UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

BACHELOR IN BIOMEDICAL ENGINEERING

BACHELOR THESIS

DESIGN AND IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS FOR DETECTION AND PREDICTION OF PATHOLOGICAL TREMOR USING EMG SIGNALS.

ALBERTO JOSÉ BELTRÁN CARRERO

2022

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN

BACHELOR IN BIOMEDICAL ENGINEERING

BACHELOR THESIS

DESIGN AND IMPLEMENTATION OF MACHINE LEARNING ALGORITHMS FOR DETECTION AND PREDICTION OF PATHOLOGICAL TREMOR USING EMG SIGNALS.

Author
ALBERTO JOSÉ BELTRÁN CARRERO

Tutor
ALEJANDRO PASCUAL VALDUNCIEL

Cotutor
ÁLVARO GUITÉRREZ MARTÍN

2022

Abstract

Essential tremor (ET) is one of the most common motor disorders, whose principal symptom is pathological tremor. This situation hampers or even impedes those patients to successfully accomplish their basic daily living activities. Nowadays, it does not exist efficient, affordable and wide-applicable treatments for managing tremor symptoms. Recent studies conducted investigations on new approaches based on the electrical stimulation of afferent pathways. This method employs strategies, such as out-of-phase and Selective Adaptive Timely Stimulation (SATS), in order to apply the electrical pulses to the muscles. They consist on stimulating a pair of antagonists muscles with electrical pulses, synchronized with periods of tremor activity. The main problem those strategies are facing is the prediction of future intervals of tremor. As a first approach, they based their predictions on tremor frequency analysis. However, this led to synchronization errors in long prediction windows. Also, reliable classification techniques are needed, in order to correctly identify tremor periods.

Some studies have evaluated the usage of kinematic signals for classification and prediction of tremor intervals, but few investigations were conducted in the sense of employing physiological signals instead. This thesis proposes and assesses the viability of using traditional machine learning (ML) algorithms (Gaussian Naive-Bayes (GNB), Random Forest (RF), K-Nearest Neighbors (KNN) and Support Vector Machines (SVM)), and deep learning (DL) models (Long Short-Term Memory (LSTM)) to classify electromyographic (EMG) signals into those presenting tremor activity and those that not. Classification task is performed using raw and filtered signals, as well as different sequence lengths (0.4 s, 0.6 s, 0.8 s and 1.0 s). Also, a LSTM based predictor is built and tested on prediction task of only EMG signals presenting tremor, using various prediction horizons. Classification datasets are composed by signals from 12 ET patients and 8 healthy subjects. Tagging process is conducted by means of signals' PSD values. The dataset for prediction only includes sequences from ET patients.

Best traditional ML algorithms were RF and SVM, reaching precision and recall values over 0.85 using raw and filtered signals, for each window length. Meanwhile, GNB and KNN had notable differences between precision and recall values, in all cases. However, every algorithm experimented improvements as shorter was the window length, as well as when using filtered signals instead of raw ones. The LSTM classifier did not improve the results obtained by traditional algorithms. Nonetheless, the model achieved greater accuracies on classifying raw signals. The LSTM predictor reached correlations over 0.9 for the next 100 ms period, being trained with 20, 30, 40 and 50 samples. For longer prediction windows, correlation values gradually decreased to 0.35 for the next 1.0 s period.

Results demonstrate that usage of ML and DL algorithms is an effective method to be implemented along with out-of-phase and SATS strategies, in order to identify tremor periods using EMG signals. However, prediction results, while promising, showed that this field needs for further investigation, so as to achieve acceptable results for longer prediction windows.

Keywords: tremor suppression, classification of tremor, prediction of tremor, essential tremor, electrical stimulation of afferent pathways, out-of-phase, SATS, machine learning, deep learning, LSTM.

Acknowledgments

For the record.

El miedo a estar solo solamente es superado por el amor en la buena compañía.

Como no hay mal ni bien que dure para siempre, no todos los días que nos depertamos lo hacemos cerca de alguien que quisiéramos que estuviera ahí, como tampoco faltan días en los que nos arrebate el sueño un beso o un abrazo. Hay veces en que los días los enfrentamos solos, pero en esos en los que hay alguien que te tiende la mano y no te pide nada a cambio, más que el esfuerzo de aprovechar su ayuda como merece, hay que ser consciente de la suerte y el deber que se tienen. Yo quiero agradecer esta ayuda a quienes no me dejaron solo.

Mi madre, irreemplazable como lo son todas, incansable como lo son por naturaleza, apoyo firme que ni el tiempo gasta ni se lleva; en todos los días de mi vida no podré alcanzar a devolver todo lo que te debo, aunque sepa que nunca me lo pidieras. Sin ti, ni este trabajo ni nada hubiera sido ni sería posible. Gracias.

Mi padre, mi hermano, referentes que me abrieron paso por un camino tan ignorado en la juventud, como veces en la historia transitado: el camino a ser buena persona, buen padre, buen hermano, buen hombre. Cada una de estas páginas tiene parte de vosotros. Gracias.

Mi familia y mis amigos, en ocasiones lo mismo, como en los árboles las raíces, no existiría sin ellas tronco, ni hojas, ni flores, ni frutos. He crecido gracias a vosotros, estando cerca o lejos, pero estando. Gracias.

A Alejandro, mi guía y director, que en su paciencia y tiempo siempre me ha guardado un hueco. Gracias.

A Álvaro, cuya no suficientemente agradecida confianza me ha traído hasta aquí. Espero, hoy, merecerla un poco más. Gracias.

Ciudad Real, junio 2022.

Contents

List of Figures

List of Tables

List of Acronyms

AI: Artificial Intelligence.

BPTT: Back Propagation Through Time.

DBS: Deep Brain Stimulation.

DL: Deep Learning.

EMG: Electromyography.

ECR: Extensor Carpi Radialis.

ET: Essential Tremor.

FCR: Flexor Carpi Radialis.

FES: Functional Electrical Stimulation.

GK: Gamma Knife.

GNB: Gaussian Naive-Bayes.

HIFU: High Intensity Focused Ultrasounds.

KNN: K-Nearest Neighbors.

LSTM: Long Short-Term Memory.

ML: Machine Learning.

MLP: Multilayer Perceptron.

NMDA: N-methyl-D-Aspartate.

PD: Parkinson's Disease.

PSD: Power Spectral Density.

RF: Random Forest.

RNN: Recurrent Neural Network.

SVM: Support Vector Machine.

UPDRS: Unified Parkinson's Disease Rating Scale.

VIM: Ventralis Intermedius.

Chapter 1

Introduction

Tremor belongs to a group of involuntary movements, such as tics, myoclonic jerk, chorea, athetosis, dystonia or hemiballism. It can be defined as involuntary, oscillatory and rhythmic movements of one or more body parts, not only including limbs [1]. It is not necessarily a sign of pathology, so there exist uncontrolled movements of body parts that are naturally present. This is called physiologic tremor and is common in situations of anxiety, fear, physical exhaustion, hypoglycemia, hyperthyroidism or alcohol withdrawal, among others [2]. Nonetheless, there are certain diseases whose primary or most acute manifestation is tremor, where it is considered as pathological. In fact, pathological tremor is one of the most common involuntary movement disorders assessed in clinical practice [1].

There are two main types of tremor: resting tremor and action tremor. The first type is related to involuntary movements that occur when the body part is relaxed, while the second type refers to tremor during voluntary movements [3]. Action tremor can also be divided into 3 subtypes:

- **Postural**: when the body part is voluntarily maintained against gravity.
- **Isometric**: during an stationary muscle contraction.
- **Kinetic**: may occur while doing any kind of voluntary movement. When tremor is amplified as the target object is reached, it is considered as a subtype of action tremor called kinetic-intention.

Pathological tremor is most common among middle-aged and older adults, due to neural and motor system ageing. Nevertheless, it can appear at any age. Some of the most important causes include: neurodegenerative diseases, stroke, head injury, drugs and toxins, demyelinating disorders, systemic illnesses or metabolic disorders [4]. The two principal diseases that are characterized by presence of tremor are Essential Tremor (ET) and Parkinson's Disease (PD) [5]. This thesis will primarily focus its investigation on ET.

1.1 Context

1.1.1 Problem description

ET is a chronic neurodegenerative disease [6] which has as primary manifestation kinetic tremor at hands and arms, even if it may eventually spread to other parts of the body. This tremor has components between 4 and 12 Hz [7] typically. Postural and rest tremor are also present in patients with ET, but it has been proved that their amplitudes are lower than those in kinetic tremor [8].

There is no consensus about the origin of ET, even a new conception of ET as a family of diseases, whose main feature is kinetic tremor of the arms, is increasing in popularity [9]. First studies in this matter proposed that a tremor pacemaker at inferior olivary nucleus was the responsible of ET [10]. Recent studies, including surgical, neurophysiological and postmortem, showed that there are other brain structures participating on tremor generation in ET: cerebellum, red nucleus, thalamus, and cerebral cortex (the cerebello–thalamo–cortical network) [5].

Regarding to impact of ET disease, some studies made across various countries since 1960 to 2019, showed that prevalence of ET in population older than 60 years is between 2,3% to 14,3% (median: 6,3%), as it increases with age [11]. But this kind of disorders not only end to motor dysfunctions, they also result in psychological problems (i.e. depression) that join difficulties to get daily life activities done [12]. In addition, it is known that the impact of motor dysfunctions, caused by diseases like ET or PD, will keep growing for the next years.

1.1.2 Therapeutic treatments

Nowadays, there exist several techniques that aim to suppress, in certain grade, pathological tremor in ET patients. They use different strategies to achieve the best possible results: from medication to surgical interventions or direct stimulation of brain structures and motor pathways.

1.1.2.1 Medication

Most common drugs used to deal with ET symptoms are propranolol and primidone. Recent studies have concluded that topiramate can be another first line drug for ET patients, so it reaches results close to those obtained with propranolol and primidone. Estimated improvement results of these drugs are shown at Table 1.1. Also, there are new alcohol based drugs in development, which are potent, but dangerous, solutions [13].

All pharmacological treatments for ET aim to handle with its symptoms, so patients can experiment an improvement on their quality of living, for as long as possible. However, drugs cannot be considered as complete solutions due to their inconveniences: they are not wide-applicable (only effective for around 50% of

Drug	Estimated improvement percentage in tremor amplitude
Propranolol	32-75
Primidone	42-76
Topiramate	30-41

Table 1.1: Most common used drugs for ET [13].

patients) [14], their side-effects leads to withdrawal of treatment in 33% of cases [15], patients can develop cognitive disorders after treatment period [16].

1.1.2.2 Thalamotomy

A thalamotomy is a surgical procedure which consists on an ablation of part of the thalamus using radio frequency through an opening, in order to improve brain function in patients. Generally, the target area for doing the ablation is the Ventralis Intermedius (VIM) [17]. Results achieved with this technique, in terms of tremor suppression, reach from 74% to 90% of total tremor [18]. Although these are good improvements, some disadvantages have to be considered: difficulties to localize intervention site precisely and irreversible partial destruction of thalamus [19].

There exists a variant of traditional thalamotomy based on Gamma Knife (GK) that use an external ablation strategy, so same process can be done through a non-invasive way. This is applicable to those patients who cannot be treated with traditional thalamotomy, due to comorbidity. Using this procedure, 63% of tremor reduction can be reached after 12 months [20]. By contrast, thalamotomy with GK has some side-effects to be studied for each case: hemiparesis, paresthesia, dysphasia, dysphagia and delayed neurological deficits [18].

1.1.2.3 High Intensity Focused Ultrasounds (HIFU)

The HIFU technique is a novel treatment for pathological tremor. It is a minimally invasive surgical process where target structures (ventral intermediate nucleus, subthalamic nucleus, and internal globus pallidus) are hit by ultrasounds, in order to conduct cerebral ablation. After 3 months, around 62% of patients can experiment an improvement, becoming around 80% after 12 months [21]. Other advantages of using ultrasounds are better precision and spatial resolution than those of thalamotomies [22]. Even though, some drawbacks of HIFU approach are that it is irreversible and its side-effects, namely: paresthesia, ataxia, gait instability and, less frequently, ischemic stroke, fingers dysesthesia and deep vein thrombosis [18].

1.1.2.4 Deep Brain Stimulation (DBS)

DBS is a surgical procedure in which implants are placed on target brain structures, whose function is to generate electrical pulses to modulate tremorgenic activity. Its main objective is to retain propagation of tremor through the cerebello–thalamo–cortical network using these pulses [23].

This procedure is an alternative to pharmacological and other approaches, with an improvement of around 50% on speech, tremor amplitude and bradykinesia. It also has the advantage of being reversible, unlike other strategies such as thalamotomy (see Section 1.1.2.2) or HIFU (see Section 1.1.2.3) [24].

However, the most inconvenience of DBS is the selective group of patients who can be treated using this method. There is a whole set of requisites to be satisfied by every patient who wants to be involved into a process of this kind, reducing the percentage of suitable subjects [25]. Moreover, there exist adverse events that may occur during the surgery (hypotension, seizures, symptomatic intracerebral hemorrhages, ischemic strokes) and after (wound infection, device malfunctioning). Typically, these events are notified in 2% of cases [24].

1.1.2.5 Functional Electrical Stimulation (FES)

FES technique is meant to be a promising, feasible alternative to the other seen approaches for handling with pathological tremor. It is based on using superficial or intramuscular electrodes [26] placed over muscles, that generate low level electrical pulses, above motor threshold, so these muscles can be activated, resulting in certain grade of contraction [27, 28, 29, 30]. It should be noted that applying pulses over the motor threshold has some inconveniences: fatigue, discomfort and interferences with voluntary movements [31, 32]. An example of a FES system is shown in Figure 1.1.

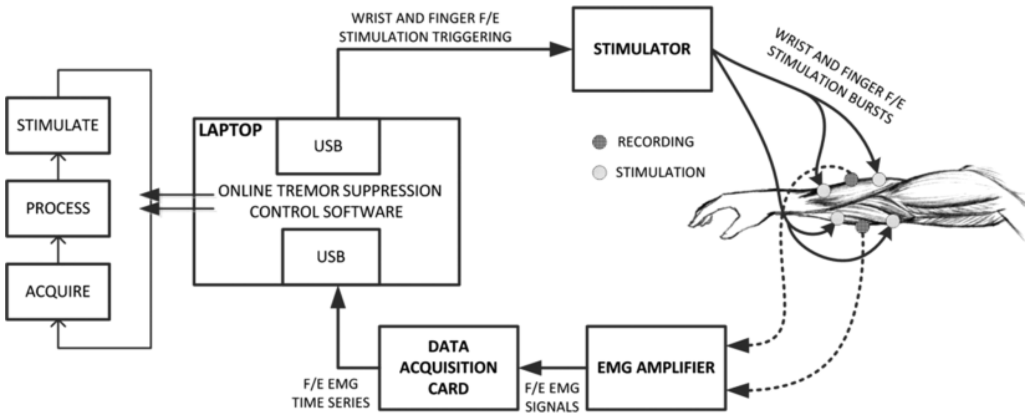


Figure 1.1: Workflow schema of a FES system [26].

Additionally, FES technique can be applied following two different strategies known as: out-of-phase and co-contraction.

- **Out-of-phase**: this strategy consists on muscle stimulation in order to generate opposite forces to those from tremor activity. Before stimulation period begins, either kinematic or electromyography (EMG) signals are analyzed. Then, tremor frequency and period are calculated so electrical pulses may be applied simultaneously to tremor intervals.

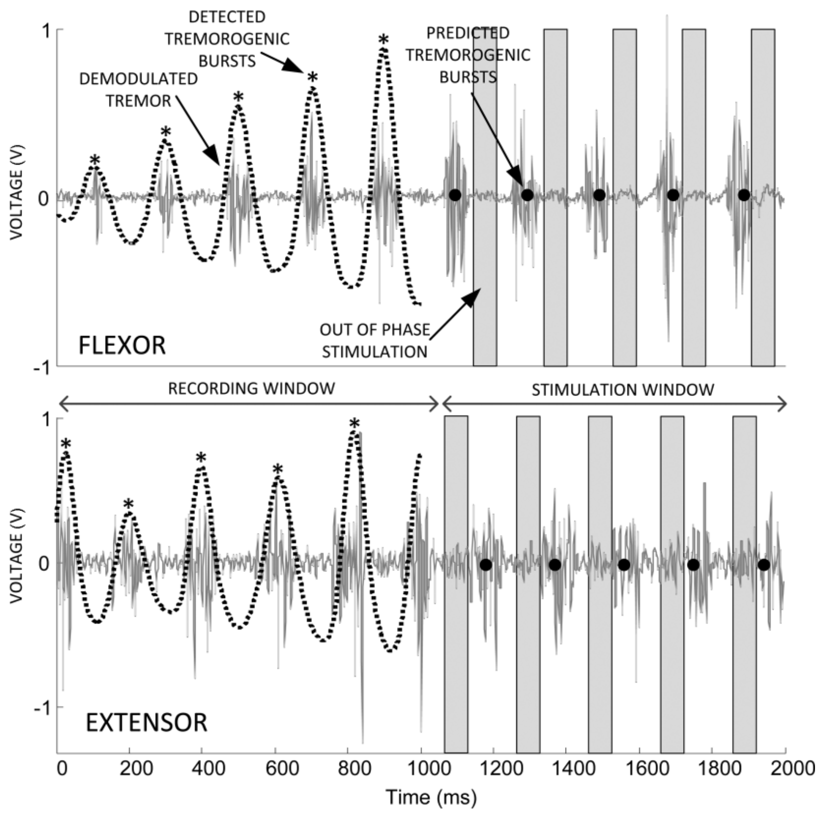


Figure 1.2: Out-Of-Phase strategy [26].

In more detail, when EMG signal are used, first step is to calculate the signal envelope so as to easily detect peaks of activity. After that, frequency and period of peaks are calculated. This is done for a pair of antagonist muscles: flexor/extensor. At this point, based on temporal metrics, it is possible to determine a prediction horizon for next peaks of activity and, using this information, stimulation periods can be intentionally planned for each antagonist muscle, so they generate opposite movement to that from tremor activity. An illustration of this strategy is shown in Figure 1.2.

The main disadvantage of this strategy is that a recording window without stimulation is needed, since electrical pulses generates artifacts in the EMG signals. That leads to a limitation for predicting future peaks, due to the non-stationary nature of EMG signals: as prediction horizon increases, synchronization of peaks will become more inaccurate.

- **Co-contraction:** as out-of-phase method, co-contraction aims to activate a pair of antagonists muscles at the same joint. Although, co-contraction is based on simultaneously activation of muscles, this leading to an increment of rigidity at the joint and forcing its stability. However, most important withdraw is that patients suffer stiffness and movement limitations [33, 34, 35].

From these two approaches, out-of-phase strategy is gaining relevance compared to co-contraction, due to its greater tremor suppression while applying fewer amount of pulses to muscles. A summary of different studies in these FES techniques is shown at Table 1.2.

Reference	Patients	Joint	Strategy	% Suppression
Javidan et al., 1992	3 ET, 4 PD	Wrist	Out-of-Phase	53±25%
Gilard et al., 1999	3 PD	Wrist/Finger	Out-of-Phase	83±2%
Grimaldi et al., 1999	1 ET, 2 PD	Wrist/Shoulder	Co-contraction	9±35%
Popovic Maneski et al., 2011	3 ET, 4 PD	Wrist	Out-of-Phase	67±13%
Widjaja et al., 2011	1 ET	Wrist	Out-of-Phase	57%
Gallego et al., 2013	2 ET, 2 PD	Wrist	Co-contraction	52±25%
Dosen et al., 2015	2 ET, 4 PD	Wrist	Out-of-Phase	60±14%
Jitkritsadakul et al., 2015	34 PD	Wrist	Co-contraction	44±33%

Table 1.2: FES studies and their results.

1.1.2.6 Electrical stimulation of afferent pathways

As an alternative to traditional FES methods, in the past years, some studies on stimulation of afferent pathways for tremor suppression have come with promising results. This approach seek to use spinal reflexes and interneurons for the purpose of cancel involuntary movements, still following opposite movement of antagonists muscles philosophy [36]. Afferent pathways suitable for being stimulated, seeing that they are involved into tremor activity, are Ia and Ib types. In addition, superficial and intramuscular electrodes are apt for its usage [37]. An schematic illustration of this method, using Ia afferent pathway, is shown in Figure 1.3.

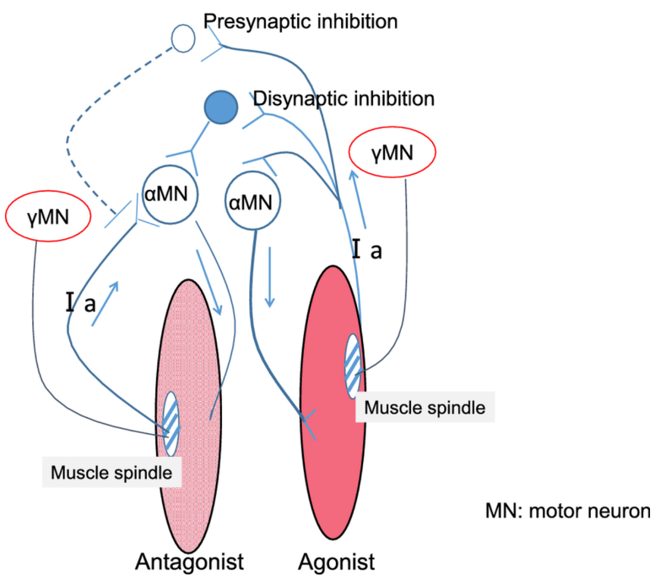


Figure 1.3: Reciprocal inhibition pathways in the spinal cord [38].

Stimulation of afferent pathways may use same strategies as FES (Out-of-Phase, Co-contraction). Nevertheless, an alternative approach is Selective Adaptive Timely Stimulation (SATS) technique. SATS seeks to overcome the prediction horizon and de-synchronization issues from out-of-phase, by using recording and stimulation windows sequentially [37] (see Figure 1.4).

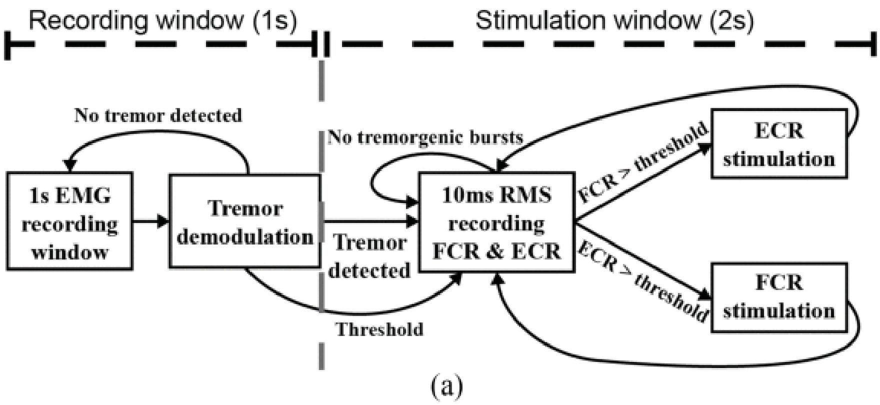


Figure 1.4: Control flow diagram of SATS strategy. [37].

Most interesting benefit of stimulating afferent pathways is that electrical pulses under motor threshold are able to reach tremor suppression, namely: an average of 52% [39]. That solves most of FES methods issues. Moreover, it does not have known side-effects yet, it is feasible for every subject, harmless and affordable [36]. Nonetheless, there are still improvements to be done, as variability of results is high (see Table 1.3).

Reference	Patients	Strategy	Pathway	% Suppression
Dosen et al., 2015	2 ET, 4 PD	Antagonist/Out-of-Phase	Ia	57±6%
Dideriksen et al., 2017	5 ET, 4 PD	IM-S/Out-of-Phase	Ia	52%
Muceli et al., 2019	2 ET, 4 PD	IM/Out-of-Phase	Ia	58%
Pascual-Valdunciel et al., 2020	9 ET	IM/SATS	Ia	32%

Table 1.3: Studies on afferent pathways stimulation. IM: intramuscular, S: superficial.

1.1.3 Machine Learning (ML) and Deep Learning (DL)

In the past 10 years, taking advantage of increasing ML and DL progress, new studies on classification and prediction tasks of tremor signals came out as a promising approach for dealing with involuntary movements. Most of them take ET and PD patients as study subjects and try to classify kinematic signals, for the purposes of identify tremor and non-tremor periods, classify tremor into unified types or evaluate improvements after surgery. The investigation in this thesis aims to develop ML and DL models, in order to evaluate their performance on classifying and predicting tremor signals.

ML refers to the design and implementation of computer programs, which can learn about data to take future decisions without human intervention. This process relays on algorithms capability of extracting rules and patterns from data. ML is also part of a bigger investigation field called Data Mining, whose purpose precisely is extracting unknown information from data by means of mathematical algorithms [40]. This leads to a two kinds of problems known as supervised learning and unsupervised learning. First regards to situations where tagged or expert reviewed data is available and the objective is to make predictions on new instances. The second refers to those problems where data is not tagged, so there is a descriptive task to be accomplished [41]. On this thesis, only supervised learning algorithms will be used.

Supervised learning is also divided into two categories, regarding to the kind of problem is faced: classification and regression. The objective of classification is to build a model capable of predicting an output class for input data. In order to do that, classification algorithms are previously fed with a training dataset, so they could fit their internal parameters based on it and *learn* about data features. On the other hand, regression refers to problems where main objective is to assign input values to a continuous function [41].

DL is a novel branch of ML that is meant to develop automatic algorithms capable of emulate human perception in different areas, such as natural language processing or image perception and description. DL algorithms are based on the mathematical model for human neurons, which is used to build complex neural networks using different architectures i.e: convolutional neural networks, recurrent neural networks (RNN), auto-encoders, etc. [42]. For the purpose of this thesis, most important architecture will be RNN, as they are meant to work with time series data.

1.1.3.1 Recurrent Neural Networks

The RNN are a group of neural networks specifically designed to work with sequential data, for example: time series information. Their main difference from other architectures is the ability to save memory states as they are processing data. This behavior is achieved by parameters share along multiple time steps. In order to share information across time, RNN implement sequence unrolling, which means that previous time steps outputs are connected to current time step input [43] (see Equation 1.1).

$$\mathbf{h}_t = f(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}) \tag{1.1}$$

where $\mathbf{h}_t, \mathbf{h}_{t-1}$ are RNN outputs of current and previous time steps, respectively; $\mathbf{W}$ is the current weights matrix, $\mathbf{U}$ the previous weights matrix, $\mathbf{x}_t$ the input vector and $\mathbf{b}$ the bias vector.

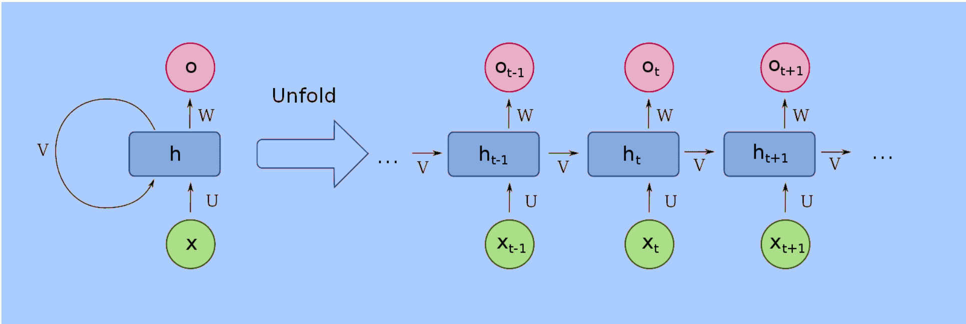


Figure 1.5: Folded and unfolded RNN architecture illustration [44].

By extending this process to the whole sequence, RNN achieve an approximation to memory notion [43]. An illustration of a RNN is shown in Figure 1.5.

Back Propagation Through Time (BPTT)

It is known that most of neural networks implement a mathematical procedure named back-propagation, in order to be able of refitting their parameters and *learn* from data. However, as RNN share their parameters across different time steps, while staying at the same iteration, this type of architecture needs for a new implementation of back-propagation which can take into account multiple time steps through the same train iteration. Figure 1.6 shows an illustration of how BPTT is performed.

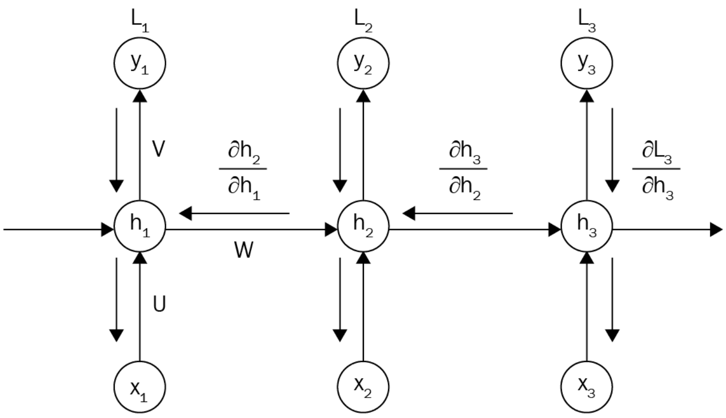


Figure 1.6: Example of a simplified BPTT computational graph [43].

BPTT includes time dimension on back-propagation process by making the loss function ($\mathcal{L}$) of the current time step dependent of previous ones. As there are many time steps to be computed, this dependency extends to the whole sequence [43]. At this point, the objective of BPTT is still the same as that of traditional back-propagation: minimize loss function, searching for best parameters values, based on gradient-descent algorithm. The weights of **W**, **U** matrices and **b** vector are updated as shown in Equations 1.2, 1.3 and 1.4.

$$\mathbf{W}_{t+1} = \mathbf{W}_t - \alpha \frac{\delta \mathcal{L}}{\delta \mathbf{W}_t} \tag{1.2}$$

$$\mathbf{U}_{t+1} = \mathbf{U}_t - \alpha \frac{\delta \mathcal{L}}{\delta \mathbf{U}_t} \tag{1.3}$$

$$\mathbf{b}_{t+1} = \mathbf{b}_t - \alpha \frac{\delta \mathcal{L}}{\delta \mathbf{b}_t} \tag{1.4}$$

where α is the learning rate.

Besides BPTT solves back-propagation for sequential data, as sequences increase in length, RNN become unable to maintain memory states from past time steps, that is, they cannot recognize long-term patterns. This is known as the vanishing gradients problem (see Appendix C). New architectures such as Long Short-Term Memory (LSTM) were created in order to overcome this issue.

Long Short-Term Memory (LSTM)

LSTM is a RNN based architecture designed to handle long-term dependencies in large sequential data, specifically, the vanishing gradient problem, in which long-term gradients tend to zero. The core of a LSTM is a memory cell which can maintain its state over time, and non-linear gating units able to regulate the incoming and outgoing information flow [45] (see Figure 1.7).

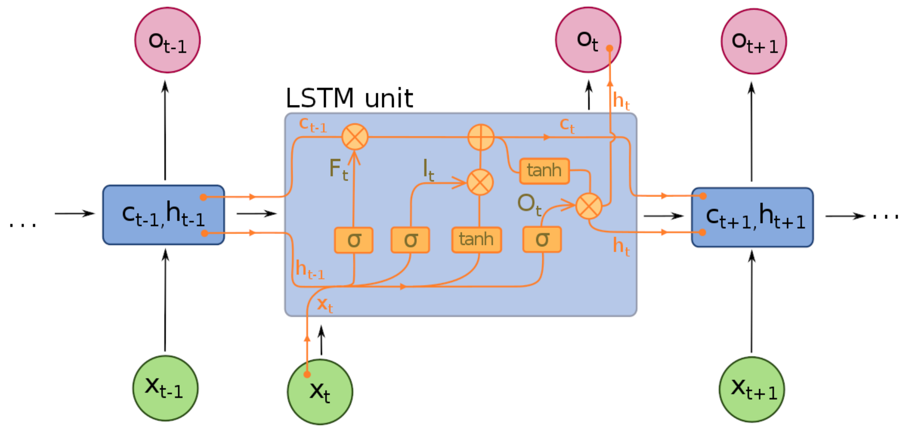


Figure 1.7: LSTM cell computational graph [44].

One LSTM cell is composed by 4 different gates, namely: input gate, forget gate, context gate and output gate. They are defined by Equations 1.5, 1.6, 1.7 and 1.8. Matrices first index corresponds the vector they process, while second index refers to the gate [43].

$$\mathbf{i}_t = \sigma(\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + \mathbf{b}_i) \tag{1.5}$$

$$\mathbf{f}_t = \sigma(\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + \mathbf{b}_f) \tag{1.6}$$

$$\mathbf{c}_t = \mathbf{f}_t \odot \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c) \tag{1.7}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + \mathbf{b}_o) \tag{1.8}$$

for $\mathbf{x}_t \in R^N$, where $\mathbf{x}_t$ is the input vector and N the feature length, at time step t, while $\odot$ represents the Hadamard product (element-wise multiplication).

Then, the output gate generates the final hidden state which is propagated to the next step:

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{c}_t) \tag{1.9}$$

It should be noted that $\mathbf{i}_t, \mathbf{f}_t, \mathbf{o}_t, \mathbf{h}_t, \mathbf{h}_{t-1}, \mathbf{c}_t, \mathbf{c}_{t-1}, \mathbf{b} \in R^H$, where H is the hidden state dimension, also known as the number of neurons per cell.

The purpose of these equations is to build a model for human memory notion. A roughly simplified way to explain this, based on the above equations, is that LSTM cells are building a representation for new ($\mathbf{i}_t$) and past ($\mathbf{f}_t$) information, by means of linear combinations between the input ($\mathbf{x}_t$), previous outputs ($\mathbf{h}_{t-1}$) and previous context ($\mathbf{c}_{t-1}$) (see Equations 1.5, 1.6), which can be considered as similar to what humans do. Then (see Equation 1.7), the current cell information ($\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + \mathbf{b}_c$) is filtered by applying the $\tanh$ function and an element-wise multiplication with the input. At this point, this information is added to that from an element-wise multiplication between the forget representation and the previous context, which still makes sense. Now, the output state is created using the new context vector (see Equation 1.8) and, finally, the new output is generated, based on new context and the output state (see Equation 1.9).

1.1.4 Application of ML and DL algorithms on tremor signals

In 2013, *Julien Stamatakis et al.* [46] used 18 extracted features from kinematic signals of PD patients to classify tremor, following an standard scale named Unified Parkinson's Disease Rating Scale (UPDRS). Results reached around 87% of accuracy. *Jeon et al.* [47] searched for the same performance by extracting 19 features from wrist kinematic signals. They trained a Support Vector Machine (SVM) whose accuracy ascended to 85.5%. In 2016, *Alam et al.* [48] determined tremor grade in PD by means of portable inertial sensors at finger/wrist and a SVM algorithm. Its accuracy reached 88.6% at rest and 78.8% for postural tremor.

In 2015, *Le Moyne et al.* [49] worked in the development of ML algorithms that could identify presence of tremor correctly, on stimulation and no-stimulation periods of DBS. Their SVM achieved 100% of accuracy.

Subsequently, a recent study carried by *Alves et al.* [50] in 2020 compared the performance of a set of ML algorithms on classifying signals from healthy and PD patients. Combining different amount of extracted features (272, 190, 136, 82 y 27), using various time intervals (1, 5, 10, 15 s), they proved that most relevant features of those signals were: average frequency, linear prediction coefficients, power ratio, power density skew and kurtosis. Also, their best algorithm was a K-Nearest Neighbor (KNN) with 90% of accuracy. In 2020, *Shahtalebi et al.* [51] proposed an architecture of bidirectional recurrent deep neural nets, called PHTNet, whose main objective was to characterize pathological tremor. Its main novelty was the separation between voluntary and tremor movements, without assuming that their spectral components should be different, which is quite inaccurate as there are superpositions between them. This project employ a dataset of signals from 81 ET and PD patients. However, this method has never been applied.

By contrast, prediction task is not as present as classification, but there are still a few studies on this field. In 2019, *Zanini et al.* [52] implemented different neural net based models (Multilayer Perceptron (MLP), MLP autoencoder, LSTM and LSTM autoencoder) for predicting EMG signals from a PD patient. They employed both raw and filtered EMG sequences of 60 s of duration and tested algorithms on predicting task. As correlation become smaller as prediction window enlarges, in this study the defined horizon was fixed to 200 ms, still under the required window size for tremor frequencies. Nonetheless, results were promising. The best model performing prediction, using raw and filtered EMG, was the MLP decoder with a linear combination of 3 layers (see Figure 1.8).

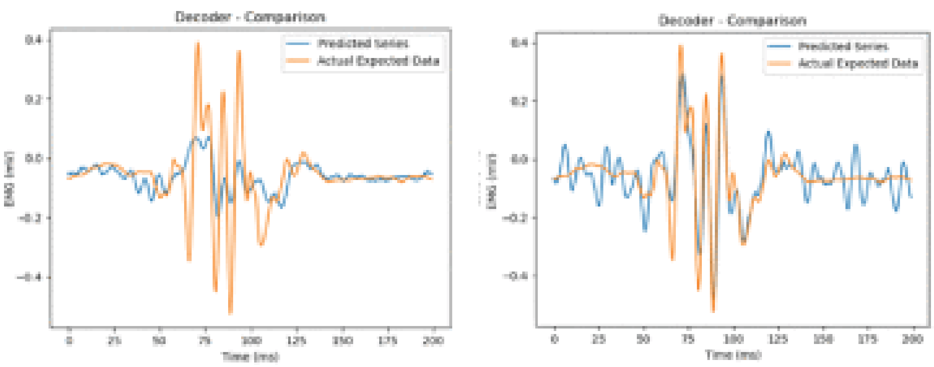


Figure 1.8: Raw EMG prediction examples using the MLP decoder [52].

At the same time, MLP and LSTM models based on a combination with autoencoder architectures performed better. They conclude that those DL models can successfully predict EMG tremor behavior, not only by means of the EMG envelope, but also using raw EMG sequences. Another recent study on tremor signals prediction was developed in 2020 by *Ibrahim et al.* [53]. This time data were composed by kinematic signals from 13 PD patients, acquired using Inertial

Movement Units (IMU). The selected model was a 2-layer CNN net followed by a MLP as the output layer. They only assessed this model on prediction horizons of 10 ms, 20 ms, 50 ms and 100 ms, where it achieved very good results: over 90% of prediction percentage accuracy in all cases (see Figure 1.9).

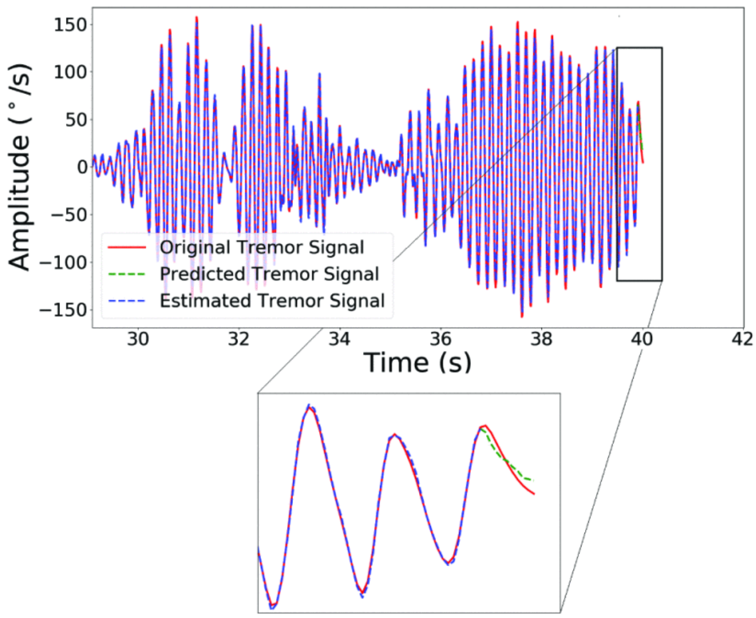


Figure 1.9: Prediction and estimation of true tremor kinematic signal. Green scattered line corresponds to last 10 ms predicted segment. [53].

Finally, they concluded that the development of wearable suppression devices, based on tremor activity prediction by means of DL models, can be a feasible approach for handling tremor. Nevertheless, there is more investigation to be done, in order to increase prediction windows as much as possible.

1.2 Motivation

Throughout Section 1.1, impact of ET, in terms of patients number and future trends, has been explained, as well as the state of art about methods for handling pathological tremor. In conclusion, today, most patients are treated using traditional pharmacological treatments. These solutions have several drawbacks, regarding to patient quality of living, due to all well known side-effects of medications, in exchange of temporal and inconsistent improvements. Surgical procedures, such as thalamotomy, DBS and HIFU appeared as suitable alternatives to drugs. However, their highest disadvantage is the surgery itself, as it could lead to difficulties and adverse events, even during and after intervention (hemiparesis, paresthesia, dysphasia, dysphagia, ischemic stroke, deep vein thrombosis, wound infection, device malfunctioning, among others). Moreover, patients do not experiment quite an improvement until several months since surgery was completed. Finally, these procedures, excepting DBS, are irreversible.

Therefore, this field claimed for new strategies to handle tremor activity, that were meant to overcome all of these inconveniences. Then, FES systems and studies on afferent pathways stimulation came with modern, minimal or even non-invasive methods. First, FES experimented a huge improvement on its results when applying more sophisticated ways to stimulate muscles, such as out-of-phase and co-contraction. Despite of still generating pulses over motor threshold, out-of-phase reduced the amount of electrical stimulation applied to muscles, then achieving a more comfortable experience for patients. It is a better alternative than even co-contraction, which leads to barely be unable to perform voluntary movements. Afferent pathways stimulation solves complications of stimulating over motor threshold, showing decent results with lower pulse amplitude. In addition, SATS strategy means better optimization for stimulation. Unfortunately, for both out-of-phase and SATS based systems, there exists an important limitation which is prediction window length.

As was discussed in Section 1.1.4, the application of new advances on Artificial Intelligence (AI) field could bring solutions to those kind of problems, electrical stimulation techniques are facing to. The main objective is to achieve a synchronous stimulation with tremor activity, using least possible number of electrical pulses. For this purpose, first, it is fundamental to classify signals, either kinematic or physiological ones, in order to identify if they corresponds to tremor or not-tremor activity, and do not begin stimulation period until it is really necessary. Then, algorithms should be able to deal with noise and artefacts, achieving high levels of accuracy on signals classification. At this point, they also should accomplish a prediction task, so systems could anticipate next stimulation and no-stimulation periods, that is, future tremor activity. Recent DL models, specially those based on RNN such as LSTM, have been designed specifically to perform well in time series prediction.

In this context, EMG signals might be the best alternative to be used in afferent pathways stimulation approaches. First of all, EMG represents physiological activity directly, without any transformation, as it is done, for example, in methods that use kinematic signals. Therefore, working with EMG may be the most feasible way to synchronize stimulation with activity from the nervous system, as final purpose is to employ afferent spinal pathways and reflexes to interfere with tremor.

In conclusion, ML and DP bring hope about future, more powerful systems, capable of predicting either kinematic or EMG signals for a sufficient interval of time, knowing that typical tremor frequencies are between 4 and 12 Hz. That is the main motivation for investigations that will be carry out on this thesis.

1.3 Objectives

This investigation aims to develop a comparative analysis between various ML algorithms, including traditional ones and DL models, based on their performance in classification and prediction tasks for EMG signals from patients who suffer pathological tremor. All these algorithms will be fed with raw and filtered signals, searching for differences in terms of performance metrics, as employing raw data means an advantage in terms of reducing pre-processing work. Also, that could be useful for implementing the algorithms into portable devices.

The study consists on 3 stages:

1. Design scripts for building databases containing EMG signals (raw or filtered), which should be able to admit different parameters of sampling rate and different window lengths. Data would be tagged using EMG features.
2. Design Machine Learning models for classification and prediction of EMG signals. There might be included traditional algorithms, such as Random Forest, KNN, SVM and Naive Bayes, and more complex neural networks based models, such as LSTM. Evaluate each model results individually.
3. Results assessment between models, focusing on their performance in terms of signal parameters (window length, sampling rate) and the differences between using raw or filtered signals.

The essential purpose of this thesis is to bring a wide set of results, which could help in the design of modern tremor suppression systems that use ML/DL algorithms, in order to provide personalized treatments adapted to patient-specific pathological tremor.

1.4 Hypothesis

Pathological tremor has a characteristic frequency, which oscillates between 4 and 12 Hz as discussed in Section 1.1.1. Therefore, a potential way to head classification task, using only EMG signals, should be tag the sequences based on tremor frequency band, while taking into account that they are not stationary (their properties change through time). Then, this method may be useful for creating the datasets.

For its part, it is well known that EMG signals are noisy and difficult to analyze, but working with ML and DL algorithms may help to avoid these inconveniences. For the purpose of evaluate their ability to handle with noisy signals, raw and filtered data are going to be used. It is expected that most algorithms will perform better with filtered signals, while still giving good results with raw data.

Traditional ML algorithms are supposed to bring performances, at least, over 70% in precision and recall metrics on classification task, using raw signals. For filtered signals, those metrics should increase to over 80% or more. On the other hand, DL models performance might be greater or, at least, equal to that of traditional algorithms, using both kind of signals, due to their more complex architecture.

Finally, expected results for prediction task are correlations between real and predicted signal above 0.7, for the first 10 samples, which corresponds to a window of 200 ms with a sample frequency of 50 Hz. Prediction horizon will be increased to a maximum of 50 samples (1 s), yet correlations are supposed to be below 0.5 as a consequence of using highly noisy signals.

Chapter 2

Materials and Methods

Throughout the following chapter, there will be explained details about data used for this study, dataset creation processes and ML and DL models for classification and prediction tasks, as well as the development tools employed during the experiments.

2.1 Materials

Twelve ET patients were selected from the Movement Disorders Clinic of Gregorio Marañón Hospital (Madrid, Spain), between April 2019 and January 2020, to participate on an experiment [37] whose objective was to evaluate if muscle afferent stimulation could be a feasible approach to reduce pathological tremor. Those patients were clinically examined by movement disorders specialists of the Neurological Department and satisfied the following criteria: diagnosis of ET according to Tremor Research Investigation Group criteria [54], present clinically postural tremor; age between 18-80 years; tremor affecting at least one of the upper limbs, with prominent wrist flexion-extension; absence of another neurological or musculoskeletal pathology; ability to understand the procedure and sign the informed consent. In addition, patients that were under any anticoagulant treatment, presented coexistence of other diseases that distort movement; or mixed or complex tremors, with involvement of multiple muscles and concomitant important medical pathology, were excluded.

2.1.1 Data acquisition and description

On the study performed by *Alejandro Pascual-Valdunciel et al.* [37], kinematic and EMG signals were recorded from ET patients. For the purpose of this thesis, only EMG signals were of interest.

Patients underwent surface (SurfStim) stimulation experiments. Before each session, a neurologist evaluated their basal condition, based on the Fahn-Tolosa-Marín [55] tremor rating scale (specific and motor tasks) and the Clinical Global Impression of Severity (CGI-S)/of Change (CGI-C) [56]. Bipolar surface electromyography (sEMG) electrodes were placed over the muscle belly of FCR and ECR, after cleaning the skin with alcohol. Surface EMG signals were acquired at 2042 Hz [37].

Apart from recruited ET patients, another eight healthy subjects participated on the study, under the same conditions exposed before. EMG signals from FCR were recorded via surface electrodes, while subjects were performing voluntary flexor contraction. These data were employed only for the classification task.

In terms of recording time, 140 records of 60 s where finally acquired from ET patients, while 76 records of 34 s came from healthy subjects. It should noted that there are two EMG records (ECR and FCR) for each of the 140, and only one (FCR) from those 76 of healthy patients.

2.1.2 Pre-processing

Besides all records included EMG signals, not every sequence satisfied minimum requisites for signal quality, so as to be acceptable on this study. After sessions were ended, each record was assessed and tagged according to its quality, based on visual inspection performed by an expert in EMG tremor signals. A 3-class scale was established as following: 0 (Not-Acceptable), 1 (Acceptable) and 2 (Good). For the purpose of this investigation, only EMG records tagged as 1 or 2 quality classes were considered and included into the datasets. Table 2.1 shows number of useful records per class.

	EMG-ECR	**EMG-FCR**
Acceptable (1)	23	46
Good (2)	41	10
Total	64	56
Available records	120	

Table 2.1: Number of useful EMG records.

Nevertheless, this criteria only applies for records from ET patients, so those from healthy subjects all satisfied requisites of signal quality.

Once data were selected by quality, an undersampling process was performed. The reason of doing this is, primarily, reducing computational cost of processing a great amount of samples per sequence, while it is not strictly necessary: the objective is making algorithms capable of rapidly identify typical tremor components in EMG signals, which are between 4 and 12 Hz. Then, undersampling helps algorithms to work faster, as no relevant signal information is lost. Raw EMG signals were undersampled by a 4 factor, that is, to 510 Hz from 2042 Hz; filtered signals sample rate was reduced to 50 Hz. Figure 2.1 shows a raw EMG signal example.

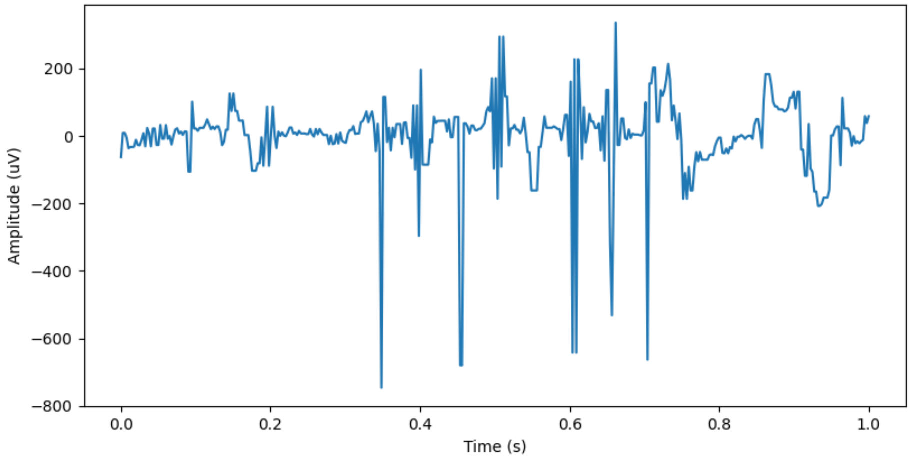


Figure 2.1: Example of raw EMG signal.

For filtered signals datasets, each sequence was passed through a Butterworth bandpass filter of order 2 and cut frequencies 4 and 10 Hz, in order to obtain the envelop. This kind of filters does not have ripple in its passband and the transition between pass/removed bands is not very abrupt. Also, the digital filter was applied forward and backward to the signals, which means a zero phase distortion [57]. Figure 2.2 shows a filtered EMG signal example.

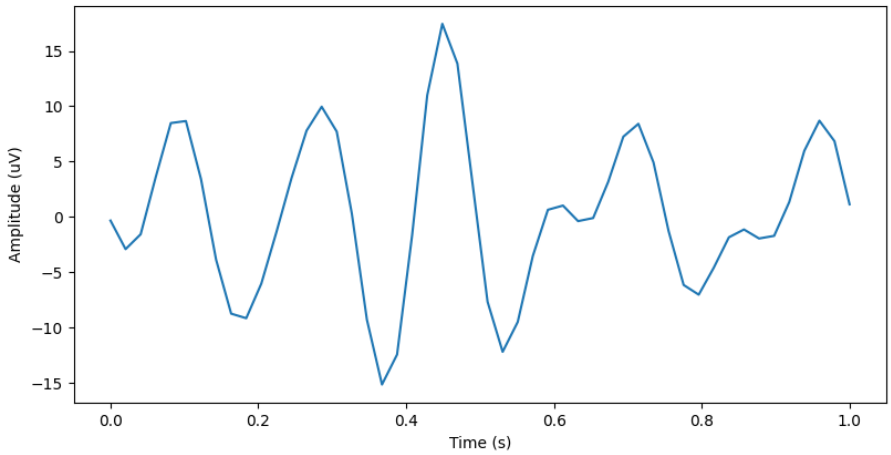


Figure 2.2: Example of filtered EMG signal.

At this point, either raw and filtered data were normalized in range $(0, 1)$. This is a good practice when working with ML and DL models, even a common requirement for some estimators which might behave badly if the individual features do not look similar to standard normally distributed data [58]. Then, two normalization algorithms known as *StandardScaler* (see Equation 2.1) and *MinMaxScaler* (see Equation 2.2) were employed (see Section 2.2.2.6).

$$z = \frac{(x - \mu)}{s} \tag{2.1}$$

where x is the sample value, μ the mean of all samples, s their standard deviation and z the standard score of sample x.

$$\begin{aligned}\mathcal{X}_{std} &= \frac{\mathcal{X} - \mathcal{X}_{min}}{\mathcal{X}_{max} - \mathcal{X}_{min}} \\ \mathcal{X}_{scl} &= \mathcal{X}_{std} \cdot (Max - Min) + Min\end{aligned} \tag{2.2}$$

where $\mathcal{X}$ is the sample, $\mathcal{X}_{max}$, $\mathcal{X}_{min}$ the maximum and minimum samples values, and Max, Min the limits of desired feature range, which in this case are 1 and 0, respectively.

Finally, it is important to mention that, as signals from patients were in mV and those from healthy subjects were in μV, all sequences were normalized to μV, in order to unify units.

2.1.3 Classification datasets

First stage of the investigation will be the classification of EMG signals in two classes: Tremor and No-Tremor. Besides, algorithms will be fed with sequences of different length. Window lengths selected for this study are: 1 s, 0.8 s, 0.6 s and 0.4 s. As there are two types of signals (raw and filtered), eight different datasets will be created in total. The defined strategy to tag the EMG sequences followed two steps:

1. Define global thresholds based on spectral components from signals to establish the limit between those presenting tremor and those that not.

2. Extract spectral components from each sequence, compare them to the corresponding threshold and tag it as Tremor or No-Tremor.

The selected spectral components to be used for this study were the Power Spectral Density (PSD) values, more specifically, the greatest value between 4 and 10 Hz. The PSD contains information about the percentage of total signal power that corresponds to each frequency component. Formal definition of PSD (see Equation 2.3) is the Fourier Transform of the correlation function, that is, the correlation between the signal and a copy of itself delayed a certain time τ [59, 60]. PSD was considered as an appropriate method for tag process, so it allows to identify objectively power distribution over typical tremor frequencies

$$S(f) = TF(R(\tau)) = \int_{-\infty}^{+\infty} R(\tau) \cdot e^{-j2f\tau\pi} d\tau \tag{2.3}$$

where R is the hop size window and k the number of available observations.

Nonetheless, as it is shown in Equation 2.3, the PSD is calculated through an infinite period of time, while the length of EMG signals is finite, in other words, available time information of the signal is limited. At this point, in substitution to calculating PSD, an spectral estimation will be performed. This is a probabilistic analysis that allows to calculate PSD values of a signal when using a finite number of observations (k). In order to use this kind of methods, signals must be considered as stationary (signal properties do not change through time) and ergodic (estimations average corresponds to real average) stochastic processes. There exist different methods to estimate PSD values. For the purpose of this study, Welch's method was employed. It is a non-parametric method that is based on the formal definition of PSD [59, 60]. In order to apply this method, a Hanning type window was used, as number of observations was 510 and 50 for raw and filtered signals, respectively.

At this point, six different instances of EMG signals, from each raw and filtered sets, were chosen to calculate thresholds. Those signals were selected manually so two came from patients whose tremor was qualitatively defined as *High*, two as *Low* and, finally, two signals from healthy patients. Then, every signal was passed through the same filter as explained in Section 2.1.2 (excepting those already filtered) and its estimated PSD was calculated. Due to the noise and artifacts present in EMG signals, PSD values were quite variable and, in some cases, those coming from healthy subjects signals were greater than values from patients' sequences. Figure 2.3 shows examples of extracted PSDs. In order to solve this situation, it was decided to calculate arithmetic means of maximum PSD values between 4 and 10 Hz and take the result as the threshold (see Equation 2.4).

$$\begin{aligned}
mean(HT_{1,i}, HT_{2,i}) &= HT_{m,i}, \\
mean(LT_{1,i}, LT_{2,i}) &= LT_{m,i}, \\
mean(HS_{1,i}, HS_{2,i}) &= HS_{m,i}, \qquad i \in W = \{1.0, 0.8, 0.6, 0.4\} \\
mean(HT_{m,i}, LT_{m,i}) &= P_{m,i}, \\
mean(P_{m,i}, Hs_{m,i}) &= T_i
\end{aligned} \tag{2.4}$$

where $HT_{1,i}, HT_{2,i}$ are PSD max. values between 4 and 10 Hz from *High* tremor signals, $LT_{1,i}, LT_{2,i}$ those from *Low* tremor, $HS_{1,i}, HS_{2,i}$ those from healthy subjects and T_i the threshold for window i.

It should be noted that PSD values change for different window lengths, so this procedure was repeated for each of the eight datasets. Defined thresholds are shown in Table 2.2.

	Raw EMG				Filtered EMG			
Window	1.0 s	0.8 s	0.6 s	0.4 s	1.0 s	0.8 s	0.6 s	0.4 s
PSD value	8.2012	7.8758	7.5504	4.5744	7.9881	7.8758	7.5504	7.3104

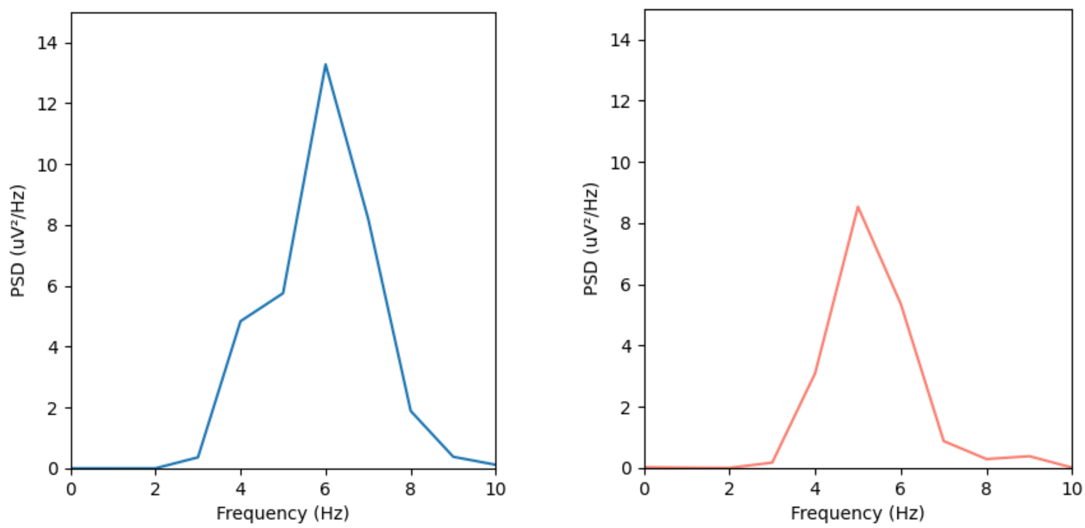
Table 2.2: PSD max. values between 4 and 10 Hz considered as tremor threshold.

The result of tag procedure, in terms of number of instances per class for each dataset is shown in Table 2.3. Number of instances per class and window is expected to be the same for raw and filtered signals, however, there are few differences, while proportions between Tremor and Not-Tremor instances are virtually consistent. The reason for this might be that there are sequences which PSD values should be closer to the threshold so, as thresholds minimally vary for filtered signals, some of those instances change their tag. The optimal method for tagging EMG sequences should be based on a consistent estimation for thresholds, using the PSD or other signal features. In this investigation, as a first approach, the estimation explained before, based on arithmetic means of PSD values from example sequences, conducted to acceptable results. Nonetheless, there is improvement to be done for future approaches.

	Raw EMG				**Filtered EMG**			
Window	1.0 s	0.8 s	0.6 s	0.4 s	1.0 s	0.8 s	0.6 s	0.4 s
Tremor	4882	5805	7162	10318	4734	6042	7576	10662
No-Tremor	5054	6543	9426	14610	5202	6306	9012	14266

Table 2.3: Total instances per class after tag process.

All datasets should be 50/50 balanced to avoid algorithms of becoming more likely to classify instances on one class. Moreover, for the purpose of helping algorithms, specially DL models, to be trained faster, select least possible number of instances is recommended. In conclusion, datasets are composed by: 8000, 10000, 14000 and 20000 instances (50% Tremor, 50% No-Tremor) for 1.0 s, 0.8 s, 0.6 s and 0.4 s window lengths, respectively.



(a) High tremor signal. *(b) Low tremor signal.*

Figure 2.3: Examples of PSD values between 1 and 10 Hz from signals with tremor.

For classification task, datasets have been divided into two subsets: train set (2/3 of total instances) and test set (1/3 of total instances). This method prevents algorithms from suffering of overfitting, so they become more capable of working with unseen data. For those more complex models (i.e DL ones), it is necessary to build another split of data, that will be used during validation process. At this point, half of the test set is separated as the validation set (1/6 of total instances).

2.1.4 Prediction dataset

Last part of this study will be the prediction of EMG signals. This task needs for another dataset, different from those built for classification, so this time it is not necessary to tag instances: the target will be another signal. Signals came only from ET patients, as the objective is predicting tremorgenic activity. Also, prediction will be performed only on filtered EMG signals, as predicting raw EMG could be excessively difficult for the model. For this purpose, sequences of 2 s length are built, so now each instance corresponds to two 1 s signals. This investigation consists on using different window lengths for training and various prediction horizons, but this will be implemented in model's own code, more specifically, in the training loop function. Moreover, dataset is built with 50% of overlapping between signals, that is, 1 s of overlap. The objective of doing this is to use every 1 s sequence available for training, otherwise those signals employed as targets would never be used for training. The dataset is composed by 7000 2 s sequences of filtered EMG.

2.2 Methods

2.2.1 Development tools

The experiments performed on this investigation have been implemented using *Python* programming language and some of its libraries and frameworks. *Python* brings usable, complete and powerful tools for data analysis and data mining to developers, what makes this programming language a great tool for working with data structures and building ML and DL models [61]. The main *Python* libraries that have been used on this investigation are: *NumPy*, *Pandas*, *SciPy*, *Scikit-Learn* and *Matplotlib*, along with an specific framework based on it named *Seaborn*.

2.2.2 Classification algorithms

2.2.2.1 Gaussian Naive-Bayes (GNB)

GNB is a variant of Naive Bayes methods. These consist on a set of supervised learning algorithms based on applying Bayes' theorem with the assumption of conditional independence between every pair of features, given the value of the class variable [58]. First of all, Bayes' theorem states that, given a class variable y and a vector of features $\mathbf{x} = (x_1, x_2, x_3, \cdots, x_n)$:

$$P(y|x_1, \cdots, x_n) = \frac{P(y)P(x_1, \cdots, x_n|y)}{P(x_1, \cdots, x_n)} \tag{2.5}$$

At this point, the "naive" assumption corresponds to conditional feature independence, that is:

$$P(x_i|y, x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n) = P(x_i|y) \tag{2.6}$$

for all i features in **x**.

This leads to a simplified version of Equation 2.5:

$$P(y|x_1, \cdots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \cdots, x_n)} \tag{2.7}$$

As $P(x_1, \cdots, x_n)$ is constant given the input, the following classification rule can be applied:

$$\begin{aligned} P(y|x_1, \cdots, x_n) &\propto P(y)\prod_{i=1}^{n} P(x_i|y), \\ \hat{y} &= argmax_y P(y)\prod_{i=1}^{n} P(x_i|y) \end{aligned} \tag{2.8}$$

Finally, by Maximum A Posteriori estimation, $P(y)$ and $P(x_i|y)$ can be estimated, being $P(x_i|y)$ the relative frequency of class y in the training set. The existent various methods based on Naive-Bayes differ on probability distribution assumed for $P(x_i|y)$. Among their main advantages, it should be mention that this methods require small amount of training data to estimate parameters, their training process is faster than most other algorithms and they achieve decent results in real-world situations. Nonetheless, Naive-Bayes classifiers are known for being bad estimators [58, 62].

GNB is a Naive-Bayes classifier used for binary classification tasks. Its particularity is that the likelihood of features is assumed to be Gaussian (see Equation 2.9) [58].

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_i^2}} \exp\left(-\frac{(x_i - \mu_i)^2}{2\sigma_i^2}\right) \tag{2.9}$$

where μ_i and σ_i are estimated using maximum likelihood.

2.2.2.2 K-Nearest Neighbors (KNN)

KNN is one of the most popular algorithms in ML, widely used for classification tasks. Its strategy is based on comparing data points to those in their "neighborhood", that is, the closest points to them. Namely, the principle behind KNN is to find a predefined number of training samples closest in distance to the new point, and predict the label from these. That number of samples can be defined as a constant (k). On the other hand, the distance can be any metric measure, such as Euclidean, Chebyshev or Minkowsky distances (see Equations 2.10, 2.11 and 2.12) [58].

$$d_{euc}(x,y) = \sqrt{\sum_{i=1}^{n}(x_i - y_i)^2} \tag{2.10}$$

$$d_{chb}(x,y) = \max_{i=1}^{n} |x_i - y_i| \tag{2.11}$$

$$d_{mnk}(x,y) = \left(\sum_{i=1}^{n} |x_i - y_i|^p\right)^{\frac{1}{p}} \tag{2.12}$$

where **x** and **y** are two n-dimensional vectors and p is the order of the Minkowski distance.

Classification is achieved through a simple majority vote of the nearest neighbors of each point: assigned data class to a point is that having the most representatives around it. The main disadvantage of KNN is that it employs every instance from the training set to classify a point, so it demands memory and compute time from the CPU.

2.2.2.3 Random forest (RF)

RF algorithm is, as well, another popular ML model frequently employed on classification problems. It is based on various decision trees working together [40].

First of all, a decision tree consists on a set of nodes and branches, where nodes correspond to some data features and branches represent decision rules. When initializing the algorithm, it starts searching for a certain number of features (that can be user specified), which are selected regarding to how well they differentiate data instances. In order to measure that, decision trees calculate the Information Gain from each node/feature. IG is defined as:

$$IG = E(ParentNode) - \sum(w_i E(ChildNode)) \tag{2.13}$$

where w_i is the relative size of the child node, that is, number of instances given by it over those given by the parent node. Also, E refers to node's entropy, which represents the grade of disorder in it [61]. Entropy is defined as:

$$E = \sum_{i=1}^{c}(-p_i \log_2 p_i) \tag{2.14}$$

where p_i is the probability that an instance belongs to one of the classes and c is the number of classes.

After this process, those features having greatest IG will be selected as nodes for the decision tree. RF repeats this procedure for a set of trees through the bagging method. It consists on dividing the training set into random N subsets (N is the number of decision trees employed) and fed each tree with one of them. This way, different trees never see the same data. After the feeding, each one generates an output y_i, which is the class assigned to the instance. Finally, that class receiving more votes become the output of the RF (see Figure 2.4).

Some advantages of using RF are their versatility and flexibility, as the user can specify number of trees, nodes, nodes' length, among others; and that it avoids well overfitting, due to the random segmentation of training data.

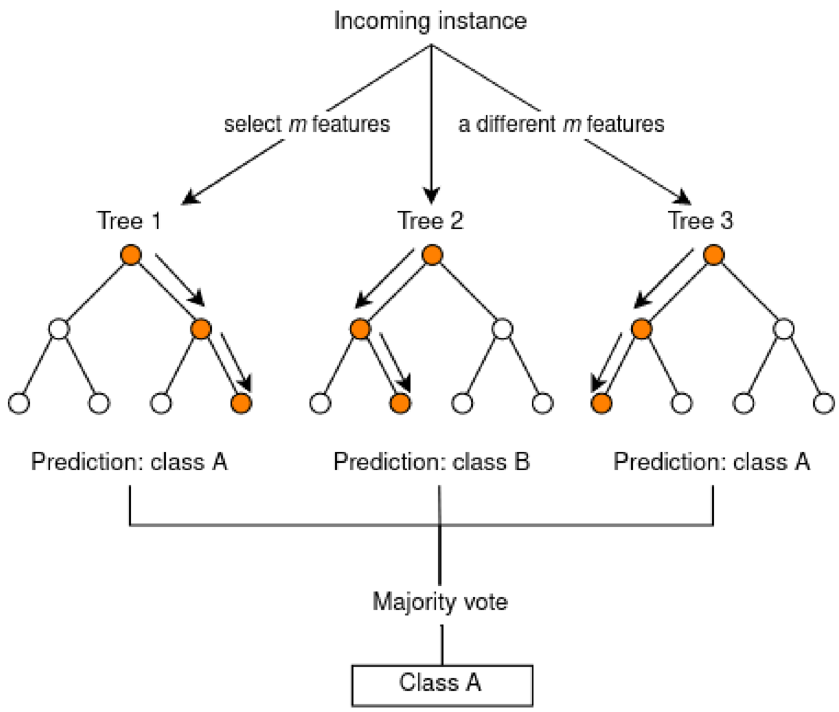


Figure 2.4: Illustration of RF workflow using 3 decision trees [63].

2.2.2.4 Support Vector Machines (SVM)

SVM is one of the most powerful algorithms among traditional ML models, and one of the most used for classification problems of pathological tremor. This is due to their versatility and great performance on binary classification tasks, including linear and non-linear problems [61].

SVM take each instance as a Ω-dimensional vector, when Ω is the number of features. SVM are able to perform multi-class classification, however, binary classification process will be explained as that is the kind of problem in this investigation. Therefore, SVM aim to separate two different groups of vectors (two classes) on a hyperplane defined as:

$$\begin{aligned} \mathbf{wx} - b &= 0, \\ \mathbf{wx} &= \sum_{i=1}^{\Omega} w_i x_i \end{aligned} \tag{2.15}$$

At this point, a hypothesis function $h(x)$ can be defined for making predictions:

$$h(x) = \begin{cases} 0 : \mathbf{wx} - b < 0, \\ 1 : \mathbf{wx} - b \geq 0 \end{cases} \tag{2.16}$$

Then, SVM algorithm faces an optimization task, whose objective is to find the values for **w** and **x** that maximizes the distance between groups. In order to achieve that, SVM come with a set of parameters which can be tuned depending on the problem to solve.

First one is *C*, known as the regularization parameter. It affects the margins' size from the hyperplane. Greater *C* values allow margins to be tighter, so the algorithm adjusts more to data and could suffer from overfitting. Lower *C* values cause the opposite: force margins to be greater, increasing algorithm's ability to generalize and, maybe, underfitting.

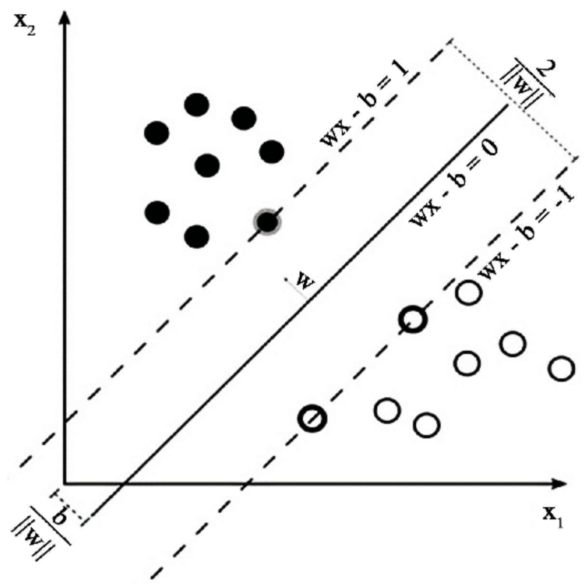


Figure 2.5: Illustration of SVM algorithm. Solid line corresponds to the hyperplane [64].

Another relevant parameter is the *kernel*, which refers to the core function of the SVM and affects the hyperplane shape. There exist various *kernel* functions that can be divided into linear and non-linear. Figure 2.6 shows some effects of changing the *kernel* function. On this investigation, the Radial Basis Function (RBF) is employed, which is defined as:

$$K(x_i, x_j) = \exp(\gamma ||x_i - x_j||^2) \tag{2.17}$$

where $||x_i - x_j||^2$ represents the square euclidean distance between two input instances (x_i, x_j) and γ is an scalar that refers to the influence of each instance in the *kernel*. Lower γ values mean that the RBF will become similar to a linear function.

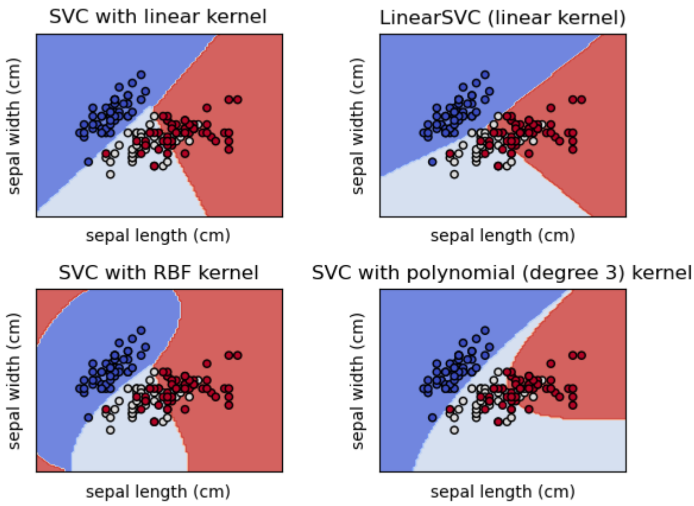


Figure 2.6: Influence of different *kernel* functions on SVM classifiers [58].

2.2.2.5 LSTM classifier

The LSTM based model for classification is shown in Figure 2.7. It is formed by three layers: two LSTM and a final linear layer, which is a perceptron. The inputs for the first LSTM layer are the time sequence values $(i_1, i_2, i_3, \cdots, i_\tau)$. The hidden states from this layer (h_i^1) are sent to the second LSTM layer as inputs. Last hidden state from this layer (h_τ^2) corresponds to the input for the linear layer, as it stores information from all the sequence. Finally, output $(\mathbf{y})$ is generated as predicted tag for input sequence $\mathbf{i}_t$.

The training loop consists on 4 stages per epoch, i.e. each time the model has been fed with the whole training set.

1. Load data and feed the model. The model generates its output.
2. Calculate loss from predictions and actual labels. In order to do that, the model uses a loss function, which in this case is the Binary Cross-Entropy (recommended for binary classification tasks) defined as [40]:
$$H_{BCE} = -\frac{1}{N}\sum_{i=1}^{N} y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \tag{2.18}$$
where y_i is the tag for instance i, $p(y_i)$ the probability of that instance belonging to class y and N the number of instances.
3. BPTT and weights update using Adam optimizer. According to [66], Adam method is ”computationally efficient, has little memory requirement, invariant to diagonal rescaling of gradients, and is well suited for problems that are large in terms of data/parameters”.
4. Validation process. This stage consists on evaluate model performance on the validation set and calculate its loss. Then, if loss is lower than that of the previous model, the current one is saved.

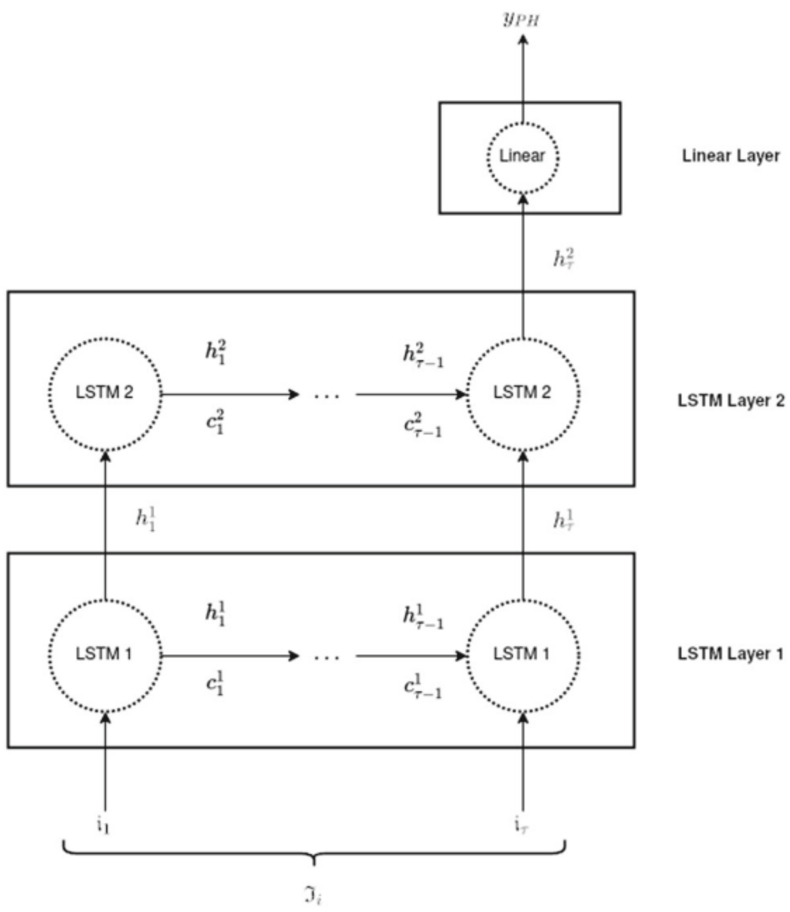


Figure 2.7: LSTM classifier [65].

2.2.2.6 Parameters selection

All algorithms has been tested using different combinations of parameters, looking for those giving better performance. Only the GNB algorithm was evaluated without any changes, as its parameters were related with giving priorities to certain classes and calculation stability [58], so they were not of interest for this study.

In order to perform models evaluation, the selected method was the K-Fold Cross Validation, with K=10. This procedure consists on dividing training data into K different splits of same size, train the algorithm on K-1 folds and evaluate the model using the last subset. This process is repeated for every split: first stage train with $\{S_1, S_2, \cdots, S_{k-1}\}$ and test on S_k, second stage train with $\{S_1, S_2, \cdots, S_{k-2}, S_k\}$ and test on S_{k-1}, etc. The result of this procedure is the model that performed the best on the test stage, which can be considered the best possible model that can be achieve using the training set and K different splits. Nonetheless, K-Fold Cross Validation was only employed for traditional ML algorithms, i.e. GNB, KNN, RF and SVM. The LSTM classifier has its own training algorithm, as explained in Section 2.2.2.5. Then, parameters selection for each algorithm was:

- **KNN**. Data normalization algorithm: *StandardScaler*.
  - Number of neighbors: {2}. For filtered EMG datasets: {4,5,6}.
  - Weights: {Distance}.
  - Algorithm: {Ball Tree}.
  - Leaf size (for Ball Tree): {30, 35}.
  - Metric: {Euclidean, Chebyshev, Minkowsky}.
- **RF**. Data normalization algorithm: *MinMaxScaler*.
  - Trees: {85, 90 ,95, 100}.
  - Criterion (function to measure split quality): {Entropy}.
  - Max. Features (when looking for the best split): $\{\log_2(N), \sqrt{N}\}$, where N is the number of data features.
  - Min. Samples per leaf: {2}.
- **SVM**. Data normalization algorithm: *StandardScaler*.
  - C: {0.1, 1, 10}.
  - γ: {1, 0.1, 0.01}.
  - Kernel: {RBF}.
- **LSTM classifier**. Data normalization algorithm: *StandardScaler*.
  - Learning rate: {0.005, 0.001, 0.0001}.
  - Hidden size: {20, 35, 50}.

2.2.2.7 Performance metrics

The evaluation of classification algorithms performance is conducted by four different metrics: precision, recall, f1-score and accuracy. These are calculated by using the confusion matrix, in which distribution of real and predicted instances are visually displayed [40]. Confusion matrix is based on the definition of four types of instances:

- **True Positive (TP)**: signals tagged as Tremor and classified as Tremor as well.
- **False Positive (FP)**: signals classified as Tremor but tagged as No-Tremor.
- **True Negative (TN)**: signals tagged and correctly classified as No-Tremor.
- **False Negative (FN)**: signals tagged as Tremor but classified as No-Tremor.

		Predicted Class	
		No-Tremor	Tremor
Real Class	No-Tremor	TN	FP
	Tremor	FN	TP

Table 2.4: Confusion Matrix Template.

Then, precision is the rate between signals correctly classified as Tremor and the total of signals classified as Tremor (see Equation 2.19). It is a measure of how good is the algorithm identifying positives as actually positives [40].

$$Precision = \frac{TP}{TP + FP} \tag{2.19}$$

Recall is similar to precision, even they usually appear together as can be considered complementary metrics. It represents the rate between signals correctly classified as Tremor and all signals tagged as Tremor, including those incorrectly classified as No-Tremor (FN)(see Equation 2.20). Recall simply represents how many positives the algorithm does not identify [40].

$$Recall = \frac{TP}{TP + FN} \tag{2.20}$$

F1-score is the harmonic mean between precision and recall. This metric is used to ensure that classifiers have good precision and recall at the same time, balancing the values of those metrics equally (see Equation 2.21).

$$F_1 = 2\frac{Precision \cdot Recall}{Precision + Recall} = \frac{TP}{TP + \frac{1}{2}(FP + FN)} \tag{2.21}$$

Finally, accuracy is a very useful metric to preview algorithm's performance, even is a good practice to combine this measure with others, such as precision and recall. It provides the rate of all correctly classified instances over the test set [40] (see Equation 2.22).

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{2.22}$$

2.2.3 Prediction models

Two models were built to perform prediction tasks. They consists on various layers, as well as the model employed for classification (see Section 2.2.2.5):

- **Model 1**: one LSTM layer, one linear layer and the sigmoid function applied right before generating the output (see Figure 2.8).
- **Model 2**: two LSTM layers, one linear layer and the sigmoid function (see Figure 2.9).

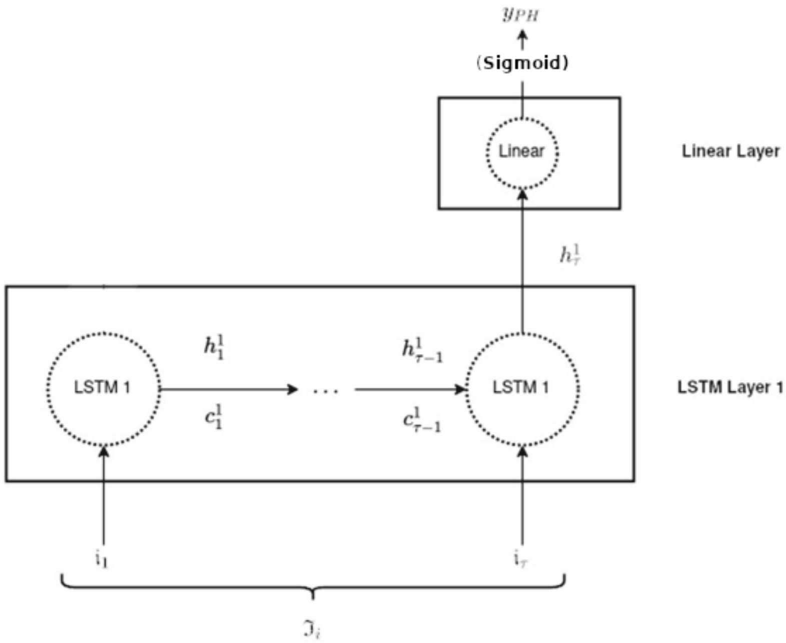


Figure 2.8: Model 1 for prediction.

The main difference between these models and the LSTM classifier is that the last LSTM layer produces a δ-dimension output, where δ corresponds to the number of predicted samples. Then, this values are passed through the sigmoid function. Regarding to the training loop, the process is still the same as explained for the LSTM classifier (see Section 2.2.2.5). However, the loss function selected for evaluating the prediction model is now the Mean Squared Error:

$$MSE = \left(\frac{1}{N}\right) \sum_{i=1}^{N} (i_i - y_i)^2 \tag{2.23}$$

where N is the number of predictions made, i_i the input value and y_i the predicted value.

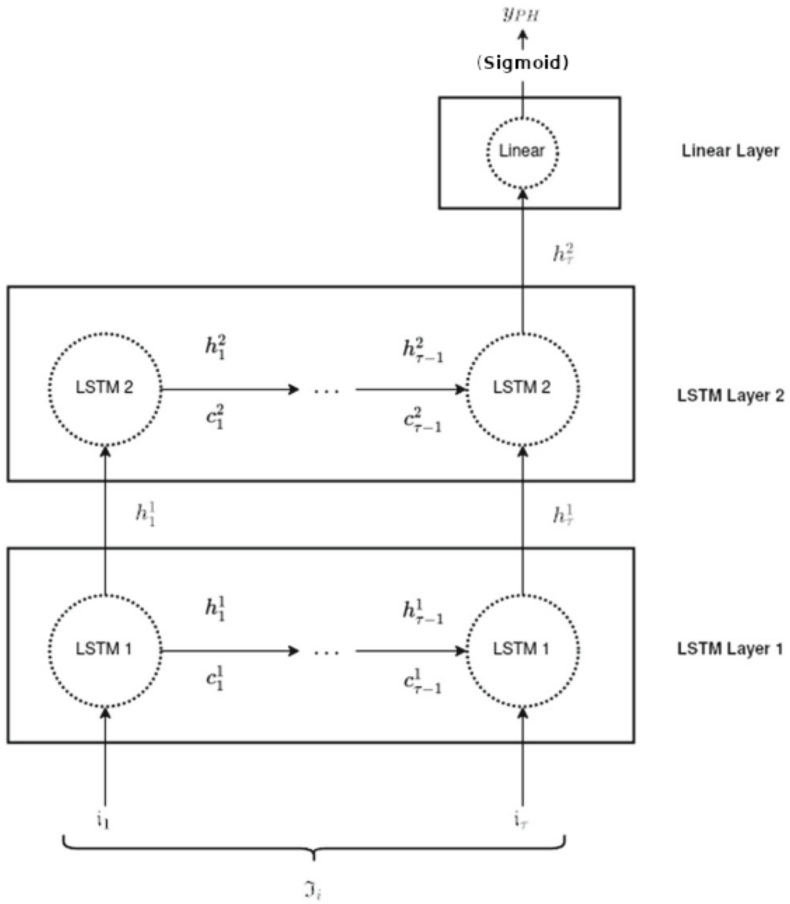


Figure 2.9: Model 2 for prediction.

2.2.3.1 Parameters selection

Prediction models were evaluated using a wide selection of different parameters:

- **Data normalization algorithm**: *MinMaxScaler*.
- **Train samples**: $\{20, 30, 40, 50\}$.
- **Prediction horizon (in number of samples)**: $\{5, 10, 20, 30, 50\}$.
- **Learning rate**: $\{0.001, 0.0005, 0.0001\}$.
- **Hidden size**: $\{20, 35, 50\}$.

Given all possible combinations between these parameters, the number of total trained models ascends to 360.

2.2.3.2 Performance metrics

In order to evaluate the prediction models, new metrics should be defined as to work with time series data. These metrics are the Mean Squared Error (see

Equation 2.23), the Root-Mean-Square Error (see Equation 2.24) and Pearson's Correlation Coefficient (see Equation 2.25).

$$RMSE = \sqrt{\left(\frac{1}{N}\right)\sum_{i=1}^{N}(i_i - y_i)^2} \tag{2.24}$$

$$PCC = r_{i,y} = \frac{\sum_{i=1}^{N}(i_i - \overline{i_i})(y_i - \overline{y_i})}{\sqrt{\sum_{i=1}^{N}(i_i - \overline{i_i})^2}\sqrt{\sum_{i=1}^{N}(y_i - \overline{y_i})^2}} \tag{2.25}$$

where N is the number of predictions made, i_i the input value and y_i the predicted value.

The Root-Mean-Square Error value provides information about the prediction accuracy of the model, calculating differences between real and predicted values. On the other hand, the Pearson's Correlation Coefficient represents the similarity between real and predicted signals, as values of this coefficient closer to 1 mean a higher linear relationship between them.

Chapter 3

Results

This chapter presents the classification and prediction results for each ML and DL model, as well as comparisons between them. The results include: performance metrics (see Sections 2.2.2.7 and 2.2.3.2), models' performance using different parameters and window lengths, and employing raw and filtered signals. Prediction results also include some examples of predicted signals using the defined prediction horizons (see Section 2.2.3.1).

3.1 Classification results

3.1.1 GNB

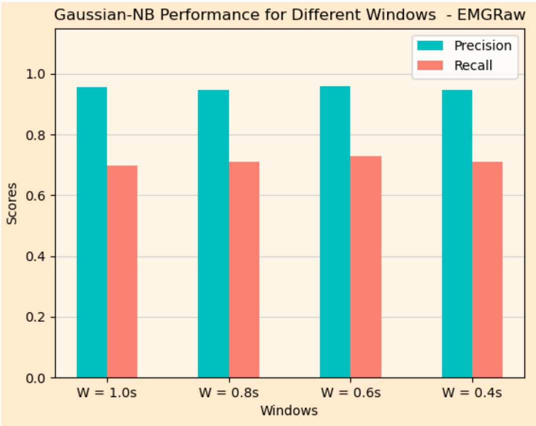
No changes were made to GNB parameters as explained in Section 2.2.2.6). Therefore, we proceed to present the results from those GNB models that performed the best after conducting a K-Fold Cross Validation process.

Table 3.1 shows the precision and recall obtained for different window lengths and using raw and filtered signals. Precision is always better than recall, while former's minimum value is 0.60 when using filtered EMG and 0.8 s window length. Nonetheless, it can be seen that, for most of the cases, shorter window lengths and usage of filtered signals suppose an improvement. This classifier achieved its best results with 0.4 s and filtered EMG: 0.9708 of precision and 0.7502 of recall.

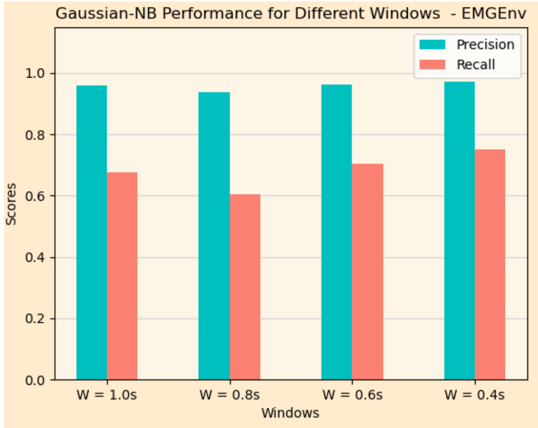
	Window							
	1.0 s		**0.8 s**		**0.6 s**		**0.4 s**	
	R	**F**	**R**	**F**	**R**	**F**	**R**	**F**
Precision	0.9556	0.9597	0.9472	0.9364	0.9587	0.9630	0.9468	0.9708
Recall	0.6965	0.6769	0.7108	0.6048	0.7274	0.7025	0.7089	0.7502

Table 3.1: GNB classification performances. R: Raw; F: Filtered.

GNB performance evolution through different windows is visually presented in Figure 3.1. Also, all confusion matrices can be found at Appendix E.1.



(a) Raw EMG results.



(b) Filtered EMG results.

Figure 3.1: GNB performance per window.

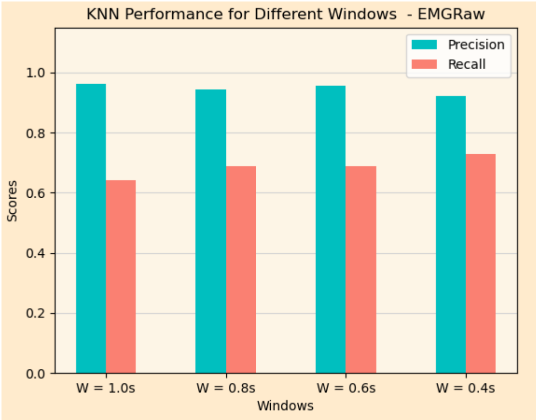
3.1.2 KNN

K-Fold Cross Validation for KNN was applied for six models, corresponding to those for raw EMG, and twelve models for filtered EMG. Tables D.2 and D.1 in Appendix D.1 show all models and their parameters.

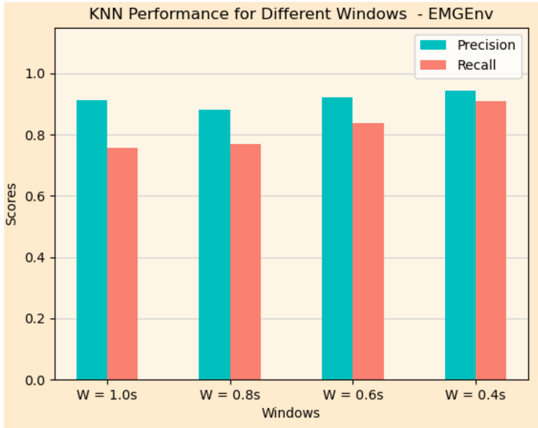
After cross validation assessment, best KNN models (see Appendix D.1) underwent the test stage. It should be noted that some models gave the same results for equal conditions (see Appendix F.1), so only one of them was finally selected, as for simplifying results. Table 3.2 shows the metrics obtained using the best KNN model per window and signal type and Figure 3.2 shows performance evolution through windows. First of all, from cross validation process we conclude that most influential parameter is the number of neighbors. After testing the models, all precision values are over 0.9 for all cases, while not making any relevant improvement with shorter windows and being greater with raw EMG signals, except for 0.4 window. On the other hand, recall is improving as window length decreases and is greater when using filtered signals. Best results came from using 0.4 s window length and filtered data: 0.94 of precision, 0.91 of recall. Confusion matrices can be seen in Appendix E.2.

	Window							
	1.0 s		**0.8 s**		**0.6 s**		**0.4 s**	
	R	**F**	**R**	**F**	**R**	**F**	**R**	**F**
Precision	0.9629	0.9123	0.9433	0.8826	0.9554	0.9211	0.9221	0.9441
Recall	0.6402	0.7583	0.6898	0.7680	0.6887	0.8384	0.7295	0.9106

Table 3.2: KNN classification performances. R: Raw; F: Filtered.



(a) Raw EMG results.



(b) Filtered EMG results.

Figure 3.2: KNN performance per window.

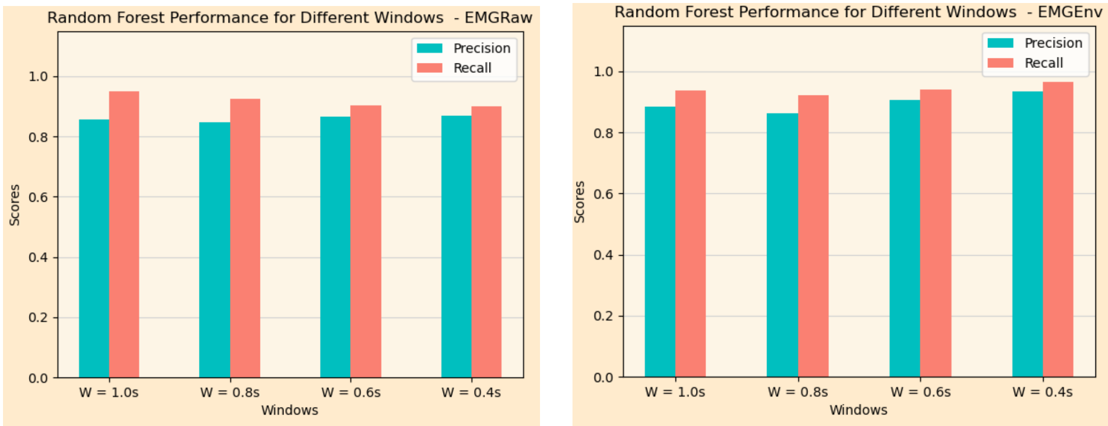
3.1.3 RF

Regarding to RF algorithm, K-Fold Cross Validation was applied for a set of eight models, which, this time, are the same for raw and filtered EMG. Table D.4 in Appendix D.2 shows all models and their parameters. All boxplots from cross validation are available in Appendix F.2. Variety in best models selection (see Appendix D.2) is greater in this case, which means that choosing appropriate values of parameters *Max. Features* and *Trees* is crucial, in order to achieve good performance results.

	Window							
	1.0 s		**0.8 s**		**0.6 s**		**0.4 s**	
	R	**F**	**R**	**F**	**R**	**F**	**R**	**F**
Precision	0.8560	0.8853	0.8503	0.8636	0.8667	0.9068	0.8698	0.9324
Recall	0.9568	0.9302	0.9356	0.9263	0.9043	0.9451	0.8976	0.9636

Table 3.3: RF classification performances. R: Raw; F: Filtered.

As shown in Table 3.3, RF performs quite well on classifying either raw and filtered signals. Unlike GNB and KNN, this algorithm finally achieves values over 0.85 for both metrics, for every window and signal type. Precision values increase as shorter is the window length. For its part, recall decreases for raw signals and increases for filtered ones (see Figure 3.3). One more time, best results were reached using 0.4 s window length and filtered signals: 0.93 of precision, 0.96 of recall. Confusion matrices can be seen in Appendix E.3.



(a) Raw EMG results. *(b) Filtered EMG results.*

Figure 3.3: RF performance per window.

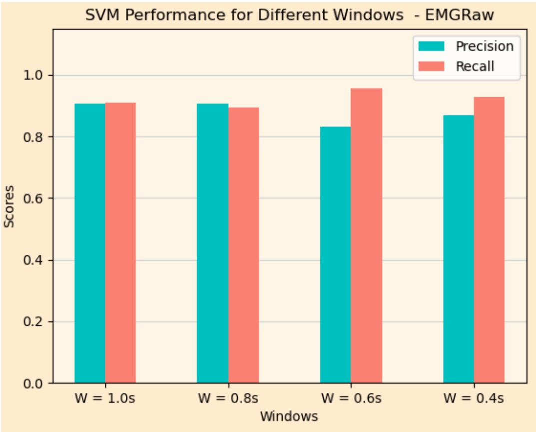
3.1.4 SVM

Nine SVM models were assessed by cross validation (see Appendix D.3) for both raw and filtered datasets. All boxplots related to this process are available in Appendix F.3. Models giving best performances have C values 1 and 10 (see Appendix D.3), which means that SVM benefits from having tighter margins. This is probably caused by high variability in data, so the algorithm needs for smaller margins in order to separate instances into classes.

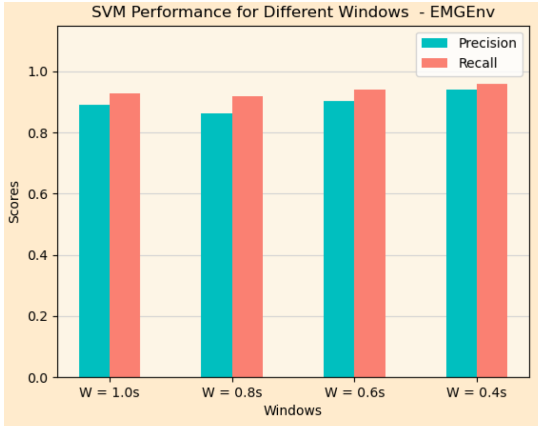
In regards to the results, they are very promising. As shown in Table 3.4 and Figure 3.4, metrics values are over 0.8 in all cases. Working with raw signals, precision and recall become more imbalanced as reducing window size. On the other hand, we can notice an increasing trend for both metrics, when using filtered data, as window size increases. Best results are reached for 0.4 s of window length and filtered EMG: 0.94 of precision and 0.96 of recall. Confusion matrices can be seen in Appendix E.4.

	Window							
	1.0 s		**0.8 s**		**0.6 s**		**0.4 s**	
	R	**F**	**R**	**F**	**R**	**F**	**R**	**F**
Precision	0.9071	0.8903	0.9047	0.8631	0.8318	0.9038	0.8690	0.9412
Recall	0.9099	0.9279	0.8935	0.9184	0.9565	0.9416	0.9270	0.9597

Table 3.4: SVM classification performances. R: Raw; F: Filtered.



(a) Raw EMG results.



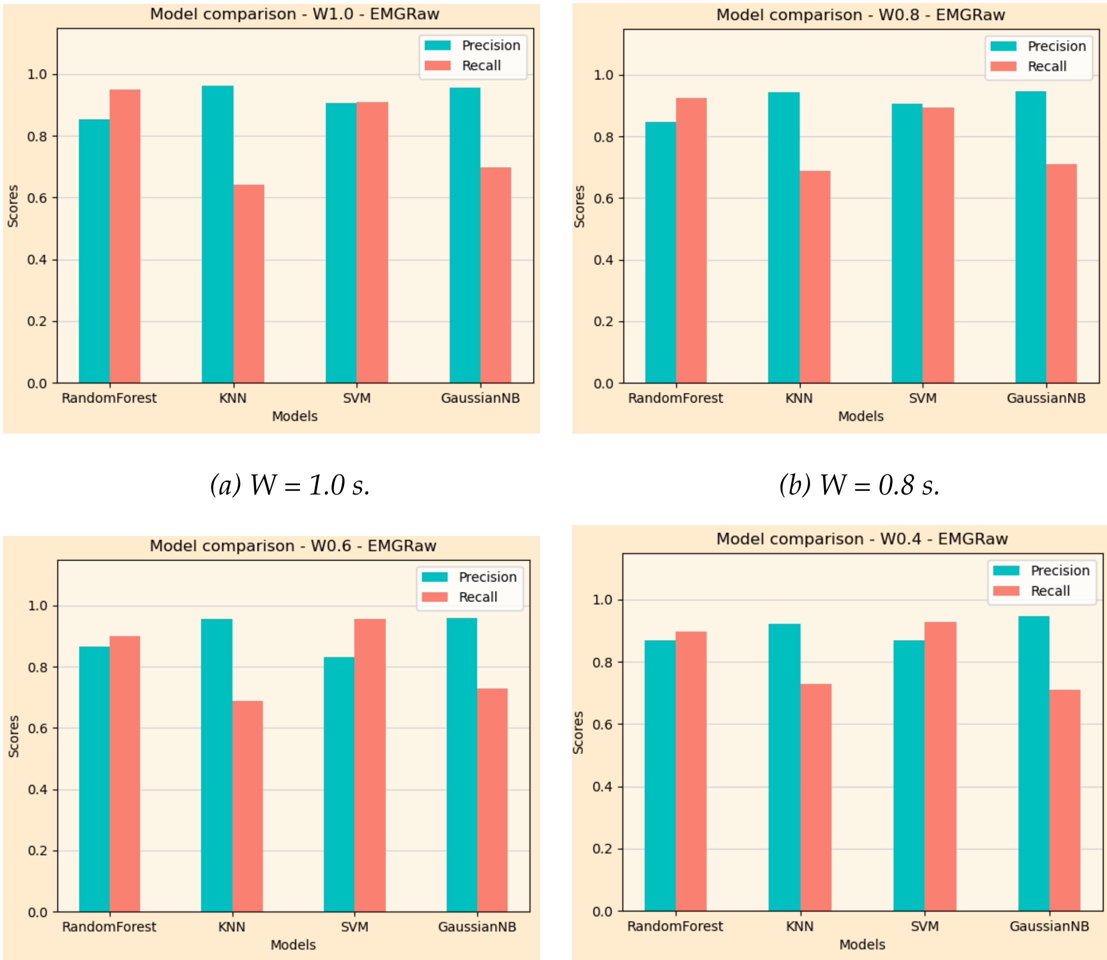
(b) Filtered EMG results.

Figure 3.4: SVM performance per window.

3.1.5 Best models comparison

We proceed to make a deeper comparison between all assessed ML models, taking into account how different window sizes affect, as well as the usage of raw and filtered signals. The main objective is to identify which models perform the best for every situation and how prominent are the differences.

Figure 3.5 shows the comparison between models using raw data for all window sizes. RF and GNB seem to perform quite the same, independently from the window length, while making tiny improvements as it becomes shorter. However, RF metrics are more balanced that those from GNB. SVM benefits from longer windows, regarding to similar precision and recall values, but still achieves its best results with 0.4 s size. GNB and KNN are the algorithms giving most imbalanced results: precision is always high but recall never takes values over 0.8. Then, this models have difficulties on identifying tremor signals as they bring many false negatives, which supposes an important problem if they are implemented in tremor handling devices. Therefore, best algorithms for working with raw signals situations are RF and SVM, while the former gives better results.



(a) W = 1.0 s.

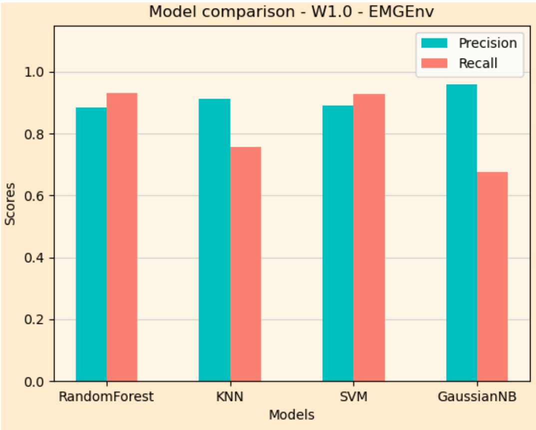
(b) W = 0.8 s.

(c) W = 0.6 s.

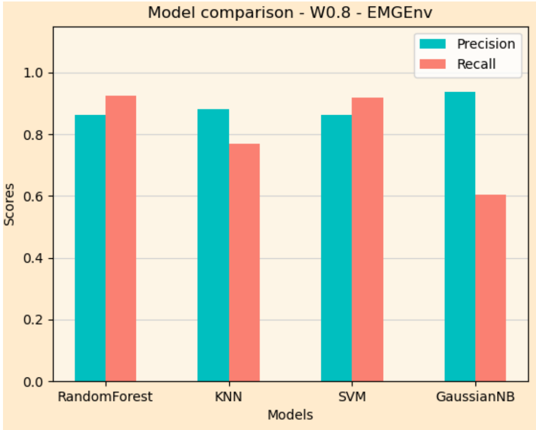
(d) W = 0.4 s.

Figure 3.5: Best models comparison. Raw EMG.

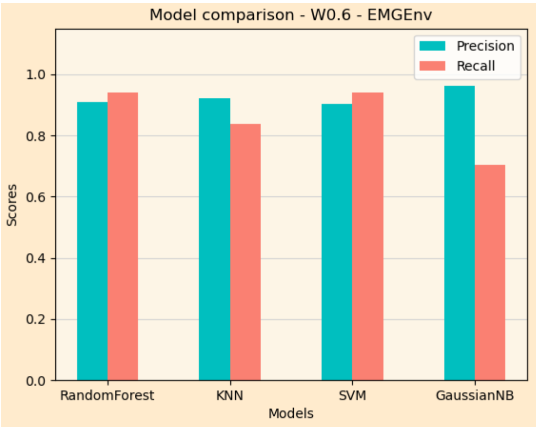
On the other hand, Figure 3.6 shows the comparison between models using filtered data. RF and SVM are still the algorithm that give better results overall. They behave similar to when using raw signals, while metrics values increases with filtered data, in most cases. The biggest difference is for KNN and GNB models, which now reach more balanced precision and recall values, specially for KNN. GNB is still far from being able to reduce the number of false negatives identified. For its part, KNN is now an appropriate algorithm for facing classification task.



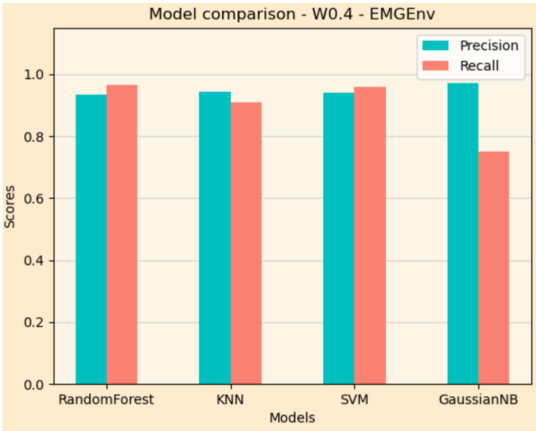
(a) *W = 1.0 s.*



(b) *W = 0.8 s.*



(c) *W = 0.6 s.*

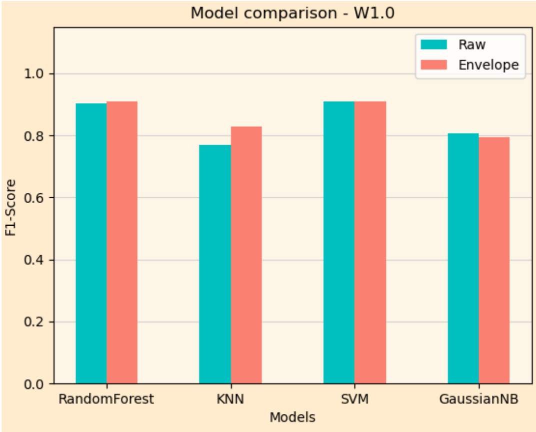


(d) *W = 0.4 s.*

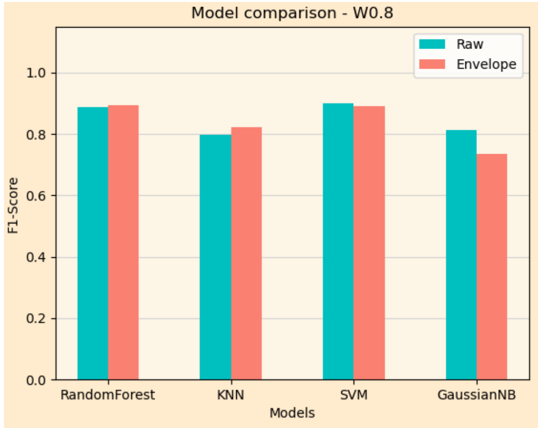
Figure 3.6: Best models comparison. Filtered EMG.

Finally, Figure 3.7 compares models' performances when using raw and filtered signals, for all window sizes. We can see models start to give better results for filtered signals as window size decreases. When using longest windows (i.e. 1.0 s and 0.8 s) models seem to perform the same, except from KNN which, as explained before, experiments greater improvements when working with filtered signals, and since 1.0 s window length performs better. Bigger differences start to appear for 0.6 s and 0.4 s windows, where f_1 scores rise gradually.

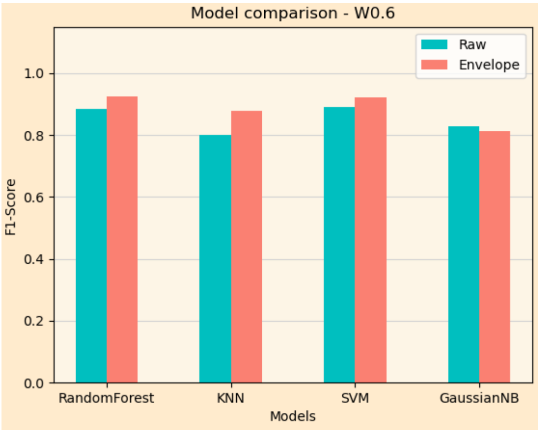
After evaluating the results from traditional ML models, a LSTM based classifier is going to be assessed in classification task, looking for better or, at least, equal results than those from models already presented.



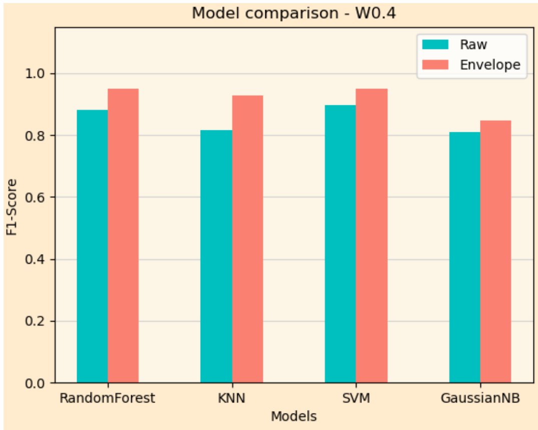
(a) *W = 1.0 s.*



(b) *W = 0.8 s.*



(c) *W = 0.6 s.*



(d) *W = 0.4 s.*

Figure 3.7: Best models comparison. Raw EMG vs Filtered EMG.

3.1.6 LSTM classifier

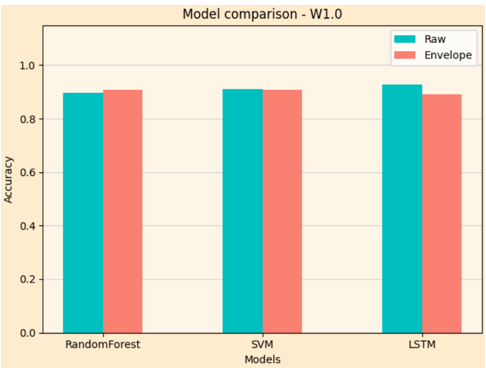
As explained in Section 2.2.2.6, six different parameters (three learning rates, three hidden sizes) were selected for building LSTM classifiers. That gives a total of nine different models to be evaluated in classification task, searching for best parameters combination per window (see Appendix D.4). After training process, those models giving the best performances (see Appendix D.4) were conducted to the test stage. It should be noted that all models passed through validation process during their train stage, in order to ensure their improvement.

Almost all proposed models were selected as best in performance for at least one situation, which means that influence of learning rate and hidden size values is decisive, so as to reach better results. Nonetheless, other parameters, such as the optimizer (*Adam*) and the loss function (*Binary Cross Entropy*), have high importance for training. It could be interesting to train this LSTM models using also different combinations of optimizers (i.e. Stochastic Gradient Descent) and loss functions.

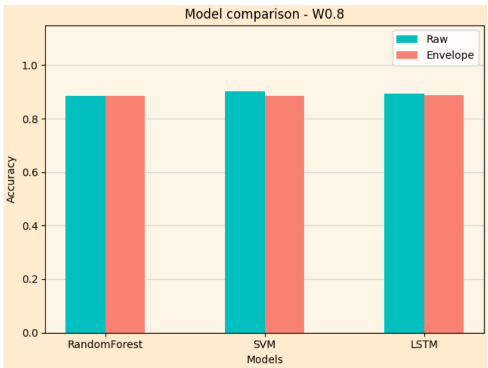
The results of LSTM classifiers are promising, with accuracies over 0.88 for all cases. Those windows where LSTM became more unstable and reaching convergency was more difficult are 0.8 s and 0.6 s. At the same time, convergency was faster and smoother when using filtered EMG signals. Train and valid loss functions become more distant as training progresses, in most cases (see Appendix G). Meanwhile, using filtered data, those functions are closer. The conclusion is that raw signals, being noisier than filtered ones, make training process more demanding for LSTM. Besides, models still perform well in classification. Best result come, as happened with the other classifiers, from using 0.4 s window size and filtered signals: 0.94 of accuracy.

	Window							
	1.0 s		**0.8 s**		**0.6 s**		**0.4 s**	
	R	**F**	**R**	**F**	**R**	**F**	**R**	**F**
Accuracy	0.9265	0.8909	0.8933	0.8891	0.9117	0.9100	0.9306	0.9352

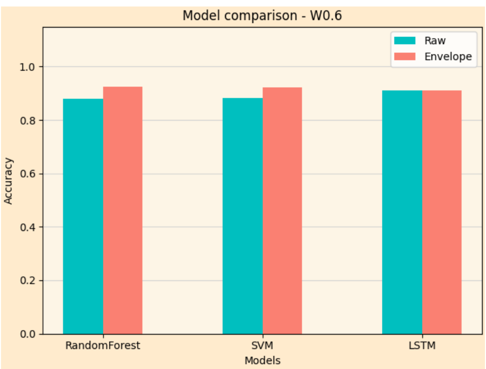
Table 3.5: LSTM classifiers performances. R: Raw; F: Filtered.



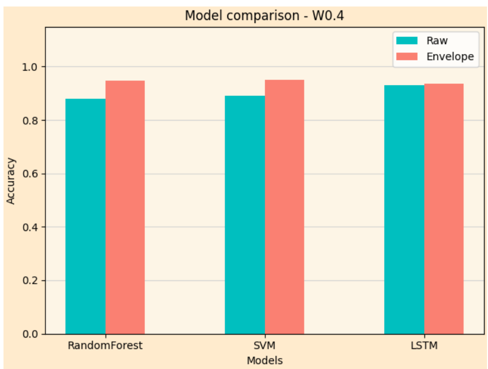
(a) W = 1.0 s.



(b) W = 0.8 s.



(c) W = 0.6 s.



(d) W = 0.4 s.

Figure 3.8: Comparison between best ML algorithms and LSTM classifier.

Finally, Figure 3.8 shows a comparison between RF and SVM, which are the algorithms that performed the best among traditional ML ones, and the LSTM. This time, the metric selected was the accuracy, as it was the computed metric for LSTM classifier. Results are quite similar among the compared models. As already explained, SVM and RF experiment some improvement when using filtered signals, which, in the case of LSTM, is barely imperceptible, except for 1.0 s window. Regarding to the influence of window length, there are some aspects to highlight. First, if we look at performances when using raw signals, the LSTM classifier keeps accuracies high, always similar to those obtained for filtered data, while the differences seen for SVM and RF are more noticeable. Also, we can observe that, from 0.6 s window length, SVM and RF start to achieve higher accuracies than LSTM, using filtered signals.

3.2 Prediction results

Table 3.6 shows the results for the best models for each combination of prediction horizon and input sample length, after training stage. Also, some examples of predicted signals are shown in Figures 3.9, 3.10 and 3.11.

Best LSTM models for prediction

S_t	P_h	N_l	H_s	L_r	**PCC**	**MSE (u^2)**	**RMSE (u)**
20	5	3	20	0.001	0.9030	0.0035	0.0508
20	10	3	20	0.0001	0.7179	0.0135	0.1014
20	20	3	20	0.001	0.5199	0.0230	0.1383
20	30	3	20	0.0005	0.4200	0.0263	0.1515
20	50	3	35	0.0001	0.3228	0.0290	0.1627
30	5	3	20	0.001	0.9023	0.0028	0.0448
30	10	3	20	0.0001	0.7233	0.0130	0.0976
30	20	3	20	0.0001	0.5156	0.0228	0.1373
30	30	3	20	0.001	0.4137	0.0263	0.1513
30	50	2	20	0.0005	0.3154	0.0292	0.1633
40	5	3	20	0.0001	0.9043	0.0028	0.0455
40	10	2	20	0.0005	0.7245	0.0130	0.0100
40	20	3	35	0.0001	0.5201	0.0225	0.1369
40	30	3	20	0.0001	0.4250	0.0259	0.1500
40	50	3	20	0.0001	0.3204	0.0294	0.1635
50	5	2	20	0.0001	0.9083	0.0026	0.0436
50	10	3	20	0.0005	0.7597	0.0115	0.0927
50	20	3	35	0.0001	0.5413	0.0218	0.1338
50	30	3	20	0.0001	0.4386	0.0262	0.1501
50	50	3	20	0.0001	0.3489	0.0296	0.1637

Table 3.6: LSTM best models' parameters and performances for prediction. S_t: training samples; P_h: prediction horizon; N_l: number of layers; H_s: hidden size; L_r: learning rate; PCC: Pearson Correlation Coefficient; MSE: Mean Squared Error; RMSE: Root-Mean-Square Error.

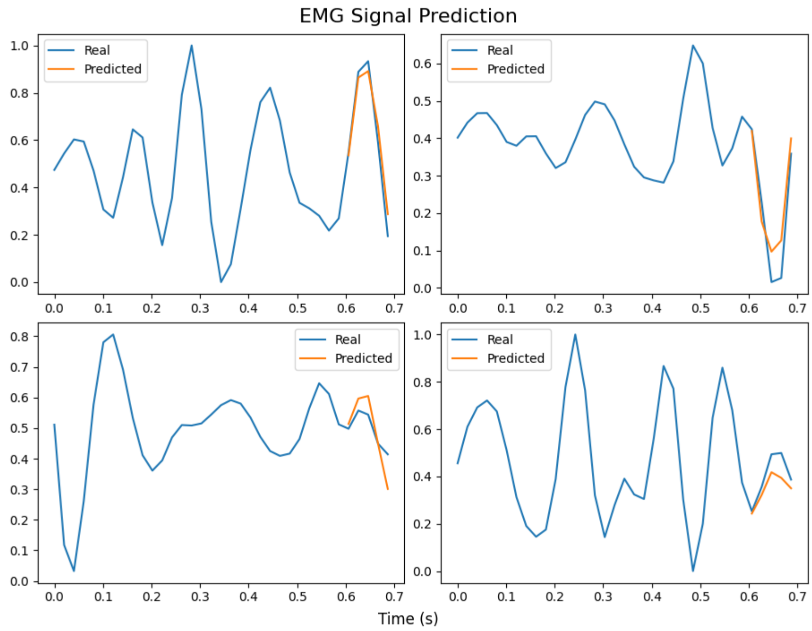


Figure 3.9: Examples of predicted EMG signals. Train samples: 30. Prediction Horizon: 5.

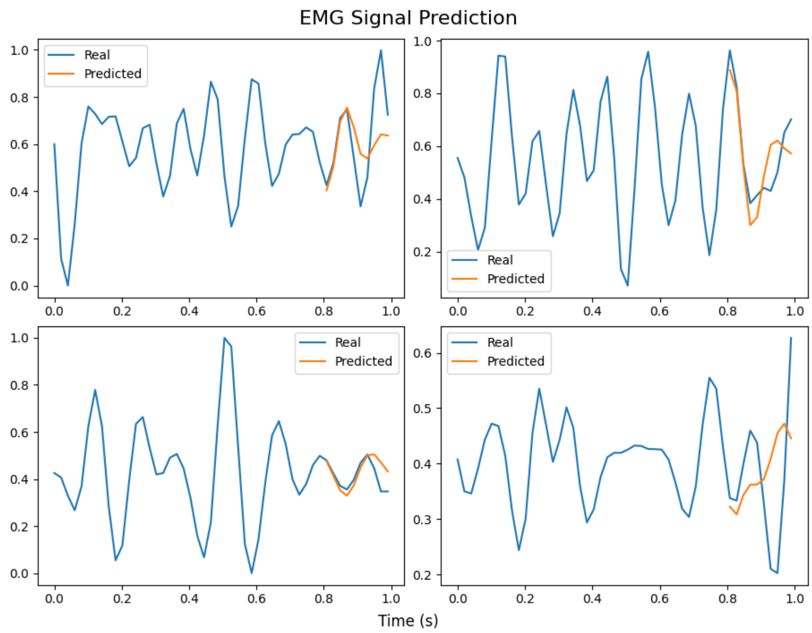


Figure 3.10: Examples of predicted EMG signals. Train samples: 40. Prediction Horizon: 10.

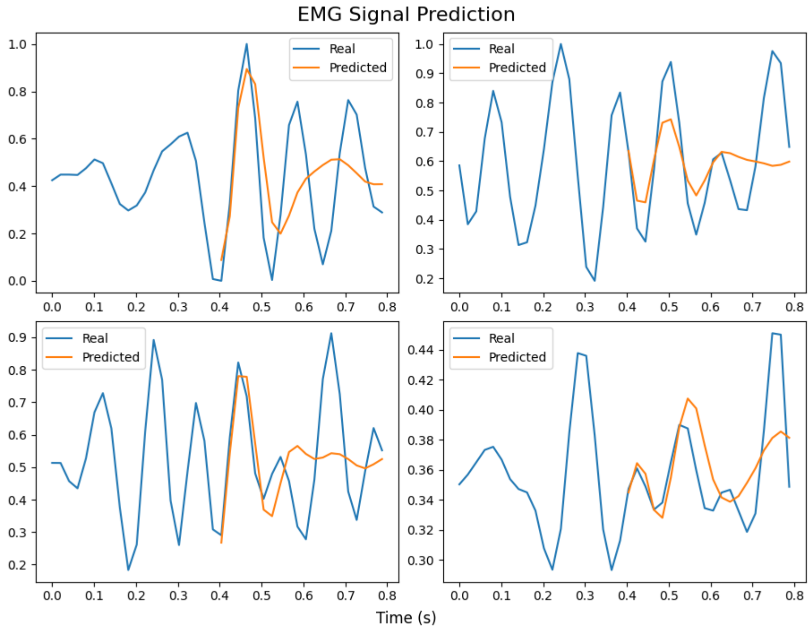


Figure 3.11: Examples of predicted EMG signals. Train samples: 20. Prediction Horizon: 20.

Results are quite similar among all models. They reach their best correlation coefficients when with shortest prediction horizon (5 samples). Then, correlation between predicted and real signals is high, always over 0.9 in all cases. We can see a very tiny improvement as training window increases to 50 samples, which corresponds to a 1 s sequence. However, improvements are almost irrelevant. Looking for longer prediction distances, models still perform well for 10 future samples, with correlation values around 0.75. These are acceptable performances, knowing that models are working with few pre-processed signals. The examples above show that, even when EMG sequences are filtered, they have still poor quality, so it is expected that predictions for long windows are going to decrease in accuracy. For prediction horizons of 20, 30 and 50 samples, correlations decline to values around 0.5, 0.4 and 0.3, respectively. Those are not sufficient at all for even trying to identify signal and tremor trends in future windows.

Chapter 4

Conclusions and future approaches

4.1 Discussion

The results for classification of EMG signals are quite promising. After comparing all assessed models, we conclude that RF and SVM provide good results, in terms of precision and recall. Also, those metrics increase when using filtered signals, while they are still acceptable with raw data. The combination between using filtered sequences and short windows leads to best results: over 0.9 for precision and recall values. Apart from traditional algorithms, LSTM classifiers also perform with accuracy values over 0.9. However, their results are not improving those from traditional ML models, even the SVM and RF gave better results in some cases. Therefore, there is no reason, regarding to raw performance metrics, to employ LSTM models for classification task in substitution of those traditional models. SVM and RF are sufficiently powerful for identifying tremor in EMG signals from the datasets employed in this investigation and under the same conditions.

On the other hand, prediction task demands for more investigation to be made. The LSTM based models showed that they are able to predict EMG tremorgenic signals for a few future samples. In those cases, when they had to predict next 5 samples, the results are more than acceptable with correlations over 0.9, even when training with a 400 ms window length. In addition, results showed that performances do not increase substantially when augmenting training samples. Nonetheless, results are still not sufficient for implementing these algorithms in tremor handling devices, as tremor band frequency is between 4 and 12 Hz, which means that models must be able to predict the next 250 ms with good accuracy. In this respect, we have seen how correlations are getting worse as prediction horizon increases. For a prediction window of only 10 samples (200 ms), correlations between real and predicted sequences reach values around 0.7. At this point, if we continue enlarging the horizon, we obtain values under 0.5, which are not yet acceptable, in order to identify future signals trends. It was expected that models would perform worse when augmenting the number of samples to predict. Nonetheless, this results might be influenced by different aspects related to the datasets employed. First of all, it was explained that EMG signals were selected regarding to its quality. However, those sequences might be too much noisy and present artifacts, all contributing to make prediction task more demanding for models. Also, it should not be forgotten that EMG signals come from a variety of ET patients, with different tremor grades, so this adds another difficulty in recognizing patterns, which, in addition, might be specific for each subject. This situation could have led to relatively high performances for the first few samples and, then, to models losing their ability to identify next steps. Fortunately, in recent years, new DL models, more powerful than LSTM based ones working with time series data, were developed, such as transformers [67]. There are still many experiments to do, in order to achieve models capable of predicting EMG signals and, consequently, tremor activity.

4.2 Conclusions

Comparing these results with those from previous studies (see Section 1.1.4), we can conclude that classification of EMG signals into Tremor and No-Tremor ones is actually possible, using either traditional ML algorithms or DL models, as it was when using kinematic signals. Furthermore, most algorithms perform good even for raw and short EMG sequences, while the SVM and RF give results over 0.9 of precision and recall in those situations. However, as classification problem can be considered as solved, the next step, prediction task, needs for better solutions. The LSTM architecture presented in this thesis, while being simpler than models from other studies based on combinations of MLP and LSTM [52], reaches correlations around 0.9 for the next 100 ms window, even when using a 400 ms EMG window for training. For a prediction horizon of 10 samples (200 ms), correlations decrease to around 0.7. Then, EMG prediction for identify tremor periods is possible with less sophisticated models, but the problem of enlarging prediction windows while preserving acceptable correlations remains unresolved.

One of the main hypothesis was that EMG sequences could be tagged by means of frequency components between 4 and 10 Hz, which is the typical band for tremor. Results showed that it is possible using, in this case, an estimation of the PSD, even when this approach assumes EMG to be stationary. Nonetheless, classifiers did not seem to suffer from a bad tagging process. Another important hypothesis was that algorithms were supposed to perform worse when classifying raw sequences, in comparison when using filtered signals. It was demonstrated that, indeed, classifiers improve their performance when they are fed with filtered sequences. Some of them, such as KNN or GNB, experiment higher improvements than others, increasing their values of recall by 0.1 and 0.2, in some cases. In RF and SVM models, we observe lower improvements, with differences between metrics obtained using raw and filtered signals of around 0.05. In short, all algorithms brought results, in terms of precision and recall values, over 0.7 in all cases, which are acceptable. Finally, it was proposed that correlations values in prediction task would decrease as the number of future samples enlarges. This is also demonstrated by the results obtained. However, for the shortest prediction windows, correlations were above

0.9, which means that predicting EMG signals is a feasible method for detecting tremor activity.

In regards to the objectives of this thesis, the design, implementation and assessment of ML and DL algorithms in classification and prediction of EMG signals were successfully accomplished. Recent techniques, such as electrical stimulation of afferent pathways, can benefit from this investigation, in terms of including the usage of EMG signals in their tremor suppression strategies. The results would help on further investigations aiming to implement these kind of technology in portable suppression devices, which, hopefully, would suppose a huge advance in tremor handling approaches.

4.3 Future approaches

Further investigations in this field should focus on various improvements:

- This study was conducted using data extracted from only ET patients. In future experiments, it would be desirable to include information from patients who suffer tremor symptoms, but caused by other motor diseases, such as PD. This can lead to a general method to handle pathological tremor, widely applicable to patients who suffer from it.
- Tagging process needs for a more consistent and accurate solution, in order to avoid variability and standardize tremor thresholds. Further studies should aim to evaluate what are the most important and useful signal features which allow to consistently tag EMG tremor sequences.
- Of course, more investigation is needed for prediction models. In future experiments, usage of new DL architectures, such as transformers, and other powerful models in time series prediction should be assessed and, hopefully, could bring higher performances for longer windows. Nonetheless, it should not be forgotten that the final goal is to implement these algorithm in portable, even affordable, devices which have limited computational power.
- Last but not least, it might be interesting to combine kinematic and physiological signals, in order to determine if models benefit from having these information given at the same time. We already now that DL architectures such as LSTM networks can be fed with multi-feature time series data, that is, with more than one feature per time step. This behavior brings the opportunity to employ not only one type of signal, but a combination of various sources of tremorgenic activity information.

References

[1] R Bhidayasiri. Differential diagnosis of common tremor syndromes. *Postgrad. Med. J.*, 81(962):756–762, 2005.

[2] Involuntary movements: Types, causes, and examples, stanford 25. https://stanfordmedicine25.stanford.edu/the25/involuntary-movements-and-tremors.html. Accessed: 2022-6-9.

[3] Paul Crawford and Ethan E Zimmerman. Tremor: Sorting through the differential diagnosis. *Am. Fam. Physician*, 97(3):180–186, 2018.

[4] Nitish Kamble and Pramod Kumar Pal. Tremor syndromes: A review. *Neurol. India*, 66(Supplement):S36–S47, 2018.

[5] Rick C Helmich, Ivan Toni, Günther Deuschl, and Bastiaan R Bloem. The pathophysiology of essential tremor and parkinson's tremor. *Curr. Neurol. Neurosci. Rep.*, 13(9):378, 2013.

[6] E D Louis. Clinical practice. essential tremor. *N. Engl. J. Med.*, 345(12):887–891, 2001.

[7] Lorraine N Clark and Elan D Louis. Essential tremor. *Handb. Clin. Neurol.*, 147:229–239, 2018.

[8] Eliezer J Sternberg, Roy N Alcalay, Oren A Levy, and Elan D Louis. Postural and intention tremors: A detailed clinical study of essential tremor vs. parkinson's disease. *Front. Neurol.*, 4:51, 2013.

[9] Elan D Louis. 'essential tremor' or 'the essential tremors': is this one disease or a family of diseases? *Neuroepidemiology*, 42(2):81–89, 2014.

[10] R Llinás and R A Volkind. The olivo-cerebellar system: functional properties as revealed by harmaline-induced tremor. *Exp. Brain Res.*, 18(1):69–87, 1973.

[11] J Benito-León and M León-Ruiz. Epidemiología del temblor esencial. *Rev. Neurol.*, 70(4):139–148, 2020.

[12] Mark Plumb and Peter Bain. *Essential tremor: The facts*. Oxford University Press, London, England, 2006.

[13] Susanne A Schneider and Günther Deuschl. The treatment of tremor. *Neurotherapeutics*, 11(1):128–138, 2014.

[14] Peter Hedera, František Cibulčík, and Thomas L Davis. Pharmacotherapy of essential tremor. *J. Cent. Nerv. Syst. Dis.*, 5:43–55, 2013.

[15] Andreas Puschmann and Zbigniew K Wszolek. Diagnosis and treatment of common forms of tremor. *Semin. Neurol.*, 31(1):65–77, 2011.

[16] J Kulisevsky, M R Luquin, J M Arbelo, J A Burguera, F Carrillo, A Castro, J Chacón, P J García-Ruiz, E Lezcano, P Mir, J C Martinez-Castrillo, I Martínez-Torres, V Puente, A Sesar, F Valldeoriola-Serra, and R Yañez. Enfermedad de parkinson avanzada. características clínicas y tratamiento (parte i). *Neurologia*, 28(8):503–521, 2013.

[17] Günther Deuschl, Jan Raethjen, Helge Hellriegel, and Rodger Elble. Treatment of patients with essential tremor. *Lancet Neurol.*, 10(2):148–161, 2011.

[18] Robert Francis Dallapiazza, Darrin J Lee, Philippe De Vloo, Anton Fomenko, Clement Hamani, Mojgan Hodaie, Suneil K Kalia, Alfonso Fasano, and Andres M Lozano. Outcomes from stereotactic surgery for essential tremor. *J. Neurol. Neurosurg. Psychiatry*, 90(4):474–482, 2019.

[19] Damián Bendersky, Pablo Ajler, and Claudio Yampolsky. El uso de la neuromodulación para el tratamiento del temblor the use of neuromodulation for the treatment of tremor. *Surg. Neurol. Int.*, 5(Suppl 5):S232–46, 2014.

[20] J R Pérez-Sánchez, R Martínez-Álvarez, N E Martínez Moreno, C Torres Diaz, G Rey, I Pareés, A Del Barrio A, J Álvarez-Linera, and M M Kurtis. Radiocirugía estereotáctica con gamma knife® como tratamiento del temblor esencial y parkinsoniano: experiencia a largo plazo. *Neurol. (Engl. Ed.)*, 2020.

[21] Shayan Moosa, Raul Martínez-Fernández, W Jeffrey Elias, Marta Del Alamo, Howard M Eisenberg, and Paul S Fishman. The role of high-intensity focused ultrasound as a symptomatic treatment for parkinson's disease. *Mov. Disord.*, 34(9):1243–1251, 2019.

[22] Mohammad Rohani and Alfonso Fasano. Focused ultrasound for essential tremor: Review of the evidence and discussion of current hurdles. *Tremor Other Hyperkinet. Mov. (N. Y.)*, 7:462, 2017.

[23] Sarah Paris-Robidas, Elodie Brochu, Marion Sintes, Vincent Emond, Mélanie Bousquet, Milène Vandal, Mireille Pilote, Cyntia Tremblay, Thérèse Di Paolo, Ali H Rajput, Alex Rajput, and Frédéric Calon. Defective dentate nucleus GABA receptors in essential tremor. *Brain*, 135(Pt 1):105–116, 2012.

[24] M Rueda. Efectividad de la estimulación cerebral profunda de núcleo subtalámico en pacientes con enfermedad de parkinson: experiencia de antioquía. *Acta neurol. colomb.*, 2016.

[25] Joel S Perlmutter and Jonathan W Mink. Deep brain stimulation. *Annu. Rev. Neurosci.*, 29(1):229–257, 2006.

[26] Strahinja Dosen, Silvia Muceli, Jakob Lund Dideriksen, Juan Pablo Romero, Eduardo Rocon, Jose Pons, and Dario Farina. Online tremor suppression using electromyography and low-level electrical stimulation. *IEEE Trans. Neural Syst. Rehabil. Eng.*, 23(3):385–395, 2015.

[27] M Javidan, J Elek, and A Prochazka. Attenuation of pathological tremors by functional electrical stimulation. II: Clinical evaluation. *Ann. Biomed. Eng.*, 20(2):225–236, 1992.

[28] A Prochazka, J Elek, and M Javidan. Attenuation of pathological tremors by functional electrical stimulation. i: Method. *Ann. Biomed. Eng.*, 20(2):205–224, 1992.

[29] D M Gillard, T Cameron, A Prochazka, and M J Gauthier. Tremor suppression using functional electrical stimulation: a comparison between digital and analog controllers. *IEEE Trans. Rehabil. Eng.*, 7(3):385–388, 1999.

[30] Lana Popović Maneski, Nikola Jorgovanović, Vojin Ilić, Strahinja Došen, Thierry Keller, Mirjana B Popović, and Dejan B Popović. Electrical stimulation for the suppression of pathological tremor. *Med. Biol. Eng. Comput.*, 49(10):1187–1193, 2011.

[31] Nicola A Maffiuletti. Physiological and methodological considerations for the use of neuromuscular electrical stimulation. *Eur. J. Appl. Physiol.*, 110(2):223–234, 2010.

[32] C Scott Bickel, Chris M Gregory, and Jesse C Dean. Motor unit recruitment during neuromuscular electrical stimulation: a critical appraisal. *Eur. J. Appl. Physiol.*, 111(10):2399–2407, 2011.

[33] Giuliana Grimaldi, Stéphane Camut, and Mario Manto. Functional electrical stimulation effect on upper limb tremor. 2011.

[34] Juan Álvaro Gallego, Eduardo Rocon, Juan Manuel Belda-Lois, and José Luis Pons. A neuroprosthesis for tremor management through the control of muscle co-contraction. *J. Neuroeng. Rehabil.*, 10(1):36, 2013.

[35] Antônio Padilha Lanari Bó, Christine Azevedo-Coste, Christian Geny, Philippe Poignet, and Charles Fattal. On the use of fixed-intensity functional electrical stimulation for attenuating essential tremor: Thoughts and progress. *Artif. Organs*, 38(11):984–991, 2014.

[36] F.O. Barroso A. Pascual-Valdunciel and J.L. Pons. Motor inhibition elicited by electrical stimulation of afferent pathways and its application in tremor suppression.

[37] Alejandro Pascual-Valdunciel, Miguel Gonzalez-Sanchez, Silvia Muceli, Beatriz Adan-Barrientos, Viviana Escobar-Segura, Javier Ricardo Perez-Sanchez, Moon Ki Jung, Andreas Schneider, Klaus-Peter Hoffmann, Juan Camilo Moreno, Francisco Grandas, Dario Farina, Jose Luis Pons, and Filipe Oliveira

Barroso. Intramuscular stimulation of muscle afferents attains prolonged tremor reduction in essential tremor patients. *IEEE Trans. Biomed. Eng.*, 68(6):1768–1776, 2021.

[38] Toshiyuki Fujiwara. The role of spinal reciprocal inhibition and intracortical inhibition in functional recovery from stroke. *Exp. Brain Res.*, 238(7-8):1701–1705, 2020.

[39] Jakob L Dideriksen, Christopher M Laine, Strahinja Dosen, Silvia Muceli, Eduardo Rocon, José L Pons, Julian Benito-Leon, and Dario Farina. Electrical stimulation of afferent pathways for the suppression of pathological tremor. *Front. Neurosci.*, 11:178, 2017.

[40] Aurelien Geron. *Hands-on machine learning with scikit-learn, keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems*. O'Reilly Media, Sebastopol, CA, 2 edition, 2019.

[41] Michael W Berry, Azlinah Mohamed, and Bee Wah Yap. *Supervised and unsupervised learning for data science*. 2020.

[42] Charu C Aggarwal. *Neural networks and deep learning: A textbook*. Springer International Publishing, Cham, Switzerland, 1 edition, 2018.

[43] Nikolas Adaloglou and Sergios Karagiannakos. Recurrent neural networks: building a custom lstm cell. *https://theaisummer.com/*, 2020.

[44] Wikipedia contributors. Recurrent neural network. https://en.wikipedia.org/w/index.php?title=Recurrent_neural_network&oldid=1090237075, May 2022. Accessed: NA-NA-NA.

[45] Klaus Greff, Rupesh Kumar Srivastava, Jan Koutník, Bas R Steunebrink, and Jürgen Schmidhuber. LSTM: A search space odyssey. 2015.

[46] Julien Stamatakis, Jérome Ambroise, Julien Crémers, Hoda Sharei, Valérie Delvaux, Benoit Macq, and Gaëtan Garraux. Finger tapping clinimetric score prediction in parkinson's disease using low-cost accelerometers. *Comput. Intell. Neurosci.*, 2013:717853, 2013.

[47] Hyoseon Jeon, Woongwoo Lee, Hyeyoung Park, Hong Ji Lee, Sang Kyong Kim, Han Byul Kim, Beomseok Jeon, and Kwang Suk Park. Automatic classification of tremor severity in parkinson's disease using a wearable device. *Sensors (Basel)*, 17(9), 2017.

[48] Md Nafiul Alam, Benjamin Johnson, Jeffrey Gendreau, Kouhyar Tavakolian, Colin Combs, and Reza Fazel-Rezai. Tremor quantification of parkinson's disease - a pilot study. In *2016 IEEE International Conference on Electro Information Technology (EIT)*. IEEE, 2016.

[49] Robert LeMoyne, Nestor Tomycz, Timothy Mastroianni, Cyrus McCandless, Michael Cozza, and David Peduto. Implementation of a smartphone wireless

accelerometer platform for establishing deep brain stimulation treatment efficacy of essential tremor with machine learning. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015.

[50] Ana Camila Alves de Araújo, Enzo Gabriel da Rocha Santos, Karina Santos Guedes de Sá, Viviane Kharine Teixeira Furtado, Felipe Augusto Santos, Ramon Costa de Lima, Lane Viana Krejcová, Bruno Lopes Santos-Lobato, Gustavo Henrique Lima Pinto, André Dos Santos Cabral, Anderson Belgamo, Bianca Callegari, Ana Francisca Rozin Kleiner, Anselmo de Athayde Costa E Silva, and Givago da Silva Souza. Hand resting tremor assessment of healthy and patients with parkinson's disease: An exploratory machine learning study. *Front. Bioeng. Biotechnol.*, 8:778, 2020.

[51] Soroosh Shahtalebi, Seyed Farokh Atashzar, Olivia Samotus, Rajni V Patel, Mandar S Jog, and Arash Mohammadi. PHTNet: Characterization and deep mining of involuntary pathological hand tremor using recurrent neural network models. *Sci. Rep.*, 10(1):2195, 2020.

[52] Rafael Anicet Zanini, Esther Luna Colombini, and Maria Claudia Ferrari de Castro. Parkinson's disease EMG signal prediction using neural networks. In *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*. IEEE, 2019.

[53] Anas Ibrahim, Yue Zhou, Mary E Jenkins, Ana Luisa Trejos, and Michael D Naish. The design of a parkinson's tremor predictor and estimator using a hybrid convolutional-multilayer perceptron neural network. *Annu Int Conf IEEE Eng Med Biol Soc*, 2020:5996–6000, 2020.

[54] Kailash P Bhatia, Peter Bain, Nin Bajaj, Rodger J Elble, Mark Hallett, Elan D Louis, Jan Raethjen, Maria Stamelou, Claudia M Testa, Guenther Deuschl, and Tremor Task Force of the International Parkinson and Movement Disorder Society. Consensus statement on the classification of tremors. from the task force on tremor of the international parkinson and movement disorder society. *Mov. Disord.*, 33(1):75–87, 2018.

[55] S. Fahn, Eduard Tolosa, and C. Marin. Clinical rating scale for tremor. 2:271–280, 01 1988.

[56] William Guy. *Clinical global impressions in ECDEU assessment manual for psychopharmacology*. National Institue of Mental Health, 1976.

[57] John Karl. *Introduction to Digital Signal Processing*. Academic Press, San Diego, CA, 1989.

[58] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[59] Alan V Oppenheim and Alan S Willsky. *Signals and Systems*. Pearson, Upper Saddle River, NJ, 2 edition, 1996.

[60] Donald B Percival and Andrew T Walden. *Spectral Analysis Physical Applications*. Cambridge University Press, Cambridge, England, 2004.

[61] Sarah Guido and Andreas C Mueller. *Introduction to machine learning with python: A guide for data scientists*. O'Reilly Media, Sebastopol, CA, 2016.

[62] Harry Zhang. The optimality of naive bayes. volume 2, 01 2004.

[63] Thomas Wood. Random forests. https://deepai.org/machine-learning-glossary-and-terms/random-forest, September 2020. Accessed: 2022-6-17.

[64] Junyi Cao, Zhongming Tian, and Zhengtao Wang. Hand gestures recognition based on one-channel surface emg signal*. *Journal of Software Engineering and Applications*, 12:383–392, 01 2019.

[65] Jaime Carrillo-Moreno, Carmen Pérez-Gandía, Rafael Sendra-Arranz, Gema García-Sáez, M Elena Hernando, and Álvaro Gutiérrez. Long short-term memory neural network for glucose prediction. *Neural Comput. Appl.*, 33(9):4191–4203, 2021.

[66] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. 2014.

[67] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. 2017.

[68] General data protection regulation (GDPR) compliance guidelines. https://gdpr.eu/, June 2018. Accessed: 2022-6-21.

Appendix A

Ethical, economic, social and environmental impacts

A.1 Introduction

Pathological tremor is one of the most common symptoms caused by motor dysfunctions, such as ET or PD. It leads to patients suffering from cannot control their own movements and not being able to carry out many of their daily life activities. It is another critical medical issue that has to be investigated, so it needs for general, affordable and effective solutions. Therefore, this thesis is involved into the experiments which aim to achieve better remedies to pathological tremor, in substitution to traditional solutions such as pharmacological approaches or surgical procedures. More in detail, this thesis aims to assess viability of using EMG signal classification and prediction for recent tremor suppression techniques, such as FES and afferent stimulation. Besides the experiments conducted throughout this thesis could help on improving those strategies, it is important to understand and evaluate the main impacts this investigation has on society.

A.2 Analysis of most relevant impacts related to this project

- **Ethical impact**. In this thesis, data collected from ET patients were used in order to carry out the experiments. It is important to ensure that the data collection process obeys the current legislation for data protection. In this case, these process is protected by the General Data Protection Regulation (GDPR) from the European Union (Regulation (EU) 2016/679) that was put into effect on May 25, 2018 [68].

- **Social impact**. This project has an straight effect on patients who suffer from pathological tremor caused by ET disease, as well as for their families. The development of new and better tremor handling techniques derives into more comfortable and personalized treatments which help to improve patients quality of living.

- **Economic impact**. The results brought by this thesis can help in the development of new treatment approaches for pathological tremor. Nowadays, most treatments are based on medication or surgical procedures, which both suppose an important investment of economic resources, while not being completely effective solutions. New strategies such as FES or afferent stimulation, to which this investigation aims to improve, are considered as the future treatments for handling tremor, being less expensive and more effective than current ones.

- **Environmental impact**. This thesis is involved into the development of sustainable and wide-applicable approaches for handling pathological tremor, namely: FES and afferent stimulation. In conclusion, further results of this investigation would have a minimal detrimental impact on the environment.

A.3 Conclusions

After performed this evaluation of the different impacts of this thesis, we can conclude that it is worth to invest on this field of investigation, which, hopefully, in the incoming future would bring better and sustainable solutions for tremor suppression systems, helping patients and society to overcome the drawbacks of this type of diseases.

Appendix B

Economic budget

This project was conducted throughout six months in collaboration with the Neural Rehabilitation Group from the Cajal Institute-CSIC. All data employed in the experiments come from the European project EXTEND-Bidirectional Hyper-Connected Neural System, in which the Cajal Institute is contributing. Regarding to the required economic budget, in order to conduct the investigations of this thesis, an estimation of costs is presented below.

- **Personnel**: in terms of personnel costs, salaries for the project leader (Biomedical Engineer) and the student were considered (see Table B.1).

	Hourly rate (€)	Hours	Total (€)
Project leader	25	85	2.125
Student	15	600	9.000
TOTAL			**11.125**

Table B.1: Personnel costs.

- **Costs from material resources**: in order to carry out the experiments made in this investigation, the required hardware has been valued, taking into account the corresponding devaluation for each component (see Table B.2).

	Useful life (years)	Units.	Cost (€)	Amortization (€/month)	Usage (months)	Total (€)
Personal computer	5	1	1.000	16,66	6	100
Graphics Card (NVIDIA GTX 1080 Ti)	4	1	800	13,33	6	80
TOTAL						**180**

Table B.2: Costs from material resources.

Finally, total costs are summarized in Table B.3.

	Cost
Personnel costs	11.125 €
Material costs	180 €
Subtotal	11.305 €
IVA	2.374,05 €
Total	13.679,05 €

Table B.3: Total costs.

Appendix C

Vanishing Gradients problem

The vanishing gradients problem refers to the fact that gradients tend to zero as time sequences are longer. This is a very important issue that RNN architectures are facing. In order to demonstrate the existence of this problem, an example based on the resolution for weight matrix **W** is shown below, being the cases of matrix **U** and vector **b** similar to this.

Let the partial derivative of loss function $\mathcal{L}$ with respect to **W** for time step t be:

$$\begin{aligned} \frac{\partial \mathcal{L}_t}{\partial \mathbf{W}_t} &= \frac{\partial \mathcal{L}_t}{\partial \mathbf{y}_t}\frac{\partial \mathbf{y}_t}{\partial \mathbf{h}_t}\cdots\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial \mathbf{W}_t} = \frac{\partial \mathcal{L}_t}{\partial \mathbf{y}_t}\frac{\partial \mathbf{y}_t}{\partial \mathbf{h}_t}\left(\prod_{t=2}^{T}\frac{\partial \mathcal{L}_t}{\partial \mathbf{W}_t}\frac{\partial \mathbf{h}_t}{\partial \mathbf{h}_{t-1}}\right)\frac{\partial \mathbf{y}_t}{\partial \mathbf{W}} \\ \mathbf{h}_t &= f(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}) \end{aligned} \tag{C.1}$$

Then, evaluating the partial derivatives in the sequential product leads to:

$$\frac{\partial \mathcal{L}_t}{\partial \mathbf{W}_t} = \frac{\partial \mathcal{L}_t}{\partial \mathbf{y}_t}\frac{\partial \mathbf{y}_t}{\partial \mathbf{h}_t}\cdots\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial \mathbf{W}_t} = \frac{\partial \mathcal{L}_t}{\partial \mathbf{y}_t}\frac{\partial \mathbf{y}_t}{\partial \mathbf{h}_t}\left(\prod_{t=2}^{T} f'(\mathbf{U}\mathbf{h}_{t-1} + \mathbf{W}\mathbf{x}_t + \mathbf{b})\mathbf{U}\right)\frac{\partial \mathbf{y}_t}{\partial \mathbf{W}} \tag{C.2}$$

where f is the activation function.

Usually, $\tanh$ and *sigmoid* are used as activation functions. $\tanh$ maps their entries between -1 and 1, while *sigmoid* function does it between 0 and 1. The derivatives of these functions are delimited to 1, which leads to the derivative of $\mathcal{L}$ tending to 0 for any t time step.

$$\begin{aligned} &\frac{\partial \mathcal{L}_t}{\partial \mathbf{W}_t} \to 0 \\ &\mathbf{W}_{t+1} = \mathbf{W}_t - \alpha\frac{\partial \mathcal{L}_t}{\partial \mathbf{W}_t} \approx \mathbf{W}_t \end{aligned} \tag{C.3}$$

Therefore, gradients would not be updated anymore, for a sufficiently great value of t.

Appendix D

Cross Validation ML models

D.1 KNN

KNN Models - Filtered EMG

ID	Algorithm	Leaf Size	Metric	Neighbors	Weights
Model 1	Ball-Tree	30	Euclidean	4	Distance
Model 2	Ball-Tree	30	Euclidean	5	Distance
Model 3	Ball-Tree	30	Euclidean	6	Distance
Model 4	Ball-Tree	30	Chebyshev	4	Distance
Model 5	Ball-Tree	30	Chebyshev	5	Distance
Model 6	Ball-Tree	30	Chebyshev	6	Distance
Model 7	Ball-Tree	35	Euclidean	4	Distance
Model 8	Ball-Tree	35	Euclidean	5	Distance
Model 9	Ball-Tree	35	Euclidean	6	Distance
Model 10	Ball-Tree	35	Chebyshev	4	Distance
Model 11	Ball-Tree	35	Chebyshev	5	Distance
Model 12	Ball-Tree	35	Chebyshev	6	Distance

Table D.1: Cross Validation KNN models for filtered EMG.

KNN Models - Raw EMG

ID	Algorithm	Leaf Size	Metric	Neighbors	Weights
Model 1	Ball-Tree	30	Euclidean	2	Distance
Model 2	Ball-Tree	30	Chebyshev	2	Distance
Model 3	Ball-Tree	30	Minkowski	2	Distance
Model 4	Ball-Tree	35	Euclidean	2	Distance
Model 5	Ball-Tree	35	Chebyshev	2	Distance
Model 6	Ball-Tree	35	Minkowski	2	Distance

Table D.2: Cross Validation KNN models for raw EMG.

	Window			
	1.0 s	**0.8 s**	**0.6 s**	**0.4 s**
Raw EMG	Model 1	Model 1	Model 1	Model 1
Filtered EMG	Model 1	Model 1	Model 3	Model 3

Table D.3: Best KNN models per window for raw and filtered EMG.

D.2 RF

RF Models

ID	Criterion	Max. Features	Min. Samples per leaf	Trees
Model 1	Entropy	log2	2	85
Model 2	Entropy	log2	2	90
Model 3	Entropy	log2	2	95
Model 4	Entropy	log2	2	100
Model 5	Entropy	sqrt	2	85
Model 6	Entropy	sqrt	2	90
Model 7	Entropy	sqrt	2	95
Model 8	Entropy	sqrt	2	100

Table D.4: Cross Validation RF models.

	Window			
	1.0 s	**0.8 s**	**0.6 s**	**0.4 s**
Raw EMG	Model 3	Model 8	Model 6	Model 8
Filtered EMG	Model 1	Model 8	Model 7	Model 4

Table D.5: Best RF models per window for raw and filtered EMG

D.3 SVM

SVM Models

ID	C	γ	Kernel	ID	C	γ	Kernel
Model 1	0.1	1	RBF	Model 6	1	0.01	RBF
Model 2	0.1	0.1	RBF	Model 7	10	1	RBF
Model 3	0.1	0.01	RBF	Model 8	10	0.1	RBF
Model 4	1	1	RBF	Model 9	10	0.01	RBF
Model 5	1	0.1	RBF				

Table D.6: Cross Validation SVM models.

	Window			
	1.0 s	**0.8 s**	**0.6 s**	**0.4 s**
Raw EMG	Model 9	Model 9	Model 8	Model 8
Filtered EMG	Model 4	Model 4	Model 4	Model 7

Table D.7: Best SVM models per window for raw and filtered EMG

D.4 LSTM classifier

LSTM classifiers

ID	Learning rate	Hidden size	ID	Learning rate	Hidden size
Model 1	0.005	20	Model 6	0.0001	50
Model 2	0.005	35	Model 7	0.0001	20
Model 3	0.005	50	Model 8	0.0001	35
Model 4	0.001	20	Model 9	0.0001	50
Model 5	0.001	35			

Table D.8: LSTM models for classification.

	Window			
	1.0 s	**0.8 s**	**0.6 s**	**0.4 s**
Raw EMG	Model 2	Model 1	Model 4	Model 3
Filtered EMG	Model 1	Model 6	Model 5	Model 9

Table D.9: Best LSTM classifiers per window for raw and filtered EMG

Appendix E

Confusion matrices from classifiers

E.1 GNB

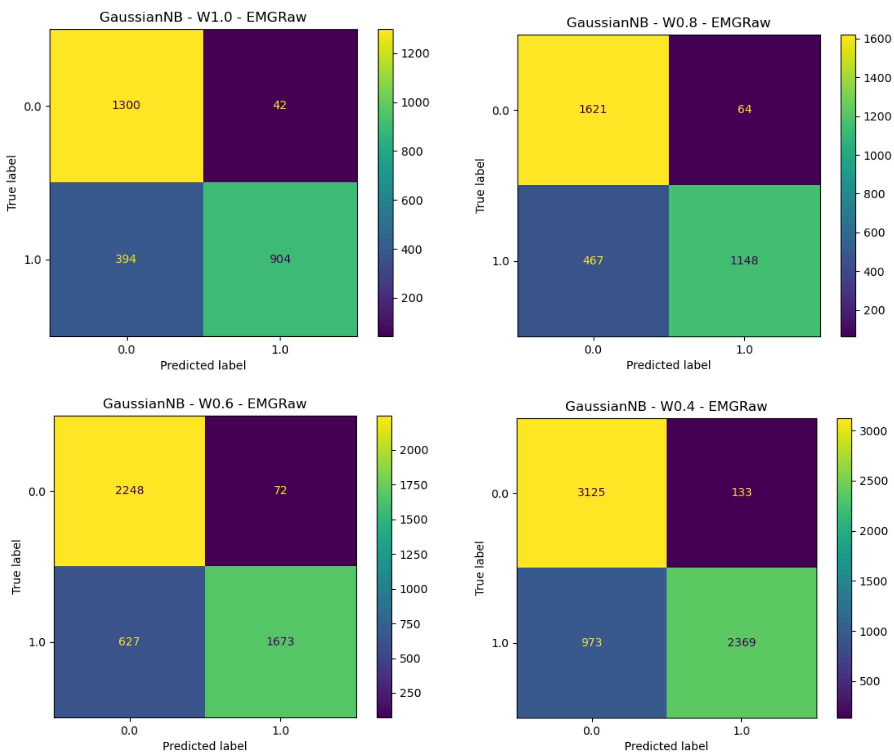


Figure E.1: Confusion matrices for GNB per window (Raw EMG).

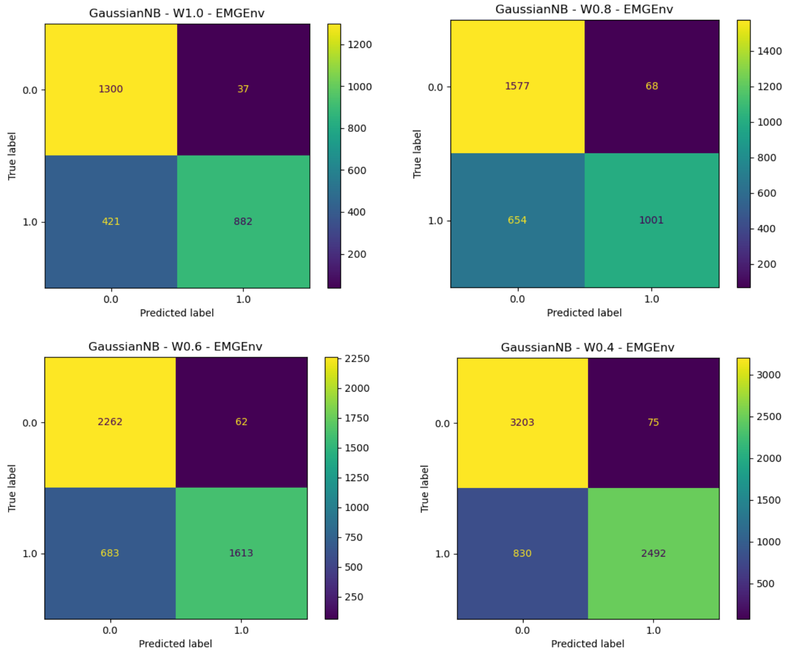


Figure E.2: Confusion matrices for GNB per window (Filtered EMG).

E.2 KNN

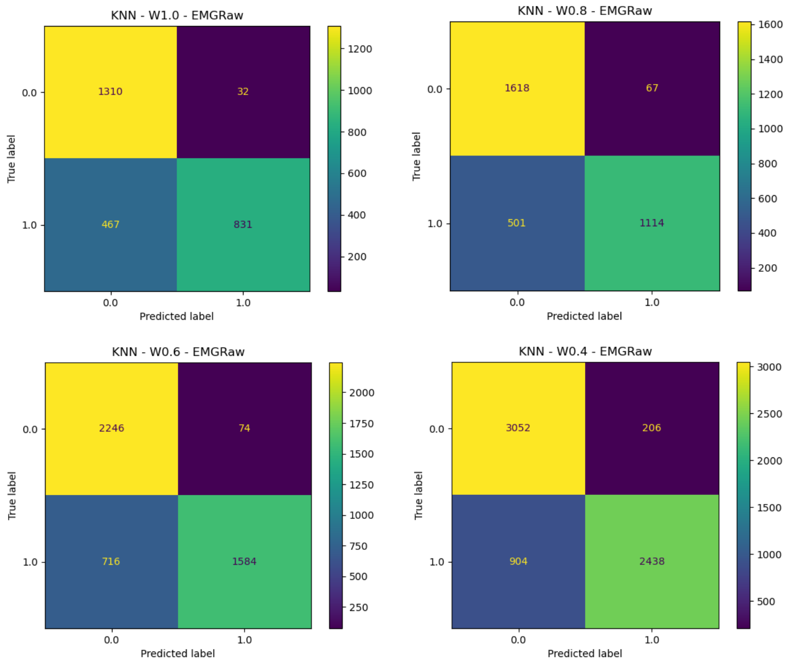


Figure E.3: Confusion matrices for KNN per window (Raw EMG).

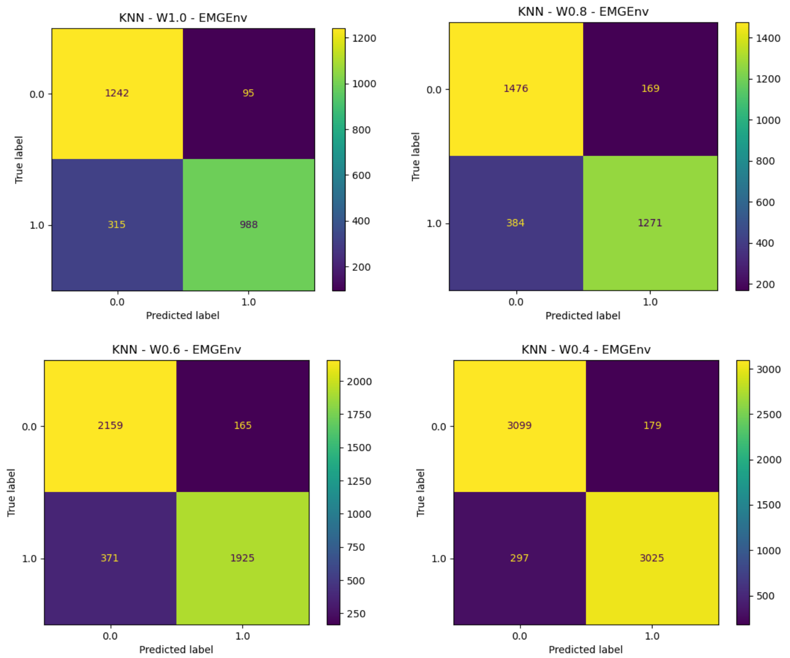


Figure E.4: Confusion matrices for KNN per window (Filtered EMG).

E.3 RF

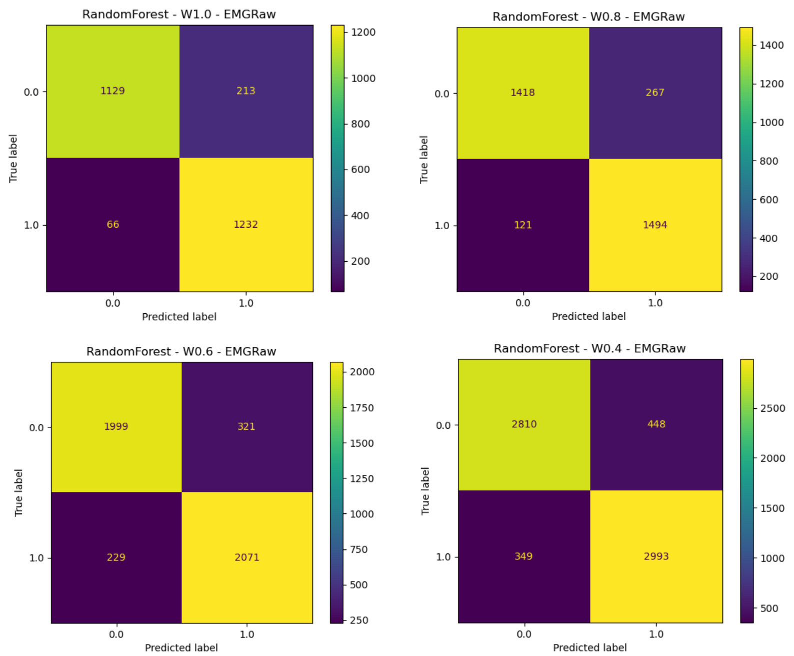


Figure E.5: Confusion matrices for RF per window (Raw EMG).

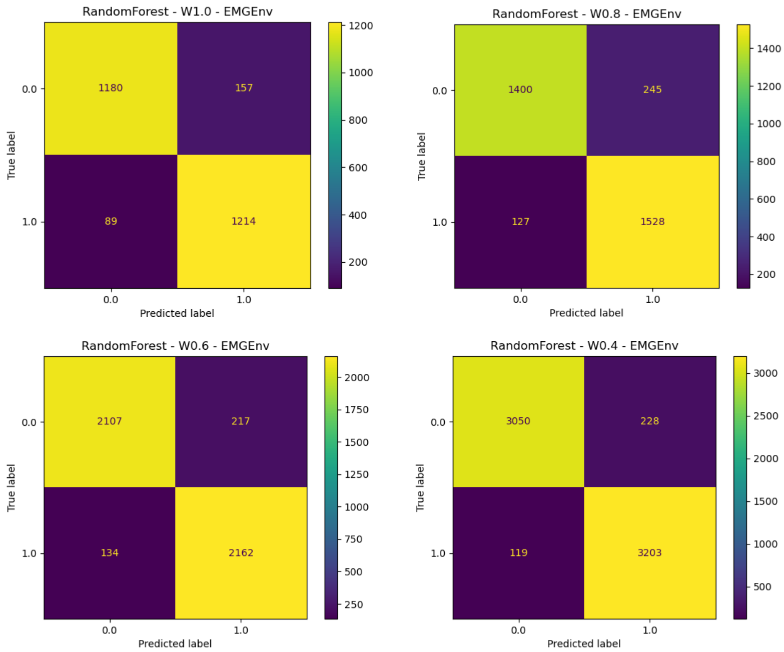


Figure E.6: Confusion matrices for RF per window (Filtered EMG).

E.4 SVM

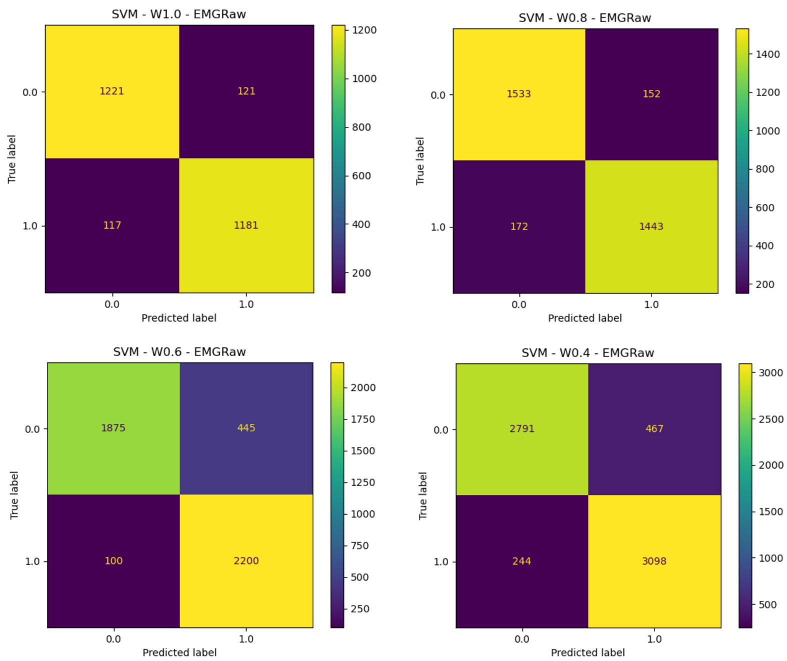


Figure E.7: Confusion matrices for SVM per window (Raw EMG).

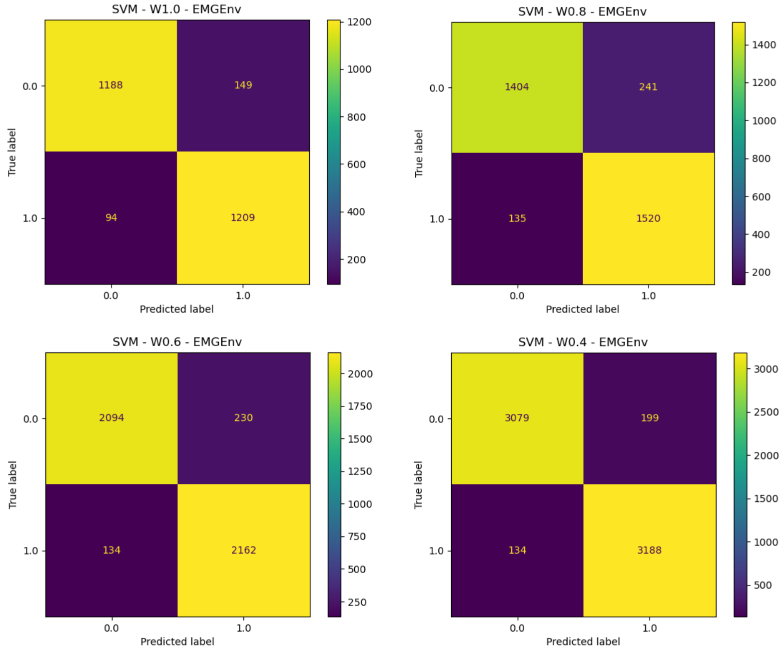


Figure E.8: Confusion matrices for SVM per window (Filtered EMG).

Appendix F

Boxplots from K-Fold Cross Validation results

F.1 KNN

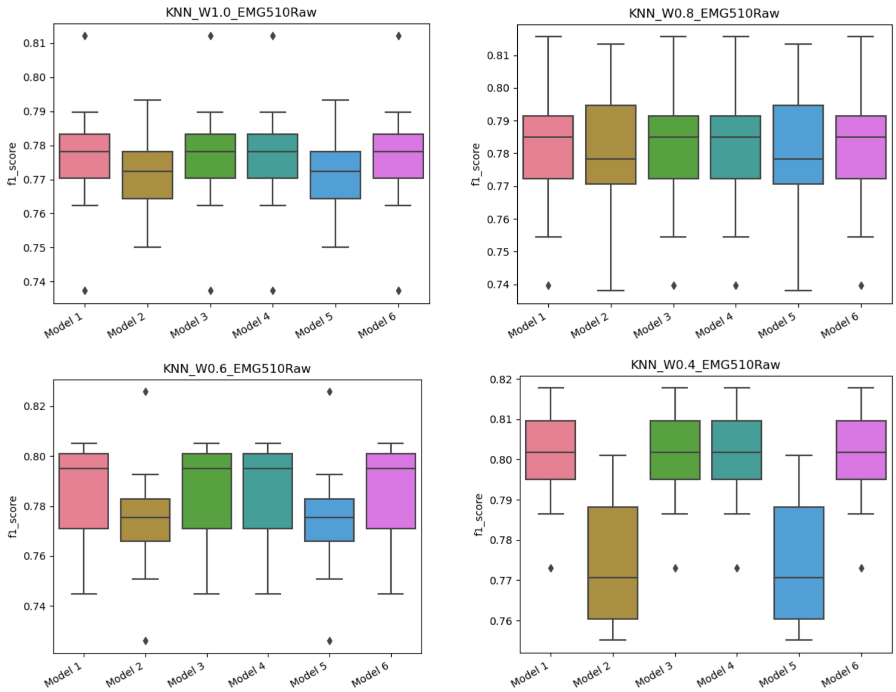


Figure F.1: K-Fold Cross Validation boxplots for KNN and raw EMG.

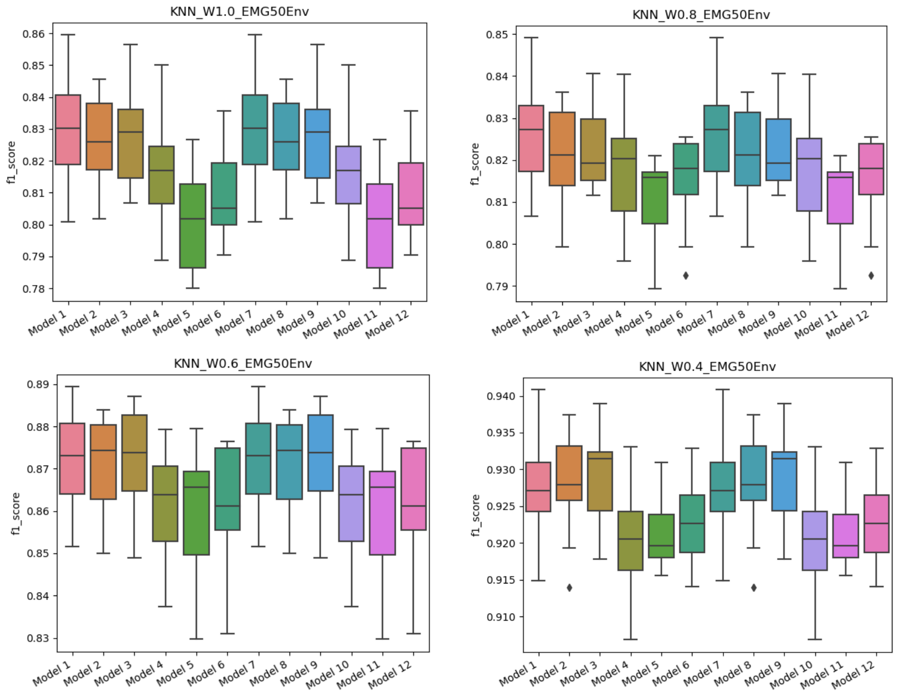


Figure F.2: K-Fold Cross Validation boxplots for KNN and filtered EMG.

F.2 RF

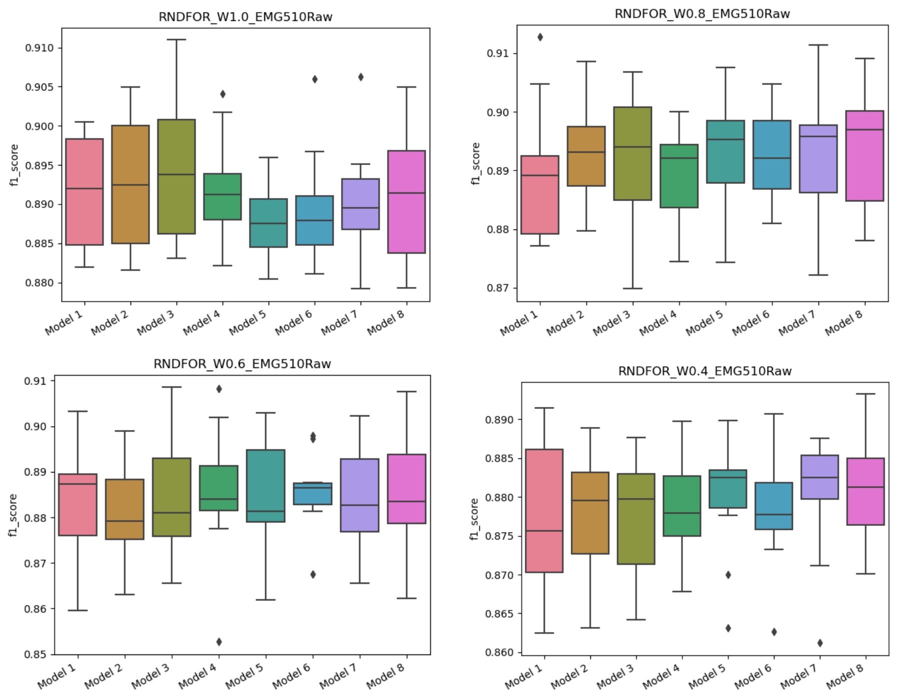


Figure F.3: K-Fold Cross Validation boxplots for RF and raw EMG.

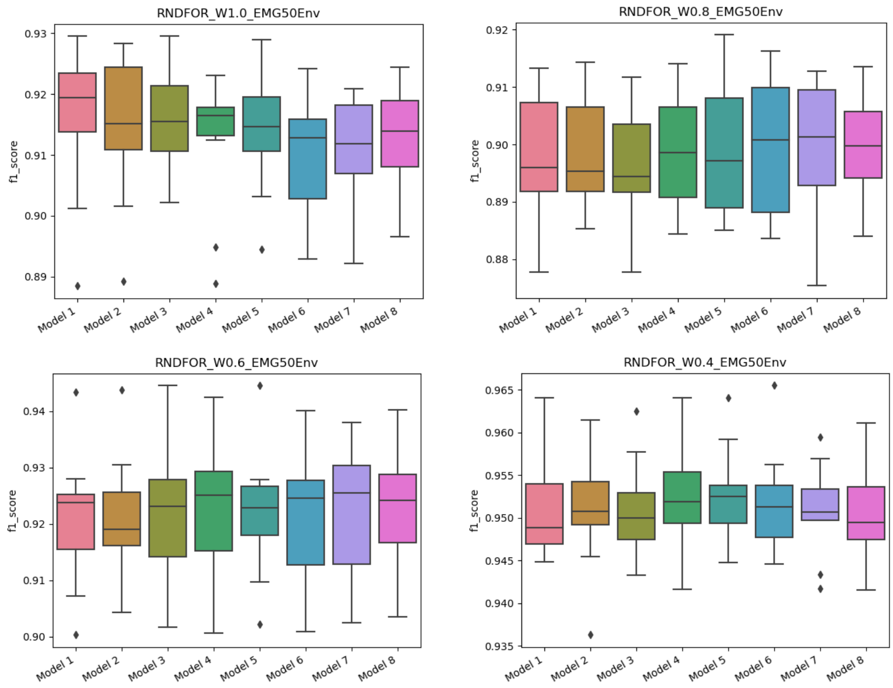


Figure F.4: K-Fold Cross Validation boxplots for RF and filtered EMG.

F.3 SVM

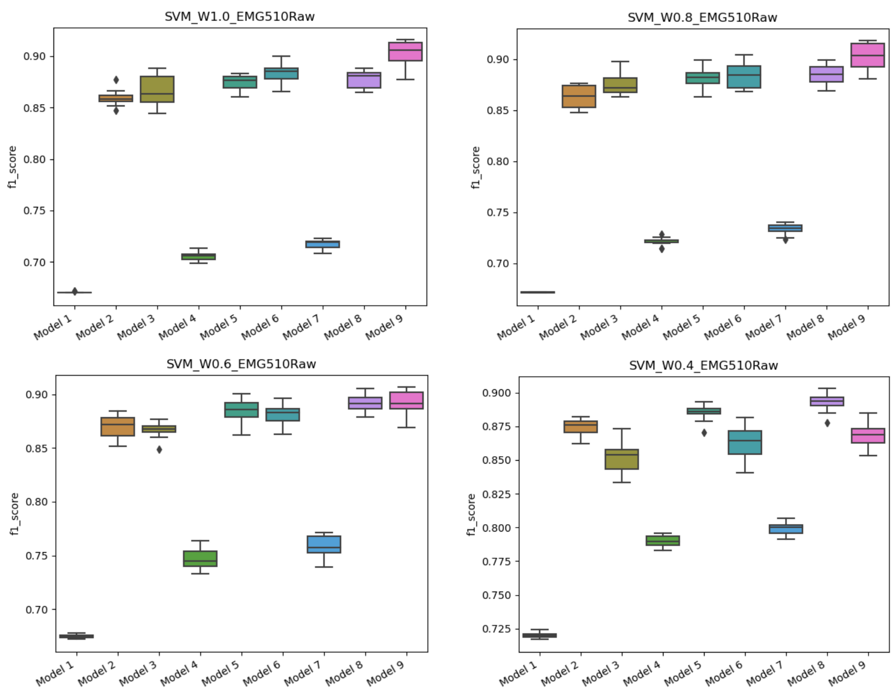


Figure F.5: K-Fold Cross Validation boxplots for SVM and raw EMG.

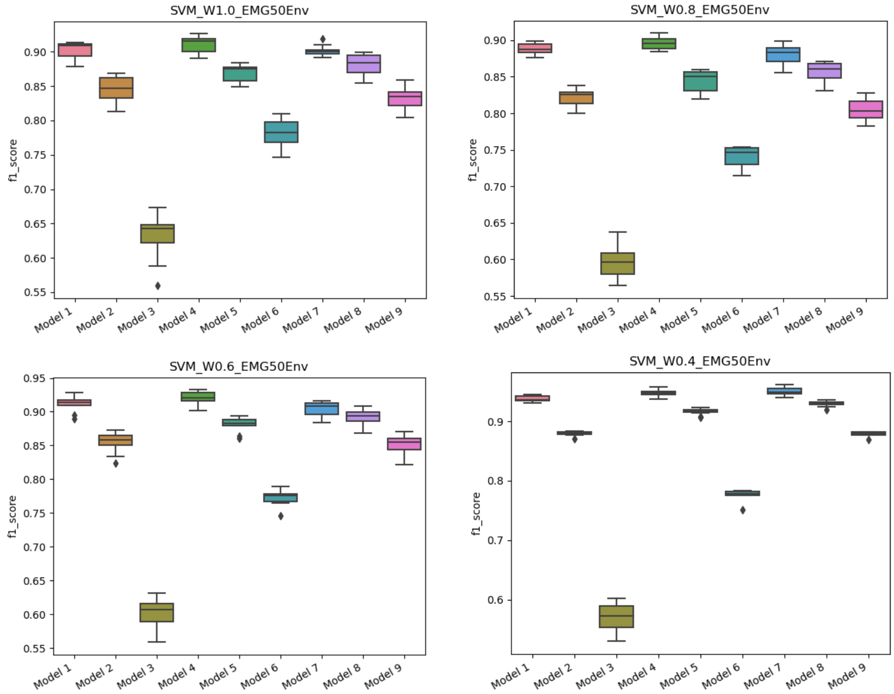
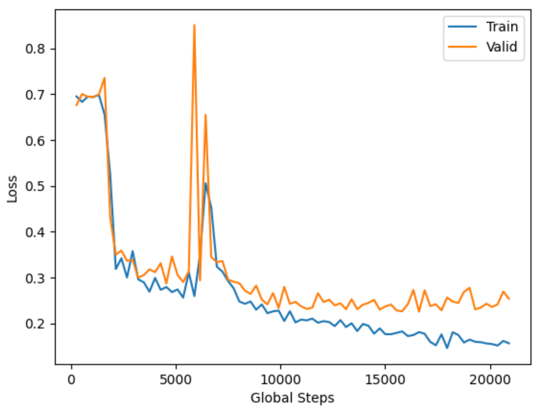


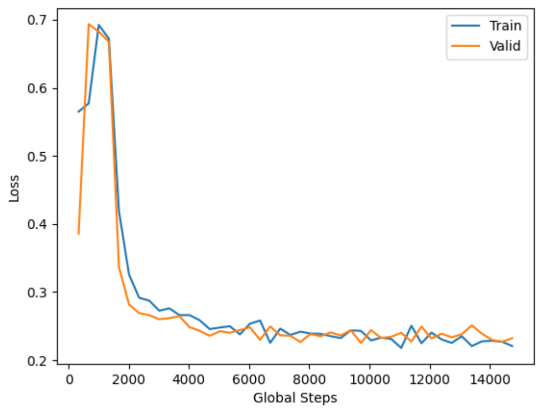
Figure F.6: K-Fold Cross Validation boxplots for SVM and filtered EMG.

Appendix G

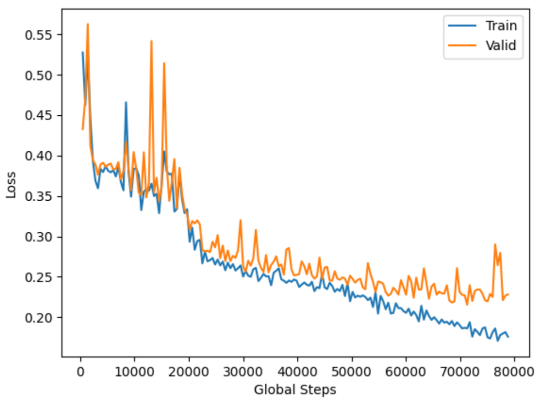
Train-Valid Loss graphs from best LSTM classifiers



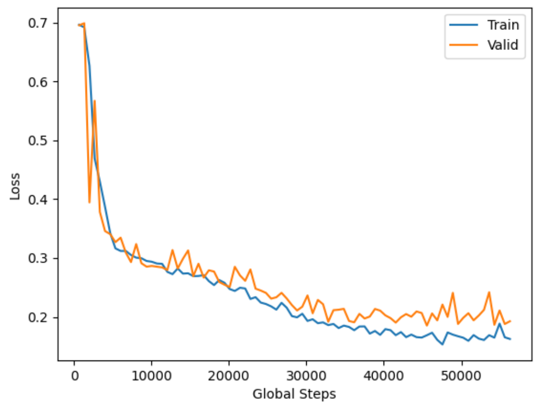
(a) W = 1.0 s.



(b) W = 0.8 s.

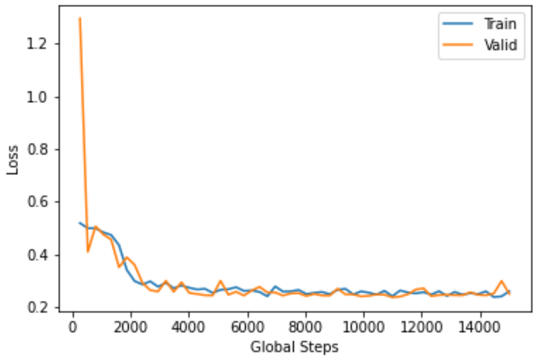


(c) W = 0.6 s.

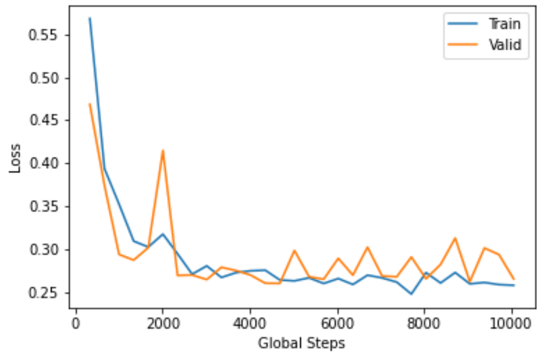


(d) W = 0.4 s.

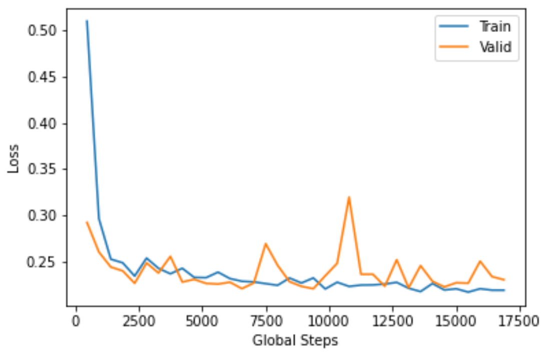
Figure G.1: Train-Valid loss graphs for best LSTM classifiers using raw EMG.



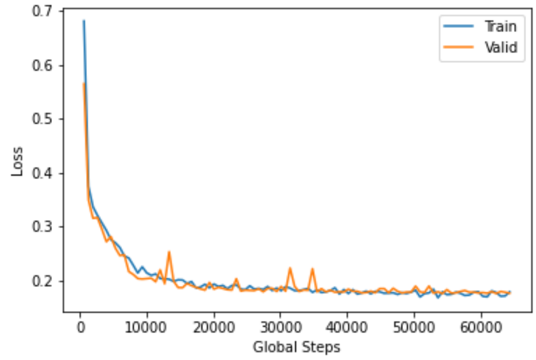
(a) W = 1.0 s.



(b) W = 0.8 s.



(c) W = 0.6 s.



(d) W = 0.4 s.

Figure G.2: Train-Valid loss graphs for best LSTM classifiers using filtered EMG.

Appendix H

Boxplots from prediction models

H.1 20 train samples

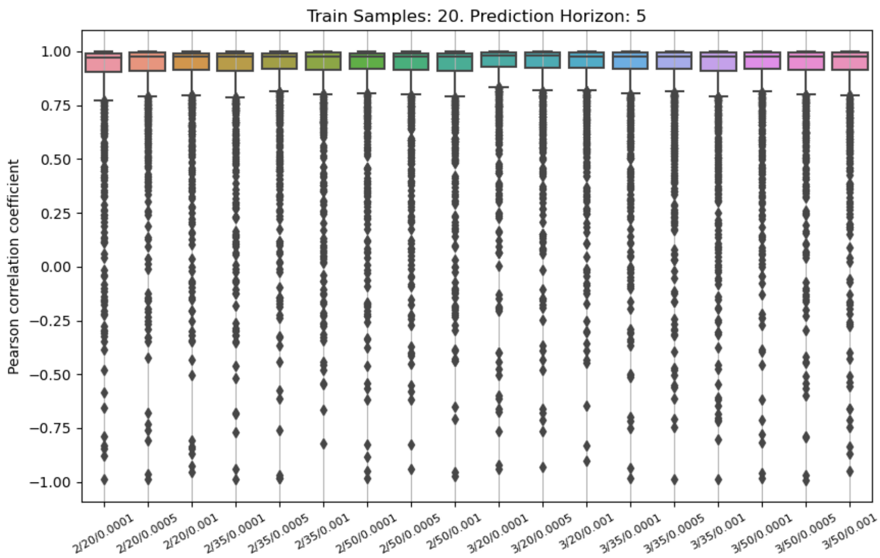


Figure H.1: Boxplots from prediction models. Train samples: 20. Prediction Horizon: 5.

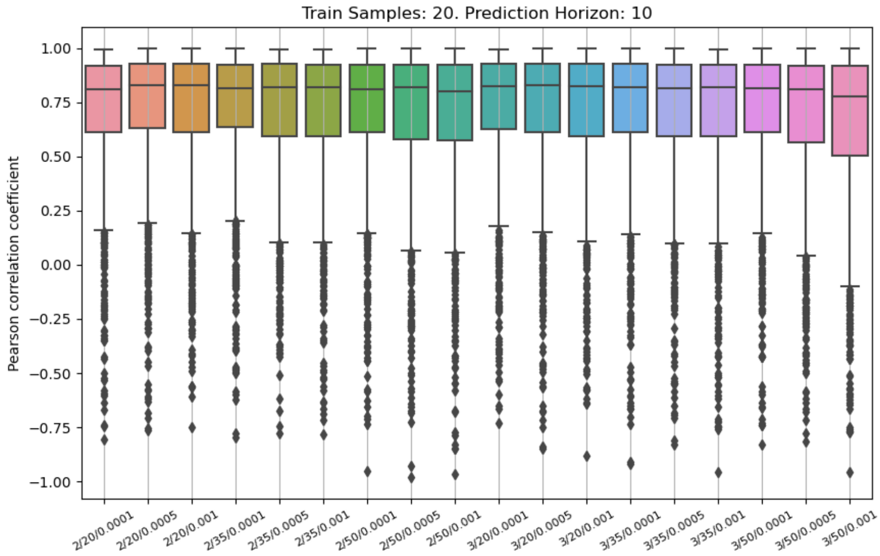


Figure H.2: Boxplots from prediction models. Train samples: 20. Prediction Horizon: 10.

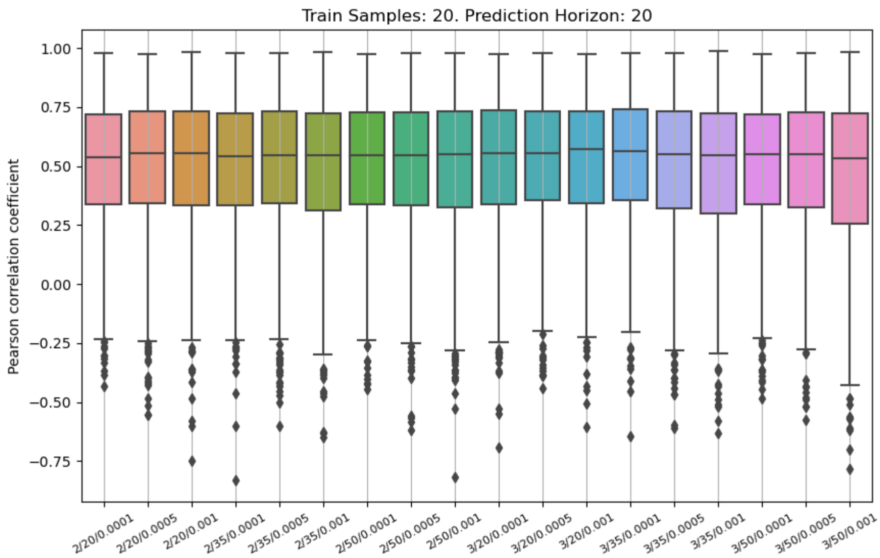


Figure H.3: Boxplots from prediction models. Train samples: 20. Prediction Horizon: 20.

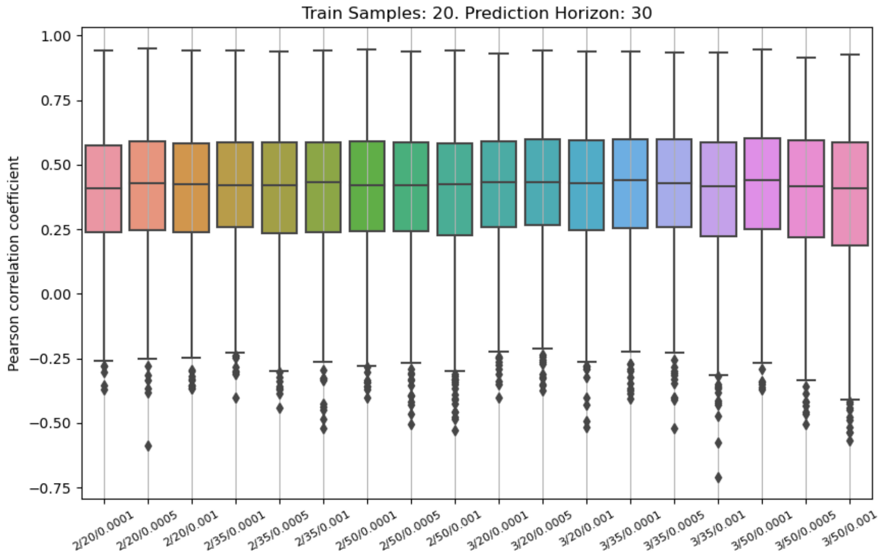


Figure H.4: Boxplots from prediction models. Train samples: 20. Prediction Horizon: 30.

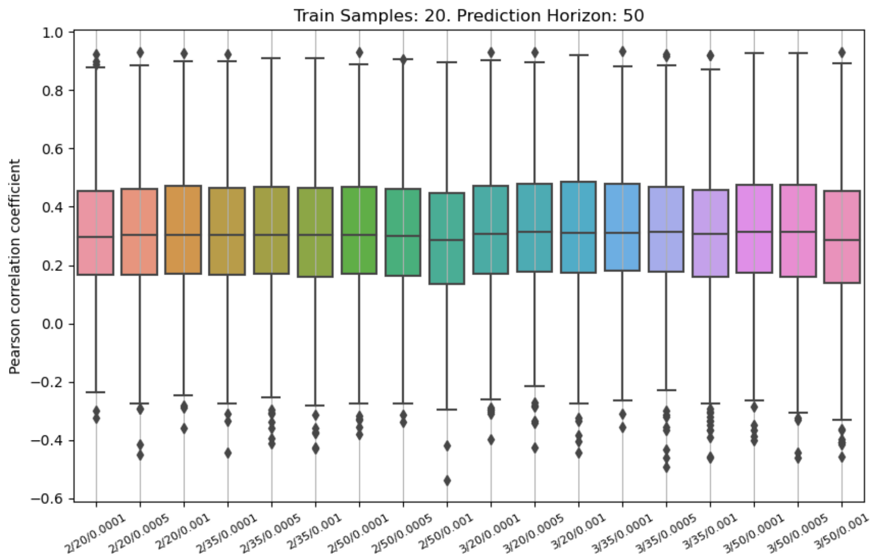


Figure H.5: Boxplots from prediction models. Train samples: 20. Prediction Horizon: 50.

H.2 30 train samples

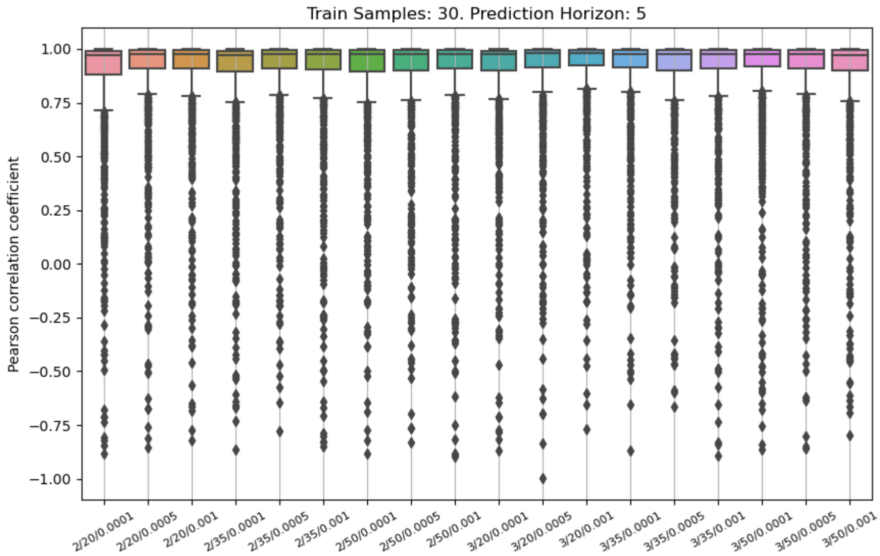


Figure H.6: Boxplots from prediction models. Train samples: 30. Prediction Horizon: 5.

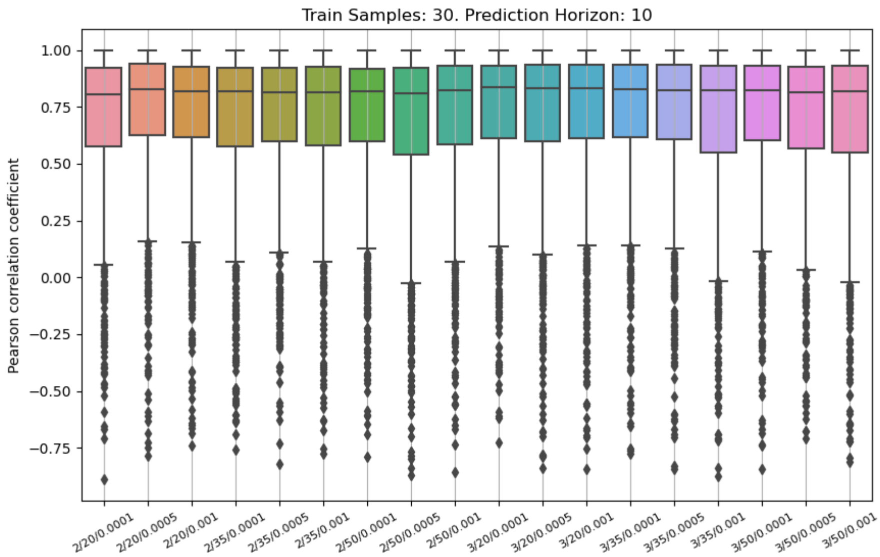


Figure H.7: Boxplots from prediction models. Train samples: 30. Prediction Horizon: 10.

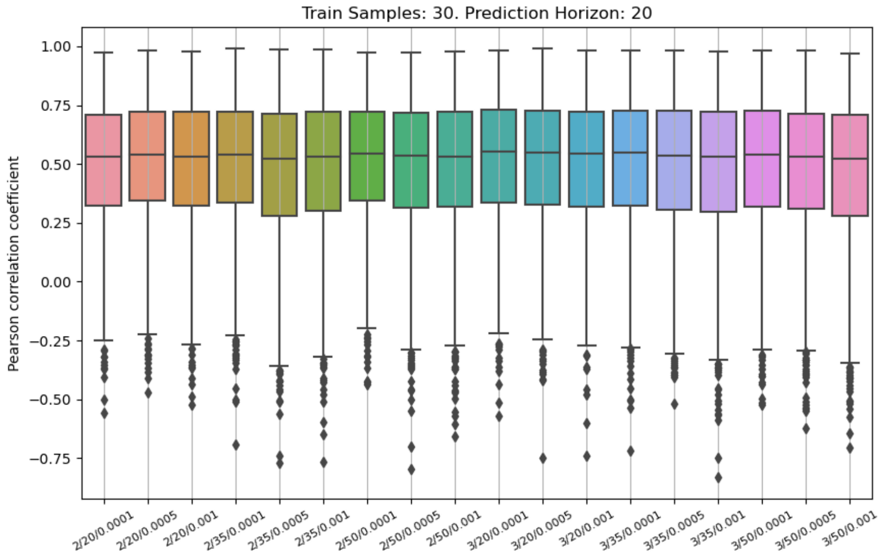


Figure H.8: Boxplots from prediction models. Train samples: 30. Prediction Horizon: 20.

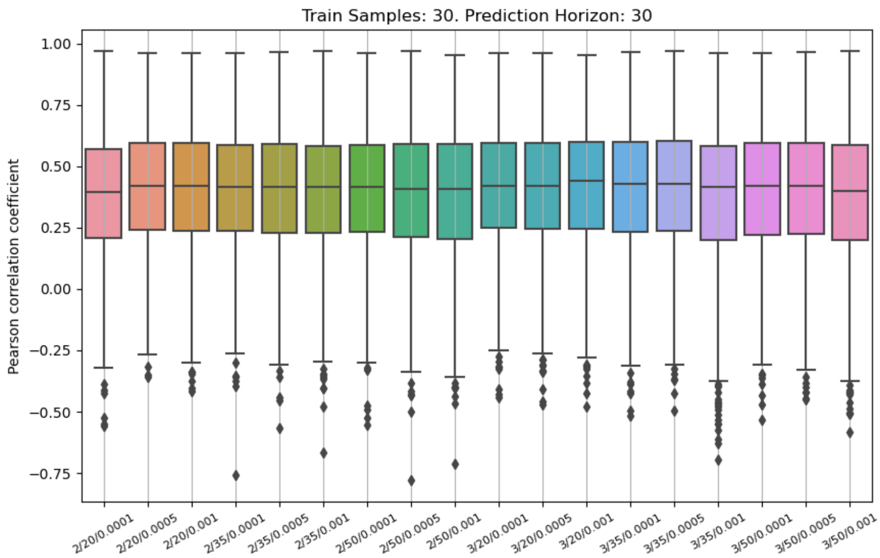


Figure H.9: Boxplots from prediction models. Train samples: 30. Prediction Horizon: 30.

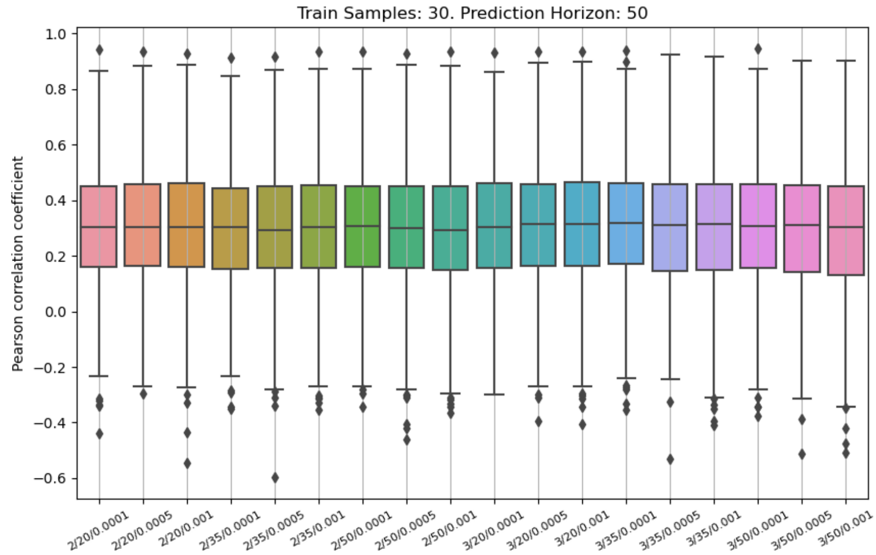


Figure H.10: Boxplots from prediction models. Train samples: 30. Prediction Horizon: 50.

H.3 40 train samples

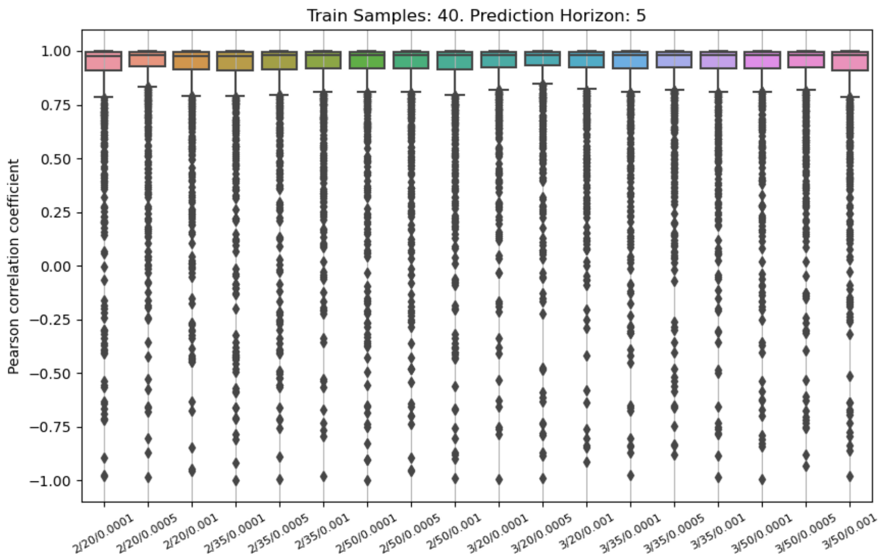


Figure H.11: Boxplots from prediction models. Train samples: 40. Prediction Horizon: 5.

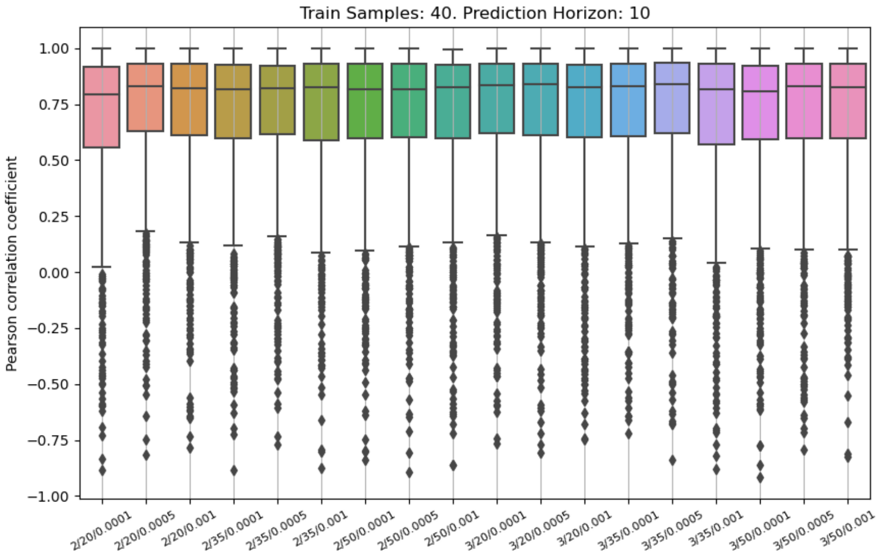


Figure H.12: Boxplots from prediction models. Train samples: 40. Prediction Horizon: 10.

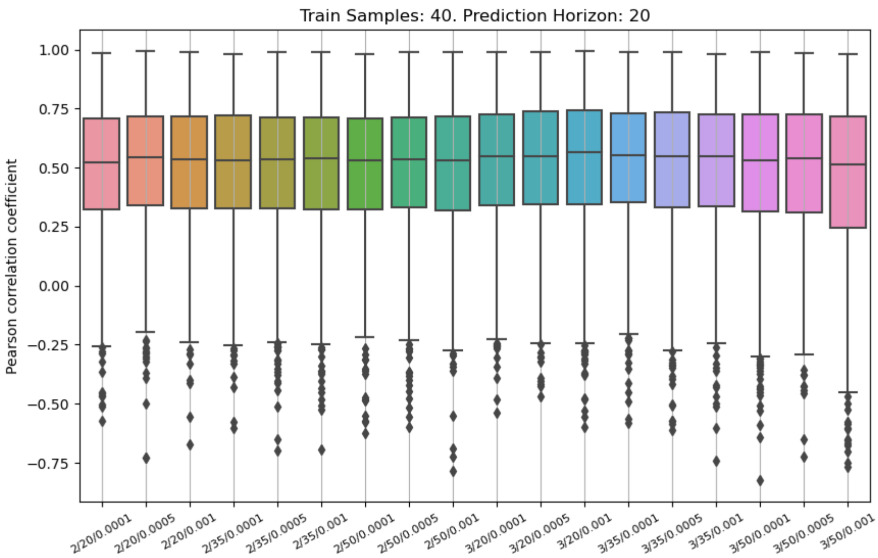


Figure H.13: Boxplots from prediction models. Train samples: 40. Prediction Horizon: 20.

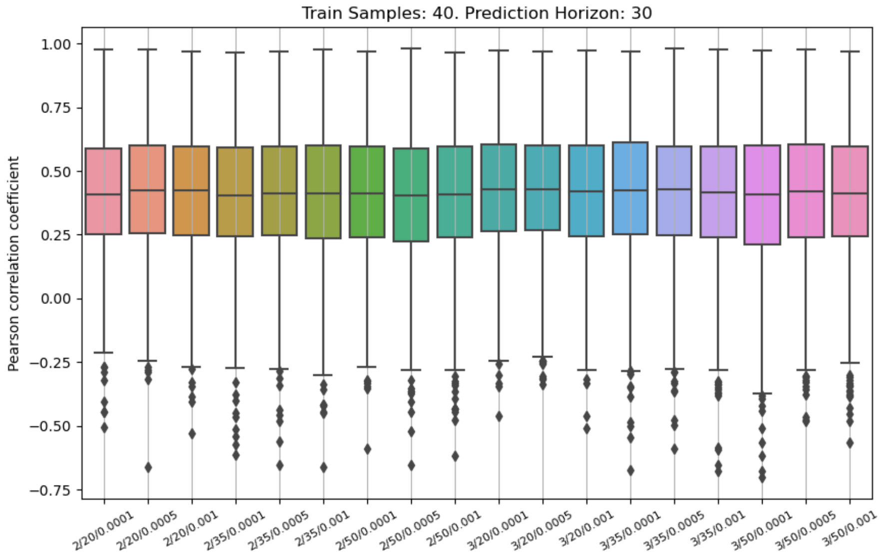


Figure H.14: Boxplots from prediction models. Train samples: 40. Prediction Horizon: 30.

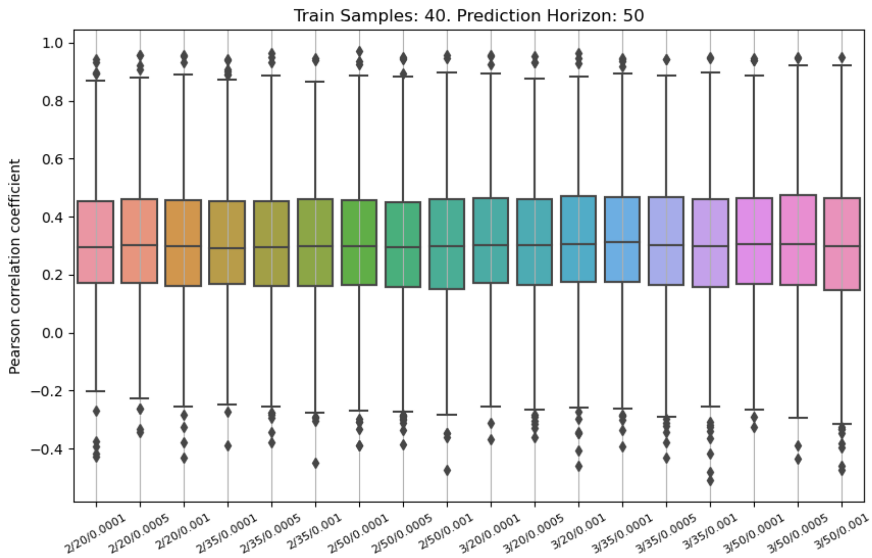


Figure H.15: Boxplots from prediction models. Train samples: 40. Prediction Horizon: 50.

H.4 50 train samples

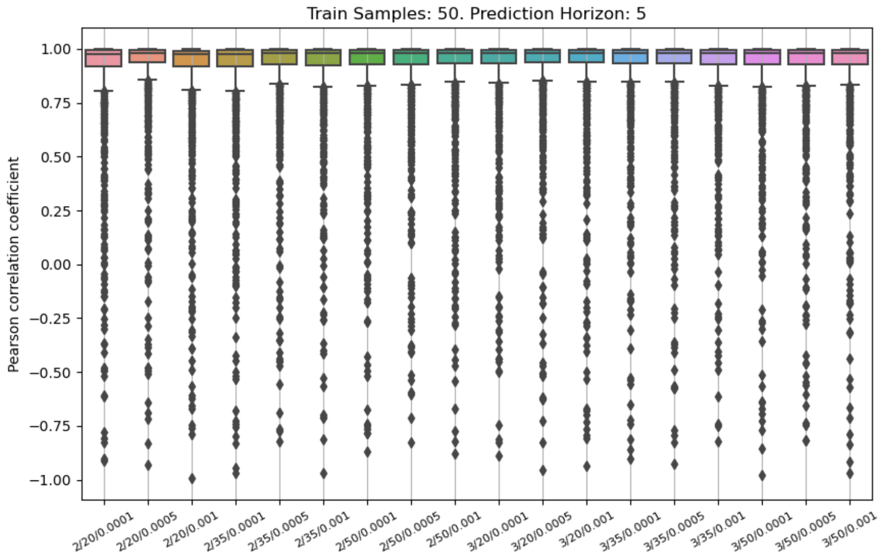


Figure H.16: Boxplots from prediction models. Train samples: 50. Prediction Horizon: 5.

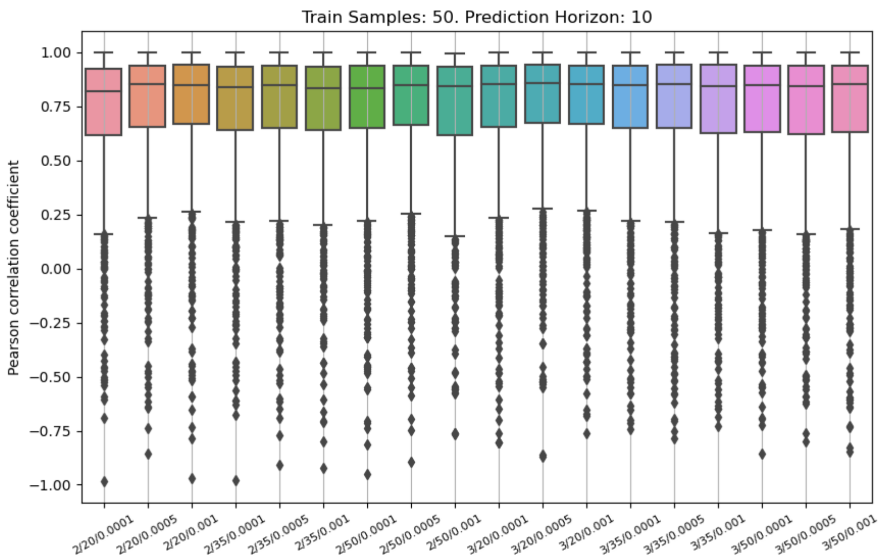


Figure H.17: Boxplots from prediction models. Train samples: 50. Prediction Horizon: 10.

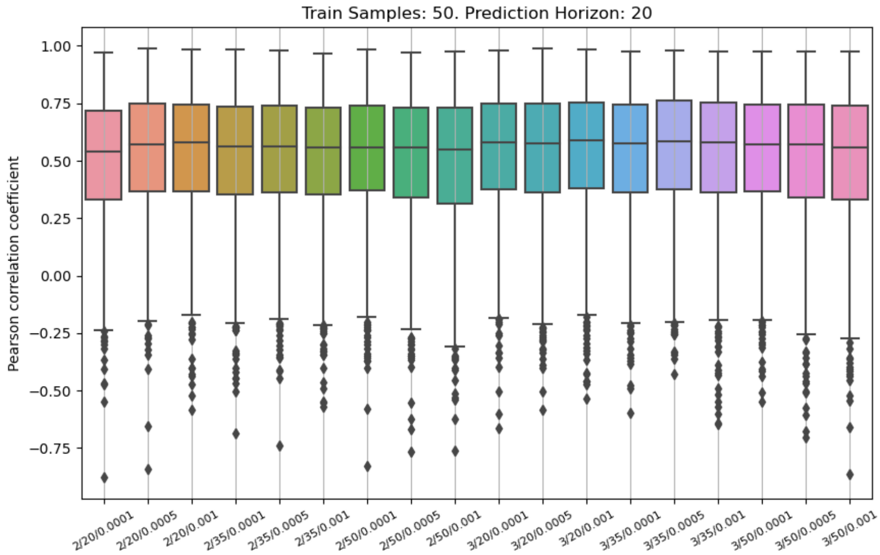


Figure H.18: Boxplots from prediction models. Train samples: 50. Prediction Horizon: 20.

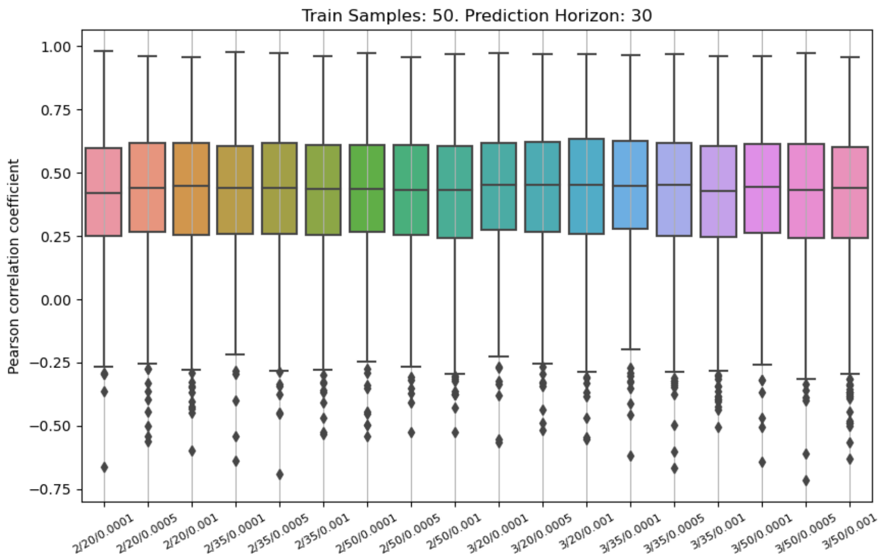


Figure H.19: Boxplots from prediction models. Train samples: 50. Prediction Horizon: 30.

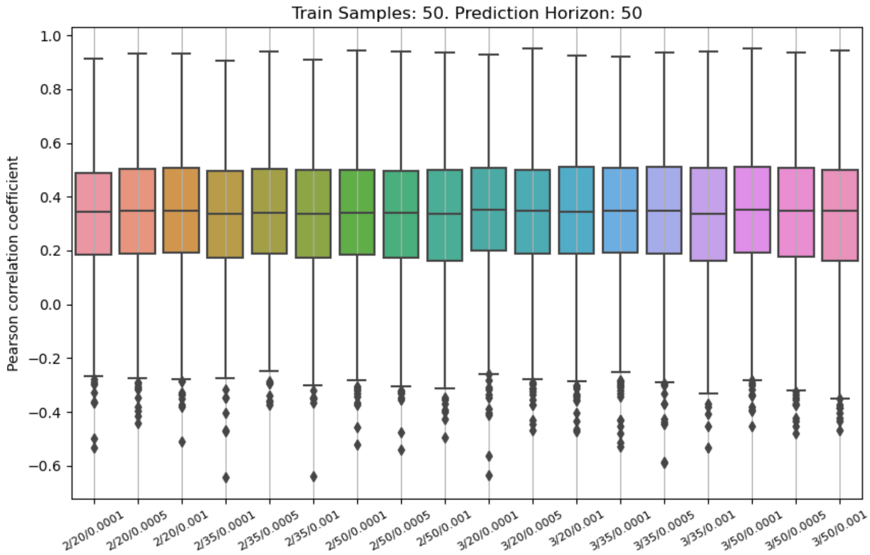


Figure H.20: Boxplots from prediction models. Train samples: 50. Prediction Horizon: 50.

Appendix I

Examples of predicted signals

I.1 20 train samples

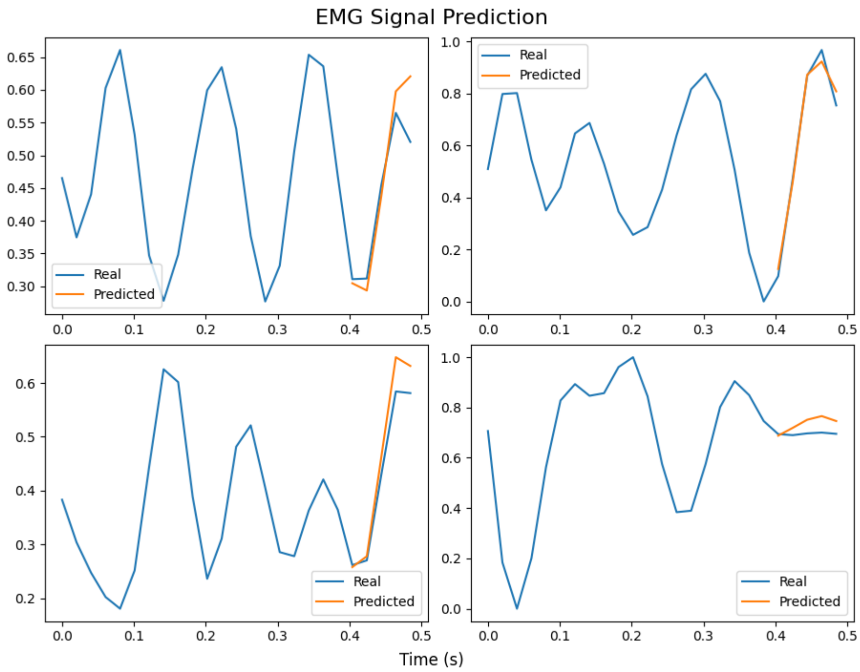


Figure I.1: Examples of predicted EMG signals. Train samples: 20. Prediction Horizon: 5.

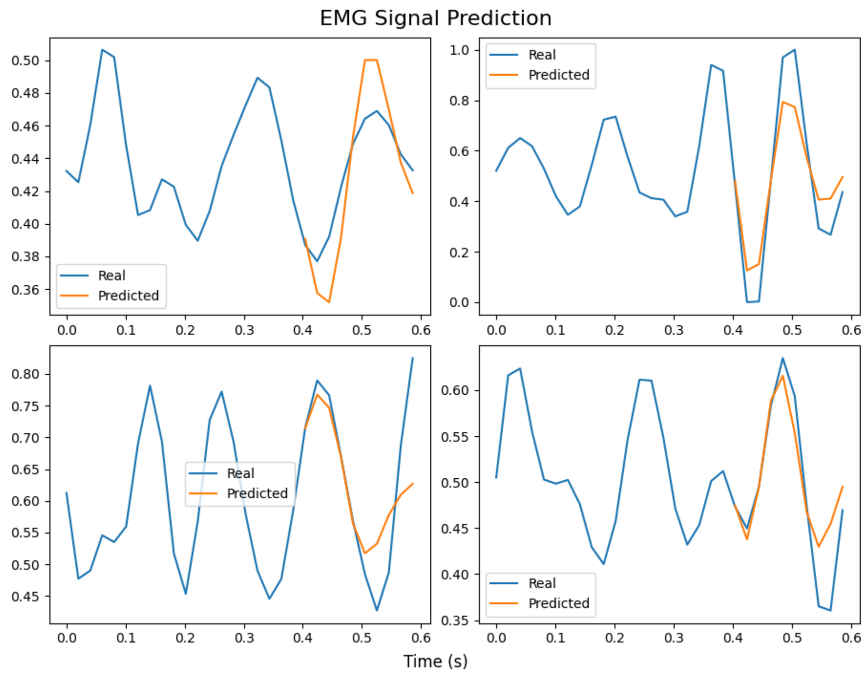


Figure I.2: Examples of predicted EMG signals. Train samples: 20. Prediction Horizon: 10.

I.2 30 train samples

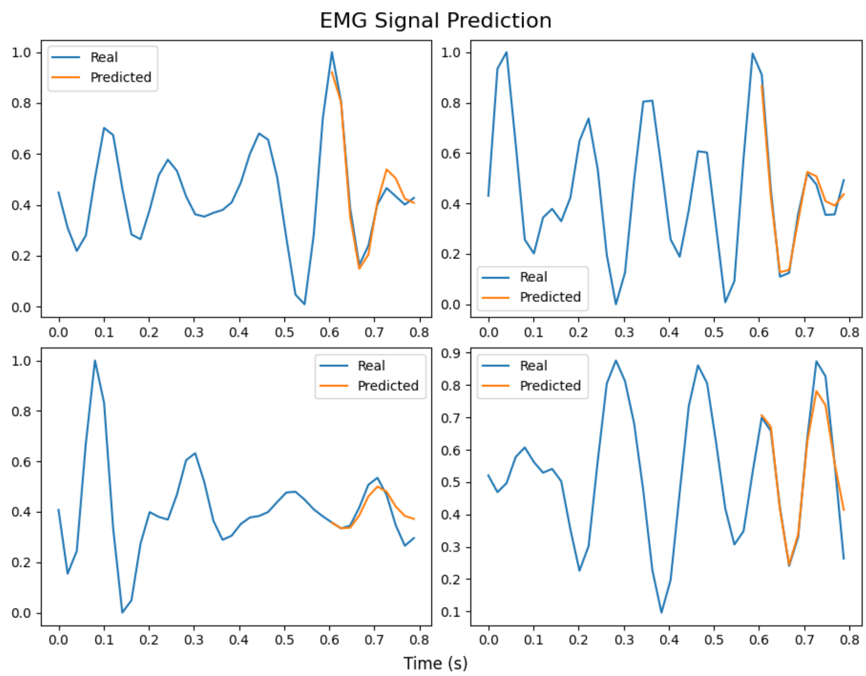


Figure I.3: Examples of predicted EMG signals. Train samples: 30. Prediction Horizon: 10.

I.3 40 train samples

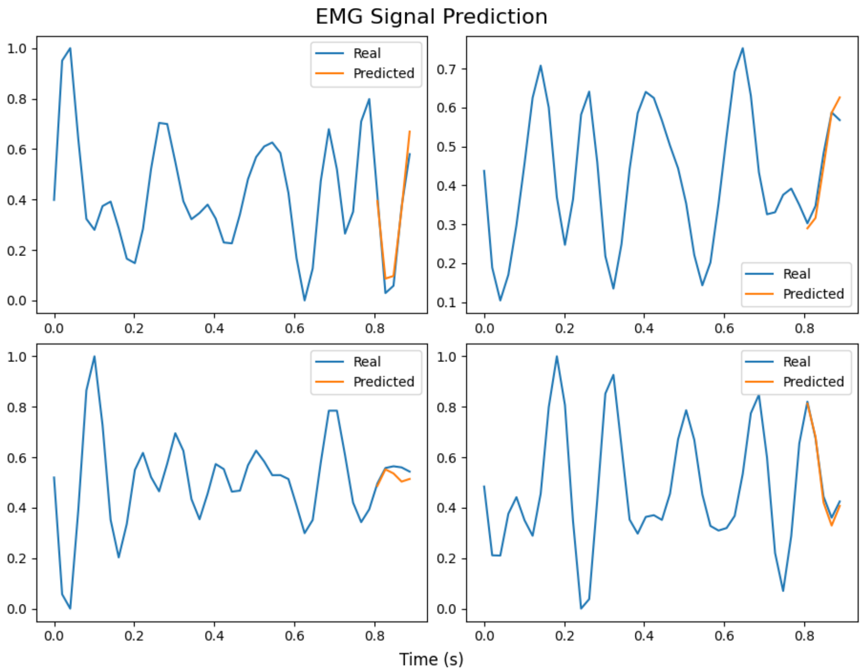


Figure I.4: Examples of predicted EMG signals. Train samples: 40. Prediction Horizon: 5.

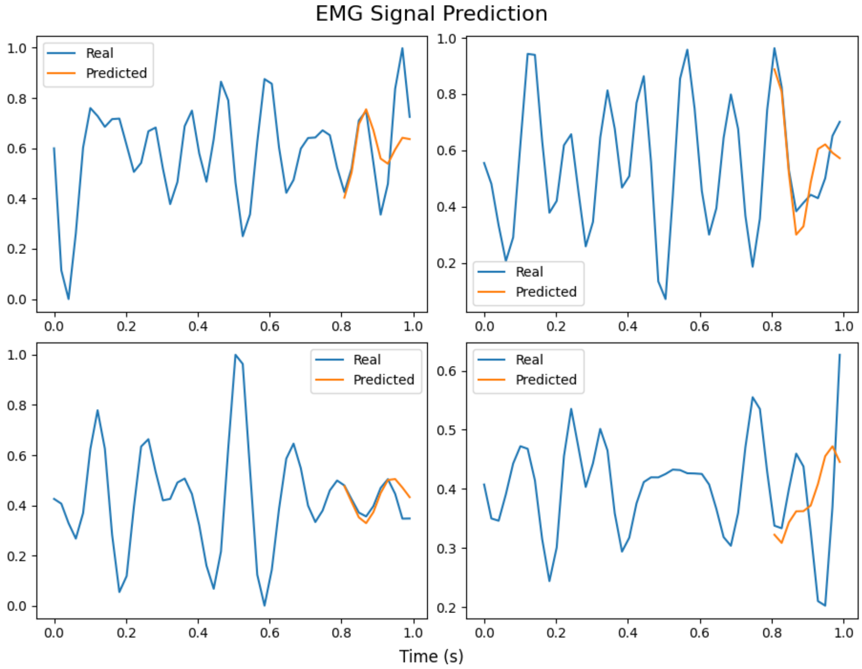


Figure I.5: Examples of predicted EMG signals. Train samples: 40. Prediction Horizon: 10.

I.4 50 train samples

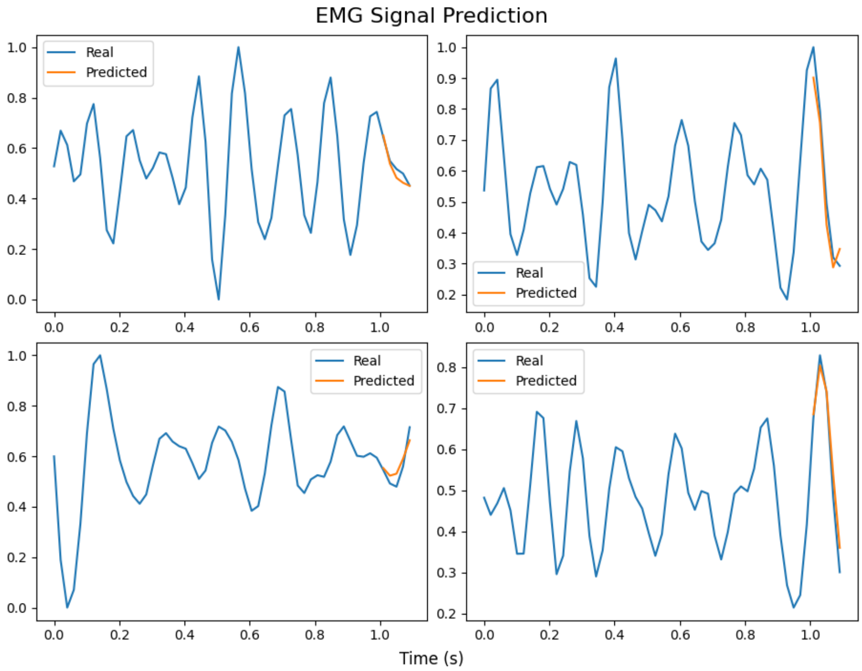


Figure I.6: Examples of predicted EMG signals. Train samples: 50. Prediction Horizon: 5.

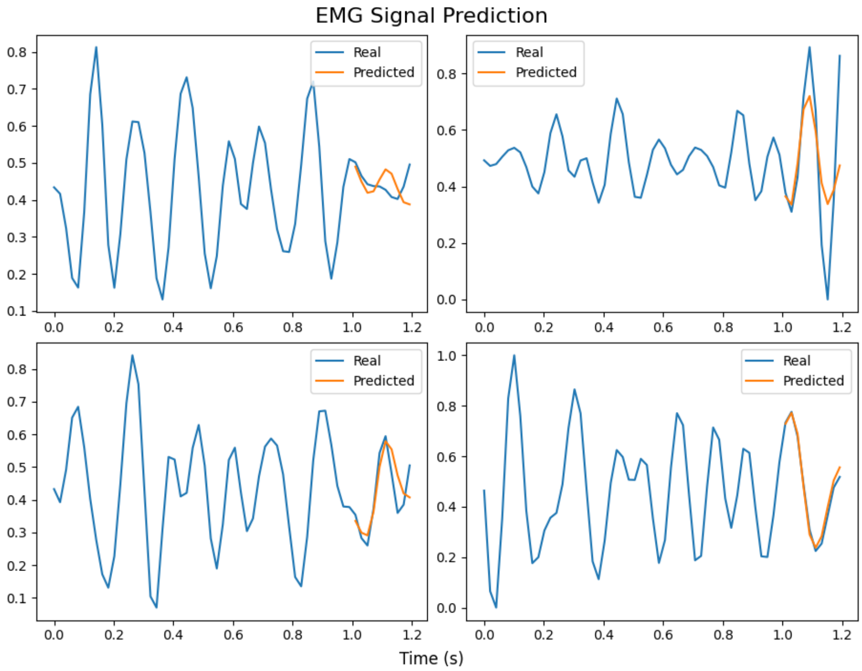


Figure I.7: Examples of predicted EMG signals. Train samples: 50. Prediction Horizon: 10.