

UNIVERSIDAD POLITÉCNICA DE MADRID

**ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN**



GRADO INGENIERÍA BIOMÉDICA

TRABAJO FIN DE GRADO

**DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS DE MACHINE
LEARNING PARA LA CARACTERIZACIÓN DE TEMBLOR PATOLÓGICO
MEDIANTE SEÑALES CINEMÁTICAS Y FISIOLÓGICAS**

VÍCTOR LOPO MARTÍNEZ

2021

UNIVERSIDAD POLITÉCNICA DE MADRID

ESCUELA TÉCNICA SUPERIOR
DE INGENIEROS DE TELECOMUNICACIÓN



GRADO INGENIERÍA BIOMÉDICA

TRABAJO FIN DE GRADO

DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS DE MACHINE
LEARNING PARA LA CARACTERIZACIÓN DE TEMBLOR PATOLÓGICO
MEDIANTE SEÑALES CINEMÁTICAS Y FISIOLÓGICAS

Autor

VÍCTOR LOPO MARTÍNEZ

Tutor

ALEJANDRO PASCUAL VALDUNCIEL

Co-tutor

ÁLVARO GUTIÉRREZ MARTÍN

Resumen

El temblor esencial es un trastorno del sistema nervioso que provoca la aparición de temblor patológico en alguna parte del cuerpo o extremidad pudiendo llegar a comprometer a la realización de actividades de la vida diaria del paciente. Actualmente, no existe ningún tratamiento eficaz y accesible para todos los pacientes contra el temblor esencial. Las nuevas líneas de investigación presentan la estimulación de las vías aferentes junto con su estrategia de estimulación *out-of-phase* como una alternativa barata e inocua capaz de superar los efectos secundarios y desventajas de los tratamientos más convencionales. La estrategia *out-of-phase* se basa en la aplicación de estimulaciones sobre un par de músculos antagonistas sincronizadas con la fase de actividad muscular del temblor.

Sin embargo, la predicción de varios ciclos de temblor basada en la frecuencia del mismo puede dar lugar a errores en la sincronización de la estimulación al ser las señales de temblor no estacionarias. Como alternativa, en este TFG se propone, explora y valida la utilización de algoritmos de *machine learning* tradicionales (KNN, SVM y Random Forest) y una red neuronal LSTM (Long Short Term Memory) para la tarea de clasificación, además de una red neuronal LSTM para la tarea de predicción. Para la evaluación de los distintos algoritmos de clasificación y predicción se crearon dos datasets con señales cinemáticas correspondientes a las variaciones temporales de los ángulos de flexo-extensión de la muñeca. Para el dataset de clasificación se utilizó un conjunto de 12 pacientes con temblor esencial y 5 sujetos sanos. Se etiquetaron los segmentos temporales en base al valor de su PSD dando un total de 18.000 segmentos. Para el dataset de predicción se utilizaron 7.000 segmentos temporales de pacientes con temblor esencial.

Los algoritmos de clasificación implementados en este TFG han permitido determinar la presencia de temblor, de manera que se estimule dentro de la estrategia *out-of-phase* únicamente cuando se presenta actividad tremorogénica. Todos los clasificadores implementados presentan exactitudes superiores al 90% en la tarea de clasificación binaria de señales cinemáticas de temblor y señales cinemáticas de no temblor. La red neuronal LSTM implementada es capaz de predecir los ciclos futuros del temblor de manera robusta y satisfactoria con un horizonte de predicción máximo de 1 segundo. Los valores de correlación de los primeros 100 ms son superiores al 0.9, y disminuyen a medida que aumenta el horizonte de predicción hasta 0.75 en la predicción futura de 1 segundo.

Se han cumplido todos los objetivos de las tareas de clasificación y predicción, y se ha demostrado que el uso de algoritmos de *machine learning* y *deep learning* pueden optimizar la estrategia *out-of-phase* en la estimulación de las vías aferentes, consiguiendo así una mayor supresión del temblor patológico.

Palabras clave: Supresión del temblor, predicción del temblor, clasificación del temblor, estimulación de las vías aferentes, *out-of-phase*, LSTM, PSD, SVM, KNN, Random Forest, temblor esencial.

Abstract

Essential Tremor is a disorder of the nervous system that causes pathological tremor in any part of the body or limb. It can affect to the realization of patient's daily living tasks. Currently, there is no effective and accessible treatment for all the patients with Essential Tremor. New lines of research present the stimulation of the afferent pathways together with its out-of-phase strategy as an inexpensive and harmless alternative, capable of overcoming the side effects and disadvantages of conventional treatments. The out-of-phase strategy is based on the application of stimulation pulses on a pair of antagonist muscles synchronized with the phase of the tremor activity.

However, the prediction of future tremor cycles based on the tremor frequency may lead to errors in the timing of the stimulation, as the tremor signals are non-stationary. As an alternative, this Thesis proposes, explores and validates the use of traditional machine learning algorithms (KNN, SVM and Random Forest) and an LSTM (Long Short Term Memory) neural network for the classification task, and a LSTM neural network for the prediction task. For the evaluation of the different classification and prediction algorithms, two datasets with kinematic signals corresponding to the time variations of the flexion/extension angles were created. A set of 12 patients with Essential Tremor and 5 healthy subjects was used for the classification dataset. Temporal sequences were labeled based on their PSD (Power Spectral Density) value, giving a total of 18,000 time sequences. 7,000 time sequences from patients with Essential Tremor were used for the prediction database.

The classification algorithms implemented in this Thesis have made it possible to determine the presence of tremor, so that it is stimulated within the out-of-phase strategy only when tremorogenic activity is present. All the implemented classifiers present accuracies higher than 90% in the binary classification task of tremor kinematic signals and non-tremor kinematic signals. The implemented LSTM neural network is capable of predicting future tremor cycles in a robust and satisfactory way, with a maximum prediction horizon of 1 second. The correlation values for the first 100 ms are greater than 0.9, and decrease as the prediction horizon increases, up to a 0.75 value for the 1 second future prediction.

All the objectives of the classification and prediction tasks have been achieved, and it has been demonstrated that the use of machine learning and deep learning algorithms can optimize the out-of-phase strategy in the stimulation of afferent pathways, so that it could be achieved a greater suppression of pathological tremor.

Keywords: Tremor suppression, tremor prediction, tremor classification, afferent pathways stimulation, out-of-phase, LSTM, PSD, SVM, KNN, Random Forest, Essential Tremor.

Agradecimientos

En primer lugar, quiero mostrar mi agradecimiento a mi madre, a mi padre y a mi hermano, por el apoyo incondicional que me han dado y por confiar siempre en mí, más incluso que yo mismo. Porque sin vosotros no estaría donde estoy.

En segundo lugar, me gustaría agradecer a mi tutor Alejandro Pascual por toda la ayuda, cooperación, oportunidades y sobre todo paciencia que ha mostrado a lo largo de todo este año, suponiendo un apoyo tanto a nivel académico como personal. También me gustaría darle las gracias a mi segundo tutor, Dr. Álvaro Gutiérrez, por todos los consejos e interés mostrado durante la realización de este proyecto.

Finalmente, agradecer a mis amigos de RDG por estar siempre aunque no siempre podamos estar, desde los botones rojo, azul y verde de las Gaunas. También a toda la gente que quiero del chami y de la uni, por vivir conmigo y acompañarme durante los mejores 4 años de mi vida.

A todos vosotros, gracias.

Índice general

Resumen	III
Abstract	IV
Agradecimientos	V
Índice General	VI
Índice de Figuras	X
Índice de Tablas	XIII
Lista de Acrónimos	XV
1. Introducción	1
1.1. Contexto	1
1.1.1. Tratamientos terapéuticos actuales	2
1.1.1.1. Medicación	2
1.1.1.2. Deep Brain Stimulation (DBS)	3
1.1.1.3. Talamotomías	4
1.1.1.4. Ultrasonidos Focalizados de Alta Intensidad (HIFU)	4
1.1.1.5. Estimulación Eléctrica Funcional (FES)	4
1.1.1.6. Estimulación de las Vías Aferentes	7
1.2. Motivación	9
1.3. Objetivos	10
1.4. Hipótesis	11

2. Estado del arte	12
2.1. Machine Learning	12
2.1.1. K-Nearest Neighbors (KNN)	12
2.1.2. Random Forest	13
2.1.3. Support Vector Machine (SVM)	15
2.2. Redes Neuronales Recurrentes	16
2.2.1. Introducción	16
2.2.2. Redes Neuronales Recurrentes	18
2.2.2.1. Back Propagation Through Time (BPTT)	19
2.2.2.2. Long Short Term Memory (LSTM)	20
2.3. Aplicación de algoritmos de ML y DL en señales de temblor patológico	22
3. Diseño y Metodología	24
3.1. Entorno de desarrollo	24
3.2. Dataset	25
3.2.1. Adquisición y descripción de los datos	25
3.2.2. Pre-procesamiento	26
3.2.3. Creación del dataset	26
3.3. Clasificación de señales cinemáticas de temblor	30
3.3.1. KNN	30
3.3.2. Random Forest	31
3.3.3. SVM	31
3.3.4. Red LSTM para clasificación	32
3.3.5. Métricas para evaluar la clasificación	33
3.4. Predicción de señales cinemáticas de temblor	34

3.4.1. Métricas para evaluar la predicción	36
4. Resultados	37
4.1. Clasificación de señales cinemáticas de temblor	37
4.1.1. Resultados KNN	37
4.1.2. Resultados Random Forest	39
4.1.3. Resultados SVM	40
4.1.4. Resumen de los resultados de los clasificadores tradicionales . .	42
4.1.5. Resultados LSTM de clasificación	42
4.2. Predicción de señales cinemáticas de temblor	44
4.3. Discusión	47
5. Conclusiones y Líneas Futuras	49
5.1. Conclusiones	49
5.2. Líneas futuras	50
Referencias	51
A. Aspectos éticos, económicos, sociales y ambientales	56
B. Presupuesto económico	57
C. Modelos de clasificación descartados	59
C.1. KNN	59
C.2. Random Forest	60
C.3. SVM	60
D. Modelos descartados de la LSTM de clasificación	61

E. Problema del desvanecimiento de gradientes en las RNNs	69
F. IMUs y su posicionamiento en la adquisición de datos	71
G. Ejemplos de predicción de señales tremorogénicas	73
H. Boxplots de los 180 modelos de predicción	79

Índice de figuras

1.1. Deep Brain Stimulation (DBS)	3
1.2. Estimulación Eléctrica Funcional (FES)	5
1.3. Estrategia de estimulación Out-Of-Phase	6
1.4. Ilustración simplificada de las fibras aferentes Ia involucradas en la flexo-extensión de una articulación para un par de músculos antagonistas	7
1.5. Estrategia SATS	9
2.1. Flujo de ejecución del Random Forest formado por 3 árboles de decisión	14
2.2. Ejemplo de modelo SVM para dos características	16
2.3. Estructura de la neurona simple	16
2.4. Ejemplos de funciones de activación	17
2.5. Gráfico computacional RNN de una capa.	18
2.6. Gráfico computacional de 3 celdas LSTM	20
3.1. Comparación de los dos tipos de señales cinemáticas del dataset	27
3.2. Comparación de la PSD de los dos tipos de señales cinemáticas del dataset	29
3.3. Arquitectura LSTM de clasificación	32
3.4. Arquitectura LSTM predicción	35
4.1. Boxplot comparativo de los modelos KNN	38
4.2. Matriz de confusión Modelo 2 KNN	39
4.3. Boxplot comparativo de los modelos Random Forest	40
4.4. Matriz de confusión del Modelo 5 Random Forest	40
4.5. Boxplot comparativo de los modelos SVM	41

4.6. Matriz de confusión SVM	42
4.7. Evolución Función de Pérdidas	43
4.8. Matriz de confusión LSTM	43
4.9. Ejemplos de predicciones	46
4.10. Boxplot con 50 muestras de entrenamiento / 5 de predicción	47
4.11. Boxplot con 50 muestras de entrenamiento / 50 de predicción	47
D.1. Función de pérdidas del modelo 2	61
D.2. Función de pérdidas del modelo 3	62
D.3. Función de pérdidas del modelo 4	63
D.4. Función de pérdidas del modelo 5	64
D.5. Función de pérdidas del modelo 6	65
D.6. Función de pérdidas del modelo 7	66
D.7. Función de pérdidas del modelo 8	67
D.8. Función de pérdidas del modelo 9	68
F.1. Inertial Measurement Unit (IMU)	71
F.2. IMUs y posición postural	72
G.1. Predicción con 30 muestras de entrenamiento / 5 de predicción	73
G.2. Predicción con 30 muestras de entrenamiento / 10 de predicción	74
G.3. Predicción con 30 muestras de entrenamiento / 20 de predicción	74
G.4. Predicción con 30 muestras de entrenamiento / 30 de predicción	75
G.5. Predicción con 30 muestras de entrenamiento / 50 de predicción	75
G.6. Predicción con 50 muestras de entrenamiento / 5 de predicción	76
G.7. Predicción con 50 muestras de entrenamiento / 10 de predicción	76

G.8. Predicción con 50 muestras de entrenamiento / 20 de predicción	77
G.9. Predicción con 50 muestras de entrenamiento / 30 de predicción	77
G.10. Predicción con 50 muestras de entrenamiento / 50 de predicción	78
H.1. Boxplot con 30 muestras de entrenamiento / 5 de predicción	79
H.2. Boxplot con 30 muestras de entrenamiento / 10 de predicción	80
H.3. Boxplot con 30 muestras de entrenamiento / 20 de predicción	80
H.4. Boxplot con 30 muestras de entrenamiento / 30 de predicción	81
H.5. Boxplot con 30 muestras de entrenamiento / 50 de predicción	81
H.6. Boxplot con 50 muestras de entrenamiento / 5 de predicción	82
H.7. Boxplot con 50 muestras de entrenamiento / 10 de predicción	82
H.8. Boxplot con 50 muestras de entrenamiento / 20 de predicción	83
H.9. Boxplot con 50 muestras de entrenamiento / 30 de predicción	83
H.10. Boxplot con 50 muestras de entrenamiento / 50 de predicción	84

Índice de tablas

1.1. Estudios FES Out-of-phase supresión del temblor	7
1.2. Estudios FES Co-contracción supresión del temblor	7
1.3. Estudios en la estimulación de las vías aferentes	8
4.1. Parámetros Modelo 2 KNN	37
4.2. Métricas KNN	38
4.3. Parámetros Modelo 5 Random Forest	39
4.4. Métricas Random Forest	39
4.5. Parámetros Modelo 9 SVM	41
4.6. Métricas SVM	41
4.7. Métricas LSTM	44
4.8. Mejores modelos de predicción	46
B.1. Costes de personal.	57
B.2. Costes de recursos materiales.	57
B.3. Costes totales.	58
C.1. Modelos descartados del KNN	59
C.2. Modelos descartados Random Forest	60
C.3. Modelos descartados de la SVM	60
D.1. Matriz de confusión Modelo 2	61
D.2. Matriz de confusión Modelo 3	62
D.3. Matriz de confusión Modelo 4	62
D.4. Matriz de confusión Modelo 5	63

D.5. Matriz de confusión Modelo 6	64
D.6. Matriz de confusión Modelo 7	65
D.7. Matriz de confusión Modelo 8	66
D.8. Matriz de confusión Modelo 9	67

Lista de Acrónimos

EP: Enfermedad de Parkinson

TE: Temblor Esencial

FES: Estimulación Eléctrica Funcional

DBS: Deep Brain Stimulation

HIFU: Ultrasonidos Focalizados de Alta Intensidad

KNN: K - Nearest Neighbors

SVM: Support Vector Machine

UPDRS: Escala Unificada de Calificación de la Enfermedad de Parkinson

LSTM: Long Short Term Memory

RNN: Redes Neuronales Recurrentes

BPTT: Back Propagation Through Time

ANN: Redes Neuronales Artificiales

MSE: Error Cuadrático Medio

HIFU: Ultrasonidos Focalizados de Alta Intensidad

VIM: Ventralis Intermedius

GK: Gamma Knife

ML: Machine Learning

DL: Deep Learning

IG: Ganancia de Información

MLP: Perceptrón Multicapa

IMU: Unidad de Medida Inercial

RMSE: Raíz del Error Cuadrático Medio

GPU: Graphical Processing Unit

CPU: Central Processing Unit

CUDA: Compute Unified Device Architecture

IMU: Inertial Measurement Unit

PSD: Densidad Espectral de Potencia

Capítulo 1

Introducción

El temblor se define como el movimiento involuntario de una parte del cuerpo o extremidad como consecuencia de oscilaciones neurales patológicas que se proyectan hacia los músculos, provocando la contracción alternante o sincrónica de músculos agonistas y antagonistas [1] [2]. El temblor se hace más evidente en las extremidades superiores, sobre todo en los movimientos de flexo-extensión de la muñeca. Las dos enfermedades más comunes que producen temblor patológico son el Temblor Esencial (TE) y la Enfermedad de Parkinson (EP). El temblor postural y el temblor cinético son más comunes dentro del TE mientras que el temblor en reposo se asocia a los pacientes con Parkinson. El TE se caracteriza por tener un rango de frecuencias que oscila entre 4 y 12 Hz, pudiendo variar en intensidad y con la edad [3]. Se estima que alrededor del 4% de la población por encima de los 65 años padece TE [4] y el 1% de la población por encima de los 60 sufre el trastorno del Parkinson [5].

1.1. Contexto

No hay consenso sobre el origen del TE. La teoría predominante es que se genera como consecuencia de circuitos cerebelo-tálamo-corticales anormales que derivan en oscilaciones patológicas en las neuronas motoras espinales que inervan los músculos afectados [3]. También se ha sugerido que la aparición de temblor es el resultado de la interacción entre los circuitos efectores supraespinales y las vías aferentes [6]. Por su parte, la EP es un desorden neurológico que actúa de manera progresiva, cuyo origen está en la neurodegeneración o pérdida de las neuronas de la sustancia nigra (estructura situada en el mesencéfalo). Esta degeneración se traduce en una falta de dopamina en el organismo, que es la encargada de transmitir la información necesaria para la correcta realización de los movimientos [7].

Más del 65% de la población que padece temblor patológico en las extremidades superiores se ve condicionada en la realización de actividades de la vida diaria [4]. Además de la incapacidad para realizar estas actividades, los pacientes sufren de alteraciones psicológicas que pueden derivar en exclusión social. El 34% de los pacientes con TE presenta depresión [8].

A continuación se muestran diferentes tipos de temblor [7]:

- **Temblor en reposo:** Se asocia de forma común a los pacientes con Parkinson y está presente cuando una extremidad está totalmente soportada en contra de la gravedad, con una frecuencia característica entre 3 y 6 Hz.
- **Temblor postural:** El temblor postural es característico de la patología de

TE. Es máximo cuando se mantiene una extremidad en una posición fija contra la gravedad. Presenta una frecuencia de 5 a 8 Hz.

- **Temblo de acción:** El temblor de acción es máximo cuando una extremidad se mueve de forma voluntaria. Ocurre en frecuencias diferentes pero siempre con valores menores de 13 Hz.
- **Temblo fisiológico:** No sólo los pacientes enfermos presentan temblor. Los sujetos sanos presentan un temblor fisiológico (que no es considerado patológico) en todas las articulaciones y músculos, siempre y cuando se encuentren en una posición en la que sea posible oscilar. Su frecuencia característica está entre los 8 y los 12 Hz variando su amplitud en función de la persona.

1.1.1. Tratamientos terapéuticos actuales

Existen distintos enfoques en el tratamiento del temblor patológico necesarios de revisar para comprender la importancia de la búsqueda de soluciones alternativas [9] [10].

1.1.1.1. Medicación

El enfoque más extendido dentro del tratamiento del temblor son los medicamentos [11], siendo un tratamiento efectivo únicamente para el 50% de los pacientes que lo sufren [12]. La medicación no es una cura, sino una manera de aliviar los síntomas, capaz de reducir el temblor un 50% con respecto al temblor total. Los efectos secundarios ligados al tratamiento con medicamentos son comunes llegando a provocar el abandono de este tipo de tratamientos en 1/3 de las ocasiones [13].

Para pacientes con TE el propranolol y la primidona son los fármacos de primera línea, reservando el resto de fármacos para casos en los que estos primeros carecen de eficacia o sus efectos secundarios dificultan su uso [13].

En el caso de los pacientes con EP la mayoría de los fármacos utilizados están orientados a reestablecer los niveles de dopamina mediante la administración de levodopa (precursor de la dopamina) y más recientemente con el uso de agonistas dopaminérgicos [7].

Estos tratamientos aportan una mejora sintomática y un aumento de la calidad de vida de al menos 5-8 años, periodo tras el cual los pacientes suelen desarrollar complicaciones motoras (fluctuaciones y discinesias) y neuropsiquiátricas, llegando a aparecer síntomas de alteraciones cognitivas importantes [14].

1.1.1.2. Deep Brain Stimulation (DBS)

En los casos en los que la medicación es insuficiente o sus efectos secundarios excesivos, una alternativa es la estimulación cerebral profunda (DBS) de los ganglios basales [15].

El DBS es un tratamiento quirúrgico no destructivo que emplea pulsos eléctricos para modular la actividad motora tremorogénica. La eficacia del DBS se atribuye a su capacidad para detener la propagación del temblor en el circuito cerebelo-tálamo-cortical interrumpiendo la actividad oscilatoria del tálamo [16].

Es un tratamiento alternativo a la medicación utilizado en el caso de que los medicamentos no hayan sido capaces de paliar los síntomas. La gran desventaja del DBS es su bajo criterio de elegibilidad y los estrictos requisitos que se deben cumplir para que el paciente pueda ser considerado apto para el procedimiento. Todo ello supone un bajo porcentaje de pacientes elegibles para esta técnica de reducción del temblor patológico [15].

Pese a ello, es una de las mejores técnicas para la supresión del temblor, pudiendo alcanzar porcentajes de mejora del 50% en términos de habla, rigidez, temblor y bradicinesia, además de ser una técnica reversible a diferencia de las talamotomías (ver Subsubsección 1.1.1.3) y la técnica HIFU (ver Subsubsección 1.1.1.4) [17].

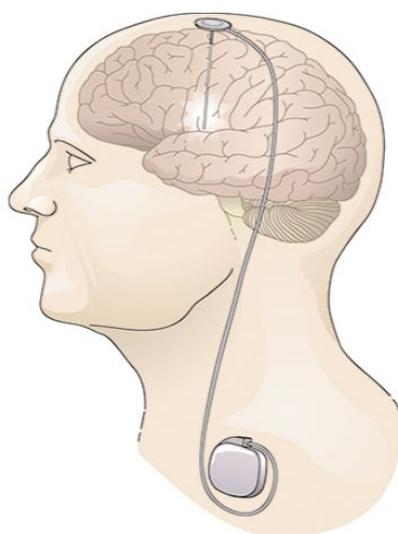


Figura 1.1: Deep Brain Stimulation (DBS)
[18]

Además del bajo criterio de elegibilidad, otros inconvenientes de esta técnica son los eventos adversos intraoperatorios como hipotensión o convulsiones, hemorragias intracerebrales sintomáticas, o ictus isquémicos derivando en una disminución de la conciencia. Las complicaciones a largo plazo del DBS pueden incluir molestias con el implante (ver Figura 1.1), infecciones de la herida, malposición, fractura de algún componente o pérdida de efecto. Generalmente estos síntomas suceden en el 2% de

los casos [17].

1.1.1.3. Talamotomías

La talamotomía por radiofrecuencia consiste en destruir núcleos del tálamo (generalmente el Ventralis Intermedius (VIM)) por medio de ablación por radiofrecuencia de manera interna a través de un orificio [19]. La talamotomía por radiofrecuencia consigue resultados de reducción del temblor patológico en un rango del 74-90% respecto al temblor total [20]. Pese a que los efectos son inmediatos, presenta desventajas como la imposibilidad de confirmar el sitio de la lesión intraoperativamente con neurofisiología, el efecto clínico no inmediato del procedimiento (los resultados aparecen de 1 a 6 meses después de la operación) y la destrucción de parte del tálamo cerebral de manera irreversible [21].

La talamotomía con Gamma Knife (GK) sobre el VIM es una opción neuroquirúrgica mínimamente invasiva para la reducción del temblor patológico con mejoras significativas y una reducción del temblor del 63% pasados 12 meses [22]. Se emplea en aquellos pacientes que no pueden realizar la talamotomía por radiofrecuencia debido a comorbilidades médicas. La principal diferencia con la talamotomía por radiofrecuencia es que se aplica radiación de manera externa sobre el VIM, sin necesidad de realizar un orificio. Los efectos secundarios incluyen hemiparesia, parestesia, disfasia, disfagia y déficit neurológico retardado. Además supone la destrucción total e irreversible del VIM [20].

1.1.1.4. Ultrasonidos Focalizados de Alta Intensidad (HIFU)

La técnica HIFU es la más novedosa en el tratamiento del temblor [23]. Es un procedimiento quirúrgico mínimamente invasivo para el tratamiento sintomático del temblor patológico. Utiliza los ultrasonidos, lo que es una ventaja frente a las radiofrecuencias al no ser radiaciones ionizantes evitando así los efectos adversos asociados. Presenta una resolución espacial y precisión superior a las talamotomías [24]. Tras el tratamiento con la técnica HIFU los pacientes pueden llegar a experimentar una mejora en el temblor desde el momento del tratamiento hasta 3 meses después en un 62% de los casos [23] y de hasta el 80% pasados los 12 meses [25]. Los efectos secundarios más frecuentes son parestesia, ataxia, inestabilidad de la marcha y, con menor frecuencia, ictus isquémico, disestesia de los dedos y trombosis venosa profunda. Se trata de una técnica irreversible [20].

1.1.1.5. Estimulación Eléctrica Funcional (FES)

La FES surge como una alternativa en el tratamiento de la supresión del temblor a las técnicas mostradas en apartados anteriores. Puede aplicarse a través de electrodos superficiales (no invasivo) o mediante electrodos intramusculares (mínimamente invasivo) [26].

Nace como una posibilidad prometedora y fácilmente aplicable. En la FES, los electrodos se sitúan sobre los músculos que van a ser activados (ver Figura 1.2) y los pulsos de corriente eléctrica de bajo nivel, por encima del umbral motor de las motoneuronas, se aplican sobre el tejido, despolarizando los nervios motores y provocando las contracciones del músculo [27] [28] [29] [30].

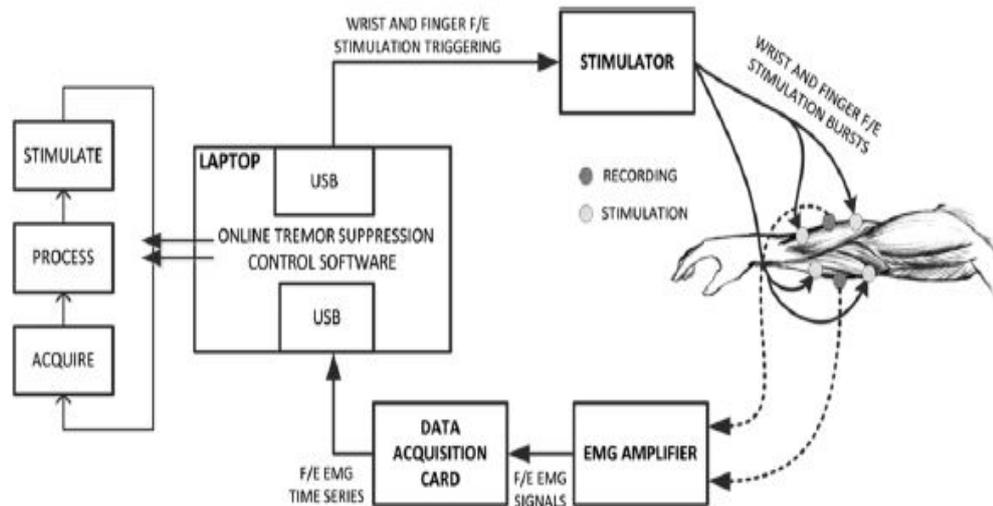


Figura 1.2: Estimulación Eléctrica Funcional (FES)
[26]

Dentro de la técnica de FES pueden ser aplicadas dos tipos de estrategias:

La estrategia FES *out-of-phase* se aplica la estimulación sobre los músculos para generar fuerzas opuestas a aquellas realizadas por las activaciones tremorogénicas detectadas mediante medidas cinemáticas o a través de señales de electromiografía (EMG).

Cuando esta estrategia es controlada mediante señales de EMG (ver Figura 1.3) se calcula la envolvente de los picos del EMG tanto en el músculo flexor como en el extensor [26]. El músculo flexor y extensor son antagonistas, de manera que cuando el flexor se activa el extensor se relaja y viceversa. Posteriormente se calculan los ciclos y la frecuencia del temblor dictados por los picos de los pulsos tremorogénicos en la envolvente del EMG. Una vez extraído el periodo del temblor (τ) tanto en el músculo flexor como en extensor, se predice el momento ($t + n\tau$) en el que se realizarán las futuras estimulaciones (siendo n el número de estimulación futura). Esta dinámica seguida se debe a que no es posible detectar el EMG durante los periodos de estimulación consecuencia de los artefactos introducidos en la señal por la propia estimulación. La desventaja de la estrategia *out-of-phase* es su limitada capacidad predictiva. El temblor patológico no es estacionario y una desincronización de los ciclos de temblor del orden de *ms* puede provocar que los pulsos de estimulación estimados no estén alineados con la actividad fisiológica, reduciéndose así la eficacia de la intervención.

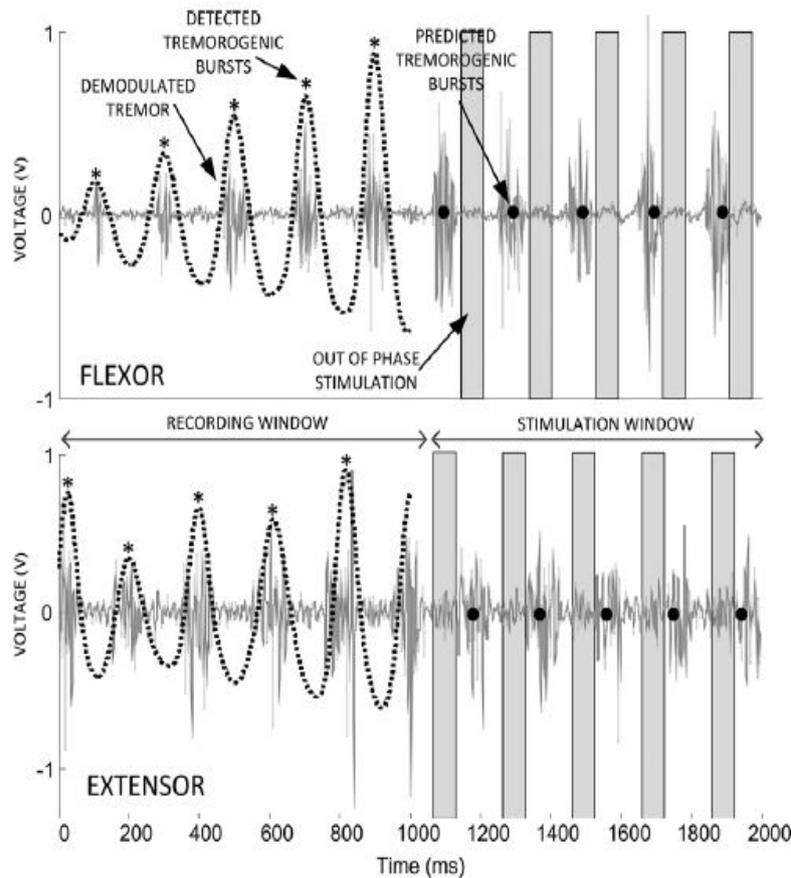


Figura 1.3: Estrategia de estimulación Out-Of-Phase [26]

La otra estrategia de estimulación es la *co-contracción*. En la *co-contracción* se estimulan de manera simultánea los músculos antagonistas que intervienen en la actividad tremorogénica de una determinada articulación. Se consigue aumentar la rigidez de la articulación por medio de la co-activación para disminuir la manifestación mecánica de las oscilaciones. La desventaja de la *co-contracción* es la reducción del temblor en base al aumento de rigidez. Este aumento de rigidez supone una molestia y una limitación de los movimientos del paciente [31] [32] [33].

La FES es una de las técnicas clásicas de estimulación neuromuscular en las cuales la intensidad del pulso supera el umbral motor de las motoneuronas provocando la contracción del músculo inervado. Este enfoque tiene desventajas como la fatiga producida en los músculos estimulados, la incomodidad de la alta intensidad de la estimulación y la interferencia con los movimientos voluntarios [34] [35].

Dentro de las dos técnicas de FES presentadas, la estrategia *out-of-phase* (ver Tabla 1.1) es capaz de lograr una supresión del temblor mayor que la estrategia de *co-contracción* (ver Tabla 1.2). Además, aumenta la comodidad del paciente durante los periodos de estimulación provocando menos fatiga y menor rigidez en el músculo [29].

Referencia	Pacientes	Articulación	Estrategia	Supresión
<i>Javidan et al., 1992</i>	3 TE, 4 EP	Muñeca	Out-of-phase	53 ±25 %
<i>Gilard et al., 1999</i>	3 EP	Muñeca/Dedo	Out-of-phase	83 ±2 %
<i>Popovic Maneski et al., 2011</i>	3 TE, 4 EP	Muñeca	Out-of-phase	67 ±13 %
<i>Widjaja et al., 2011</i>	1 TE	Muñeca	Out-of-phase	57 %
<i>Dosen et al., 2015</i>	2 TE, 4 EP	Muñeca	Out-of-phase	60 ±14 %

Tabla 1.1: Estudios FES Out-of-phase supresión del temblor

Referencia	Pacientes	Articulación	Estrategia	Supresión
<i>Grimaldi et al., 1999</i>	1 TE, 2 EP	Muñeca/Hombro	Co-contracción	9 ±35 %
<i>Gallego et al., 2013</i>	2 TE, 2 EP	Muñeca	Co-contracción	52 ±25 %
<i>Jitkrisadukul et al., 2015</i>	34 EP	Muñeca	Co-contracción	44 ±33 %

Tabla 1.2: Estudios FES Co-contracción supresión del temblor

1.1.1.6. Estimulación de las Vías Aferentes

La estimulación neuromuscular de las vías aferentes por debajo del umbral motor nace como una solución alternativa y prometedora en la supresión del temblor. Se estimulan las vías aferentes para producir una modulación a nivel espinal a través de las interneuronas y los circuitos reflejos. Es una técnica accesible a todos los pacientes, inocua, barata y sin efectos adversos conocidos [36].

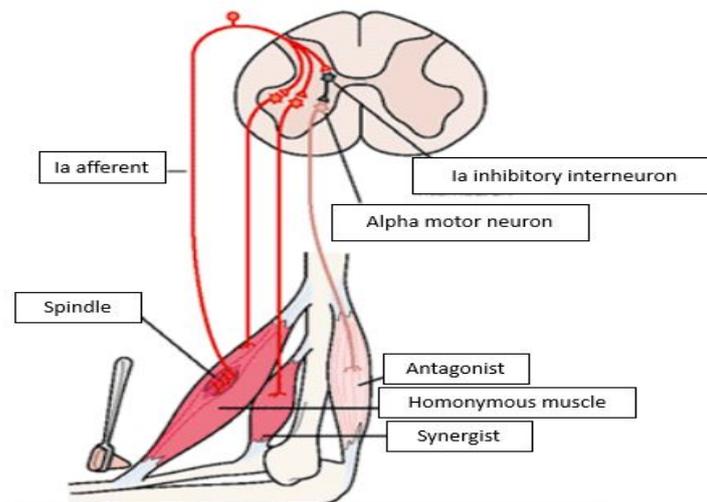


Figura 1.4: Ilustración simplificada de las fibras aferentes Ia involucradas en la flexo-extensión de una articulación para un par de músculos antagonistas [37]

La activación de las vías aferentes por debajo del umbral motor puede lograr una supresión del temblor significativa con un valor promedio del 52 % [38]. Sin embargo, la variabilidad es alta entre los pacientes que consiguen una buena respuesta y aquellos que apenas responden a la estimulación. Se consiguen solucionar los problemas de

fatiga del músculo y la interferencia con los movimientos voluntarios causados por la estrategia FES.

Las vías aferentes conocidas que pueden estar involucradas en la generación o amplificación del temblor y por tanto pueden ser tratadas en el tratamiento de reducción del temblor son las Ia y las Ib. Las vías aferentes pueden estimularse desde la superficie o a través de electrodos intramusculares [39]. En la Tabla 1.3 se resumen los principales estudios llevados a cabo con el objetivo de disminuir el temblor en las extremidades superiores por medio de la modulación de las vías aferentes.

La Tabla 1.3 muestra resultados prometedores y sugiere que la estimulación de las vías aferentes puede suponer una alternativa a los tratamientos de temblor más tradicionales. La heterogeneidad de los resultados se debe a diversos factores como la inclusión de pacientes pertenecientes a diferentes grupos de enfermedades (TE y EP), o los diferentes protocolos empleados en términos de intensidad, estrategia de estimulación y la situación de los electrodos (superficiales e intramusculares).

Una estrategia alternativa al *out-of-phase* y a la *co-contracción* es la Selective and Adaptive Timely Stimulation (SATS). La técnica SATS busca solucionar el problema del *out-of-phase* y conseguir que los pulsos de estimulación no caigan fuera de fase mediante el uso de ventanas secuenciales de lectura y estimulación [39]. En la ventana de lectura se mide la actividad del EMG y si supera un cierto umbral se aplica la estimulación (ver Figura 1.5).

Referencia	Pacientes	Estrategia	Vías	Supresión
<i>Dosen et al., 2015</i>	2 TE, 4 EP	Estimulación out-of-phase sobre el músculo antagonista	Ia	57+6 %
<i>Dideriksen et al., 2017</i>	5 TE, 4 EP	Estimulación superficial e intramuscular en estrategia out-of-phase	Ia	52 %
<i>Muceli et al., 2019</i>	2 TE, 4 EP	Estimulación intramuscular en estrategia out-of-phase	Ia	58 %
<i>Pascual-Valdunciel et al., 2020</i>	9 TE	Estimulación intramuscular con SATS	Ia	32 %

Tabla 1.3: Estudios en la estimulación de las vías aferentes

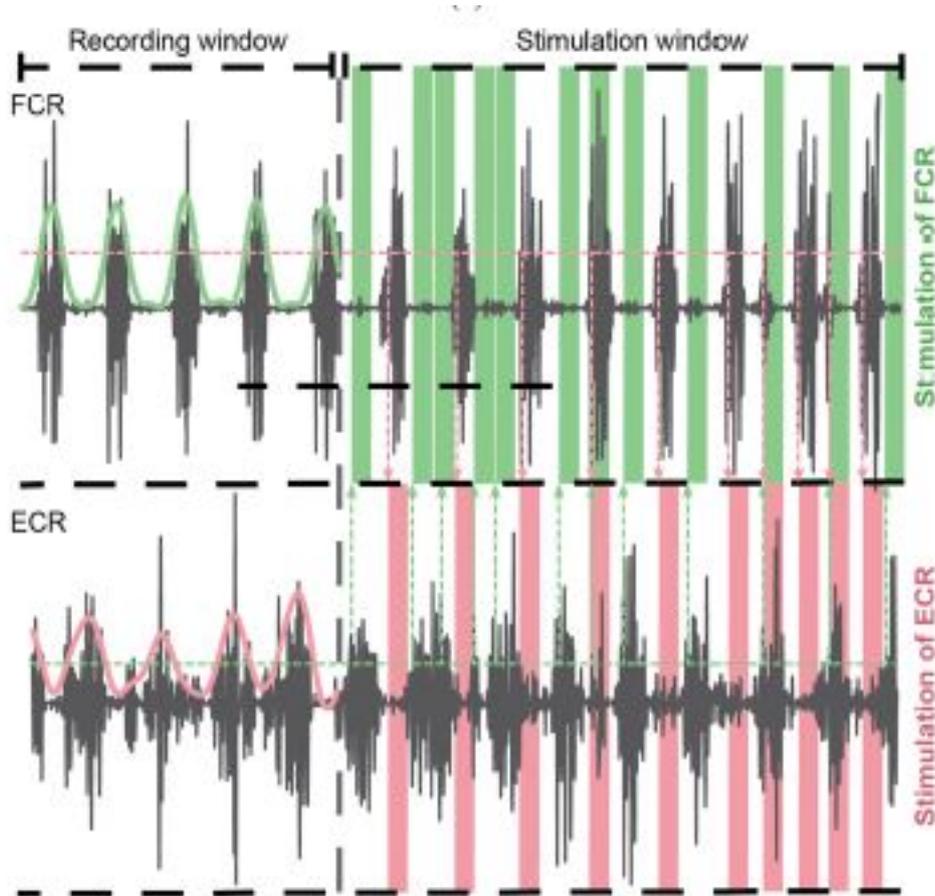


Figura 1.5: Estrategia SATS
[39]

1.2. Motivación

El tratamiento del temblor patológico está enfocado mayoritariamente en la administración de fármacos. Las nuevas líneas de investigación y el desarrollo de nuevas tecnologías junto con el procesamiento de datos abren las puertas a la búsqueda de soluciones vanguardistas. Una de las técnicas de tratamiento más prometedoras en los estudios de la última década es la Estimulación Eléctrica Funcional (FES). Sin embargo, los problemas de fatiga del músculo y rigidez provocados por la estimulación por encima del umbral motor dieron paso a la estimulación de las vías aferentes. Dentro de las estrategias de estimulación presentadas en el estado del arte se encuentran la *out-of-phase* [26], *co-contracción* [31] [32] [33] y *SATS* [39].

La *co-contracción* presenta el problema de la rigidez e interferencia con los movimientos voluntarios en base a la co-activación de los músculos antagonistas en el movimiento de flexo-extensión. La estrategia *out-of-phase* supera los efectos adversos de la *co-contracción*, sin embargo, la predicción de varios ciclos de temblor basada en la frecuencia del mismo puede dar lugar a errores en la sincronización de la

estimulación. La técnica *SATS* basada en el uso de ventanas secuenciales de lectura y estimulación mejora el funcionamiento teórico del *out-of-phase*, ya que prescinde de la predicción. No obstante, está limitada por la resolución temporal, ya que existe una latencia entre la lectura del EMG, la detección de actividad temblorosa y la consecuente estimulación. La magnitud de esta latencia, aún siendo del orden de milisegundos puede provocar que la estimulación no esté óptimamente sincronizada con la actividad fisiológica. Además, dicha estrategia hasta ahora ha sido testada en condiciones de laboratorio en temblor postural o de reposo, donde no existe contribución de movimientos voluntarios y por lo tanto las señales de EMG de la banda de temblor tienen una elevada relación señal/ruido y su procesamiento es sencillo.

La implementación de algoritmos de clasificación permite detectar la actividad tremorogénica durante la ventana de lectura de la estrategia *out-of-phase* de manera similar al ojo clínico. Por su parte, los algoritmos de predicción posibilitan la predicción de la frecuencia y amplitud de los próximos ciclos de temblor de manera robusta, a diferencia de las técnicas presentadas en el párrafo anterior. Los problemas de clasificación y predicción pueden abordarse con un procesamiento de señal basado en *machine learning* y redes neuronales, superando las limitaciones de los algoritmos empleados hasta la fecha y mejorando potencialmente la aplicación de las estrategias de control de la estimulación de las vías aferentes para la supresión del temblor.

1.3. Objetivos

Los objetivos de este Trabajo de Fin de Grado se van a dividir en tres áreas. Por un lado el estudio del estado del arte e investigaciones llevadas a cabo dentro del procesamiento de señales cinemáticas o de EMG de temblor patológico para diagnóstico o predicción; el diseño e implementación de los algoritmos de clasificación de ML para la caracterización objetiva del temblor; y por último el diseño e implementación de una red neuronal LSTM (Long Short Term Memory) para la predicción futura de los ciclos de temblor en pacientes con TE.

El objetivo de clasificación es que tanto los clasificadores tradicionales como las redes neuronales LSTM sean capaces de clasificar y distinguir entre señales cinemáticas de temblor y de no temblor. Se trata por tanto de un problema de clasificación binaria en el que se buscarán los parámetros óptimos para mejorar el rendimiento de cada modelo. Todo ello con un preprocesado mínimo de la señal y sin extracción de características. Supone un enfoque diferente a los estudios existentes en la actualidad basados en la extracción de características. Se consigue así una clasificación similar a la que sería capaz de realizar el ojo clínico.

El objetivo de la predicción es encontrar, aplicar y testear un algoritmo capaz de predecir los ciclos de temblor en base a la cinemática. Esta predicción debe ser robusta y estar suficientemente correlada con la señal real de temblor para poder aplicarla y optimizar así las estrategias de control de la estimulación de las vías aferentes para mejorar los resultados de la supresión de temblor.

1.4. Hipótesis

Las señales cinemáticas de temblor en pacientes con TE se definen por su trayectoria, frecuencia y amplitud, presentando una frecuencia característica entre 4 y 12 Hz. La señal cinemática extraída en un paciente con TE presenta una componente periódica dominante con fluctuaciones, amplitudes y frecuencias características diferentes a un sujeto sano. Sin embargo, no todas las situaciones son ideales ya que las señales de no temblor presentan fluctuaciones a consecuencia del temblor fisiológico. El temblor patológico de los pacientes con TE es variable en el tiempo en términos de amplitud y frecuencia.

La hipótesis dentro de problema de clasificación es que los algoritmos tradicionales serán capaces de clasificar de manera correcta al menos un 90% de las instancias. Por su parte, las redes neuronales LSTM presentan una arquitectura más compleja y se caracterizan por su capacidad para tratar con series temporales, de manera que los resultados obtenidos por las redes neuronales LSTM se estima que serán mejores que los obtenidos por los clasificadores tradicionales.

La hipótesis de la tarea de predicción es que la red neuronal LSTM sea capaz de predecir las 10 primeras muestras con una correlación superior a 0.9 y que dicha correlación vaya disminuyendo a medida que aumente el horizonte de predicción. El último horizonte de predicción a evaluar son 50 muestras, que con una frecuencia de muestreo de 50 Hz se corresponde con 1 segundo a futuro. Se espera que la correlación en este horizonte de predicción sea mayor de 0.7. La predicción de las futuras 50 muestras debe mostrar un alto grado de correlación con respecto a la señal real para que la sincronización de las estimulaciones y la actividad tremorogénica sea máxima. También se estima que el valor del error cuadrático medio y la raíz del error cuadrático medio aumenten con el aumento del horizonte de predicción.

Capítulo 2

Estado del arte

El machine learning (ML) es el diseño y estudio de herramientas informáticas que utilizan la experiencia pasada para tomar decisiones futuras, es decir, es el estudio de programas capaces de aprender de los datos. El objetivo fundamental del ML es hallar reglas y patrones generalizables en un conjunto de datos [40]. Dentro del ML existe una división entre los algoritmos de clasificación, de manera que existen algoritmos de aprendizaje supervisado y algoritmos de aprendizaje no supervisado. Sin embargo, este TFG se centra únicamente en la implementación de algoritmos de aprendizaje supervisado.

El aprendizaje supervisado es un proceso de generación de conocimiento en el cual un grupo de ejemplos o datos etiquetados son conocidos previamente. Este tipo de modelos aprenden de estos resultados e incorporan ajustes en los parámetros interiores para poder adaptarse a datos nuevos que ingresarán en el sistema. Permite que los modelos sean alimentados con un conjunto de datos para posteriormente realizar predicciones sobre nuevos datos que aún no habían sido procesados. El objetivo del aprendizaje supervisado es por tanto utilizar las entradas para predecir los valores de las salidas [41].

El Deep Learning (DL) nace como una rama del ML y se define como la nueva evolución del ML. Se basa en algoritmos automáticos que buscan imitar la percepción humana. La diferencia principal entre el ML y el DL radica en el tipo de algoritmos utilizados. Mientras que el ML utiliza algoritmos más tradicionales como árboles de decisión, el DL se basa en arquitecturas de redes de neuronas con un gran número de capas. Es un algoritmo automático y estructurado que emula el aprendizaje humano.

2.1. Machine Learning

Existen muchos algoritmos que tratan la tarea de clasificación supervisada sobre un conjunto de datos etiquetados. En este TFG se implementan 3 tipos diferentes de algoritmos de aprendizaje supervisado que son el *K-Nearest Neighbors (KNN)*, el *Random Forest (RF)* y las *Support Vector Machine (SVM)*.

2.1.1. K-Nearest Neighbors (KNN)

El algoritmo KNN es uno de los métodos de clasificación más sencillos y tradicionales. Se emplea comúnmente para clasificar valores buscando los puntos de datos más similares por cercanía. Dentro de sus desventajas está el hecho de que utiliza todo el dataset para entrenar cada punto, por lo que requiere del uso de mucha

memoria y procesamiento de la Central Processing Unit (CPU).

Se parte de un dataset en el cual \mathbf{X} es el conjunto de instancias totales, \mathbf{x} es una instancia del conjunto total e \mathbf{y} es la etiqueta de \mathbf{x} . El método KNN estudia la probabilidad condicionada de \mathbf{y} dado \mathbf{x} y clasifica esa observación en base a la clase que consiga mayor probabilidad [42]. Dado un número entero k , el KNN busca en las k observaciones más cercanas a una determinada instancia x_0 del conjunto de test y calcula la probabilidad condicionada de que x_0 pertenezca a la clase j (ver Ecuación 2.1).

$$P_r(Y = j|X = x_0) = \frac{1}{k} \sum_{i \in N} I(y_i = j) \quad (2.1)$$

Cuando una nueva instancia \mathbf{x} entra, busca k ejemplos de entrenamiento similares y cercanos a \mathbf{x} y devuelve la etiqueta de la mayoría de estos k vecinos. La cercanía de las instancias se mide a través de una función de distancia como puede ser la distancia euclídea.

Para extraer las k observaciones más cercanas a x_0 es necesario calcular la distancia euclídea de x_0 con el resto de instancias de entrenamiento. Suponiendo que \mathbf{x} es un punto con coordenadas (x_1, x_2, \dots, x_p) y \mathbf{z} es un punto con coordenadas (z_1, z_2, \dots, z_p) se define la distancia euclídea en la Ecuación 2.2.

$$d(x, z) = \sqrt{\sum_{i=1}^p (x_i - z_i)^2} \quad (2.2)$$

2.1.2. Random Forest

El Random Forest es un ensamble de varios árboles de decisión [40]. Es un algoritmo de *machine learning* versátil que puede realizar tareas de clasificación [40]. Un árbol de decisión consiste en un conjunto de nodos y ramas, donde los nodos representan atributos de la señal y las ramas son las reglas de decisión. Un nodo padre se divide por medio de dos ramas en dos nodos hijos.

En una secuencia temporal los atributos son los valores de la señal para cada instante de tiempo. Una parte esencial dentro de los árboles de decisión es la selección de atributos para actuar como nodos, de forma que la partición de datos a través del nodo derive en clases individuales.

La entropía es la medida del desorden donde p_i es la probabilidad de que una instancia pertenezca a una clase determinada y c es el número de clases (ver Ecuación 2.3) [42]. En un conjunto de datos equitativamente distribuido la entropía adquiere un valor de 1.

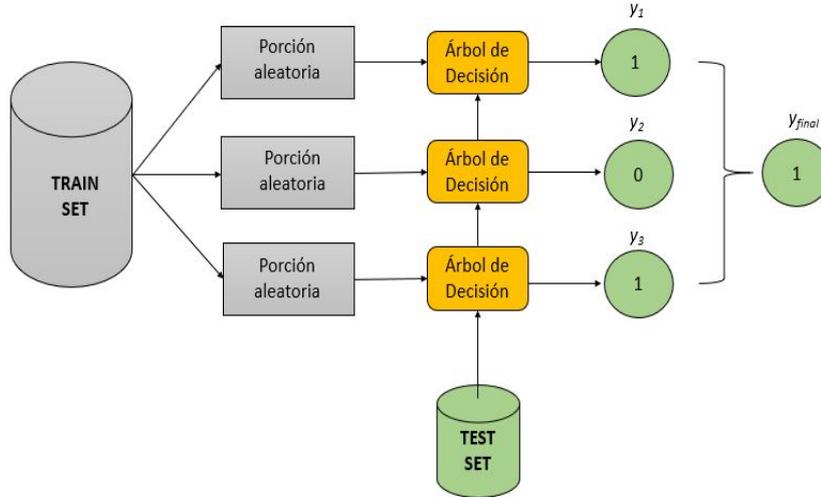


Figura 2.1: Flujo de ejecución del Random Forest formado por 3 árboles de decisión

$$E = \sum_{i=1}^c (-p_i \log_2 p_i) \quad (2.3)$$

La entropía dentro de los árboles de decisión se utiliza para calcular la Ganancia de Información (IG) del nodo especificado por un determinado atributo. Se define la IG en la Ecuación 2.4, siendo w_i el tamaño relativo, es decir, el número de instancias a las que ha dado lugar el nodo hijo con respecto al total de instancias que existían en el nodo padre.

$$IG = E(\text{Nodo padre}) - \sum (w_i E(\text{Nodo hijo})) \quad (2.4)$$

Finalmente será el atributo con mayor IG el que se utilizará como nodo para hacer la división de los datos en sus dos ramas. Este procedimiento se repite hasta extender el número de nodos deseados que nos permitan discriminar y hacer la clasificación final de los datos.

El random forest es el ensamble de varios árboles de decisión combinados con *bagging*. El *bagging* consiste en la subdivisión en porciones del conjunto de entrenamiento para alimentar con cada una de estas porciones un árbol, de manera que ningún árbol llega a ver todos los datos del conjunto de entrenamiento (ver Figura 2.1). Esto permite que el modelo generalice mejor sobre un conjunto de datos nuevos que no se hayan vistos durante la etapa de entrenamiento. Cada árbol (i) saca su output y_i , que se utiliza para asignar el valor de la etiqueta final (y_{final}) a la nueva instancia del conjunto de test. En la Figura 2.1 se muestra el flujo de ejecución con un total de 3 árboles, pudiendo este extenderse a un mayor número de árboles.

2.1.3. Support Vector Machine (SVM)

Las SVM son un modelo de machine learning versátil y potente, capaz de realizar clasificaciones binarias, lineales y no lineales. Una de sus ventajas es su capacidad para detectar valores atípicos. Se trata de uno de los modelos más populares y en la literatura correspondiente a la clasificación del temblor patológico de los más utilizados.

Las SVM toman cada instancia y la interpretan como un vector de características dentro de un gran espacio dimensional. Si cada instancia tiene Ω características la SVM verá la instancia dentro de un espacio Ω -dimensional. Dentro de una clasificación binaria, la SVM separa los dos grupos a través de un hiperplano cuya ecuación viene definida por dos parámetros (\mathbf{x} y \mathbf{w}) (ver Ecuación 2.5).

$$\mathbf{w}\mathbf{x} - b = 0 \quad (2.5)$$

La expresión $\mathbf{w}\mathbf{x}$ se desarrolla en $w_1x_1 + w_2x_2 + \dots + w_\Omega x_\Omega$ siendo Ω la dimensionalidad del espacio. Una vez definido el hiperplano se puede utilizar para hacer predicciones a través de la función de hipótesis $h(x)$ (ver Ecuación 2.6).

$$h(x) = \begin{cases} 0 & : \mathbf{w}\mathbf{x} + b < 0 \\ 1 & : \mathbf{w}\mathbf{x} + b \geq 0 \end{cases} \quad (2.6)$$

Para hallar los valores óptimos de \mathbf{w} y \mathbf{b} se debe resolver un problema de optimización. El hiperparámetro más importante de las SVM es C . El parámetro de regularización C influye en el tamaño de los márgenes del hiperplano (ver Figura 2.2). Cuanto mayor sea el valor de C más estrechos serán los márgenes y el modelo tenderá a sobreajustarse a los datos de entrenamiento limitando así su capacidad de generalizar frente a nuevas instancias. Por el contrario, un valor muy pequeño de C supone que el modelo tenga unos márgenes muy grandes y generalice demasiado.

Un factor relevante dentro de las SVM es el kernel. Existe la posibilidad de que no sea posible establecer un hiperplano capaz de dividir el conjunto de datos de manera lineal. Las SVM ofrecen una solución transformando el espacio a otras dimensiones en las que sí sea capaz de separar linealmente los datos.

Una de estas funciones que permiten transformar el espacio es la Radial Basis Function (RBF). La RBF (ver Ecuación 2.7) contiene un parámetro γ . Un valor bajo de γ hace que la SVM se asemeje a un modelo lineal, mientras que un valor alto hace que el modelo esté altamente influenciado por los ejemplos de los vectores soporte.

$$K(x_i, x_j) = \exp(\gamma \|x_i - x_j\|^2) \quad (2.7)$$

Donde la función $\|x_i - x_j\|^2$ es el cuadrado de la distancia euclídea entre dos instancias de entrada. Por su parte γ es un escalar que define la influencia de cada instancia en el modelo.

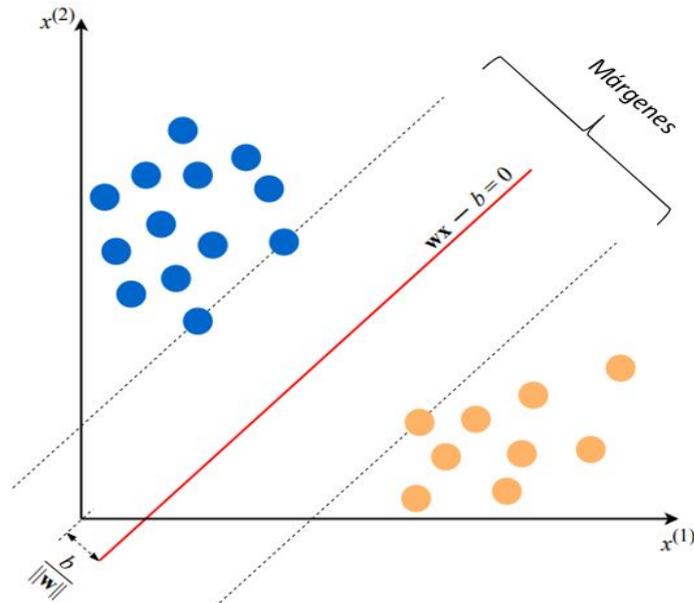


Figura 2.2: Ejemplo de modelo SVM para dos características [42]

2.2. Redes Neuronales Recurrentes

2.2.1. Introducción

Las series temporales consisten en datos que pueden ser ordenados como observaciones y ofrecer información útil, de manera que los patrones temporales de una determinada entrada pueden ayudar a la predicción de la salida de otra entrada distinta [43]. Las Redes Neuronales Artificiales (ANN) están sufriendo un incremento de su uso en la construcción de modelos computacionales dentro del ámbito de la neurociencia [44]. Las ANNs emergen como una solución alternativa al ML con un amplio rango de aplicaciones dentro de las cuales se encuentra el tratamiento de series temporales [45].

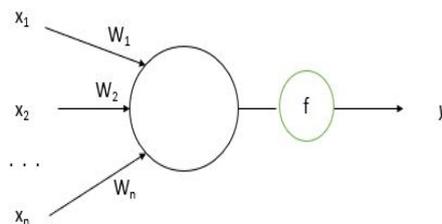


Figura 2.3: Estructura de la neurona simple

El concepto fundamental dentro de las redes neuronales son las neuronas. El modelo computacional más simple de una neurona (ver Figura 2.3) recibe una serie

de entradas (x_1, x_2, \dots, x_n) que multiplica por sus respectivos pesos (W_1, W_2, \dots, W_n) y pasa a través de una función de activación (ver Figura 2.4) para obtener una salida (y) (ver Ecuación 2.8) [40].

$$y = f\left(\sum_i x_i W_i\right) = f(x_1 W_1 + x_2 W_2 + \dots + x_n W_n) \quad (2.8)$$

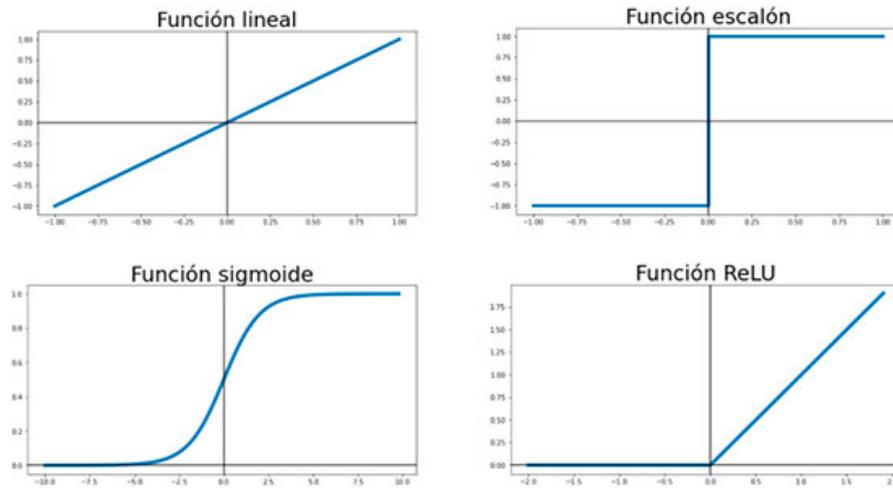


Figura 2.4: Ejemplos de funciones de activación [46]

La combinación de neuronas artificiales en una o varias capas da lugar a una ANN. Se distinguen 3 tipos de capas diferentes: la capa de entrada, la capa oculta y la capa de salida [40].

En una red con conexiones hacia adelante, *feedforward*, las neuronas de la capa de entrada no actúan como neuronas propiamente dichas, sino que se encargan de transmitir las entradas hacia las neuronas de la capa oculta. Los datos se propagan por las capas ocultas hasta llegar a la capa de salida que proporciona la respuesta de la red. Existe una conectividad total entre las neuronas de una capa y la capa siguiente para definir la relación entre las entradas y las salidas [40].

Para entrenar ANNs *feedforward* se utiliza generalmente el algoritmo de *back-propagation*. Se trata de un algoritmo de aprendizaje supervisado donde los pesos de la red se actualizan en base a la función de pérdidas (\mathcal{L}), que evalúa la similitud de la salida de la red con la salida real. El algoritmo de *back-propagation* busca por tanto hallar el mínimo de \mathcal{L} mediante la aplicación del método de descenso de gradientes. Por tanto, cada parámetro w de la red se modifica para cada entrada hasta encontrar los parámetros óptimos (ver Ecuación 2.9).

$$w_n = w_{n-1} - \alpha \frac{\partial \mathcal{L}}{\partial w} \quad (2.9)$$

Sin embargo, una ANN *feedforward* no es la mejor elección para el tratamiento de series temporales ya que la inexistencia de conexiones entre neuronas de una misma capa y su estructura sin realimentación entre capas le impiden tener dependencia temporal.

2.2.2. Redes Neuronales Recurrentes

Las Redes Neuronales Recurrentes (RNN) son una familia de redes neuronales que permiten trabajar con el procesamiento de secuencias temporales, capaces de agregar un estado de memoria. Las RNNs utilizan la compartición de parámetros a lo largo del modelo. Esta compartición de parámetros hace posible que el modelo sea capaz de generalizar y aplicar los ejemplos ya vistos sobre secuencias nuevas. En este caso los datos secuenciales se corresponden con series temporales [40].

La idea fundamental de las RNNs [47] es usar no sólo los datos de entrada sino también salidas previas para hacer la predicción en el paso de tiempo actual. Las RNNs producen una salida en cada paso de tiempo y tienen una conexión recurrente entre las neuronas ocultas. Introducido el comportamiento del MLP se puede definir una RNN de una sola capa (ver Ecuación 2.10).

$$\mathbf{h}_{(t)} = f(\mathbf{W}\mathbf{x}_t + \mathbf{U}\mathbf{h}_{t-1} + \mathbf{b}) \quad (2.10)$$

Donde \mathbf{h}_t y \mathbf{h}_{t-1} son las salidas de la neurona RNN en los pasos de tiempo actual y previo respectivamente. Por su parte \mathbf{U} es la matriz de pesos en el paso anterior y \mathbf{W} es la matriz de pesos en el paso actual. Se observa el comportamiento recurrente en el que la salida de la neurona en el paso anterior realimenta a la neurona como su nueva entrada (ver Figura 2.5).

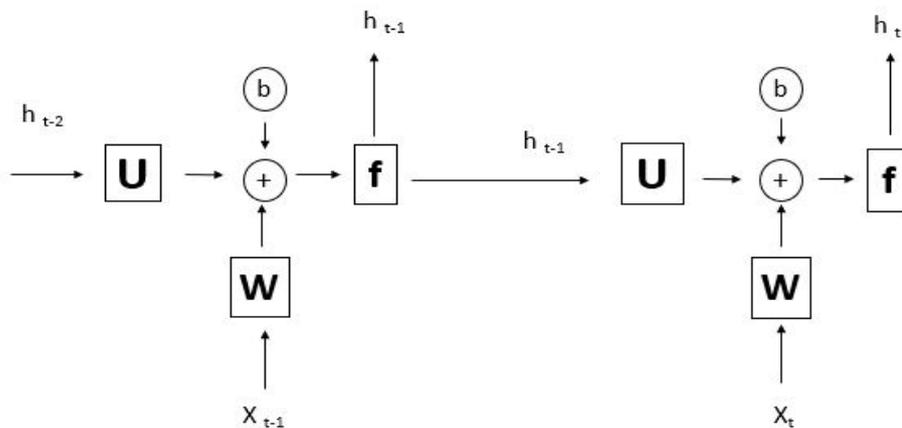


Figura 2.5: Gráfico computacional RNN de una capa.

2.2.2.1. Back Propagation Through Time (BPTT)

Las RNNs pueden entrenarse con una modificación del *back-propagation*, el BPTT. En el BPTT entra también la dimensión del tiempo, de manera que la \mathcal{L} de un determinado instante de tiempo depende del instante de tiempo anterior. El error es propagado hacia atrás desde el último hasta el primer instante de tiempo.

Los parámetros iniciales de la RNN se inicializan con un valor aleatorio y el objetivo es modificar los parámetros internos de la red (pesos y sesgos) para conseguir que la salida se aproxime lo máximo posible al valor deseado. El BPTT es un algoritmo para calcular el mínimo de la función de pérdidas. Encontrar el mínimo de la función de pérdidas (\mathcal{L}) supone encontrar los mejores parámetros de la RNN. Existen varias funciones de pérdidas que pueden ser usadas en función de la tarea a realizar, para tareas de clasificación se usa la Cross Entropy y para tareas de regresión el Error Cuadrático Medio (MSE). El algoritmo de retropropagación propaga el error de \mathcal{L} computando los gradientes de la misma. Luego se lleva a cabo la actualización de pesos y sesgos utilizando un optimizador.

Un coeficiente importante a tener en cuenta en el proceso de entrenamiento es la tasa de aprendizaje del algoritmo (α). La tasa de aprendizaje actúa en la velocidad de convergencia del modelo en la búsqueda del mínimo de \mathcal{L} . Para asegurar la convergencia, los métodos de optimización establecen un valor de α muy pequeño [48]. De esta forma la convergencia está asegurada a costa de un mayor tiempo de procesamiento. Si el valor de α es muy alto el método puede diverger y no encontrar el mínimo de \mathcal{L} , por lo que se debe establecer un valor intermedio y óptimo de α (ver Ecuaciones 2.11, 2.12 y 2.13) [49].

$$W_{t+1} = W_t - \alpha \frac{\partial \mathcal{L}}{\partial W_t} \quad (2.11)$$

$$U_{t+1} = U_t - \alpha \frac{\partial \mathcal{L}}{\partial U_t} \quad (2.12)$$

$$b_{t+1} = b_t - \alpha \frac{\partial \mathcal{L}}{\partial b_t} \quad (2.13)$$

Se puede observar en las Ecuaciones 2.11, 2.12 y 2.13 como las matrices de pesos (W y U) y sesgo (b) dependen del valor de α así como del valor anterior y de la derivada parcial de la función de pérdidas con respecto a estos parámetros (W , U y b).

El algoritmo BPTT dentro de las RNNs tiene un inconveniente relacionado con la desaparición de los gradientes en pasos de tiempo muy anteriores al actual, que impide a la red aprender patrones a largo plazo (ver Anexo E). Dicho problema se ve solucionado mediante el uso de redes neuronales Long Short Term Memory (LSTM).

2.2.2.2. Long Short Term Memory (LSTM)

Las redes neuronales LSTMs son un tipo de RNNs capaces de solucionar las limitaciones de las RNNs en el tratamiento de dependencias temporales largas al solucionar el problema del desvanecimiento de gradientes [50] [51]. La idea detrás de las redes neuronales LSTM es que introducen un bucle automático para permitir que los gradientes puedan fluir durante un tiempo prolongado sin desaparecer. Esta es la principal contribución de la memoria a largo y corto plazo. Las LSTM contienen celdas computacionalmente capaces de controlar el flujo de la información, suponiendo esto un incremento de complejidad con respecto a las RNNs. La base de las celdas LSTM son estructuras conocidas como puertas [47].

El funcionamiento de la LSTM viene definido por 4 puertas, la puerta de olvido, la puerta de entrada, la puerta de estado y la puerta de salida (ver Figura 2.6).

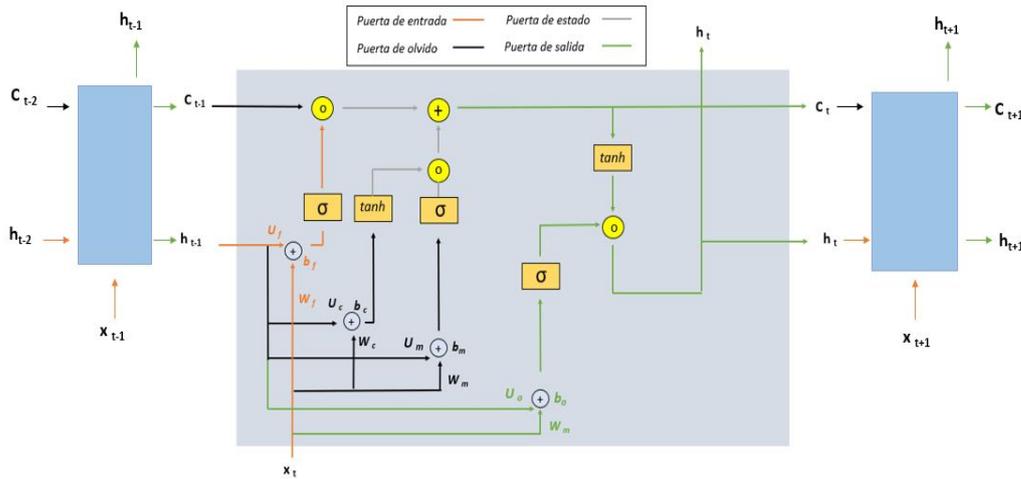


Figura 2.6: Gráfico computacional de 3 celdas LSTM

Puerta de olvido

Su función principal es decidir qué información debe quedarse en la red LSTM y cuál debe ser desechada. Esto sucede debido a que la secuencia temporal guarda patrones irrelevantes. La puerta de olvido se define acorde a las Ecuaciones 2.14 y 2.15.

$$f_t = \sigma(b_f + W_f x_t + U_f h_{t-1}) \quad (2.14)$$

$$c_f = c_{t-1} \circ f_t \quad (2.15)$$

La función de activación es una sigmoide lo que provocará que el valor de f_t se encuentre entre 0 y 1, t representa el paso de tiempo, x_t es la nueva secuencia temporal, h_t es el estado oculto que contiene las salidas de la red LSTM, b_f es el

sesgo de la puerta de olvido, U_f los pesos recurrentes y W_f los pesos aplicados a la nueva entrada. Finalmente, la f_t será multiplicada a través del producto Hadamard por el estado de la celda en el paso de tiempo anterior. El subíndice 'f' en las matrices y sesgos hace referencia a la relación de pertenencia de los mismos con la puerta de olvido [50] [51].

Puerta de entrada

La puerta de entrada emplea dos unidades funcionales. La primera es la función de activación \tanh cuyas salidas se encuentran entre -1 y 1 para decidir si cambiar el estado de la celda (\bar{c}_t). La función sigmoide es responsable de determinar la magnitud del cambio (m_t) (ver Ecuaciones 2.16 y 2.17) [52].

$$\bar{c}_t = \tanh(b_c + W_c x_t + U_c h_{t-1}) \quad (2.16)$$

$$m_t = \sigma(b_m + W_m x_t + U_m h_{t-1}) \quad (2.17)$$

Puerta de estado

Los resultados de las dos unidades funcionales de la puerta de entrada se multiplican y se suman al estado de la celda tras haber pasado por la puerta de olvido (c_f) para actualizar el estado de la celda (c_t) (ver Ecuación 2.18) [52].

$$c_t = c_f + \bar{c}_t \circ m_t \quad (2.18)$$

Es el estado [53] [54] de la celda (c_t) donde están codificadas las relaciones y dependencias temporales a largo plazo. La diferencia con respecto a las RNNs y el problema del desvanecimiento de gradientes radica en c_t , que es capaz de conseguir que los gradientes no converjan a 0 [52].

Puerta de salida

La puerta de salida usa una matriz (W_o) entrenada y un bias (b_o) para extraer información relevante de la entrada actual y la salida previa acorde a la Ecuación 2.19. Esta información se combina con la nueva celda ajustada (c_t) para predecir la próxima salida (h_t) como se muestra en la Ecuación 2.20. Esta salida se usa de manera recurrente en la próxima iteración. En el caso de que haya varias capas, esta salida actúa como entrada de la próxima capa. En caso contrario esta salida sería la predicción [52].

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (2.19)$$

$$h_t = o_t \circ \tanh(c_t) \quad (2.20)$$

2.3. Aplicación de algoritmos de ML y DL en señales de temblor patológico

Dentro de la literatura existen líneas de investigación en las que se trata tanto la tarea de clasificación del temblor patológico a través de algoritmos tradicionales de clasificación de señales cinemáticas como el problema de la predicción futura del temblor. La mayor parte de los estudios son sobre pacientes con TE y EP al ser estas las enfermedades más comunes que lo generan. Pese a existir estudios tanto de clasificación como de predicción, la clasificación del temblor es más abordada que la predicción del mismo.

En 2013, *Julien Stamatakis et al.* [55] buscaron una solución al problema de la subjetividad y variabilidad entre evaluadores en la calificación clínica del temblor en pacientes con Parkinson mediante la Escala Unificada de Calificación de la Enfermedad de Parkinson (UPDRS). Las señales de temblor estaban etiquetadas con el valor correspondiente en la escala UPDRS, y sus clasificadores consiguieron exactitudes en torno al 87% a partir de la extracción de 18 características de señales cinemáticas de una actividad de golpeteo de los dedos. Un estudio llevado a cabo por *Jeon et al.* [56] años más tarde trataba el mismo problema. Extrajeron 19 características de la señal cinemática de los movimientos con respecto a la muñeca y consiguieron una exactitud del 85.5% a través de la implementación de una SVM. En 2015 *Le Moyne et al.* [57] buscaron cuantificar la presencia de mejoría en pacientes sobre los que se aplicó DBS. Las SVM implementadas fueron capaces de alcanzar una exactitud del 100% en un trabajo de clasificación binaria entre las señales cinemáticas de temblor cinético con el DBS encendido y apagado. También usaron el procedimiento de extracción de características. Un año más tarde *Alam et al.* [58] cuantificaron la gravedad del temblor de la EP utilizando señales adquiridas de sensores inerciales portátiles en el dedo y en la muñeca. La exactitud obtenida en la tarea de clasificación por la SVM fue del 88.6% para el temblor en reposo y del 78.8% para el temblor postural.

El último estudio en lo relativo a la clasificación de señales de temblor fue llevado a cabo por *Alves et al.* en 2020 [59]. Su objetivo era comparar la exactitud de varios algoritmos de aprendizaje automático para clasificar datos relacionados con sujetos sanos y pacientes con la EP. Al igual que los estudios realizados anteriormente se llevó a cabo un proceso de extracción de características, combinando características en el dominio del tiempo y de la frecuencia. Con estas características alimentaron 7 algoritmos (KNN, SVC, regresión logística, random forest, análisis discriminante lineal, árbol de decisión y Naive Bayes). La exactitud de los clasificadores se evaluó en base a un número distinto de características (272, 190, 136, 82 y 27) y durante distintos intervalos de tiempo (1, 5, 10 y 15 segundos). Los resultados mostraron que las características más importantes eran la frecuencia media, los coeficientes de predicción lineal, la relación de potencia, la desviación de la densidad de potencia y la curtosis. De todos los clasificadores probados el KNN fue el que mejor resultados obtuvo en términos de exactitud, clasificando bien más del 90% de las instancias.

Los estudios de predicción de temblor son más escasos y apenas existen 3 en los últimos años. En 2019, *Zanini et al.* [60] proponen la comparación de varios

modelos basados en redes neuronales usando el Perceptrón Multicapa (MLP), MLP autoencoder, LSTM y una LSTM autoencoder. El objetivo es predecir el EMG del paciente con Parkinson para poder anticiparse a él y mejorar así técnicas de tratamiento como la FES. Sin embargo, sólo fueron capaces de predecir hasta 200 ms futuros a partir de los cuales los errores aumentaban limitando así su utilidad en la optimización de la FES.

Un año después, *Ibrahim et al.* [61] trabajaron en la implementación de un perceptrón convolucional multicapa capaz de predecir los movimientos voluntarios y el temblor patológico de un conjunto de pacientes en base a los datos recogidos por las Unidades de Medida Inercial (IMU). Fueron capaces de predecir la señal futura hasta 100 ms. Aún obteniendo resultados positivos, un horizonte de predicción de 100ms no permite predecir un ciclo completo de temblor entre 4 y 6 Hz, las frecuencias más comunes en pacientes, por lo que su utilidad para el control de la estimulación periférica está limitada.

Finalmente, *Shahtalebi et al.* [62] propusieron con PHTNet un marco de eliminación del temblor patológico de la mano en tiempo real a través de redes neuronales recurrentes bidireccionales profundas sin llegar a aplicarlo. Otros métodos asumen que los contenidos espectrales de los movimientos voluntarios e involuntarios son distintos, permitiendo obtener uno en base a la eliminación del otro. Sin embargo, PHTNet es un sistema que permite diferenciar movimientos voluntarios del temblor sin hacer este tipo de asunciones frecuenciales, ya que la superposición de los contenidos espectrales de los movimientos voluntarios e involuntarios es bastante natural. PHTNet no sólo es capaz de extraer los movimientos voluntarios e involuntarios sino que también permite la predicción de movimientos voluntarios futuros con un horizonte de predicción de 1 muestra. Se desarrolla sobre un conjunto de datos de 81 pacientes con TE y EP empleando señales de acción real y pseudosintetizadas (componente voluntario sintetizado + señal real de temblor postural o de reposo).

Capítulo 3

Diseño y Metodología

En este capítulo se describe la metodología empleada para implementar y validar los algoritmos de clasificación y predicción de señales cinemáticas de temblor. Primero se presenta el entorno de desarrollo utilizado, la descripción de las señales usadas, su pre-procesamiento y la creación del dataset final.

A continuación se describen los algoritmos implementados, su arquitectura y los parámetros seleccionados para las tareas de clasificación y predicción. Finalmente, se describen las métricas empleadas para evaluar la validez de los modelos.

3.1. Entorno de desarrollo

Python ha sido la herramienta utilizada para la creación de los clasificadores y predictores. *Python* es un lenguaje de programación que cuenta con iteraciones rápidas de datos, lo que favorece su aplicación dentro del campo del ML y del DL [63].

Para el tratamiento y manejo de los datos se hizo uso de la librería *NumPy*, especializada en el cálculo numérico y análisis de grandes volúmenes de datos. *NumPy* incorpora una clase de objetos llamados arrays que permiten representar colecciones de datos n-dimensionales y ofrece funciones muy eficientes para su manipulación [63].

Para la implementación de los clasificadores tradicionales (KNN, SVM y Random Forest) se utilizó la librería *scikit-learn*. *Scikit-learn* es una de las librerías más extendidas para proyectos de ML al proporcionar una alta gama de algoritmos de aprendizaje supervisado y no supervisado en *Python* [40].

Para el desarrollo de las redes neuronales LSTM se utiliza *Pytorch*. *Pytorch* es una librería de *Python* que facilita la creación de modelos de DL. La estructura de datos fundamental dentro de *Pytorch* son los tensores. Los tensores son el símil en *Pytorch* de los arrays n-dimensionales de *NumPy*. Una de las ventajas que presenta *Pytorch* es que los tensores pueden ser computados en la Graphical Processing Unit (GPU) a través de la Compute Unified Device Architecture (CUDA) [64].

La diferencia principal de la GPU con respecto a la CPU, es que la CPU presenta varios núcleos optimizados para el procesamiento de una serie secuencial, mientras que la GPU tiene una arquitectura en paralelo con un mayor número de núcleos. Esto permite que las GPUs sean capaces de procesar grandes cantidades de datos en paralelo de forma más eficiente y con una reducción del tiempo computacional con respecto a las CPUs [64].

3.2. Dataset

Para llevar a cabo las tareas de clasificación y predicción es necesario un dataset con el que entrenar los algoritmos. En esta sección se describe el proceso de adquisición de los datos así como su posterior procesado.

12 pacientes con TE de la Clínica de Desórdenes del Movimiento del hospital Gregorio Marañón (Madrid) fueron seleccionados entre abril de 2019 y enero 2020 [39]. Dichos pacientes fueron examinados por especialistas de desórdenes del movimiento del Departamento de Neurología. Los criterios de inclusión incluían el diagnóstico de TE acorde con el Tremor Research Investigation Group [65]. Los pacientes, con una edad entre 62-82 años, presentaban temblor postural con afección en al menos una de las extremidades superiores y con un temblor de flexo-extensión de muñeca prominente. No presentaban trastornos neurológicos adicionales ni otras patologías musculoesqueléticas. Todos ellos participaron en el estudio de manera voluntaria y firmaron el consentimiento informado. Los experimentos fueron registrados en un estudio sobre la aplicación de estimulación eléctrica de las vías aferentes realizado y cuyos resultados fueron publicados previamente a la elaboración de este trabajo [39]. Para complementar el dataset de clasificación se realizaron registros de 5 sujetos sanos replicando las mismas condiciones utilizadas en los pacientes.

3.2.1. Adquisición y descripción de los datos

Para la adquisición de las señales cinemáticas y fisiológicas del temblor se empleó un Sistema de Captura de Movimiento (Technaid S.L, España) (ver Anexo F). Este sistema incluye unos adaptadores, un conjunto de correas, un sistema inalámbrico de activación, un hardware de adquisición inalámbrico, un software de captura de movimiento Tech MCS Studio y un conjunto de 7 Inertial Measurement Unit (IMU).

Para conseguir una adquisición precisa de las señales cinemáticas, las IMUs integran un magnetómetro, giroscopio y acelerómetro 3D que ofrecen los valores de los componentes del campo magnético, la velocidad angular y la aceleración respectivamente sobre cada uno de los 3 ejes (x, y, z). Los ángulos se almacenaban en forma de cuaterniones y se muestrearon a 50 Hz. Se colocaron un total de 2 IMUs en cada brazo en las siguientes localizaciones:

- 1 IMU sobre el lado dorsal de cada mano.
- 1 IMU sobre el lado dorsal de cada antebrazo.

Se adquirieron las señales cinemáticas del temblor postural de los pacientes. De todas las rotaciones presentes en el movimiento tembloroso se trabajó con los movimientos de flexo-extensión de la muñeca, al presentar estos un temblor más marcado que el resto de articulaciones superiores. Se extrajeron 60 segundos de grabación por paciente, en los cuales el paciente se encontraba con los brazos estirados en contra de la gravedad.

3.2.2. Pre-procesamiento

El software de captura del movimiento Tech MCS Studio almacena las variaciones de los ángulos en forma de cuaterniones. Un cuaternión es una extensión de los números reales (ver Ecuación 3.1) donde w, x, y, z son número reales y i, j, k son números imaginarios.

$$q = w + xi + yj + zk \quad (3.1)$$

Los cuaterniones se utilizan porque su almacenamiento ocupa menos espacio que las rotaciones representadas en ángulos de Euler, sin embargo, su representación resulta menos intuitiva. Para facilitar el uso y manejo de los ángulos se llevó a cabo la conversión de cuaterniones a ángulos de Euler. Finalmente los ángulos quedaron representados en base a las rotaciones con respecto a los ejes de las IMUs (x, y, z). En este TFG se utilizaron los movimientos de flexo-extensión de muñeca, que se corresponden con las rotaciones alrededor del eje y de las IMUs. Las rotaciones quedan definidas como ϕ, θ y ψ , siendo ϕ los ángulos de la rotación con respecto al eje x , θ los ángulos de la rotación con respecto al eje y y ψ los ángulos de la rotación con respecto al eje z . El dataset almacena por tanto la variación temporal de θ .

Con el fin de eliminar las componentes frecuenciales del temblor fisiológico y obtener las señales temblorosas provocadas por el TE se aplicó sobre las señales cinemáticas un filtro IIR paso banda entre 4 y 10 Hz tipo Butterworth de orden 3. Los filtros de tipo Butterworth son una familia de filtros que no tienen rizado en la banda de paso y cuya transición entre la banda de paso y la banda eliminada es poco abrupta. Una vez obtenidos los coeficientes del filtro paso banda, se utilizaron para realizar un filtrado digital de fase cero procesando los datos de entrada. Después de filtrar los datos en la dirección de avance, invierte la secuencia filtrada y la vuelve a ejecutar. De esta forma se obtiene una distorsión de fase cero y una función de transferencia del filtro igual a la magnitud cuadrada de la función de transferencia de filtro original [66].

3.2.3. Creación del dataset

Existen dos datasets diferentes, uno para evaluar la tarea de clasificación y otro para la predicción.

El dataset de clasificación consta de señales cinemáticas del ángulo de flexo-extensión de la muñeca en pacientes enfermos (12) y en sujetos sanos (5). Con todas las grabaciones de las señales cinemáticas se formaron 18.000 secuencias temporales de 1 segundo cada una. Cada segmento de 1 segundo está procesado de acuerdo con los métodos ya definidos (ver Subsección 3.2.2).

Teniendo en cuenta que la frecuencia de muestreo de las IMUs es de 50 Hz, el número de muestras de cada instancia es de 50 (ver Ecuación 3.4). Cada una de

las columnas del dataset se corresponde con el valor del ángulo de flexo-extensión de muñeca en un determinado momento temporal, y la última columna almacena la etiqueta correspondiente a ese segmento temporal.

Se trata de un dataset equilibrado, con un 50 % (9.000 instancias) de los datos etiquetados como temblor (etiqueta 1) y el 50 % (9.000 instancias) restante como no temblor (etiqueta 0). Es esencial tener un dataset balanceado con el fin de que los clasificadores no se vean afectados a consecuencia de una desproporción entre el número de instancias clasificadas como temblor y el número de instancias clasificadas como no temblor. En la Figura 3.1 se observa la comparación de las señales cinemáticas temblorosas y no temblorosas.

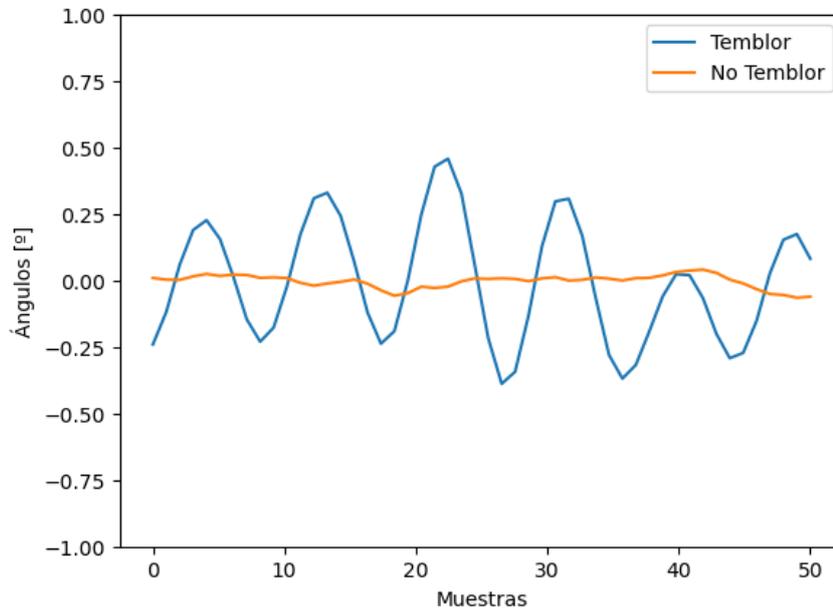


Figura 3.1: Comparación de los dos tipos de señales cinemáticas del dataset

Para el etiquetado del dataset para las tareas de clasificación supervisada se usó el valor de la Densidad Espectral de Potencia (PSD) de las señales procesadas. La PSD contiene información acerca del porcentaje de la potencia total de la señal que corresponde a cada una de las componentes en frecuencia. Todo el rango temporal de la señal influye en su PSD. La definición formal de la PSD (ver Ecuación 3.2) es la Transformada de Fourier (TF) de la función de autocorrelación, siendo la función de autocorrelación la similitud de la señal con una réplica de sí misma desplazada un tiempo τ [67] [68].

$$S(f) = TF(R(\tau)) = \int_{-\infty}^{\infty} R(\tau) \cdot e^{-j2f\tau\pi} d\tau \quad (3.2)$$

Para poder calcular la PSD de la señal cinemática es necesario tener un conocimiento absoluto de ella, sin embargo, el conocimiento de la señal cinemática de temblor (longitud finita) es limitado. No es posible calcular la PSD de una señal a partir de observaciones, por lo que se lleva a cabo una estimación espectral

de la misma. La estimación espectral es un tipo de análisis probabilístico que permite estimar la PSD a partir de un número finito k de observaciones, siendo $k = 50$. Al trabajar con un análisis probabilístico se considera la señal cinemática como un proceso estocástico con dos propiedades importantes: la estacionariedad (las propiedades estadísticas de la señal se mantienen a lo largo del tiempo) y la ergodicidad (el promedio de las estimaciones se corresponde con el promedio real). Para estimar la PSD de las señales cinemáticas se utilizó un método no paramétrico conocido como el método de Welch, que basa su estimación en la definición formal de la PSD. La estimación espectral de la PSD queda definida, siendo $M = 50$ el número de muestras de cada segmento temporal (ver Ecuación 3.3) [67] [68].

$$\hat{S}(f, k) = \frac{1}{M} \int_{k=-\frac{M}{2}}^{\frac{M}{2}} x(t, k) \cdot e^{-j2ft\pi} dt \quad (3.3)$$

Se llevó a cabo un etiquetado heurístico del dataset a través de la PSD. Se siguió esta dinámica por dos motivos principales. El primer motivo es que la PSD ofrece una caracterización objetiva del temblor en base a características propias de las señales cinemáticas y sus frecuencias, aportando una objetividad mayor en el etiquetado. El segundo motivo es que pese a que un paciente esté diagnosticado con TE, no necesariamente todas las señales de 1 segundo de duración extraídas durante las grabaciones del temblor deben presentar actividad tremorogénica. Con la PSD se asegura que los segmentos etiquetados con temblor verdaderamente presentan actividad tremorogénica, y así los algoritmos de clasificación no sufren por un mal etiquetado de las señales cinemáticas [67] [68].

$$\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,50} & 0 \\ x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,50} & 1 \\ x_{3,1} & x_{3,2} & x_{3,3} & \cdots & x_{3,50} & 1 \\ \vdots & \vdots & \vdots & \cdots & \vdots & \vdots \\ x_{18000,1} & x_{18000,2} & x_{18000,3} & \cdots & x_{18000,50} & 0 \end{bmatrix} \quad (3.4)$$

Una vez obtenida la PSD se puede observar la distribución de potencia en cada una de las frecuencias. Las señales temblorosas de pacientes con TE muestran de carácter general un valor máximo entre los 4 y los 10 Hz, al ser esta la frecuencia característica del temblor de esta patología. Por contra, la señal asociada a un paciente sano no mostrará tal pico en esas componentes en frecuencia. Para diferenciar entre señales cinemáticas temblorosas y no temblorosas se ha establecido un umbral en base al valor máximo de la PSD de la señal entre las frecuencias de 4 y 10 Hz.

Se ha considerado como señal temblorosa aquella cuyo valor de ángulo pico pico (V_{pp}) es superior o similar a 0.4 grados. Dicho valor se determinó de forma experimental en base a los datos registrados en sanos, donde no había temblor. El valor de la PSD asociado a una señal de $V_{pp} = 0.4^\circ$ es de $0.006 \frac{V^2}{Hz}$. En la Figura 3.2 se observa la comparación de la PSD de las señales cinemáticas temblorosas y no temblorosas.

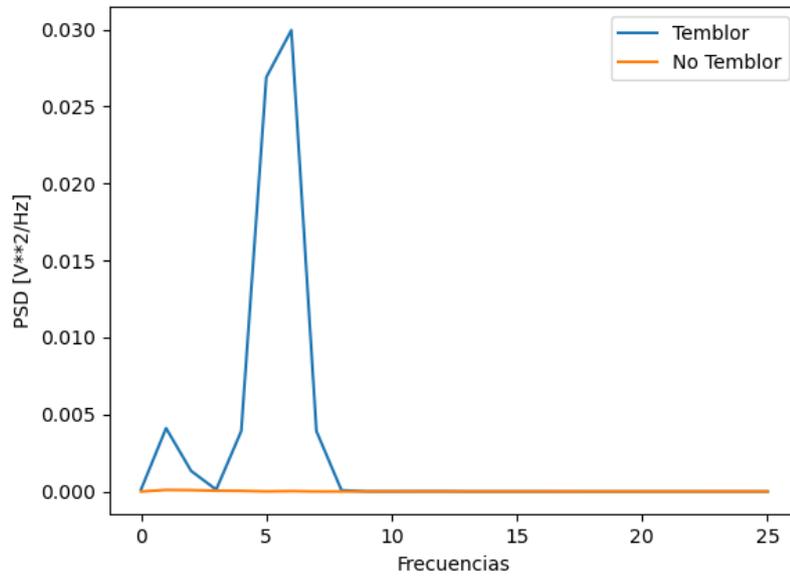


Figura 3.2: Comparación de la PSD de los dos tipos de señales cinemáticas del dataset

El dataset para la predicción, a diferencia de la creada para la tarea de clasificación, no tendrá etiquetado y simplemente almacenará las secuencias temporales. Las secuencias temporales del dataset de predicción (ver Ecuación 3.5) son similares a las creadas para la tarea de clasificación. Únicamente cambia la longitud de las secuencias temporales para poder evaluar las redes LSTM con diferentes horizontes de predicción. Para realizar la tarea de predicción y entrenar la LSTM se necesitan señales temblorosas, por lo que ya no se emplearon las señales de sujetos sanos. El dataset consta de 7.000 segmentos de señales cinemáticas de pacientes con TE de 2 segundos cada una.

$$\begin{bmatrix} x_{1,1} & x_{1,2} & x_{1,3} & \cdots & x_{1,100} \\ x_{2,1} & x_{2,2} & x_{2,3} & \cdots & x_{2,100} \\ x_{3,1} & x_{3,2} & x_{3,3} & \cdots & x_{3,100} \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ x_{7000} & x_{7000,2} & x_{7000,3} & \cdots & x_{7000,100} \end{bmatrix} \quad (3.5)$$

Las bases de datos de clasificación y predicción se dividieron en 3 conjuntos cada una:

- **Conjunto de entrenamiento:** El conjunto de entrenamiento representa un 75 % del total de los datos y se utiliza para entrenar el modelo que luego se usará para la clasificación y predicción de nuevas instancias.
- **Conjunto de validación:** El conjunto de validación representa un 15 % del total de los datos y se utiliza para validar los modelos y encontrar el más óptimo asociado a la tarea de clasificación y predicción.
- **Conjunto de prueba:** El conjunto de prueba representa un 15 % del total de los datos y se utiliza en la etapa final para evaluar la calidad del modelo y ver

su capacidad de clasificar y predecir nuevas instancias que no hayan sido vistas en las etapas de entrenamiento y validación anteriores.

3.3. Clasificación de señales cinemáticas de temblor

En esta sección se definen los parámetros establecidos en cada uno de los clasificadores, métricas usadas para evaluar la calidad de los modelos y la arquitectura de las redes LSTM de clasificación.

3.3.1. KNN

Para entrenar el KNN las señales del dataset de clasificación (ver Subsección 3.2.3) fueron normalizadas transformando cada valor a un rango entre 0 y 1. La transformación viene definida por la Ecuación 3.6:

$$\hat{x}_i = \frac{x_i - x_{min}}{x_{max} - x_{min}} \quad (3.6)$$

siendo x_i el valor específico de la serie temporal a ser normalizado. Donde $i \in (1, \dots, N)$ siendo $N = 50$, x_{min} es el valor mínimo de los ángulos del dataset y x_{max} es el valor máximo de los ángulos del dataset.

Para la búsqueda del mejor modelo se crearon varios modelos cambiando los parámetros y se evaluaron en el conjunto de validación mediante la técnica *K-Fold-Cross-Validation* para encontrar aquel con los parámetros óptimos. El método *K-Fold-Cross-Validation* consiste en dividir el conjunto de validación en $k = N$ grupos del mismo tamaño, empleando $k-1$ grupos para entrenar el modelo y el grupo sobrante como validación. Después se usan otros $k-1$ grupos para entrenar y se evalúa sobre el *fold* sobrante, un total de $k = N$ veces. Se ha elegido un valor de $N = 10$. Una vez obtenido el modelo óptimo, se entrenó con las instancias del conjunto de entrenamiento para finalmente testarlo en el conjunto de prueba. Ninguno de los conjuntos comparte instancias para garantizar la independencia estadística entre conjuntos y evitar que el modelo se guíe por patrones erróneos.

Tras la búsqueda de parámetros óptimos se ha establecido un número de 6 vecinos más cercanos para realizar la clasificación de la señal del conjunto de prueba. Se ha elegido el *ball tree* como el algoritmo para computar y encontrar los 6 vecinos más cercanos. El *ball tree* sitúa todas las instancias en un espacio n-dimensional sobre el que crea dos hiperesferas. Cada una de las instancias pertenece a una de las dos hiperesferas, específicamente a aquella cuya distancia euclídea con el centroide sea menor. Cada hiperesfera se subdivide a su vez en dos hiperesferas secundarias y así sucesivamente hasta que la hiperesfera contenga K instancias (siendo $K = 6$ el número de vecinos). El parámetro *weights* del modelo indica la influencia que tienen los vecinos más cercanos al clasificar la nueva instancia. Se ha establecido este parámetro con el

valor *distance* que provoca que las instancias más cercanas a la instancia a clasificar tengan un peso mayor que aquellas más alejadas, es decir, el vecino 1 (más cercano) influye más que el vecino 6 (menos cercano) en la clasificación de la nueva instancia. Quedan definidos por tanto los parámetros del modelo a continuación:

- Número de vecinos = 6
- Weights = Distance
- Algorithm = Ball Tree

3.3.2. Random Forest

Tras la búsqueda de parámetros óptimos con el *k-fold-cross-validation* se ha obtenido que el número óptimo de árboles de decisión del Random Forest es 100. Es decir, va a evaluar la clasificación de cada instancia del conjunto de prueba en esos 100 árboles y le asignará la etiqueta más común dentro de las 100 etiquetas obtenidas. La calidad de la división de los nodos viene definida por un parámetro denominado *criterion*. Se usa la entropía como medida para calcular la ganancia de información de cada nodo. También se ha definido el número de características máximas a tener en cuenta en la búsqueda del mejor nodo. Las características se corresponden con cada uno de los instantes de tiempo de la señal cinemática y se limita a un total de \sqrt{N} siendo $N = 50$ el número total de características. Los parámetros del modelo que mejores resultados dio en el proceso de validación quedan resumidos a continuación:

- Número de árboles = 100
- Criterion = Entropy
- Máximo número de características = Sqrt

3.3.3. SVM

Para implementar la SVM es necesaria la definición de dos parámetros: C y el kernel. El kernel es una función capaz de transformar el espacio a otras dimensiones en las que el modelo pueda separar linealmente los datos a través de un hiperplano. Existen muchas funciones que pueden realizar esta transformación, y de entre todas ellas se ha seleccionado la función *rbf* (ver teoría SVM 2.1.3).

El segundo parámetro a definir es C. C controla el número y la severidad de las clasificaciones erróneas dictaminadas por el hiperplano en el proceso de entrenamiento. Un valor bajo de C indica que los errores se penalizan menos y más observaciones pueden estar en el lado contrario del margen. Esto provoca que el modelo no consiga ajustarse a los datos y la generalización sea excesiva. Por contra, un valor elevado de C indica que los errores se penalizan demasiado, lo que deriva en un modelo sobreajustado a los datos e incapaz de generalizar ante nuevas observaciones.

El valor óptimo para este SVM con este conjunto de datos es 10. Quedan definidos a continuación los parámetros del modelo SVM más óptimo encontrado tras el proceso de validación:

- Kernel = RBF
- C = 10

3.3.4. Red LSTM para clasificación

Se evaluó el modelo de clasificación con distinto número de neuronas ocultas (z)(20, 35 y 50) y learning rate (0.0001, 0.001 y 0.005) dando un total de 9 modelos diferentes.

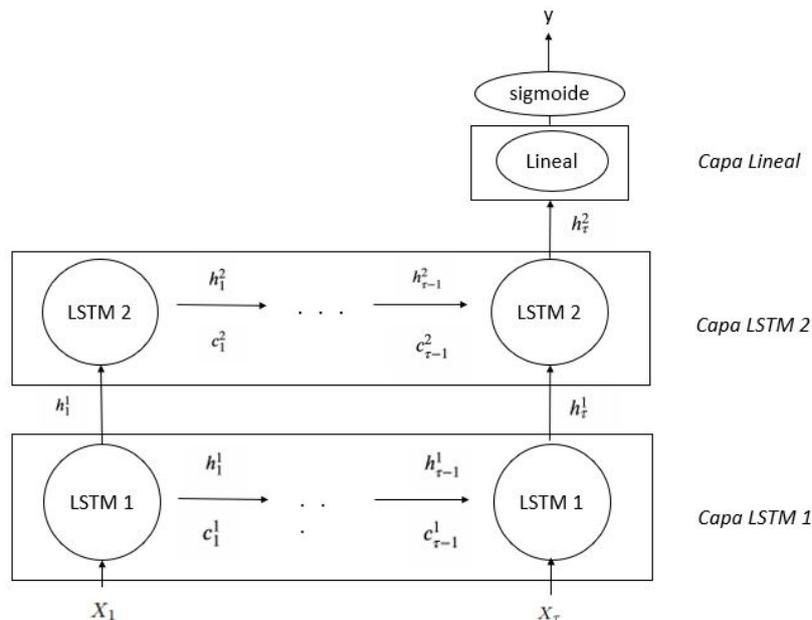


Figura 3.3: Arquitectura LSTM de clasificación [69]

El clasificador se compone de 3 capas, 2 capas de LSTM y una capa lineal final (ver Figura 3.3). La capa de la LSTM posee z neuronas ocultas y la capa lineal final es un perceptrón. La salida del perceptrón pasa por una función sigmoide para dar el resultado final (y) de la clasificación de esa instancia. Cada valor de la secuencia temporal ($X_1 \dots X_\tau$) pasa por la LSTM y en cada paso de tiempo se van actualizando tanto el estado de la celda (c_t^1) como el estado oculto (h_t^1). La entrada en cada paso de tiempo de la segunda LSTM será el estado oculto de la primera LSTM para ese paso de tiempo. Será el estado oculto del último paso de tiempo (h_τ^1) el que se pasará a la siguiente capa LSTM como su entrada. La segunda LSTM usará esta entrada para sacar el estado oculto h_τ^2 del último paso de tiempo a la capa lineal, ya que será este estado oculto el que almacene información de todos los pasos de tiempo

anteriores. El estado final (h_T^2) es un vector de n neuronas ocultas. La capa lineal recibe h_T^2 y genera una salida de 1 dimensión que pasará por una función sigmoide para finalmente obtener la etiqueta predicha de la instancia x (predicción binaria). La arquitectura de red LSTM es entrenada en base a las entradas, realizando la predicción (y) y comparándola con las etiquetas reales de las entradas recibidas por medio de la función de pérdidas. Finalmente se lleva a cabo el proceso de BPTT y el optimizador *Adam* actualiza los pesos de la red. Este proceso sucede cada época. Una época se corresponde con el paso de todos los datos del conjunto de entrenamiento por la red. Tras cada época se valida el modelo en el conjunto de validación para evitar el sobreajuste, de manera que si la función de pérdidas de la época actual es menor que la función de pérdidas del último modelo guardado, el modelo guardado se actualiza con los nuevos pesos. El optimizador encargado de actualizar los pesos de la red tras el BPTT es el *Adam* con un valor ε de learning rate. El cálculo de la función de pérdidas sobre la que llevar a cabo el descenso de gradientes se obtiene con el método *Binary Cross-Entropy* (ver Ecuación 3.7), recomendado para tareas de clasificación binaria [40].

$$H_{BCE} = -\frac{1}{N} \sum_{i=1}^N y_i \cdot \log(p(y_i)) + (1 - y_i) \cdot \log(1 - p(y_i)) \quad (3.7)$$

siendo y la etiqueta de la instancia, $p(y)$ la probabilidad de que esa instancia pertenezca a la etiqueta y , y N el número total de instancias.

El objetivo del conjunto de prueba es evaluar el rendimiento del último modelo de la red neuronal LSTM guardado (aquel que presentaba un mejor rendimiento en el conjunto de validación) sobre un conjunto de instancias que no hayan sido vistas antes.

3.3.5. Métricas para evaluar la clasificación

Para evaluar el rendimiento de los clasificadores propuestos en la Sección 3.3 es necesario definir una serie de métricas. Las métricas presentadas en esta sección permiten evaluar la calidad de la clasificación mediante el cálculo de verdaderos positivos (VP), verdaderos negativos (VN), falsos positivos (FP) y falsos negativos (FN). La primera métrica a considerar es la matriz de confusión (ver Ecuación 3.8) que muestra la distribución de los valores reales y de las predicciones [40]. La correspondencia entre la matriz de confusión y las señales es la siguiente:

- VP: Señales de temblor clasificadas como temblor
- FP: Señales de no temblor clasificadas como temblor
- VN: Señales de no temblor clasificadas como no temblor
- FN: Señales de temblor clasificadas como no temblor

		Valores predicción	
		Positivo	Negativo
Valores actuales	Positivo	VP	FN
	Negativo	FP	VN

(3.8)

A partir de la matriz de confusión se pueden extraer otras 4 métricas: Exactitud, precisión, sensibilidad y especificidad. La exactitud contempla el número de instancias totales bien clasificadas (ver Ecuación 3.9). Mide la relación de predicciones correctas en el conjunto de prueba [40].

$$Exactitud = \frac{VP + VN}{VP + VN + FN + FP} \quad (3.9)$$

La exactitud es una buena métrica para evaluar la calidad de un modelo, sin embargo, para un estudio más profundo del clasificador es necesario estudiar otras métricas (precisión y sensibilidad). La precisión es la exactitud de las predicciones positivas, es decir, de las señales clasificadas como temblor cuántas verdaderamente lo son (ver Ecuación 3.10) [40].

$$Precisión = \frac{VP}{VP + FP} \quad (3.10)$$

La precisión es una métrica que suele ir acompañada de la sensibilidad (ver Ecuación 3.11). La sensibilidad es la relación de instancias positivas que el clasificador detecta de manera correcta, es decir, el número de VP que se le escapan al clasificador [40].

$$Sensibilidad = \frac{VP}{VP + FN} \quad (3.11)$$

Finalmente, se emplea la especificidad que es número de instancias etiquetadas correctamente como no temblor con respecto al total de señales de no temblor (ver Ecuación 3.12) [40].

$$Especificidad = \frac{VN}{VN + FP} \quad (3.12)$$

3.4. Predicción de señales cinemáticas de temblor

Para la tarea de predicción se usa el dataset de predicción en lugar del de clasificación. Se realiza el proceso de normalización de datos seguido en la tarea de clasificación (ver Subsección 3.3.4). Se evaluó el modelo de predicción con distinto número de capas de LSTM (n)(1 y 2), neuronas ocultas (z)(20, 35 y 50), learning

rate (α)(0.0001, 0.001 y 0.0005), número de muestras en el horizonte de predicción (δ)(5, 10, 20, 30 y 50) y número de muestras para entrenar (τ)(30 y 50) dando un total de 180 modelos diferentes.

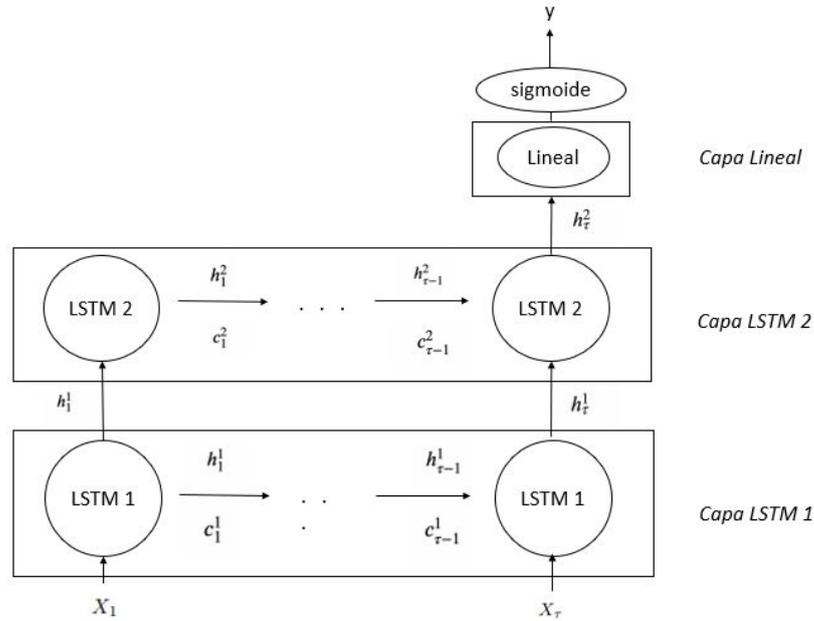


Figura 3.4: Arquitectura LSTM predicción [69]

Suponemos para la explicación de la arquitectura y el proceso de entrenamiento que el clasificador se compone de 3 capas (2 capas de LSTM y una capa lineal) (ver Figura 3.4). Cada valor de la secuencia temporal ($X_1 \cdots X_\tau$) pasa por la primera capa LSTM y en cada paso de tiempo se van actualizando tanto el estado de la celda (c_t^1) como el estado oculto (h_t^1). El estado oculto de la primera LSTM de cada paso de tiempo se pasará como entrada a la segunda LSTM. La capa lineal recibe los estados del último paso de tiempo de la segunda LSTM (h_τ^2) (vector de n neuronas) y genera una salida de δ dimensiones que pasará por una función sigmoide para finalmente obtener los futuros δ valores de la secuencia temporal. La LSTM hace predicciones en base a las entradas que recibe y compara el valor predicho (y) con el valor real de la señal de entrada por medio de la función de pérdidas, que en este caso, a diferencia de la clasificación, se obtiene por medio del Error Cuadrático Medio (MSE) (ver Ecuación 3.13). Finalmente se lleva a cabo el proceso de BPTT y el optimizador *Adam* actualiza los pesos de la red en base a α . Este proceso sucede cada época. El proceso de validación es similar al de clasificación (ver Subsección 3.3.4).

$$MSE = \left(\frac{1}{N}\right) \sum_{i=1}^N (t_i - y_i)^2 \quad (3.13)$$

Finalmente, se testea el modelo sobre el conjunto de prueba y se evalúa su rendimiento sobre un conjunto de instancias nuevas que no haya visto el modelo

ni durante la etapa de entrenamiento ni durante la etapa de validación.

3.4.1. Métricas para evaluar la predicción

Para evaluar el rendimiento de la red neuronal LSTM propuesta en la Sección 3.4 es necesario definir una serie de métricas. Las métricas presentadas en esta sección se han elegido por su capacidad para evaluar series temporales. Siendo N el número de muestras a ser evaluadas, $Y=y_1, \dots, y_n$ las salidas propuestas por el predictor y $T=t_1, \dots, t_n$ las salidas reales, se define el MSE (ya descrito como la función de pérdidas de la red LSTM)(ver Ecuación 3.13), la Raíz del Error Cuadrático Medio (RMSE) y el coeficiente de correlación de Pearson.

El RMSE (ver Ecuación 3.14) es la raíz del MSE y proporciona una idea de la exactitud de las predicciones del modelo.

$$RMSE = \sqrt{\left(\frac{1}{N}\right) \sum_{i=1}^N (t_i - y_i)^2} \quad (3.14)$$

Otra de las métricas con las que se ha trabajado es el coeficiente de correlación de Pearson ($r_{t,y}$) (ver Ecuación 3.15). Este coeficiente aporta información sobre la similitud entre las predicciones y la señal real. Un valor del coeficiente de correlación de Pearson cercano a 1 implica una alta similitud entre las predicciones y las señales reales.

$$r_{t,y} = \frac{\sum_{i=1}^N (t_i - \bar{t}_i)(y_i - \bar{y}_i)}{\sqrt{\sum_{i=1}^N (t_i - \bar{t}_i)^2} \sqrt{\sum_{i=1}^N (y_i - \bar{y}_i)^2}} \quad (3.15)$$

Capítulo 4

Resultados

En este capítulo se evaluará el rendimiento de los clasificadores y predictores en base a las métricas especificadas en la Subsección 3.3.5 y en la Subsección 3.4.1. Se dividirá en dos secciones, una para los resultados de clasificación obtenidos por medio de los clasificadores tradicionales y la LSTM de clasificación, y otra sección para los resultados de la predicción llevada a cabo por la red neuronal LSTM de predicción.

4.1. Clasificación de señales cinemáticas de temblor

Esta sección se subdivide en 4 subsecciones, una para cada uno de los clasificadores empleados (KNN, Random Forest, SVM y LSTM).

4.1.1. Resultados KNN

Para la evaluación del *KNN* se crearon un total de 6 clasificadores (ver Anexo C.1) variando los parámetros del número de vecinos y la función *weights* que indicaba la influencia de los vecinos más cercanos sobre la clasificación de una nueva instancia (ver Sección 3.3). Se comparó la exactitud de cada modelo sobre el conjunto de validación para obtener aquel con los parámetros óptimos ya definidos en la Sección 3.3. Se evaluaron los modelos a través del *k-fold-cross-validation* con un valor de $k = 10$. Se obtienen por tanto 10 valores de exactitud para cada uno de los *folds* creados, valores que se utilizan para realizar el boxplot (ver Figura 4.1). El modelo que mejor se adapta y clasifica los datos es el Modelo 2. La media de las exactitudes del Modelo 2 (90.88 %) y su mediana (91.11 %) son superiores a las obtenidas en el resto de modelos. Además no tiene valores atípicos y el 75 % de las exactitudes obtenidas son superiores a las medianas del resto de modelos. Los parámetros del Modelo 2 (ver Tabla 4.1) vienen explicados en la Sección 3.3.

Modelo 2		
Número de vecinos	Weights	Algorithm
6	<i>Distance</i>	<i>Ball Tree</i>

Tabla 4.1: Parámetros Modelo 2 KNN

Se escogió este modelo para entrenarlo sobre un mayor número de instancias del conjunto de entrenamiento y posteriormente se evaluó su rendimiento en el conjunto de prueba para extraer las métricas especificadas en la Subsección 3.3.5. A partir de la matriz de confusión (ver Figura 4.2) se obtienen las métricas de la Tabla 4.2.

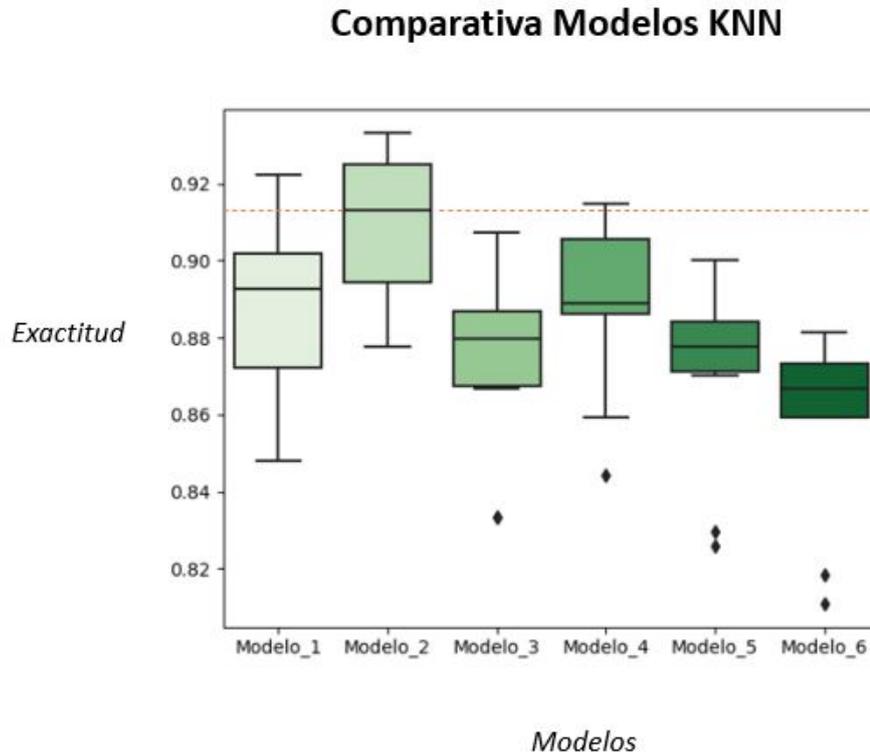


Figura 4.1: Boxplot comparativo de los modelos KNN

Métricas KNN			
Exactitud	Precisión	Sensibilidad	Especificidad
94 %	90 %	99 %	89 %

Tabla 4.2: Métricas KNN

El *KNN* clasifica correctamente el 94 % de las instancias, presentando la mayor parte de los errores en señales que eran de no temblor pero fueron clasificadas como temblor. Se observa que tiene mayor facilidad para distinguir las señales de temblor que aquellas sin temblor, por eso los valores de la sensibilidad son tan altos. Pese a todo, los resultados son satisfactorios y se puede concluir que es un modelo capaz de realizar la clasificación binaria de señales cinemáticas de temblor.

		MATRIZ DE CONFUSIÓN	
Reales	TEMBLOR	1362	18
	NO TEMBLOR	146	1174
		TEMBLOR	NO TEMBLOR
		Predicciones	

Figura 4.2: Matriz de confusión Modelo 2 KNN

4.1.2. Resultados Random Forest

Para la evaluación del *Random Forest* se crearon un total de 18 modelos (ver Anexo C.2) variando los siguientes parámetros: Número de árboles totales, medida para calcular la IG de cada nodo (*criterion*) y el máximo número de características a tener en cuenta en la elección del nodo. Se siguió el mismo procedimiento de validación especificado en el *KNN*. El modelo óptimo se define en la Tabla 4.3.

Modelo 5 Random Forest		
Número de árboles	Criterion	Máximo número de características
100	Entropy	Sqrt

Tabla 4.3: Parámetros Modelo 5 Random Forest

El modelo elegido ha sido el número 5 con un valor medio de exactitud del 95.03 %. Se ha elegido el *Modelo 5* porque su mediana es superior al resto de modelos. Existen otros modelos con valores de mediana parecidos pero el modelo 5 presenta una variabilidad muy baja entre el 1 y 2 cuartil, a diferencia de modelos con medianas similares lo que le convierte en el modelo óptimo dentro de los Random Forest (ver Figura 4.3).

Métricas Random Forest			
Exactitud	Precisión	Sensibilidad	Especificidad
96 %	97 %	95 %	96 %

Tabla 4.4: Métricas Random Forest

Al igual que con el *KNN* se entrenó el modelo con las instancias del conjunto de entrenamiento para evaluarlo posteriormente sobre el conjunto de prueba. La matriz de confusión obtenida sobre el conjunto de prueba se muestra en la Figura 4.4. A partir de la matriz de confusión se obtienen las métricas especificadas en la Tabla 4.4.

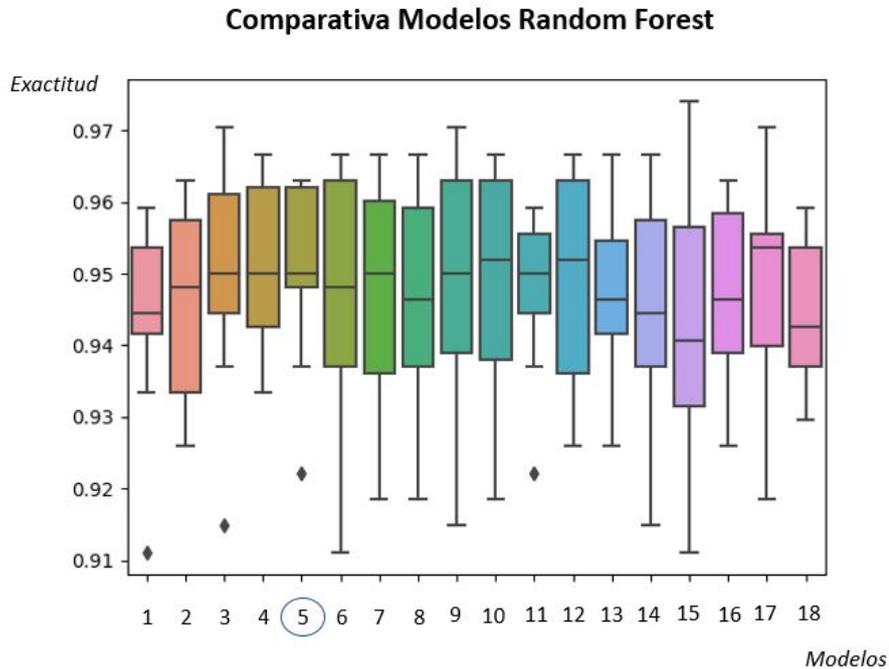


Figura 4.3: Boxplot comparativo de los modelos Random Forest

		MATRIZ DE CONFUSIÓN	
<i>Reales</i>	TEMBLOR	1306	74
	NO TEMBLOR	47	1273
		<i>TEMBLOR</i>	<i>NO TEMBLOR</i>
		<i>Predicciones</i>	

Figura 4.4: Matriz de confusión del Modelo 5 Random Forest

El *Random Forest* ha tenido un desempeño ligeramente mejor que el *KNN*. El número de errores (FP y FN) está más repartido que en caso del *KNN* viéndose esto reflejado en los valores de exactitud, precisión y especificidad.

4.1.3. Resultados SVM

Para la evaluación del *SVM* se crearon un total de 12 modelos (ver Anexo C.3) variando el parámetro *C* y la función kernel. El modelo elegido ha sido el número 9 (ver Tabla 4.5) ya que presenta un rendimiento muy superior al resto de modelos (ver Figura 4.5). Su exactitud media es del 89%, no presenta valores de exactitud atípicos

y la varianza de los cuartiles es baja.

Modelo 9 SVM	
C	Función kernel
10	<i>rbf</i>

Tabla 4.5: Parámetros Modelo 9 SVM

Existe una gran diferencia entre los modelos 7, 8 y 9 en comparación con los otros modelos, y esto se debe a la función kernel que se ha utilizado (*rbf*). Por tanto, la función *rbf* se ajusta mejor a los datos que la función sigmoide, lineal o polinomial. El valor de $C = 10$, hace que el modelo se ajuste más a los datos de entrenamiento que los modelos con $C = 1$ y $C = 0,1$, sin llegar al sobreajuste.

Métricas SVM			
Exactitud	Precisión	Sensibilidad	Especificidad
91 %	86 %	98 %	84 %

Tabla 4.6: Métricas SVM

La matriz de confusión del modelo 9 tras evaluarlo en el conjunto de prueba se ve en la Figura 4.6. A partir de la matriz de confusión se obtienen las métricas especificadas en la Tabla 4.6.

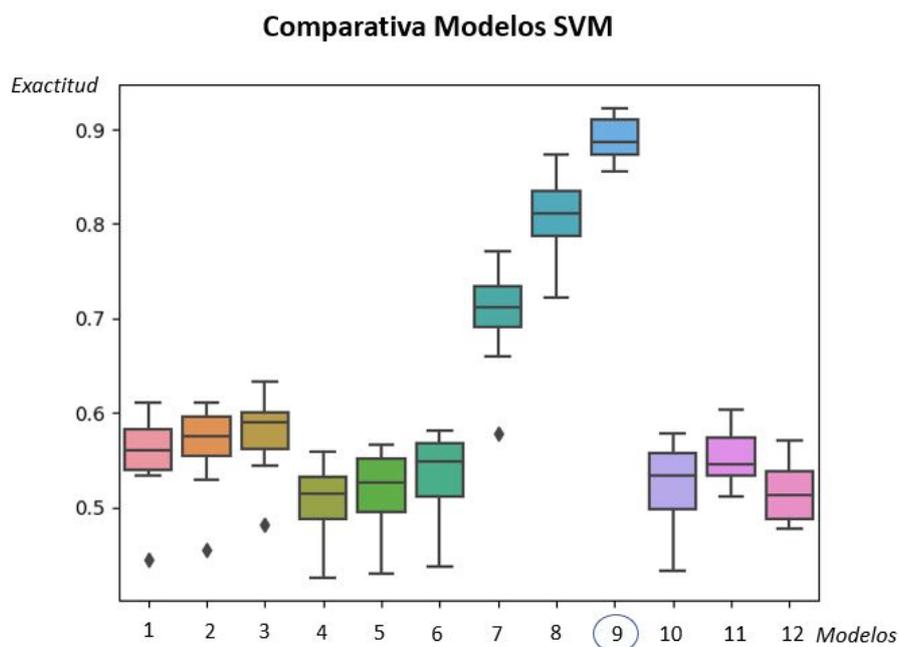


Figura 4.5: Boxplot comparativo de los modelos SVM

		MATRIZ DE CONFUSIÓN	
Reales	TEMBLOR	1355	25
	NO TEMBLOR	212	1108
		TEMBLOR	NO TEMBLOR

Predicciones

Figura 4.6: Matriz de confusión SVM

El *SVM* es el clasificador que peor exactitud presenta (91 %). Además es el que tiene mayor tasa de FP, quedando esto reflejado en el bajo valor de la precisión y de la especificidad.

4.1.4. Resumen de los resultados de los clasificadores tradicionales

Todos los modelos óptimos son capaces de clasificar correctamente más de un 90 % de la instancias del conjunto de prueba. Los resultados de las métricas del *SVM* son parecidos a los obtenidos por el *KNN*. Ambos clasifican mejor las señales de temblor y concentran la mayor parte de sus errores en la clasificación de señales de no temblor (FP). Sin embargo, los resultados del *KNN* son ligeramente superiores a los del *SVM* en términos de exactitud, precisión, sensibilidad y especificidad. Por su parte, el *Random Forest* es el clasificador que mejores valores de exactitud ha conseguido, y por tanto aquel que clasifica mejor un mayor número de instancias. El número de errores (FP y FN) está más repartido que en los otros clasificadores, aumentando así la precisión y especificidad, y disminuyendo la sensibilidad.

4.1.5. Resultados LSTM de clasificación

Para la evaluación de la LSTM de clasificación se crearon 9 modelos variando el *learning rate* (0.0001, 0.001 y 0.005) y el número de neuronas de cada capa de la LSTM (20, 35 y 50) (ver Subsección 3.3.4). Se entrenó cada uno de los modelos, se evaluó la función de pérdidas tanto en el conjunto de entrenamiento como en la etapa de validación, y finalmente cada uno de los modelos se testeó sobre el conjunto de prueba con los valores óptimos de los pesos de la red.

Todos los resultados de los modelos fueron satisfactorios con exactitudes superiores al 93 %. El modelo que mejor rendimiento ofreció fue la LSTM formada por 50 neuronas y un *learning rate* de 0.005. Al aumentar el número de neuronas de la LSTM se consigue dar más complejidad a la red, por lo que es capaz de adaptarse a más patrones de las señales de entrenamiento.

La función de pérdidas se va evaluando en cada época y se utiliza para actualizar los pesos de la red. En las primeras épocas la función de pérdidas tiene un valor de 0.7, y a medida que la red ve más datos se van optimizando los pesos de la red, y por tanto va disminuyendo la función de pérdidas. La etapa de validación evita el sobreajuste al conjunto de entrenamiento. El modelo final con los parámetros óptimos será aquel que presente un valor menor en la función de pérdidas dentro de la etapa de validación (ver Figura 4.7).

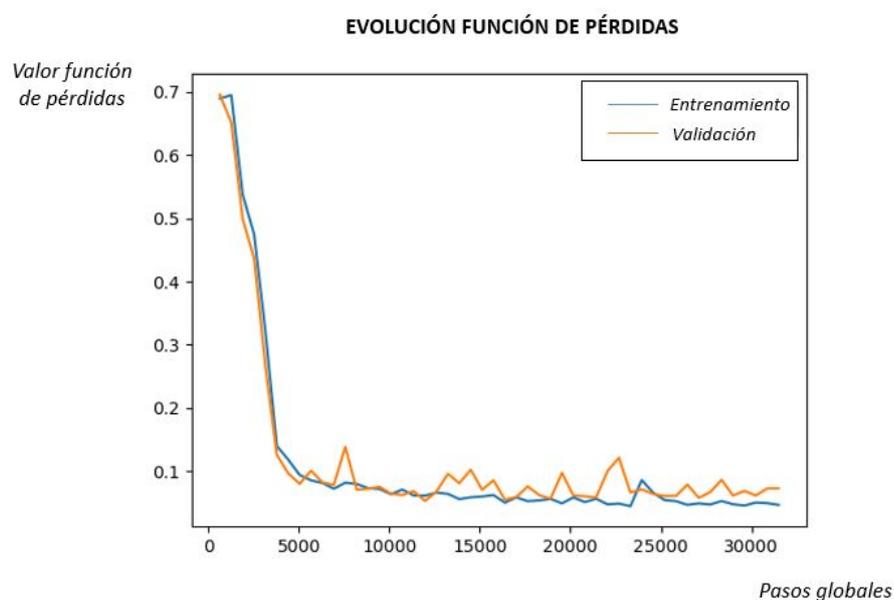


Figura 4.7: Evolución Función de Pérdidas

		MATRIZ DE CONFUSIÓN	
Reales	TEMBLOR	1282	38
	NO TEMBLOR	41	1339
		TEMBLOR	NO TEMBLOR
		Predicciones	

Figura 4.8: Matriz de confusión LSTM

A partir de la matriz de confusión se obtiene la exactitud, sensibilidad, precisión y especificidad (ver Tabla 4.7).

Los resultados de la LSTM de clasificación superan al *SVM* y al *KNN* en términos de exactitud, precisión y especificidad. Esto se debe a que el número de FP es menor. Los FN y FP están muy repartidos, lo que se traduce en valores de exactitud, precisión,

Métricas LSTM			
Exactitud	Precisión	Sensibilidad	Especificidad
97 %	97 %	97 %	97 %

Tabla 4.7: Métricas LSTM

sensibilidad y especificidad iguales.

4.2. Predicción de señales cinemáticas de temblor

Para la tarea de predicción de señales cinemáticas de temblor se variaron los parámetros especificados en la Sección 3.4 creando un total de 180 modelos.

Para organizar los resultados se compararon los distintos modelos en base al número de muestras de entrenamiento (30 y 50) y al número de muestras del horizonte de predicción (5, 10, 20, 30 y 50). Teniendo en cuenta que en cada modelo se varían el número de capas LSTM (1 y 2), el número de neuronas de cada capa (20, 35 y 50) y el valor del *learning rate* (0.0001, 0.0005 y 0.001), esto da lugar al estudio de 18 modelos para cada combinación entre el número de muestras de entrenamiento y el horizonte de predicción. En las Figuras 4.10 y 4.11 se muestran dos ejemplos de boxplots, pudiendo observarse el resto en el Anexo H.

Para cada boxplot se han extraído todos los modelos y se ha encontrado aquel que mejor rendimiento obtuvo en base a la mediana de la correlación entre las predicciones y las señales reales del conjunto de test. Cuanto más se aproxime a 1 la correlación mejores serán las predicciones realizadas por el modelo. También se muestran los valores de la media del MSE y la media del RMSE del mejor modelo para cada una de las situaciones. Esta información queda recogida en la Tabla 4.8.

En las Figuras del Anexo H se observan valores muy altos de correlación, lo que indica que los modelos han realizado una predicción satisfactoria de la señal cinemática del temblor. Se ve como a medida que aumenta el horizonte de predicción los errores en la predicción aumentan y la correlación disminuye. Los 180 modelos presentan unos resultados de correlación elevados por lo que se ajustan bastante bien a las señales de temblor.

Se observa de manera general una ligera mejora de los modelos con 2 capas LSTM con respecto a aquellos con 1 capa LSTM. De hecho, todos los modelos de la Tabla 4.8 tienen 2 capas LSTM. De la Tabla 4.8 se deduce que los modelos con un mayor número de neuronas (35 y 50) predicen mejor las señales que aquellos con 20 neuronas. El aumento del número de neuronas y de capas LSTM indica una tendencia hacia mejores resultados, y esto se debe a que aumenta la complejidad de red y el número de pesos a optimizar, dando lugar a una red capaz de ajustarse a patrones más complejos de las señales de temblor que aquellas redes con un menor número de neuronas y capas. Sin embargo, no se muestra una tendencia clara en el valor óptimo del *learning rate*, por lo que sería un parámetro a optimizar y varía en función del modelo.

Pese a todo, seguir aumentando el número de capas y neuronas no implicaría una mejora significativa de los resultados ya que es complicado conseguir una optimización de los parámetros mejor que la realizada por los modelos creados en este TFG. De hecho, un aumento excesivo de la complejidad de la red puede provocar la imposibilidad de encontrar los pesos y sesgos óptimos de la misma.

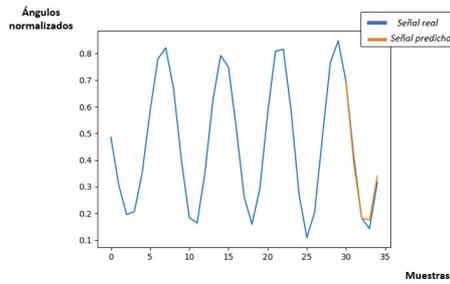
La mediana de la correlación varía desde valores del 0.9987 para las 5 primeras muestras (ver Figura 4.10) predichas hasta valores de 0.7563 para la predicción futura de 50 muestras (ver Figura 4.11). Los valores del MSE y RMSE aumentan con el aumento del horizonte de predicción acorde a las hipótesis realizadas (ver Sección 1.4).

En la Figura 4.9 se muestran 4 ejemplos de predicciones sobre distintas señales del conjunto de test (el resto de ejemplos se muestran en el Anexo G). En cada una de las figuras se han cambiado el número de muestras empleadas para entrenar y el número de muestras del horizonte de predicción. Las figuras con 50 muestras de entrenamiento han sido extraídas con el modelo formado por 2 capas LSTM, 35 neuronas y 0.0005 de *learning rate*. Por otro lado, las figuras con 30 muestras de entrenamiento han sido extraídas con el modelo formado por 2 capas LSTM, 35 neuronas y 0.001 de *learning rate*.

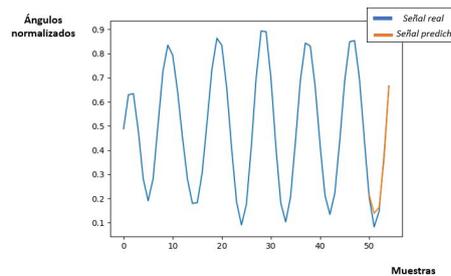
Se observa de manera gráfica como las predicciones empeoran cuanto mayor es el horizonte de predicción, sin embargo, los modelos realizan una predicción satisfactoria de las futuras 50 muestras, ya que incluso el último ciclo de temblor predicho no presenta apenas desfase con respecto al último ciclo de temblor de la señal real. La capacidad de predicción y los altos valores de correlación entre las señales predichas y las señales reales demuestran que la predicción por medio de redes LSTM es una solución idónea para la optimización de la técnica de estimulación de las vías aferentes en la estrategia *out-of-phase*,

Muestras entrenar	Muestras predecir	Modelo óptimo	Correlación	MSE ($^{\circ 2}$)	RMSE ($^{\circ}$)
30	5	2 Capas LSTM 50 Neuronas 0.0005 Lr	0.9982	0.00018	0.01135
30	10	2 Capas LSTM 35 Neuronas 0.001 Lr	0.9798	0.00203	0.03890
30	20	2 Capas LSTM 35 Neuronas 0.001 Lr	0.9007	0.01032	0.08898
30	30	2 Capas LSTM 50 Neuronas 0.001 Lr	0.8336	0.01687	0.11600
30	50	2 Capas LSTM 35 Neuronas 0.001 Lr	0.7176	0.03050	0.16330
50	5	2 Capas LSTM 50 Neuronas 0.0005 Lr	0.9982	0.00018	0.01135
50	10	2 Capas LSTM 35 Neuronas 0.0005 Lr	0.985	0.00160	0.03480
50	20	2 Capas LSTM 35 Neuronas 0.001 Lr	0.9210	0.00840	0.08200
50	30	2 Capas LSTM 35 Neuronas 0.001 Lr	0.8707	0.01400	0.10800
50	50	2 Capas LSTM 35 Neuronas 0.0005 Lr	0.7563	0.02516	0.14620

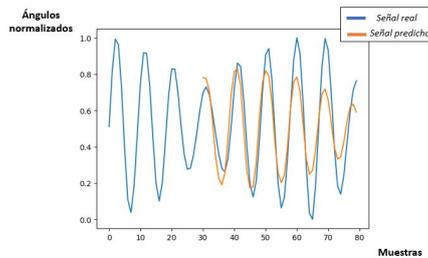
Tabla 4.8: Mejores modelos de predicción



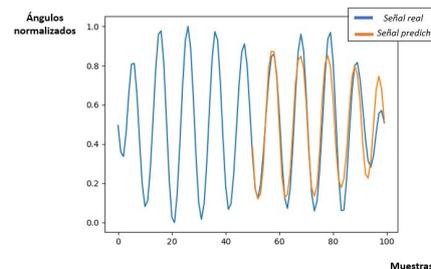
(a) 30 de Entrenamiento y 5 de Predicción



(b) 50 de Entrenamiento y 5 de Predicción



(c) 30 de Entrenamiento y 50 de Predicción



(d) 50 de Entrenamiento y 50 de Predicción

Figura 4.9: Ejemplos de predicciones

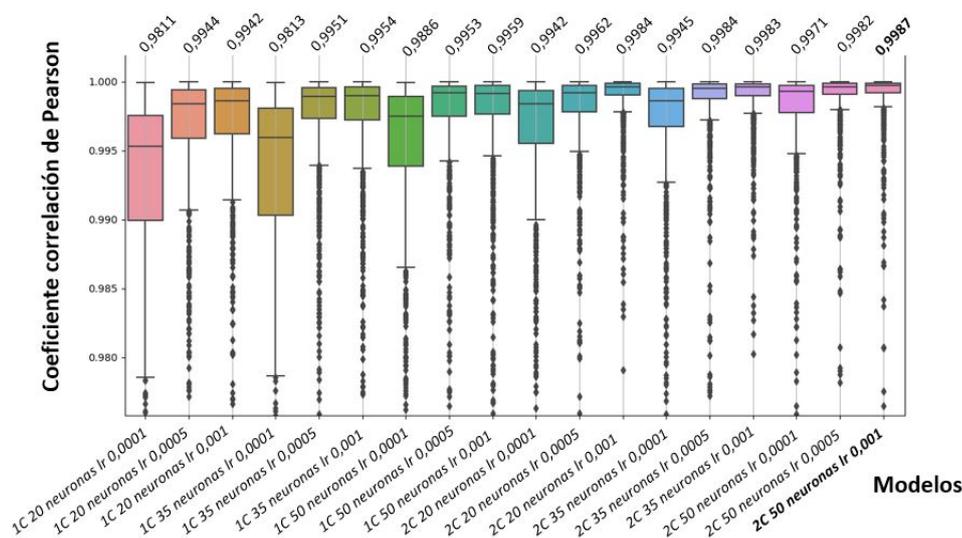


Figura 4.10: Boxplot con 50 muestras de entrenamiento / 5 de predicción

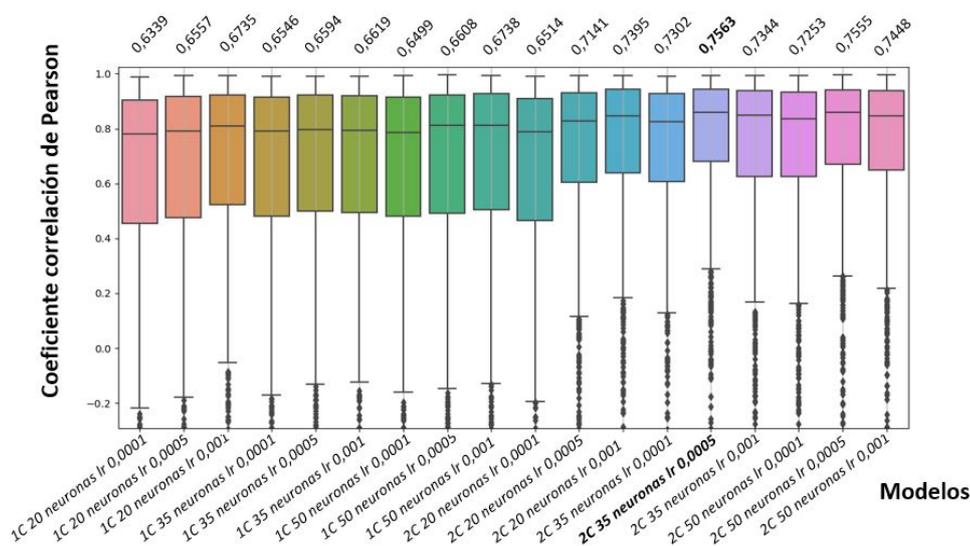


Figura 4.11: Boxplot con 50 muestras de entrenamiento / 50 de predicción

4.3. Discusión

En este TFG se han implementado y evaluado algoritmos tanto de clasificación como de predicción. Se obtuvieron resultados de 6 modelos de KNN, 18 modelos de Random Forest, 12 modelos de SVM y 9 modelos de redes neuronales LSTM para la tarea de clasificación. Para la tarea de predicción se evaluaron un total de 180 modelos basados en redes neuronales LSTM. Se obtuvo el mejor modelo para cada tipo de clasificador y para la predicción. Tanto la tarea de clasificación como la de predicción tienen como objetivo optimizar la estrategia *out-of-phase* en la estimulación de las vías

aferentes. Los clasificadores se utilizarían para detectar la existencia de temblor en el paciente. En caso de que se detecte temblor por medio de los clasificadores, se usarían los predictores para predecir los ciclos futuros y estimular de manera sincronizada a la actividad tremorogénica.

Los valores de exactitudes de la clasificación varían desde el 91 % en el SVM hasta el 97 % en la LSTM. Tanto el KNN como el SVM presentan valores de sensibilidad más altos, de manera que casi siempre que el paciente presentase temblor se estimularía a través de la estrategia *out-of-phase* en el tratamiento de estimulación de las vías aferentes. Pese a que tengan muy pocos FN no son los modelos más adecuados ya que su exactitud, precisión y especificidad son más bajas en comparación con la red LSTM y el Random Forest. Esto implica que en momentos en los que el paciente no presentase temblor se estaría estimulando sin necesidad, afectando de manera negativa a la reducción del temblor. El Random Forest y la LSTM presentan unos resultados más equilibrados, aumentando mucho la precisión y la especificidad con respecto al KNN y al SVM. Esto supone que se reducirían el número de estimulaciones realizadas en los momentos en los que el paciente no presentase temblor. La comparación entre el Random forest y la LSTM indica que es mejor utilizar la LSTM, al mostrar ésta una ligera mejora en términos de exactitud, sensibilidad y especificidad con respecto al Random Forest. Pese a todo, las diferencias entre modelos son mínimas y todos presentan unos resultados de clasificación satisfactorios con exactitudes superiores al 90 %.

Una vez realizada la clasificación del temblor y obtenida la decisión de si es necesario estimular o no, se entra en la etapa de predicción de los ciclos futuros de temblor. En la Tabla 4.8 se observa que cuanto menor es el horizonte de predicción mejores son los valores de la correlación, el MSE y el RMSE. Sin embargo, una predicción de 5 muestras futuras no aporta mucha información de la señal futura. Un horizonte de predicción de 50 muestras con una frecuencia de muestreo de 50 Hz se corresponde con 1 segundo en el futuro, y pese a tener un valor de correlación menor aporta más información. Esta correlación permite sincronizar las estimulaciones con la actividad tremorogénica del paciente para conseguir una mayor reducción del temblor. El modelo más óptimo por tanto sería aquel con 2 capas LSTM, 35 neuronas por capa y un *learning rate* de 0.0005 ya que es el modelo con mayor correlación dentro de aquellos con un horizonte de predicción de 1 segundo. Es el más óptimo porque ofrece una ventana temporal mayor para realizar las estimulaciones de manera sincronizada en la técnica de estimulación de las vías aferentes.

Tanto los resultados de clasificación como de predicción muestran la posibilidad de optimizar el control de las estimulaciones en la técnica de estimulación de las vías aferentes para conseguir una mayor supresión del temblor patológico.

Capítulo 5

Conclusiones y Líneas Futuras

En este capítulo se resumen las conclusiones más importantes del proyecto y los resultados de este TFG, así como se proponen líneas futuras de investigación sobre el mismo.

5.1. Conclusiones

Dentro de las estrategias empleadas en la Estimulación de las Vías Aferentes destaca el *out-of-phase*. La desventaja fundamental de la estrategia *out-of-phase* es que la predicción de futuros ciclos de temblor basada en la frecuencia del mismo puede dar lugar a errores en la sincronización de la estimulación. Este problema se ve solucionado con un procesamiento de la señal basado en el *machine learning* y redes neuronales, superando sus limitaciones y mejorando la aplicación de las estimulaciones de las vías aferentes.

La implementación de algoritmos de clasificación permiten detectar la presencia de actividad tremorogénica para decidir si estimular las vías aferentes. La implementación de algoritmos de predicción permiten predecir los futuros ciclos de temblor para superar así las limitaciones propias del *out-of-phase*, relativas a la desincronización de las estimulaciones con la actividad tremorogénica. Se consigue un aumento de la eficiencia en la neuromodulación y una mayor supresión del temblor patológico.

La creación de dos datasets, uno para la clasificación y otro para la predicción, que contienen las variaciones de los ángulos de flexo-extensión de muñeca permiten la implementación tanto de algoritmos de clasificación como de predicción con un limitado procesamiento de la señal y sin necesidad de extraer características de la misma (técnica más empleada en el estado del arte).

Dentro de los clasificadores la red LSTM de clasificación mostró un rendimiento superior al resto de modelos con valores de exactitud, precisión, sensibilidad y especificidad del 97%. Para la tarea de predicción se eligió como el mejor modelo la red neuronal de 2 capas LSTM, 35 neuronas y un *learning rate* de 0.0005 porque presentó un mejor rendimiento para la predicción de 50 muestras futuras (1 segundo). Existen otros modelos con valores de correlación más altos. Sin embargo, estos modelos tienen un horizonte de predicción más pequeño, limitando la información futura de la señal a 100 ms (5 muestras), 200 ms (10 muestras), 400 ms (20 muestras) o 600 ms (30 muestras).

Se han cumplido los objetivos e hipótesis realizadas tanto en la tarea de clasificación como en la tarea de predicción, mostrando este proyecto como una posible forma de optimizar la estrategia *out-of-phase* en la estimulación de las vías aferentes.

5.2. Líneas futuras

Las líneas futuras de investigación consistirían por una parte en mejorar los resultados de clasificación y de predicción, y por otro lado en extender este TFG para conseguir la supresión del temblor de otras articulaciones u otro tipo de movimientos:

- El uso de Redes Neuronales Convolucionales capaces de procesar datos como series temporales, pueden ser una alternativa más a probar en la búsqueda de mejores resultados en la clasificación de señales de temblor.
- El estudio se ha realizado sobre la variación de los ángulos de flexo-extensión de muñeca, sin embargo, el temblor patológico de los pacientes con TE no se limita ni a la flexo-extensión ni a la muñeca (pese a que este sea el más marcado). Por lo tanto, se podría realizar el mismo estudio para diferentes rotaciones y articulaciones superiores.
- El dataset está formado en base a los datos extraídos de 12 pacientes diferentes. Esto implica que tanto la red LSTM de predicción como la de clasificación hayan ido actualizando sus pesos y encontrando los parámetros óptimos en función de las señales de los 12 pacientes. Sin embargo, un dataset de 1 único paciente permitiría que la red aprendiese patrones en base a las señales propias de ese paciente, convirtiéndola en una red mucho más específica y por tanto dando mejores resultados para las señales de dicho paciente. Sería necesario extraer una cantidad suficiente de datos para poder llevar a cabo una tarea de DL como la reflejada en este proyecto.
- Finalmente, otra posible mejora sería combinar las señales cinemáticas de temblor con señales electromiográficas de temblor. La combinación de ambas señales aportaría más información a las redes LSTM permitiéndoles desempeñar una mejor labor tanto en las tareas de clasificación como en las tareas de predicción.

Bibliografía

- [1] R. C. Helmich, I. Toni, G. Deuschl, and B. R. Bloem. The pathophysiology of essential tremor and Parkinson's tremor. *Curr Neurol Neurosci Rep*, 13(9):378, Sep 2013.
- [2] E. D. Louis, M. Bares, J. Benito-Leon, S. Fahn, S. J. Frucht, J. Jankovic, W. G. Ondo, P. K. Pal, and E. K. Tan. Essential tremor-plus: a controversial new concept. *Lancet Neurol*, 19(3):266–270, 03 2020.
- [3] J. A. Gallego, J. L. Dideriksen, A. Holobar, J. Ibáñez, J. L. Pons, E. D. Louis, E. Rocon, and D. Farina. Influence of common synaptic input to motor neurons on the neural drive to muscle in essential tremor. *J Neurophysiol*, 113(1):182–191, Jan 2015.
- [4] V. Castrillo-Fraile, E. C. Peña, J. M. T. Gabriel Y Galán, P. D. Delgado-López, C. Collazo, and E. Cubo. Tremor Control Devices for Essential Tremor: A Systematic Literature Review. *Tremor Other Hyperkinet Mov (N Y)*, 9, 2019.
- [5] O. Tysnes and A. Storstein. Epidemiology Of Parkinson's Disease. *J Neural Trans*, 17:901–905, 2017.
- [6] J. A. Gallego, J. L. Dideriksen, A. Holobar, J. Ibáñez, V. Glaser, J. P. Romero, J. Benito-León, J. L. Pons, E. Rocon, and D. Farina. The phase difference between neural drives to antagonist muscles in essential tremor is associated with the relative strength of supraspinal and afferent input. *J Neurosci*, 35(23):8925–8937, Jun 2015.
- [7] E.D. and Haines and A. G. and Mihailoff. *Principios de Neurociencia*. 2019.
- [8] M. Plumb and P. Bain. *Essential Tremor: The facts*. Oxford University Press USA, 2006.
- [9] Pedro Emilio Bermejo Velasco. Tratamiento del temblor esencial. *Medicina Clínica*, 129(6):222–227, 2007.
- [10] A. Gironell. Therapeutical strategies for essential tremor. *Med Clin (Barc)*, 129(16):632–637, Nov 2007.
- [11] Carmen Gasca-Salas. Treatment of cognitive impairment in parkinson disease. *Medicina Clínica*, 148:232–237, 2017.
- [12] P. Hedera, F. Cibulčík, and T. L. Davis. Pharmacotherapy of essential tremor. *J Cent Nerv Syst Dis*, 5:43–55, Dec 2013.
- [13] A. Puschmann and Z. K. Wszolek. Diagnosis and treatment of common forms of tremor. *Semin Neurol*, 31(1):65–77, Feb 2011.
- [14] 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. Martínez-Castrillo, I. Martínez-Torres, V. Puente, A. Sesar, F. Valldeoriola-Serra, and R. Yañez.

- Enfermedad de parkinson avanzada. características clínicas y tratamiento (parte i). *Neurología*, 28(8):503–521, 2013.
- [15] S. Perlmutter and W. Jonathan. Deep brain stimulation. *Annual Review of neuroscience*, 29:229–57, 2006.
- [16] S. Paris-Robidas, E. Brochu, M. Sintès, V. Emond, M. Bousquet, M. Vandal, M. Pilote, C. Tremblay, T. Di Paolo, A. H. Rajput, A. Rajput, and F. Calon. Defective dentate nucleus GABA receptors in essential tremor. *Brain*, 135(Pt 1):105–116, Jan 2012.
- [17] Claudia Lucia Moreno Lucia. Efectividad de la estimulación cerebral profunda del núcleo subtalámico en pacientes con enfermedad de Parkinson: experiencia en Antioquia. *Acta Neurología Colombiana*, 30:139 – 140, 07 2014.
- [18] Kalia Suneil and Andrés Lozano. Deep brain stimulation for parkinson’s disease and other movement disorders. *Current opinion in neurology*, 26:374–80, 8 2013.
- [19] G. Deuschl, J. Raethjen, H. Hellriegel, and R. Elble. Treatment of patients with essential tremor. *The Lancet Neurology*, 10(2):148–161, 2011.
- [20] R. F. Dallapiazza, D. J. Lee, P. De Vloo, A. Fomenko, C. Hamani, M. Hodaie, S. K. Kalia, A. Fasano, and A. M. Lozano. Outcomes from stereotactic surgery for essential tremor. *J Neurol Neurosurg Psychiatry*, 90(4):474–482, 04 2019.
- [21] D. Bendersky, P. Ajler, and C. Yampolsky. El uso de la neuromodulación para el tratamiento del temblor. *Surgical neurology international*, 5:232–46, 2014.
- [22] 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. *Neurología*, 2020.
- [23] S.Moosa, R.Martínez-Fernández, WJ.Elias, M., H. Eisenberg, and PS. Fishman. The role of high-intensity focused ultrasound as a symptomatic treatment for parkinson’s disease. 2019.
- [24] M. Rohani and A. Fasano. Focused Ultrasound for Essential Tremor: Review of the Evidence and Discussion of Current Hurdles. *Tremor Other Hyperkinet Mov (N Y)*, 7:462, 2017.
- [25] S. R. Schreglmann, S. Hägele-Link, B. Werner, E. Martin, and G. Kägi. Focused ultrasound ablation as tremor treatment. *Nervenarzt*, 89(6):674–681, Jun 2018.
- [26] S. Dosen, S. Muceli, J. L. Dideriksen, J. P. Romero, E. Rocon, J. Pons, and D. Farina. Online tremor suppression using electromyography and low-level electrical stimulation. *IEEE Trans Neural Syst Rehabil Eng*, 23(3):385–395, May 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, Sep 1999.
- [30] L. Popović Maneski, N. Jorgovanović, V. Ilić, S. Došen, T. Keller, M. B. Popović, and D. B. Popović. Electrical stimulation for the suppression of pathological tremor. *Med Biol Eng Comput*, 49(10):1187–1193, Oct 2011.
- [31] G. Grimaldi, S. Camut, and M. Manto. Functional electrical stimulation effect on upper limb tremor. *Int. J. Bioelectromagn*, 13:123–124, 2011.
- [32] J. Á. Gallego, E. Rocon, J. M. Belda-Lois, and J. L. Pons. A neuroprosthesis for tremor management through the control of muscle co-contraction. *J Neuroeng Rehabil*, 10:36, Apr 2013.
- [33] A. P. Bó, C. Azevedo-Coste, C. Geny, P. Poignet, and C. Fattal. On the use of fixed-intensity functional electrical stimulation for attenuating essential tremor. *Artif Organs*, 38(11):984–991, Nov 2014.
- [34] N. A. Maffiuletti. Physiological and methodological considerations for the use of neuromuscular electrical stimulation. *Eur J Appl Physiol*, 110(2):223–234, Sep 2010.
- [35] C. S. Bickel, C. M. Gregory, and J. C. Dean. Motor unit recruitment during neuromuscular electrical stimulation: a critical appraisal. *Eur J Appl Physiol*, 111(10):2399–2407, Oct 2011.
- [36] A. Pascual-Valdunciel, F. O. Barroso, and J. L. Pons. Motor inhibition elicited by electrical stimulation of afferent pathways and its application in tremor suppression.
- [37] E. R. Kandel, J. H. Schwartz, T. M. Jessell, S. A. Siegelbaum, and A. J. Hudspeth. *Principles of Neural Science*. 5 edition, 2013.
- [38] J. L. Dideriksen, C. M. Laine, S. Dosen, S. Muceli, E. Rocon, J. L. Pons, J. Benito-Leon, and D. Farina. Electrical Stimulation of Afferent Pathways for the Suppression of Pathological Tremor. *Front Neurosci*, 11:178, 2017.
- [39] A. Pascual-Valdunciel, M. Gonzalez-Sanchez, S. Muceli, B. Adan-Barrientos, V. Escobar-Segura, J. R. Perez-Sanchez, M. K. Jung, A. Schneider, K. P. Hoffmann, J. C. Moreno, F. Grandas, D. Farina, J. L. Pons, and F. O. Barroso. Intramuscular Stimulation of Muscle Afferents Attains Prolonged Tremor Reduction in Essential Tremor Patients. *IEEE Trans Biomed Eng*, 68(6):1768–1776, Jun 2021.
- [40] Aurélien Géron. *Aprende Machine Learning con Scikit-Learn, Keras y Tensorflow*. O’reilly, 2 edition, 2019.
- [41] Trevor Hastie, Robert Tibshirani, and Jerome Friedman. *The Elements of Statistical Learning*. 2 edition, 2017.

-
- [42] Andriy Burkov. *The Hundred-Page Machine Learning Book*. 2019.
- [43] Sakib Zargar. Introduction to sequence learning models: Rnn, lstm, gru. 04 2021.
- [44] Guangyu Robert Yang and Xiao-Jing Wang. Artificial neural networks for neuroscientists: A primer. *Neuron*, 107(6):1048–1070, 2020.
- [45] Feng-Lei Fan, Jinjun Xiong, Mengzhou Li, and Ge Wang. On interpretability of artificial neural networks: A survey. *IEEE Transactions on Radiation and Plasma Medical Sciences*, pages 1–1, 2021.
- [46] Francisco Alonso. *Redes Neuronales y Deep Learning*. 2 edition.
- [47] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. *Deep Learning*. MIT Press, 2016. <http://www.deeplearningbook.org>.
- [48] Pedro Carvalho, Nuno Lourenço, and Penousal Machado. Evolving learning rate optimizers for deep neural networks, 2021.
- [49] Changyong Yu, Xin Qi, Haitao Ma, Xin He, Cuirong Wang, and Yuhai Zhao. Llr: Learning learning rates by lstm for training neural networks. *Neurocomputing*, 394:41–50, 2020.
- [50] Christian Bakke Vennerød, Adrian Kjærran, and Erling Stray Bugge. Long short-term memory rnn, 2021.
- [51] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.
- [52] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.
- [53] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks, 2013.
- [54] Justin Simon Bayer. Learning sequence representations. 2015.
- [55] J. Stamatakis, J. Ambroise, J. Crémers, H. Sharei, V. Delvaux, B. Macq, and G. Garraux. Finger tapping clinimetric score prediction in Parkinson’s disease using low-cost accelerometers. *Comput Intell Neurosci*, 2013:717853, 2013.
- [56] H. Jeon, W. Lee, H. Park, H. J. Lee, S. K. Kim, H. B. Kim, B. Jeon, and K. S. Park. Correction: Automatic Classification of Tremor Severity in Parkinson’s Disease Using a Wearable Device. *Sensors* 2017, 17, 2067. *Sensors (Basel)*, 18(1), 12 2017.
- [57] R. LeMoyné, N. Tomycz, T. Mastroianni, C. McCandless, M. Cozza, and D. Peduto. Implementation of a smartphone wireless accelerometer platform for establishing deep brain stimulation treatment efficacy of essential tremor with machine learning. *Annu Int Conf IEEE Eng Med Biol Soc*, 2015:6772–6775, 2015.
- [58] Md Nafiul Alam, Benjamin Johnson, Jeffrey Gendreau, Kouhyar Tavakolian, Colin Combs, and Reza Fazel-Rezai. Tremor quantification of parkinson’s disease - a pilot study. pages 0755–0759, 2016.

- [59] 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. *Frontiers in Bioengineering and Biotechnology*, 8:778, 2020.
- [60] Rafael Anicet Zanini, Esther Luna Colombini, and Maria Claudia Ferrari de Castro. Parkinson's disease emg signal prediction using neural networks. pages 2446–2453, 2019.
- [61] 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. pages 5996–6000, 2020.
- [62] S. Shahtalebi, S. F. Atashzar, O. Samotus, R. V. Patel, M. S. Jog, and A. Mohammadi. PHTNet: Characterization and Deep Mining of Involuntary Pathological Hand Tremor using Recurrent Neural Network Models. *Sci Rep*, 10(1):2195, 02 2020.
- [63] Raúl González Duque. *Python para todos*.
- [64] Eli Stevens, Luca Antiga, and Thomas Viehmann. *Deep learning with PyTorch*. Manning, 2020.
- [65] K. P. Bhatia, P. Bain, N. Bajaj, R. J. Elble, M. Hallett, E. D. Louis, J. Raethjen, M. Stamelou, C. M. Testa, and G. Deuschl. 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, 01 2018.
- [66] Juan Vignolo Barchiesi. *Introducción al Procesamiento Digital de Señales*. 2008.
- [67] A. Oppenheim and A. Willsky. *Signals and Systems*. MIT, 2 edition, 1997.
- [68] D. Percival and A. Walden. *Spectral Analysis for Physical Applications*. Cambridge University Press, 1993.
- [69] Jaime Carrillo, Carmen Pérez, Rafael Sendra, Gema García, M.Helena Hernando, and Álvaro Gutiérrez. Long short-term memory neural network for glucose prediction. *Neural Computing and Applications*, 33:374–80, 8 2020.

Apéndice A

Aspectos éticos, económicos, sociales y ambientales

El TE es un desorden neurológico que causa la aparición de movimientos involuntarios de una parte del cuerpo o extremidad como consecuencia de oscilaciones neurales patológicas que se proyectan hacia los músculos. Dichos movimientos pueden llegar a afectar de manera severa a la realización de actividades de la vida diaria. La aparición de nuevas técnicas como la estimulación de las vías aferentes nace como una alternativa a los tratamientos convencionales. La optimización de la técnica de estimulación de las vías aferentes puede suponer una mayor neuromodulación y reducción del temblor patológico, siendo este el objetivo de este TFG.

- **Aspectos sociales:** Este proyecto tiene un impacto social directo sobre los pacientes con Temblor Esencial y sus familiares. Los pacientes podrán tener un tratamiento más optimizado y con mejores resultados. Todo ello, derivará en una mejora de la calidad de vida tanto de los pacientes como de sus familiares.
- **Aspectos éticos y profesionales:** Los datos de los pacientes utilizados de este proyecto están amparados bajo el Reglamento general de protección de datos (GDPR) de la Unión Europea. Estos fueron utilizados exclusivamente para solucionar un problema en ámbito de la salud y potenciar los tratamientos de los pacientes.
- **Aspectos económicos:** Este proyecto tiene un impacto económico tanto en el paciente como en el sistema nacional de salud. Los gastos del sistema nacional de salud y del paciente en tratamientos convencionales como medicamentos disminuye en favor de un tratamiento más barato como es la estimulación de las vías aferentes.
- **Aspectos ambientales:** Este proyecto también tiene un impacto ambiental, ya que la técnica de estimulación de las vías aferentes es una técnica sostenible y apta para todos los pacientes.

Apéndice B

Presupuesto económico

Este proyecto se ha desarrollado a lo largo de 6 meses en colaboración con el Grupo de Neurorrehabilitación del CSIC-Instituto Cajal. Los datos fueron aportados por el proyecto europeo EXTEND - Bidirectional Hyper-Connected Neural System en el cual participa el propio CSIC. Se ha llevado a cabo una aproximación del presupuesto económico necesario para la realización de este proyecto teniendo en cuenta costes humanos, de software y equipamiento técnico.

- **Personal:** Esta sección del presupuesto considera los salarios del personal que se ha visto involucrado en el proyecto: Director de Proyecto (Ingeniero) y el estudiante de ingeniería autor de este TFG (ver Tabla B.1). (ver Tabla B.1).

	Coste horario (€)	Horas	Total (€)
Director del trabajo	25	85	2.125
Estudiante de ingeniería	15	600	9.000
TOTAL			11.125

Tabla B.1: Costes de personal.

- **Costes de recursos materiales:** Esta sección tiene en cuenta el coste del software y el equipamiento técnico utilizado para el desarrollo de este proyecto. El coste total ha sido calculado teniendo en cuenta la devaluación de los bienes materiales usados (ver Tabla B.2).

	Tiempo de vida (años)	Uds.	Coste (€)	Amortización (€/mes)	Uso (meses)	Total (€)
Ordenador personal GTX	5	1	1000	16,66	4	66,66
1080 Ti	4	1	1000	20.83	4	83,32
	1					
TOTAL						149,98

Tabla B.2: Costes de recursos materiales.

	Coste
Costes de personal	11.125€
Costes de material	149,98€
Subtotal	11.274,98€
IVA	2.367,74€
Total	13.642,72€

Tabla B.3: Costes totales.

Apéndice C

Modelos de clasificación descartados

En este apéndice se muestran los distintos modelos (con sus respectivos parámetros) que fueron descartados debido a la existencia de otro modelo más óptimo (los presentados en los resultados del TFG) que obtuvo mejores resultados en las métricas a evaluar.

C.1. KNN

Un total de 5 modelos del clasificador *KNN* fueron descartados (ver Tabla C.1).

MODELO	Número de vecinos	Weights	Algoritmo
<i>Modelo 1</i>	6	Uniform	Ball Tree
<i>Modelo 3</i>	12	Uniform	Ball Tree
<i>Modelo 4</i>	12	Distance	Ball Tree
<i>Modelo 5</i>	20	Uniform	Ball Tree
<i>Modelo 6</i>	20	Distance	Ball Tree

Tabla C.1: Modelos descartados del KNN

C.2. Random Forest

Un total de 17 modelos del clasificador *Random Forest* fueron descartados (ver Tabla C.2).

MODELO	Número de árboles	Criterion	Número de características
<i>Modelo 1</i>	100	Gini	Auto
<i>Modelo 2</i>	100	Gini	Sqrt
<i>Modelo 3</i>	100	Gini	Log2
<i>Modelo 4</i>	100	Entropy	Auto
<i>Modelo 6</i>	100	Entropy	Log2
<i>Modelo 7</i>	200	Gini	Auto
<i>Modelo 8</i>	200	Gini	Sqrt
<i>Modelo 9</i>	200	Gini	Log2
<i>Modelo 10</i>	200	Entropy	Auto
<i>Modelo 11</i>	200	Entropy	Sqrt
<i>Modelo 12</i>	200	Entropy	Log2
<i>Modelo 13</i>	50	Gini	Auto
<i>Modelo 14</i>	50	Gini	Sqrt
<i>Modelo 15</i>	50	Gini	Log2
<i>Modelo 16</i>	50	Entropy	Auto
<i>Modelo 17</i>	50	Entropy	Sqrt
<i>Modelo 18</i>	50	Entropy	Log2

Tabla C.2: Modelos descartados Random Forest

C.3. SVM

Un total de 11 modelos del clasificador *SVM* fueron descartados (ver Tabla C.3).

MODELOS	C	Kernel
<i>Modelo 1</i>	0.1	Lineal
<i>Modelo 2</i>	1	Lineal
<i>Modelo 3</i>	10	Lineal
<i>Modelo 4</i>	0.1	Polinomial
<i>Modelo 5</i>	1	Polinomial
<i>Modelo 6</i>	10	Polinomial
<i>Modelo 7</i>	0.1	Rbf
<i>Modelo 8</i>	1	Rbf
<i>Modelo 10</i>	0.1	Sigmoide
<i>Modelo 11</i>	1	Sigmoide
<i>Modelo 12</i>	10	Sigmoide

Tabla C.3: Modelos descartados de la SVM

Apéndice D

Modelos descartados de la LSTM de clasificación

En este apéndice se muestran los distintos modelos de redes neuronales LSTM (con sus respectivos parámetros) que fueron descartados debido a la existencia de otro modelo más óptimo (el presentado en los resultados del TFG) que obtuvo mejores resultados en las métricas a evaluar. **Modelo 2 LSTM de clasificación**

Los parámetros del modelo 2 de la LSTM de clasificación son los siguientes: Número de neuronas = 20, learning rate = 0.0001.

Real temblor	1309	11
Real no temblor	114	1266
	Predicción temblor	Predicción no temblor

Tabla D.1: Matriz de confusión Modelo 2

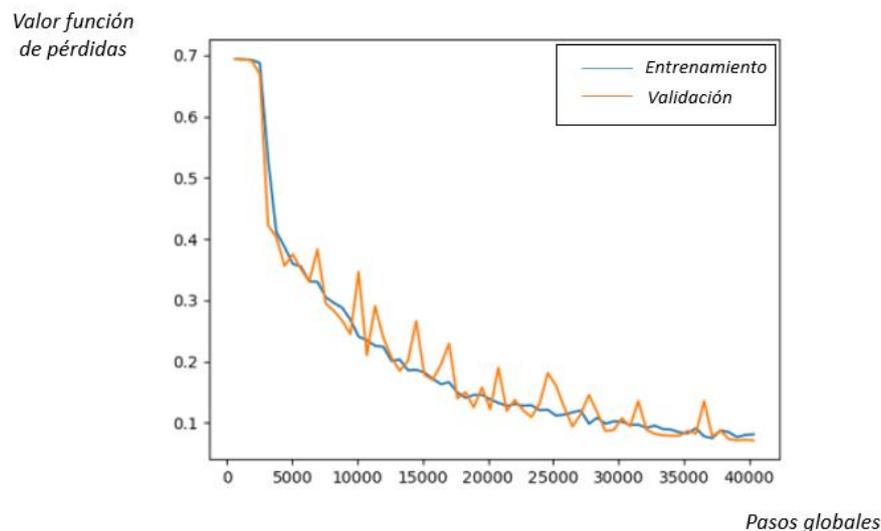


Figura D.1: Función de pérdidas del modelo 2

Modelo 3 LSTM de clasificación

Los parámetros del modelo 3 de la LSTM de clasificación son los siguientes:
Número de neuronas = 20, learning rate = 0.001.

Real temblor	1132	188
Real no temblor	1	1379
	Predicción temblor	Predicción no temblor

Tabla D.2: Matriz de confusión Modelo 3

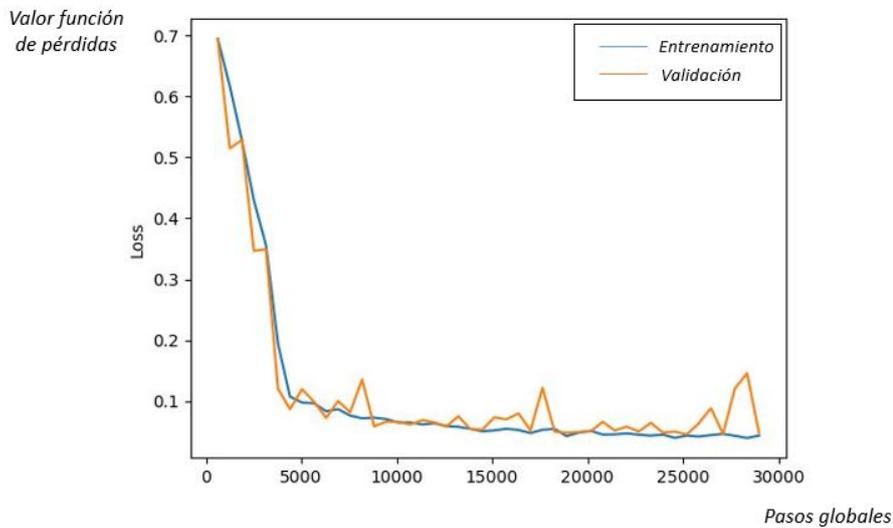


Figura D.2: Función de pérdidas del modelo 3

Modelo 4 LSTM de clasificación

Los parámetros del modelo 4 de la LSTM de clasificación son los siguientes:
Número de neuronas = 20, learning rate = 0.005.

Real temblor	1260	60
Real no temblor	19	1361
	Predicción temblor	Predicción no temblor

Tabla D.3: Matriz de confusión Modelo 4

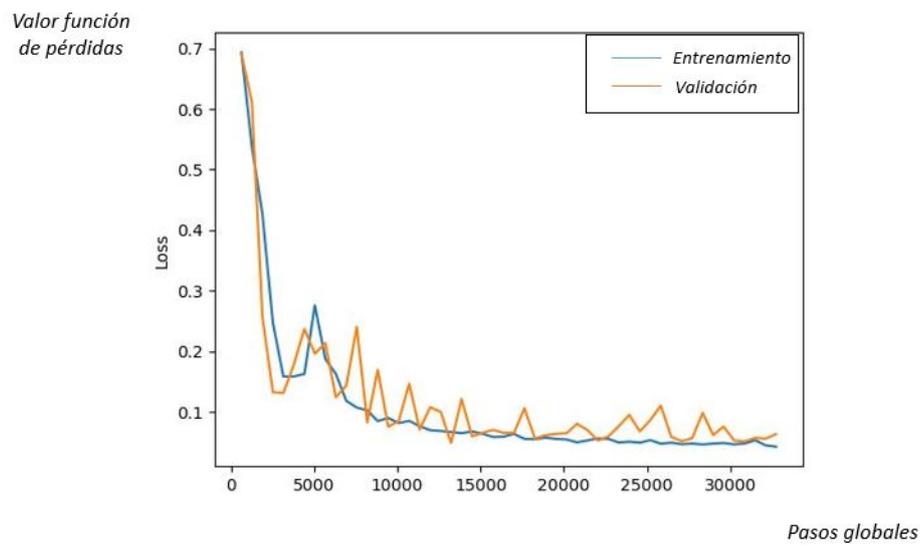


Figura D.3: Función de pérdidas del modelo 4

Modelo 5 LSTM de clasificación

Los parámetros del modelo 5 de la LSTM de clasificación son los siguientes: Número de neuronas = 35, learning rate = 0.0001.

Real temblor	1297	23
Real no temblor	84	1296
	Predicción temblor	Predicción no temblor

Tabla D.4: Matriz de confusión Modelo 5

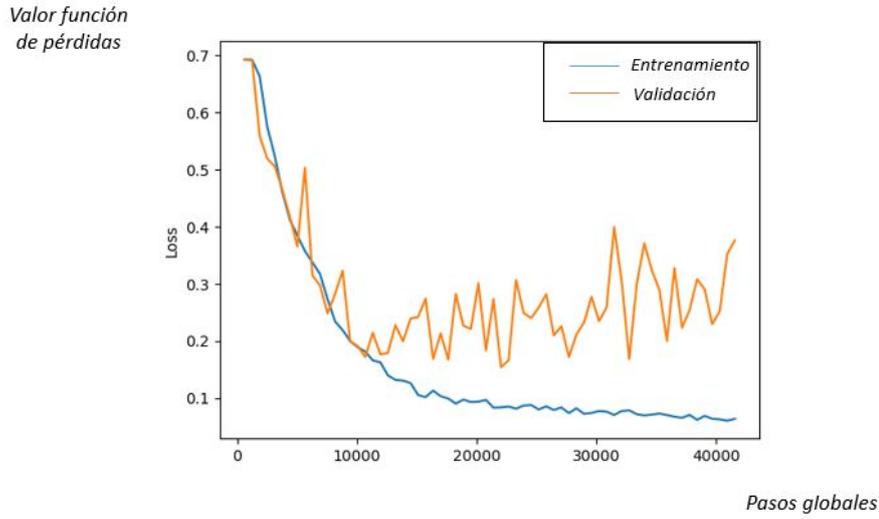


Figura D.4: Función de pérdidas del modelo 5

Modelo 6 LSTM de clasificación

Los parámetros del modelo 6 de la LSTM de clasificación son los siguientes:
 Número de neuronas = 35, learning rate = 0.001.

Real temblor	1317	3
Real no temblor	110	1270
	Predicción temblor	Predicción no temblor

Tabla D.5: Matriz de confusión Modelo 6

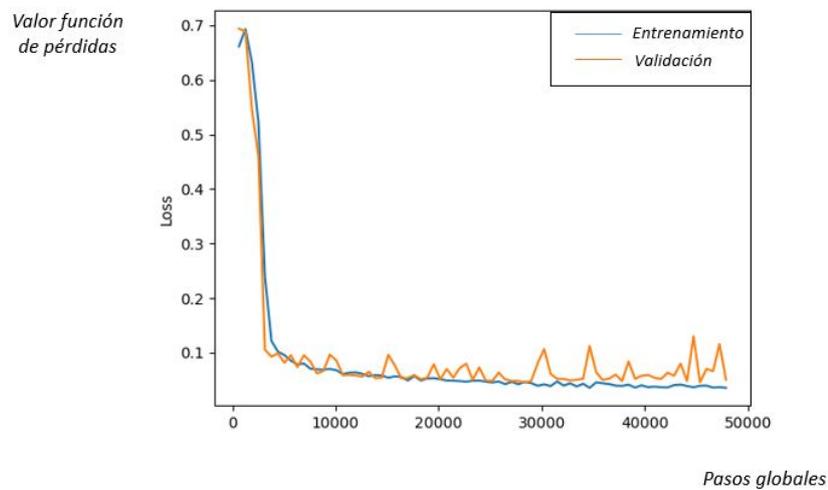


Figura D.5: Función de pérdidas del modelo 6

Modelo 7 LSTM de clasificación

Los parámetros del modelo 7 de la LSTM de clasificación son los siguientes:
 Número de neuronas = 35, learning rate = 0.005.

Real temblor	1167	153
Real no temblor	1	1379
	Predicción temblor	Predicción no temblor

Tabla D.6: Matriz de confusión Modelo 7

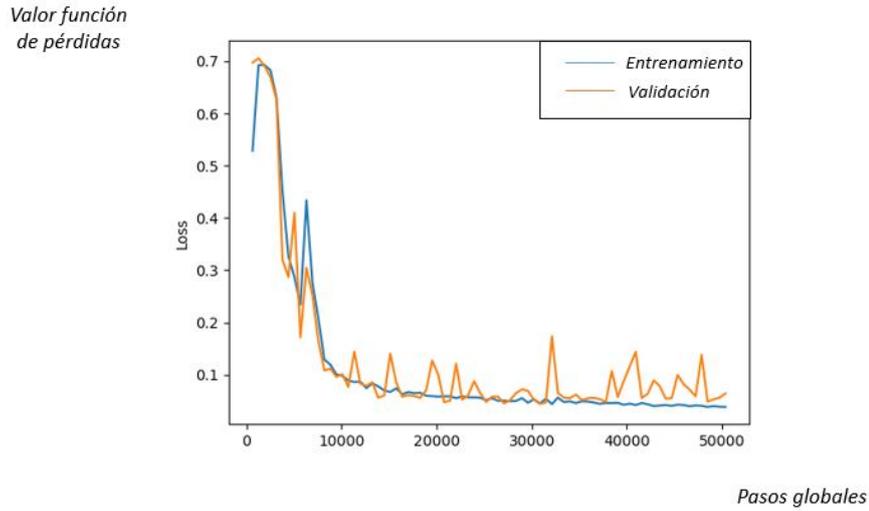


Figura D.6: Función de pérdidas del modelo 7

Modelo 8 LSTM de clasificación

Los parámetros del modelo 8 de la LSTM de clasificación son los siguientes:
 Número de neuronas = 50, learning rate = 0.0001.

Real temblor	1273	47
Real no temblor	16	1364
	Predicción temblor	Predicción no temblor

Tabla D.7: Matriz de confusión Modelo 8

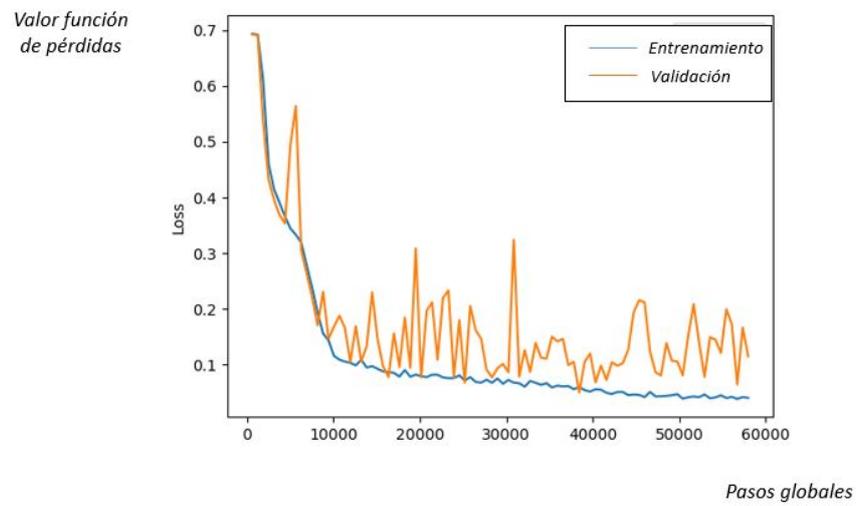


Figura D.7: Función de pérdidas del modelo 8

Modelo 9 LSTM de clasificación

Los parámetros del modelo 9 de la LSTM de clasificación son los siguientes:
 Número de neuronas = 50, learning rate = 0.001.

Real temblor	1193	127
Real no temblor	2	1378
	Predicción temblor	Predicción no temblor

Tabla D.8: Matriz de confusión Modelo 9

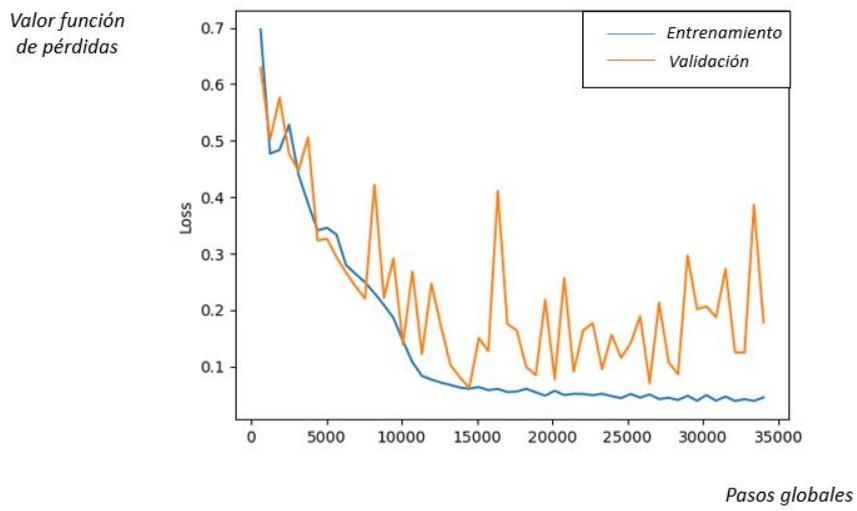


Figura D.8: Función de pérdidas del modelo 9

Apéndice E

Problema del desvanecimiento de gradientes en las RNNs

En este apéndice se explica y demuestra el problema del desvanecimiento de gradientes de las RNNs que les impide tener dependencias temporales a largo plazo.

Se muestra como ejemplo del problema a resolver la situación con respecto a W , siendo similar a los casos de U y b . La derivada parcial de la función de pérdidas con respecto a W en el paso de tiempo t puede ser definida a través de la Ecuación E.1:

$$\frac{\partial \mathcal{L}_t}{\partial W_t} = \frac{\partial \mathcal{L}_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \dots \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W} = \frac{\partial \mathcal{L}_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \left(\prod_{t=2}^T \frac{\partial h_t}{\partial h_{t-1}} \right) \frac{\partial h_1}{\partial W} \quad (\text{E.1})$$

Si se observa la Ecuación 2.10 y se evalúa la derivada parcial dentro del productorio $\left(\frac{\partial h_t}{\partial h_{t-1}} \right)$, al insertarla dentro de la Ecuación E.1 se llega a la Ecuación E.2 donde f es la función de activación:

$$\frac{\partial \mathcal{L}_t}{\partial W_t} = \frac{\partial \mathcal{L}_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \dots \frac{\partial h_2}{\partial h_1} \frac{\partial h_1}{\partial W} = \frac{\partial \mathcal{L}_t}{\partial y_t} \frac{\partial y_t}{\partial h_t} \underbrace{\left(\prod_{t=2}^T f'(U h_{t-1} + W x_t + b) U \right)}_{\vartheta} \frac{\partial h_1}{\partial W} \quad (\text{E.2})$$

Se ha sustituido en la Ecuación E.1 la derivada parcial del estado oculto por ϑ . Dentro de ϑ se encuentra la derivada de la función de activación. De forma general se usan las funciones *tanh* y *sigmoide*. Una propiedad importante de estas funciones es que mapean sus entradas en torno a -1 y 1 para la *tanh* y entre 0 y 1 para la *sigmoide*. ϑ tiende a desaparecer cuando el paso de tiempo (t) es grande y esto se debe a que la derivada de la función de activación (*tanh* y *sigmoide*) es menor que 1. Esto provoca que para algún paso de tiempo t , la derivada parcial de \mathcal{L} con respecto al parámetro W o U tienda a 0 (ver Ecuación E.3) :

$$\frac{\partial \mathcal{L}_t}{\partial W} \rightarrow 0 \quad (\text{E.3})$$

Se observa en la Ecuación 2.11 que la actualización del peso en un determinado paso de tiempo es nulo (ver Ecuación E.4).

$$W_{t+1} = W_t - \alpha \frac{\partial \mathcal{L}}{\partial W_t} \approx W_t \quad (\text{E.4})$$

De esta forma se demuestra el problema de las RNNs con el desvanecimiento de gradientes. Cuando el paso de tiempo es suficientemente grande, la derivada parcial de \mathcal{L} se vuelve 0 y por tanto no se actualizan los pesos.

Apéndice F

IMUs y su posicionamiento en la adquisición de datos

En este Anexo se muestran imágenes de las IMUs empleadas en el proceso de adquisición de los datos, la posición postural de los pacientes y sujetos sanos y el posicionamiento de las IMUs en los brazos.

La Figura F.1 muestra las Inertial Measurement Unit (IMU) del Sistema de Captura de Movimiento (Technaid S.L, España).



Figura F.1: Inertial Measurement Unit (IMU)

Los pacientes con TE y sujetos sanos fueron colocados en posición de temblor postural (ver Figura F.2(a)). El temblor postural es característico de la patología de TE. Es máximo cuando se mantiene una extremidad en una posición fija contra la gravedad y presenta una frecuencia de 5 a 8 Hz. Se colocaron 2 IMUs en cada brazo, una situada sobre la mano y otra situada sobre el antebrazo para adquirir la variación del ángulo de flexo-extensión de la muñeca a lo largo del tiempo (ver Figura F.2(b)).



(a) Posición postural en la adquisición de datos



(b) Posicionamiento de las IMUs

Figura F.2: IMUs y posición postural

Apéndice G

Ejemplos de predicción de señales tremorogénicas

En este Capítulo se muestran imágenes de las predicciones realizadas por la red neuronal LSTM variando el número de muestras de entrenamiento y el horizonte de predicción. Las figuras con 50 muestras de entrenamiento han sido extraídas con el modelo formado por 2 capas LSTM, 35 neuronas y 0.0005 de *learning rate*. Por otro lado, las figuras con 30 muestras de entrenamiento han sido extraídas con el modelo formado por 2 capas LSTM, 35 neuronas y 0.001 de *learning rate*.

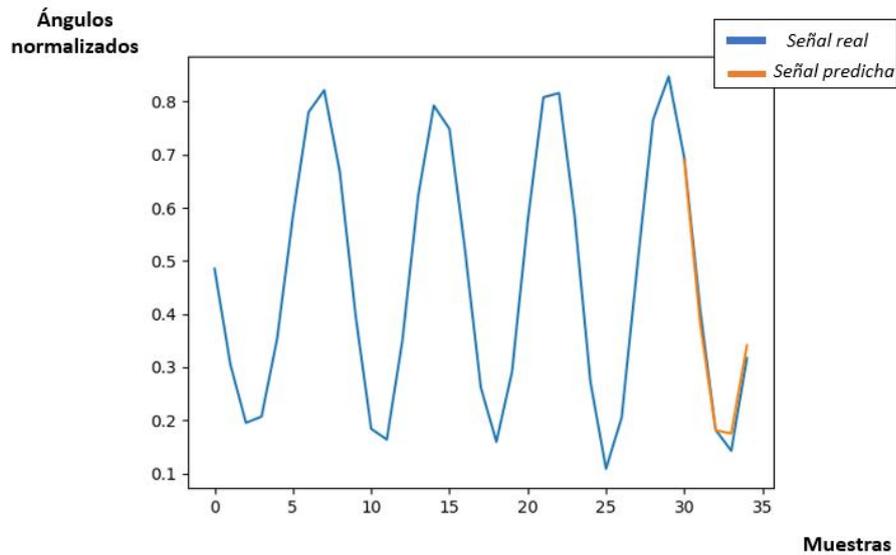


Figura G.1: Predicción con 30 muestras de entrenamiento / 5 de predicción

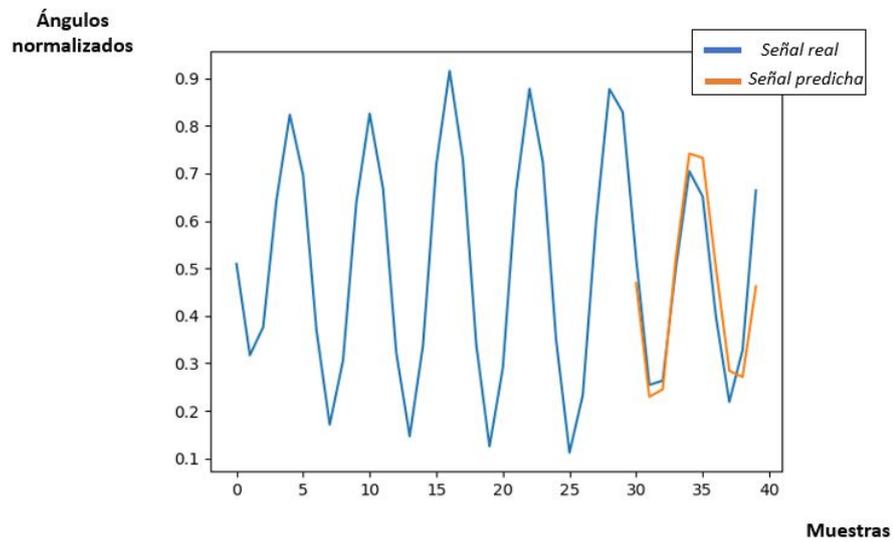


Figura G.2: Predicción con 30 muestras de entrenamiento / 10 de predicción

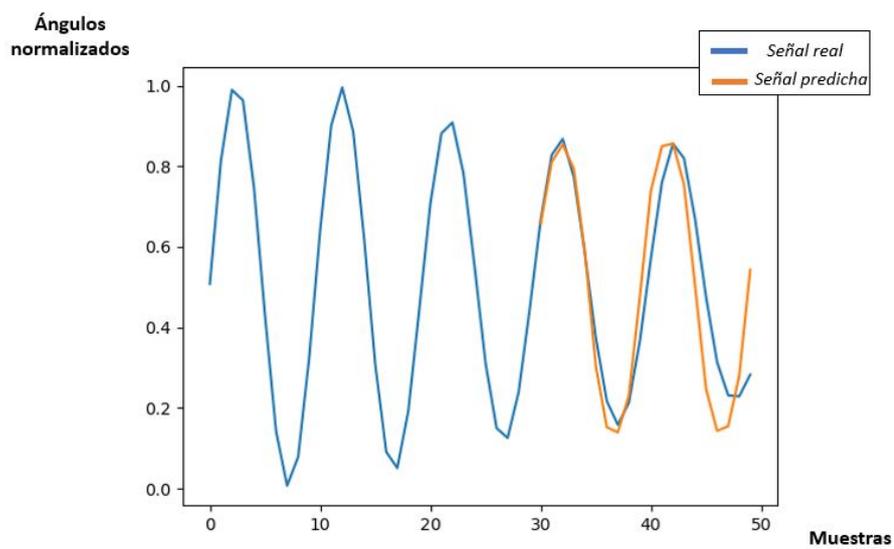


Figura G.3: Predicción con 30 muestras de entrenamiento / 20 de predicción

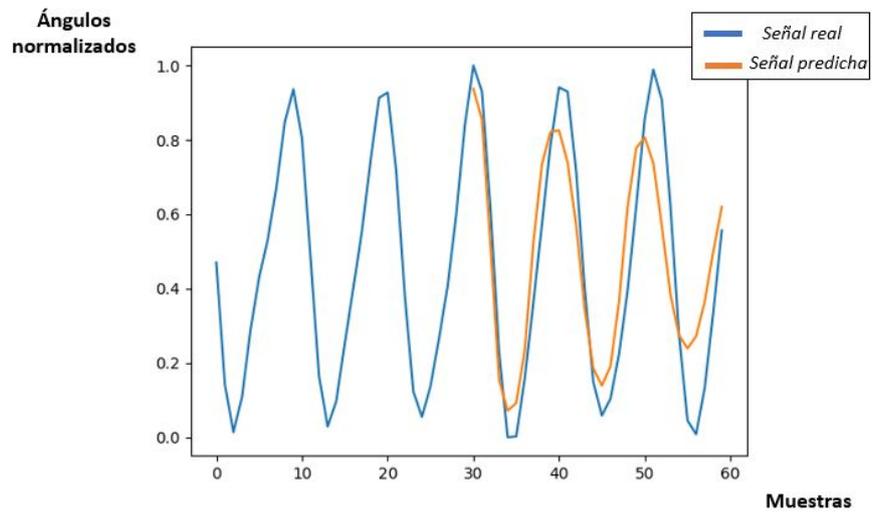


Figura G.4: Predicción con 30 muestras de entrenamiento / 30 de predicción

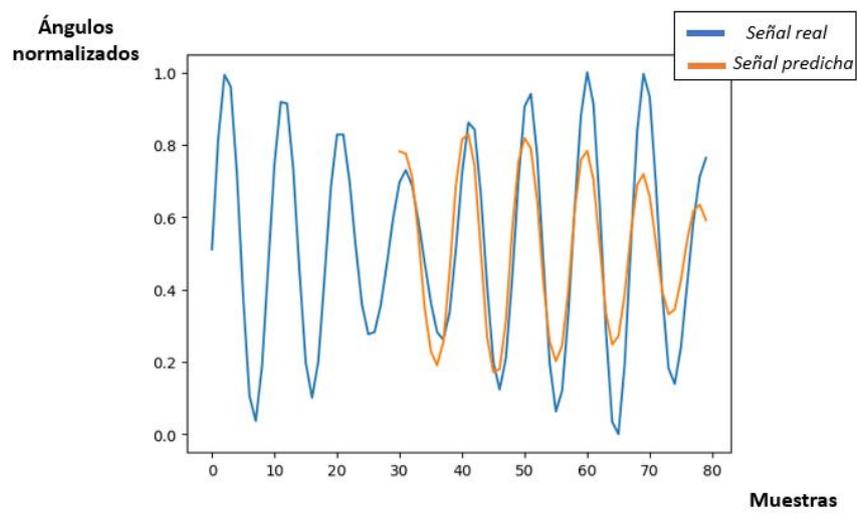


Figura G.5: Predicción con 30 muestras de entrenamiento / 50 de predicción

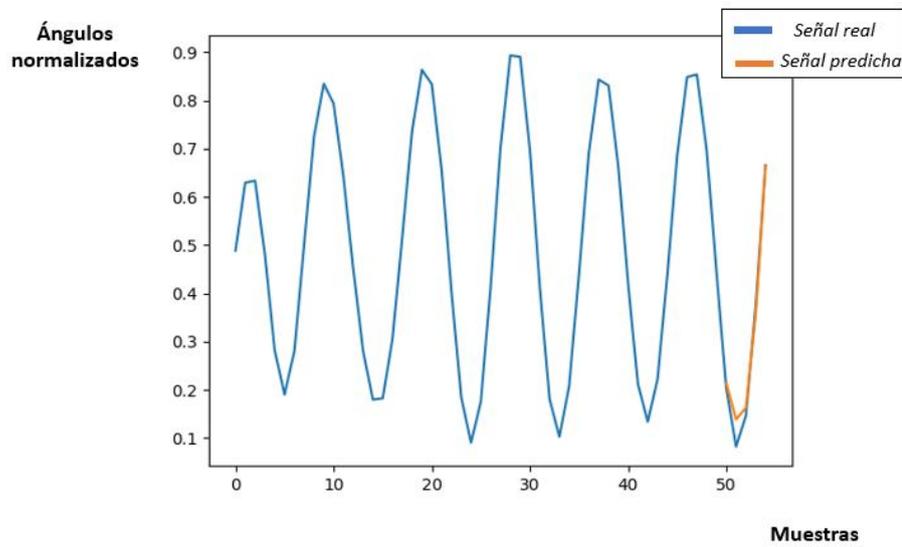


Figura G.6: Predicción con 50 muestras de entrenamiento / 5 de predicción

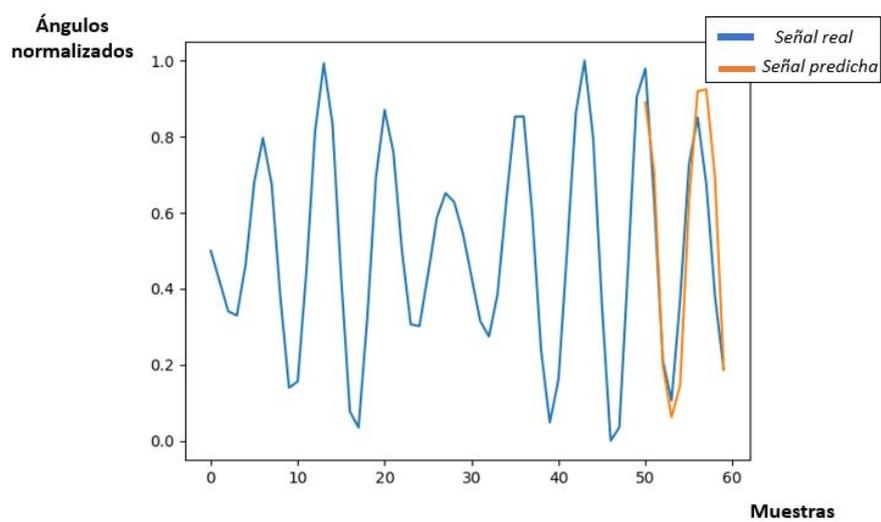


Figura G.7: Predicción con 50 muestras de entrenamiento / 10 de predicción

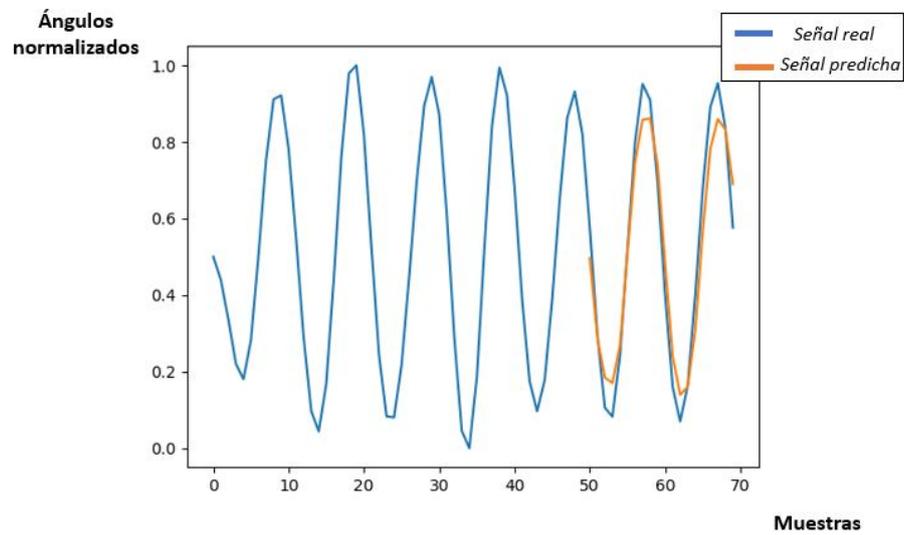


Figura G.8: Predicción con 50 muestras de entrenamiento / 20 de predicción

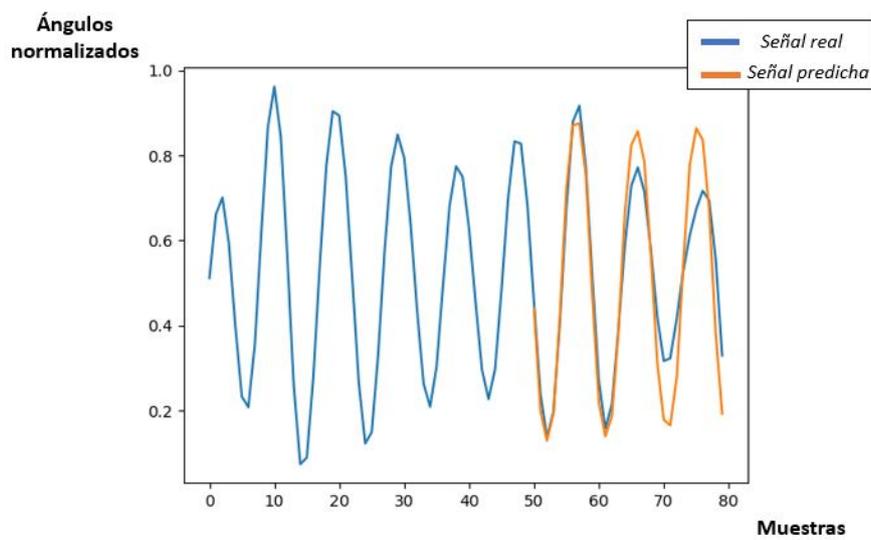


Figura G.9: Predicción con 50 muestras de entrenamiento / 30 de predicción

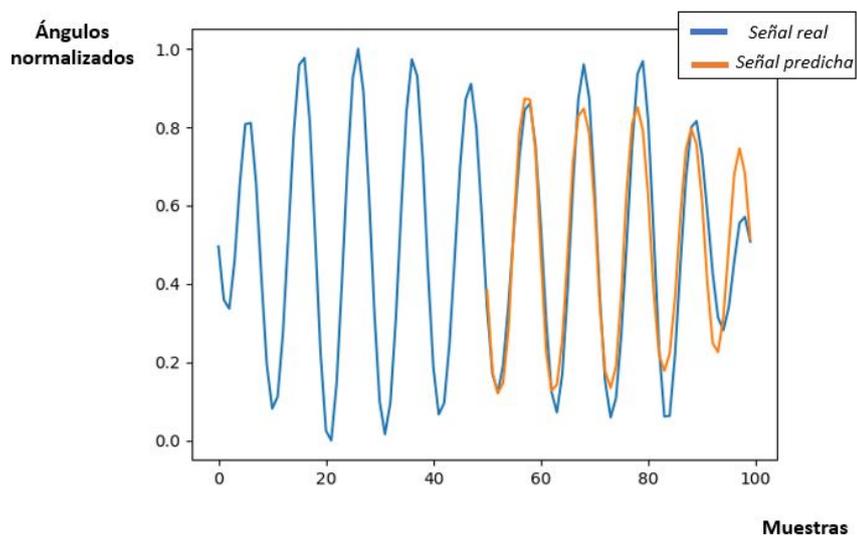


Figura G.10: Predicción con 50 muestras de entrenamiento / 50 de predicción

Apéndice H

Boxplots de los 180 modelos de predicción

En este Capítulo se muestran imágenes de los boxplots de los 180 modelos variando el número de muestras de entrenamiento (30 y 50) y el horizonte de predicción (5, 10, 20, 30 y 50). En cada modelo se varían el número de capas LSTM (1 y 2), el número de neuronas de cada capa (20, 35 y 50) y el valor del *learning rate* (0.0001, 0.0005 y 0.001), dando lugar a 18 modelos por boxplot.

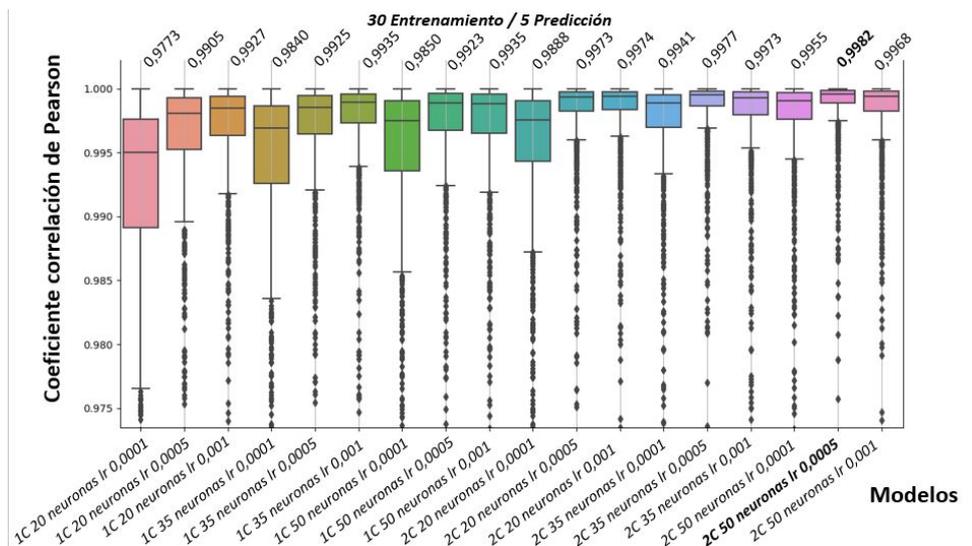


Figura H.1: Boxplot con 30 muestras de entrenamiento / 5 de predicción

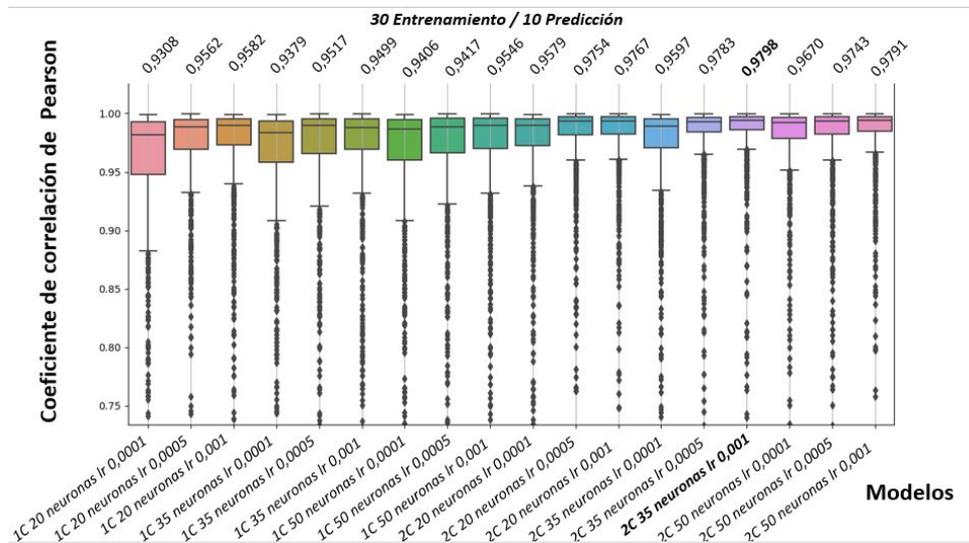


Figura H.2: Boxplot con 30 muestras de entrenamiento / 10 de predicción

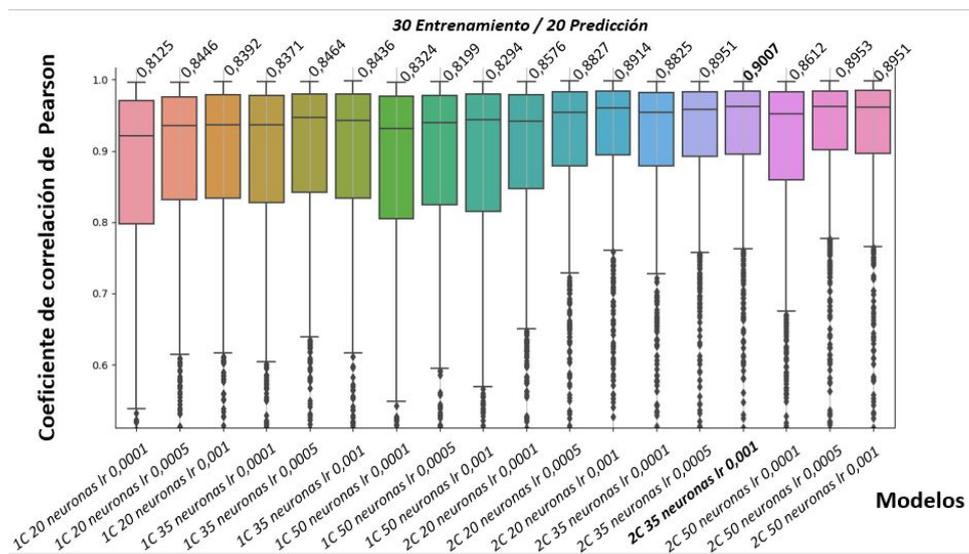


Figura H.3: Boxplot con 30 muestras de entrenamiento / 20 de predicción

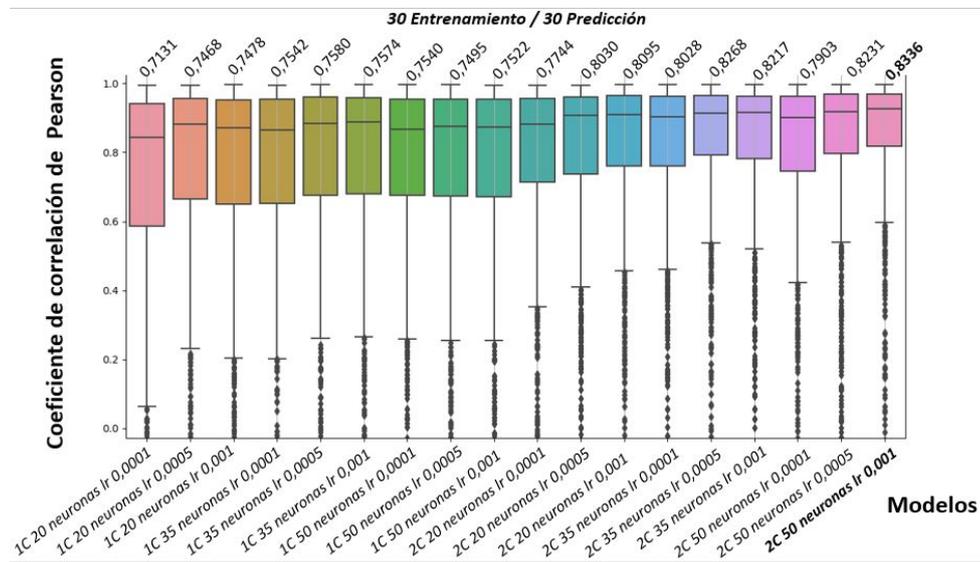


Figura H.4: Boxplot con 30 muestras de entrenamiento / 30 de predicción

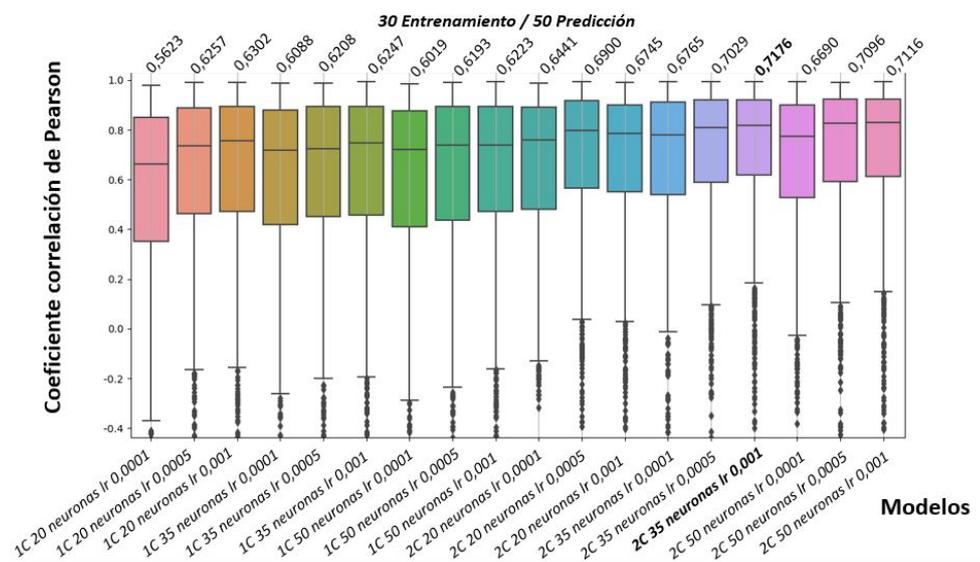


Figura H.5: Boxplot con 30 muestras de entrenamiento / 50 de predicción

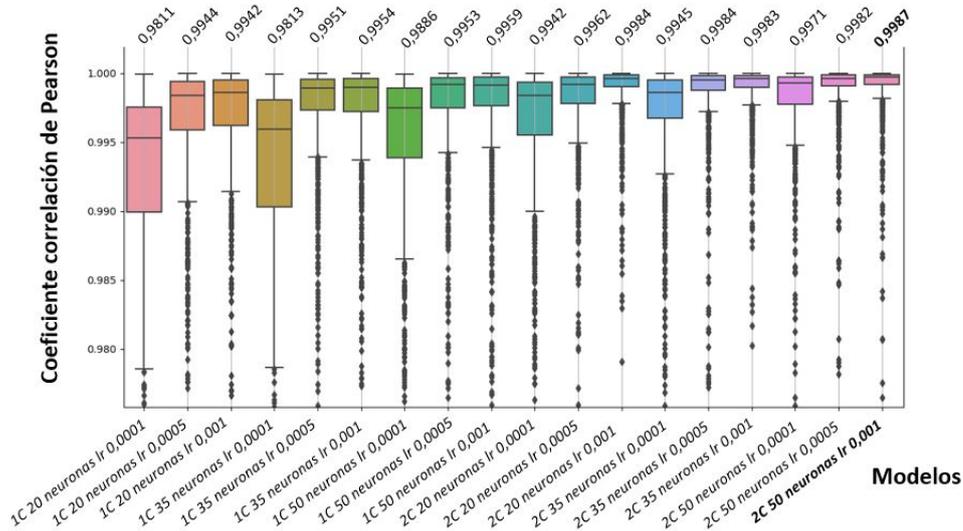


Figura H.6: Boxplot con 50 muestras de entrenamiento / 5 de predicción

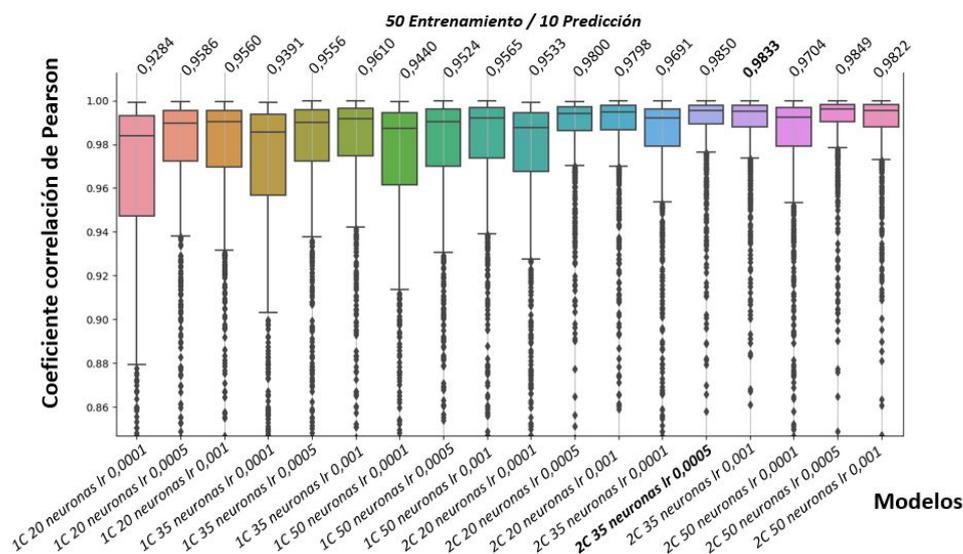


Figura H.7: Boxplot con 50 muestras de entrenamiento / 10 de predicción

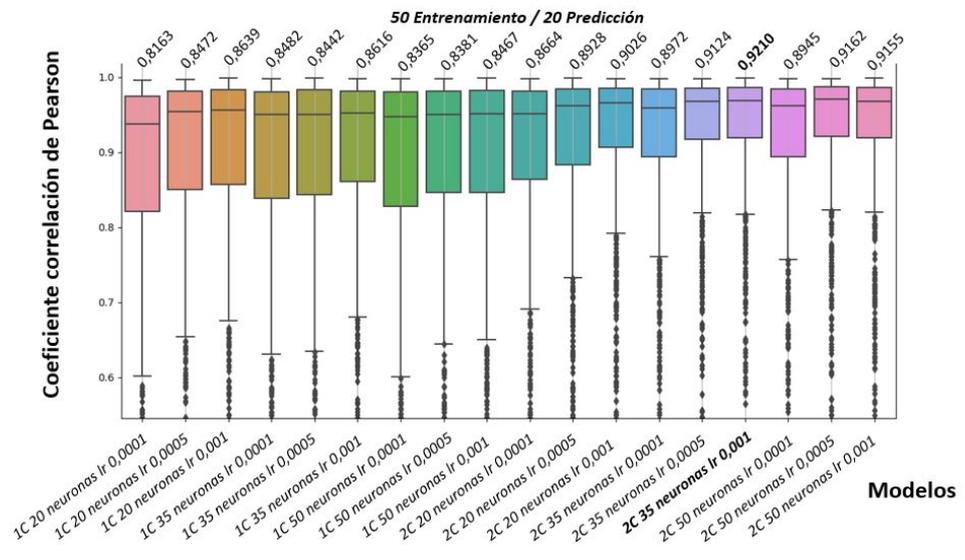


Figura H.8: Boxplot con 50 muestras de entrenamiento / 20 de predicción

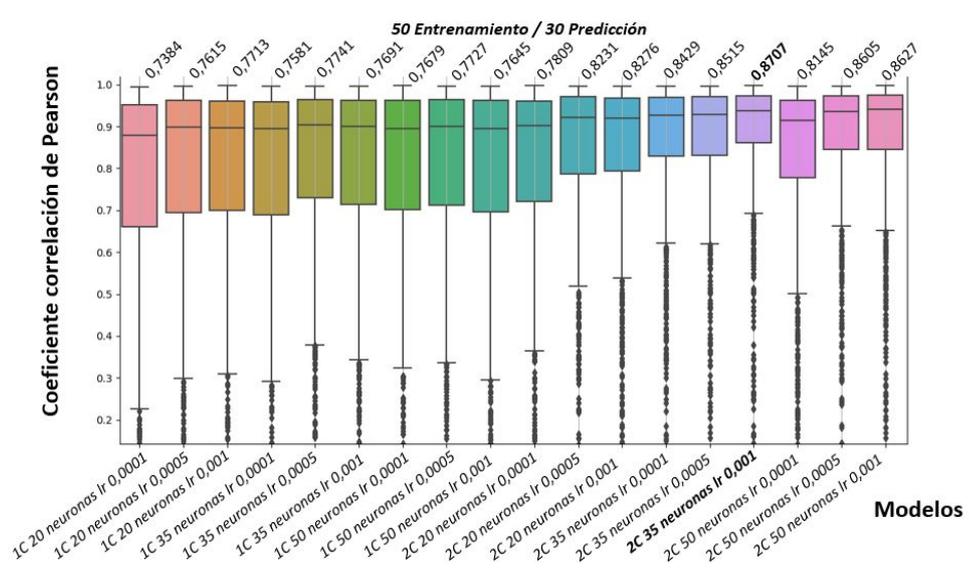


Figura H.9: Boxplot con 50 muestras de entrenamiento / 30 de predicción

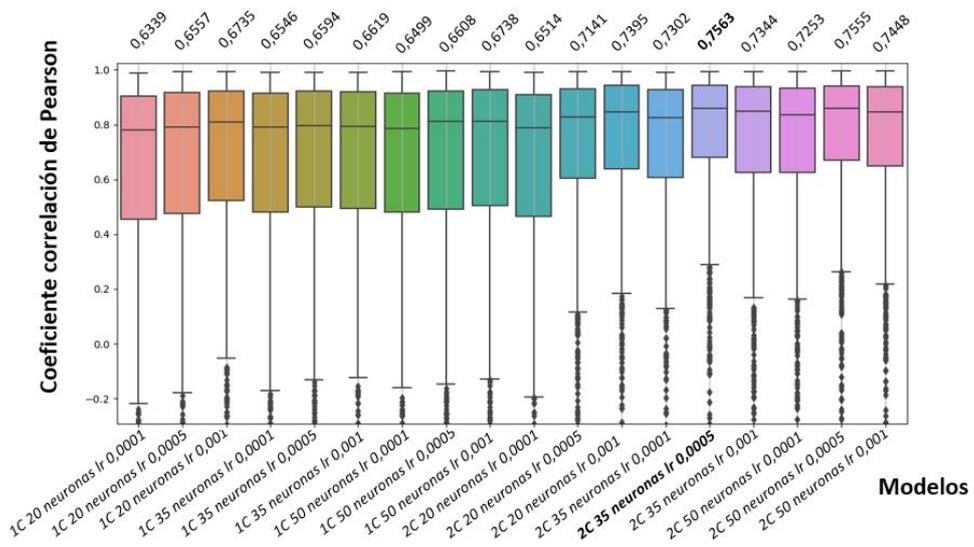


Figura H.10: Boxplot con 50 muestras de entrenamiento / 50 de predicción