POLITECNICO
MILANO 1863

SCUOLA DI INGEGNERIA INDUSTRIALE
E DELL'INFORMAZIONE

Exploring novel interaction strategies in live music performances based on muscle signals and deep learning.

Tesi di Laurea Magistrale in
Music and Acoustic Engineering - Ingegneria della Musica e dell'Acustica

Davide Lionetti, 10751438

Advisor:
Prof. Massimiliano Zanoni

Co-advisors:
Ing. Paolo Belluco

Academic year:
2022-2023

Abstract: The objective of this paper is to propose an innovative interaction method based on the conjunction of electromyographic signals and deep learning analysis, integrated into a Digital Musical Instrument for guitarists. This interaction strategy exploits the correlation between the musicians' emotional state and their muscular activity, monitored using surface Electromyography (sEMG) wearable sensors. The proposed method investigates how to effectively adapt the guitar sound by dynamically tracking musical intentions through an analysis of muscular contractions. To accomplish this, a Recurrent Neural Network (RNN) based on Bidirectional Long-Short Term Memory (BLSTM) has been developed to interpret sEMG signals. The musicians provide their gestural vocabulary, each associated with a corresponding pedalboard sonic preset, this association is used to train the gesture classification network; We collected from multiple musicians various sEMG acquisitions (related to popular guitar techniques) in order to create a training dataset; it will be published to support similar applications. The selection of the most effective features, in synergy with the best set of muscles, was conducted to optimize the learning rate of the gesture recognition model. The digital signal processing is carried out with Max/Msp 8, where we define the guitar pedalboard. Finally, an evaluation strategy has been presented, involving the collection of feedback from an expert guitarist through a questionnaire; the goal of this work is to help other researchers integrate muscle signals into artistic performance by proposing a protocol for mapping sEMG with sound.

Key-words: Human Computer Interaction, Surface Electromyography, Gesture Classification, Digital Musical Instrument, Bidirectional LSTM

1. Introduction

Embodied engagement with music is a key element of musical experience, and the gestural properties of musical sound have been studied from multiple disciplinary perspectives, including Human-Computer Interaction (HCI), New Interface of Musical Expression (NIME), musicology, and the cognitive sciences [22].In recent years, the rapid evolution in wearable devices aided by machine learning has introduced novel interaction strategies in today's live music performances. Exploring new gestural interaction strategies is the core for establishing artistic practices where the performer's body is deeply engaged in forms of corporeal interplay with the music by means of motion and physiological sensing [53]. In this scenario, Digital Musical Instruments (DMIs) are based on the creation of action–sound mappings, in which the relationships between the physical energy of the input action may not necessarily correspond to that of the output sound [15], exploiting some Gestural Interface (GI). A GI is a device that measures bodily behaviors (gestures) to manipulate digital media [44]. In this research, Gestural Interaction Design (GID) has been tackled by using a custom wearable sensor made by our partner, LWT3. Using it, we investigate the concepts of *musical intention* in relation to Electromyography (EMG) for DMIs application [51]. EMG is employed in the biomedical and HCI fields as a highly sensitive way to capture human movement and has been used as a signal suitable to sense musical gestures[52, 62]. An EMG signal measures the voltage difference generated by the motor units within muscles during voluntary or involuntary muscle contractions, providing insights into the neuromuscular activation patterns and muscle function. in the normal EMG investigation, the muscles signal of the patient are sensed by skin-inserted needles. In this study, we employ surface electromyography (sEMG), a non-invasive technique that involves the placement of electrodes on the skin to measure muscle activity. we leverage the sEMG signals for modeling the musician's isometric muscular activity which does not translate into movement; sEMG allows the classification of motionless gestures, such as different levels of contraction while performing the same guitar technique. Based on a systematic literature review, we found no attempt to control guitar pedalboard with the musician's muscle signal, and the majority of the research that used sEMG for DMIs development is based on forearm muscles traced by a GI called Myo Armband [38]. A related example is the 'air' guitar, a study by Erdem et al. [17] investigating the development of an air guitar DMI (via ML methods using Myo armband). Due to the lack in the market of affordable sEMG GI, one of the objectives of this paper is to introduce a new wearable sEMG sensor board fully developed in the LWT3 laboratory. Thanks to this innovative GI, we have the possibility to evaluate the impact of a *full-body muscle selection* in the design of a DMI. With this wearable device, we studied the most activated muscles during guitar playing, by proposing a set of arm muscles that maximize grading performance; To guarantee an integrated user experience we design the pedalboard using a well-known (among musicians) sound programming language called *Max/Msp*, with a dedicated Graphical User Interface, that allows an intuitive interaction with sound effects.

In the following sessions, we present the proof of concept of a sensor-augmented system for electric guitar, based on a Bidirectional Long Short Time Memory (BLSTM) recurrent neural network, which adapts and customizes the sound accordingly to an sEMG analysis; it maps the user's muscle activation into the guitar effects' parameters, automatically adapting the sound with the detected musical intension. The musician provides a number of sounds presets by labeling each one with a specific gesture to apply training based on supervised learning; the smooth change between the vocabulary of gestures creates a novel overlay of effects that aims to enhance creativity and broaden the compositional tools palette.
All the development process was conducted together with an advanced level guitarist who takes part in multiple acquisition sessions guiding all the experimental stages explained in the last section: the sound to intentions mapping, the design of the digital pedal-board with Max/Msp, the choosing of the gesture vocabulary and the muscle selection. This article is structured as follow, Section 2, 3.1 present the state of the art and the theoretical background. Section 4 presents all the stages of the protocol design. Section 5 presets all the experiments, the results and the evaluation strategy. Section 6 highlight the conclusions and future works.

2. State of the Art

2.1. Digital Musical Instrument

In the realm of musical instrument craft industry, each generation of luthiers has harnessed the technological advancements of their time to produce innovative creations. Just as Antonio Stradivari utilized the latest discoveries in varnishes, glues, and woods during the 1700s to craft his extraordinary violins, contemporary luthiers have the opportunity to embrace cutting-edge digital technologies. The emergence of digital lutherie [31] calls for a combination of skilled craftsmanship, profound understanding of the materials employed, and artistic sensibility, mirroring the traditional craftsmanship required for acoustic instruments. With the integration of electronics into luthierie, a new class of instruments known as digital musical instruments (DMIs) has been invented, leading to the development of various applications [12]. DMIs extend conventional instruments with sensors, actuators, and digital signal processing techniques in an effort to balance technical and artistic novelty with established playing techniques. Sensor-based augmentations consist of enhancing the instrument with an interface composed of sensors dedicated to the tracking of performer's gestures, which are repurposed into electronically generated sounds via mapping techniques. Augmented Musical Instruments (or AMIs) are created by embedding sensors and actuators on traditional instruments, allowing the performer to control and modify the instrument's sonic output, introducing DSP controllers in the physical instrument to ease the interaction between the musician and sound effects. In the literature, there are some previous attempts related to the creation of augmented guitars able to introduce a higher level of embodiment compared to the traditional instrument [56]. Among the most interesting, the *Moog Guitar* is an electric guitar with onboard sliders that control augmentation of the guitar's traditional sound by sending electromagnetic energy into strings trying to convert an electric guitar into a Moog synthesizer. MacConnell and colleagues presented the *RANGE Guitar* as an AMI built with an electric guitar and variable (membrane) potentiometers, non-invasively installed on the instrument. The embedded processor receives the membrane potentiometers data to control a guitar effect patch programmed in Pure Data[35]. Traditional audio effect units and commercial DSP solutions tend to disregard the importance of intuitive control interfaces that take advantage of the guitar player's natural performance technique, forcing the musician to interact with musical parameters by way of non-musical gestures: turnin a knob or adjusting a fader [20]. This conflicts with the guitarist's normal gestural interaction, fails to convey the exact musical intention, and can even act as a distraction for the musician [37]. In recent years, the concept of DMI has been expanded to include a new class of AMIs called *Smart Musical Instrument* (SML), first presented by Turchet and collegues in [57] and deeply described in [55]. Such instruments encompass different strands of technologies, including Internet of Things (IoT)[10] (e.g., wireless sensor networks), networked music performance systems , sensors and actuator augmentations typical of augmented instruments, and embedded intelligence tailored to support real-time audio and sensor processing. Smart instruments are instances of *Musical Things*, within the *Internet of Musical Things* (IoMusT) paradigm [58], which refers to the network of interoperable devices dedicated to the production and/or reception of musical content. The IoMusT ecosystem gathers devices and services that connect performers and audiences to support performer-performer and audience-performers interactions. Such interactions might be both co-located and remote, by leveraging, respectively, wireless local area networks and the Internet. The most intresting (and one of the first SMI never built) is the *Sensus* Smart Guitar[1] developed by MIND Music Labs3 (see Figure 1). This hybrid electro-acoustic guitar, described in [57], consists of a hollow body guitar augmented with multiple sensors embedded in various parts of the instrument, on-board processing, a system of multiple actuators attached to the soundboard, and interoperable wireless communication (using state-of-the art protocols for wireless transmission and reception such as Wi-Fi and Bluetooth, as well as for exchange of musical data such as MIDI and OSC).

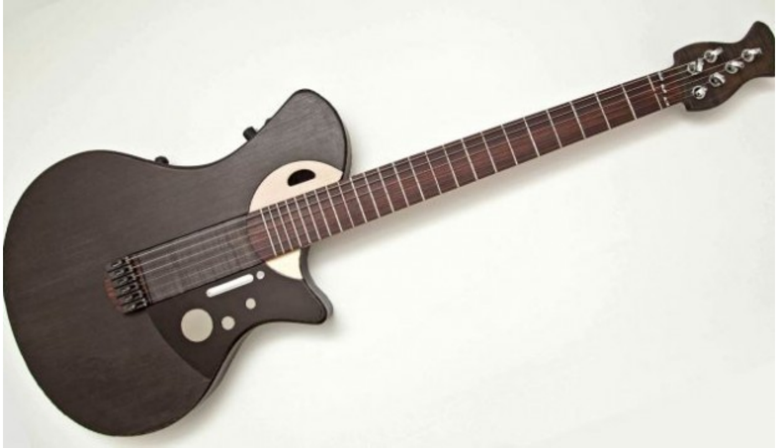


Figure 1: The *Sensus* Smart Musical Instrument

2.2. Wearable Devices

The wearable device domain is being actively researched for the sake of enhancing ease of use, comfort, and non-invasiveness of monitoring physiological signals and sometimes psychological or emotional state, which can be detected by analyzing biometrical data from different sensors. Wearable devices include any device mounted on the body and can capture noninvasive signals from the human body through the use of different types of sensors. Examples of wearable devices are smart watches, armbands, chest straps, shoes, helmets, glasses, lenses, rings, patches, textiles, and hearing aids [47].

There are numerous well-known signals that are read from the human body in literature to identify the vital signs and other information about the health or mental state of the subject. Examples of these sensors include skin temperature [50] and electrodermal activity (EDA) sensor used to record the skin conductance that varies with the sympathetic state of the subject [16]. Other examples include an electrocardiogram (ECG) sensor to capture

[1] Youtube Video demostraction of the Sensus smart guitar: `https://www.youtube.com/watch?v=fqzEQnsSIoY`

electrical changes in the skin corresponding to heartbeats [25]. To capture features of the electrical activity of the brain, the health of muscles and the nerve cells, electroencephalogram (EEG) and electromyography (EMG) sensors are used [13, 30]; these sensors can also be used with amputees, who cannot execute the movements, but have the intention to do so [60]. Bio-signals are also fused with other sensing methods, such as Inertial Measurement Units (IMU), small, low-cost, highly portable devices that incorporate accelerometers and gyroscopes. When these devices are paired with magnetometers, the resulting arrays are also known as Magnetic, Angular Rate and Gravity (MARG) sensors. These sensor arrays allow the tracking of acceleration, rotational velocity and orientation relative to the earth's magnetic field of whatever they are attached to. They are used extensively in aviation, robotics and HCI. Their increasing affordability and small size have made them a very common feature of mobile and wearable devices and other consumer electronics. They are commonly referred to as 9DoF (9 Degrees of Freedom) sensors, as they comprise three tri-axis sensors, resulting in a total of nine sensitive axes: 3D accelerometers, 3D gyroscopes, and 3D magnetometers. EMG with IMU is one of the most explored fusion techniques. A reason for this could be the commercial availability of this combination with the Delsys EMG bio-amplifiers [27] and the Myo armband by Thalmic Labs. Whereas the Myo armband is designed to be used as an affordable interaction system for the HCI sector, the Delsys sensor is intended for medical research and is unaffordable for wearable device development. In [21], Gijsberts et al. utilized the NinaPro database [4] to demonstrate the need for the accelerometer data in decoding the human hand gestures. They concluded that the highest accuracy is obtained when both modalities are integrated with a multi-modal classifier.

Following the technological advances in the design of system-on-chip (SoC), the development and use of wearable devices have achieved high growth rates. The increasing adoption of smart wearable technology products among consumers is driving industry growth. The global wearable technology market was valued at USD 61.30 billion in 2022 and is expected to expand at a compound annual growth rate (CAGR) of 14.6 from 2023 to 2030 according to the wearable technology market industry report by Grand View Research [2]. Despite this growth, there has been only one commercial armband based on the combination of sEMG and IMU for HCI application, quoted above (i.e. the *Myo armband*) that is currently *out of production*. There are no affordable wearable devices on the market at the moment that use sEMG meant for HCI applications. This research aims to mitigate this lack in the market by proposing a case study based on a sEMG sensor board designed at the LWT3 lab, with plans to turn it into a real wearable device for HCI applications.

Based on the systematic literature review conducted in [30], it was found that among the 27 analyzed systems focused on sEMG hand gesture recognition, 21 of them utilized the - at that time - commercially available device Myo armband, with a sampling rate of 200 Hz. The remaining six systems employed custom-made devices, which shared a design similar to the Myo armband, and were configured with a sampling rate of 1000 Hz.

The Myo armband is composed of 8 EMG surface dry sensors. These sensors measure the electrical activity of the muscles of the forearm at a sampling rate of 200 Hz (Fig. 2- 1a) with 8 bits of resolution for each sensor. The forearm muscles are responsible for the movements of the different parts of the hand (Fig. 2- 1b). The Myo transmits its measurements to the computer via BLE (Bluetooth Low Energy). Additionally, the Myo contains an inertial measurement unit (IMU) with 9 degrees of freedom (accelerometer, gyroscope, and orientation in the x, y, and z axes). Finally, the Myo comes with a proprietary system for recognizing 5 gestures of the hand: pinch, fist, open, wave in, and wave out (Fig. 2- 1c) [9].

[2] Grand View Research wearable technology trends analysis: `https://www.grandviewresearch.com/industry-analysis/wearable-technology-market`

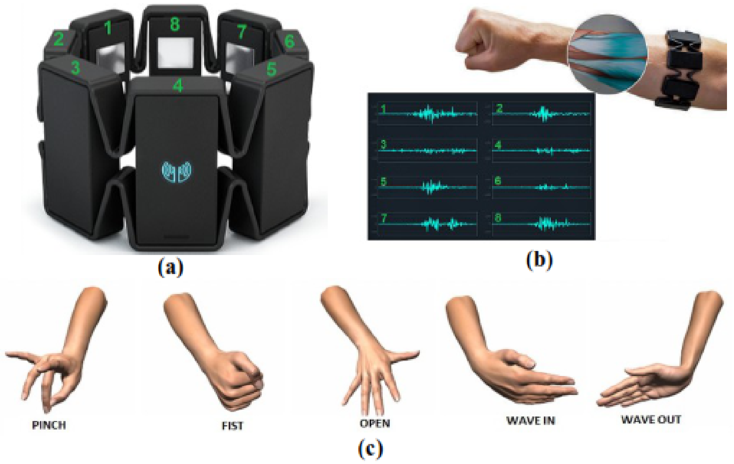


Figure 2: Myo armband: (a) location of the surface EMG sensors on the device, (b) placement of the armband on the forearm (the sensor with the blue logo goes on the posterior part of the forearm), and (c) gestures that are recognized by the proprietary software of the Myo.

2.3. sEMG Analysis Aided By Machine Learning

Most of the studies in the literature, concerning electromyography (EMG), investigate pattern recognition criteria based on Machine Learning. The main goal in this field is to improve the performance of myoelectric control for upper limb prostheses with respect to current clinical approaches based on direct control. One of the first attempts was made in 1993 by William Putnam who proposed a real-time computer control system based on a neural network for pattern recognition of EMG from physically disabled users [42]. In recent years EMGs has been extensively utilized also outside of clinical applications in the domain of human-computer interaction (HCI). An input device developed using EMGs is a natural means of HCI, because the electrical activity induced by the human's arm muscle movements can be interpreted and transformed into computer's control commands. The robust and accurate decoding of information from EMG signals represents a critical challenge, addressed by numerous types of research aided by deep learning techniques: hand gesture classification [33], [36], silent speech recognition [32], [29], stroke rehabilitation [41] and robot arm control [3]. Buongiorno et al. [14] wrote a brief survey about DL in EMG processing that includes also other applications such as sleep stage identification and emotion classification. For a further review of all the major methods used for EMG classification in the field of HCI you can refer to this systematic literature review [1].
Recent studies have investigated the role that gesture representation has in sound-action mapping to build guitar DMIs. A study by Erdem et al. [18] investigating the development of an air guitar DMI (via ML methods) focused on excitation (Excitation is a phase where there is energy transfer between a music performer and a musical object. The excitation phase is preceded by a prefix (movement trajectory towards the point of contact with the musical object) and followed by a suffix (movement away from the musical object)) actions when implementing sound-action mapping for the air guitar. The authors used several excitation categories to investigate whether action-sound gesture coupling can be used to create action-sound mapping for a novel DMI (air guitar) which does not rely on a physical controller (i.e., a physical instrument), using EMG data from the Myo GI to do so. The study used a recurrent neural network (RNN) and trained it with data from semi-professional musicians/participants to predict guitar gestures. The authors then created an RNN model by mapping sEMG data to the RMSof the instrument's audio signal and producing a synthesised guitar sound (i.e., a physical model) [45].

3. Theoretical Backgroud

3.1. Surface Electromiography

Nowadays, there are various technologies available for capturing the movement of the body. IMU are currently the most frequently used devices to do so, thanks to their availability on the market and modest price. However, they are still far from being optimal. Although accelerometers are widely used for capturing body movement due to their affordability and availability, they are not well-suited for measuring isometric strain, as this type of muscle contraction does not result in any movement. Among other modalities for capturing movements, there is the electromyography (EMG), a technique that derives from the medical field and is able to recognize the electrical activity produced by skeletal muscles. The *surface* Electromyography (sEMG) technique allows non-invasive detection of the electrical activity of the muscles by placing electrode pads on the skin with minimal invasiveness. It is called "surface" because in the normal EMG investigation, the muscles of the patient are reached by needles inserted directly into the muscle fibers. The measured signal is a composition of the electric activity from the nearest muscular tissue, which is related to the contraction of the muscles.

Thus, sEMG is used to record the electrical activity of skeletal muscles via surface sensors, capturing the electrical potentials generated during two distinct states of a muscle. In the first state, when the muscle is at rest, individual muscle fibers exhibit an electric potential of approximately -80 mV [60]. The second state occurs when the muscle contracts, leading to the generation of electric potentials within motor units (MUs), comprising muscle fibers and motor neurons. These electric potential differences arise when a motor neuron triggers a neuromuscular junction by transmitting two intracellular action potentials in opposite directions. These potentials propagate through depolarization and repolarization of each muscle fiber [46]. The summation of intracellular action potentials from all muscle fibers within a motor unit is known as a motor unit action potential (MUAP). Consequently, during muscle contraction, the EMG signal represents a linear combination of multiple MUAP trains [60]. There are three types of muscle contractions:

1. **Isometric** contractions: refer to muscle contractions in which the length of the muscle remains constant while tension is generated. Isometric contractions can be detected through sEMG signals: a constant level of muscle activity without any visible movement.
2. **Isotonic** contractions: involve muscle contractions in which tension remains constant while the length of the muscle changes. Isotonic contractions can be detected through sEMG signals, at a level of muscle activity corresponding to the change in muscle length.
3. **Isokinetic** contractions: are a type of muscle contraction in which the muscle generates constant tension throughout its range of motion at a fixed velocity. These contractions are usually achieved with the aid of specialized equipment. Isokinetic contractions can be detected through sEMG signals, and muscle activity will show a constant level of intensity throughout the range of motion.

The sensing of an sEMG channel requires at least two electrodes, for measuring the differential voltage, and a third one as a DC reference. sEMG signal is composed of two main parts - the baseline period and the contraction period - and ranges from -5mV to 5mV. The electrodes are built to optimize the electrode-skin impedance, and decrease the crosstalk of adjacent units, through specific materials. Due to the stochastic, nonlinear and nonstationary nature of this type of signal, it is impractical to analyze the raw myoelectric signals, and that is why the typical procedure is computing some low-level features before the classification. We now present the most common low-level features for sEMG analysis, which can be classified into three categories: time domain, frequency domain, and time-frequency domain. We have considered only the first category because of their computational simplicity, suitability for real-time application, and wide use in research and clinical practice; Time Domain (TD) features are widely utilized due to their strong performance and computational efficiency, as they do not necessitate transformation and consequently demand minimal computational time. Among the prominent TD features frequently employed, the following four have demonstrated particular significance in sEMG analysis accordingly to [48].

1. Root Mean Square (RMS), it is related to the constant force and non-fatiguing contraction. It relates to standard deviation, which can be expressed as

$$RMS = \sqrt{\frac{\sum_{k=1}^{N} x_k^2}{N}}. \tag{1}$$

   where N denotes the length of the signal and x_n represents the EMG signal in a segment n. In the realm of time domain (TD) analysis, RMS emerges as the primary representative feature based on signal amplitude.

2. Waveform Length(WL) : Waveform length (WL) is the cumulative length of the waveform over the time segment. WL is related to the waveform amplitude, frequency and time. It is given by

$$WL = \sum_{k=1}^{N-1} |x_{n+1} - x_n| \tag{2}$$

3. Willison amplitude (WAMP) is the number of counts for each change in the EMG signal amplitude that exceeds a predefined threshold where threshold value is 10 mV. It is defined as Equation 3 is the same as

before, but with just one label:

$$WAMP = \sum_{i=1}^{N-1} \begin{cases} 1 & \text{if } |x(i) - x(i+1)| > \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{3}$$

WAMP is related to the firing of motor unit action potentials (MUAP) and the muscle contraction level, the threshold is set to 5 mV accordingly to the best threshold evaluation presented in [39].

4. Zero crossing (ZCR) is the number of times that EMG signals crosses zero in a window of lentgh N. The threshold value is 20 mV. It can be formulated as .

$$ZCR = \frac{1}{2(N-1)} \sum_{i=1}^{N-1} \begin{cases} 1 & \text{if } x(i) \cdot x(i+1) < 0 \text{ and } |x(i) - x(i+1)| > \text{threshold} \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

ZCR is related to Slope Sign Change. These features provide a rough estimate of the properties in frequency domain.

The feature with the highest noise rejection ratio was found to be the WAMP (equation 3) accordingly to EMG feature selection presented in [39] with a threshold that has to be set as a percentage of full range during a calibration phase. However, research on EMG signal processing showed that there is not a single best feature extraction process or formula, and that it largely depends on the data and required characteristics of the system. EMG signals can be used for a variety of applications including clinical applications, HCI and interactive computer gaming.

3.2. Artificial Neural Network

Mathematical modelling of cutting processes is very important, not only for understanding the nature of the process itself, but also for planning and optimizing the machining operations. Nevertheless, hard machining involves many complex and nonlinear relationships between different variables and parameters. Modelling of these relationships is a difficult task but some artificial intelligence based tools, have been proved their ability for matching complex non-linear relationships. The most popular and deeply studied techniques in soft computing are the artificial neural networks. A Feed Forward Neural Network (FFNN) is the most basic type of ANN, It has only forward connections in between the neurons, as shown in Figure 3. In this group are included multilayer perceptrons (MLP), radial basis function networks (RBF) and self-organizing maps (SOM).

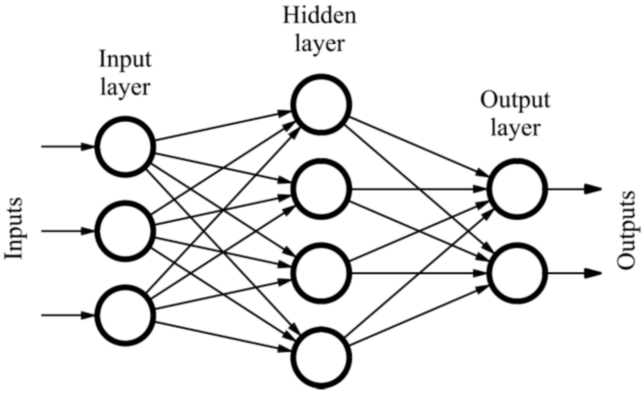


Figure 3: Example of a feed-forward neural network.

Artificial Neural networks (ANN) have a rich history that dates back to the 1940s, arousing as an attempt to model the human brain structure and functioning. They are inspired by mathematical models of individual neurons and interconnected networks of neurons, mimicking the way the human brain functions. The network is composed for several simple units, called neurons, arranged in certain topology, and connected with each other. Neurons are organized into layers. Depending upon their position, layers are denominated input layer, hidden layer or output layer. A neural network may contain several hidden layers. In recent years, deep neural networks have gained significant attention due to their exceptional performance across various research domains,

as extensively discussed in [34]. Deep learning (DL) entails the acquisition of abstract representations of input data through the utilization of multiple hidden layers. The deeper a network is, the larger the number of hidden layers it includes. With the ever-increasing availability of data, ANNs have become a dominant and popular technique for machine learning tasks in the recent past. Over time, numerous network structures have been proposed to effectively match different types of input data such as images, video or sound; The most used are recurrent neural networks (RNN), and convolutional neural networks (CNN).
ANNs are universal approximators, meaning that they are capable of modelling any form of relationship in the data, especially non-linear relationships. To learn complex not linear patter ANN uses specific activation functions such as sigmoid, hyperbolic tangent (tanh), rectifier linear unit (ReLU) (shown in Figure 4). These functions play a crucial role in enhancing the representation power and learning capabilities of ANNs. They determine the output of a neuron or a node receiving as arguments the weighted sum of previous neurons. The activation function is applied to this sum to introduce non-linear transformations to the network's intermediate or final outputs. The sigmoid function, characterized by a smooth and S-shaped curve, maps the network's input to a bounded output range, typically between 0 and 1. Its mathematical form is well-suited for introducing non-linearity in ANNs, enabling the modeling of complex relationships and capturing non-linear patterns in the data. The sigmoid activation function plays a crucial role in facilitating gradient propagation during backpropagation (i.e. weights update), contributing to the efficient learning process and convergence of ANNs.

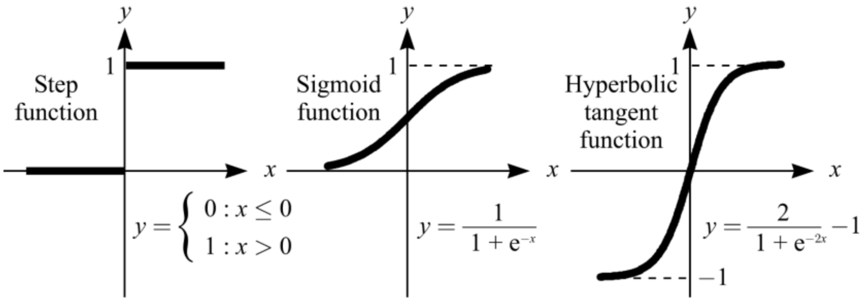

Step function
1
y
x
$y = \begin{cases} 0 : x \leq 0 \\ 1 : x > 0 \end{cases}$
Sigmoid function
1
y
x
$y = \frac{1}{1 + e^{-x}}$
Hyperbolic tangent function
1
y
x
$y = \frac{2}{1 + e^{-2x}} - 1$
−1

Figure 4: Typical activaction function used in ANN.

3.2.1 Deep Learning Fundamental Concepts

There are two fundamental systematic approaches for training an ANN:

1. **Supervised Learning**: In this training strategy for ANNs, the network learns from labeled data, where inputs are associated with corresponding output labels. The goal is to minimize the discrepancy between predicted outputs and true labels, achieved through iterative weight adjustments using optimization techniques like gradient descent. Supervised Learning requires using both the input and the target values for each sample in the training set. In this approach, two fundamental strategies exist to guide the learning process based on the desired output characteristics. These strategies, namely classification and regression, play crucial roles in various applications. If you want your model to recognize a discrete number of states you need a classifier, instead if you want to map your input to a continuous output you need a regression model.

    - *Classification*: Classification is the task of assigning a category, or class, to an item. In a supervised learning scenario, the training dataset is constituted by items labelled with the category they belong to. The training dataset is then used to build a model that will assign labels to new unlabelled items, or instances, that have not been seen before. In our study, the training set consists of muscle data belonging to a specific vocabulary of gestures drawn from the most common guitar techniques. Recognition of one of the vocabulary gestures sets a user-made sound preset. Classification of gestures based on how they unfold over time can be done by using various temporal modelling approaches, like recurrent neural networks, that we implemented both for classification and regression.

    - *Regression*: Regression is the task of estimating the relationship between an independent variable, called feature, and a dependent, output variable. This is done by building a statistical model that explains how the variables are related, and thus allows to infer the value of the dependent variable given the independent variable. The model describing this continuous function is built using a set of discrete samples of independent variables (the input) paired with the corresponding values of the dependent variables (the output). To implement a regression model one requires labelled data (input paired with corresponding output), hence is a supervised learning problem. In the context of musical interaction, regression is an attractive approach as it allows us to define a continuous sound effects modulation function based on the muscle contraction.

2. **Unsupervised Learning**: This training approach involves training ANNs on unlabeled data, where the network discovers hidden patterns, relationships, or structure within the data. It aims to learn without explicit target labels, using techniques like clustering, dimensionality reduction, and generative models. Unsupervised learning is used for tasks such as data exploration, anomaly detection, and feature learning

In the context of training a DL model, one epoch refers to a complete iteration through the entire training dataset during model training. The number of epochs determines how many times the training algorithm will process the entire dataset. Choosing the appropriate number of epochs is a critical decision in the training process, as it directly impacts the model's performance and convergence.
For the development of this project we use a supervised learning approach, proposing two ANNs that work in parallel: one gesture classifier and one regression network described in 4.7.1.

3.2.2 Recurrent Neural Network

Recurrent neural networks (RNN) have been created to address the problem of learning the contest of a sequence of data (e.g. sequence of frames of a video to interpret the present frame).

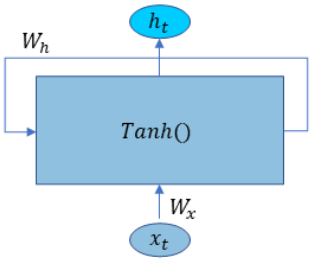


Figure 5: The standard recurrent cell of a recurrent neural network with tanh() as activation function.

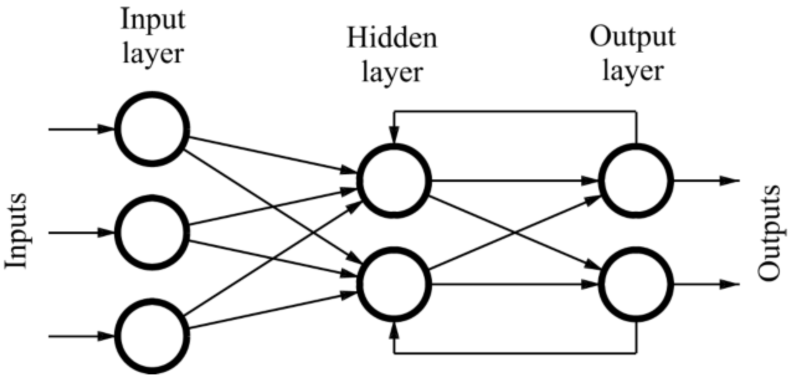


Figure 6: Sample of a recurrent neural network .

In RNNs, the recurrent layers or hidden layers consist of recurrent cells whose states are affected by both past states and current input with feedback connections. The recurrent layers can be organized in various architectures to form different RNNs. Therefore, RNNs are mainly distinguished by the type of recurrent cell and network architecture; different cells and inner connections enable RNNs to possess different characteristics. Usually RNNs are networks that consist of *standard recurrent cells* using tanh or *sigmoid* activation function. Figure 6 shows a schematic of the standard recurrent sigma cell. The mathematical expressions of the standard recurrent sigma cell are written as follows:

$$\boldsymbol{h_t} = \tanh\left(\boldsymbol{W_h h_{t-1}} + \boldsymbol{W_x} x_t + b\right) \tag{5}$$

the small, bold letters represent vectors while the capital, bold letters represent matrices; where x_t , h_t denote the input, the recurrent information of the cell at time t; W_h and W_x are the weights; and b is the bias. The model is recurrent because the state at time t depends on the state at time $t-1$ (which in turn depends on $t-1$, and so on) through the recurrent weights W_h. A recurrent cell therefore integrates information over all previous times to produce the output state vector y_t at time t.
Learning the weights W_h and W_x for a recurrent layer is computationally challenging, since the gradient calculation depends on the entire state sequence. The standard approach is **back-propagation through time** (BPTT). BPTT approximates the full gradient by unrolling state vectors up to a finite number k of time steps, and applying standard backpropagation to estimate gradients over length-k input sub-sequences. This standard RNN cell is known to suffer from vanishing gradient descent. In this, the gradients that are used to update the weights during backpropagation become very small. Multiplying weights with a gradient that is close to zero prevents the network from learning new weights. This stopping of learning results in the RNN forgetting what is seen in longer sequence, thus recurrent networks that consist of standard recurrent cells are not capable of handling long-term dependencies. Hochreiter (1991) and Bengio, Simard, and Frasconi (1994) analyzed fundamental reasons for the long-term dependencies problem: error signals flowing backward in time tend to either blow up or vanish. In order to deal with the problem of *long-term dependencies*, Hochreiter and Schmidhuber (1997) proposed the *Long Short Time Memory cell.*

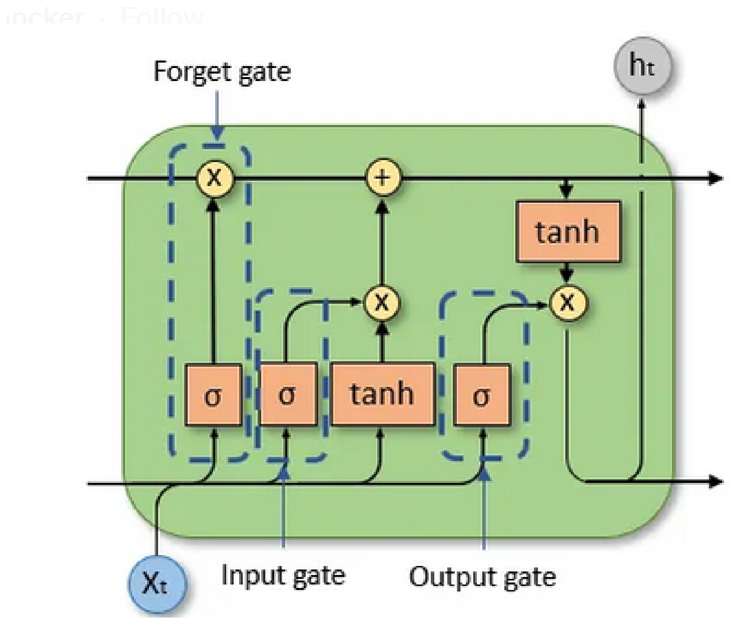


Figure 7: The LSTM cell architecture, ⊗ denotes the element-wise vector multiplication

They have improved the storage capacity of the standard recurrent cell tanh by introducing multiple "gates" composed of sigmoid cells (in the figure 7 represented as σ); thus the LSTM cell can be divided into 4 components: forgetting gate, input gate, output gate, and cell status. These gates function as filters and control the flow of information and determine what information is retained or ignored. All three gates use the sigmoid activation function. Only the weights and biases differ. The state of the cell is updated through the forgetting port and the input port.

1. The forget gate decides how much information from previous states to throw away, in other words, how much long memory shall be kept. For this, a sigmoid activation function is used which states the importance of the cell state. The output varies between 0 and 1 and states how much information is kept, i.e., 0, keep no information and 1, keep all information of the cell state.

$$\boldsymbol{forgetGate_t} = \sigma(\boldsymbol{W_{f,x}} x_t + \boldsymbol{W_{f,h} h_{t-1}} + \boldsymbol{b_f}) \tag{6}$$

   Based on h_{t-1} (previous hidden state) and x_t (current input at time-step t), this decides a value between 0 and 1 for each value in cell state c_{t-1}. For all 1's, all the information is kept as it is, for all 0's all the information is discarded and with other values it decides how much information from previous state is to be carried to the next state.

2. The input gate decides which information shall be added to the cell state and thus the long-term memory. Here, a sigmoid layer decides which values are updated.

$$\boldsymbol{inputGate_t} = \sigma(\boldsymbol{W_{i,x}} x_t + \boldsymbol{W_{i,h} h_{t-1}} + \boldsymbol{b_i}) \tag{7}$$

3. The output gate decides which parts of the cell state build the output. Hence, the output gate is responsible for the short-term memory.

$$\boldsymbol{outputGate_t} = \sigma(\boldsymbol{W_{o,x}} x_t + \boldsymbol{W_{o,h} h_{t-1}} + \boldsymbol{b_o}) \tag{8}$$

4. Cell state serves as the memory of an LSTM. The cell state is updated through the forget gate and the input gate.

$$\boldsymbol{cellState_t} = (\boldsymbol{f_t} \otimes \boldsymbol{c_{t-1}} + \boldsymbol{i_t} \otimes \tanh(\boldsymbol{h_{t-1}})) \tag{9}$$

$\otimes$ denotes the element wise vector multiplication. The first term in the above equation determines how much of the long-term memory is kept while the second terms adds new information to the cell state.

The hidden state of the current time step is then determined by the output gate and a *tanh* function which limits the cell state between -1 and 1.

$$\boldsymbol{h_t} = \boldsymbol{o_t} \odot \tanh(c_t) \tag{10}$$

RNN emerge as promising models for gesture recognition classifiers based on EMG signals due to their ability to identify patterns in dynamic time series; they have been widely used in different fields to forecast time series data and model energy system behavior. In the field of Human–Machine Interfaces and electromyography, multiple studies have explored the application of RNNs in gesture recognition systems in order to determine whether is or not suitable for gesture prediction and prosthesis control [54].

Along with the LSTM, one of the most used RNN architectures is the bidirectional recurrent neural network (BRNN) first described by Schuster and Paliwal [1997]. While traditional LSTMs effectively capture dependencies in one direction, they lack the ability to exploit information from future time steps, limiting their potential in certain applications. Bidirectional LSTM networks were introduced to address this limitation by incorporating two LSTM layers: one processing the input sequence in the forward direction and another in the backward direction. This means that for every point in a given sequence, the BRNN has complete, sequential information about all points before and after it. Also, because the net is free to use as much or as little of this context as necessary. This bidirectional nature empowers the network to capture dependencies from past and future contexts simultaneously, enabling more comprehensive modeling of sequential data. The first layer is responsible for processing the input sequence in the forward direction and the second in the backward direction. The outputs of both layers are concatenated, providing a holistic representation that captures information from both past and future contexts. This bidirectional flow enhances the network's ability to capture complex dependencies and improve the performance.

Due to the two-sided connectivity of basic units, the BRNN architecture is able to capture the relationship of inputs better than a one-sided RNN architecture. Indeed, sharing the features that are learned by each basic unit reduces the misclassification and improves overall accuracy. Also, considering the same inputs for all basic units enhances the classifier performance by providing prior information about the inputs. In [5] various architectures of RNN at different signal lengths are compared for hand gesture classification using an eight-channel sEMG dataset. BRNN with the same inputs and RNN with the sequential inputs show better performance than other architectures, those architectures achieve the accuracy of 0.93 in 500 msec of batch size.

4. Protocol Design

The protocol pipeline for acquiring, processing, and classifying sEMG signals in real-time to control the guitar sound, depicted in Figure 8, can be summarized as follows:

1. **Electrode Placement**: Electrodes are placed on the skin following the protocols of the Atlas of Muscle Innervation Zone [7]. These electrodes are connected to a wearable board, which captures analog sEMG signals and performs analog-to-digital conversion (ADC).
2. **Prefiltering Stage**: A prefiltering stage is applied to ensure signal fidelity. This stage consists of an anti-aliasing filter and a high-pass filter (with a cutoff frequency of 5Hz) applied in cascade.
3. **Data Acquisition and Processing**: The wearable board transmits the signals to a computer system running a proprietary data acquisition platform called Row Power. This platform performs crucial operations such as feature extraction and signal packaging. The packaged data is then simultaneously fed into two separate recurrent neural network (RNN) models.

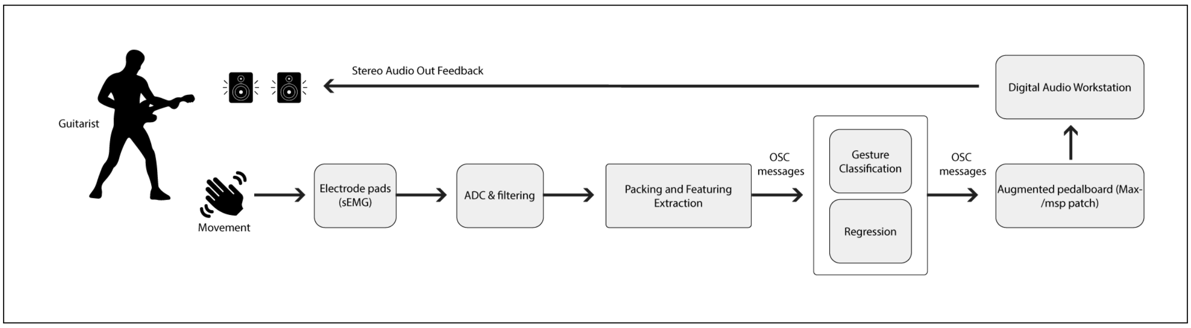

Guitarist
Stereo Audio Out Feedback
Digital Audio Workstation
Movement
Electrode pads (sEMG)
ADC & filtering
Packing and Featuring Extraction
OSC messages
Gesture Classification
Regression
OSC messages
Augmented pedalboard (Max-/msp patch)

Figure 8: This image shows the entire signal flow, from its generation and acquisition from the user's body to the modification of the sound by the Max/Msp patch.

4. **Gesture Classification**: The first RNN model is dedicated to gesture classification, facilitating the selection of pedalboard presets based on the recognized gestures.
5. **Effect Modulation**: The second RNN model operates as a regression model, enabling continuous modulation of the chosen effects based on the sEMG signals.
6. **Max/Msp 8 Patch**: The predicted parameters from the RNN models are sent to a Max/Msp 8 patch using the Open Sound Control (OSC) network protocol. The Max/Msp 8 patch encapsulates a series of five Virtual Sound Technology (VST) plugins arranged in a sequential manner to achieve the desired audio effects and modifications.
7. **Audio Output and Processing**: The audio output from the Max/Msp 8 patch is routed to a PA system. Ableton, a Digital Audio Workstation (DAW), is employed as an intermediary between the Max/Msp 8 patch and the PA system. Ableton provides a robust platform for audio mixing, signal processing, and playback control, ensuring high-quality sound reproduction.

This protocol pipeline enables the real-time control of guitar sound using sEMG signals, involving electrode placement, signal acquisition and processing, gesture classification, effect modulation, and audio output with the aid of Max/Msp 8 and Ableton.

4.1. Data Acquisition Platform

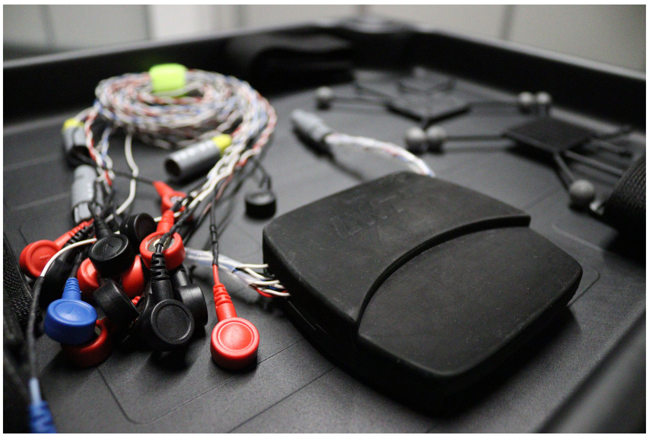


Figure 9: Data acquisition platform: 8-channels sEMG werable sensor board. Two sides, four channels for each side.

To introduce the sEMG signals to the artistic practice, in a reliable and robust way, in the laboratory we designed a non-invasive wearable sensor. We were able to investigate multiple muscle areas using this wearable computer system able to track the biometric signals of all the body. This device aims to be a bridge between the performer's internal world (observed through biopotential signals) and the world of the media arts (hardware and software used to create music and audio-visual contents, such as Max/Msp). The acquisition is made by our wearable computer device shown in figure 9 with a sampling rate of 1000Hz with 22bit resolution ADC for 8 channels in double differentials configuration. Then the data is sent to the software by wired USB 2.0

communication channel.

4.2. Electrode Pads Placement

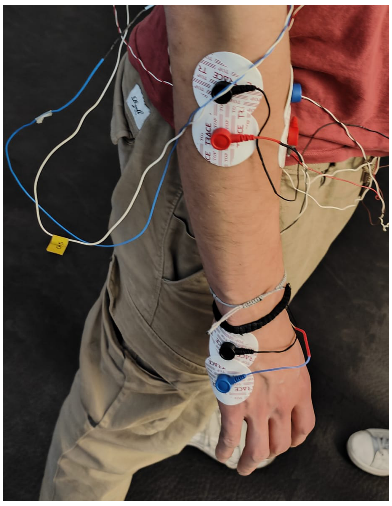


Figure 10: This photo shows an example of electrode placement on the guitarist's arm during muscle selection.

The electrode pads placement is critical for accurate sEMG measurements. The electrode pads should be placed over the belly of the muscle, with wires leading away from the muscle. It is essential to ensure that the skin is clean and dry before applying the pads, and avoid placing them over bony areas or areas with excessive hair. For the experiments, we use Ag/AgCl foam TOP TRACE ST 50 RLI electrodes (Ø 50 mm) (see Fig. 10). Next, it is important to connect the cables securely to the sEMG amplifier, ensuring that there is no interference from other sources. For the correct placement the protocols of Atlas of Muscle Innervation Zone has been followed [7]. Furthermore, we fixed the cables to the body with elastic straps to compensate for external electromagnetic disturbances due to the cables' movements during the recording.

4.3. Data Elaboration Platform

The data elaboration platform facilitates the acquisition of surface electromyography (sEMG) signals from the 8-channel sensor board described in 4.1. It is designed to capture and record synchronized sensor data, enabling real-time visualization of muscular activities through a dynamic plot. Furthermore, the platform incorporates a range of sophisticated post-processing computations, designed to assess and evaluate the performance of the acquired data. In the context of live performances, the observation and analysis of sEMG signals enable the tracking of the artist's expressive intentions manifested through varying degrees of muscular contractions. To enhance the live elaboration of sEMG acquisition during guitar performances, it becomes imperative to mitigate the impact of movement artifacts and electromagnetic interferences. To address this challenge, a pre-filtering stage is incorporated, aimed at attenuating high-frequency motion artifacts and ultra-low-frequency electromagnetic noise. This pre-filtering is achieved through the implementation of a fifth-order Butterworth bandpass filter, with cut-off frequencies set at [30, 300] Hz. In addition, to minimize the influence of electromagnetic disturbances stemming from nearby electronic devices, a harmonic band-stop Notch filter is applied. This filter is specifically designed to suppress the interference at a center frequency of 50 Hz, commonly associated with power line noise. The quality factor of the Notch filter is set at 0.8, ensuring effective attenuation of the targeted frequency while preserving the integrity of the sEMG signal. By implementing the pre-filtering stage as described, a more refined analysis, interpretation, and subsequent processing of the acquired sEMG data is enabled. This pre-filtering approach enhances the accuracy and reliability of investigations into the underlying muscle activity.

4.4. Feature Selection

In order to enhance the discrimination capabilities among various guitar techniques based on the acquired sEMG signals, a feature selection stage is executed. The primary objective of this feature selection process is to identify the most informative and discriminative features that effectively differentiate between different guitar gestures.

By adhering to specific selection criteria, we aim to achieve a high level of inference and accuracy in distinguishing the selected gestures. In this study, we focus on identifying the two most significant low-level features in the time domain, which are crucial for characterizing the unique aspects of the guitar-playing techniques. We compute the feature over a sliding window of 500ms with no overlaps, this ensures the best trade-off between classification performance and low latency for real-time interaction. The model shows the best performance while using the combination of two features in the time domain:

1. **Root Mean Square** (RMS) (Equation 1) to track the amount of power activation.The sEMG amplitude is considered the most meaningful feature because it is directly proportional to the exerted force, for this reason, the RMS is one of the most popular features used in EMG signal analysis [40]. It has been shown that sEMG amplitude analysis is more useful than frequency domain analysis, for gesture recognition tasks based on supervised learning algorithms [2].

2. **Zero Crossing Rate** (ZCR) (Equation 4) a valuable and computationally efficient feature that encapsulates the approximate frequency behavior of the signal. ZCR is a widely used feature in audio signal processing, it is used to recognize harmonic sounds from noisy ones, and it is employed in speech recognition [6]; applied to the zero-mean EMG signal it's an indicator of the neurons firing rate, directly proportional to the level of activation of muscular fiber. By quantifying the rate at which the signal crosses the zero amplitude threshold, the ZCR provides a succinct representation of the frequency dynamics present within the analyzed acquisition; behaving as an accurate descriptor of the amount of activation over time during a specific gesture.

Based on the combination of ZCR and RMS, we present in the next section an evaluation method to classify the best muscles having a set of specific gestures to classify(i.e the one of a guitarist).

4.5. Muscle Selection

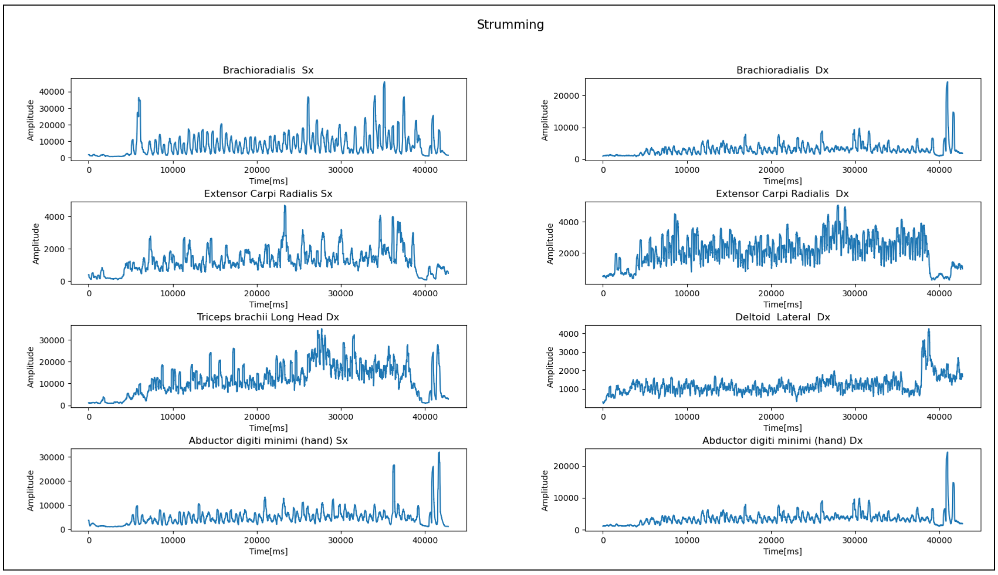


Figure 11: Example of plots of eight different muscles during the second stage of the selection using the RMS feature. it is noticeable how the discarded hand muscles show lower level of activation

The selection of appropriate muscles is paramount in ensuring accurate gesture classification, as distinct guitar techniques elicit unique patterns of muscle activation. To establish a robust muscle selection criterion, we conducted multiple 30 seconds for each gesture. After that two consecutive stages were performed:

1. In the first stage, we analyzed the average activation over time of each muscle by plotting the ZCR for each of the seven gestures we aimed to classify. This allowed us to assess the extent to which each muscle was activated on average, during the set of gestures. By comparing the average ZCR values with the specified threshold, we identified the top eight muscles for the second step.

2. In the second stage, we divided the twelve muscles into groups of eight and evaluated the average RMS

values for each group on the seven gestures. We plotted the RMS values for the eight selected muscles to examine the average amplitude levels, as shown in Figure 16; this allows us to determine the amplitude threshold. For instance, we excluded the hand muscles from selection due to their lower RMS values, which contributed to a lower average RMS for the group compared to other sets that did not include hand muscles.

By computing the average ZCR for each muscle, we obtained an indication of the level of muscle activation over time during the execution of specific gestures. Through extensive testing, we observed that in a resting position (with our configuration test), the ZCR values ranged from 40 to 70, while the RMS around 2.3 mV compared to the baseline. These observed ranges serve to define a heuristic **threshold for muscle classification**: 50 slop change and 4.6 mV, for ZCR and RMS respectively, into a 500 ms window.
Consequently, we rank the muscles based on their mean ZCR and RMS values, discarding those muscles that fall below the established threshold, thereby ensuring the selection of only the most relevant and activated muscles for accurate gesture discrimination. We present the result and the list of selected muscles in section 5.1.

4.6. Dataset Creation

The proposed tool is intra-subject, meaning that it is built upon the stylistic decision of the user, that train it. For this reason, we define a procedural approach to train the system. The dataset was meticulously constructed following a strict procedure:

1. **S**even guitar techniques were selected namely: *fingerpicking, strumming, bending, down picking, alternate picking, tapping, and pull-off/hammer-on.*
2. Each technique was paired with a corresponding guitar riff [3] that best represented its characteristic motion, to ensure consistent movements across different participants,
3. Two acquisitions were performed for each technique, asking the tester to play at different levels of muscle contraction, in order to detect the maximum and minimum contraction to train the regression mode, which aims to differentiate the varying levels of force exerted during the gestures.
4. Each acquisition was performed with the same electric guitar in a standing position, with a fixed duration (30 s) and a constant tempo of 100 beats per minute (BPM).
5. Finally, a built-in function within Row Power was utilized to export a filtered RMS version of each acquisition, saving the maximum and the minimum values for each muscles to apply a normalization between $[0, 1]$.

The dataset was collected from four guitarists (both male and female) with varying levels of expertise, spanning from beginner to professional, and with diverse anthropometric measures such as heights, weights, and muscle shapes, all carefully documented.
To the best of our knowledge, there are no public sEMG datasets of guitar techniques. We built four datasets with fours subject, that are publicly available at the GitHub repository of this research for further studies[4].
All acquisitions capture the signals from the eight selected muscles as presented in Section 5.1; additionally, audio files of each acquisition were saved for potential future application.
Each acquisition was meticulously labeled and organized into dedicated folders using Row Power software. The RMS-processed acquisitions were then loaded into a pandas data frame, enabling the proper shaping of the dataset. By concatenating a target dataset alongside each acquisition, the supervised learning procedure was facilitated. This target dataset included references to the corresponding class assigned to each gesture acquisition. This tagging process, wherein each example is labeled with its associated class, adheres to the standardized procedure employed prior to training the model, in a supervised fashion.
Each dataset is normalized between $[0, 1]$ with a Mix-Max strategy, to enable the model to better extract inferences by promoting fair treatment of features, facilitating faster convergence, and enhancing the network's ability to generalize its learned knowledge. Normalization ensures that all features are treated equally, allowing the ANN to learn from each feature's meaningful patterns without bias. The dataset was split into two parts: 80% was used for training, and 20% was used as a validation set to allow the process of hyper-parameter tuning (i.e. updating of the weights) and validation of the training step.

[3] A riff is a repeated pattern, or melody, often played by the rhythm section instruments or solo instrument, that forms the basis or accompaniment of a musical composition.

[4] GitHub repository of the research

4.7. Proposed Model

For the purpose of arms gesture classification and real-time regression, this section presents the steps followed to develop a novel model, showing key concepts that allow the implementation of a customized architecture based on a BRNN. Specifically, the model incorporated a Bidirectional Long Short-Term Memory (BLSTM) layer, coupled with two consecutive fully connected Dense layers. The schematic representation of the proposed model architecture can be observed in Figure 13.

The utilization of an RNN structure with LSTM configuration has been widely acknowledged as an effective approach for capturing temporal patterns in time series data. Numerous researchers have highlighted the RNN's capacity to retain information, rendering it highly suitable for applications involving time series analysis. In this study, we proposed a hybrid RNN architecture, comprising a recurrent layer followed by two dense layers, to enhance overall performance. Notably, it has been demonstrated that incorporating dense layers in conjunction with the recurrent layer can yield superior outcomes compared to models solely relying on recurrent layers [26]. To further enhance the model architecture, we explored the implementation of a bidirectional LSTM, allowing the recurrent layer to operate in both the past and future directions. By incorporating bidirectionality, we aimed to capture more comprehensive temporal dependencies and improve the model's predictive capabilities. The network was developed using Python programming language, utilizing the latest version of TensorFlow (2.12.0). The Keras API was employed to implement the layers, while the Scikit-learn library facilitated the integration of useful preprocessing functions. The source code and scripts are openly accessible on Google Colab [4].

4.7.1 Model Design

The first crucial aspect that requires careful consideration when employing a Long Short-Term Memory (LSTM) network is the selection of the input data shape. Specifically, the input data shape in a recurrent neural network (RNN) dictates the number of samples to be analyzed collectively to identify recurrent patterns within the input stream. In essence, it determines the memory capacity of the system, enabling it to capture and retain temporal dependencies effectively. Every LSTM layer accept as input a 3-Dimensional tensor composed of (number of batches, number of time steps, number of features) as shown in Figure 12. It is mandatory to set the shapes for the last two dimensions, while the first (i.e the number of batches) could vary. In our specific case, the number of features remains fixed at eight, representing the RMS values of each sEMG channel. The size of the time step corresponds to the number of features (i.e., the values of the 8 channels) that the network will analyze at one time; the bigger the time step the longer the temporal context tracked by the network. The third dimension can vary and represents the number of time steps provided as input to the network during training. We fixed the length of the time step to 256, taking into consideration that the sample time is 1 ms, 256 sample windows allow an actuation speed of around 200–300 ms considering a serial communication with the computational model.
Based on the input 3D shape, a reshaping process is required, in which the training set is divided into groups of sizes equal to the number of time steps. Applying a supervised learning techniques we need to create for each class a correspondent target set of two dimension: the first equal to the number of baths obtained during the reshaping of the training set and the second equal to the number of output classes.

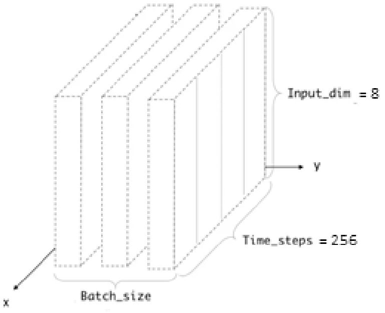


Figure 12: The chosen input shape of the model, the batch size is a varying parameter.

One of the key point to reach good performances is the architecture of the model: In this stage we avaluate numerous architecture firstly stacking multiple LSTM, then reinforcing the generalization and features extraction

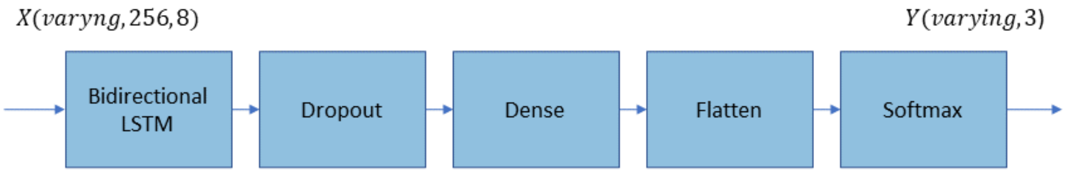


Figure 13: The model architecture diagram

of the RNN stacking at the end multiple Dense layer. The main difficulty was finding an architecture with not too many parameters, suitable for real-time usage. Stacking multiple LSTM with a high number of units (i. e. neurons), resulting in architecture with number of parameters of the order of millions; We discover that a single BLST with fewer units performs much better than multiple stacked LSTM layers. Only one BLSTM was not able to reach high performance, for this reason, we add multiple Dense layers after it to reinforce the inference of the architecture. After various experiments, we reach the best performance with the architecture described in Figure 13. The system consists of one BLSTM, two Dense layers with activation function *tanh and ReLu* respectively, a Dropout regularization layer, a Flatten layer to reduce the dimensionality and a final SOFTMAX layer to compute the class probabilities.

Dropout is a commonly used technique to avoid overfitting by improving the generalization ability of the network. During the training phase, a dropout layer randomly "drops out" (i.e., temporarily removes) a certain proportion of the units (i.e. neurons) in the preceding layer as shown in Figure 14. The dropped-out units are temporarily ignored during both forward and backward propagation, effectively making the network less reliant on specific units. By doing so, dropout prevents the network from relying too heavily on any individual neuron, which encourages the network to learn more robust and generalizable features. The output of the last layer is

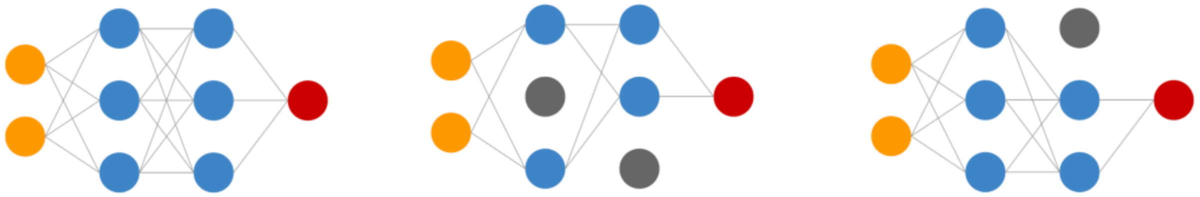


Figure 14: A representation of how a Dropout layer randomly “drops” neurons during training with the aim of forcing the network to learn more robust representations.

a flattened vector, to reduce the shape of the data from 3D (RNN work with 3D tensor) to a 2D before feeding the vector to the output layer. The output layer integrates a SOFTMAX activation function to classify which gesture is being performed. The SOFTMAX activation function is one of the most widely used functions for multi-class classification problems because of its performance in predicting the probability of each class [23]; It predicts the probability of a class C_j given an input X as shown in Equation 11.

$$p_j = P(C_j|X) = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}} \tag{11}$$

where:

- p_j is the output probability of the class j
- C is the total number of classes that are being predicted
- z_j represents the linear combination of the weights and the previous layer activations in matrix form, and can be defined as $z_j = \mathbf{W}z^{l-1}$
  It is important to note that the SOFTMAX output layer has the same units as the number of classes that are being predicted.

4.7.2 Model Optimization

Once the model is defined, the next step was to select a cost function in order to optimize it through training and obtain the desired results in the classifier. In general, when using an output layer with a SOFTMAX activation function,a categorical cross entropy loss function 12 is selected, which allows optimizing the cost function 13 that sums the loss over the training examples. The categorical cross entropy loss allows a correct penalization of incorrect predictions and improve the performance in multi-class classification problems

$$\text{Categorical Cross Entropy Loss} = \theta = -\sum_{i=1}^{C} y_i \log(p_i) \tag{12}$$

$$\text{Cost Function} \quad J(\theta) = \frac{1}{m}\sum_{i=1}^{m}\left(-\sum_{j=1}^{C} y_{i,j} \log(p_{i,j})\right) \tag{13}$$

where:

- C represents the number of classes in the classification task.
- y_i denotes the true probability (ground truth) of class i.
- p_i represents the predicted probability of class i assigned by the model.

The batch size was set during the hyper-parameter tuning along with the number of training epochs (Training iterations). The optimization algorithm will be explained in detail in the next paragraph. In order to minimize the cost function, it is necessary to update the weights W and biases b of each layer. This must be done by an optimization algorithm such as gradient descent. In this research, the Adaptive Moment Estimation **ADAM** algorithm was selected; that is, a combination of *Gradient Descent* With *Momentum and Root MeanSquare Propagation.* This optimization algorithm is considered to be one of the best optimization algorithms to train neural networks [28]. The learning rate represents the step size or magnitude at which the weights of the neural network are updated during the training process. During each iteration of the training process, the ADAM algorithm computes the gradients of the weights with respect to the loss function and then scales these gradients by the learning rate before updating the weights. The update rule for the weights using ADAM optimization with a learning rate α can be represented as follows:

$$w_{\text{new}} = w_{\text{old}} - \alpha \times \text{gradient} \tag{14}$$

Here, w_{new} represents the updated weight value, w_{old} represents the current weight value, and gradient represents the computed gradient of the weights with respect to the loss function. The learning rate α controls the magnitude of the weight updates. In summary, the learning rate α determines the step size at which the weights are updated during training, influencing the convergence speed and stability of the learning process. Selecting an appropriate learning rate is essential for achieving optimal performance. If the learning rate is set too high and the training becomes unstable, reducing the learning rate can help stabilize the process. Conversely, if the learning rate is set too low and the training becomes slow, increasing the learning rate can accelerate convergence.

In order to prevent overfitting we add a regularization term called L1 kernel regularization. it is a technique that adds a regularization term into the loss function to impose a penalty on the absolute values of the weights (kernels) in the network. This regularization technique encourages the model to have sparse weight values, meaning it tends to set many weights to exactly zero. The L1 kernel regularization term is defined as the sum of the absolute values of all the weights in the network, multiplied by a regularization parameter λ :

$$\text{Loss with L1 regularization} = -\sum_{i=1}^{C} y_i \log(p_i) + \lambda \sum_{i} |w_i| \tag{15}$$

Here, w_i represents an individual weight in the network, and the summation is taken over all the weights in the network. The regularization parameter λ controls the strength of the regularization effect, with larger values promoting sparser weights. The L1 regularization term encourages the network to learn sparse weight values because it introduces a penalty for non-zero weights. When the loss is minimized, the network will tend to assign smaller absolute values to less important weights, effectively driving some of them to zero. By promoting sparsity in the weights, L1 regularization can simplify the model and improve its ability to generalize to unseen data.

4.7.3 Hyperparameters tuning

After the architecture is decided, the next key step for the model optimization, is the process known as hyper-parameter tuning. The hyper-parameter are values used to control the learning of the model, for example, the number of neurons of each layer, the number of training epochs (Training iterations), the learning rate η... This process is performed with the objective of obtaining an optimal performance of the model, and correcting issues like high bias and high variance. In this research, manual hyper-parameter tuning was performed. This optimization was done by tuning the following hyperparameters: the learning rate η, the batch size, the number of training epochs, the model architecture, the number of neurons of each layer, and its activation function. During the training, we investigated the proper number of time steps after which the weights are updated, called the *batch size*. This hyperparameter can change performance dramatically and must be chosen carefully, along with the number of *epochs*, over which the process of updating the weights will be iterated.
To determine the optimal number of epochs we applied a well know techquince called *Early stopping*. It involves monitoring the model's performance on a validation dataset (i. e. data that the model does not see during the training) and stopping the training when the validation loss function does not decrease for a specified number of epochs. Early stopping helps identify the point at which the model achieves the best trade-off between capturing the underlying patterns and generalizing to unseen data. The result with the chosen hyperparameters can be seen in Figure 3

4.7.4 Evaluation Metrics

Once the model is trained, it is important to define a metric to determine its performance. There are many metrics to track the performance of a ANN, most of them based on the confusion matrix, which contains the values of true positives (TP), true negatives (TN), false positives (FP) and false negatives (FN), values that come in handy to calculate these metrics. This research evaluated the performance of the gesture classifier using the accuracy, commonly used in the literature [24], which can be mathematically defined as shown in Equation 16

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \tag{16}$$

As Equation 16 shows, the accuracy measures how much the model is correctly predicting on the entire data-set and represents the probability that a prediction made by the model is correct. The accuracy does not take into consideration the dataset class distribution, which could be misleading if the model is trained with imbalanced data sets in which some classes have fewer examples than other classes. For this reason during the training set creation, we ensured to have the same number of examples for each class.
In order to analyze and evaluate the proposed model an objective and a subjective evaluation were performed. Initially, the model was tested over a test set; then it was tested in a real-time gesture classification scenario. The accuracy of the model computed over the test set can be seen in Figure 17 where we show the result using a confusion matrix [5]; while subjective evaluation is described in 5.5.

5. Experimental Setup and Evaluation

This section presents the experimental setup and evaluation strategies followed for the creation of the described intelligent DMI. The development of this project was guided by different experiments that took place in the LWT3 laboratory; All of them were conducted using the same right-hand solid-body electric guitar, performed in a stationary standing position. The resulting architecture, which shows the signal flows of the proposed system, is presented in Figure 8. The experiments first involved a single guitarist, then multiple ones and were conducted in three phases, all consisting of multiple muscle acquisitions:

Phase 1: Muscle Selection

- Objective: Define muscle selection criteria and evaluate the best upper limb muscles for guitar gesture classification.
- Result: Identified the 8 arm muscles that maximize classification results.

Phase 2: Mapping strategy

- Objective: Define the gestures and the VSTs for mapping muscles to sound.

[5] A confusion matrix is a square matrix used to evaluate the performance of a multi-class model. It summarizes the predicted and actual labels of the data in a tabular form, showing the classifier's performance and identifying areas for improvement.

- Result: Development of the RNNs' architecture, the training process, and the Max/msp patch.

Phase 3: Evaluation

- Objective: Evaluate the system's performance and effectiveness with three additional guitarists.
- Result: Validation questionnaire.

The initial two phases were guided by experiments conducted with a single guitarist involved from the very beginning of the development process. Multiple sessions of sEMG acquisition were performed, which guided the mapping of sound to intentions, the design of the digital pedal-board using Max/Msp, and the determination of the gesture vocabulary. The user is required to provide their own gesture vocabulary and VST pedals to construct the augmented pedalboard. In this experiment, the selected gesture vocabulary consists of seven guitar techniques, chosen on the basis of their popularity, low gesture similarity, and inclusion of only left limb movements (e.g., pull-off and hammer-on). Due to variations in individual anatomies and muscle usage, even seemingly identical gestures can differ in muscle activation. The careful selection of features and muscles, as explained in the previous sections, required fine-tuning through multiple acquisitions.
A step-by-step evaluation experiment was established to apply the muscle selection criteria defined in Section 4.5:

1. Seven guitar gestures were performed at fixed BPM (Beats per minute, i.e 120).
2. All the acquisition has the same lengths of 30 seconds.
3. Multiple groups of eight muscles were collected to identify the best set.

Performance data for the seven technical gestures was collected for each tested muscle. Muscles with consistently low RMS values (left anterior deltoid, left and right triceps, and hand muscles) were discarded in this phase. To further refine the muscle selection and identify the muscles specifically activated during each guitar gesture, the Zero-Crossing Rate (ZCR) was utilized to evaluate mean activation over a time interval. The combination of these two features, as extensively described in Section 4.4, facilitated the identification of the optimal set of eight arm muscles for classifying each specific gesture, as depicted in Table 1.

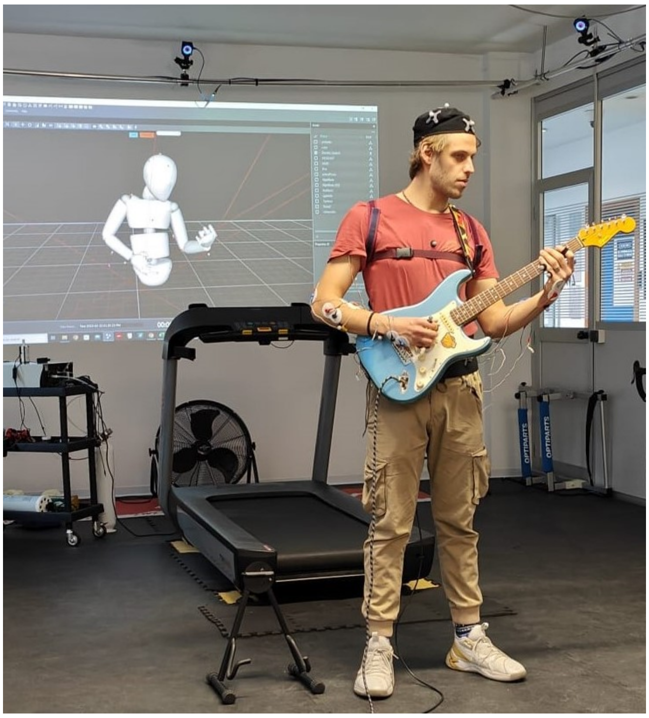


Figure 15: Guitarist who follow the first two experimental stages, during data acquisition in the lab, wearing the proprietary data acquisition system.

5.1. Selected Muscles

We propose a method for selecting muscles (listed in Table 1) relevant to guitar performance through a systematic evaluation experiment. We analyzed twelve upper limbs muscles selecting the best eight following the criteria explained in 4.5. The guitarist is instructed to perform a set of standard technical gestures belonging to

Muscle selection results

Gestures Muscles	Tapping		Finger picking		Bending		Strumming		Pull off-hammer on		Down picking		Alternate picking	
	ZCR	RMS	ZCR	RMS	ZCR	RMS	ZCR	RMS	ZCR	RMS	ZCR	RMS	ZCR	RMS
Left Flexor Carpi Radialis	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
Left Extensor Carpi Radialis		✓	✓	✓	✓			✓	✓	✓	✓	✓	✓	✓
Left Bicep Brachii Short Head					✓	✓					✓	✓		
Left Brachioradialis	✓	✓	✓	✓			✓	✓			✓	✓	✓	✓
Right Flexor Carpi Radialis	✓	✓	✓	✓			✓	✓			✓	✓	✓	✓
Right Extensor Carpi Radialis	✓	✓	✓	✓			✓	✓			✓	✓	✓	✓
Right Bicep Brachii Short Head					✓	✓					✓	✓		
Right Anterior Deltoid							✓	✓			✓	✓		
Left Anterior Deltoid														
Right Triceps														
Left Triceps														
Right hand														
Left hand														
Right Brachioradialis														

Table 1: The best muscle for classifying a specific guitar gesture based on the average of ZCR and RMS for a standard right-hand guitarist

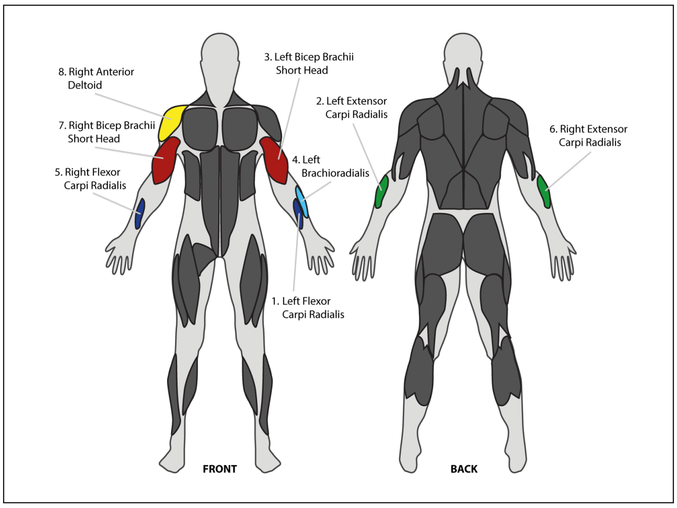

8. Right Anterior Deltoid
3. Left Bicep Brachii Short Head
2. Left Extensor Carpi Radialis
6. Right Extensor Carpi Radialis
7. Right Bicep Brachii Short Head
4. Left Brachioradialis
5. Right Flexor Carpi Radialis
1. Left Flexor Carpi Radialis
FRONT
BACK

Figure 16: Selection of the best eight muscles for guitar gesture classification. This image shows the electrode adhesive placement area, highlighting the chosen muscles.

the most common guitar techniques: *finger picking, strumming, bending/vibrato, pull off - hammer on, tapping, down picking* and *alternative picking*. He performed them at a fixed tempo of 120 Beats per Minute (BPM) for a duration of 30 seconds each. During this phase, we exclude muscles with consistently low RMS values, namely the left anterior deltoid, left and right triceps, and hand muscles. In order to further refine the muscle selection process and identify the specific muscle activation patterns associated with each guitar gesture, we utilize the Zero-Crossing Rate (ZCR) to evaluate the average activation level within predefined time intervals. By combining these two features, as elaborated in Section 4.4, we are able to determine the optimal set of eight upper limb muscles for accurately classifying each specific gesture, as illustrated in Table 1. while performing The twelve analyzed muscles are listed in Table 1, and the discarded ones are the last four. For each of the eight selected ones, you could see in which gesture they perform better. As a result, we present here the most activated muscle during guitar playing.

1. **Forearm muscles**: including the *flexor carpi radialis the extensor carpi radialis* and *brachioradialis*, they are responsible for the wrist movement during guitar playing, which is particularly activated during techniques such as strumming, arpeggio, tremolo picking, double pick and strong pick.
2. **Right anterior Deltoid muscle**: This muscle is involved in shoulder movement, which is important for detecting techniques such as strong picking, strumming, and chord changes (i. e left-hand movement when changing chords). The Deltoid muscles are responsible for abduction and flexion of the right arm at the shoulder, it is crucial to detect techniques such as strumming and chord changes. We notice that the anterior right deltoid muscle has slightly higher values, than the lateral, as it tends to be more consistently active during guitar playing. While the left deltoid is completely discarded for low activation proved by low values of both ZC and RMS.
3. **Biceps muscles**: These muscles are involved in elbow movement, which is high activation during techniques such as bending and vibrato, and they are responsible for elbow flexion strongly activated during string bending and strong picking.

We discarded the triceps muscle and the right brachioradialis, which fall behind the thresholds defined in 4.5. The hand muscles show too little SNR due to cross-talk. The cross-talk of the hand muscle is more significant than the other part of the body because of higher tissue density concentrated in a smaller area which induces weaker signals due to the muscle's reduced size. The final muscle selection is presented in figure 16.

5.2. Training Process

We split the training into two parts once for the classification and the other for the regression. In the first part, we train the Gesture classification: the guitarist defines four sonic presets in the Max/Msp path, tagging each one to the intention he wants to convey. Based on those we create three classes in the model named: calm, happy, and frenetic; then we ask the musician to associate each class to the corresponding guitar gestures (among the seven guitar techniques defined in section 4.5) that he wants to link to that specific intention. He associates the calm class with fingerpicking and rest position. The happy class with strumming, down picking and bending. The frenetic with tapping, pull off/hammer on, and alternate picking. The classifier model changes the Max/Msp preset by sending an OSC message as soon as a specific gesture is classified. In the second part, we train the regression model; it modulates a group of guitar effects based on the amount of muscular contraction. The musician chooses to modulate the Mix potentiometer (i.e the parameter that controls the blending of the effects) of the chorus, reverb and delay, and the drive potentiometer of the overdrive and distortion pedals. The regression model has one output for each parameter, changing them via OSC messages during the performance. We train this model by collecting two acquisitions for each gesture with opposite contraction levels: a harder and a relaxed one. Eventually, the musician associates each acquisition with the desired values to create the target dataset.

5.3. Model Evaluation

Now we present the final architecture of the classifier model, specifying the number of units for each hidden layer, the activation functions, and the input/output shape. The architecture of the regression model is identical besides the output shape of (varying, 6) having 6 pedalboard's parameters to control, and the output activation function equal to ReLu.

Layer	Output Shape	Number of parameters	Units
Input	(varying, 256, 8)	-	-
BLSTM(kernel regularizer=l1(0.001))	(varying, 256, 32)	3200	32
Dropout(0.1)	(varying, 256, 32)	0	0
Dense (activation= 'tanh')	(varying, 256, 64)	2112	64
Dense(activation= 'relu')	(varying, 256, 32)	2080	32
Flatten()	(varying, 8192)	0	0
Dense(activation= 'softmax')	(varying, 3)	24579	3
Model Summary			
Total Parameters			31,971
Trainable Parameters			31,971
Output Shape			(varying, 3)

Table 2: Architecture of the classifier model with 3 classes. Units indicate the number of neurons for each layer

5.3.1 Hyperparameter Tuning Results

In this section, we present the results of our hyperparameter tuning for the classification and regression of sEMG signals, taking into consideration that we aim to define a lightweight model suitable for real-time application and its embedding into wearable devices. We considered various hyperparameters and performed an extensive manual search to find the optimal values. After evaluating several combinations, we selected the values of the hyperparameters presented in Table 3 succeeding in defining two efficient models, one for regression and one for classification, weighing 735 Kb and 450 Kb, respectively.

Hyperparameter	Value
Input Time Steps	256
Regularization Strategy	L1(factor= 0.001)
Dropout Rate	0.1
Optimization Algorithm	Adam
Learning Rate	0.0001
Loss Function	Categorical Cross Entropy
Batch Size	32
Early Stopping	Monitor 'loss function' with patience 10
Early Stopping	Monitor 'validation loss' with patience 10

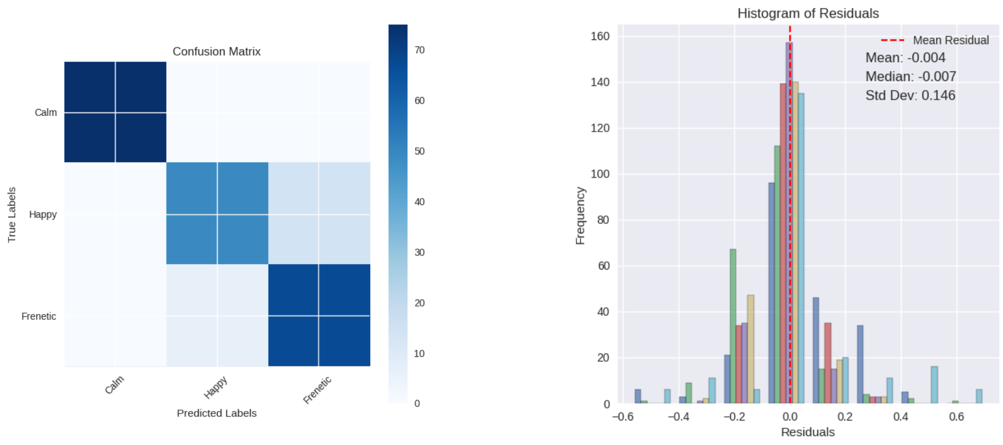
Table 3: Model's Hyperparameters Values

The values of the hyperparameters were chosen based on their performance in terms of accuracy and convergence during the training process. We tried to strike a balance between model complexity and generalization capability, achieving good performance with only 31,971 parameters (we started from an initial architecture with 2 million parameters), resulting in a compact model suitable for real-time applications. Using these hyperparameters, we trained the classifier BLSTM model on our sEMG dataset and evaluated its performance on a separate test set.

The results of the training evaluation are presented in Figures 17, 18, both for the classifier and the regression model.

5.3.2 Training evaluation

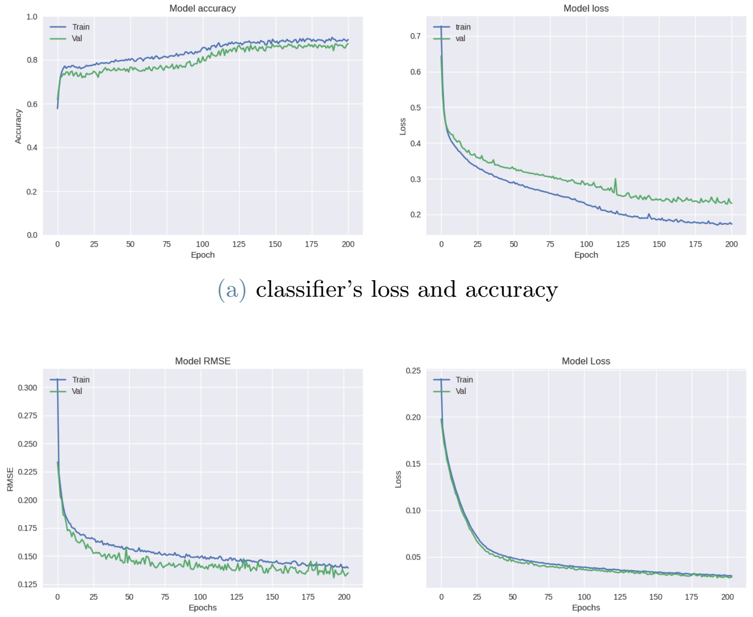
In this section, we present training evaluation plots for the two proposed models; during training, we monitored the loss function and accuracy to ensure convergence (Fig. 18). Finally, we evaluate the model on the test set by plotting the confusion matrix and regression residuals (Fig 17).



(a) classifier's confusion matrix

(b) regression residuals

Figure 17: (a) Confusion matrix of the classifier with three classes over the test dataset, with an overall accuracy of 0.905. (b) Histogram of the regression's residuals, the bins' Gaussian distribution shows the effectiveness of the model.



(a) classifier's loss and accuracy

(b) regression's loss and RMSE

Figure 18: (a) Evaluation plot of the classifier's training, with a loss: 0.1801 - accuracy: 0.9245.(b) Evaluation plot of the regression RNN, whit an MSE loss: 0.0316 and a RMSE: 0.1464 over the test set.

5.4. Sound Processing Unit

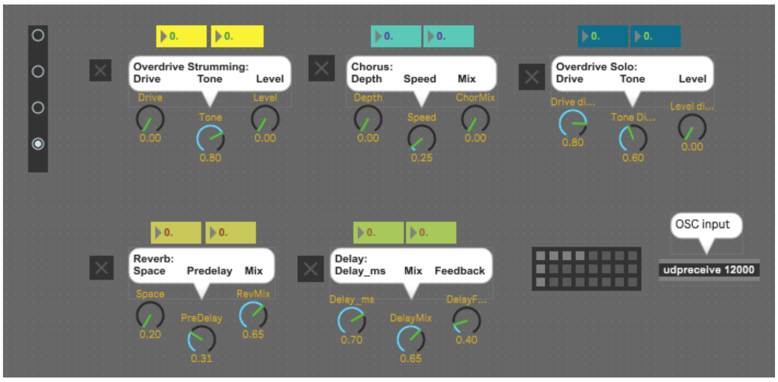


Figure 19: Max/Msp patch for sound parameters control

For the sound implementation of the augmented pedal board, We asked the guitarist to provide us with the VSTs he usually uses during his performance. He chose five different effects:

- **Overdrive**: *Overdrive TSC 1.1* develop by *Mercuriall* with three knobs to control: drive, tone, level.
- **Distorsion**: *Distortion Greed smasher* develop by *Mercuriall* with three knobs to control: drive, tone, level.
- **Chorus**: *Chorus WS-1* develop by *Mercuriall* with three knobs to control: depth, speed, mix.
- **Reverb**: *Pro-R* develop by Fab Filter with three knobs to control: space, pre-delay (ms), mix.
- **Delay**: *Valhalla Super Massive Delay* with three knobs to control: delay time (ms), mix, feedback.

To design an effective and integrated user experience, we use *Cycling '74 Max/Msp 8*, a visual programming language for music and multimedia, widely used by musicians. It controls the five imported VST with the `vst object`. The data communication between Row Pawer and the Max/Msp patch was implemented using the OSC network communication protocol [61]. We add three knobs for each VST, allowing the modulation of the effect's main parameters by the recursive model. These knobs are directly controlled by OSC messages allowing the regressive network to communicate with the augmented pedalboard. The Max/Msp patch receives the output of the two RNN models through two different OSC messages: one for setting the sonic preset and one containing a list of float in a [0,1] range for the real-time parameters mapping. The artist can save and recall the sonic preset using the `preset object` and can activate/deactivate each vst with the `tugle button` .

5.5. Evaluation Questionnaire

For the final evaluation, we had an experienced guitarist try the system. We evaluated the system in a performance scenario, after the training process (described in 5.2) to adapt the two RNNs according to the subject. To collect the musician's perspective, we design a questionnaire following the typical DMI evaluation methodologies ([8] [43]), focusing our attention on the ability of the system to understand the musical intentions.

1. **Playability:** How much control do you feel that you have over the tool while using it?
   Grade: 7/10
   Comment: The system well recognizes changes in the intensity of movements and modifies the sound accordingly.

2. **Learnability:** How easy was learning to use it?
   Grade: 9/10
   Comment: Just as with all guitar instruments and effects it takes a moment to adjust, which in this case is very brief

3. **Expressiveness:** How much does it help you to enhance your creativity and express your musical intention?
   Grade: 10/10
   Comment: The potentials are endless

4. **Enjoyability:** How enjoyable was your experience while using it?
   Grade: 10/10
   Comment: There were times where the system did exactly what I wanted it to do.

5. **Novelty:** How much novelty does it introduce to your performance
   Grade: 10/10
   Comment: Changing sound without interaction with a pedalboard makes the performance more organic

6. **Effectiveness:** How much is it able to track your musical intention?
   Grade: 8/10
   Comment: Very good premise, but it will still improve with time

7. **Sound quality:** Rate the perceived sound quality obtained through the digital pedalboard setup.
   Grade: 8/10
   Comment: Excellent

8. How much does it change the way you interact with your pedalboard?
   Grade: 10/10
   Comment: Completely .

9. How much were we able to improve the system following your feedback after the first try?
   Grade: 8/10 **Comment:** It would also be interesting to detect wrist movements with their changes in intensity

10. Can you describe your experience of being involved in the development of a smart instrument (expectations, results, impressions...)?
    Answer: A very useful experience to understand what the actual potential of these tools is, the results are very compelling and I look forward to future developments.

11. How confident would you feel using it in a live performance?
    Grade: 7/10
    Comment: Very confident in case I play solo, in band contexts it would take more time to adapt.

12. What are the main limitations of the system in order to use it in a live scenario?
    Answer: The wearable interface system might be a limiting factor for some in certain situations, but considering the particularity of the instrument, it is something that you budget for in order to have this kind of experience.

13. What should we carry on doing, and what could we do better, in order to work with artists like you to develop cutting edge performance technologies that you would use on concerts?
    Answer: Continue on this path by listening to feedback from testers in a passionate manner.

6. Conclusions

The objective of this research is to propose an innovative interaction method based on electromyographic signals and deep learning analysis, integrated into a DMI for guitarists; in particular to develop a tool that musicians may use to sonify their gestures during the performance, giving to the musician an innovative medium to explore the instrument.

One of the main difficulties in analyzing the sEMG signal comes from its noisy characteristics. Compared to other biosignals, sEMG provides a lower Signal to Noise Ratio (SNR) caused by inherent equipment noise, electromagnetic radiation, motion artifacts, and the interaction of different neighboring muscle, called cross-talk. Hence, a preprocessing stage is needed to filter out the unwanted noises. This difficulty becomes more critical when resolving a multiclass classification problem. To increase the classification performance, therefore, we adopted multi-channel EMG sensors to detect relevant sEMG patterns by a combined signal analysis, together with a strict muscle selection based on the specific gesture vocabulary.

With the proposed data acquisition protocol the best muscle groups for guitarist gestures classification have been selected, the results were used for the implementation of the proposed tool, which is able to modulate a set of effects accordingly to the user's muscle activation signals, consequently changing the sound of the guitar during a performance stimulating creativity and exploration. To assess this result, the outcome of a questionnaire, proposed in 5.5, has been analyzed after the final experiment. We have described in depth a method to integrate the sEMG signal into the development of a DMI for guitarists, but it is applicable to many other instruments, by using a custom Gestural Interface able to handle full body mapping. The two proposed Deep learning models gather information related to the gestures and the amount of exerted contraction, communicating with the Max/Msp patch, where the digital pedal board was implemented, to change the sound in real time. Furthermore, an evaluation strategy based on a questionnaire to collect feedback from tester musicians has been presented, with the aim of paving the way for other researchers interested in integrating muscle signals into artistic performances. Finally, the machine learning model is designed to be lightweight and thus can be easily inserted and run on embedded systems, typical of wearable computers.

6.1. Future Works

This work is preparatory to the Interactive Machine Learning (IML) paradigm, that could be applied into the training process. IML is a supervised learning technique that designs and implements algorithms to facilitate the learning process with the help of human interaction. One of the most notable advantages of using IML in this domain is that mappings between gesture and sound can be interactively "shown" by the user to the system, capable of "learning" them [19] instead of being manually coded, which could become excessively complex and time-consuming and it needs specific skills uncommon among musicians. In addition, the method explained for the muscle selection can be extended to other instruments and musicians. Furthermore, users suffer from the inconvenience of carrying several cabled electrodes. To solve this issue, our partner LWT3 is prototyping a sensor suit that is able to embed the cables in elastic tissue.

References

[1] Md Rezwanul Ahsan, Muhammad I Ibrahimy, Othman O Khalifa, et al. Emg signal classification for human computer interaction: a review. *European Journal of Scientific Research*, 33(3):480–501, 2009.

[2] Zainal Arief, Indra Adji Sulistijono, and Roby Awal Ardiansyah. Comparison of five time series emg features extractions using myo armband. In *2015 international electronics symposium (IES)*, pages 11–14. IEEE, 2015.

[3] Panagiotis K Artemiadis and Kostas J Kyriakopoulos. Emg-based control of a robot arm using low-dimensional embeddings. *IEEE transactions on robotics*, 26(2):393–398, 2010.

[4] Manfredo Atzori and Henning Müller. The ninapro database: a resource for semg naturally controlled robotic hand prosthetics. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7151–7154. IEEE, 2015.

[5] Reza Bagherian Azhiri, Mohammad Esmaeili, and Mehrdad Nourani. Real-time emg signal classification via recurrent neural networks. In *2021 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 2628–2635. IEEE, 2021.

[6] Rajesh G Bachu, S Kopparthi, B Adapa, and Buket D Barkana. Voiced/unvoiced decision for speech signals based on zero-crossing rate and energy. In *Advanced techniques in computing sciences and software engineering*, pages 279–282. Springer, 2010.

[7] M. Barbero, R. Merletti, and A. Rainoldi. *Atlas of Muscle Innervation Zones: Understanding Surface Electromyography and Its Applications.* Springer Milan, 2012.

[8] Jeronimo Barbosa, Joseph Malloch, Marcelo M Wanderley, and Stéphane Huot. What does" evaluation" mean for the nime community? 2015.

[9] Marco E Benalcázar, Andrés G Jaramillo, A Zea, Andrés Páez, Víctor Hugo Andaluz, et al. Hand gesture recognition using machine learning and the myo armband. In *2017 25th European Signal Processing Conference (EUSIPCO)*, pages 1040–1044. IEEE, 2017.

[10] Eleonora Borgia. The internet of things vision: Key features, applications and open issues. *Computer Communications*, 54:1–31, 2014.

[11] Remco R Bouckaert, Eibe Frank, Mark A Hall, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H Witten. Weka—experiences with a java open-source project. *The Journal of Machine Learning Research*, 11:2533–2541, 2010.

[12] Till Bovermann, Alberto de Campo, Hauke Egermann, Sarah-Indriyati Hardjowirogo, and Stefan Weinzierl. *Musical instruments in the 21st century.* Springer, 2017.

[13] Ricardo Buettner, Janek Frick, and Thilo Rieg. High-performance detection of epilepsy in seizure-free eeg recordings: A novel machine learning approach using very specific epileptic eeg sub-bands. In *ICIS*, 2019.

[14] Domenico Buongiorno, Giacomo Donato Cascarano, Antonio Brunetti, Irio De Feudis, and Vitoantonio Bevilacqua. A survey on deep learning in electromyographic signal analysis. In *Intelligent Computing Methodologies: 15th International Conference, ICIC 2019, Nanchang, China, August 3–6, 2019, Proceedings, Part III 15*, pages 751–761. Springer, 2019.

[15] Filipe Calegario, Marcelo M. Wanderley, Stéphane Huot, Giordano Cabral, and Geber Ramalho. A method and toolkit for digital musical instruments: Generating ideas and prototypes. *IEEE MultiMedia*, 24(1):63–71, 2017.

[16] Franca Delmastro, Flavio Di Martino, and Cristina Dolciotti. Cognitive training and stress detection in mci frail older people through wearable sensors and machine learning. *Ieee Access*, 8:65573–65590, 2020.

[17] Cagri Erdem, Qichao Lan, Julian Fuhrer, Charles Patrick Martin, Jim Tørresen, and Alexander Refsum Jensenius. Towards playing in the'air': Modeling motion-sound energy relationships in electric guitar performance using deep neural networks. In *Proceedings of the SMC Conferences*, pages 177–184. Axea sas/SMC Network, 2020.

[18] Cagri Erdem, Qichao Lan, Julian Fuhrer, Charles Patrick Martin, Jim Tørresen, and Alexander Refsum Jensenius. Towards playing in the'air': Modeling motion-sound energy relationships in electric guitar performance using deep neural networks. In *Proceedings of the SMC Conferences*, pages 177–184. Axea sas/SMC Network, 2020.

[19] Jules Françoise, Norbert Schnell, and Frédéric Bevilacqua. Mad: mapping by demonstration for continuous sonification. In *ACM SIGGRAPH 2014 Studio*, pages 1–1. 2014.

[20] Steven Gelineck and Stefania Serafin. A quantitative evaluation of the differences between knobs and sliders. In *Proceedings of the 2009 conference on New Interfaces for Musical Expression (NIME09)*, 2009.

[21] Arjan Gijsberts, Manfredo Atzori, Claudio Castellini, Henning Müller, and Barbara Caputo. Movement error rate for evaluation of machine learning methods for semg-based hand movement classification. *IEEE transactions on neural systems and rehabilitation engineering*, 22(4):735–744, 2014.

[22] Rolf Inge Godøy and Marc Leman. *Musical gestures: Sound, movement, and meaning.* Routledge, 2010.

[23] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep learning.* MIT press, 2016.

[24] Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*, 2020.

[25] Awni Y Hannun, Pranav Rajpurkar, Masoumeh Haghpanahi, Geoffrey H Tison, Codie Bourn, Mintu P Turakhia, and Andrew Y Ng. Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network. *Nature medicine*, 25(1):65–69, 2019.

[26] Hansika Hewamalage, Christoph Bergmeir, and Kasun Bandara. Recurrent neural networks for time series forecasting: Current status and future directions. *International Journal of Forecasting*, 37(1):388–427, 2021.

[27] Delsys Incorporated. Delsys—wearable systems for movement science, 2023.

[28] Imran Khan Mohd Jais, Amelia Ritahani Ismail, and Syed Qamrun Nisa. Adam optimization algorithm for wide and deep neural network. *Knowledge Engineering and Data Science*, 2(1):41–46, 2019.

[29] Matthias Janke, Michael Wand, and Tanja Schultz. A spectral mapping method for emg-based recognition of silent speech. In *B-Interface*, pages 22–31, 2010.

[30] Andrés Jaramillo-Yánez, Marco E Benalcázar, and Elisa Mena-Maldonado. Real-time hand gesture recognition using surface electromyography and machine learning: A systematic literature review. *Sensors*, 20(9):2467, 2020.

[31] Sergi Jordà. Instruments and players: Some thoughts on digital lutherie. *Journal of New Music Research*, 33(3):321–341, 2004.

[32] Chuck Jorgensen, Diana D Lee, and Shane Agabont. Sub auditory speech recognition based on emg signals. In *Proceedings of the International Joint Conference on Neural Networks, 2003.*, volume 4, pages 3128–3133. IEEE, 2003.

[33] Kirkpong Kiatpanichagij and Nitin Afzulpurkar. Use of supervised discretization with pca in wavelet packet transformation-based surface electromyogram classification. *Biomedical Signal Processing and Control*, 4(2):127–138, 2009.

[34] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *nature*, 521(7553):436–444, 2015.

[35] Duncan MacConnell, Shawn Trail, George Tzanetakis, Peter Driessen, Wyatt Page, and N Wellington. Reconfigurable autonomous novel guitar effects (range). In *Proceedings of the international conference on sound and music computing*. Stockholm Sweden, 2013.

[36] Takamitsu Matsubara and Jun Morimoto. Bilinear modeling of emg signals to extract user-independent features for multiuser myoelectric interface. *IEEE Transactions on Biomedical Engineering*, 60(8):2205–2213, 2013.

[37] Jim W Murphy, Ajay Kapur, and Carl Burgin. The helio: A study of membrane potentiometers and long force sensing resistors for musical interfaces. In *NIME*, pages 459–462. Citeseer, 2010.

[38] Kristian Nymoen, Mari Romarheim Haugen, and Alexander Refsum Jensenius. Mumyo–evaluating and exploring the myo armband for musical interaction. 2015.

[39] Angkoon Phinyomark, Chusak Limsakul, and Pornchai Phukpattaranont. Emg feature extraction for tolerance of white gaussian noise. In *Proc. International Workshop and Symposium Science Technology*, pages 178–183, 2008.

[40] Angkoon Phinyomark, Pornchai Phukpattaranont, and Chusak Limsakul. Feature reduction and selection for emg signal classification. *Expert systems with applications*, 39(8):7420–7431, 2012.

[41] Božidar Potočnik, Matjaž Divjak, Filip Urh, Aljaž Frančič, Jernej Kranjec, Martin Šavc, Imre Cikajlo, Zlatko Matjačić, Matjaž Zadravec, and Aleš Holobar. Estimation of muscle co-activations in wrist rehabilitation after stroke is sensitive to motor unit distribution and action potential shapes. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(5):1208–1215, 2020.

[42] W. Putnam and R.B. Knapp. Real-time computer control using pattern recognition of the electromyogram. In *Proceedings of the 15th Annual International Conference of the IEEE Engineering in Medicine and Biology Societ*, pages 1236–1237, 1993.

[43] P. J. Charles Reimer and Marcelo M. Wanderley. Embracing Less Common Evaluation Strategies for Studying User Experience in NIME. In *NIME 2021*, apr 29 2021. https://nime.pubpub.org/pub/fidgs435.

[44] Chris Rhodes, Richard Allmendinger, and Ricardo Climent. New interfaces and approaches to machine learning when classifying gestures within music. *Entropy*, 22(12):1384, 2020.

[45] Chris Rhodes, Richard Allmendinger, and Ricardo Climent. New interfaces and approaches to machine learning when classifying gestures within music. *Entropy*, 22(12):1384, 2020.

[46] Javier Rodriguez-Falces, Javier Navallas, and Armando Malanda. Emg modeling. *Computational Intelligence in Electromyography Analysis-A Perspective on Current Applications and Future Challenges*, pages 3–36, 2012.

[47] Farida Sabry, Tamer Eltaras, Wadha Labda, Khawla Alzoubi, and Qutaibah Malluhi. Machine learning for healthcare wearable devices: the big picture. *Journal of Healthcare Engineering*, 2022, 2022.

[48] Jorge Arturo Sandoval-Espino, Alvaro Zamudio-Lara, José Antonio Marbán-Salgado, J Jesús Escobedo-Alatorre, Omar Palillero-Sandoval, and J Guadalupe Velásquez-Aguilar. Selection of the best set of features for semg-based hand gesture recognition applying a cnn architecture. *Sensors*, 22(13):4972, 2022.

[49] Pavel Senin. Dynamic time warping algorithm review. *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, 855(1-23):40, 2008.

[50] Pekka Siirtola, Heli Koskimäki, Henna Mönttinen, and Juha Röning. Using sleep time data from wearable sensors for early detection of migraine attacks. *Sensors*, 18(5):1374, 2018.

[51] Atau Tanaka. Intention, effort, and restraint: The emg in musical performance. *Leonardo*, 48(3):298–299, 2015.

[52] Atau Tanaka, Balandino Di Donato, Michael Zbyszynski, and Geert Roks. Designing gestures for continuous sonic interaction. 2019.

[53] Atau Tanaka and Marco Donnarumma. The body as musical instrument. *The Oxford handbook of music and the body*, pages 79–96, 2019.

[54] Alejandro Toro-Ossaba, Juan Jaramillo-Tigreros, Juan C Tejada, Alejandro Peña, Alexandro López-González, and Rui Alexandre Castanho. Lstm recurrent neural network for hand gesture recognition using emg signals. *Applied Sciences*, 12(19):9700, 2022.

[55] Luca Turchet. Smart musical instruments: vision, design principles, and future directions. *IEEE Access*, 7:8944–8963, 2018.

[56] Luca Turchet, Michele Benincaso, and Carlo Fischione. Examples of use cases with smart instruments. In *Proceedings of the 12th international audio mostly conference on augmented and participatory sound and music experiences*, pages 1–5, 2017.

[57] Luca Turchet, Andrew McPherson, Carlo Fischione, et al. Smart instruments: Towards an ecosystem of interoperable devices connecting performers and audiences. In *Proceedings of Sound and Music Computing Conference*, pages 498–505. http://www. smcnetwork. org/, 2016.

[58] Luca Turchet, Fabio Viola, György Fazekas, and Mathieu Barthet. Towards a semantic architecture for the internet of musical things. In *2018 23rd Conference of Open Innovations Association (FRUCT)*, pages 382–390. IEEE, 2018.

[59] Federico Ghelli Visi and Atau Tanaka. Interactive machine learning of musical gesture. *Handbook of artificial intelligence for music: Foundations, advanced approaches, and developments for creativity*, pages 771–798, 2021.

[60] Jay M Weiss, Lyn D Weiss, and Julie K Silver. *Easy Emg-E-Book: A Guide to Performing Nerve Conduction Studies and Electromyography.* Elsevier Health Sciences, 2021.

[61] Matthew Wright. Open sound control: an enabling technology for musical networking. *Organised Sound*, 10(3):193–200, 2005.

[62] Michael Zbyszyński, Balandino Di Donato, Federico Ghelli Visi, and Atau Tanaka. Gesture-timbre space: Multidimensional feature mapping using machine learning and concatenative synthesis. In *Perception, Representations, Image, Sound, Music: 14th International Symposium, CMMR 2019, Marseille, France, October 14–18, 2019, Revised Selected Papers 14*, pages 600–622. Springer, 2021.

A. Appendix A

A.1. Model Customization

Gesture unfolds over time, and gestures that may look similar in terms of displacement in space may differ radically in expressivity depending on their temporal evolution; by incorporating user examples in the training step, the model can be customized to the artist's gesture vocabulary[59]. During the development of the training process of our model, We define a protocol to assemble the training dataset, with the goal of customizing the model according to the artist's taste. Our protocol also investigates how many sEMG different gesture acquisitions are needed to obtain a smooth interaction between the guitars effects and the musician's muscle's signal. In the first stage of this research, we investigated the best approach using two models in parallel: Dynamic Time Warping (DTW) for gesture classification, and Artificial Neural Networks(ANN) for the real-time tuning of the sonic preset through regression. These two model were provided by WEKA framework, an open-source collection of algorithms for data analysis and ML predictive modeling [11]. We used Dynamic Time Warping to deal with the gesture temporal modeling [49]; it is a technique that allows the temporal alignment of incoming time series (e.g. motion features changing over time) to previously saved gesture templates. The

DTM model was responsible for selecting the sonic preset of the pedalboard. it captured continuously the gesture match while running. For continuous regression we tested three algorithms : (i) linear regression (LR), (ii) polynomial regression (PR), and (iii) neural network (Multilayer perceptron ANN). We evaluated the ANN approach as the best way to implement regression; it was effectively able to continuously alter selected effects parameters during the performance based on muscle contractions. Merging the time-dependent DTW model with the regression one we obtained a time-evolving system, weighted by the artist's decision. In our application, the DTW network selects the sonic preset of the pedalboard created by the artist during the training stage. The gesture match was computed continuously while running. We downsample the input so examples have max length of 13 s. Continuous matches use a minimum length of 6 s, match width of 10 s and match hop size of 2 s. For continuous regression, we used a Multilayer perceptron ANN with 3 hidden layers, to continuously alter selected effect parameters during the performance based on muscle contractions. Merging the time-dependent DTW network with a regression model we obtain a time-evolving system, weighted by artistic decision. The user can tune continuously the sound effects parameters directly inside the model that was connected through an OSC network to the digital sound processor(i.e. Max/Msp 8 patch). Users may rely on tight action-feedback loops in which they modify model behavior through changes to the training data, followed by real-time experimentation with models to evaluate them and inform further modifications. This two model are free to access in the Git Hub page of this research [4].
This initial exploration process was critical to the development of a custom model and the creation of an ad hoc training dataset, clarifying all the issues that need to be considered during the development of a digital musical instrument that needs to function in real time.

Abstract in lingua italiana

Il seguente lavoro presenta lo sviluppo di un medotodo d'interazione innovativo che sfrutta la correlazione tra lo stato emotivo dell'artista e la sua attività muscolare, tramite l'utilizzo di sensori indossabili basati sull'elettromiografia di superficie (sEMG). A questo fine , è stato progettato un protocollo per tracciare dinamicamente le intenzioni musicali, adattando il suono della chitarra in tempo reale per trasmettere efficacemente le intenzioni sonore del musicista. Questo è stato ottenuto grazie alla progettazione di una rete neurale ricorrente (RNN) basata un una rete bidirezionale (BLSTM) per interpretare i segnali muscolari, attuando una classificazione di note tecniche chitarristiche. Per addestrare il modello di classificazione dei gesti, il musicista fornisce esempi gestuali propri della sua performace, assocaindone ogni uno ad un corrispondente preset sonoro. Per supportare future pubbicazini sarà pubblicato un dataset di acquisizioni sEMG relative a varie tecniche chitarristiche, creato durante la fase di allenamento, basato su vari soggetti. La combinazione tra la selezione delle migliori caratteristiche del segnale (i.e features), unita alla valutazione del miglior insieme di muscoli, ha permesso di ottimizzare il tasso di apprendimento del modello. L'elaborazione del segnale digitale è stata sviluppata utilizzando Max/Msp, per l'assemblento della pedaliera di effetti chitarristici. Infine, viene presentata una strategia di valutazione basata su un questionario per la raccolta di feedback. L'obiettivo finale di questo lavoro è quello di aiutare altri ricercatori nell'introduzione dei segnali muscolari come mezzo di interazione, durante performance artistiche.

Parole chiave: Interazione uomo macchina, elettromiografia di superficie, classificazione dei gesti, strumenti musicali digitali, LSTM bidirezionale

Acknowledgements

I want to thank all the LWT3 memebers for being like a second house for me during my intership while I was delevoping this project, especially Paolo Belluco for the continous support and all the vital advices, Samuele Polistina for the Max/msp support, Luigi Attoresi for the tailoring work to my torn pants, Roberto Alessandri for being the best fellow artist friend I could imagine. I also want to thank Giusy Caruso, who inspired me a lot and strengthened my dream of becoming an artistic researcher, I will become like you one day Giusy. I want to thank my professor Massimiliano Zanoni for the briliant knowledge that he gave to me during his course and the support for this project, I am very grateful to him for reminding me that you can also be an artist being a programmer.

I want to thank the three guinea pigs who sacrificed themselves for the cause and gave me their time, believing in my project, without you this work would be much weaker. Thank you very much alessandro, Cecilia and Cristian.

Then I really have to thank the great support that my family has given me. Without you guys believing in me none of this would have ever happened. I have been studying for 7 years now to combine my great passion for music with technology, only thanks to you who allowed me to dream, I can never show you just with words how grateful I am to you. I love you.

A very, very big thank you goes to my beautiful girlfriend Arianna, for the moral, physical, spice-physical, food and dairy, laughable, crazy, insurmountable support. Without you my journey would surely have been sadder and less playful.

I want to thank with a huge kiss my two best friends in Milan, Francesca and Annachiara, for giving me the most beautiful evenings, the laughter, the passions, the advices, the support, the crazy, drunken nights, the concerts, the dances, the madness, the trips to Sardinia, the tears...

I want to thank two special people I have had the honor of meeting over the Milan years, who always reminded me how beautiful life was, how spiritual a single tear or laugh could be, and how much beauty there can be in every simple smile. Thank you very much Franboscky and Antonio, I will never forget about you.

I would like to thank all the musicians I have jammed with over the years in piazza Leonardo and in Polifonia, especially the great Giack, who have given me thousands of inspiring ideas and encouraged me to write my own pieces.

Vorrei ringraziare il mio migliore amico da sempre che è rimasto al mio fianco nonostante la distanza, che mi ha sempre porso un abbraccio di sostegno quando ne sentissi la mancanza, che sarà sempre nei miei ricordi come il miglior veneto della nostra nazioone, un grande bacio al mio Nicolone. (fare le rime in inglese è una skill che non ho ancora curato)