

Article

Enhancing sEMG-Based Finger Motion Prediction with CNN-LSTM Regressors for Controlling a Hand Exoskeleton

Mirco Vangi , Chiara Brogi , Alberto Topini , Nicola Secciani  and Alessandro Ridolfi 

Department of Industrial Engineering, University of Florence, 50139 Florence, Italy; chiara.brogi@unifi.it (C.B.); alberto.topini@unifi.it (A.T.); nicola.secciani@unifi.it (N.S.); alessandro.ridolfi@unifi.it (A.R.)

* Correspondence: mirco.vangi@unifi.it

Abstract: In recent years, the number of people with disabilities has increased hugely, especially in low- and middle-income countries. At the same time, robotics has made significant advances in the medical field, and many research groups have begun to develop low-cost wearable solutions. The Mechatronics and Dynamic Modelling Lab of the Department of Industrial Engineering at the University of Florence has recently developed a new version of a wearable hand exoskeleton for assistive purposes. In this paper, we will present a new regression method to predict the finger angle position of the first joint from the value of the sEMG of the forearm and the previous position of the finger itself. To acquire the dataset necessary to train the regressor a specific graphical user interface was developed which was able to acquire sEMG data from a Myo armband and the finger position from a Leap Motion Controller. Two long short-term memory (LSTM) models were compared, one in its standard configuration and the other with a convolutional layer, yielding significantly better performance for the second one, with an increase in R^2 coefficient from an average value of 0.746 to 0.825, leading to the conclusion that a convolutional layer could increase performance when few sensors are available.

Keywords: finger motion; hand exoskeletons; electromyography; CNN-LSTM; regression analysis



Citation: Vangi, M.; Brogi, C.; Topini, A.; Secciani, N.; Ridolfi, A. Enhancing sEMG-Based Finger Motion Prediction with CNN-LSTM Regressors for Controlling a Hand Exoskeleton. *Machines* **2023**, *11*, 747. <https://doi.org/10.3390/machines11070747>

Academic Editor: Zheng Chen

Received: 21 June 2023

Revised: 7 July 2023

Accepted: 14 July 2023

Published: 17 July 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

With the start of the “fourth industrial revolution”, also known as “industry 4.0”, robots have become essential support for healthcare systems both in clinical environments and on rehabilitation paths [1–4]. According to the most recent World Health Organization (WHO) report, around 1.3 billion people—about 16% of the global population—experience a significant disability [5]. Disabilities can result from several factors, some of them related to the lifestyle of the subject, others derived from genetic neurodegenerative deformities or neurodegenerative diseases, and still others after transient phenomena such as strokes. Nonetheless, people living in low- and middle-income countries (LMIC) have poor access to rehabilitation services, and in many of these countries, the density of rehabilitation-trained practitioners is often below 10 per 1 million population [6]. Indeed, medical devices and treatments are generally very expensive and, therefore, not accessible to the poorest. Rehabilitation treatments also have non-negligible costs and are often dropped prematurely by those unable to afford them. As a result of such considerations, creating a low-cost rehabilitation device that is also easy to use for non-trained personnel is necessary to provide more people with a quality of life equal to unimpaired persons.

In particular, one of the most important tools for human beings is the hand; for this reason, various hand-assistance devices have been developed over the last few decades [7]. Despite being oriented toward the same application field, these devices differ in several aspects, such as dimension, materials, weight, and control system. Concerning the last reported feature, one of the most popular control system strategies is using surface electromyography (sEMG). The sEMG signals are an electrical indication of the neuromuscular

actuation connected with a contracting muscle. Nevertheless, such kinds of signals are extremely complicated, influenced by the anatomical and physiological properties of muscles, the control plan of the peripheral nervous system, and the attributes of the instrumentation utilized to identify and watch them [8]. In recent years, different types of control were tested, such as a brain–computer interface (BCI) [9] or bilateral control [10], but these methods, although performing very well, are not suitable for a wearable and portable version of assistive devices that can help people during their daily activities. Devices to perform a BCI take up entire rooms and are, therefore, not portable, while bilateral control does not allow independent limb movement and, more importantly, presupposes the functioning of the other limb. Conversely, one approach that allows for wearable and portable devices while at the same time being able to identify user intent is based on sEMG signals. These signals are generally sampled at forearm height and classified to recognize a set of gestures that can be performed with the hand. Many studies regarding the classification of sEMG signals have been published in various areas, including robotics [11], prosthetic arms [12], sign language classification [13], and remote-controlled devices [14].

The main objective of the research activity into which this work fits is to develop a portable exoskeleton for hand-impaired subjects, which may help them in their daily activities. The design constraints that the device must meet are focused on making it easy to use during daily activities but, above all, on safety. To be fully safe, a device must not allow movement over the natural range of motion of each digit. The best way to ensure this is to place mechanical stops preventing the structure from crossing such limits. Robustness and ease of use are two necessary characteristics for such a device to be completely wearable. For the mechanical part, it is necessary for the system to be compact, as it has to be worn on the back of the subject's hand, light in weight, so as not to fatigue the patient unnecessarily, and easily scalable for different limb sizes among subjects, in order to reduce the costs of single device production. Furthermore, to reduce costs and allow more people to access this technology, the entire structure was designed to be 3D printed using plastic materials. Promising mechanical results have already been reached [15–17] and with the most recent version of the device (Figure 1) we have reached the possibility of controlling each finger independently. This crucial change opens new challenges in the control strategy. Indeed, the previous solution [18,19] allowed only full opening and closing of the hand through an sEMG classifier.

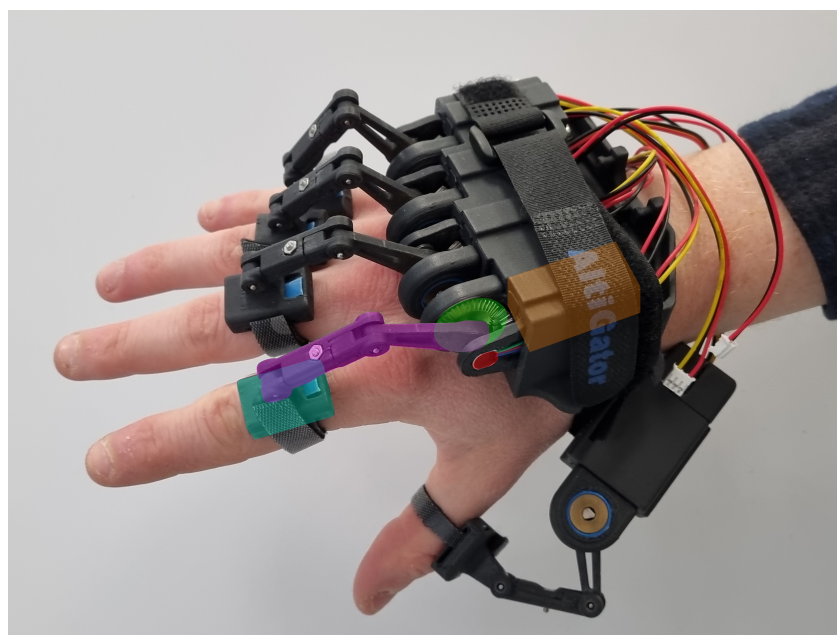


Figure 1. Most recent exoskeleton prototype designed by the Mechatronics and Dynamic Modelling Lab (MDM Lab).

The exoskeleton is of the “rigid” type because it acts on the finger through a “rigid kinematic chain” that can be customized according to the user’s anthropometric dimensions, composed of three links which couple with the kinematic chain formed by the finger’s bones. The kinematic chain actively implements the flexion/extension of the metacarpophalangeal (MCP) joint while passively following its ab/adduction. Therefore, the exoskeleton has 10 degrees of freedom (DOFs), 5 actuated and 5 passive. Each finger of the exoskeleton presents an electric motor (orange in Figure 1) connected to a gear transmission (green in Figure 1) which acts directly on the rigid kinematic chain (purple and cyan in Figure 1). The only point of contact between the kinematic chain is the end effector (cyan in Figure 1) that pushes or pulls the finger to assist the user in the movements. The control loop is closed thanks to an encoder (red in Figure 1) that measures the flexion/extension of the MCP joint. Due to its mechanical architecture, the exoskeleton can also be categorized as a “single-phalanx” because it has only one point of contact with the finger on the medial phalanx.

In this paper, the authors present a new protocol for developing a finger angular position regressor relying on sEMG signals taken on the forearm and finger angular positions measured in the previous instants. To accomplish this, a graphical user interface (GUI) was developed to facilitate the acquisition of the dataset needed to study and develop a deep learning (DL) algorithm capable of predicting the user’s intention. In particular, we use the sEMG data from a Myo armband and the finger position from the encoders placed on the device. The next step will be to directly implement the regressor found with the GUI on the exoskeleton control unit, making the system fully wearable.

The paper is organized as follows. First, in Section 2, a review of the state of the art related to wearable assistive and rehabilitation devices is presented. In Section 3, the control strategy idea description is reported, and then, in Section 3.3, we describe the acquisition protocol carried out through the GUI. Finally, in Sections 4 and 5, some results on different DL models are discussed and compared with each other.

2. Related Work and Paper’s Contribution

Various approaches have been investigated in the field of hand exoskeletons designed for rehabilitative or assistive purposes. Guo et al. [20] presented a soft robotic glove for post-stroke hand function rehabilitation. This glove is a combination of an extendable joint and a rigid part; each interphalangeal MCP joint is actuated by inflating and deflating through a set of micro air pumps and two valves, so each hand can be controlled independently. The control unit proposed in the article was driven by electroencephalography (EEG) signals captured by 14 electrodes on the subject’s head in a predefined and fixed position. Due to its inherent nature, such a system does not lend itself to being a portable system because it requires at least a compressed air tank and a system of pipes and valves, nor is it easy to use by a non-expert subject due to the constraints on sensor placement for EEG signals. Tan et al. [21] instead present a hand-assisted rehabilitation robot based on a master–slave control system. In this configuration, the user wears a glove equipped with flex sensors that collect data. These data are used to estimate the fingers’ angle and position and passed to a specifically designed rehabilitation robot that can be mounted on the patient’s impaired hand. This solution has proven to be an improvement concerning traditional rehabilitation therapy but is unsuitable for daily use because it uses the signal from one hand to control the other, making it impossible to perform different movements with the hands. A portable and wearable version of a hand exoskeleton, called ReHand, was designed by Wang et al. [22]. The ReHand exoskeleton is realized by two modules: one to control the four long fingers and one only to control the thumb. The robot actuation is realized by two DC motors, one for the long fingers and one for the thumb, and a gears combination that generate a continuous force of 5 Nm on the long fingers and a force of 70.56 mNm that can help the user both in extension and flexion of all digits. The user can control the exoskeleton through sEMG signals or voice commands. Although this solution performs well, it does not apply to our case because it uses only two sEMG sensors

which is not enough to discriminate all possible movements of the fingers. A new trend in wearable robotics, particularly hand assistive and rehabilitation devices, is the use of tendon-driven continuum structures. The high structural compliance of continuum robotic structures can improve the working safety of the exoskeletons. Typically, these types of devices are made of a sequence of rigid links connected by wires, the task of which is to emulate human tendons. The actuation of the devices is, therefore, realized by pulling the wires. A good example of this type of assistive device was designed by Delph et al. [23] in 2013. This glove is actuated by five different Bowden cables, each connected to a servo stored in a backpack on the back of the user. A more portable version of a soft robotic glove is presented in [24], where the authors show an actuated glove designed to help people in the grasping movement. The system is actuated by a single brushless motor actuating all the eight DOFs of the glove to ensure portability.

sEMG signals are widely used for controlling hand exoskeletons as they allow measurement of muscle activity in a non-invasive way, and different approaches have been investigated over the past few decades. In a previous work, [18] proposed a new classification method based on sEMG signals from only two sensors, respectively positioned at the extensor and flexor muscle bands. This method allows three hand configurations to be distinguished: rest, open, and closed. These are the fundamental gestures to replicate without losing too much efficacy in activities of daily living (ADLs) assistance [25]. The use of sEMG allows the control system to be not invasive and easy to use for everyone; however, because of the high complexity of sEMG signals, it is almost impossible to create a mathematical model that links these to their effect; indeed, approximately 20 types of muscles are related to fingers' activities, half of which are located in the hand itself (i.e., intrinsic hand muscles), while the other half are located in the forearm (i.e., extrinsic hand muscles) [26].

For these reasons, the most popular approach is related to the use of machine learning (ML) and artificial neural networks (ANNs). There are two major types of supervised ML problems, called classification and regression. In classification problems, the goal is to predict a class label, which is a choice from a predefined list of possibilities, instead in regression tasks, the goal is to predict a real number (or a floating-point number in programming terms) [27]. Whether it is a classification or a regression problem, the basic steps for implementing neural networks are the same. We start with a data acquisition process, which may involve one or more sensors, after that usually a pre-processing of the data is performed, which generally relies on a filtering operation designed to reduce the noise or undesired frequency contributions, next, we can perform a feature extraction consisting of obtaining some characteristics in the time or frequency domain, then, we conclude with the training step where a loss function is minimized in an iterative process. ML algorithms are particular structures able to improve their performance based on the training data. Once we have collected our data, we divide them into three groups: one used for training, one for validation, and the last one for testing. Concerning how to split the data into training, validation, and test subsets, more procedures are possible; many previous studies split the data into 80% training and 20% test [28], without an explicit validation dataset, others prefer to split the data into 70% training, 15% validation, and 15% test [8], and some others follow an equal split in 50% training and 50% test [29]. The decision on the amount of data to be used for training, validation, and test processes should be guided by the desire to reach a valid trade-off between training time, prediction performance, and generalization with respect to new input data.

Mostly, investigated problems solved by sEMG signals and ML algorithms mainly concern hand gesture recognition, so they are classification problems and not regression problems. For example, Bisi et al. [14] use sEMG signals from a Myo armband from Thalmic Lab (some examples are available on GitHub <https://github.com/balandinodidonato/MyoToolkit/tree/master>, accessed on 17 April 2023) and a k-nearest neighbor (KNN) classifier to recognize six different gestures and use them to control a simple differential drive robot. In particular, they collect raw data from a Myo armband and use five sample windows to extract six different

features: mean average value (MAV), simple square integral (SSI), root mean square (RMS), log detector (LOG), and variance (VAR). After feature extraction, they obtain a vector of 48 values for each time window, so they apply a principal component analysis (PCA) to reduce the complexity of the problem. The final algorithm uses only RMS and MAV to identify the six gestures with high accuracy (around 95.29%). Another widely used ML classification algorithm is the support vector machine (SVM) algorithm which may be used to solve several classification problems. SVM maps training examples to points in space so as to maximize the width of the gap between the categories. New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall [30]. For example, in [31], the authors use a 150 ms sample window from which they extract only one feature, MAV, to identify one gesture between fist, finger spread, wave-in, wave-out, pronation, supination, rest, and average.

The latest trends in the development of hand exoskeleton systems (HESs) are moving towards solutions that allow the movements of every single finger independently. This demand arises from the need to make precision movements with the fingers that cannot be performed except by independently actuating the fingers of the hand. For example, grasping an item of cutlery or taking a book from a shelf. To achieve the ability to discriminate each muscle from another, and, thus, activate only part of the fingers, we need highly dense information coming from the forearm muscles. A preliminary solution may be to increase the number of electrodes by placing them in strategic spots in the area of interest. For example, in [28], the authors use six different wireless sEMG sensors to record muscle activity from the shoulder to forearm. Nevertheless, this solution does not apply well to our problem since there remains the complication of knowing where to apply the electrodes to obtain the correct signal. Indeed, our goal is to create a device that is completely usable by everyone, even those without specific knowledge of human anatomy. To solve the electrode placement problem, we can use dense arrays of sEMG sensors like in [32–35]. The high number of electrodes and the proximity of one to another reduce the positioning error when placing the array in the measuring zone but, at the same time, increase the cost and complexity of the system while also making it more difficult to fit. From the state-of-the-art analysis presented, it appears that a good compromise between these two approaches is to use an armband-shaped device equipped with electromyographic sensors, such as a Myo armband. The main advantages are that it is easy to wear and has sufficient sensor density.

Table 1 summarizes some of the main features of the state-of-the-art devices used in assistance and rehabilitation (for the sake of brevity, just those that seemed of most interest to our work are hereafter reported).

Table 1. Main features of some of the state-of-the-art devices for assistance or rehabilitation.

Reference	Actuated Fingers	Independent Fingers	Control System	Type of Therapy
Guo et al. [20]	All	No	BCI	Rehabilitation
Tan et al. [21]	All	Thumb, others	Master–Slave	Rehabilitation
Wang et al. [22]	All	Thumb, others	2-Channel sEMG or Voice	Assistive
Secciani et al. [36]	Long fingers	No	2-Channel sEMG	Assistive
Esposito et al. [37]	Long fingers	No	FMG ¹	Assistive
Yurkewich et al. [38]	All	No	8-Channel sEMG	Assistive
Zhao et al. [39]	All	Thumb, others	1-Channel sEMG	Assistive
Sandison et al. [40]	All	All	Tablet Application	Assistive
Bouterraa et al. [41]	All	All	2-Channel sEMG	Rehabilitation

¹ Force myography.

Another issue related to the training of ML algorithms is the generation of appropriate labels to bind to the input data. A classic approach in classification problems is to establish an acquisition protocol where a predefined sequence of movements is performed at specific time instants, so it is easy to bind input and output. For regression problems, this method cannot be used since the outputs are real numbers, so they can assume infinite values. Therefore, in the case of a regression problem, the output of the ML algorithms are collected simultaneously from one or more sensors. For example, in [42], the authors use an angle sensor to capture the knee angle and tie it to the sEMG signals of the quadriceps. Fazil et al. [43] instead use motion capture to measure the wrist joint angles and link them with sEMG data from below the elbow. Regarding the hand, a widely used device is the Leap Motion Controller, a simple device realized by two near-infrared (NIR) cameras capable of detecting the major joints of the hand. In [13], for example, Leap data were used to recognize the gesture and translate it to the equivalent Indonesian sign language meaning. In this paper, we present a specially designed GUI that facilitates the data acquisition process, including sEMG signals, from the Myo armband, and hand kinematic information from the Leap. By using this interface, we collect a dataset and conduct a performance test to determine if adding a convolutional layer to a DL algorithm improves its functionality in a situation with a limited number of sensors.

3. Methods

3.1. Hardware

The Myo armband (Figure 2) is a commercial device developed and distributed by Thalmic Labs in 2014. The Myo armband is partially adjustable in diameter thanks to its rubber structure that, once positioned at the height of the largest section of the forearm (more specifically at the height of the brachioradialis muscle), can monitor muscle activity with eight pairs of sEMG sensors. The Myo armband is also equipped with a 9-axis inertial measurement unit (IMU) which allows information on the device pose to be gathered. The Myo armband results in a low-cost and high-performance device capable of acquiring sEMG signals at 200 Hz and inertial data at 50 Hz sampling frequency. For the sake of completeness, the sEMG signal bandwidth is 500 Hz, thus requiring a Nyquist sampling rate of 1000 Hz; however, several previous studies have shown that the signals obtained from the Myo armband are still valid [13,14,29,31,44].

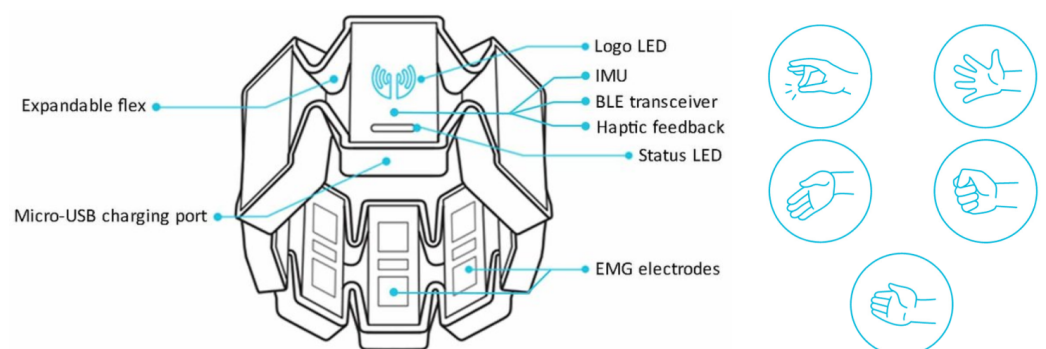


Figure 2. The Myo armband and the gestures recognized by the standard classifier already implemented in the device.

The data acquired by the Myo sensors are sent, via the Bluetooth low-energy (BLE) module embedded into the bracelet, to other electronic devices. Finally, the Myo armband is equipped with batteries, rechargeable via a USB connection, making it fully wearable.

The Leap Motion Controller (Figure 3) from Ultraleap (www.ultraleap.com, accessed on 17 April 2023) is an optical hand-tracking module that captures the movement of users' hands and fingers so they can interact naturally with digital content [45]. It is able to determine the position of the hand and its various elements, such as fingers and joints. Ultraleap provides different language libraries to communicate with the device, including

a Python 2.7 software development kit that has been adapted to work in a Python 3-based environment through the WebSocket interface already implemented by Ultraleap.

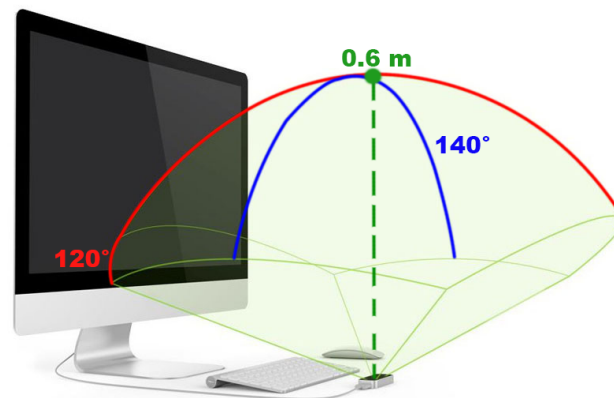


Figure 3. Leap Motion Controller with its range of visibility.

The Leap Motion Controller can provide a large variety of information related to hand dynamics and kinematics. In particular, we are interested in the tips' positions and velocities, MCP, and proximal interphalangeal (PIP) joints' angular positions and velocities of the hand palm. Additionally, with the angular position of the hand joints and the elapsed time between two successive acquisitions, we can also calculate the angular velocity of the joints. The angular position and velocity of each joint have to be expressed with respect to the neighboring proximal joint (i.e., for MCP we use palm and for PIP we use MCP). Once the WebSocket is started, simply connecting to the localhost domain at port 6437, we can receive the information obtained from the device in the form of JSON messages, an extremely powerful tool since they allow access to information by keywords rather than by indexes. In addition to the hardware device, we also use the Ultraleap *Visualizer* software (Figure 4) to have visual feedback on what the Leap actually sees during the acquisition sequence. The *Visualizer* is a proprietary software able to replicate the real hand in a virtual environment using the position of each hand's joints; it displays a variety of tracking data such as the device data rate and can enable/disable the visualization of the camera images.

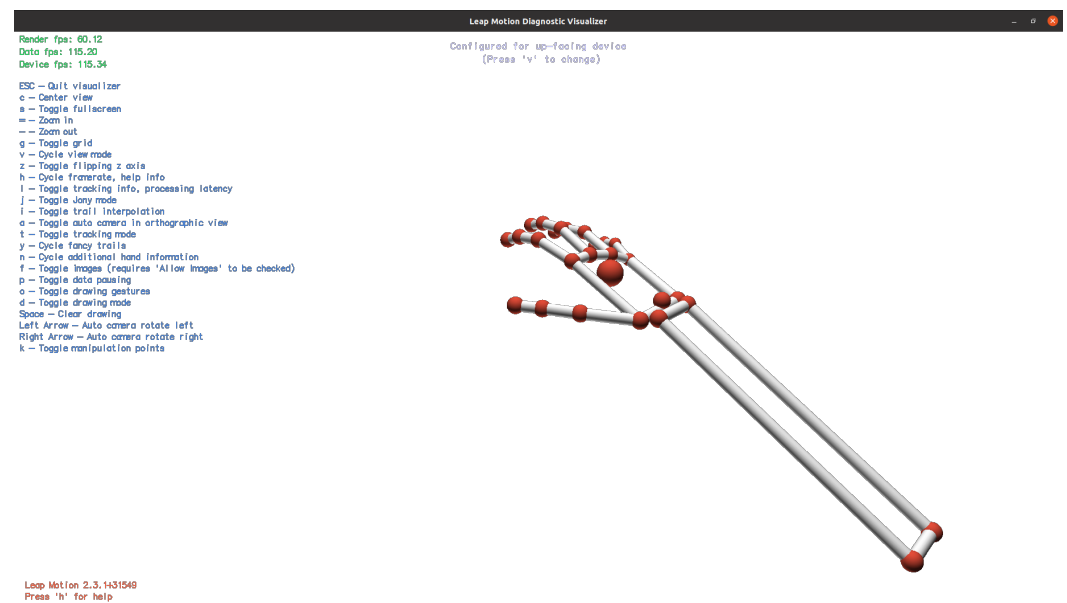


Figure 4. Ultraleap Visualizer interface.

Using the Leap Motion Controller, we are no longer constrained to perform predefined gestures by following on-screen instructions as is generally the case with data acquisition.

Users can theoretically perform any movements they want as long as the hand remains in the field of view of the instruments. However, a specified acquisition protocol is defined to ensure that all the gestures of interest are performed. The user's hand must stay in the field of view of the Leap because the data received from the device represent the output of the regression algorithm during the training process. If this information is missing, the GUI stops the acquisition and warns the user. The Leap Motion Controller firmware identifies each finger joint and in its correspondence, it poses a reference frame. With simple goniometric calculations and compositions of the rotation matrix we can determine the orientation of the joint's frame with respect to the palm's frame. The flexion/extension angle of the first joint of the finger, i.e., the MCP joint, is generally represented as the angle between a frame fixed on the hand's palm and a frame fixed on the MCP joint. In particular, with the frame placed by the Leap Motion Controller on the palm and the MCP, this angle corresponds to the rotation angle along the x-axis.

3.2. Regressor Structure

ML algorithms are particular structures able to improve their performance based on the training data. Once we have collected our data, we divide them into three groups: one used for training, one for validation, and the last one for testing. Both for regression and classification problems, the standard workflow before the training process is divided into three steps: data acquisition, pre-processing, and feature extraction. The data acquisition process will be described in detail in Section 3.3 of this paper when we describe the GUI designed and used for this work. The pre-processing process usually acts as a filter to isolate only the frequency band of interest. In particular, sEMG signals are bio-electrical signals with a low signal-to-noise ratio so the process of amplification and transmission is a critical stage that could increase that ratio. So, the pre-processing filter action is necessary to reduce the initial noise component. In the present application, all of these steps are performed by the Myo armband firmware before sending the data through the BLE. Identification of the region of muscle activation in the sEMG signal is typically based on features extracted from the data both in the time domain and the frequency domain. The most used ones are MAV [8,13,14,31], WL [8,13,31], zero crossing (ZC) [8,13,31], slope sign change (SSC) [31,35], and RMS [8,13,14,31,33]. Since the chosen features will have to be computed online to realize exoskeleton control, and considering that the maximum delay on control can be 300 ms so as not to annoy the user [46], it is necessary to choose features that are not too computationally expensive. For the same reason, another important parameter is the time window used to compute the features. Although the classic approach is overlapping windows, different lengths from 100 ms [47] to 260 ms [48] have been investigated in other works and considered for this one. It is necessary to optimize the length of the windows from which to extrapolate the various features and use only those features that are useful for the ANN algorithm. Given the low sampling rate of the Myo armband, a longer analysis window provides more sEMG data to the control system and is expected to have greater accuracy in deducing the user's intent [49].

However, a longer window increases the delay of the control system, leading to possible annoyances (e.g., functional disturbances, interruptions, hesitations in fluent movement generation, etc.) in the use of the system. While the maximum window length is 300 ms, the minimum length not to obtain a too high variance and bias in the frequency domain features is 125 ms [50]. According to these constraints, and considering an overlapping windowing scenario (Figure 5), we decided to segment the data into 80-sample windows (400 ms), at 200 Hz of the sampling rate of the Myo armband, with 30 samples (150 ms) of overlapping. This way, the delay that was introduced between one control signal and the next is equivalent to the time it takes to acquire 50 new samples (250 ms), leaving 50 ms to the regressor algorithm to generate the command action and to the actual execution.

Different features were considered for this work and are summarized in Table 2. However, to minimize the computational time of the controller, it is not possible, let alone necessary, to compute all features. In fact, by several trials with different combinations of

these features, we have found that the combination of RMS (F_1) and IEMG (F_7) proved to be the best combination in our case study.

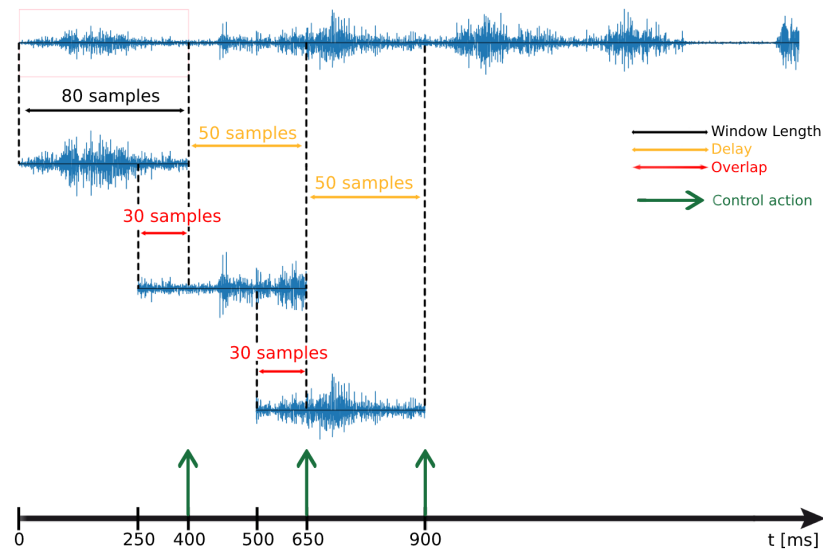


Figure 5. Display of delays in a case of overlapping windows.

Table 2. Tested features.

Feature	Equation
Root Mean Square (F_1)	$\sqrt{\frac{1}{N} \sum_{i=1}^N x_i^2}$
Mean Absolute Value (F_2)	$\frac{1}{N} \sum_{i=1}^N x_i $
Mean Absolute Deviation (F_3)	$\frac{1}{N} \sum_{i=1}^N x_i - \bar{x} $
Waveform Length (F_4)	$\sum_{j=1}^{N-1} x_j - x_{j+1} $
Zero Crossing (F_5)	$\sum_{i=1}^{N-1} f_{zc}(x_i, x_{i+1})^*$
Difference Absolute Standard Deviation (F_6)	$\sqrt{\frac{1}{N} \sum_{i=1}^{N-1} (x_{i+1} - x_i)^2}$
Integral EMG (F_7)	$\sum_{i=1}^N x_i $
Envelope (F_8)	$\max\{x\}$
Fast Fourier Transform (F_9)	$\sum_{k=1}^N x_k e^{-j\frac{2\pi}{N}kq}, q = 1, \dots, N$
Mean Value (F_{10})	$\frac{1}{N} \sum_{i=1}^N x_i$

$$^* f_{zc}(x_i, x_{i+1}) = \begin{cases} 1 & \text{if } x_i \cdot x_{i+1} < 0 \\ 0 & \text{if } x_i \cdot x_{i+1} > 0 \end{cases}$$

Our HES is also equipped with five encoders, one for each finger, that measure the fingers' flexion angle. We, therefore, decided to use not only sEMG signals as inputs to the neural network but also the average of the previous MCP angular positions on the same time window as the sEMG. Providing both sEMG signals and previous finger poses is equivalent to providing the neural network with information about the actual state (position) of the system in addition to the sEMG input. Figure 6 presents a simplified diagram showing the input and output of the regressor.

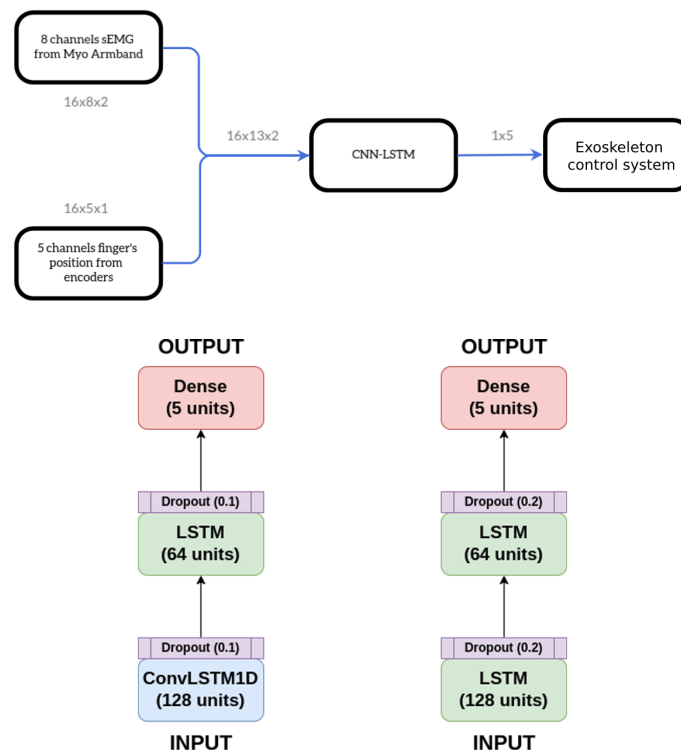


Figure 6. Model structure overviews and specific structures of the two models below.

Many recent works have shown that combining convolutional neural networks (CNNs) and long short-term memory (LSTM) can increase the accuracy of the ANN algorithm [51–55]. LSTM is a particular type of recurrent neural network (RNN) specifically designed to perform well both with short and long term data without running into the problem of the gradient vanishing. In particular, LSTM is very strong in extrapolating the temporal characteristics of the signal while CNNs are particularly capable of discerning spatial properties, so, combining these two algorithms makes it possible to extrapolate both spatial and temporal features. Spatial features allow us to determine which muscle was activated while temporal characteristics provide information on how much and when that muscle was activated, leading to a more complete understanding of the forearm’s muscle activity.

Considering that the model is to be implemented within an STM32 integrated, we have made a very lightweight model consisting of relatively few layers and with a small number of units within each. In particular, we propose a three-layer model:

1. ConvLSTM1D layer: this particular layer, present in TensorFlow V2, already implements a combination of a CNN and LSTM. It is similar to an LSTM layer, but the input transformations and recurrent transformations are both convolutional.
2. LSTM layer: this implements a standard LSTM layer as presented by Hochreiter in 1997 with *tanh* as the activation function [56].
3. Dense layer: this is used to reduce the dimensionality of the previous layer’s output.

The proposed model was compared with a model that does not use a convolutional layer in order to determine how much the former improves on the latter improves its performance.

3.3. Graphical User Interface

The most sensitive part of using DL algorithms is the dataset acquisition for the training process. The training dataset should be as large and heterogeneous as possible so as to provide the neural network model with a sufficient generalizing capability and behave well when new and never-seen-before data are presented to it. Given the crucial

relevance of such an operation, the creation of an interface that can simplify this process has arisen as necessary to make this innovative technology more accessible. The main task of the developed GUI is to collect and label the dataset needed for the regressor training without performing any feature extraction and training process. In the future, an update is planned so that the controller training takes place directly within the interface itself so that the user is provided with the final controller at the end of the acquisition protocol. Nonetheless, in this first version, the interface was just used to facilitate the acquisition of data needed to study which regression method was the most suitable for our problem. The GUI can be seen as being composed of six independent sections, as we can see in Figure 7.

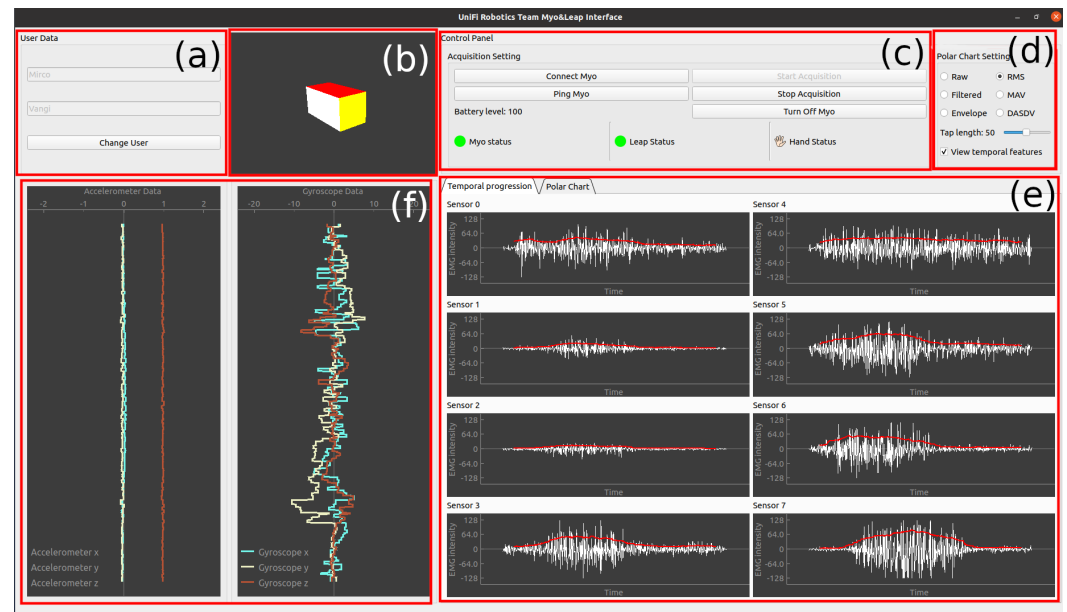


Figure 7. GUI overview. (a) User's name and surname. (b) 3D cube replicating the forearm's pose. (c) GUI commands and connection info. (d) Setting for the polar chart and the second function on the temporal plot. (e) Real-time sEMG evolution in temporal view. (f) Gyroscope and accelerometer data.

In the top left of the GUI (division a) is placed a panel where the user is supposed to insert their name and surname; this information is, thus, used to create a dedicated folder where the data collected during the actual acquisition cycle are saved. The 3D box in division (b) is meant to replicate the pose of a reference frame centered on the correspondence of the Myo armband, providing visual feedback of the forearm pose.

Division (c) presents all the control commands required to enable connection with the Myo armband, start and stop the data acquisition, and, at the end of the process, turn off the Myo armband. Moreover, some status indicators are present to inform the user on the connection status of the devices and whether the Leap Motion Controller is detecting at least one hand. Through division (d), it is possible to choose which feature to display on the polar graph and, as a second feature on the temporal graphs (in red), also to select the size of the window from which to extrapolate the chosen feature. Division (e) is dedicated to showing the real-time sEMG signals collected by the Myo armband. Two different display modes are designed to make it straightforward for the user to understand the signals, and, thus, be able to recognize whether there are any acquisition errors occurring: temporal progression, more comprehensible for non-expert users, and polar representation, most useful in the research field, providing spatial information of the sEMG signals. Lastly, division (f) shows the accelerometer data related to the frame posed on the forearm and cited before.

Is it important to mention that our GUI does not show any of the data captured by the Leap, in fact, it is deliberately chosen to omit the display of data received by the Leap and to use, as visual feedback for the user, the device's own proprietary software called

Visualizer. The acquisition cycle needs to follow a predetermined sequence in the operations to be performed by the user. The first essential step is to enter an identifier (this could be simply first and last name), then, search in the Bluetooth signal range for a Myo armband device and connect to it. Automatically, it is also verified that a Leap device is connected to the workstation. Once both Myo armband and Leap are identified and connected, the user can start the acquisition cycle by pressing the dedicated button on the (c) section of the GUI. As long as sEMG data from the Myo armband and hand data from Leap are collected, the sEMG and IMU graphs are updated at 50 Hz (refresh rate of the most common monitors). During the acquisition cycle, we continuously control whether the connection with the devices is still held and whether the Leap Motion Controller can still individuate a hand in its operational range. If one of those conditions is no longer met, the data acquired are ignored until it is met again; in addition, the final file contains one row indicating that some data were missing at that time. The data acquisition from the Myo armband and the Leap occurs simultaneously at a frequency of 200 Hz, the maximum possible for the Myo. The combined use of the Myo armband and Leap in this way for dataset construction does not force the user to reproduce a predefined sequence of gestures. As long as the hand remains within the range of the Leap, all kinematic information will be captured and linked to the corresponding time windows of the sEMG data.

3.4. Test Methodology

The presented framework, composed of the Myo armband, the Leap, and a dedicated GUI, was tested by following a data acquisition protocol involving performing a series of hand openings, closings, and independent finger movements. In particular, we decided to perform three complete opening and closing movements of all fingers, spaced out by a rest phase, and three complete flexions and extensions of each finger in sequence from the thumb to the small finger. The data were collected by the dedicated GUI and saved in a CSV file, and then used a second time to train and test the neural network.

Two metrics were considered to evaluate the performance and be able to compare the models: mean absolute percentage error (MAPE) (Equation (1)) and R^2 score (Equation (2)):

$$\text{MAPE} = \frac{1}{N} \sum_{i=0}^{N-1} \left| \frac{\hat{y}_i - y_i}{y_i} \right| \quad (1)$$

$$R^2 = 1 - \frac{\sum_{i=0}^{N-1} (\hat{y}_i - y_i)^2}{\sum_{i=0}^{N-1} (\bar{y} - y_i)^2} \quad (2)$$

where \bar{y} is the mean of the observed value y and \hat{y} is the vector of the predicted value. The MAPE is an adimensional metric expressing how close predictions are to ground truth values. The lower the value, the better the prediction. It easily permits comparing among different models and makes it easy to understand if a prediction is good or not at a glance [57–59]. The R^2 score, instead, is a fundamental metric that is used to evaluate the performance of a regression-based ML model. It measures the amount of variance in the predictions that can be predicted by the dataset [60]. The closer the score is to 1, the better the predictions.

Because hand exoskeletons are extremely personal devices, they must be custom-designed from both a mechanical and control point of view. For that reason, we use a dataset generated by considering a single subject performing the acquisition protocol. As can be guessed from the protocol itself, the dataset obtained does not appear to have a high dimensionality. In fact, it is designed to be able to be performed quickly without creating too much annoyance to the user and to allow for rapid re-training of the network over time. We have estimated that the whole acquisition protocol can be performed in two minutes, so the dataset will have around 24,000 samples.

We decided to use 75% of the dataset for training the model and the remaining 25% of the dataset for testing the trained model. During the training process, at each iteration, we

used 20% of the training data to validate the model by controlling the loss-function value to establish whether the model started to overfit the training data.

4. Results

In Figure 8, the predictions performed with double LSTM layers are presented. In blue, the predictions performed by the network are highlighted, i.e., the estimated angular positions of the MCP joints, while in orange, the same quantities measured by the Leap and considered as ground truth are plotted. It is essential to notice that the thumb finger has a larger range of motion (ROM) in abduction compared to the long fingers. The prediction of the thumb's flexion angle is, therefore, more challenging and, at the same time, more critical. The model with only LSTM layers is composed of a first LSTM layer with 128 units followed by a second layer with 64 units; the output layer, instead, is composed of a dense layer with a five-dimension vector as the output of the layer, and, thus, of the ANN.

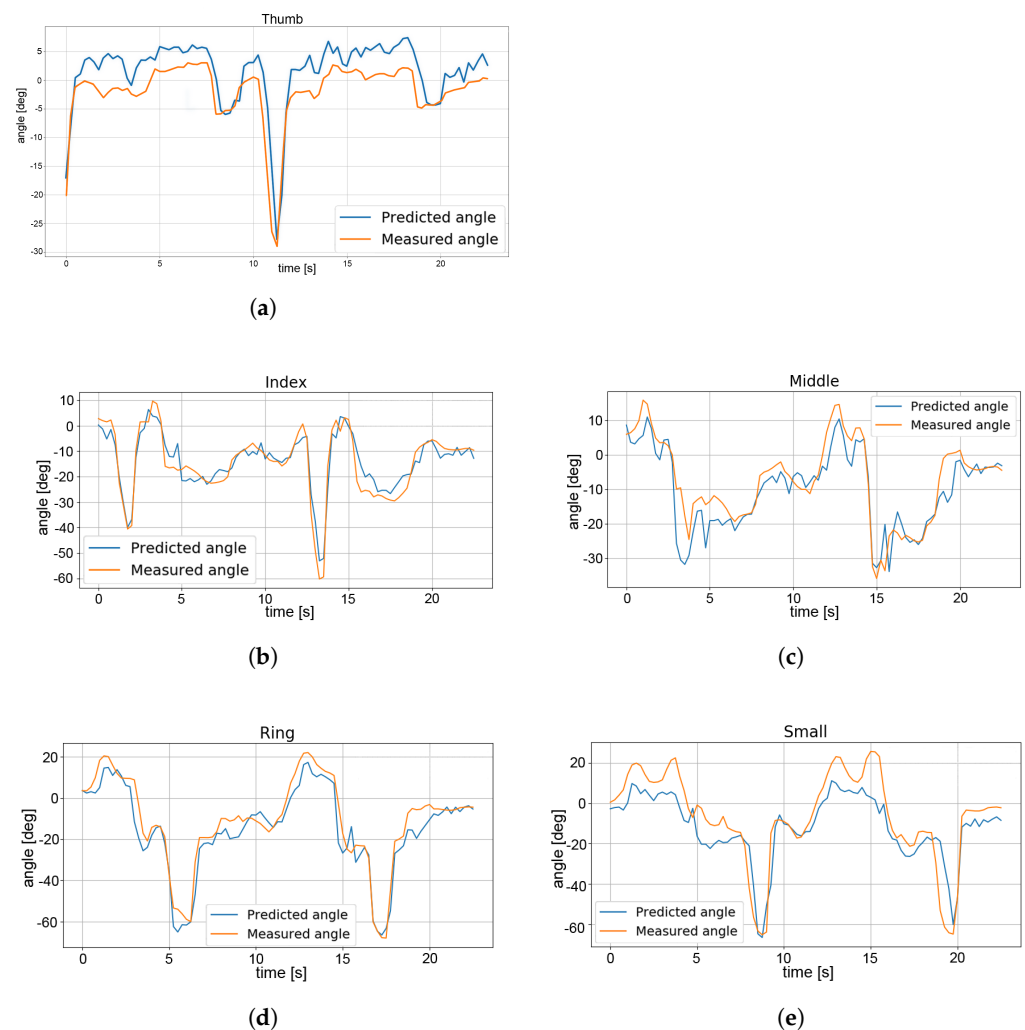


Figure 8. LSTM regressor. Comparison of predicted (in blue) and measured (in orange) finger angle values. (a) Thumb, (b) middle finger, (c) index finger, (d) ring finger, (e) small finger.

Figure 9 shows the prediction outcome from the model with a ConvLSTM1D layer with 128 units and a kernel size of two, followed by a classic LSTM layer with 64 units. The output layer is always realized by a dense layer that outputs a vector of five elements, one for each finger. For these images, the same color notation as in the previous images is used, wherein blue is the prediction and orange is the ground truth.

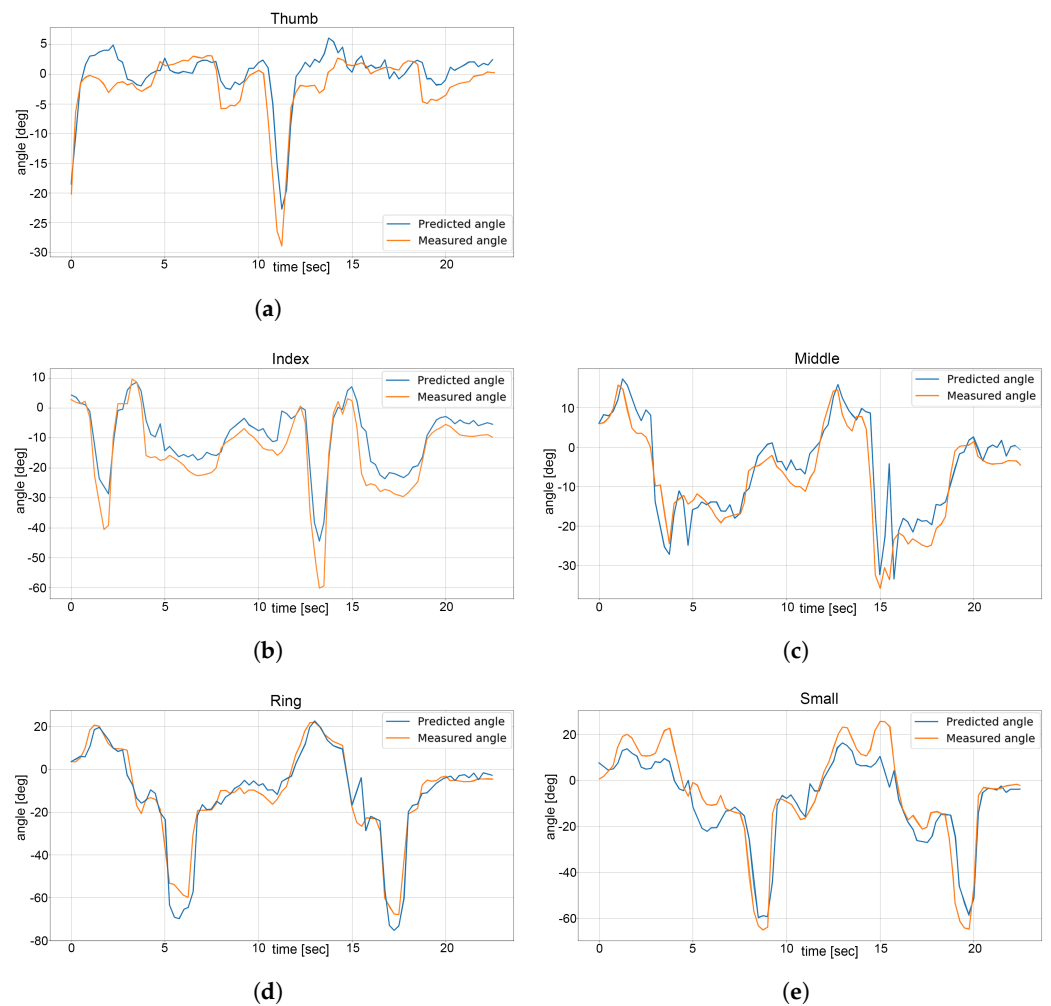


Figure 9. CNN-LSTM regressor. Comparison of predicted (in blue) and measured (in orange) finger angle values. (a) Thumb, (b) middle finger, (c) index finger, (d) ring finger, (e) small finger.

Finally, Table 3 summarizes the results for the LSTM and CNN-LSTM models.

Table 3. Metrics' values for the presented models.

Model Type	Metric	Value				
		Thumb	Index	Middle	Ring	Small
LSTM	MAPE	7.6596	0.5882	2.7816	0.5975	1.1663
	R^2	0.4360	0.8848	0.7692	0.9002	0.7419
CNN-LSTM	MAPE	4.1307	0.4760	1.2120	0.3654	0.9568
	R^2	0.6237	0.8660	0.8323	0.9436	0.8596

5. Discussion and Conclusions

As can be seen from the presented results in Section 4, the model with the CNN-LSTM layer leads to a significant improvement, especially for the thumb. From a quantitative perspective, by considering the MAPE values we can see an improvement for each finger, with an average increase of 36.27%; on the other hand, for the R^2 score, we have an average increase of 13.96%, with a just minimal decrease for the index finger (2.12%). By comparing Figures 8a and 9a we can see how the model with the CNN-LSTM performs better in predicting the angle of thumb flexion from the sEMG value and position data. Furthermore, for the other fingers, we can easily see that the predictions are more accurate in the CNN-

LSTM model with respect to the LSTM-only architecture. Table 4 summarizes the increase in performance of the CNN-LSTM model for each finger.

Table 4. Performance increase for each finger from LSTM model to CNN-LSTM model.

Finger	MAPE	R^2
Thumb	46.07%	43.05%
Index	19.07%	−2.12%
Middle	56.42%	8.20%
Ring	38.85%	4.82%
Small	17.96%	15.86%

In conclusion, in this research activity we have proposed a new type of regressor for the prediction of the finger flexion angles relying on sEMG data captured on the forearm. The proposed algorithm uses just eight sample points for the sEMG signals. A convolutional layer was introduced in the regressor structure to deal with the low number of independent sEMG signals. Consequently, the final structure of the algorithm presents a ConvLSTM1D layer, an LSTM layer, and a final dense layer. To simplify the acquisition of the dataset a specific GUI was designed. Such an interface can be simultaneously connected to a Myo armband and a Leap Motion Controller, devices that are widely used in the research field due to their low costs and ease of use, facilitating the data gathering procedure and making it uniform. In fact, the realized GUI allows for a more straightforward process of acquiring, among many things, sEMGs and angular positions of the fingers. That dataset was then used to train the designed regressor capable of predicting the angular position of the MCP joint to be implemented on a hand exoskeleton with five degrees of freedom, one per finger. The model with the convolutional layer was then compared with a similar model that implements only the LSTM and the dense layer. The ANN model was tested offline and turned out to perform better with respect to the model with just the LSTM model. We can, thus, infer that adding a convolutional layer, which also considers the spatial pattern of sEMG signals, does improve the prediction performance of the neural network in the case of low-resolution sensors. In the case of a large number of sensors, in fact, it is possible to discriminate which muscle is contracting even by sEMG signals alone, due to having a denser quantization of space.

Although the obtained results are very encouraging, there is still some research work to be done. Indeed, this paper presents a preliminary work aimed at becoming aware of what strategies may be effective in the context of hand exoskeleton control, but, for the sake of completeness, it is important to mention that some key steps still need to be achieved before the work can be applied to a commercial device. Firstly, the algorithm was not tested online with the exoskeleton. Secondly, the sEMG signals generated by impaired subjects, which are extremely different in terms of strength, structure, and intensity, shall be analyzed and tested with the proposed regression procedure. Additionally, a further major point to investigate is the behavior of the overall system consisting of the Myo armband and the exoskeleton. In fact, the encoder measurements might differ in precision and accuracy from the angular position measurements provided by the Leap device. Finally, the human–machine interface (HMI) pattern will be studied: in effect, the behavior of subjects wearing the exoskeleton might evolve over time as they becomes more and more aware of the whole system and are, thus, able to generate signals more easily recognizable by the regression model itself.

Author Contributions: Conceptualization, N.S., A.T. and C.B.; methodology, M.V.; software, M.V. and A.T.; validation, M.V. and A.T.; formal analysis, N.S. and A.T.; investigation, M.V.; resources, C.B. and A.R.; data curation, M.V. and A.T.; writing—original draft preparation, M.V.; writing—review and editing, N.S., A.T. and A.R.; visualization, C.B., N.S. and A.T.; supervision, N.S. and A.R.; project administration, A.R.; funding acquisition, A.R. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Data are available upon request.

Acknowledgments: We acknowledge the support of the European Union by the Next Generation EU project ECS00000017 'Ecosistema dell'Innovazione' Tuscany Health Ecosystem (THE, PNRR, Spoke 4: Spoke 9: Robotics and Automation for Health) and of the Fondazione CR Firenze by the R3COVER project.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results.

Abbreviations

The following abbreviations are used in this manuscript:

WHO	World Health Organization
LMIC	Low- and middle-income countries)
sEMG	Surface electromyography
BCI	Brain–computer interface
MCP	Metacarpophalangeal
DOF	Degree of freedom
GUI	Graphical user interface
DL	Deep learning
EEG	Electroencephalography
ADL	Activities of daily living
ML	Machine learning
ANN	Artificial neural network
KNN	k-nearest neighbor
MAV	Mean absolute value
SSI	Simple square integral
RMS	Root mean square
LOG	Log detector
VAR	Variance
PCA	Principal component analysis
SVM	Support vector machine
HES	Hand exoskeleton system
NIR	Near-infrared
IMU	Inertial measurement unit
BLE	Bluetooth low energy
WL	Waveform length
PIP	Proximal interphalangeal
ZC	Zero crossing
SSC	Slope sign change
CNN	Convolutional neural network
LSTM	Long short-term memory
RNN	Recurrent neural network
MAPE	Mean absolute percentage error
ROM	Range of motion
HMI	Human–machine interface

References

1. Ren, Y.; Wu, Y.N.; Yang, C.Y.; Xu, T.; Harvey, R.L.; Zhang, L.Q. Developing a Wearable Ankle Rehabilitation Robotic Device for in-Bed Acute Stroke Rehabilitation. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2017**, *25*, 589–596. [CrossRef]
2. ReWalk Robotics, I. ReWalk-More than Walking. Available online: <https://rewalk.com/> (accessed on 23 February 2023).
3. Bionics, E. Ekso Bionics. Available online: <https://eksobionics.com/> (accessed on 23 February 2023).
4. Bui, K.D.; Wamsley, C.A.; Shofer, F.S.; Kolson, D.L.; Johnson, M.J. Robot-Based Assessment of HIV-Related Motor and Cognitive Impairment for Neurorehabilitation. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2021**, *29*, 576–586. [CrossRef] [PubMed]
5. Barrett, D.; Kamenov, K.; Pearce, E.; Cieza, A. *Global Report on Health Equity for Persons with Disabilities*; World Health Organization: Geneva, Switzerland, 2022.
6. Gimigliano, F.; Negrini, S. The world health organization “rehabilitation 2030: A call for action”. *Eur. J. Phys. Rehabil. Med.* **2017**, *53*, 155–168. [CrossRef] [PubMed]
7. Kabir, R.; Sunny, M.S.H.; Ahmed, H.U.; Rahman, M.H. Hand Rehabilitation Devices: A Comprehensive Systematic Review. *Micromachines* **2022**, *13*, 1033. [CrossRef]
8. Mokhlesabadifarrahani, B.; Gunjan, V.K. *EMG Signals Characterization in Three States of Contraction by Fuzzy Network and Feature Extraction*; Springer: Berlin/Heidelberg, Germany, 2015.
9. Buch, E.; Weber, C.; Cohen, L.G.; Braun, C.; Dimyan, M.A.; Ard, T.; Mellinger, J.; Caria, A.; Soekadar, S.; Fourkas, A.; et al. Think to move: A neuromagnetic brain-computer interface (BCI) system for chronic stroke. *Stroke* **2008**, *39*, 910–917. [CrossRef]
10. Kawasaki, H.; Ito, S.; Ishigure, Y.; Nishimoto, Y.; Aoki, T.; Mouri, T.; Sakaeda, H.; Abe, M. Development of a hand motion assist robot for rehabilitation therapy by patient self-motion control. In Proceedings of the 2007 IEEE 10th International Conference on Rehabilitation Robotics, Noordwijk, The Netherlands, 13–15 June 2007; pp. 234–240.
11. Fleischer, C.; Hommel, G. A human–exoskeleton interface utilizing electromyography. *IEEE Trans. Robot.* **2008**, *24*, 872–882. [CrossRef]
12. Sakib, N.; Islam, M.K. Design and Implementation of an EMG Controlled 3D Printed Prosthetic Arm. In Proceedings of the 2019 IEEE International Conference on Biomedical Engineering, Computer and Information Technology for Health (BECITHCON), Dhaka, Bangladesh, 28–30 November 2019; pp. 85–88. [CrossRef]
13. Wibawa, A.D.; Sumpeno, S. Gesture recognition for Indonesian Sign Language Systems (ISLS) using multimodal sensor leap motion and myo armband controllers based-on naïve bayes classifier. In Proceedings of the 2017 International Conference on Soft Computing, Intelligent System and Information Technology (ICSIT), Denpasar, Indonesia, 26–29 September 2017; pp. 1–6.
14. Bisi, S.; De Luca, L.; Shrestha, B.; Yang, Z.; Gandhi, V. Development of an EMG-controlled mobile robot. *Robotics* **2018**, *7*, 36. [CrossRef]
15. Allotta, B.; Conti, R.; Governi, L.; Meli, E.; Ridolfi, A.; Volpe, Y. Development and experimental testing of a portable hand exoskeleton. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 5339–5344. [CrossRef]
16. Conti, R.; Meli, E.; Ridolfi, A.; Bianchi, M.; Governi, L.; Volpe, Y.; Allotta, B. Kinematic synthesis and testing of a new portable hand exoskeleton. *Meccanica* **2017**, *52*, 2873–2897. [CrossRef]
17. Secciani, N.; Bianchi, M.; Ridolfi, A.; Vannetti, F.; Volpe, Y.; Governi, L.; Bianchini, M.; Allotta, B. Tailor-made hand exoskeletons at the university of florence: From kinematics to mechatronic design. *Machines* **2019**, *7*, 22. [CrossRef]
18. Secciani, N.; Topini, A.; Ridolfi, A.; Meli, E.; Allotta, B. A novel point-in-polygon-based sEMG classifier for hand exoskeleton systems. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2020**, *28*, 3158–3166. [CrossRef]
19. Secciani, N.; Bianchi, M.; Ridolfi, A.; Vannetti, F.; Allotta, B. Assessment of a hand exoskeleton control strategy based on user’s intentions classification starting from surface EMG signals. In Proceedings of the Wearable Robotics: Challenges and Trends: Proceedings of the 4th International Symposium on Wearable Robotics, WeRob2018, Pisa, Italy, 16–20 October 2018; Springer: Berlin/Heidelberg, Germany, 2019; pp. 440–444.
20. Guo, N.; Wang, X.; Duanmu, D.; Huang, X.; Li, X.; Fan, Y.; Li, H.; Liu, Y.; Yeung, E.H.K.; To, M.K.T.; et al. SSVEP-Based Brain Computer Interface Controlled Soft Robotic Glove for Post-Stroke Hand Function Rehabilitation. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2022**, *30*, 1737–1744. [CrossRef]
21. Tan, X.; Luo, Q.; Yang, S.; Jiang, Y. Hand-assisted rehabilitation robot based on human-machine master-slave motion mode. In Proceedings of the 2020 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), Chongqing, China, 29–30 October 2020; pp. 362–365. [CrossRef]
22. Wang, D.; Meng, Q.; Meng, Q.; Li, X.; Yu, H. Design and Development of a Portable Exoskeleton for Hand Rehabilitation. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2018**, *26*, 2376–2386. [CrossRef]
23. Delph, M.; Fischer, S.; Gauthier, P.; Martinez-Luna, C.; Clancy, E.; Fischer, G. A soft robotic exomusculature glove with integrated sEMG sensing for hand rehabilitation. In Proceedings of the 2013 IEEE 13th International Conference on Rehabilitation Robotics (ICORR), Seattle, WA, USA, 24–26 June 2013. [CrossRef]
24. Xiloyannis, M.; Cappello, L.; Khanh, D.B.; Yen, S.C.; Masia, L. Modelling and design of a synergy-based actuator for a tendon-driven soft robotic glove. In Proceedings of the 2016 6th IEEE International Conference on Biomedical Robotics and Biomechatronics (BioRob), Singapore, 26–29 June 2016; pp. 1213–1219. [CrossRef]
25. Montagnani, F.; Controzzi, M.; Cipriani, C. Independent long fingers are not essential for a grasping hand. *Sci. Rep.* **2016**, *6*, 35545. [CrossRef]

26. Perotto, A.O. *Anatomical Guide for the Electromyographer: The Limbs and Trunk*; Charles C Thomas Publisher: Springfield, IL, USA, 2011.
27. Müller, A.C.; Guido, S. *Introduction to Machine Learning with Python: A Guide for Data Scientists*; O'Reilly Media, Inc.: Newton, MA, USA, 2016.
28. Liu, J.; Ren, Y.; Xu, D.; Kang, S.H.; Zhang, L.Q. EMG-based real-time linear-nonlinear cascade regression decoding of shoulder, elbow, and wrist movements in able-bodied persons and stroke survivors. *IEEE Trans. Biomed. Eng.* **2019**, *67*, 1272–1281. [[CrossRef](#)]
29. Tepe, C.; Erdim, M. Classification of EMG Finger Data Acquired with Myo Armband. In Proceedings of the 2020 International Congress on Human-Computer Interaction, Optimization and Robotic Applications (HORA), Ankara, Turkey, 26–28 June 2020; pp. 1–4.
30. Wikipedia Contributors. Support Vector Machine—Wikipedia, The Free Encyclopedia. 2023. Available online: https://en.wikipedia.org/w/index.php?title=Support_vector_machine&oldid=1144271534 (accessed on 29 March 2023).
31. Tepe, C.; Erdim, M. Classification of surface electromyography and gyroscopic signals of finger gestures acquired by Myo armband using machine learning methods. *Biomed. Signal Process. Control* **2022**, *75*, 103588. [[CrossRef](#)]
32. Stapornchaisit, S.; Kim, Y.; Takagi, A.; Yoshimura, N.; Koike, Y. Finger angle estimation from array EMG system using linear regression model with independent component analysis. *Front. Neurobotics* **2019**, *13*, 75. [[CrossRef](#)]
33. Yokoyama, M.; Koyama, R.; Yanagisawa, M. An evaluation of hand-force prediction using artificial neural-network regression models of surface EMG signals for handwear devices. *J. Sens.* **2017**, *2017*, 3980906. [[CrossRef](#)]
34. Jaber, H.A.; Rashid, M.T.; Fortuna, L. Robust hand gesture identification using envelope of HD-sEMG signal. In Proceedings of the International Conference on Information and Communication Technology, Baghdad, Iraq, 15–16 April 2019; pp. 203–209.
35. Zhang, Y.; Chen, Y.; Yu, H.; Yang, X.; Sun, R.; Zeng, B. A feature adaptive learning method for high-density semg-based gesture recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2021**, *5*, 1–26. [[CrossRef](#)]
36. Secciani, N.; Brogi, C.; Pagliai, M.; Buonamici, F.; Gerli, F.; Vannetti, F.; Bianchini, M.; Volpe, Y.; Ridolfi, A. Wearable robots: An original mechatronic design of a hand exoskeleton for assistive and rehabilitative purposes. *Front. Neurobot.* **2021**, *15*, 750385. [[CrossRef](#)]
37. Esposito, D.; Centracchio, J.; Andreozzi, E.; Savino, S.; Gargiulo, G.D.; Naik, G.R.; Bifulco, P. Design of a 3D-Printed Hand Exoskeleton Based on Force-Myography Control for Assistance and Rehabilitation. *Machines* **2022**, *10*, 57. [[CrossRef](#)]
38. Yurkewich, A.; Kozak, I.J.; Ivanovic, A.; Rossos, D.; Wang, R.H.; Hebert, D.; Mihailidis, A. Myoelectric untethered robotic glove enhances hand function and performance on daily living tasks after stroke. *J. Rehabil. Assist. Technol. Eng.* **2020**, *7*, 2055668320964050. [[CrossRef](#)]
39. Zhao, L.; Xie, C.; Song, R. Design and validation of a wearable hand exoskeleton system. In Proceedings of the 2020 5th International Conference on Advanced Robotics and Mechatronics (ICARM), Shenzhen, China, 18–21 December 2020; pp. 559–563.
40. Sandison, M.; Phan, K.; Casas, R.; Nguyen, L.; Lum, M.; Pergami-Peries, M.; Lum, P.S. HandMATE: Wearable robotic hand exoskeleton and integrated android app for at home stroke rehabilitation. In Proceedings of the 2020 42nd Annual International Conference of the IEEE Engineering in Medicine & Biology Society (EMBC), Montreal, QC, Canada, 20–24 July 2020; pp. 4867–4872.
41. Bouteraa, Y.; Abdallah, I.B.; Elmogy, A.M. Training of hand rehabilitation using low cost exoskeleton and vision-based game interface. *J. Intell. Robot. Syst.* **2019**, *96*, 31–47. [[CrossRef](#)]
42. Sun, Z.; Zhang, X.; Liu, K.; Shi, T.; Wang, J. A Multi-Joint Continuous Motion Estimation Method of Lower Limb Using Least Squares Support Vector Machine and Zeroing Neural Network based on sEMG signals. *Neural Process. Lett.* **2022**, *55*, 2867–2884. [[CrossRef](#)]
43. Fazil, M.; Meng, Z.; Kang, J. CNN-based Controller for Multi-DoF Prosthetic Wrist using sEMG Data during Activities of Daily Living. In Proceedings of the 2022 9th IEEE RAS/EMBS International Conference for Biomedical Robotics and Biomechatronics (BioRob), Seoul, Republic of Korea, 21–24 August 2022; pp. 01–06.
44. Arozi, M.; Ariyanto, M.; Kristianto, A.; Setiawan, J.D. EMG signal processing of Myo armband sensor for prosthetic hand input using RMS and ANFIS. In Proceedings of the 2020 7th International Conference on Information Technology, Computer, and Electrical Engineering (ICITACEE), Semarang, Indonesia, 24–25 September 2020; pp. 36–40.
45. Ultraleap Inc. Leap Motion Controller. Available online: <https://www.ultraleap.com/product/leap-motion-controller/> (accessed on 23 February 2023).
46. Englehart, K.; Hudgins, B. A robust, real-time control scheme for multifunction myoelectric control. *IEEE Trans. Biomed. Eng.* **2003**, *50*, 848–854. [[CrossRef](#)] [[PubMed](#)]
47. Yang, J.; Pan, J.; Li, J. sEMG-based continuous hand gesture recognition using GMM-HMM and threshold model. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, Macao, 5–8 December 2017; pp. 1509–1514.
48. Côté-Allard, U.; Fall, C.L.; Campeau-Lecours, A.; Gosselin, C.; Laviolette, F.; Gosselin, B. Transfer learning for sEMG hand gestures recognition using convolutional neural networks. In Proceedings of the 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Banff, AB, Canada, 5–8 October 2017; pp. 1663–1668.

49. Smith, L.H.; Hargrove, L.J.; Lock, B.A.; Kuiken, T.A. Determining the optimal window length for pattern recognition-based myoelectric control: Balancing the competing effects of classification error and controller delay. *IEEE Trans. Neural Syst. Rehabil. Eng.* **2010**, *19*, 186–192. [[CrossRef](#)]
50. Oskoei, M.A.; Hu, H. Myoelectric control systems—A survey. *Biomed. Signal Process. Control* **2007**, *2*, 275–294. [[CrossRef](#)]
51. Huang, D.; Chen, B. Surface EMG decoding for hand gestures based on spectrogram and CNN-LSTM. In Proceedings of the 2019 2nd China Symposium on Cognitive Computing and Hybrid Intelligence (CCHI), Xi'an, China, 21–22 September 2019; pp. 123–126.
52. Karnam, N.K.; Dubey, S.R.; Turlapaty, A.C.; Gokaraju, B. EMGHandNet: A hybrid CNN and Bi-LSTM architecture for hand activity classification using surface EMG signals. *Biocybern. Biomed. Eng.* **2022**, *42*, 325–340. [[CrossRef](#)]
53. Bao, T.; Zaidi, S.A.R.; Xie, S.; Yang, P.; Zhang, Z.Q. A CNN-LSTM hybrid model for wrist kinematics estimation using surface electromyography. *IEEE Trans. Instrum. Meas.* **2020**, *70*, 1–9. [[CrossRef](#)]
54. Chen, L.; Fu, J.; Wu, Y.; Li, H.; Zheng, B. Hand gesture recognition using compact CNN via surface electromyography signals. *Sensors* **2020**, *20*, 672. [[CrossRef](#)]
55. Nahid, N.; Rahman, A.; Ahad, M.A. Deep learning based surface EMG hand gesture classification for low-cost myoelectric prosthetic hand. In Proceedings of the 2020 Joint 9th International Conference on Informatics, Electronics & Vision (ICIEV) and 2020 4th International Conference on Imaging, Vision & Pattern Recognition (icIVPR), Kitakyushu, Japan, 26–29 August 2020; pp. 1–8.
56. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
57. Ahmad, S.; Latif, H.A. Forecasting on the crude palm oil and kernel palm production: Seasonal ARIMA approach. In Proceedings of the 2011 IEEE Colloquium on Humanities, Science and Engineering, Penang, Malaysia, 5–6 December 2011; pp. 939–944. [[CrossRef](#)]
58. Tiwari, S.; Sabzehgar, R.; Rasouli, M. Short Term Solar Irradiance Forecast Using Numerical Weather Prediction (NWP) with Gradient Boost Regression. In Proceedings of the 2018 9th IEEE International Symposium on Power Electronics for Distributed Generation Systems (PEDG), Charlotte, NC, USA, 25–28 June 2018; pp. 1–8. [[CrossRef](#)]
59. Pant, J.; Sharma, R.K.; Juyal, A.; Singh, D.; Pant, H.; Pant, P. A Machine-Learning Approach to Time Series Forecasting of Temperature. In Proceedings of the 2022 6th International Conference on Electronics, Communication and Aerospace Technology, Coimbatore, India, 1–3 December 2022; pp. 1125–1129. [[CrossRef](#)]
60. Wikipedia Contributors. Coefficient of Determinatio—Wikipedia, The Free Encyclopedia. 2023. Available online: https://en.wikipedia.org/w/index.php?title=Coefficient_of_determination&oldid=1148160373 (accessed on 22 May 2023).

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.