)$ and the dot product with a function $k(x, x^{(i)}) = \phi(x) \cdot \phi(x^{(i)})$ called a kernel. The $\cdot$ operator represents an inner product analogous to $\phi(x)^T \phi(x^{(i)})$. For some feature spaces, we may not use the vector inner product. In some infinite dimensional spaces, we need to use other kinds of inner products, for example, inner products based on integration rather than a summation. After replacing dot products with kernel evaluations, we can make predictions using the function

$$f(x) = b + \sum_{i} \alpha_i k(x, x^{(i)}). \tag{3.6}$$

This function is nonlinear for x, but the relationship between $\phi(x)$ and $f(x)$ is linear, as well the relationship between α and $f(x)$ is linear. The kernel-based function is exactly equivalent to preprocessing the data by applying $\phi(x)$ to all inputs and then learning a linear model in the newly transformed space. The kernel trick is powerful for two reasons:

1. it allows us to learn models that are nonlinear as a function of x using convex optimization techniques that are guaranteed to converge efficiently. This is possible thanks to $\phi(x)$ being fixed and only α is optimised.

2. the kernel function k often admits an implementation that is significantly more computationally efficient than naively constructing two $\phi(x)$ vectors and explicitly taking their dot product.

The most commonly used kernel is the Gaussian kernel

$$k(u, v) = \mathcal{N}(u - v; 0, \sigma^2 I) \tag{3.7}$$

where $\mathcal{N}(x; \mu, \Sigma)$ is the standard normal density. This kernel is also known as the radial basis function (RBF) kernel because its value decreases along lines in v space radiating outward from u. The Gaussian kernel corresponds to a dot product in an infinite-dimensional space. We can think of the Gaussian kernel as performing a kind of template matching. A training example x associated with training label y becomes a template for class y. When a test point x' is near x according to Euclidean distance, the Gaussian kernel has a large response, indicating high similarity. The model then puts a large weight on the associated training label y. Overall, the prediction will combine many such training labels weighted by the similarity of the corresponding training examples.

Linear and Non-linear SVMs (LSVM and KSVM) have two main hyperparameters that must be tuned: the regularisation hyperparameter C, while KSVM has both C and the kernel coefficient γ. For the problem of tuning the hyperparameters and assessing the performance of the final model, please refer to Section 3.4.

3.2.3 XGBoost

The eXtreme Gradient Boosting (XGBoost) model [CG16] uses a gradient boosting framework [Fri01] and is also a decision-tree-based ensemble method. Boosting techniques are based on improving a single weak model by combining it with many other weak models to generate a collectively strong one. Gradient boosting represents an extension of this concept where the process of additively generating weak models is formalised as a gradient descent algorithm over an objective function. Gradient Boosting minimises errors for the next model by using the error gradient (hence the name gradient boosting) to optimise the prediction. Figure 3.4 shows a simplified view of XGBoost where the possible decision paths of the algorithm are shown in red.

Both RF and XGBoost build a model consisting of multiple decision trees. The difference is in how the trees are built and combined. Indeed, RF uses the bagging technique to build full decision trees in parallel from random bootstrap samples of the dataset. The final prediction is an average of all of the decision tree predictions. On the other hand, XGboost iteratively trains a set of shallow decision trees where each iteration uses the error residuals of the previous model to fit the next model. The final prediction is a weighted sum of all tree predictions. The bagging technique used by RF minimises variance and overfitting, while the boosting technique is used by XGBoost bias and underfitting.

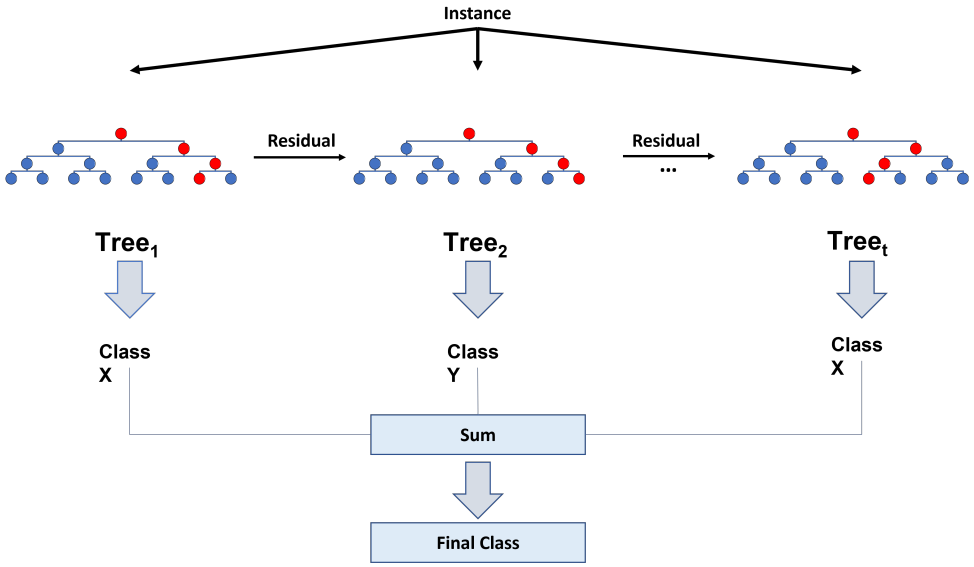


Figure 3.4: A simplified view of the eXtreme Gradient Boosting classifier.

Moreover, contrarily to RF, XGBoost does not optimise each tree independently but optimises the entire ensemble to trade off accuracy (i.e., the error on the data) and complexity of the ensemble, measured with different criteria like the depth of each tree n_d or the minimum loss reduction required to make a further partition on a leaf node of the tree m_l. This optimisation occurs using gradient descent that implies the correct learning rate l_r selection.

In XGBoost we need to tune the learning rate of the gradient l_r, the max dept of 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 features to randomly sample from the whole featured during each node of each tree creation n_f. For the problem of tuning the hyperparameters and assessing the performance of the final model, please refer to Section 3.4.

3.2.4 Feature Engineering

This section describes how the features can be extracted and engineered from the raw data to improve the shallow model recognition performance. Note that this step is not mandatory but often recommended. Feature Engineering allows obtaining aggregates for the time series that compose the dataset, as visible in Figure 3.5. These aggregates can better explain the phenomena under investigation and are usually chosen with one segmentation criteria. Segmentation criteria are manifold but can be identified into two groups: time/space-dependent or event-dependent. The first groups of segmentation criteria use fixed-width sliding windows that can observe the behaviour of a signal over time or space. Although this approach is straightforward, it can pro-

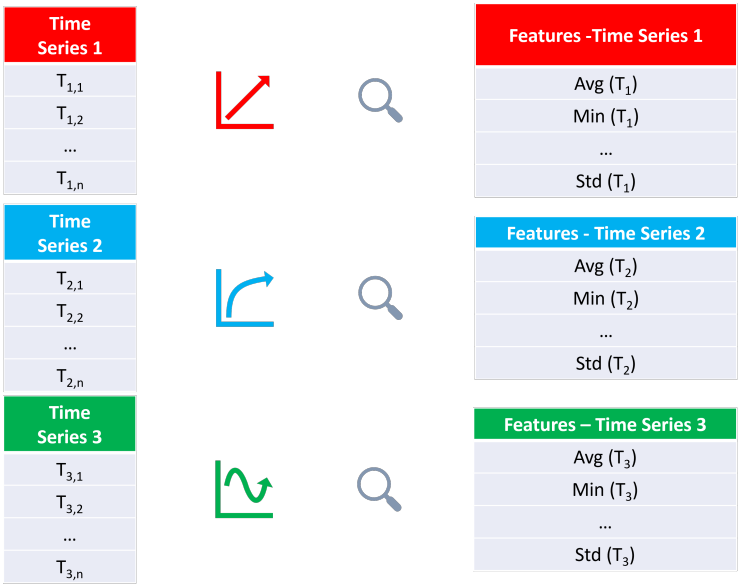


Figure 3.5: A simplified view of the Feature Engineering phase.

duce high recognition performance when applied, and it belongs to the state-of-the-art for many years in computer vision applications. The second groups of segmentation criteria use events to observe the behaviour of a signal according to some events. This approach implies good knowledge of the problem faced to identify the best events that can be used to segment the signals. For example, the analysis of criminals at airports takes place on events that people do not habitually perform (e.g. leaving a suitcase unattended). However, these two ways of segmenting signals solve the problem in a mirror-image manner. The first time/space-based group uses a bottom-up approach, in which one starts from the low-level signal to study a given situation. The second event-dependent approach is dual to the first in that it uses a top-down approach based on high-level information.

In human movement applications, a typical approach is to start from the time series describing the recorded movement and sample them with fixed-width sliding windows. These windows slide over the different signals and provide information on what is happening in a given period. Windows are usually applied with an overlapping approach to consider the preceding and the following information. For example, if we want to automatically detect people's actions, we are interested in studying the behaviour of their joints over time. Let us assume that the duration of this action is 10 seconds. By choosing an overlapping of 50%, the windows will slide over 3 different intervals, i.e., in seconds $\{0 \dots 5\}$, seconds $\{2.5 \dots 7.5\}$ and $\{5 \dots 10\}$. Note that this heuristic has already been successfully employed in many works in the literature [ROOS+16, CDSFS19, DVO+20, RBHP21], where statistical measures were able to describe human actions. In these works, measurements such as the mean, signal-pair correlation and signal magnitude area in the time and frequency domains are extracted, as observable in Table 3.1. The Fast

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	Interquantile range
entropy	Signal Entropy
correlation	Correlation coefficient
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 3.1: List of measures for computing feature vectors.

Fourier Transform was employed to find the frequency components for each window. A new set of features, including the energy of the different frequency bands, skewness and frequency kurtosis, were also used to improve learning performance. Table 3.1 contains the list of all the measures applied to the time and frequency domain signals in this thesis works.

3.2.5 Feature Ranking

It is not always sufficient to build models but also to understand how these models exploit, combine, and extract information. This operation allows us to understand if the learning process has a cognitive meaning, or in other words, it can capture the underline phenomena and not just spurious correlations [GE03,CL17]. This process happens by comparing the knowledge of the experts with the information learned by the models. One way to reach this goal is to perform the Feature Ranking (FR) phase, which allows the detection of the importance of those features. Those features are known to be relevant from a physical perspective and are also taken into account appropriately, i.e., ranked as highly important by the learned models. The failure of the learned model to properly account for the features, which are relevant from a cognitive point of view,

might indicate poor quality in the measurements, poor learning ability of the model, or spurious correlations. FR, therefore, represents a fundamental phase of model checking and verification since it should generate results consistent with the available knowledge of the phenomena under exam provided by the experts.

In the literature, several measures and approaches are available for FR techniques. One of the most effective relies on the Permutation Test combined with the Mean Decrease in Accuracy (MDA) metric, where the importance of each feature is estimated by removing the association between the input (i.e., the raw time series or the extracted feature) and the outcome of the model. For this purpose, the values of the features are randomly permuted [Goo13], and the resulting increase in error is measured. In this way, we can remove the influence of the correlated features. This technique has been applied mainly to RF [SAVdP08, GPTM10] but can easily be applied to other ML models. The key idea of the Permutation Test combined with the MDA is the evaluation of two quantities for each DT: the error on the out-of-bag samples, which are used during prediction, and the error computed on the out-of-bag ones after permuting the input values. Finally, the difference between these two values is averaged over the different trees in the ensemble, and this quantity represents the raw importance score for the input variable under the exam.

It is remarkable to highlight that this technique can be applied to raw time series and extracted features. The main difference in these cases lies in the level of granularity. When applied to time series, much more information is removed from the input data since all features derived from that particular time series are also removed. On the other hand, when applied to extracted features, one does not remove all the input of a time series, only the spurious or non-informative ones.

3.3 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) [ZLX+16]. 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.

3.3.1 Long-Short-Term Memory Network

To date, one of the most effective sequence models used in practical applications is called gated RNNs, which include models such as Long Short-Term Memory network (LSTM) [HS97] and Gated Recurrent Unit (GRU) [CGCB14]. Gated RNNs rely on the idea of creating paths through time that have derivatives that neither vanish nor explode [GBC16]. These architectures generalise this to connection weights, allowing the network to process information over a long duration. However, not all this information is relevant and it might be more useful for the network to forget the old state. A practical example is represented by a sequence that present sub-sequences. In this case, if we want to accumulate evidence inside each sub-sequence, we need a mechanism to forget the old state by setting it to zero. Gated RNNs allow performing this operation automatically.

Introducing self-loops to produce pathways where the gradient can flow for long periods is relevant to the initial short-term memory model (LSTM) [HS97]. A key addition was to make the weight of this self-loop context-dependent rather than fixed [GSC00]. By making the weight of this auto-loop gated (controlled by another hidden unit), the integration time scale can be changed dynamically. The LSTM block diagram is reported in Figure 3.6. Instead of a unit that

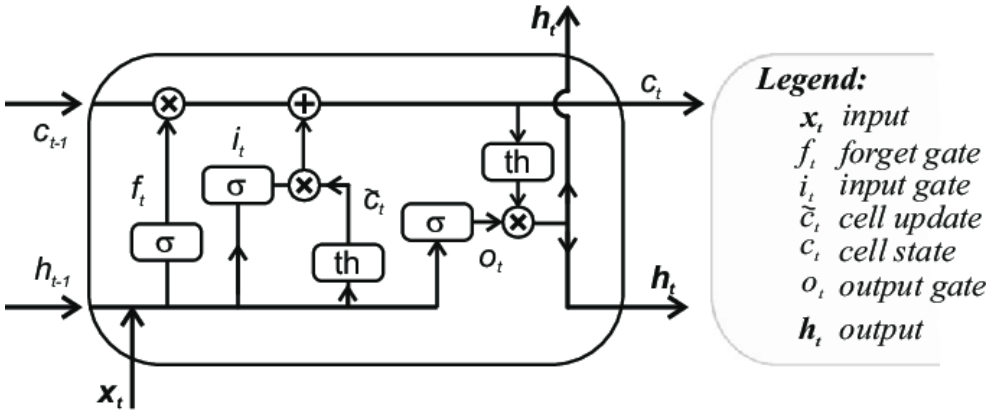


Figure 3.6: The LSTM cell in detail.

simply applies an elementary nonlinearity to the affine transformation of inputs and recurrent units, LSTM recurrent networks have "LSTM cells" that have an internal recurrence (a self-loop), in addition to the outer recurrence of the RNN. Each cell has the same inputs and outputs as a traditional recurrent network but has more parameters and a system of control units that con-

trol the information flow. The most important component is the state unit $s_i^{(t)}$ which has a linear self-loop. However, in this case, the weight of the self-loop (or the associated time constant) is controlled by a leaky gate unit $f_i^{(t)}$ (for time step t and cell i), which sets this weight to a value between 0 and 1 via a sigmoid unit:

$$f_i^{(t)} = \sigma\left(b_i^f + \sum_j U_{i,j}^f x_j^{(t)} + \sum_j W_{i,j}^f h_j^{(t-1)}\right), \tag{3.8}$$

where $x(t)$ is the current input vector and $h^{(t)}$ is the currently hidden layer vector, containing the outputs of all the LSTM cells, and b^f, U^f, and W^f are respectively biases, input weights and recurrent weights for the forget gates. The LSTM cell internal state is thus updated as follows, but with a conditional self-loop weight $f_i^{(t)}$:

$$s_i^{(t)} = f_i^{(t)} s_i^{(t-1)} + g_i^{(t)} \sigma\left(b_i + \sum_j U_{i,j} x_j^{(t)} + \sum_j W_{i,j} h_j^{(t-1)}\right), \tag{3.9}$$

where b, U and W respectively denote the biases, input weights and recurrent weights into the LSTM cell. The external input gate unit $g_i^{(t)}$ is computed similarly to the forget gate (with a sigmoid unit to obtain a gating value between 0 and 1), but with its parameters:

$$g_i^{(t)} = \sigma\left(b_i^g + \sum_j U_{i,j}^g x_j^{(t)} + \sum_j W_{i,j}^g h_j^{(t-1)}\right), \tag{3.10}$$

The output $h_i^{(t)}$ of the LSTM cell can also be switched off via the output gate $q_i^{(t)}$, which also uses a sigmoid unit for gating:

$$h_i^{(t)} = \tanh(s_i^{(t)}) q_i^{(t)} \tag{3.11}$$

$$q_i^{(t)} = \sigma\left(b_i^\circ + \sum_j U_{i,j}^\circ x_j^{(t)} + \sum_j W_{i,j}^\circ h_j^{(t-1)}\right), \tag{3.12}$$

which has the parameters b°, U°, and W° for its biases, input weights and recurrent weights, respectively. Among the variants, one can choose to use the state of cell $s_i^{(t)}$ as an additional input (with its weight) in the three ports of the i-th unit, as shown in Figure 3.6. This would require three additional parameters. LSTM networks have been shown to learn long-term dependencies more easily than simple recurrent architectures, first on artificial datasets designed to test the ability to learn long-term dependencies [BSF94, HS97, HBF+01].

As classical (LSTM) and bidirectional (BLSTM) LSTM networks, we rely on a standard architectures [GBC16, LBRF20, JTAW20] 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 [KB14] empowered with a one-cycle learning rate [HHM$^+$21] 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 Tables 5.11 and 5.16). Note that in this case, the hyperparameters configuration space is much larger than the shallow models. For the problem of tuning the hyperparameters and assessing the performance of the final model, please refer to Section 3.4.

Unfortunately, as we will discuss in Section 6, 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.

3.3.2 Temporal Convolutional Network

Convolutional networks [L$^+$89] are a specialised type of neural network for processing topological grid data (i.e., a 1D grid for time series or a 2D grid for images). These architectures are known as Convolutional Neural Networks (CNN) or Temporal Convolutional Networks (TCN) when referring to time series problems. The name comes from the mathematical operation they use, namely convolution, a linear operation. Convolutional networks have been tremendously successful in practical applications. Research into convolutional network architectures proceeds so rapidly that a new best architecture for a given benchmark is announced every few weeks to months, rendering it impractical to describe the best architecture in print. However, the best architectures have consistently been composed of the building blocks described here.

The convolution operation leverages three main properties: sparse interactions, parameter sharing and equivariant representations. A practical benefit of convolution is that it can work with inputs of variable sizes. Instead of the classical multiplication of matrices, typical of traditional neural networks, convolutional networks have sparse interactions. This is possible due to the fact that the kernel is smaller than the input. For this reason, we can store fewer parameters, which has several benefits: (1) reduces the memory used by the model, (2) improves the efficiency, and (3) fewer operations are needed. In a deep convolutional network, units in the deeper layers may indirectly interact with a larger portion of the input, allowing the network to efficiently describe interactions between many variables by constructing such interactions from simple building blocks capable of describing sparse interactions.

The second property, i.e., parameter sharing, refers to using the same parameter for more than one function in a model. Instead of using each element of the weight matrix only once, as in traditional neural networks, convolutional networks reuse the kernel at each input position.

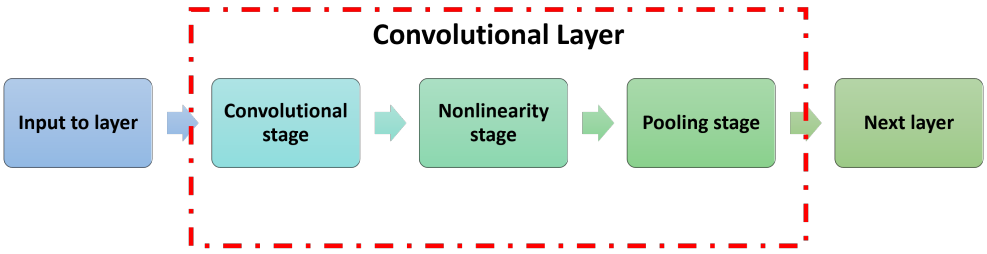
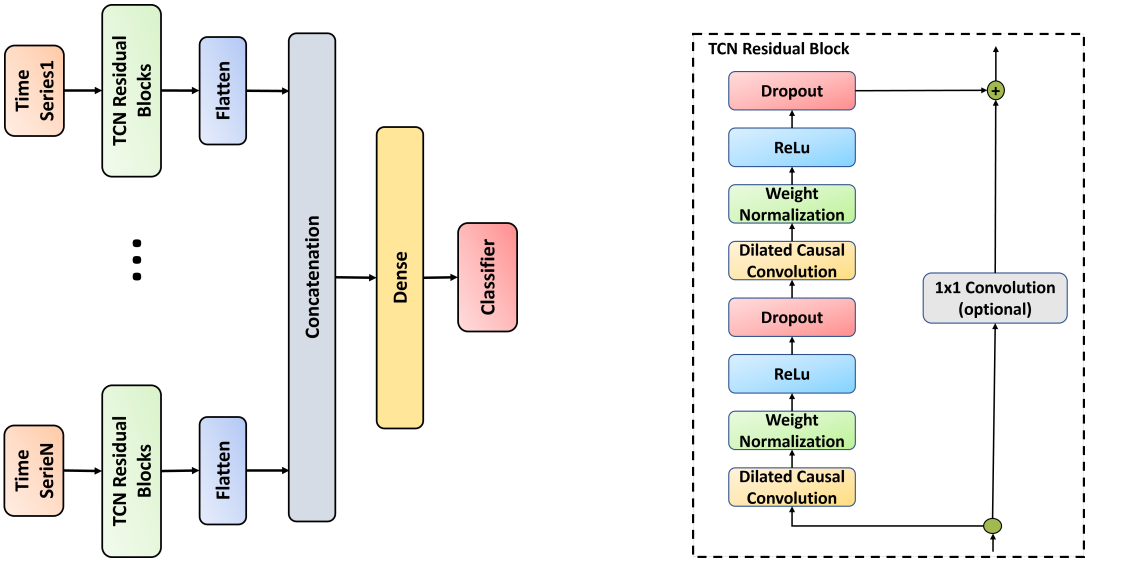


Figure 3.7: An example of Convolutional Network stages.

Therefore, convolution is dramatically more efficient than dense matrix multiplication, used in traditional neural networks, for memory requirements and statistical efficiency. The parameter sharing property causes the third property of convolution operation, i.e., the equivariance to translation. When processing time series data, convolution produces a kind of timeline showing when different features appear in the input. If we shift an event in time in the input, the same representation of it will appear in the output, just shifted in time.

Usually, a convolutional network consists of three stages, as shown in Figure 3.7. In the first stage, several convolutions are performed in parallel, producing a set of linear activations. Then, in the second stage, a nonlinear function is used on each linear activation. Finally, in the third stage, a pooling function is used to modify the layer output further. The pooling function replaces the architecture output at a certain location with a summary statistic of the neighbourhood.

To overcome LSTM limitations, we decided to substitute the LSTM blocks with Temporal Convolutional Network (TCN) residual blocks [BKK18, LYC17] that can focus on multiple temporal scales for each raw time series independently. The proposed architecture is reported in Figure 3.8. The peculiarities of the proposed deep multiple temporal scale architectures based on TCN are mainly three: (i) the convolutions in the architecture are causal; namely, there is no information leakage from future to past; (ii) the architecture can handle different sequences lengths and map it to an output sequence of the same length like the LSTMs; and (iii) can handle effectively long history. For what concerns (i) the TCN uses causal convolutions. For what concerns (ii), it is due to the use of a 1D fully-convolutional network model where each hidden layer has the same length as the input layer; zero padding of length $(\text{kernel size} - 1)$ is added to preserve the previous length. As for the (iii), we employed dilated convolution that enables a large receptive field [YK15] without employing too deep TCN residual blocks. The network has been trained, like the LSTM-based architectures, with the ADAM optimiser empowered with a 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



(a) High-level representation of the architecture. (b) TCN-based layer architecture details.

Figure 3.8: The proposed Deep Multi-Scale Models architecture based on TCN.

L2 regularisation on the weight of the entire network C (see Tables 5.11 and 5.16). Note that, in this case, the configuration space of the hyperparameters explodes even further than LSTMs, as, in practice, we have possibly different configurations of the kernel size for each time series.

For this reason, as we will see in Section 5.4, to reduce this hyperparameter configuration space and contemporary the 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 ones. It is reasonable to assume that many of these series contain redundant information. 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 the permutation importance with a 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 contributed according to the mean decrease of accuracy. Due to this reduction in the number of input time series, LSTM- and TCN-based architectures strongly reduce the number of weights to tune and the hyperparameters configuration search space.

For the problem of tuning the hyperparameters and assessing the performance of the final model, please refer to Section 3.4.

3.4 Model Selection & Error Estimation

The main challenge in ML is that we must perform well on new, previously unseen data, i.e., not just those used for training our model. This ability is called generalisation. When we train an ML model, we have access to a training set that we can use to measure the training error (i.e., the measure we want to reduce). This represents an optimisation problem. However, with the ML models, we also want the generalisation error, also known as the test error, to be low. The generalisation error is defined as the expected value of the error on new input. In this case, the expectation is taken on several possible inputs, drawn from the distribution of inputs we expect the system to encounter in practice. Typically, the generalisation error of a machine learning model is estimated by measuring its performance on a set of test examples collected separately from the training set.

Model Selection (MS) and Error Estimation (EE) face and address the problem of tuning and assessing the performance of a learning algorithm [One19]. Resampling techniques are commonly used by researchers and practitioners since they work well in most situations and this is why we will exploit them in this work. Other alternatives exist, based on the Statistical Learning Theory, but they tend to underperform resampling techniques in practice [One19]. Resampling techniques leverage on a simple idea: $\mathcal{D}_n$ is resampled many (n_r) times, with or without replacement, and three independent datasets called learning, validation and test sets, respectively $\mathcal{L}_l^r$, $\mathcal{V}_v^r$, and $\mathcal{T}_t^r$, with $r \in \{1, \cdots, n_r\}$ are defined. Note that

$$\mathcal{L}_l^r \cap \mathcal{V}_v^r = \oslash, \quad \mathcal{L}_l^r \cap \mathcal{T}_t^r = \oslash, \quad \mathcal{V}_v^r \cap \mathcal{T}_t^r = \oslash, \tag{3.13}$$

and

$$\mathcal{L}_l^r \cup \mathcal{V}_v^r \cup \mathcal{T}_t^r = \mathcal{D}_n \text{ for all } r \in \{1, \cdots, n_r\}. \tag{3.14}$$

Then, to select the optimal configuration of hyperparameters $\mathcal{H}$ of the algorithm $\mathscr{A}_{\mathcal{H}}$ in a set of possible ones $\mathfrak{H} = \{\mathcal{H}_1, \mathcal{H}_2, \cdots\}$, namely to perform the MS phase, the following procedure has to be applied:

$$\mathcal{H}^* : \quad \arg\min_{\mathcal{H} \in \mathfrak{H}} \sum_{r=1}^{n_r} M(\mathscr{A}_{\mathcal{H}}(\mathcal{L}_l^r), \mathcal{V}_v^r), \tag{3.15}$$

where $\mathscr{A}_{\mathcal{H}}(\mathcal{L}_l^r)$ is a model learned by $\mathscr{A}$ with the hyperparameters $\mathcal{H}$ based on the the data in $\mathcal{L}_l^r$ and where $M(f, \mathcal{V}_v^r)$ is a desired metric. Since the data in $\mathcal{L}_l^r$ are independent of the ones in $\mathcal{V}_v^r$, the intuition is that $\mathcal{H}^*$ should be the configuration of hyperparameters which allows achieving optimal performance, according to the desired metric, on a set of data that is independent, namely previously unseen, concerning the training set.

Then, to evaluate the performance of the optimal model, namely the model learned with the optimal hyperparameters based on the available data, which is

$$f_{\mathscr{A}}^* = \mathscr{A}_{\mathcal{H}^*}(\mathcal{D}_n) \tag{3.16}$$

or, in other words, to perform the Error Estimation (EE) phase, the following procedure has to be applied:

$$M(f^*_{\mathscr{A}}) = \frac{1}{n_r}\sum_{r=1}^{n_r} M(\mathscr{A}_{\mathcal{H}^*}(\mathcal{L}_l^r \cup \mathcal{V}_v^r), \mathcal{T}_t^r). \tag{3.17}$$

Since the data in $\mathcal{L}_l^r \cup \mathcal{V}_v^r$ are independent from the ones in $\mathcal{T}_t^r$, $M(\mathscr{A}_{\mathcal{H}^*}(\mathcal{L}_l^r \cup \mathcal{V}_v^r)$ will be an unbiased estimator of the true performance of the final model [One19].

In this thesis, the complete k-fold cross-validation is exploited [Koh95, One19] since, together with bootstrap, represent a state-of-the-art approach to the problem of MS and EE. Then we need to set

$$n_r \leq \binom{n}{k}\binom{n-\frac{n}{k}}{k}, \quad l = (k-2)\tfrac{n}{k}, \quad v = \tfrac{n}{k}, \text{ and } t = \tfrac{n}{k} \tag{3.18}$$

and the resampling must be done without replacement [Koh95].

In this thesis, we will observe different resampling strategies. We will describe them in Chapter4 as they are designed to study the ability and robustness of the algorithm to extract information from data. Furthermore, these strategies will derive directly from different hierarchies in the analysed datasets. The learning algorithm will be tested on the amount of information accessible to it for an intuition of its operation in a real-world case study.

3.5 Metrics

For what concerns the metrics $M(f)$ exploited for evaluating the performance of a model f learned from the data based on the methods described above, we have to recall that many different metrics are available in literature [Agg15]. In this work, we will report just the most common ones. To define them, let us first consider a subset of the available data $\mathcal{T}_t$, also called test set, coming from μ, but different from $\mathcal{D}_n$ since the data that have been used to learn f should be different to the ones exploited to evaluate its performance so to avoid overfitting [One19]. Let us define the elements in the confusion matrix, the True Positive

$$\text{TP}(f) = \sum_{(X,Y)\in\mathcal{T}_m : Y=1} \mathbb{1}\{f(X) = 1\}, \tag{3.19}$$

the True Negative

$$\text{TN}(f) = \sum_{(X,Y)\in\mathcal{T}_m : Y=0} \mathbb{1}\{f(X) = 0\}, \tag{3.20}$$

the False Positive

$$\text{FP}(f) = \sum_{(X,Y)\in\mathcal{T}_m : Y=0} \mathbb{1}\{f(X) = 1\}, \tag{3.21}$$

and the False Negative

$$\text{FN}(f) = \sum_{(X,Y)\in\mathcal{T}_m:Y=1} \mathbb{1}\{f(X) = 0\}. \tag{3.22}$$

Then we can also define accuracy as

$$\text{accuracy}(f) = \frac{\text{TP}(f) + \text{TN}(f)}{\text{TP}(f) + \text{FN}(f) + \text{TN}(f) + \text{FP}(f)}, \tag{3.23}$$

the balanced accuracy as

$$\text{balanced accuracy}(f) = \frac{\frac{TP(f)}{TP(f)+FN(f)} + \frac{TN(f)}{TN(f)+FP(f)}}{2}, \tag{3.24}$$

the precision as

$$\text{precision}(f) = \frac{TP(f)}{TP(f) + FP(f)}, \tag{3.25}$$

the recall as

$$\text{recall}(f) = \frac{TP(f)}{TP(f) + FN(f)}, \tag{3.26}$$

the F1 score

$$\text{F1 score}(f) = 2 * \frac{\text{precision}(f) * \text{recall}(f)}{\text{precision}(f) + \text{recall}(f)}, \tag{3.27}$$

the Matthews correlation coefficient

$$\text{MCC}(f) = \frac{TP * TN - FP * FN}{\sqrt{(TP + FP) * (TP + FN) * (TN + FP) * (TN + FN)}}, \tag{3.28}$$

and the Area Under the Receiver Operating Characteristic Curve (ROC-AUC), which is the area under the $TP(f)$ rate against the $FP(f)$ rate curve.

Chapter 4

Applications & Data

4.1 TELMI Dataset

The current section will present the dataset explored in the analysis of the papers:

1. *D'Amato, Vincenzo, et al. "Understanding violin players' skill level based on motion capture: a data-driven perspective." Cognitive Computation 12.6 (2020): 1356-1369*;
2. *D'Amato, Vincenzo, et al. "Accuracy and intrusiveness in data-driven violin players skill levels prediction: Mocap against myo against kinect." International Work-Conference on Artificial Neural Networks. Springer, Cham, 2021.*

In this study, we employed the data collected during the H2020 ICT-TELMI Project[1]. The purposes of the project were essentially twofold: the first was to understand how people learn to play the violin, and the second was to design technologies that would support the learning phases. Royal College of Music of London (RCM) suggested collecting data regarding a series of typical exercises performed during the learning path of classical violin conservatoire programs. The pedagogical materials were divided into three groups:

- posture exercises for beginners, including how to handle the instrument, bow techniques, and fingering;
- techniques studies, including vibrato and different articulation exercises;
- repertoire pieces as an example of expressive performances.

The use of both custom and preexisting exercises was deliberate and selected from different sources:

[1] `http://telmi.upf.edu/`

Source		Mocap	MYO	Kinect
Skill Level	**n° of violinists**	**n° of exercises**	**n° of exercises**	**n° of exercises**
Experts	3	32	29	31
Beginners	2	20	12	12
Total	5	52	41	43

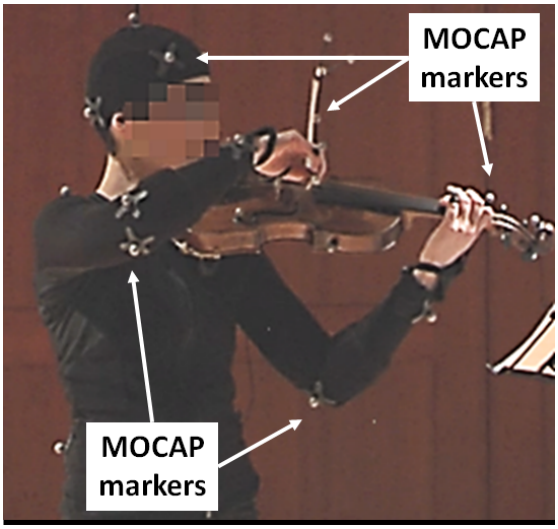
Table 4.1: Information about the collected data: players, exercises, and data sources.

- some were taken from the standard published catalogue of exercises, e.g., Schradieck, Ševčík, and Kreutzer;
- other ones are sourced or adapted from the Associated Board of the Royal Schools of Music (ABRSM) examination syllabus;
- the last ones were developed by M. Mitchell, a high-level performer and teacher from RCM, to address specific techniques and focus on the capabilities offered by non-notated feedback (e.g. bowing exercises).

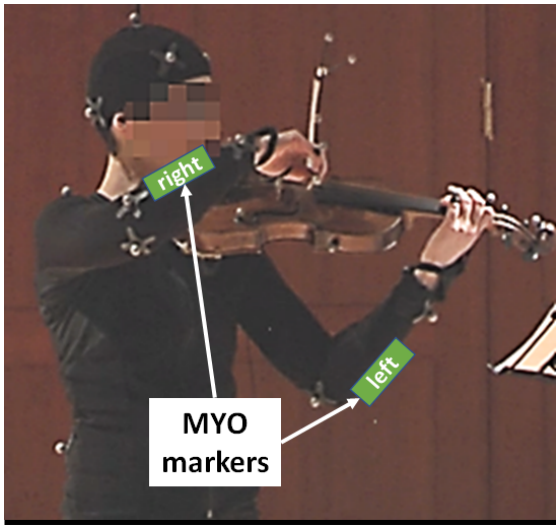
All the recordings took place at the Casa Paganini InfoMus research centre of the University of Genova[2]. We collected data about 5 violinists (3 experts selected by the RCM and 2 beginners) playing 5 bow-violin techniques such as left-hand articulation, bowing techniques and repertoire pieces. Musicians' performance movement-related data were collected using three different sources: Mocap, MYO sensors, and Kinect. In addition, since Mocap allows the extension of markers, we tracked the bow and violin to enrich the details of the players' movements. The information about the collected data (players, exercises, and data sources) is summarized in Table 4.1. Table 4.1 shows that for Mocap more exercises are available (since at the beginning of the data collection MYO and KINNECT were not available), while the number of players is the same across the different sources. Note also, from Table 4.1 more exercises are available for experts than for beginners. Regarding the Mocap data, exploiting the experience of the expert players and the teachers of the RCM, we extracted 14 low-level features starting from the Mocap skeleton data using the EyesWeb XMI platform[3] [CCVG07]. According to the literature, these features potentially comprehensively describe the movements of violinists and then are necessary to study their skill level. These are the 14 low-level features: mean Shoulders' velocity, shoulder low back asymmetry, upper body kinetic energy, left/right shoulder height, bow-violin incidence, distance low/middle/upper bow-violin, hand-violin incidence, left/right head inclination, and left/right wrist roundness. Note that Mocap requires markers to be placed on both the players and the violin, thus intruding on their habits (see Figure 4.1a). Physiological data were collected using 2 MYO sensors located on both forearms of each musician, as depicted in Figure 4.1b. A MYO device is equipped with eight electromyographic (EMG) sensors that measure muscle tension and an inertial measurement unit (IMU) with a triaxial accelerometer, gyroscope

[2] www.casapaganini.org

[3] http://www.infomus.org/eyesweb_eng.php



(a) Mocap sensors on a player.



(b) MYO sensors on a player.

Figure 4.1: Mocap and MYO sensors on a player.

and magnetometer. The accelerometer component measures linear accelerations, the gyroscopes measure angular accelerations, and the magnetometer measures magnetic fields. The magnetometer was exploited in conjunction with the accelerometer and gyroscope data to determine the absolute heading. The Kinect, like the Mocap, can reconstruct a person's skeleton. However, bow and violin positions cannot be tracked using this device and then we miss 4 features concerning the Mocap data: the feature regarding bow-violin incidence and the 3 concerning bow-violin distances. Note that Kinect is the most unobtrusive technology, being also the cheapest since it can work in the wild with no device to be placed on the player.

In this experiment, our goal was to understand if it was possible to use Mocap, Kinect, or MYO data produced by players to label it automatically as an expert or beginner. Moreover, we would like to understand the trade-off between accuracy and intrusiveness.

4.2 EmoPain-weDRAW-Unige-Maastricht Dance

The current section will present the dataset explored in the analysis of the paper:

1. *D'Amato, Vincenzo, et al. "Keep it Simple: Handcrafting Feature and Tuning Random Forests and XGBoost to face the Affective Movement Recognition Challenge 2021." 2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW). IEEE, 2021.*

In this study, we employed the data made available in Affective Movement Recognition Challenge 2021, where the purpose was to investigate three different datasets. In Section 4.2.1 we will describe the EmoPain dataset [AKRP$^+$15] related to Task 1 "Protective Behaviour Detection based on Multimodal Body Movement Data". In Section 4.2.2 we will describe the weDraw dataset [ONJ$^+$20] related to Task 2 "Detection of Reflective Thinking based on Body Movement Data". Finally, in Section 4.2.3 we will describe the Unige-Maastricht Dance dataset [CEREZ$^+$16, NMP$^+$17, VAM$^+$19] related to Task 3 "Detection of Lightness and Fragility in Dance Movement based on Multimodal Data".

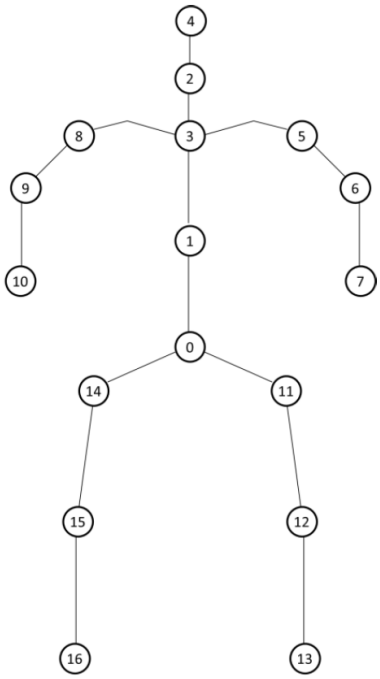
4.2.1 Task 1: EmoPain Dataset

The emoPain dataset consists of anonymised 3D full-body joint positions and concomitant back muscle activity data for 19 people with chronic low-back pain from the EmoPain dataset [AKRP$^+$15]. The data included the corresponding protective behaviour labels obtained from clinician observers [AKRP$^+$15] and the exercise type. The Challenge organisers provided training, validation, and test partitions which contain instances from 10, 4, and 5 people with chronic pain, respectively. Note that the test partition does not include the protective behaviour labels or the exercise type. More in detail, in this task, the available data comprised anonymised 3D full-body joint positions and 4 groups of muscle activity, both shown in Figure 4.2, which describe the activities of 19 different people with chronic low-back pain, from the EmoPain dataset [AKRP$^+$15]. The provided data, already segmented by the organisers, had a fixed length of 180-frame (i.e., 3 sec) for each participant. More in detail, training, validation and test sets include 5827, 1844 and 2744 windows from 10, 4 and 5 for people with chronic pain, respectively. Finally, each window provides a three-dimensional position for 17 joints (17 times 3 for a total of 51 features) and the upper envelope of rectified surface electromyography data from 4 muscle groups as shown in Figure 4.2.

In this task, the purpose was to build a model able to predict if chronic pain is present or absent in a given window.

4.2.2 Task 2: weDraw Dataset

WeDraw consists of anonymised 3D full-body joint positions, shown in Figure 4.2a, for 24 children from the weDraw-1 Movement dataset [ONJ$^+$20]. The data included the corresponding reflective thinking labels based on expert observer annotation [ONJ$^+$20] and the corresponding maths problem-solving activities. The Challenge organisers provided training, validation, and test partitions which contain instances from 13, 5 and 6 children, respectively. Note that the test partition does not include the reflective thinking labels or the exercise type. The provided data, already segmented by the organisers, had a fixed length of 5 sec for each child. More in detail, the



(a) Skeleton for Task 1 and Task 2.

(b) Electromyography for Task 1.

Figure 4.2: Type of data made available for Task 1 and Task 2 of the Affective Movement Recognition Challenge 2021.

Figure 4.3: An example of a dancer's performance in the Unige-Maastricht Dance dataset.

training, validation, and test sets for this task included 2090, 792 and 672 windows, respectively. In this task, the purpose was to build a model able to predict if reflective thinking is present or absent in a given window.

4.2.3 Task 3: Unige-Maastricht Dance Dataset

Unige-Maastricht provides accelerometer data captured from wrists, ankles, and waist, videos with faces blurred, and audio respiration data for 13 dancers [CEREZ$^+$16, NMP$^+$17, VAM$^+$19]. The data included corresponding labels for the dance type (lightness or fragility). These labels are based on both observer annotations (the organisers provided an excel file where 5 experts annotating the fragments) and on the neuroscientific experiment described in [VAM$^+$19]. An example of a dancer's performance is shown in Figure 4.3. We used only kinetic data acquired from accelerometers to have a similar approach to the previous two tasks in this analysis.

More in detail, for the majority of the 13 participants, we have 51 raw features collected using IMU [DVO$^+$21] and MYO [DVO$^+$21] sensors. For each dancer, data is present in segments of 10sec. In this task, the organisers did not provide a predefined division of training, validation and test sets. For this reason, we decided to use 7, 1, and 1 dancers for training, validation and test sets, respectively. In this task, the purpose was to build a model able to detect the fragility or the lightness in the dancer's performance.

4.3 Ellipsis Dataset

The current section will present the dataset explored in the analysis of the paper:

1. *D'Amato, Vincenzo, et al. "The Importance of Multiple Temporal Scales in Motion Recognition: from Shallow to Deep Multi Scale Models." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022.*

Figure 4.4: Example of data acquisition with the graphics tablet[4].

This study aimed to understand automatically the person who drew an ellipse, and it consisted of further exploitation of a previous experiment [SBCS15]. The authors of [SBCS15] used the two-thirds power law to study the perception of movement. In particular, the two-thirds power law can model the relation between velocity and curvature typical of human movement [VT82,BSLR18]: when the velocity decreases, the curvature increases and vice versa. The two-thirds power law can describe a variety of movement tasks and investigate the muscle districts involved in these tasks, including planar drawing movements [VT82, BSLR18] or the perception of movements [SBCS15, BSLR18]. These approaches rely on the fact that human movement is characterised by geometric and kinematic patterns that can be explained by a limited number of laws of motion. Unfortunately, the two-thirds power law is a too simple model to be exploited in practice. The definition of a richer model, capable of explaining the differences between one individual to another, is necessary. This challenge is difficult in real-world scenarios. We designed an experiment in a simplified scenario, inspired by the work of Scocchia et al. [SBCS15], who explore the different perceptions of individuals in observing a moving dot along an elliptical trajectory. We designed and collected data on different individuals who were all asked to draw an ellipse on a graphics tablet: the goal is to detect each person from the details of how s/he draws the ellipse.

We collected data using a graphics tablet[4], under an ordinary lighting condition and vertically positioned respect to the sitting participant (see Figure 4.4). We collected data about 14 right-

[4]Wacom Bamboo slate; temporal resolution: 200 samples/s; resolution: 1748 by 2551; Active area: 210 × 297 mm

handed subjects who had to draw several times an ellipse. We varied the hand (left and right) and the drawing speed (slow, medium, and fast according to the perceptual sensibility of the subject). The direction of the ellipses is different based on the hands (clockwise for the right hand and anticlockwise for the left hand) to make the drawing phase more natural and instinctive. For each combination, the participants had to repeat the draw 10 times where we discarded the first 2 and the last one to avoid a border effect. Each combination in the experiment was repeated 10 times. The resulting dataset consists of the following recordings: 14 participants, 2 hands, 3 speeds, 7 kept ellipses, and 10 repetitions of the experiment. From the total of 5880 recordings, we selected ≈5663, since some of them were corrupted. For each recording, we collected a time series reporting the position of the pencil on the graphics tablet $(x(t), y(t))$ and the pressure $p(t)$ with a sampling rate of 0.01 seconds. From the position, we compute the angular velocity $v(t)$, and the radius of curvature $r(t)$, which with the $p(t)$, are the most representative information on the movement.

4.4 Ball Exchange Dataset

The current section will present the dataset explored in the analysis of the paper:

1. *D'Amato, Vincenzo, et al. "The Importance of Multiple Temporal Scales in Motion Recognition: when Shallow Model can Support Deep Multi Scale Models." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022.*

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 three 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.

In this study, we employed the data collected during the EnTimeMent FET PROACTIVE project[5]. 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 detail, the full-body Mocap configuration (i.e., the Sports Marker Set by Qualysis[6]) has been used to collect data that allows detecting the location of 24 main joints of the body (skeleton)

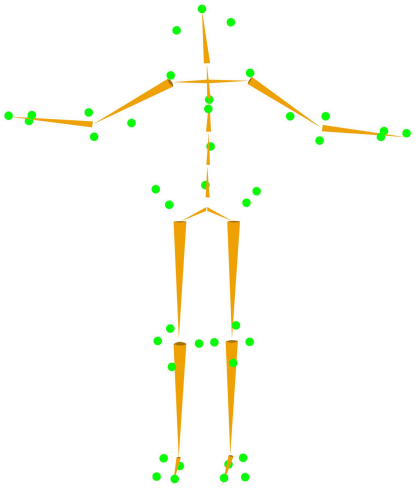


Figure 4.5: Physical markers (in green) and the resulting skeleton of 24 main joints (in orange) of the body.

based on 42 markers (see Figure 4.5) keeping track of the launcher and the receiver. Recordings were performed at *Casa Paganini - InfoMus Research Center of Genoa*[7]. The movements of 26 participants were acquired and 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 three different intentions (fair, aggressive, or deceptive) at a given distance of approximately 3 metres. 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 weight of the ball

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

[5] `https://entimement.dibris.unige.it/`
[6] `https://www.qualisys.com/`
[7] `www.casapaganini.org`

An example of the launches made with different ball weight and different intentions are shown in Figure 4.6.

During the experiment, participants started with the light ball. They had to exchange the ball a random number of times (from 10 to 30), first with fair, then with aggressive, and finally with a 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, the problem of balance, etc.).

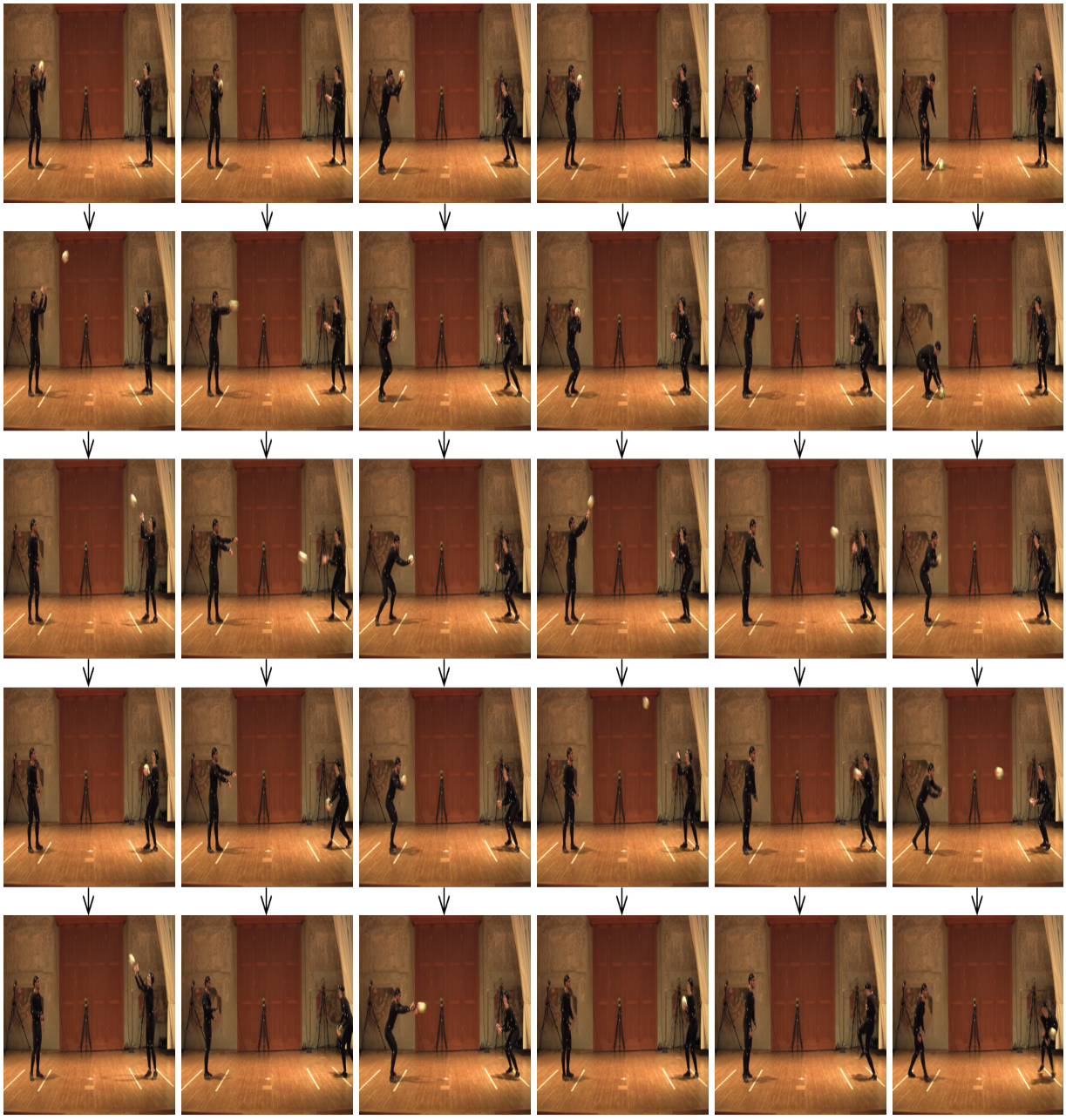
Participants had to throw the ball using two hands: this choice facilitates the involvement of the full body in both the launch and the reception phases, avoiding too complex movement and too high speed in the launch, as typically happens with single-hand launches.

Note that launching and receiving a ball contain both symmetric (launching requires an expansion while receiving a compression) and asymmetric (in the launching, one foot is usually behind the other) actions that 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 a 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 4.2.

Then, the raw datasets have been normalised as follows. For the launcher, the $24x3$ time series has been translated into a 24 time series corresponding to the 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 of a 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 4.2.



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

Figure 4.6: Example of launches for different Ball Weights (Light or Heavy) and different Launch Intentions (Fair, Aggressive, or Deceptive).

	Ball						
	Light			Heavy Ball			
	Intention			Intention			
Group	Fair	Aggressive	Deceptive	Fair	Aggressive	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 4.2: Raw Dataset

Chapter 5

Experimental Results

5.1 TELMI Dataset

The current section will present the results obtained in the papers:

1. *D'Amato, Vincenzo, et al. "Understanding violin players' skill level based on motion capture: a data-driven perspective." Cognitive Computation 12.6 (2020): 1356-1369*;
2. *D'Amato, Vincenzo, et al. "Accuracy and intrusiveness in data-driven violin players skill levels prediction: Mocap against myo against kinect." International Work-Conference on Artificial Neural Networks. Springer, Cham, 2021.*

In this section, we will report the results of applying the methodology presented in Chapter 3 over the data described in Chapter 4.1. In all experiments, we set $n_r = 100$. Experts (violinists 1 to 3) are labeled with $Y = 0$ and beginners (violinists 4 and 5) with $Y = 1$. The full list of hyperparameters is reported in Table 5.1.

To understand the extrapolation capability of the data-driven models, we studied two scenarios:

1. Leave One Person Out (LOPO): in this scenario, the model has been trained with all the

Algorithm	**Hyperparameters**
RF	$n_t : \{1000\}$ $n_f : \{d^{1/3}, d^{1/2}, d^{3/4}\}$

Table 5.1: Hyperparameters and Hyperparameters search space for all algorithms tested in the analysis of the TELMI dataset.

subjects except one that will be exploited to test the resulting model;

2. Leave One Exercise Out (LOEO): in this scenario, the model has been trained with all the exercises except one that will be exploited to test the resulting model.

The two scenarios just differ in the definitions of the learning, validation and test sets, which are the subset of data exploited for building, tuning and testing the models. For instance, in the LOPO scenario the learning, validation and test sets have been created by randomly selecting data from one person to be inserted in the test set, from another person to be inserted in the validation set, and from the remaining ones to be inserted into the learning set. For the LOEO scenario, we have the same procedure as the LOPO one but where exercises are considered instead of the people.

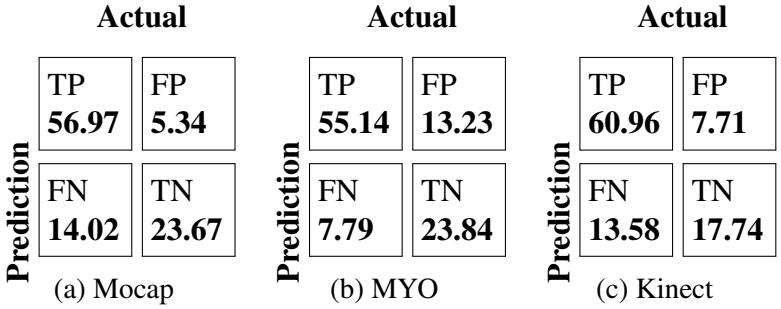
5.1.1 Recognition Performances for LOPO and LOEO

Let us present the results for the LOPO scenario. Table 5.2 reports the accuracy on different datasets for each violinist. Moreover, the overall confusion matrix is reported in Table 5.3 and for completeness, in Table 5.4, the other main classification metrics, such as overall accuracy, precision, recall and ROC-AUC, are described. Observing Table 5.2, we can notice how the three different datasets achieve quite high recognition performances ($> 78\%$ for both Mocap and MYO and $> 74\%$ for Kinect). Therefore, the lack of accuracy is not so relevant using different data sources as the recognition performances are very close to each other. Table 5.2 shows further information, namely which violinists are easier to predict correctly as experts or beginners. For instance, violinists 1 and 3 are good exponents of the expert class as they are easily recognised accurately in all the data sources. On the contrary, very different recognition performances are obtained in the beginner class. Indeed, in the analysis of Mocap data, we can observe how the violinist 4 is a good exponent of this class. However, exploiting MYO and Kinect data, the violinist 5 is the most representative for the beginner class. This behaviour is due to the different cardinality of the beginner class. Indeed, as we observed in Table 4.1, this is the most penalised class for both MYO and Kinect data. This different balance of datasets produces different results in the beginner class. In general, it is interesting to observe how MYO data has a much more balanced behaviour for beginner class: the True Negative scores – associated with novice class – and its recall are the highest compared to the others, as shown in Tables 5.3 and 5.4.

Let us present the recognition performances in the LOEO scenario. Tables 5.5, 5.6 and 5.7 are the counterparts, for the LOEO scenario, of Table 5.2, 5.3 and 5.4 for the LOPO scenario. Recognition metrics in the LOEO scenario are higher than the LOPO ones. Indeed, this happens because, in the training phase, more information is available for the LOPO one; namely, we have to predict if an exercise was performed by an expert or beginner violinist but have other ones played by the same violinist in the training set. In this scenario, we observe how the recognition performances are pretty high ($> 87\%$ for Kinect and $> 96\%$ for Mocap and MYO). It is also

Data		Mocap				MYO				Kinect			
Exp/ Beg	**Viol**	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**
0	1	98.19	0.51	97.06	99.02	100.00	0.00	100.00	100.00	89.99	2.26	84.00	94.00
	2	77.68	2.55	70.54	82.17	54.57	6.72	39.53	68.99	96.75	0.92	93.44	99.18
	3	99.73	0.40	98.47	100.00	93.33	3.78	81.68	97.71	80.40	0.39	79.39	81.68
1	4	88.89	3.05	79.82	92.98	66.34	9.48	30.36	82.14	35.21	3.01	26.79	42.86
	5	34.47	5.90	21.90	54.29	80.18	5.27	64.76	90.48	68.08	9.38	37.14	83.81
Mean		**79.79**	**2.25**	**73.56**	**85.69**	**78.88**	**3.52**	**63.27**	**87.86**	**74.09**	**3.62**	**64.15**	**80.31**

Table 5.2: Accuracy % in LOPO scenario



(a) Mocap (b) MYO (c) Kinect

Table 5.3: Average confusion matrix (in %) in LOPO scenario between all exercises and violinists.

interesting to observe how some exercises performed by violinists are always predicted correctly in the three different datasets that we explored. For instance, Right-hand exercises played by violinist 1 achieve the 100% of accuracy in all datasets, as shown in Table 5.5. Despite this, we notice how outcomes change a lot on different data sources. This is the case of exercises performed by violinists 4 and 5 where the cardinality of the datasets and the available exercises influences these outcomes, as mentioned before discussing the scores obtained in the LOPO scenario. Moreover, observing exercises in Table 5.5, we obtained results that may seem counter-intuitive compared to what we might expect. Indeed, we might think that the Technique exercise is the most discriminating one analysing the skill level of each violinist. In a context where only kinetic data are analysed, this is not true as we do not consider the quality of sound reproduced by each musician. Considering the analysis performed, we have further focused on the richness of movements in exercises played, or in other words, the exercises with more motions in their executions. Analysing the LOEO scenario, we can assert which exercises are more difficult to predict correctly for each violinist. For instance, observing Table 5.5 and comparing the results with those observed in Table 5.2, we can easily conclude that the weakness in the predictions of violinist 2 in Mocap data derived from exercises such as Left hand and Technique. This reasoning can be applied to all exercises and violinists presented in Tables 5.2 and 5.5.

5.1.2 Feature Ranking

In both LOPO and LOEO scenarios, we trained our model with features extracted and engineered from the raw data discussed in Chapter 4.1. To understand the relevance of each feature in terms of the violinist's skill level, we applied the MDA method discussed in Section 3. The results of the FR are shown in Table 5.8. In particular, they are easily comparable as we used similar features extracted from skeletons of Mocap data and Kinect data. A correct match between Mocap and Kinect features is impossible in terms of a different number of features and exact position in the rank. For instance, upper body kinetic energy is the main informative feature in

Data	**Mocap**				**MYO**				**Kinect**			
Measure	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**
Accuracy	79.79	2.25	73.56	85.69	78.88	3.52	63.27	87.86	74.09	3.62	64.15	80.31
Precision	81.61	1.73	77.30	85.21	64.44	3.64	56.74	73.62	69.54	2.66	59.80	74.47
Recall	62.79	3.21	55.25	73.56	75.37	4.62	59.63	84.47	56.65	6.28	37.89	68.32
ROC-AUC	0.91	0.01	0.88	0.94	0.86	0.03	0.76	0.92	0.74	0.02	0.67	0.78

Table 5.4: Average accuracy (%), precision (%), recall (%), and ROC-AUC between all exercises and violinists in LOPO scenario.

Data			Mocap				MYO				Kinect			
Exp/ Beg	**Viol**	**Exercise**	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**
0	1	Articulation	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	74.50	7.19	58.33	91.67
		Expressive	100.00	0.00	100.00	100.00	-	-	-	-	87.00	12.55	75.00	100.00
		Left hand	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	97.19	3.37	84.14	100.00
		Right hand	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00
		Technique	89.83	6.28	83.33	95.83	-	-	-	-	97.76	1.39	92.86	100.00
	2	Articulation	99.40	0.56	98.89	100.00	98.00	1.51	96.88	100.00	99.92	0.83	91.67	100.00
		Expressive	100.00	0.00	100.00	100.00	-	-	-	-	100.00	0.00	100.00	100.00
		Left hand	86.23	4.18	78.33	98.33	98.47	1.50	96.67	100.00	96.35	3.14	92.08	100.00
		Right hand	100.00	0.00	100.00	100.00	85.19	4.91	75.00	100.00	100.00	0.00	100.00	100.00
		Technique	88.01	5.02	82.29	100.00	100.00	0.00	100.00	100.00	86.00	3.91	83.33	91.67
	3	Articulation	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	99.79	0.47	98.72	100.00
		Expressive	100.00	0.00	100.00	100.00	-	-	-	-	64.67	7.96	33.33	66.67
		Left hand	100.00	0.00	100.00	100.00	99.33	2.01	93.33	100.00	100.00	0.00	100.00	100.00
		Right hand	100.00	0.00	100.00	100.00	100.00	0.00	100.00	100.00	96.50	7.96	66.67	100.00
		Technique	99.81	0.40	98.96	100.00	100.00	0.00	100.00	100.00	79.03	2.37	74.62	86.06
1	4	Articulation	100.00	0.00	100.00	100.00	94.83	1.64	91.38	98.28	-	-	-	-
		Expressive	100.00	0.00	100.00	100.00	-	-	-	-	79.00	9.87	50.00	100.00
		Left hand	98.56	3.76	88.89	100.00	95.67	0.82	91.67	95.83	-	-	-	-
		Right hand	87.07	7.97	64.28	100.00	-	-	-	-	-	-	-	-
		Technique	100.00	0.00	100.00	100.00	-	-	-	-	59.97	5.07	48.89	69.35
	5	Articulation	100.00	0.00	100.00	100.00	99.75	2.50	75.00	100.00	99.63	0.54	98.85	100.00
		Expressive	100.00	0.00	100.00	100.00	-	-	-	-	91.50	18.88	50.00	100.00
		Left hand	100.00	0.00	100.00	100.00	98.54	0.92	96.55	100.00	100.00	0.00	100.00	100.00
		Right hand	100.00	0.00	100.00	100.00	89.00	1.11	88.89	100.00	33.08	7.38	15.39	46.15
		Technique	95.08	0.83	93.33	98.33	98.40	1.00	97.78	100.00	87.17	2.82	75.00	91.67
Mean			**96.98**	**5.31**	**94.51**	**99.59**	**98.15**	**2.08**	**95.37**	**99.81**	**87.85**	**6.30**	**78.04**	**93.47**

Table 5.5: Accuracy % in LOEO scenario.

	Actual	
Prediction	TP **60.10**	FP **2.21**
	FN **0.84**	TN **36.85**

	Actual	
Prediction	TP **67.49**	FP **0.88**
	FN **1.38**	TN **30.25**

	Actual	
Prediction	TP **64.01**	FP **4.67**
	FN **4.97**	TN **26.35**

Table 5.6: Average confusion matrix (in %) between all exercises and violinists. From top to bottom, from left to right represented data source is Mocap, Kinect and MYO, in the LOEO scenario.

Mocap data, whereas, in the analysis of Kinect data, the same one is in the 4^{th} rank position. Despite this, we can assume that this feature is relevant for both datasets exploited. A common rank position is achieved by a feature called hand violin incidence (in 3^{rd} position for both Mocap and Kinect data). We also notice the top-ranking position of features describing the wrist roundness, the overall upper body kinetic energy, and the hand violin incidence. This suggests that these features are informative in the skill level classification of violin players. On the contrary, a redundant feature is left shoulder height placed in the last position for both data sources.

The central column of Table 5.8 represents relevant features for MYO data. In this case, it is easy to observe how we find information on the left hand in the first positions of the ranking. These characteristics highlight how the main features in ranking the different skill levels of violinists are the way the violin is held and the movements of the hand holding it. Furthermore, the acceleration of the left hand is not one of the main characteristics in the distinction between more and less experienced players. This is reasonable as the MYO sensor is positioned on the violinist's forearm since their movement is restricted during the musician's performance.

We try to summarise and combine the results obtained from the different data sources we analysed. We can observe from Table 5.8 how, for all data sources, the main informative motion describes movements very rich in kinetic energy and the hand holding the violin, providing us insights into the musicians' confidence with the instrument. This demonstrates the reasonableness of the results obtained with the different data sources. Moreover, domain experts validated our results in terms of feature ranking, agreeing with the results achieved by our algorithm. This process highlights how our model learns, in an effective way, movements concerning the correct execution of music tasks.

Data	**Mocap**				**MYO**				**Kinect**			
Measure	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**	**mean**	**std**	**min**	**max**
Accuracy	96.98	5.31	94.51	99.59	98.15	2.08	95.37	99.81	87.85	6.30	78.04	93.47
Precision	94.36	1.16	91.81	98.17	97.17	0.82	95.63	99.35	84.97	1.56	80.59	89.93
Recall	97.77	0.57	96.35	99.09	95.63	0.78	93.79	97.52	84.12	1.49	80.12	87.58
ROC-AUC	0.99	0.00	0.99	1.00	0.99	0.00	0.99	1.00	0.93	0.01	0.91	0.95

Table 5.7: Average accuracy (%), precision (%), recall (%), and ROC-AUC between all exercises and violinists in LOEO scenario.

Data	Mocap	MYO	Kinect
Rank	**Raw features**	**Raw features**	**Raw features**
1	upper body kinetic energy	left rotation	right wrist roundness
2	mean shoulder's velocity	left gyroscope	left wrist roundness
3	hand violin incidence	left EMG	hand violin incidence
4	distance lower bow violin	right gyroscope	upper body kinetic energy
5	left wrist roundness	right acceleration	left head inclination
6	right shoulder height	left acceleration	right shoulder height
7	right head inclination	right EMG	mean shoulder's velocity
8	right wrist roundness	right rotation	right head inclination
9	shoulder low back asymmetry	-	shoulder low back asymmetry
10	left head inclination	-	left shoulder height
11	bow violin incidence	-	-
12	distance upper bow violin	-	-
13	distance middle bow violin	-	-
14	left shoulder height	-	-

Table 5.8: Feature ranking of the original raw features (from top to least importance) for different data sources.

5.2 EmoPain-WHOLO-WeDraw Datasets

The current section will present the results obtained in the papers:

1. *D'Amato, Vincenzo, et al. "Keep it Simple: Handcrafting Feature and Tuning Random Forests and XGBoost to face the Affective Movement Recognition Challenge 2021." 2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW). IEEE, 2021.*

This section reports the results of exploiting the methodology presented in Chapter 3 on the tasks of the Affective Movement Recognition Challenge 2021 using the data described in Chapter 4.2.

We recall the pipeline of our approach and the parameter exploited in the experiments:

1. consider one of the three tasks of the Affective Movement Recognition Challenge 2021;
2. implements the feature engineering phase described in Section 3.2.4;
3. for each one of the considered algorithms (RF and XGBoost), we built a model using the MS and EE strategies defined in Section 3.4. The full list of hyperparameters is reported in Table 5.9.
4. with the optimal model retrained on all the data with the optimal hyperparameters we predicted the labels for the test samples, submitted them to the challenge organisers and

Algorithm	Hyperparameters
RF	n_t : {1000} n_f : {$d^{1/3}$, $d^{1/2}$, $d^{3/4}$ } n_l : {1,3,5}
XGBoost	l_r : { 0.01,0.05,0.1,0.3} n_d : { 3,4,5,6,7,8,9} m_l : { 0,0.1,0.2,0.3,0.4,0.5}

Table 5.9: Hyperparameters and Hyperparameters search space for all algorithms tested in the analysis of EmoPain-weDRAW-Unige-Maastricht Dance datasets.

got the results in terms of ACC, F1-0, F1-1, and MCC.

Since in this work we need to be able to extrapolate on previously unseen persons, we perform a particular resampling strategy: the Leave One Person Out (LOPO). In this setting, the model has been trained with all the subjects except one (i.e., we created $\mathcal{L}_l^r$ using the samples of all the subjects except one) that will be exploited to validate the resulting model (i.e., we created $\mathcal{V}_v^r$ using the samples of the remaining subject).

Table 5.10 reports recognition performances with RF and XGBoost on the three tasks of the Affective Movement Recognition Challenge 2021. The results are the ones provided by the challenge organisers. As shown in Table 5.10, RF and XGBoost have comparable performance, as there is no clear winner. This indicates clearly that the extracted features well represent the phenomena and the learning algorithm does not produce significant differences. Nevertheless, RF wins in two out of the three tasks. Note also that the final performance is in line with the results in the state-of-the-art, bearing in mind that a fair comparison is not possible as the results are obtained in different settings.

Finally, to be fully open and reproducible, we released all the codes to obtain the results reported in the article in a GIT repository[1].

5.3 Ellipsis Dataset

The current section will present the results obtained in the papers:

1. *D'Amato, Vincenzo, et al. "The Importance of Multiple Temporal Scales in Motion Recognition: from Shallow to Deep Multi Scale Models." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022.*

[1] `https://github.com/lucaoneto/ACII_AffectMove_2021`

Task	Algorithm	ACC	F1-0	F1-1	MCC
1	RF	0.84	0.90	0.48	0.41
	XGBoost	0.83	0.90	0.48	0.40
Task 1 Winner:	**RF**	**0.84**	**0.90**	**0.48**	**0.41**
2	RF	0.67	0.79	0.30	0.14
	XGBoost	0.66	0.78	0.29	0.12
Task 2 Winner:	**RF**	**0.67**	**0.79**	**0.30**	**0.14**
3	RF	0.76	0.76	0.76	0.53
	XGBoost	0.77	0.78	0.76	0.56
Task 3 Winner:	**XGBoost**	**0.77**	**0.78**	**0.76**	**0.56**

Table 5.10: Recognition Performances with RF and XGBoost on the three tasks of the Affective Movement Recognition Challenge 2021. The results are the ones provided by the challenge organisers.

In this section, we will report the results of applying the methodology presented in Section 3 over the data described in Section 4.3.

The complete list of the hyperparameters tested is reported in Table 5.11.

In our experiment, we studied two scenarios to understand the extrapolation capability of the different models described in Sections 3.2 and 3.3:

- Leave One Hand of one subject Out (LOHO): in this scenario, the models have been trained with all the subjects' data except the ones related to one hand of one subject that has been kept apart for testing purposes;
- Leave One Speed of one subject Out (LOSO): in this scenario, the models have been trained with all the subjects' data except the ones related to one speed of one subject which has been kept apart for testing purposes;

Therefore, the two scenarios just differ in the definition of the three sets $\mathcal{L}_l^r$, $\mathcal{V}_v^r$, and $\mathcal{T}_t^r$, which are the subset of data employed for building, tuning, and testing the models. For instance, in the LOHO scenario $\mathcal{L}_l^r$, $\mathcal{V}_v^r$, and $\mathcal{T}_t^r$ have been created by randomly selecting data from one hand of one subject to be inserted in $\mathcal{T}_t^r$, from another hand of a different subject to be inserted in $\mathcal{V}_v^r$, and from the remaining ones to be inserted into $\mathcal{L}_l^r$.

Algorithm	Hyperparameters
RF	$n_f : \{d^{1/3}, d^{1/2}, d^{3/4}\}$ $n_l : \{1\}$ $n_t : \{1000\}$
LSTM	l_r: $\{0.0001, 0.0005, 0.001\}$ $d_{r,0}$: $\{0.1, 0.15, \ldots, 0.5\}$ $d_{r,i}$: $\{0.1, 0.15, \ldots, 0.5\}$ C: $\{0.0001, 0.0005, 0.001, 0.005\}$ n_i: $\{25, 50, 75, 111\}$ h_l: $\{1, 2, 3, 4\}$
TCN	l_r: $\{0.0001, 0.0005, 0.001\}$ $d_{r,0}$: $\{0.1, 0.15, \ldots, 0.5\}$ $d_{r,i}$: $\{0.1, 0.15, \ldots, 0.5\}$ C: $\{0.0001, 0.0005, 0.001, 0.005\}$ h_l: $\{1, 2, 3\}$ n_i: $\{32, 64, 128\}$ $k_{s,i}$: $\{3, 5, 7, 9, 11\}$

Table 5.11: Hyperparameters and Hyperparameters search space for all algorithms tested in the analysis of the Ellipsis dataset.

5.3.1 Feature Engineering

Shallow models require adequate engineering, from the raw time series, of features that can adequately capture the phenomena under investigation by exploiting domain knowledge. In our case, we segmented the ellipse (and thus the associated time series) in different ways. In particular, each ellipse has been segmented (split) according to five different criteria (see Figure 5.1):

1. the ellipse is divided into segments characterised by high and low curvature (see Figure 5.1a);
2. the ellipse is divided into two symmetric parts according to the longest diagonal (see Figure 5.1b);
3. the ellipse is divided into four parts: the two more curved and the two more linear (see Figure 5.1c);
4. the ellipse is divided into six parts as depicted in Figure 5.1d;
5. all the previous split criteria (Figures 5.1a-5.1d) are considered.

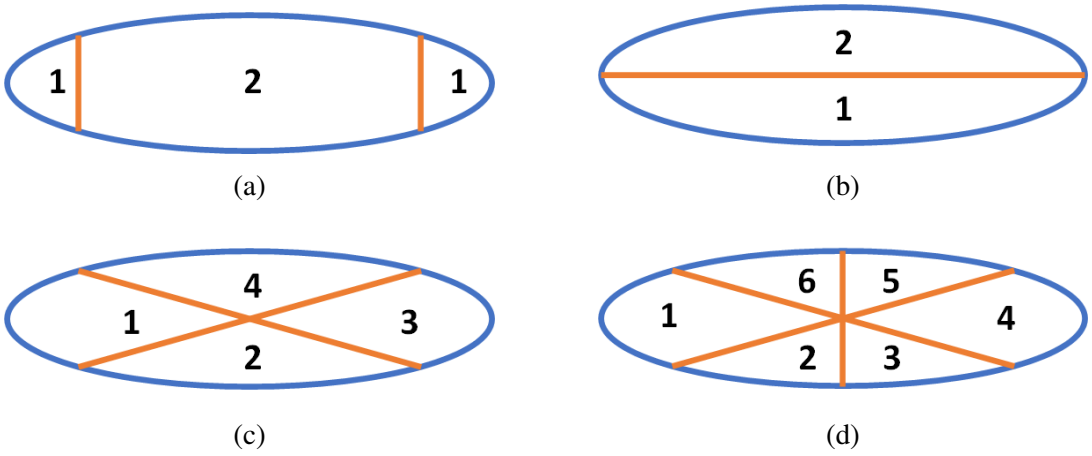


Figure 5.1: The ellipse criteria of segmentation.

On top of this feature engineering step, we applied a series of state-of-the-art classification algorithms [FDCBA14, WAF16]: RF, Support Vector Machines (with linear and Gaussian kernel), XGBoost, K-Nearest Neighbors. However, we decided to report only the results obtained using the best performing method of this family to face this problem: RF. Note that this is something that happens in many real-world problems. For example, results in Kaggle `www.kaggle.com`, which is the most popular Machine Learning competition website, show how RF and XGBoost algorithms are the top winner algorithms.

5.3.2 Recognition Performances for LOHO and LOSO

Table 5.12 and 5.13 report the percentage of accuracy in the LOHO and LOSO scenarios when exploiting RF (with the different criteria of segmentation of the ellipses described in Section 5.3.1, LSTM, and TCN (see Section 3.3) for each of the 14 subjects together with the average across the subjects.

Table 5.12 and 5.13 allow to observe that:

- As one might expect, performances on LOSO are generally higher than the ones on LOHO for all subjects and algorithms. This is the natural consequence of the fact that in the LOHO scenario we are asking for a more complex extrapolation capability for the algorithms;
- TCN consistently outperforms RF and LSTM in all scenarios, also demonstrating consistent performance across subjects;
- RF is quite competitive and outperforms, for some subjects, even the TCN. Nevertheless, for some subjects, performance is quite poor;
- RF in case (a) and (b) performs quite well. These results indicate that segmenting too little or too much of the ellipse is not a good solution while putting all the possible seg-

Alg. / Subj.	RF					LSTM	TCN
	(a)	**(b)**	**(c)**	**(d)**	**(e)**		
1	98.0±0.1	98.4±0.2	99.7±0.2	100.0±0.1	100.0±0.1	90.5±1.5	97.8±0.7
2	99.1±0.1	99.4±0.2	99.6±0.1	99.9±0.1	99.9±0.1	83.9±2.1	99.1±0.2
3	96.6±0.5	97.6±0.4	98.2±0.3	97.9±0.3	97.1±0.5	92.8±2.2	98.2±1.0
4	68.1±1.5	71.3±1.5	70.7±2.3	69.4±2.9	71.0±1.9	85.8±2.2	86.9±1.7
5	99.8±0.1	99.8±0.1	99.8±0.1	100.0±0.1	100.0±0.1	92.2±3.1	98.9±0.2
6	75.7±2.3	91.5±0.9	81.2±1.6	75.3±1.4	92.5±1.0	64.7±3.5	90.4±1.0
7	99.0±0.1	97.8±0.8	99.9±0.1	100.0±0.1	86.0±0.1	91.6±2.2	98.7±0.3
8	100.0±0.1	100.0±0.1	100.0±0.1	100.0±0.1	100.0±0.1	85.1±3.5	99.2±0.4
9	98.3±0.3	98.6±0.5	99.8±0.2	100.0±0.1	99.9±0.1	90.0±1.3	97.5±0.7
10	98.1±0.3	98.5±0.6	99.8±0.1	100.0±0.1	100.0±0.1	87.6±1.4	98.6±0.7
11	98.7±0.2	98.7±0.3	99.7±0.2	99.9±0.1	99.8±0.1	86.0±2.6	99.0±0.1
12	97.8±0.1	98.2±0.3	99.5±0.7	99.8±0.5	99.6±0.4	90.9±1.7	97.6±0.4
13	98.9±0.2	98.9±0.1	99.4±0.1	99.7±0.1	99.7±0.1	92.4±1.8	98.4±0.7
14	98.4±0.2	98.3±0.4	99.9±0.1	100.0±0.1	99.9±0.1	93.5±1.8	99.3±0.3
Avg.	**94.8±0.4**	**96.2±0.5**	**96.2±0.4**	**95.9±0.4**	**96.1±0.3**	**87.6±2.2**	**97.1±0.6**

Table 5.12: LOHO ACC when exploiting RF (with the different criteria of segmentation of the ellipses described in Section 5.3.1), LSTM, and TCN (see Section 3.3) for each of the 14 subjects together with the average across the subjects.

Alg. / Subj.	RF					LSTM	TCN
	(a)	**(b)**	**(c)**	**(d)**	**(e)**		
1	98.7±0.2	99.2±0.2	100.0±0.1	100.0±0.1	100.0±0.1	97.6±0.7	99.7±0.2
2	99.1±0.1	99.4±0.2	99.5±0.1	100.0±0.1	99.7±0.1	92.4±2.1	99.3±0.3
3	97.7±0.2	98.9±0.2	98.2±0.1	98.3±0.1	96.8±0.2	97.8±1.0	98.2±0.5
4	81.1±0.8	96.8±0.5	90.9±0.8	92.6±0.6	91.3±0.3	97.0±0.6	96.3±0.9
5	99.7±0.1	99.7±0.1	99.9±0.1	100.0±0.1	100.0±0.1	98.1±0.8	99.6±0.4
6	85.9±1.0	93.8±1.3	87.7±0.6	84.7±0.8	89.7±0.1	96.7±1.4	96.7±0.7
7	99.5±0.1	99.6±0.1	100.0±0.1	100.0±0.1	99.5±0.1	95.9±0.8	99.2±0.4
8	100.0±0.1	100.0±0.1	100.0±0.1	100.0±0.1	100.0±0.1	93.8±2.9	99.5±0.5
9	98.4±0.1	99.2±0.2	99.8±0.1	99.9±0.1	100.0±0.1	97.9±0.4	99.6±0.5
10	98.9±0.2	99.3±0.2	100.0±0.1	100.0±0.1	100.0±0.1	95.2±0.8	99.4±0.2
11	98.9±0.1	99.5±0.1	99.8±0.1	100.0±0.1	99.8±0.1	96.8±0.6	99.7±0.3
12	97.9±0.1	98.3±0.2	99.7±0.3	99.7±0.3	98.2±0.1	97.8±0.7	99.7±0.2
13	99.0±0.1	99.0±0.1	99.6±0.1	99.8±0.1	97.8±0.1	98.6±0.5	99.9±0.1
14	98.2±0.1	99.2±0.3	100.0±0.1	100.0±0.1	100.0±0.1	99.5±0.3	100.0±0.1
Avg.	**96.7±0.2**	**98.7±0.3**	**98.2±0.2**	**98.2±0.2**	**98.1±0.1**	**96.8±1.0**	**99.1±0.4**

Table 5.13: LOSO ACC when exploiting RF (with the different criteria of segmentation of the ellipses described in Section 5.3.1), LSTM, and TCN (see Section 3.3) for each of the 14 subjects together with the average across the subjects.

Alg. / Met.	RF (a)	RF (b)	RF (c)	RF (d)	RF (e)	LSTM	TCN
ACC	94.8±0.4	96.2±0.5	96.2±0.4	95.9±0.4	96.1±0.3	87.6±2.2	97.1±0.6
REC	94.8±0.4	96.2±0.5	96.2±0.4	95.9±0.4	96.1±0.3	87.6±2.2	97.1±0.6
PRE	94.8±0.3	96.2±0.5	96.2±0.4	95.9±0.4	96.1±0.4	87.6±2.5	97.1±0.4
ROC-AUC	0.9±0.1	0.9±0.1	0.9±0.1	0.9±0.1	0.9±0.1	0.9±0.1	0.9±0.1

(a) LOHO

Alg. / Met.	RF (a)	RF (b)	RF (c)	RF (d)	RF (e)	LSTM	TCN
ACC	96.7±0.2	98.7±0.3	98.2±0.2	98.2±0.2	98.1±0.1	96.8±1.0	99.1±0.4
REC	96.7±0.2	98.7±0.3	98.2±0.2	98.2±0.2	98.1±0.1	96.8±1.0	99.1±0.4
PRE	96.7±0.2	98.7±0.3	98.2±0.2	98.3±0.2	98.1±0.2	96.8±1.2	99.2±0.1
ROC-AUC	0.9±0.1	0.9±0.1	0.9±0.1	0.9±0.1	0.9±0.1	0.9±0.1	0.9±0.1

(b) LOSO

Table 5.14: ACC, REC, PRE, and ROC-AUC, averaged over the 14 subjects when exploiting RF (with the different criteria of segmentation of the ellipses described in Section 5.3.1), LSTM, and TCN (see Section 3.3).

mentation, as in case (e), does not guarantee optimal performance. These segmentations designed to capture multiple temporal scales are by construction, fixed and not customised for the specific problem. The TCN-based architecture, instead, actually learns the correct temporal scale to focus on;

- LSTM, as expected, is the algorithm with the lowest performance. This is due to its ability to capture different temporal scales being too limited.

For completeness, we report ACC, REC, PRE, and ROC-AUC in Table 5.14 averaged over the 14 subjects.

We decided not to report the results obtained with other shallow (e.g. Support Vector Machines) or deep (e.g. bi-directional LSTM) algorithms, as their performance is inferior to that of RF, LSTM and the TCN-based architecture. Instead, we reported LSTM to show that naive deep

architectures do not outperform classical methods such as RF. However, the complete set of results is available in our public repository[2].

5.3.3 Feature Ranking

As described in Section 5.1.2, in order to better understand how and what the different RF and TCN models actually learned from the data, Table 5.15 reports the sections ranking [3] performed with RF in the different sectioning scenarios (see Section 5.3.1) and Figure 5.2 reports the attention maps of TCN (see Section 3.3), averaged across subjects, for $p(t)$, $v(t)$, and $r(t)$ for both LOHO and LOSO scenarios.

Table 5.15 and Figure 5.2 allow to observe that:

- As might be expected, the most important sections of the two scenarios (LOHO and LOSO) do not appear to be the same, as they try to extract different information (hand and speed). When using shallow models (i.e. RF) for sections (a), (b) and (c) sections retain the same importance in both LOHO and LOSO scenarios whereas for sections (d) and (e), the ranking is very different. When using deep models (i.e., TCN), instead, only for $v(t)$ the attention map remains similar for both LOHO and LOSO scenarios;
- For both LOHO and LOSO scenarios, shallow models identify as the most informative sections those who are closer to the initial part of the drawing in all the analysed sectioning criteria. On the other hand, deep models generally find the final parts of the drawing as the most informative. This shows how different the perception of the two models is. The shallow ones focus on the "preparation" of the movement, while the deep ones focus more on the "completion" of the movement. The deep model, in this case, perceives the movement in a way which seems more similar to a human: human beings tend to become more confident in labelling a movement when it tends to be completed;
- shallow models primarily focus on more "linear" sections concerning the more "curved ones". The opposite happens for deep models. Also, in this case, deep model perception is more similar to human one: human tends to distinguish movements based on the most complex parts;
- Finally, note that shallow models tend to focus on sections based on the particular choice of the sectioning criteria and cannot perceive and define their one way of understanding the movement. Deep models, on the other hand, by construction, can do this by defining attention maps according to the particular problem and implicitly defining their dissection criteria to then be able to perceive the different time scales of movement.

[2] `https://github.com/lucaoneto/IJCNN2022_Ellipses`

[3] The letters indicate the sectioning and the numbers indicate the specific section, see Figure 5.1, so note that c.1 is the same as d.1 and c.3 is the same as d.4.

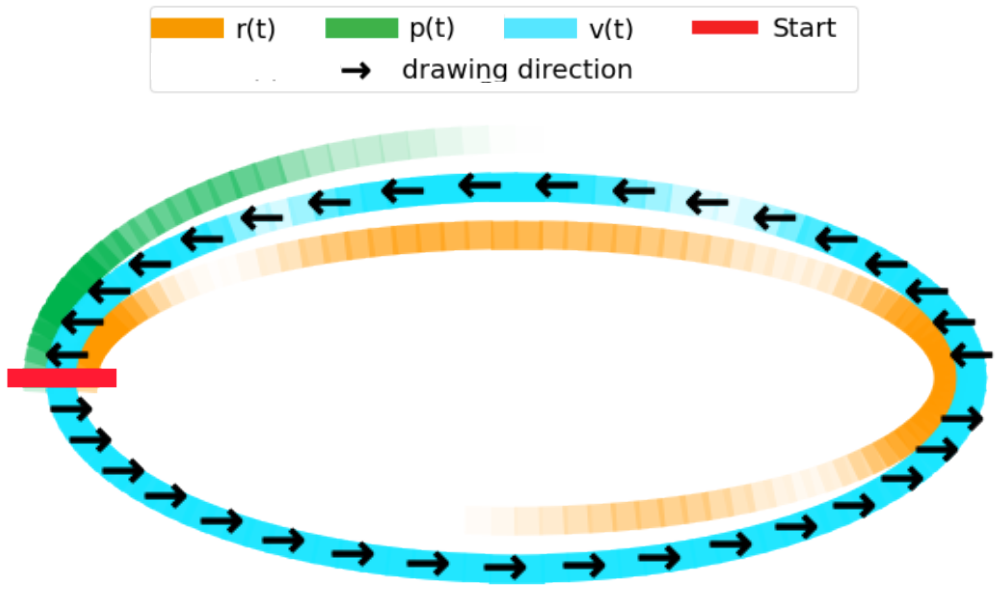
Sectioning	Rank 1	2	3	4	5	6	7	8	9	10	11	12
(a)	a.2	a.1										
(b)	b.1	b.2										
(c)	c.4	c.2	c.1	c.3								
(d)	d.3	d.2	d.4	d.1	d.5	d.6						
(e)	d.3	d.2	c.1 (d.1)	c.3 (d.4)	b.1	c.2	d.6	d.5	a.2	b.2	a.1	c.4

(a) LOHO

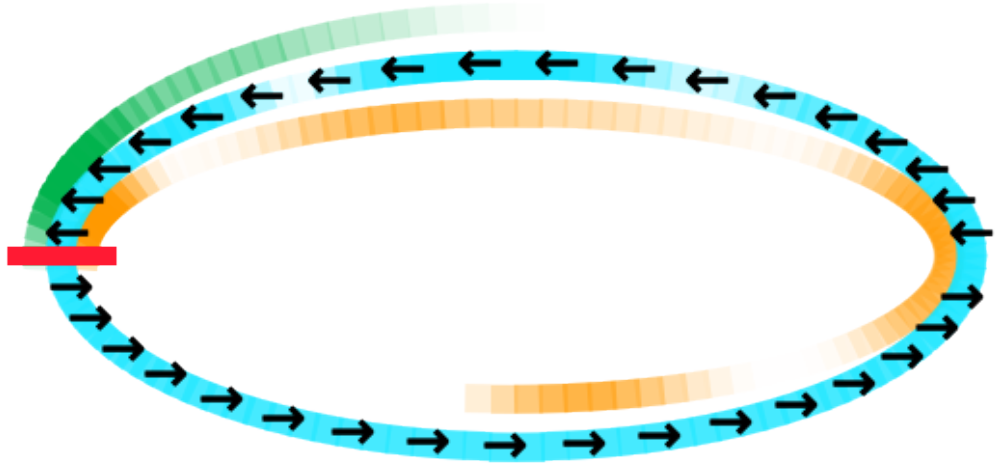
Sectioning	Rank 1	2	3	4	5	6	7	8	9	10	11	12
(a)	a.2	a.1										
(b)	b.1	b.2										
(c)	c.4	c.2	c.1	c.3								
(d)	d.2	d.4	d.3	d.1	d.5	d.6						
(e)	c.1 (d.1)	d.3	c.2	a.1	b.2	b.1	d.6	a.2	d.5	c.3 (d.4)	c.4	d.2

(b) LOSO

Table 5.15: Sections ranking[3] performed with RF in the different sectioning scenarios for both LOHO and LOSO scenarios.



(a) LOHO



(b) LOSO

Figure 5.2: Attention maps of TCN, averaged across subjects, for $v(t)$, $r(t)$, and $p(t)$ for both LOHO and LOSO scenarios. The more intense the colour, the more important the particular part of the input time series.

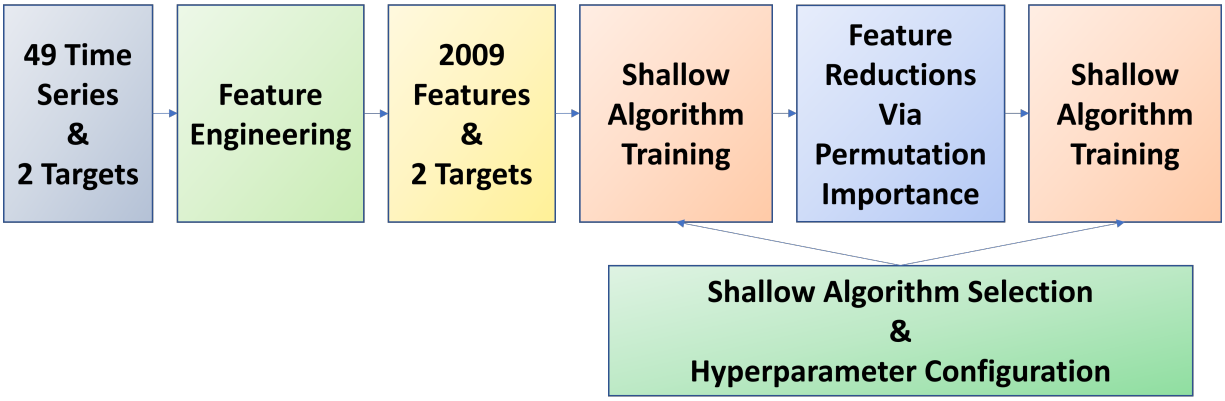


Figure 5.3: Pipeline for the Shallow Models.

.

5.4 Ball Exchange Dataset

The current section will present the results obtained in the papers:

1. *D'Amato, Vincenzo, et al. "The Importance of Multiple Temporal Scales in Motion Recognition: when Shallow Model can Support Deep Multi Scale Models." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022.*

In this section, we report the results of applying the methodology presented in Section 3 to the data described in Section 4.4. To further improve the performance of the shallow models, we decided to add a dimensionality reduction step to remove all non-informative variables, which can be numerous, as the feature engineering step is rather comprehensive [VDMPVdH09]. In particular, for each training phase of each model, we applied the permutation feature importance method [Bre01, FRD19, Mol20], 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 5.3.

Note that, in this case, the configuration space of the hyperparameters explodes further for LSTMs, as we have possibly different configurations of the kernel size for each time series. Therefore, to reduce the configuration space of the hyper-parameters and, at the same time, the number of weights in the final network (for both LSTM- and TCN-based deep models), we decided to reduce the number of input series from the original 49. It is reasonable to assume that many of these series contain redundant information. 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 once again the permutation importance

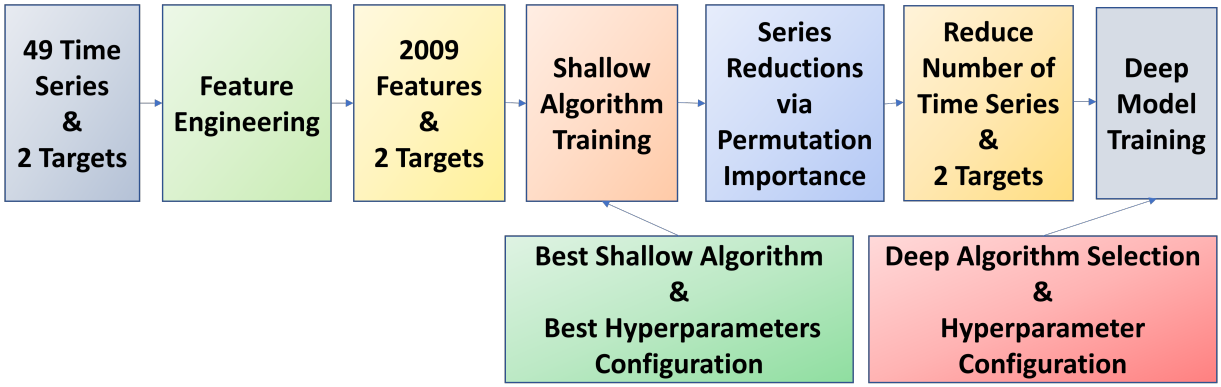


Figure 5.4: Pipeline for the Deep Models.

.

with a 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. Due to this reduction in the number of input time series, LSTM- and TCN-based architectures greatly reduce the number of weights to be adjusted and the search space of the hyperparameter configuration. The pipeline we proposed for deep models is depicted in Figure 5.4.

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 configurations for all tested algorithms is reported in Table 5.16.

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 3.2 and 3.3:

- 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 that 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 that 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 that has been kept apart for testing purposes.

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

5.4.1 Recognition Performance

As observed in Section 4.4, we want to address two different scopes, i.e., 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 5.17,5.18,5.19, and 5.20 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).

Observing Tables 5.17, 5.18, 5.19, and 5.20 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 capable of obtaining results close to TCN ones in two of the three proposed extrapolation scenarios, i.e., the simplest one (LOIO and LOBO). Note that, the shallow models (LSVM, KSVM, RF, and XGBoost) are often competitive with (or better than) the classical deep ones (LSTM and BLSTM), while TCN is always the top-performing method.

The same behaviour can be observed in Tables 5.21, 5.22, 5.23, and 5.24, which are the counterpart of Tables 5.17, 5.18, 5.19, and 5.20 when the target is the launch intention.

In conclusion, we can make a series of observations based on the experimental results. TCN can outperform all other algorithms no matter the target (ball weight or launch intention), scenarios (LOIO, LOBO, and LOGO), and metrics exploited. Shallow models generally outperform classical deep models but cannot reach the performance of TCN. In some simple extrapolation scenarios, LSVM can compete with TCN. This confirms what we discussed so far: it is not easy for deep models to outperform well-calibrated shallow models. However, if deep models use the knowledge gained from shallow models, they can achieve even greater improvements in recognition performance in very complex extrapolation scenarios.

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**

Table 5.17: Predicting the Ball Weight in LOIO scenario: 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. / 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**

Table 5.18: Predicting the Ball Weight in LOBO scenario: 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. / 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**

Table 5.19: Predicting the Ball Weight in LOGO scenario: 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	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 5.20: 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.

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**

Table 5.21: Predicting the Launch Intention in LOIO scenario: 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. / 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**

Table 5.22: Predicting the Launch Intention in LOBO scenario: 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. / 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**

Table 5.23: Predicting the Launch Intention in LOGO scenario: 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 5.24: 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.

Chapter 6

Conclusions & Future Perspectives

In this thesis, we focused on the analysis of human movement using data-driven techniques. In particular, we focused on the non-verbal aspects of human movement, with an emphasis on full-body movements. These movements, to be analysed, can be acquired using non-invasive sensors (e.g., video camera, Kinect) or invasive sensors (e.g., Mocap, IMU). The difference between the two approaches is mainly related to three factors. The first factor is the discomfort of wearing a device (more or less obtrusive) on one's body during movement. In the most extreme cases, this can degenerate into wearing actual suits (e.g. Mocap). The risk of using this type of sensor is to cause stress to the participant, compromising their performance in the long term. On the other hand, invasive devices are much more accurate and robust in acquiring human movement data. Indeed, these sensors do not need to deal with problems such as light conditions or possible occlusions typical of video camera sensors. Unfortunately, however, the precision of detail of these sensors comes at a price. Their use is often in the research field as it is not an affordable expense for everyone.

In any case, these types of sensors can acquire information on human movement, often described as time series. At each instant of time when the action of interest is performed, these sensors provide instant-by-instant (timestamp) information either on the spatial position of the skeletal joint or on the muscular activation of a particular muscle.

In order to process this data, several methodologies in the literature exist related to the different disciplines that study human movement. As already mentioned, the focus of this thesis has been on data-driven methods. These methods can interpret the information in the data by searching for rules, associations or patterns that can represent the relationships between input (e.g. the human action acquired with sensors) and output (e.g. the type of action performed). Furthermore, these models may represent a new research frontier as they can analyse large masses of data and focus on aspects that even an expert user might miss.

The literature on data-driven models proposes two families of methods that can process time

series and human movement. The first family, called shallow models, extract features from the time series that can help the learning algorithm find associations in the data. These features are identified and designed by domain experts who can identify the best ones for the problem faced. On the other hand, the second family avoids this phase of extraction by the human expert since the models themselves can identify the best set of features to optimise the learning of the model.

This thesis aimed to understand how human actions can be modelled by a data-driven model. A better design of these actions can, in the future, lead to numerous advantages when applied to everyday life. In particular, an integration of this technology in the homes of people with physical and/or motor problems can produce numerous benefits: from simple monitoring of health conditions to a prediction of a degenerative health condition, from the prevention of injuries due to critical postures to simple support for everyday life. The idea of this thesis was to understand whether there is a correlation between how people perceive and interpret their own and others' movements. The movement of human beings appears to respond to a complex motor system that contains signals at different hierarchical levels [WCH+12, BBKK17, ZDlTH12, Aur12]. For example, an action such as "grasping a glass on a table" represents a high-level action, but to perform this task, the body needs several motor inputs that include the activation of different joints of the body (shoulder, arm, hand, fingers, etc.). Each of these different joints/muscles have a different size, responsiveness, and precision with a complex non-linearly stratified temporal dimension where every muscle has its temporal scale. Parts such as the fingers responds much faster to brain input than more voluminous body parts such as the shoulder. The cooperation we have when we perform an action produces smooth, effective, and expressive movement in a complex multiple temporal scale cognitive task. Following this layered structure, the human body can be described as a kinematic tree, consisting of joints connected. As a first approximation, we can state that larger muscles are slower and are characterised by a slower perceptual response over time for the smaller muscles. Nevertheless, some movements of larger muscles can be fast: for example, the small corrections to keep us in balance to compensate for a loss of balance, to avoid the risk of falling. Note also that the multiple temporal scales nature of the human movement, also characterises how humans perceive other people's movements [Hol09, GdLL15].

Although it is nowadays well known that human movement and its perception are characterised by multiple temporal scales [WCH+12,BBKK17,ZDlTH12,MHA+16,Hol09,GdLL15,SHTF+19, Aur12], very few works in the literature are focused on studying this particular property. For instance, Ihlen et al. [IV10] provided quantitative support for studying the multiple temporal scales in human action and perception using wavelet-based multifractal analysis in the response series of four cognitive tasks (simple response, word naming, choice decision and interval estimation). Camurri et al. [CVP+16] demonstrate that computational models of expressive qualities should operate at different temporal scales starting from previous research on human perception and dance theories [ND19]. Authors of [CVP+16] propose a framework where features are computed at different levels, i.e., low-level features (e.g., velocity) are computed instantaneously, while higher ones (e.g., impulsiveness) are computed on a larger temporal scale. In image recognition tasks like object detection, semantic segmentation, and action recognition, Temporal Con-

volutional Networks (TCNs) with dilated convolutions [RHGS15,CPK+17,DSND19] have been widely adopted to increase receptive field sizes without increasing model complexity. Indeed, by applying dilated convolutions with different filter sizes, multiple temporal scales can be efficiently captured and the use of this mathematical operation can handle larger temporal contexts efficiently. Recent research, carried out in the European FET PROACTIVE Project EnTimeMent[1], focuses its attention on addressing the importance of multiple temporal scales in movement analysis and prediction. Inside EnTimeMent, Beyan et al. [BKV+21] propose an approach that can model the dynamics of full-body movement data represented on multiple temporal scales where features are processed by two independent and parallel shallow TCNs.

Therefore, with this thesis, we needed a method to apply one property of the human motion domain, multi-temporal scales, to deep learning models, the only data-driven models that can be extended to handle this property. We asked ourselves two questions: what if we applied knowledge about how human movements are performed to deep learning models? Can this knowledge improve current automatic recognition standards?

In this thesis, we have tried to answer these questions. Note that in order to obtain effective, complete and robust answers, we have analysed both families of methods whenever possible. The results demonstrated that although the majority of research follows the direction of deep models because, when there is a lot of data available these provide better results, shallow models remain a high standard to overcome to date. In the analysed datasets, these (shallow) models produce distinctly high recognition performance, often better than (deep) models specifically designed to handle time series problems. This assumption proved particularly truthful in our experiments, both for small and large datasets. The deep models analysed in this thesis are of two types: the first, used as a baseline to compare the results of the surface models and our proposal, based on a recursive architecture called LSTM that represents the state-of-the-art to date; the second, used to evaluate our proposal, based on the different intrinsic time scales of human motion. The results showed that LSTM architectures achieved far lower recognition performance than shallow models. On the other hand, our proposed architecture was able to outperform both models (shallow and deep).

The list of works presented in this thesis, in which we tried to answer the previous research questions, can be observed below:

1. *D'Amato, Vincenzo, et al. "Understanding violin players' skill level based on motion capture: a data-driven perspective." Cognitive Computation 12.6 (2020): 1356-1369*;
2. *D'Amato, Vincenzo, et al. "Accuracy and intrusiveness in data-driven violin players skill levels prediction: Mocap against myo against kinect." International Work-Conference on Artificial Neural Networks. Springer, Cham, 2021*;
3. *D'Amato, Vincenzo, et al. "Keep it Simple: Handcrafting Feature and Tuning Random*

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

Forests and XGBoost to face the Affective Movement Recognition Challenge 2021." 2021 9th International Conference on Affective Computing and Intelligent Interaction Workshops and Demos (ACIIW). IEEE, 2021;

4. *D'Amato, Vincenzo, et al. "The Importance of Multiple Temporal Scales in Motion Recognition: when Shallow Model can Support Deep Multi Scale Models." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022*;

5. *D'Amato, Vincenzo, et al. "The Importance of Multiple Temporal Scales in Motion Recognition: from Shallow to Deep Multi Scale Models." 2022 International Joint Conference on Neural Networks (IJCNN). IEEE, 2022*.

Let us now look at the conclusions for each problem addressed in more detail.

The main objective of the TELMI dataset was to understand which technology (Mocap, MYO or Kinect) and which motion characteristics can be used to efficiently and effectively distinguish a professional violin player from a student while saving on sensor intrusiveness and accuracy. We engineered peculiar features starting from different sources (Mocap, MYO, and Kinect) that we used for training a data-driven classifier to distinguish between two levels of violinist experience, namely Beginners and Experts. We studied two extrapolation scenarios (i.e., extrapolating over players and extrapolation over exercises). In these two scenarios, we compared the accuracy we lose by using Mocap, MYO or Kinect data, ordered from the most invasive and expensive technology to the least intrusive and cheapest. We discovered that using the Kinect in the most interesting scenario (i.e., extrapolating over players) reduces the recognition performance by 4% (out of 79%). This means that the loss in accuracy is negligible for having a fully unintrusive and affordable supporting tool. Finally, we studied the most predictive raw features ranked by the algorithms to predict the quality of a violinist to corroborate the significance of the results. We observed how recognition performances depend directly on the confidence in the instrument and mainly on movements of the left hand that holds the violin. Results, both in terms of accuracy and insight into the cognitive problem, support the proposal and the proposed technique as a support tool for students to monitor and enhance their home study and practice. In conclusion, we demonstrated how Kinect could provide an affordable and effective application to assist students in learning violin.

Let us now look at the results obtained in the Affective Movement Recognition Challenge 2021. The challenge involved three datasets on body movement, which is a fundamental component of everyday life both in the execution of actions that constitute physical functioning and in the rich expression of affect, cognition and intention. The datasets are based on a deep understanding of the requirements of automatic sensing technology for chronic pain physical rehabilitation, mathematical problem solving and interactive dance contexts. To address this challenge, we relied on a single, simple but effective approach that is still competitive with the most advanced results in the literature on all three datasets. Our approach was based on a two-step procedure: first, we carefully created features capable of fully and concisely representing the raw data and

then applied Random Forest and XGBoost, carefully tuned with rigorous statistical procedures, on them to provide the predictions. As required by the challenge, we reported the results in terms of three different metrics: accuracy, F1 score and Matthew's correlation coefficient. These results were provided by the organisers of the challenge, as the true labels of the test set have not yet been released.

In the ellipsis dataset, we investigated how deep and shallow data models can support the understanding of human movement and, in particular, its multiple time-scale nature. We showed how shallow data-driven models, which achieve reasonably good recognition performance, require a usually complex phase of handcrafting of the features based on domain-specific knowledge, thus limiting the ability to extract all the possible information from the data. Therefore, we propose a new deep multi-scale data-driven model based on temporal convolutional networks that can automatically learn features from data at different time scales and outperform state-of-the-art shallow models in terms of recognition performance. We tested the effectiveness of our approach in a customised motion recognition experiment, i.e. the detection of a person drawing an ellipse on a graphics tablet based on the speed, pressure and curvature of the drawing motion. Exploiting the intrinsic hierarchy in the dataset, we considered two different extrapolation scenarios, namely on hand and on speed, to understand the potentiality of the proposed architecture. Results, both in terms of performance and interpretability of the model, support the need and the usefulness of studying human movement at different temporal scales employing multi-temporal scale data-driven models. In particular, we observed how the differences between the proposed model and a traditional one are not so evident in terms of recognition performance, but in terms of the interpretability of the model and what it has learned. Shallow models tend to perform well on some subjects and poorly on others, and the information extracted from the data is usually different from human intuition. On the other hand, the proposed architecture tends to achieve consistent performance across subjects and extract information more in line with human intuition.

In the ball exchange dataset, we argued that to analyse human movement, it is necessary to model multiple time scales that fully describe its complexity. 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 a research frontier that can provide new insights, but current approaches cannot adequately address the need to model 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 handcrafting features according to the domain knowledge. The second one is based on deep models that can be extended to handle multiple time scales but are difficult to exploit because there are too many architecture configurations. For this reason, we proposed a new deep model at multiple time scales, based on the temporal convolutional network, capable of learning features from data at different time scales, overcoming the state-of-the-art of deep and shallow models, similar to the one presented for the previous analysis. Furthermore, this model exploits shallow models to tune the architecture configuration. We then collected data and tested our proposal in a specially designed experiment to prove the validity of our approach.

Specifically, we collected motion capture data on dyad actions in which two people exchange a ball. Since the weight of the ball and the throwing intentions change, we demonstrated how it is possible to automatically detect the weight of the ball or the intention behind the throw based on motion data. The results support both the proposal and the need to use deep multi-scale models as a tool to understand better human movement and its nature at multiple time scales.

The deep neural network we propose aims at a radical paradigm and technology shift in the analysis of human movement, in which the time frame for the analysis is grounded on new neuroscientific, biomechanical, psychological and computational evidence, and dynamically adapted to the human time frame that controls the phenomena under investigation. In the future, our proposed model may present a new baseline/benchmark for studying human movement in terms of recognition performance and explainability. This model can also potentially be applied to technologies that study and process human motion to make them more precise and accurate. For instance, the current generation of motion capture and motion analysis systems could be positively influenced by this multi-scale approach, providing them with complete new functionality to achieve a new generation of time-aware multi-sensory motion perception and prediction systems. Other applicative scenarios can include health (healing and support of everyday life of persons with chronic pain and disability), performing arts (e.g. dance), sports, and entertainment group activities, with and without living architectures. In addition, the deep neural network we propose can enable new forms of human-machine interaction (affective human-machine synchronisation) and human-human entrainment experiences, mainly involving the non-verbal, embodied and immersive, active and affective dimensions of qualitative gesture. Another benefit of data-driven models is that the proposed model and approach can effectively reduce the originally highly dimensional and redundant raw sensor observations to something more manageable for the inference of, for instance, emotions. Finally, to further validate the assumptions made in this thesis and the robustness of our approach, the proposed model will be tested and refined iteratively on new human motion datasets with single, dyadic and small group interactions.

List of Figures

List of Tables

Bibliography

[AAO17] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare. Credit card fraud detection using machine learning techniques: A comparative analysis. In *International conference on computing networking and informatics (ICCNI)*, 2017.

[Abe13] B. Abernethy. *Biophysical foundations of human movement*. Human Kinetics, 2013.

[ABR00] 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.

[Ado02] R. Adolphs. Neural systems for recognizing emotion. *Current opinion in neurobiology*, 12(2):169–177, 2002.

[Agg15] C. C. Aggarwal. *Data Mining: the Textbook*. Springer, 2015.

[AGO+13] D. Anguita, A. Ghio, L. Oneto, Parra P. X., and J. L. Reyes Ortiz. A public domain dataset for human activity recognition using smartphones. In *Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning*, 2013.

[AKH+18] Bruce Abernethy, Vaughan Kippers, Stephanie J Hanrahan, Marcus G Pandy, Ali McManus, and Laurel Mackinnon. *Biophysical Foundations of Human Movement*. Human Kinetics, 2018.

[AKRP+15] M. S. H. Aung, S. Kaltwang, B. Romera-Paredes, B. Martinez, A. Singh, M. Cella, M. Valstar, H. Meng, A. Kemp, M. Shafizadeh, et al. The automatic detection of chronic pain-related expression: requirements, challenges and the multimodal emopain dataset. *IEEE transactions on affective computing*, 7(4):435–451, 2015.

[APM+16] P. Alborno, S. Piana, M. Mancini, R. Niewiadomski, G. Volpe, and A. Camurri. Analysis of intrapersonal synchronization in full-body movements displaying

different expressive qualities. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, 2016.

[ARB18] Q. A. Al-Radaideh and D. Q. Bataineh. A hybrid approach for arabic text summarization using domain knowledge and genetic algorithms. *Cognitive Computation*, 10(4):651–669, 2018.

[Arg13] Michael Argyle. *Bodily communication*. Routledge, 2013.

[Aur12] D. Aur. From neuroelectrodynamics to thinking machines. *Cognitive Computation*, 4:4–12, 2012.

[BBKK17] 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.

[BKK18] 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.

[BKV+21] 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.

[BLB+17] A. Bagnall, J. Lines, A. Bostrom, J. Large, and E. Keogh. The great time series classification bake off: a review and experimental evaluation of recent algorithmic advances. *Data mining and knowledge discovery*, 31(3):606–660, 2017.

[Bre01] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[BSD22] Q. Bian, D. ET Shepherd, and Z. Ding. A hybrid method integrating a musculoskeletal model with long short-term memory (lstm) for human motion prediction. In *44th Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC)*, 2022.

[BSF94] Y. Bengio, P. Simard, and P. Frasconi. Learning long-term dependencies with gradient descent is difficult. *IEEE transactions on neural networks*, 5(2):157–166, 1994.

[BSLR18] M. Badarna, I. Shimshoni, G. Luria, and S. Rosenblum. The importance of pen motion pattern groups for semi-automatic classification of handwriting into mental workload classes. *Cognitive Computation*, 10(2):215–227, 2018.

[Bul16] Peter E Bull. *Posture & gesture*, volume 16. Elsevier, 2016.

[BVM22] S. Boner, C. Vogt, and M. Magno. Tiny tcn model for gesture recognition with multi-point low power tof-sensors. In *IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, 2022.

[BWFDS19] P. J. Bota, C. Wang, A. L.N. Fred, and H. P. Da Silva. A review, current challenges, and future possibilities on emotion recognition using machine learning and physiological signals. *IEEE Access*, 7:140990–141020, 2019.

[C+22] X. Cao et al. Human motion recognition information processing system based on lstm recurrent neural network algorithm. *Journal of Ambient Intelligence and Humanized Computing*, pages 1–13, 2022.

[CCPC12] L. Chaby, M. Chetouani, M. Plaza, and D. Cohen. Exploring multimodal social-emotional behaviors in autism spectrum disorders: an interface between social signal processing and psychopathology. In *International Conference on Privacy, Security, Risk and Trust and International Conference on Social Computing*, 2012.

[CCVG07] A. Camurri, P. Coletta, G. Varni, and S. Ghisio. Developing multimodal interactive systems with eyesweb xmi. In *International conference on New interfaces for musical expression*, 2007.

[CDCLC05] A. Cappozzo, U. Della Croce, A. Leardini, and L. Chiari. Human movement analysis using stereophotogrammetry: Part 1: theoretical background. *Gait & posture*, 21(2):186–196, 2005.

[CDSFS19] N. D. Cilia, C. De Stefano, F. Fontanella, and A. Scotto. A ranking-based feature selection approach for handwritten character recognition. *Pattern Recognition Letters*, 121:77–86, 2019.

[CDT15] B. Caramiaux, M. Donnarumma, and A. Tanaka. Understanding gesture expressivity through muscle sensing. *ACM Transactions on Computer-Human Interaction (TOCHI)*, 21:1–26, 2015.

[CEREZ+16] A. Camurri, K. El Raheb, O. Even-Zohar, Y. Ioannidis, A. Markatzi, J. M. Matos, E. Morley-Fletcher, P. Palacio, M. Romero, A. Sarti, et al. Wholodance: towards a methodology for selecting motion capture data across different dance learning practice. In *International Symposium on Movement and Computing*, 2016.

[CG16] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. In *acm sigkdd international conference on knowledge discovery and data mining*, 2016.

[CGCB14] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*, 2014.

[CL17] C. S. Calude and G. Longo. The deluge of spurious correlations in big data. *Foundations of science*, 22(3):595–612, 2017.

[CMMS11] 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.

[CO67] W. S Condon and W. D. Ogston. A segmentation of behavior. *Journal of psychiatric research*, 1967.

[CPK+17] 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*, 40:834–848, 2017.

[CS09] J. Chandler and N. Schwarz. How extending your middle finger affects your perception of others: Learned movements influence concept accessibility. *Journal of Experimental Social Psychology*, 45(1):123–128, 2009.

[CV95] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[CVP+16] 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 *Proceedings of the 3rd International Symposium on Movement and Computing*, 2016.

[DG09] B. De Gelder. Why bodies? twelve reasons for including bodily expressions in affective neuroscience. *Philosophical Transactions of the Royal Society B: Biological Sciences*, 364(1535):3475–3484, 2009.

[DGPAC19] N. De Giorgis, E. Puppo, P. Alborno, and A. Camurri. Evaluating movement quality through intrapersonal synchronization. *IEEE Transactions on Human-Machine Systems*, 49(4):304–313, 2019.

[DP07] G. Dong and J. Pei. *Sequence data mining*. Springer Science & Business Media, 2007.

[DSND19] 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 (WACV)*, 2019.

[Dub20] P. Duboue. *The Art of Feature Engineering: Essentials for Machine Learning*. Cambridge University Press, 2020.

[DVO+20] 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*, 12(6):1356–1369, 2020.

[DVO+21] V. D'Amato, E. Volta, L. Oneto, G. Volpe, A. Camurri, and D. Anguita. Accuracy and intrusiveness in data-driven violin players skill levels prediction: Mocap against myo against kinect. In *International Work-Conference on Artificial and Natural Neural Networks*, 2021.

[EB02] M. O Ernst and M. S Banks. Humans integrate visual and haptic information in a statistically optimal fashion. *Nature*, 415(6870):429–433, 2002.

[EB04] M. O Ernst and H. H Bülthoff. Merging the senses into a robust percept. *Trends in cognitive sciences*, 8(4):162–169, 2004.

[Eco16] Umberto Eco. *Trattato di semiotica generale*. La Nave di Teseo Editore spa, 2016.

[EF69] P. Ekman and W. V. Friesen. The repertoire of nonverbal behavior: Categories, origins, usage, and coding. *semiotica*, 1(1):49–98, 1969.

[EF74] P. Ekman and W. V Friesen. Detecting deception from the body or face. *Journal of personality and Social Psychology*, 29(3):288, 1974.

[EK20] M. W. Eysenck and M. T. Keane. *Cognitive psychology: A student's handbook*. Psychology press, 2020.

[Eno08] R. M. Enoka. *Neuromechanics of human movement*. Human kinetics, 2008.

[ESD21] S. Eltanani, T. O Scheper, and H. Dawes. K-nearest neighbor algorithm: Proposed solution for human gait data classification. In *IEEE Symposium on Computers and Communications (ISCC)*, 2021.

[FAO+21] I. C. Ferreira, M. V.C. Aragão, E. M. Oliveira, B. T. Kuehne, E. M. Moreira, and O. AS Carpinteiro. The development of the open machine-learning-based anti-spam (open-malbas). *IEEE Access*, 9:138618–138632, 2021.

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

[FDCBA14] 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*, 15(1):3133–3181, 2014.

[FRD19] 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*, 20(177):1–81, 2019.

[Fri01] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

[FT05] N. Fragopanagos and J. G Taylor. Emotion recognition in human–computer interaction. *Neural Networks*, 18(4):389–405, 2005.

[FWH+19] 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.

[GA10] A. M Green and D. E Angelaki. Multisensory integration: resolving sensory ambiguities to build novel representations. *Current opinion in neurobiology*, 20(3):353–360, 2010.

[GBC16] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. MIT press, 2016.

[GdLL15] R. L. Goldstone, J. R. de Leeuw, and D. H. Landy. Fitting perception in and to cognition. *Cognition*, 135:24–29, 2015.

[GE03] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3(Mar):1157–1182, 2003.

[GMW+22] J. Gao, C. Ma, D. Wu, X. Xu, S. Wang, and J. Yao. Recognition of human motion intentions based on bayesian-optimized xgboost algorithm. *Journal of Sensors*, 2022, 2022.

[Goo13] P. Good. *Permutation tests: a practical guide to resampling methods for testing hypotheses*. Springer Science & Business Media, 2013.

[GPTM10] R. Genuer, J. M. Poggi, and C. Tuleau-Malot. Variable selection using random forests. *Pattern Recognition Letters*, 31(14):2225–2236, 2010.

[GS06] A. Gallace and C. Spence. Multisensory synesthetic interactions in the speeded classification of visual size. *Perception & psychophysics*, 68(7):1191–1203, 2006.

[GSC00] F. A Gers, J. Schmidhuber, and F. Cummins. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

[GSD+18] 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*, 62:729–754, 2018.

[HBF+01] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, et al. Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, 2001.

[Hen64] F. M Henry. Physical education: An academic discipline. *Journal of Health, Physical Education, Recreation*, 35(7):32–69, 1964.

[HFL22] F. He, T. Fu, and W.C. Lee. Rel-cnn: Learning relationship features in time series for classification. *IEEE Transactions on Knowledge and Data Engineering*, 2022.

[HHK+06] G. K. Haugstad, T. S Haugstad, U. M Kirste, S. Leganger, S. Wojniusz, I. Klemmetsen, and U. F Malt. Posture, movement patterns, and body awareness in women with chronic pelvic pain. *Journal of psychosomatic research*, 61(5):637–644, 2006.

[HHM+21] 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*, 57:2964–2971, 2021.

[HJK21] S. Hafeez, A. Jalal, and S. Kamal. Multi-fusion sensors for action recognition based on discriminative motion cues and random forest. In *International Conference on Communication Technologies (ComTech)*, 2021.

[Hol09] A. O. Holcombe. Seeing slow and seeing fast: two limits on perception. *Trends in cognitive sciences*, 13(5):216–221, 2009.

[HRF+18] E. Halilaj, A. Rajagopal, M. Fiterau, J. L. Hicks, T. J Hastie, and S. L. Delp. Machine learning in human movement biomechanics: Best practices, common pitfalls, and new opportunities. *Journal of biomechanics*, 81:1–11, 2018.

[HS97] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[Hua14] G. B. Huang. An insight into extreme learning machines: random neurons, random features and kernels. *Cognitive Computation*, 6(3):376–390, 2014.

[HUE+19] 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*, 118:91–96, 2019.

[IA02] C. E Izard and B. P Ackerman. Self-organization of discrete emotions, emotion patterns, and emotion-cognition. *Emotion, development, and self-organization: Dynamic systems approaches to emotional development*, page 15, 2002.

[IPOS13] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu. Human3. 6m: Large scale datasets and predictive methods for 3d human sensing in natural environments. *IEEE transactions on pattern analysis and machine intelligence*, 36(7):1325–1339, 2013.

[IV10] E.A.F. Ihlen and B. Vereijken. Interaction-dominant dynamics in human cognition: Beyond $1/f\alpha$ fluctuation. *Journal of Experimental Psychology: General*, 139(3):436, 2010.

[JEP+21] J. Jumper, R. Evans, A. Pritzel, T. Green, and Others. Highly accurate protein structure prediction with alphafold. *Nature*, 596(7873):583–589, 2021.

[JGS+20] 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*, 51:2391–2406, 2020.

[JJK21] M. Javeed, A. Jalal, and K. Kim. Wearable sensors based exertion recognition using statistical features and random forest for physical healthcare monitoring. In *International Bhurban Conference on Applied Sciences and Technologies (IBCAST)*, 2021.

[Joh73] G. Johansson. Visual perception of biological motion and a model for its analysis. *Perception & psychophysics*, 14(2):201–211, 1973.

[JTAW20] 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*, pages 1–13, 2020.

[KB14] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412. 6980*, 2014.

[KBB12] A. Kleinsmith and N. Bianchi-Berthouze. Affective body expression perception and recognition: A survey. *IEEE Transactions on Affective Computing*, 4(1):15–33, 2012.

[KBM+07] K. P Körding, U. Beierholm, W. J. Ma, S. Quartz, J. B Tenenbaum, and L. Shams. Causal inference in multisensory perception. *PLoS one*, 2(9):e943, 2007.

[KDVdS17] L. Keuninckx, J. Danckaert, and G. Van der Sande. Real-time audio processing with a cascade of discrete-time delay line-based reservoir computers. *Cognitive Computation*, 9(3):315–326, 2017.

[KIK+18] 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*, 115:24–35, 2018.

[KLJ03] A. Kvåle, A. E Ljunggren, and T. B Johnsen. Examination of movement in patients with long-lasting musculoskeletal pain: reliability and validity. *Physiotherapy Research International*, 8(1):36–52, 2003.

[Koh95] R. Kohavi. A study of cross-validation and bootstrap for accuracy estimation and model selection. In *International Joint Conference on Artficial Intelligence*, 1995.

[KSG+13] M. Karg, A.A. Samadani, R. Gorbet, K. Kühnlenz, J. Hoey, and D. Kulić. Body movements for affective expression: A survey of automatic recognition and generation. *IEEE Transactions on Affective Computing*, 4(4):341–359, 2013.

[KWM11] J. R Kwapisz, G. M Weiss, and S. A Moore. Activity recognition using cell phone accelerometers. *ACM SigKDD Explorations Newsletter*, 12(2):74–82, 2011.

[L+89] Y. LeCun et al. Generalization and network design strategies. *Connectionism in perspective*, 19(143-155):18, 1989.

[LBRF20] D. Liciotti, M. Bernardini, L. Romeo, and E. Frontoni. A sequential deep learning application for recognising human activities in smart homes. *Neurocomputing*, 396:501–513, 2020.

[LC12] T.W. Lu and C.F. Chang. Biomechanics of human movement and its clinical applications. *The Kaohsiung journal of medical sciences*, 28(2):S13–S25, 2012.

[LDZ+19] Q. Lei, J. X. Du, H. B. Zhang, S. Ye, and D. S. Chen. A survey of vision-based human action evaluation methods. *Sensors*, 19(19):4129, 2019.

[LJKP21] S. Lee, H. Ji, J. Kim, and E. Park. What books will be your bestseller? a machine learning approach with amazon kindle. *The Electronic Library*, 2021.

[LPGA21] G. Lisca, C. Prodaniuc, T. Grauschopf, and C. Axenie. Less is more: Learning insights from a single motion sensor for accurate and explainable soccer goalkeeper kinematics. *IEEE Sensors Journal*, 21(18):20375–20387, 2021.

[LS16] 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*, 48:973–993, 2016.

[LYC17] 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.

[LZLQ21] 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.

[LZQ+21] H. Li, Q. Zheng, X. Qi, W. Yan, Z. Wen, N. Li, and C. Tang. Neural network-based mapping mining of image style transfer in big data systems. *Computational Intelligence and Neuroscience*, 2021, 2021.

[MF69] A. Mehrabian and J. T Friar. Encoding of attitude by a seated communicator via posture and position cues. *Journal of Consulting and Clinical Psychology*, 33(3):330, 1969.

[MHA+16] 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*, 6:24831, 2016.

[MM01] G. Martino and L. E Marks. Synesthesia: Strong and weak. *Current Directions in Psychological Science*, 10(2):61–65, 2001.

[MMN17] D. Micucci, M. Mobilio, and P. Napoletano. Unimib shar: A dataset for human activity recognition using acceleration data from smartphones. *Applied Sciences*, 7(10):1101, 2017.

[Mol20] C. Molnar. *Interpretable machine learning*. Lulu.com, 2020.

[MS12] M. Mehu and K. R. Scherer. A psycho-ethological approach to social signal processing. *Cognitive processing*, 13:397–414, 2012.

[MW06] C. M Mulvenna and V. Walsh. Synaesthesia: supernormal integration? *Trends in cognitive sciences*, 10(8):350–352, 2006.

[MW11] Micah M Murray and Mark T Wallace. *The neural bases of multisensory processes*. CRC Press, 2011.

[NBW+05] P. M Niedenthal, L. W Barsalou, P. Winkielman, S. Krauth-Gruber, and F. Ric. Embodiment in attitudes, social perception, and emotion. *Personality and social psychology review*, 9(3):184–211, 2005.

[ND19] J. Newlove and J. Dalby. *Laban for all*. Routledge, 2019.

[NMP+17] R. Niewiadomski, M. Mancini, S. Piana, P. Alborno, G. Volpe, and A. Camurri. Low-intrusive recognition of expressive movement qualities. In *ACM international conference on multimodal interaction*, 2017.

[One19] L. Oneto. *Model Selection and Error Estimation in a Nutshell*. Springer, 2019.

[ONJ+20] T. Olugbade, J. Newbold, R. Johnson, E. Volta, P. Alborno, R. Niewiadomski, M. Dillon, G. Volpe, and N. Bianchi-Berthouze. Automatic detection of reflective thinking in mathematical problem solving based on unconstrained bodily exploration. *IEEE Transactions on Affective Computing*, 2020.

[OOA16] I. Orlandi, L. Oneto, and D. Anguita. Random forests model selection. In *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2016.

[OSDCI17] J. Oliva, J. I. Serrano, M. D. Del Castillo, and A. Iglesias. Cross-linguistic cognitive modeling of verbal morphology acquisition. *Cognitive Computation*, 9(2):237–258, 2017.

[PD12] I. Poggi and F. D'Errico. Social signals: a framework in terms of goals and beliefs. *Cognitive Processing*, 13:427–445, 2012.

[Pie98] J. P. Piek. *Motor behavior and human skill: a multidisciplinary approach*. Human Kinetics, 1998.

[PLAH21] A. T. Purnomo, D.B. Lin, T. Adiprabowo, and W. F. Hendria. Non-contact monitoring and classification of breathing pattern for the supervision of people infected by covid-19. *Sensors*, 21(9):3172, 2021.

[PPBS01] F. E. Pollick, H. M. Paterson, A. Bruderlin, and A. J. Sanford. Perceiving affect from arm movement. *Cognition*, 82(2):B51–B61, 2001.

[PR00] R. W Picard and W. Rosalind. Toward agents that recognize emotion. *VIVEK-BOMBAY-*, 13(1):3–13, 2000.

[PS13] CV Parise and C Spence. Audiovisual cross-modal correspondences in the general population. In *Oxford Handbook of Synesthesia*, pages 790–815. Oxford University Press, 2013.

[PS20] D. A Pisner and D. M Schnyer. Support vector machine. In *Machine learning*. Elsevier, 2020.

[PS22] T. W. Pribadi and T. Shinoda. Hand motion analysis for recognition of qualified and unqualified welders using 9-dof imu sensors and support vector machine (svm) approach. *Hand*, 13(1), 2022.

[PSOC16] S. Piana, A. Staglianò, F. Odone, and A. Camurri. Adaptive body gesture representation for automatic emotion recognition. *ACM Transactions on Interactive Intelligent Systems*, 6(1):1–31, 2016.

[Rar67] G. L. Rarick. The domain of physical education as a discipline. *Quest*, 9(1):49–52, 1967.

[RBHP21] A. Roy, B. Banerjee, A. Hussain, and S. Poria. Discriminative dictionary design for action classification in still images and videos. *Cognitive Computation*, 13:698–708, 2021.

[RHGS15] S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. *Advances in neural information processing systems*, 28:91–99, 2015.

[RM08] L. Rokach and O. Z. Maimon. *Data Mining with Decision Trees: Theory and Applications*. World Scientific, 2008.

[ROOS+16] J. L. Reyes-Ortiz, L. Oneto, A. Sama, X. Parra, and D. Anguita. Transition-aware human activity recognition using smartphones. *Neurocomputing*, 171:754–767, 2016.

[RPC22] V Radhika, Ch Rajendra Prasad, and A Chakradhar. Smartphone-based human activities recognition system using random forest algorithm. In *International Conference for Advancement in Technology (ICONAT)*, 2022.

[RRI+22] M. K.A. Rahman, N. E.A. Rashid, N. N. Ismail, N. A.Z. Zakaria, Z. I. Khan, S. A.E. Ab Rahim, and F. N.M. Isa. Hand gesture recognition based on continuous wave (cw) radar using principal component analysis (pca) and k-nearest neighbor (knn) methods. *JOIV: International Journal on Informatics Visualization*, 6(1-2):188–194, 2022.

[RS12] A. Reiss and D. Stricker. Introducing a new benchmarked dataset for activity monitoring. In *16th international symposium on wearable computers*, 2012.

[SAVdP08] Y. Saeys, T. Abeel, and Y. Van de Peer. Robust feature selection using ensemble feature selection techniques. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, 2008.

[SBCS15] L. Scocchia, N. Bolognini, S. Convento, and N. Stucchi. Cathodal transcranial direct current stimulation can stabilize perception of movement: evidence from the two-thirds power law illusion. *Neuroscience letters*, 609:87–91, 2015.

[SGK22] S. A. Siddiqui, L. Gutzeit, and F. Kirchner. The influence of labeling techniques in classifying human manipulation movement of different speed. *arXiv preprint arXiv:2202.02426*, 2022.

[SHTF+19] S. Sepp, S. J. Howard, S. Tindall-Ford, S. Agostinho, and F. Paas. Cognitive load theory and human movement: Towards an integrated model of working memory. *Educational Psychology Review*, 31:1–25, 2019.

[SLJ+15] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015.

[SP21] A. Stergiou and R. Poppe. Multi-temporal convolutions for human action recognition in videos. In *International Joint Conference on Neural Networks*, 2021.

[Spe11] C. Spence. Crossmodal correspondences: A tutorial review. *Attention, Perception, & Psychophysics*, 73(4):971–995, 2011.

[SSBD14] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory To Algorithms*. Cambridge University Press, 2014.

[SSFK21] K. Shioiri, K. Saho, M. Fujimoto, and Y. Kobayashi. Radar-based gait classification of elderly non-fallers and multiple fallers using machine learning. In *IEEE 3rd Global Conference on Life Sciences and Technologies (LifeTech)*, 2021.

[SSS+17] D. Silver, J. Schrittwieser, K. Simonyan, et al. Mastering the game of go without human knowledge. *Nature*, 550(7676):354–359, 2017.

[STC04] J. Shawe-Taylor and N. Cristianini. *Kernel methods for pattern analysis*. Cambridge university press, 2004.

[Ste12] B. E. Stein. *The new handbook of multisensory processing*. Mit Press, 2012.

[SU17] S. Scardapane and A. Uncini. Semi-supervised echo state networks for audio classification. *Cognitive Computation*, 9(1):125–135, 2017.

[SYM21] F. Sakagami, H. Yamada, and S. Muramatsu. Accuracy improvement of human motion recognition with mw-fmcw radar using cnn. In *International Symposium on Antennas and Propagation (ISAP)*, 2021.

[Sze10] R. Szeliski. *Computer vision: algorithms and applications*. Springer Science & Business Media, 2010.

[TB22] D. Thakur and S. Biswas. An integration of feature extraction and guided regularized random forest feature selection for smartphone based human activity recognition. *Journal of Network and Computer Applications*, page 103417, 2022.

[THP21] L. Tong, J. He, and L. Peng. Cnn-based pd hand tremor detection using inertial sensors. *IEEE Sensors Letters*, 5(7):1–4, 2021.

[TKL11] Julia Trommershauser, Konrad Kording, and Michael S Landy. *Sensory cue integration*. Computational Neuroscience, 2011.

[TML+22] L. Tong, H. Ma, Q. Lin, J. He, and L. Peng. A novel deep learning bi-gru-i model for real-time human activity recognition using inertial sensors. *IEEE Sensors Journal*, 22(6):6164–6174, 2022.

[TMPP21] K. Tonchev, A. Manolova, R. Petkova, and V. Poulkov. Human skeleton motion prediction using graph convolution optimized gru network. In *XXX International Scientific Conference Electronics (ET)*, 2021.

[TTS+21] H.Y. Tang, S.H. Tan, T.Y. Su, C.J. Chiang, and H.H. Chen. Upper body posture recognition using inertial sensors and recurrent neural networks. *Applied Sciences*, 11(24):12101, 2021.

[TZY22] J. Tang, J. Zhang, and J. Yin. Temporal consistency two-stream cnn for human motion prediction. *Neurocomputing*, 468:245–256, 2022.

[VAM+19] M. J. Vaessen, E. Abassi, M. Mancini, A. Camurri, and B. de Gelder. Computational feature analysis of body movements reveals hierarchical brain organization. *Cerebral Cortex*, 29(8):3551–3560, 2019.

[VDMPVdH09] L. Van Der Maaten, E. Postma, and J. Van den Herik. Dimensionality reduction: a comparative. *Journal of Machine Learning Research*, 10(66-71):13, 2009.

[VdSRDG07] J. Van den Stock, R. Righart, and B. De Gelder. Body expressions influence recognition of emotions in the face and voice. *Emotion*, 7(3):487, 2007.

[VJ04] P. Viola and M. J Jones. Robust real-time face detection. *International journal of computer vision*, 57:137–154, 2004.

[VMHB+18] T. Von Marcard, R. Henschel, M. J Black, B. Rosenhahn, and G. Pons-Moll. Recovering accurate 3d human pose in the wild using imus and a moving camera. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018.

[VP15] 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*, 1:10–17, 2015.

[VPB09] A. Vinciarelli, M. Pantic, and H. Bourlard. Social signal processing: Survey of an emerging domain. *Image and vision computing*, 27:1743–1759, 2009.

[VPH+11] 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*, 3:69–87, 2011.

[VT82] P. Viviani and C. Terzuolo. Trajectory determines movement dynamics. *Neuroscience*, 7(2):431–437, 1982.

[VTW+21] C. Vong, T. Theptit, V. Watcharakonpipat, P. Chanchotisatien, and S. Laitrakun. Comparison of feature selection and classification for human activity and fall recognition using smartphone sensors. In *Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunication Engineering*, 2021.

[WAF16] M. Wainberg, B. Alipanahi, and B. J. Frey. Are random forests truly the best classifiers? *The Journal of Machine Learning Research*, 17(1):3837–3841, 2016.

[Wal98] H. G Wallbott. Bodily expression of emotion. *European journal of social psychology*, 28(6):879–896, 1998.

[WCH+12] M. Wijnants, R. Cox, F. Hasselman, A. Bosman, and G. Van Orden. A trade-off study revealing nested timescales of constraint. *Frontiers in physiology*, 3:116, 2012.

[Wei22] H. Weiwei. Classification of sport actions using principal component analysis and random forest based on three-dimensional data. *Displays*, 72:102135, 2022.

[WHT06] J. Ward, B. Huckstep, and E. Tsakanikos. Sound-colour synaesthesia: To what extent does it use cross-modal mechanisms common to us all? *Cortex*, 42(2):264–280, 2006.

[WLHL16] P. Wang, Z. Li, Y. Hou, and W. Li. Action recognition based on joint trajectory maps using convolutional neural networks. In *Proceedings of the 24th ACM international conference on Multimedia*, 2016.

[WTF19] 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.

[WXWL18] H. Wang, L. Xu, X. Wang, and B. Luo. Learning optimal seeds for ranking saliency. *Cognitive Computation*, 10(2):347–358, 2018.

[WZJ21] P. Wang, Y. Zhang, and W. Jiang. Application of k-nearest neighbor (knn) algorithm for human action recognition. In *IEEE 4th Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC)*, 2021.

[WZL+17] B. Wang, R. Zhu, S. Luo, X. Yang, and G. Wang. H-mrst: a novel framework for supporting probability degree range query using extreme learning machine. *Cognitive Computation*, 9(1):68–80, 2017.

[XZ22] Y. Xu and L. Zhao. Inception-lstm human motion recognition with channel attention mechanism. *Computational and Mathematical Methods in Medicine*, 2022, 2022.

[YK15] F. Yu and V. Koltun. Multi-scale context aggregation by dilated convolutions. *arXiv preprint arXiv:1511. 07122*, 2015.

[YLZ+21] W. Yu, R. Liu, D. Zhou, Q. Zhang, and X. Wei. An improved gru network for human motion prediction. In *IEEE 7th International Conference on Virtual Reality (ICVR)*, 2021.

[YLZJ19] G. Yao, T. Lei, J. Zhong, and P. Jiang. Learning multi-temporal-scale deep information for action recognition. *Applied Intelligence*, 49:2017–2029, 2019.

[YYY22] P. Yin, L. Yang, and M. Yang. Research on recognition of human motion state based on force and motion sensor fusion. In *IEEE 2nd International Conference on Power, Electronics and Computer Applications (ICPECA)*, 2022.

[YZH+22] X. Yi, Y. Zhou, M. Habermann, S. Shimada, V. Golyanik, C. Theobalt, and F. Xu. Physical inertial poser (pip): Physics-aware real-time human motion tracking from sparse inertial sensors. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022.

[ZC18] A. Zheng and A. Casari. *Feature engineering for machine learning: principles and techniques for data scientists*. O'Reilly Media, Inc., 2018.

[ZDlTH12] 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*, 35(3):582–596, 2012.

[ZLX+16] 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.

[ZMY+21] Z. Zheng, H. Ma, W. Yan, H. Liu, and Z. Yang. Training data selection and optimal sensor placement for deep-learning-based sparse inertial sensor human posture reconstruction. *Entropy*, 23(5):588, 2021.

[ZWS+18] H. G. Zhang, L. Wu, Y. Song, C. W. Su, Q. Wang, and F. Su. An online sequential learning non-parametric value-at-risk model for high-dimensional time series. *Cognitive Computation*, 10(2):187–200, 2018.

[ZY21] H. Zhou and G. Yu. Research on pedestrian detection technology based on the svm classifier trained by hog and ltp features. *Future Generation Computer Systems*, 125:604–615, 2021.