Aprendizaje automático para la predicción de las etapas del sueño con dispositivos wearables

1st Marina Castro Olea
Facultad de Informática
Universidade da Coruña
A Coruña, España
m.castro6@udc.es

2nd Verónica Bolón Canedo
Facultad de Informática
Universidade da Coruña
A Coruña, España
veronica.bolon@udc.es

3rd Eva Pardo Otero
Facultad de Biología
Universidade da Coruña
A Coruña, España
eva.pardo@udc.es

4th Patricia Concheiro Moscoso
Facultad de Ciencias de la Salud
Universidade da Coruña
A Coruña, España
patricia.concheiro@udc.es

***Resumen*—Los dispositivos wearables están transformando la monitorización del sueño al proporcionar datos en tiempo real, accesibles y recopilados en entornos naturales. Sin embargo, la precisión de estos datos es variable, lo que limita su fiabilidad comparada con métodos tradicionales como la polisomnografía (PSG). Este trabajo analiza la efectividad de los modelos de aprendizaje profundo, específicamente de las redes neuronales LSTM y RNN, en la predicción de las diferentes etapas del sueño utilizando datos de dispositivos wearables. Los hallazgos indican que los modelos RNN tienen un rendimiento levemente superior a los LSTM en lo que respecta a la precisión.**

***Palabras clave*—Monitorización del sueño, dispositivos wearables, aprendizaje automático, red neuronal, modelos LSTM, modelos RNN, clasificación de etapas del sueño.**

I. Introducción

I-A. Introducción al aprendizaje profundo en el contexto de la salud

El aprendizaje profundo es una rama del aprendizaje automático que ha surgido como una herramienta transformadora en el ámbito de la salud. Este enfoque, que emula el funcionamiento de las redes neuronales del cerebro humano, permite analizar grandes volúmenes de datos complejos, impulsando aplicaciones innovadoras que están cambiando tanto la investigación como la práctica clínica [1].

Una de las aplicaciones más importantes del aprendizaje profundo en el ámbito de la salud es su capacidad para procesar imágenes médicas. Las herramientas basadas en esta tecnología han demostrado ser extremadamente eficaces en la detección temprana de enfermedades como el cáncer mediante el análisis de radiografías, resonancias magnéticas y tomografías computarizadas. La alta precisión en la interpretación de estas imágenes no solo acelera el diagnóstico, sino que también facilita tratamientos más adecuados y personalizados [2].

Además de la imagen médica, el aprendizaje profundo potencia el análisis de datos clínicos y genéticos. Al integrar información de diversas fuentes, como registros médicos electrónicos y datos genómicos derivados de la secuenciación del ADN, se pueden descubrir patrones que no son detectables mediante análisis tradicionales [3]. Esto allana el camino para una medicina más predictiva y personalizada, y permite a los profesionales de la salud desarrollar tratamientos adaptados a las características individuales de cada paciente [4].

No obstante, la adopción del aprendizaje profundo en el ámbito de la salud también plantea desafíos importantes. La calidad de los datos, la privacidad de la información y la interpretabilidad de los modelos son aspectos cruciales que deben abordarse para asegurar la eficacia y la seguridad de estas tecnologías. Es esencial desarrollar marcos éticos y regulaciones sólidas para fomentar un uso responsable y fiable del aprendizaje profundo en la atención sanitaria [5].

I-B. Dispositivos wearables y monitorización de sueño

Los dispositivos wearables se han convertido en un elemento muy presente en la vida cotidiana, ya que ofrecen una gran cantidad de datos relacionados con la actividad física, la frecuencia cardíaca y los patrones de sueño. Estos dispositivos, equipados con diversos sensores como acelerómetros y giroscopios [6], pueden recopilar datos continuamente y ofrecer una visión general de los patrones de sueño de un individuo. Investigaciones anteriores han demostrado el potencial de los dispositivos portátiles en la monitorización del sueño, destacando su capacidad para detectar con precisión las fases del sueño, como el sueño ligero, el sueño profundo y el sueño REM [7].

Por ejemplo, un estudio de Zhang et al. (2018) sobre el análisis y la visualización de las etapas del sueño basado en redes neuronales profundas, mostró la eficacia de los modelos de aprendizaje profundo para clasificarlas con precisión moderada (entre el 60 % y el 64 %), lo que demuestra la viabilidad del enfoque, aunque con margen de mejora [8]. La creciente disponibilidad de datos procedentes de estos dispositivos y los avances en la tecnología de aprendizaje profundo han abierto

nuevas vías para la exploración de la predicción de las fases del sueño [9].

I-C. Fases y Etapas del sueño

El sueño es un proceso biológico esencial que desempeña un papel crucial en el bienestar físico y mental de las personas [10]. Este complejo fenómeno se organiza en ciclos, divididos en varias fases y etapas, cada una con funciones específicas que son vitales para la recuperación del organismo, el procesamiento emocional y la consolidación de la memoria [11]. Comprender las distintas fases del sueño es fundamental para valorar su importancia en la salud general.

El ciclo del sueño se compone de dos grandes fases: el sueño NREM (movimientos oculares no rápidos) y el sueño REM (movimientos oculares rápidos). Ambos tipos están subdivididos en etapas que cumplen diferentes funciones fisiológicas. El sueño NREM se divide en tres etapas: N1, N2 y N3 [12].

La etapa N1 es la etapa inicial del sueño, en la que se produce la transición de la vigilia al sueño. Durante esta etapa, la persona experimenta una somnolencia ligera y los movimientos oculares se vuelven lentos. Esta etapa es breve y dura solo unos minutos, pero es crucial para preparar el cuerpo para las etapas más profundas del sueño [13].

La etapa N2, que representa aproximadamente el 50 % del tiempo total de sueño, se caracteriza por una disminución significativa de la actividad muscular y un descenso en la temperatura corporal, lo que favorece un estado de relajación y descanso. En esta etapa, las ondas cerebrales comienzan a mostrar patrones específicos conocidos como husos del sueño y complejos K, que son esenciales para proteger el cerebro de estímulos externos y para la consolidación de la memoria [13].

La etapa N3, también conocida como sueño profundo o de ondas lentas, es la más reparadora. Durante esta etapa, las ondas cerebrales son más lentas y grandes, conocidas como ondas delta, y se producen procesos de recuperación y regeneración celular. Esta etapa es crucial para la restauración física, el crecimiento y la reparación de tejidos, y el fortalecimiento del sistema inmunológico. Además, el sueño profundo es fundamental para la liberación de hormonas de crecimiento y la eliminación de toxinas acumuladas en el cerebro durante el día [14].

Tras completar las etapas del sueño NREM, el ciclo del sueño progresa hacia el sueño REM. Esta fase es particularmente importante para el procesamiento emocional y la consolidación de la memoria. Durante el sueño REM, la actividad cerebral se asemeja a la de la vigilia, con ondas cerebrales rápidas y desincronizadas. Es en esta fase donde los sueños son más vívidos y se experimentan movimientos oculares rápidos. El sueño REM juega un papel clave en la integración de las experiencias y los conocimientos adquiridos durante el día, y facilita el aprendizaje y la adaptación emocional [15].

El ciclo del sueño se repite varias veces a lo largo de la noche. A medida que avanza la noche, los períodos de sueño REM se vuelven más prolongados, mientras que las etapas de sueño profundo disminuyen, como se puede observar en la Figura 1. Este equilibrio entre las diferentes fases y etapas del sueño es esencial para mantener una salud física y mental óptima. Las alteraciones en este ciclo pueden tener efectos perjudiciales en la memoria, el estado de ánimo y el bienestar general, entre otros [16].

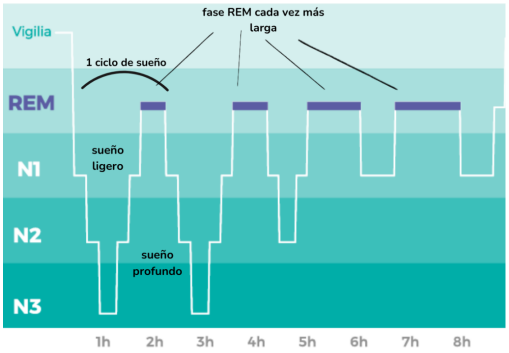

Figura 1. Esquema representativo de los ciclos de sueño a lo largo de una noche. Se distinguen las fases de vigilia, sueño ligero (N1), sueño intermedio (N2), sueño profundo (N3) y sueño REM. Cada ciclo completo de sueño tiene una duración aproximada de 90 minutos y se repite entre cuatro y seis veces por noche. Se aprecia cómo la fase REM se prolonga progresivamente en los ciclos posteriores, siendo más extensa hacia el final del periodo de sueño [17].

I-D. El sueño y su importancia para la salud

Las alteraciones del sueño afectan a una parte considerable de la población mundial, con repercusiones significativas en la calidad de vida y en la salud física y mental [18]. Cuando se ve alterado, puede dar lugar a una variedad de enfermedades y trastornos que van desde problemas leves hasta condiciones graves que requieren intervención médica [19].

Uno de los trastornos del sueño más comunes es el insomnio, que se caracteriza por la dificultad para iniciar o mantener el sueño. Las causas del insomnio son diversas, incluyendo factores como el estrés, la ansiedad y la depresión, que a menudo se retroalimentan mutuamente, exacerbando estos problemas emocionales. El insomnio no solo reduce la cantidad de sueño, sino que también afecta a su calidad, provocando síntomas como fatiga, irritabilidad y dificultades de concentración en las actividades diarias [20].

Otro trastorno relevante son las apneas del sueño, que se caracterizan por episodios repetidos de obstrucción de las vías respiratorias durante el sueño. Esta condición provoca un sueño fragmentado y no reparador, que da como resultado somnolencia diurna excesiva y una disminución del rendimiento cognitivo [21]. Las apneas del sueño influyen en el desarrollo de problemas como la hipertensión y las enfermedades cardiovasculares, además de que también están asociadas con un mayor riesgo de accidentes laborales y de tráfico debido a la somnolencia diurna [22].

Además, los trastornos del ritmo circadiano, como el síndrome de fase retrasada del sueño y el síndrome de fase adelantada del sueño, son ejemplos comunes de alteraciones del sueño que pueden surgir debido a cambios en los hábitos de vida, como el trabajo nocturno. Estos trastornos desajustan el ciclo natural del sueño [23].

La relación entre el sueño y la salud es bidireccional; no solo las alteraciones del sueño pueden contribuir al desarrollo de enfermedades, sino que la presencia de ciertas patologías también puede afectar a la calidad del sueño [24]. Por ejemplo, enfermedades crónicas como la diabetes y la obesidad pueden interrumpir el sueño y crear un ciclo vicioso que agrava ambos problemas [25].

I-E. Visión general sobre la clasificación de las etapas del sueño

La clasificación de las fases del sueño es un aspecto crucial de la monitorización y el análisis del sueño. El método estándar para clasificar las fases del sueño es la polisomnografía (PSG), que implica el registro simultáneo de varias señales fisiológicas, como el electroencefalograma (EEG) y el electromiograma (EMG).

Aunque la polisomnografía es el método de referencia para clasificar los estadios del sueño, su naturaleza laboriosa, costosa y la incomodidad que causa a los pacientes [26] puede interferir con el sueño y conducir a conclusiones erróneas.

En diferentes estudios se hace mención de métodos de aprendizaje profundo para la clasificación de las etapas del sueño; las arquitecturas de aprendizaje profundo, como las redes neuronales convolucionales, han mostrado resultados prometedores en la predicción precisa de las etapas del sueño utilizando datos de diversas fuentes, incluidos los dispositivos portátiles [27].

II. Objetivos

El objetivo principal de este proyecto es poder clasificar las etapas de sueño a partir de datos de wearables aplicando diferentes modelos de machine learning. Para poder conseguirlo, se han concretado los siguientes objetivos más específicos.

- Recopilación de datos simultánea. Recopilar datos de diferentes sujetos gracias al uso del dispositivo Xiaomi Mi Band 7 a través de las aplicaciones móviles *Gadgetbridge* de código abierto y *ZeppLife*, que es la oficial de los dispostivos Xiaomi, de manera simultánea para garantizar la coherencia en los datos.
- Etiquetado con *ZeppLife*. Las etiquetas de las diferentes clases de sueño se obtienen a través de la aplicación *ZeppLife*, una aplicación de seguimiento del sueño ampliamente utilizada, para evaluar su precisión y eficacia en la predicción de las etapas del sueño.
- Procesamiento y limpieza de datos. Realizar un procesamiento adecuado de los datos para eliminar valores atípicos, corregir errores y garantizar la calidad de los datos para su posterior análisis.
- Selección de características. Identificar las características relevantes de los datos que puedan influir en las diferentes etapas del sueño, como la frecuencia cardíaca, la intensidad de actividad, el tipo de actividad o los pasos diarios, entre otros.
- Desarrollo de modelos de machine learning. Implementar modelos de machine learning, como redes neuronales recurrentes *(RNNs)* y/o long short – term memory networks *(LSTMs)*, para predecir las etapas del sueño basadas en las características seleccionadas.
- Evaluación del modelo. Evaluar la precisión y el rendimiento del modelo utilizando métricas adecuadas para este tipo de modelo.

III. Metodología

III-A. Obtención de los datos

Los datos que se han utilizado proceden de tres sujetos anónimos que han llevado la pulsera Xiaomi Mi Band 7 durante un periodo de tiempo aproximado de un mes. Durante este periodo, la pulsera ha ido midiendo cada minuto y se han ido recopilando las mediciones gracias a las aplicaciones de *Gadgetbridge* y *ZeppLife*. *Gadgetbridge* es una aplicación de código abierto que nos permite conectar diversos dispositivos wearables, sin importar la marca, a un smartphone. Es una alternativa a *ZeppLife* y nos permite acceder a los datos crudos recolectados y así poder analizarlos [28]. Los datos recogidos por *ZeppLife* se almacenan en la nube, en los servidores del proveedor Huami, cada vez que se realizan sincronizaciones con *ZeppLife*. Mientras que en *Gadgetbridge* se almacenan localmente en el dispositivo móvil en una carpeta o en la nube personal del usuario. Para poder conectar el dispositivo wearable a *Gadgetbridge* se necesita una clave de acceso restringido generada por *ZeppLife* a la que solo se puede acceder con un teléfono rooteado [29].

Las variables que obtenemos de *Gadgetbridge* son las siguientes:

- 'TIMESTAMP': hora en formato Unix (número de segundos transcurridos desde las 00:00:00 UTC del 1 de enero de 1970)
- 'DEVICE_ID' : identificador del dispositivo conectado.
- 'USER_ID': identificador del usuario.
- 'HEART_RATE': pulsaciones por minuto. Cuando no se mide debido a cualquier problema, registra el valor 255.
- 'RAW_INTENSITY': relacionada con la cantidad de movimiento, probablemente datos del acelerómetro de la pulsera. Son códigos numéricos donde valores más altos indican mayor movimiento.
- 'UNKNOWN1': variable desconocida.
- 'STEPS': número de pasos.
- 'RAW_KIND': valores discretos, específicos del dispositivo, que se envían procesados desde la pulsera y se recogen por *Gadgetbridge*. Indican diferentes situaciones; por ejemplo, para la Mi Band 7, el valor 115 aparece cuando la pulsera no está puesta.
- 'SLEEP', 'DEEP_SLEEP', 'REM_SLEEP': son datos codificados del sueño, están procesados por *ZeppLife*. Los desarrolladores de *Gadgetbridge* establecen umbrales de forma aproximada para que las fases y etapas del sueño que muestra su aplicación se correspondan lo más posible con las que se muestran en *ZeppLife*.

Por parte de la aplicación *ZeppLife*, añadimos las siguientes variables:

- 'ZpTimeStamp': hora en formato Unix.

- ‘ZpModes’: etiquetas para las diferentes etapas del sueño codificadas por un valor numérico que se muestran en la Tabla I. Vamos a usar esta variable como *ground truth.*
- ‘ZpStages’: etiquetas con los nombres de las diferentes etapas del sueño.

Tabla I
CODIFICACIÓN NUMÉRICA DE LOS DISTINTOS ESTADOS DE SUEÑO EN LA VARIABLE ZPMODES

Mode	Estados de sueño [30]
4	Sueño ligero
5	Sueño profundo
7	Despierto
8	Rapid Eye Movement (REM)

Los datos fueron recogidos finalmente en un archivo ‘csv’ conjunto, dividido en varias hojas donde se encuentran los datos que corresponden a ambas aplicaciones (*Gadgetbridge* y *ZeppLife*) de cada usuario.

Para poder utilizar estos datos, se tuvo que separar en archivos ‘csv’ independientes. Este procedimiento se realizó en Visual Studio Code (1.92.2) en lenguaje Python (3.10.12) utilizando un entorno de Anaconda (2.4.0). Como se ha comentado, al estar los datos de las diferentes aplicaciones por separado, se tuvo que hacer un ‘left join’ de cada usuario tomando la variable ‘TIMESTAMP’ como clave primaria. Al hacer este procedimiento, como los datos que corresponden al tiempo despierto no tenían valores en las variables ‘ZpStages’ y ‘ZpModes’, se tuvo que añadir los valores ‘awake’ y ‘7’, y por último se guardaron. De esta manera, ya se tienen los tres conjuntos de datos de los tres usuarios preparados para empezar el proyecto.

III-B. Análisis exploratorio de los datos

En esta sección del proyecto, se ha optado por utilizar el entorno de Google Colab, dado que ofrece la capacidad de trabajar con GPUs, lo cual permite realizar las operaciones de procesamiento de datos de manera más eficiente y rápida. Para realizar el análisis exploratorio de los datos, se utilizaron las siguientes herramientas y librerías de software:

- **Python 3.10.12**: es la versión del lenguaje de programación Python utilizada.
- **NumPy 1.26.4**: una librería usada para el manejo y análisis de datos numéricos.
- **Pandas 2.1.4**: una librería utilizada para el análisis y manipulación de datos estructurados, así como para gestionar información en formato tabular.
- **Matplotlib 3.7.1**: una librería empleada para generar gráficos y visualizaciones de datos.
- **Seaborn 0.13.1**: una librería utilizada para la representación estadística y optimización de gráficos.

En primer lugar, se realizó un análisis preliminar de las variables con el objetivo de asegurar que el entorno y los datos estuvieran en condiciones adecuadas para el análisis posterior.

Se realizó un análisis estadístico de las variables involucradas, examinando métricas clave como la media, la desviación estándar, entre otras. Esta evaluación inicial es fundamental para determinar la relevancia de las variables en el análisis. Por ejemplo, una desviación estándar igual a cero indica que todos los valores de esa variable son idénticos, lo que sugiere una falta de variabilidad y, por ende, la ausencia de información nueva o útil para el análisis. Como ocurre con las variables ‘DEVICE_ID’ y ‘USER_ID’, que se procede a eliminarlas. El resto de las estadísticas de las demás variables se pueden ver en el Notebook del proyecto.

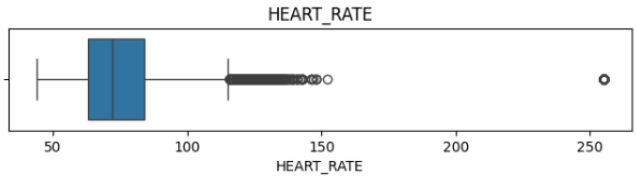

Figura 2. Valores atípicos de la variable ´HEART_RATE’ del conjunto de datos 1.

Posteriormente, se generaron gráficos individuales para cada variable con el fin de visualizar los datos y detectar posibles valores atípicos. Durante este proceso, se identificó una anomalía significativa en la variable ‘HEART_RATE’ que se puede ver en la Figura 2, donde se observaron valores inusualmente altos, aproximadamente de 255 latidos por minuto. Estos valores son atípicos, ya que, por lo que se explicó anteriormente, son problemas en la medición, y se procede a eliminarlos del conjunto de datos para evitar que distorsionen los resultados del análisis.

Aunque no se encontraron valores nulos, se identificó un patrón inusual en los datos correspondientes al conjunto de datos 2 durante el período del 12 al 19 de abril de 2024. En este intervalo, el dispositivo no registró mediciones de sueño, mostrando constantemente que el sujeto estaba despierto. Este comportamiento coincide con los valores anómalos observados en la variable de ‘HEART_RATE’, lo que sugiere un posible fallo en el dispositivo durante ese período.

Se llevó a cabo un análisis de la variable ‘Zp_Stages’ con el objetivo de evaluar la distribución de sus posibles valores. El análisis reveló una marcada predominancia del estado ‘awake’ (despierto), el cual constituye casi la totalidad de las observaciones, mientras que los otros estados representan aproximadamente un 25 % del total, como se puede ver en la Figura 3.

Esta asimetría en la distribución de los datos es un factor crítico a considerar al realizar la partición del conjunto de datos en subconjuntos de entrenamiento, validación y prueba (train, validation y test). La disparidad en la representación de los diferentes estados podría sesgar los resultados del modelo si no se maneja adecuadamente, lo que subraya la necesidad de aplicar técnicas específicas de muestreo [31] o ponderación que aseguren una representación equilibrada de todas las clases durante el proceso de modelado [32].

Con el objetivo de explorar las relaciones entre las variables recogidas por el dispositivo de monitorización, se calculó la matriz de correlación de Pearson. Este análisis permite identificar qué variables podrían estar más asociadas entre sí

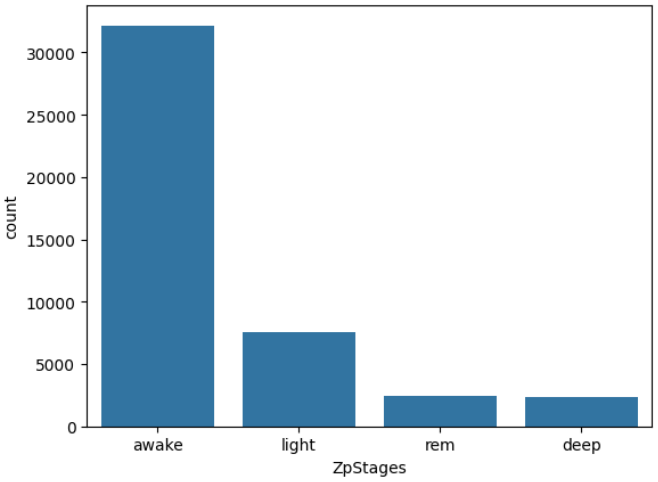

Figura 3. Distribución de las observaciones por etapa del sueño (ZpStages) en el conjunto de datos 1. Se observa un claro desbalanceo de clases, con una predominancia de registros en la etapa awake (despierto) frente a las etapas de sueño (light, rem, deep).

Figura 4. Matriz de correlación de Pearson entre las variables del conjunto de datos. Las variables asociadas al sueño (SLEEP, DEEP_SLEEP, REM_SLEEP) muestran una fuerte correlación positiva con RAW_KIND, mientras que las relacionadas con la actividad física (RAW_INTENSITY, STEPS) presentan correlaciones negativas, lo que refleja una oposición esperada entre sueño y movimiento.

y, por tanto, aportar información relevante para la clasificación de las etapas del sueño. En la Figura 4 se muestra dicha matriz, donde se destacan las correlaciones más significativas entre las variables fisiológicas, de actividad y de sueño.

Este procedimiento emplea el coeficiente de correlación de Pearson, que mide la fuerza y la dirección de la relación lineal entre pares de variables. El coeficiente de Pearson oscila entre -1 y 1, donde un valor absoluto cercano a 1 indica una relación lineal fuerte, ya sea positiva o negativa; un valor cercano a 0 sugiere una relación débil o inexistente [33].

Las variables con una correlación elevada (cercana a 1 o -1) pueden ser consideradas redundantes y, por lo tanto, podrían ser eliminadas del análisis para simplificar el modelo y reducir la multicolinealidad [34].

Además, la matriz de correlación facilita la identificación de variables que podrían ser particularmente relevantes como características para los modelos de redes neuronales. Las variables que muestran una correlación significativa con la variable objetivo o entre sí pueden ofrecer información valiosa para mejorar la capacidad predictiva del modelo.

Este análisis preliminar permite seleccionar características que potencian el desempeño del modelo y optimizan el proceso de entrenamiento, asegurando así que las redes neuronales se beneficien de las variables más informativas y menos redundantes.

III-C. Preprocesamiento de los datos

Tras el análisis exploratorio, se decide:

- Eliminar los valores atípicos de la variable 'HEART_RATE' (tanto el periodo etiquetado continuamente como 'despierto' como aquellos valores con valor igual a 255).
- Eliminar variables irrelevantes para el modelo: 'DEVICE_ID', 'USER_ID', 'UNKNOWN1', 'TIMESTAMP', 'Zp_Modes', 'Zp_Stages', 'Zp_Datetime'.

III-D. Selección de características

Se realizó un proceso sistemático para la selección de características, con el fin de identificar y escoger las variables más significativas para el análisis. Para ello, se utilizó la función *SelectKBest()* de la librería Sklearn [35]. A esta función se la ha configurado para usar la función *f_classif()* que realiza el cómputo del análisis de varianza (ANOVA) F-value para los datos que se le introduzcan.

Este procedimiento y la matriz de correlación permitió determinar cuáles de las características presentes en el conjunto de datos eran más pertinentes para el modelo. En particular, las características identificadas como más relevantes fueron: 'HEART_RATE', 'RAW_INTENSITY', 'RAW_KIND', 'LIGHT_SLEEP', 'DEEP_SLEEP' y 'REM_SLEEP'. Estas variables se seleccionaron en función de su capacidad para proporcionar información significativa y de su potencial para mejorar el rendimiento del modelo predictivo. Posteriormente, se procedió a la eliminación de las columnas no seleccionadas, con el objetivo de simplificar el conjunto de datos y concentrar el análisis en las características más informativas. Se eliminaron las columnas 'LIGHT_SLEEP', 'DEEP_SLEEP' y 'REM_SLEEP' a pesar de ser significativas, ya que son números codificados con significado desconocido y no son comunes en todos los dispositivos wearables. Se añadió la variable 'STEPS' ya que es la siguiente en tener relevancia.

Este paso es crucial para reducir la complejidad del modelo y evitar la inclusión de variables redundantes o irrelevantes que podrían afectar negativamente a su rendimiento. Además, para completar el proceso de preparación de datos, se asignaron las etiquetas correspondientes a las variables objetivo. Este paso asegura que el modelo utilice las etiquetas adecuadas para la variable objetivo, lo que permite una alineación precisa entre las características seleccionadas y las etiquetas correspondientes durante la formación del modelo.

III-E. Métodos y algoritmos de aprendizaje profundo utilizados

En este proyecto se han utilizado dos algoritmos de redes neuronales, que son las redes neuronales recurrentes (RNN, siglas en inglés) y las redes neuronales de memoria a corto - largo plazo (LSTM, siglas en inglés) Las RNN son un tipo de red neuronal artificial que utiliza datos secuenciales o datos de series temporales [36]. En principio, las RNN pueden asignar a cada salida todo el historial de entradas anteriores [37]. La clave está en que las conexiones recurrentes permiten que exista una memoria de entradas anteriores en el estado interno de la red y, por tanto, que influya en la salida de la red, siendo esta característica decisiva para el análisis de series temporales, como es este proyecto [38]. Para las arquitecturas RNN estándar, el conjunto de contextos a los que se puede acceder en la práctica es bastante limitado. El problema es que la influencia de una entrada determinada en la capa oculta y, por tanto, en la salida de la red, decae o se dispara exponencialmente a medida que recorre las conexiones recurrentes de la red. Este efecto se conoce en la literatura como el problema del gradiente evanescente [39].

Una red LSTM es equivalente a una RNN convencional, salvo que las unidades sumadoras de la capa oculta se sustituyen por bloques de memoria [40]. De esta manera, se intenta combatir el problema del gradiente evanescente y de mantener información relevante durante períodos más largos de tiempo [41].

III-F. Construcción y entrenamiento de modelos para la predicción del sueño

Antes de la implementación de las redes neuronales (RNN y LSTM), se realizó un preprocesamiento de los datos basado en la segmentación de la serie temporal en secuencias de longitud fija. Este procedimiento consiste en generar subconjuntos consecutivos de datos con un número determinado de pasos temporales (en este caso, secuencias de 5 y 10 minutos), asignando a cada uno de ellos la etiqueta correspondiente. De esta forma, se obtiene un conjunto estructurado de pares entrada-salida que preserva la dimensión temporal de la información y permite a los modelos aprender patrones secuenciales. Este preprocesamiento es fundamental para que las arquitecturas recurrentes puedan captar las dependencias a lo largo del tiempo. Las redes fueron desarrolladas utilizando las bibliotecas Scikit-learn (1.6.1), TensorFlow (2.18.0) y Keras (3.8.0).

- **Estructura del Modelo LSTM:**
  - Capa de entrada: La red comienza con una capa de entrada que toma una secuencia de longitud fija (definida por el parámetro sequence_length), con un número de características determinado por la forma de la secuencia de entrada.
  - Capa LSTM: Se emplea una capa LSTM con 64 unidades y con return_sequences=True, lo que permite que la salida de la capa LSTM sea una secuencia en sí misma, apta para pasar a la siguiente capa recurrente.
  - Dropout: Después de la primera capa LSTM, se añade una capa de Dropout del 20 % para prevenir el sobreajuste.
  - Segunda Capa LSTM: Se utiliza una segunda capa LSTM con 32 unidades y return_sequences=False, lo que reduce la secuencia a una única salida.
  - Capa densa: Se añaden dos capas densas; la primera con 25 neuronas y activación ReLU, y la segunda con tantas neuronas como clases de salida (num_classes), usando una función de activación softmax para clasificación multiclase.
- **Estructura del Modelo RNN:**
  - Capa de entrada: La capa de entrada toma secuencias de la misma longitud y estructura que en el modelo LSTM.
  - Capa RNN: Se emplea una capa SimpleRNN con 64 unidades y return_sequences=True para generar una secuencia de salida.
  - Dropout: Al igual que en el modelo LSTM, se introduce una capa de Dropout para prevenir sobreajuste.
  - Segunda Capa RNN: La segunda capa recurrente tiene 32 unidades, con return_sequences=False, reduciendo la secuencia a una única salida.
  - Capa densa: Se añaden dos capas densas, de la misma forma que en la LSTM: una con 25 neuronas y activación ReLU, y otra con activación softmax para las clases de salida.

Para erradicar el problema del desbalanceo de los datos se ha optado por la ponderación de los pesos de las diferentes clases mediante la función *compute_class_weight()* [42] de la librería Sklearn. Los pesos se calculan:

$$w_i = \frac{N}{n_i \times C} \tag{1}$$

Donde:

- w_i es el peso asignado a la clase i.
- N es el número total de muestras.
- n_i es el número de muestras de la clase i.
- C es el número de clases.

Ambos modelos se compilan con el optimizador *Adam* y con la función de pérdida *categorical_crossentropy*. Asimismo, se entrenaron durante 30 épocas utilizando diferentes tamaños de lote, que se incrementaron progresivamente entre 32, 64, 128 y 256.

Una vez diseñadas las redes neuronales, se implementan diversas estrategias de análisis para evaluar su rendimiento bajo diferentes condiciones. Estas estrategias incluyen:

- Se entrenaron modelos RNN con ventanas temporales de tamaño 5 y 10 minutos, con el fin de analizar el impacto del contexto temporal en la precisión de la clasificación, variando el tamaño de lote entre 32 y 256 para analizar su influencia en el rendimiento del modelo.
- De igual forma, se entrenaron modelos LSTM utilizando los mismos tamaños de ventana (5 y 10 minutos) y rangos de tamaño de lote (32 a 256), con el objetivo de comparar

su comportamiento frente a las RNN en la detección de patrones temporales.

Los modelos fueron entrenados de manera independiente con dos conjuntos de datos distintos, los cuales se dividieron internamente en subconjuntos de entrenamiento, validación y prueba (70 %, 20 % y 10 % respectivamente). Posteriormente, la evaluación final de los modelos se realizó sobre un tercer conjunto de datos independiente, no utilizado durante las fases de entrenamiento ni validación, con el fin de garantizar una estimación objetiva de su desempeño.

III-G. Evaluación de la precisión de los modelos de aprendizaje profundo

La evaluación de la precisión de los modelos diseñados se realiza a través de la función *Evaluation(model, X_test, y_test)*. Esta función tiene como objetivo cuantificar el rendimiento del modelo aplicado a un conjunto de datos de prueba (X_test y y_test) utilizando métricas de evaluación.

- Cálculo de métricas:
  - **Pérdida (Loss):** es una métrica fundamental para evaluar el rendimiento de un modelo de aprendizaje profundo. Representa el error cometido por el modelo al comparar sus predicciones con los valores reales durante el entrenamiento y la validación.
  - **Accuracy:** La precisión es la proporción de predicciones correctas sobre el total de casos evaluados. Formalmente, se puede definir como:

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}} \tag{2}$$

    donde:
    - TP = Verdaderos positivos (True Positives)
    - TN = Verdaderos negativos (True Negatives)
    - FP = Falsos positivos (False Positives)
    - FN = Falsos negativos (False Negatives)
  - **F1-Score:** El F1-Score es la media armónica entre la precisión y la exhaustividad (Recall). Se utiliza para balancear ambos aspectos en escenarios donde las clases están desbalanceadas.

$$\text{F1-Score} = 2 \cdot \frac{\text{Precision} \cdot \text{Recall}}{\text{Precision} + \text{Recall}} \tag{3}$$

  - **Precisión:** La precisión es la proporción de verdaderos positivos sobre el total de predicciones positivas. Indica la exactitud de las predicciones positivas del modelo.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}} \tag{4}$$

  - **Recall:** El recall, o exhaustividad, es la proporción de verdaderos positivos sobre el total de casos reales positivos. Mide la capacidad del modelo para identificar todos los casos positivos.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}} \tag{5}$$

  - **Matriz de confusión (Confusion Matrix):** Es una tabla que muestra la cantidad de predicciones correctas e incorrectas distribuidas entre las diferentes clases. Se utiliza para visualizar el rendimiento del modelo en cada clase específica. La matriz tiene la siguiente estructura general:

		Predicción	
		Clase Positiva	**Clase Negativa**
Real	**Positiva**	TP	FN
	Negativa	FP	TN

- **Porcentaje de aciertos y fallos:** se compara la diferencia entre las clases predichas y las reales, y se cuenta el número de predicciones incorrectas y correctas. A partir de esto, se calcula el porcentaje de aciertos y fallos.

IV. Resultados

En esta sección se presentan los resultados obtenidos tras el entrenamiento, validación y prueba de los modelos generados con las arquitecturas RNN y LSTM. Se usaron dos conjuntos de datos de entrenamiento distintos, denominados Dataset 1 y Dataset 2, que incorporan diferentes preprocesamientos.

El objetivo principal es comparar el desempeño de las distintas configuraciones probadas y analizar cómo afectan los factores diferentes al rendimiento final. Por último, se realizó una generalización de los modelos usando el tercer conjunto de datos que denominaremos Dataset 3.

Dado el elevado número de configuraciones, se han incluido únicamente los resultados más representativos en tablas y figuras. El resto de los valores detallados se encuentra en el Notebook del proyecto.

En las matrices de confusión obtenidas, las clases predichas y reales están representadas mediante etiquetas numéricas (0, 1, 2, 3) que se corresponden con las distintas etapas del sueño. Concretamente, la clase 0 representa el sueño ligero (etiqueta original 4), la clase 1 corresponde al sueño profundo (etiqueta 5), la clase 2 al estado de vigilia o despierto (etiqueta 7), y la clase 3 a la fase REM (etiqueta 8). Esta reclasificación se realizó para facilitar la interpretación y el tratamiento computacional de las diferentes etapas durante el entrenamiento y evaluación de los modelos.

IV-A. Métricas de evaluación durante el entrenamiento

- Modelos LSTM, entrenados con Dataset 1.
  - **Tamaño de ventana igual a 5.** En primer lugar, se utilizó el conjunto de datos 1, que corresponde al usuario 1, para entrenar la red neuronal durante 30 épocas, registrando las métricas comentadas en el apartado anterior. En este caso, se observó un rendimiento relativamente homogéneo en los distintos tamaños de lote, con un F1-score ponderado alrededor de 0,69–0,70. Se observó que a medida que el tamaño de lote disminuía, las métricas globales como accuracy y F1-score mejoraban de forma consistente, alcanzando su punto más alto con el tamaño de lote de 32. El modelo con tamaño de

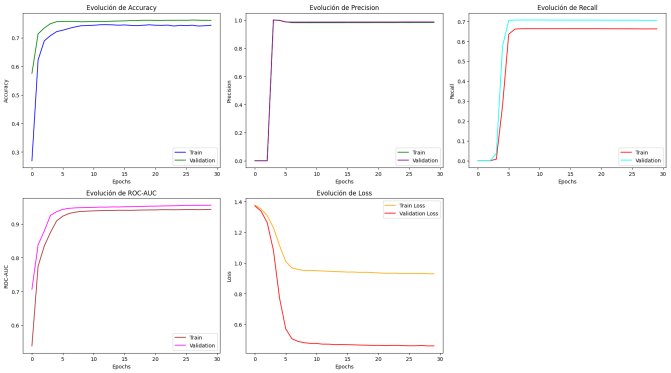

Figura 5. Métricas del entrenamiento y validación en el modelo LSTM con un tamaño de ventana igual a 5 y un batchsize igual a 32 usando el conjunto de datos 1.

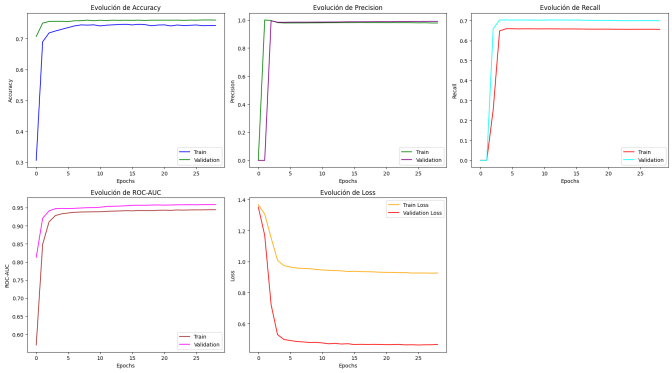

Figura 6. Métricas del entrenamiento y validación en el modelo LSTM con un tamaño de ventana igual a 10 y un batchsize igual a 32 usando el conjunto de datos 1.

lote 32 obtuvo la mejor combinación de métricas: F1-score de 0,6953, accuracy del 0,7114 y un ROC-AUC de 0,9306, siendo además el que presentó la menor loss (0,5746), como se puede ver en la Figura 5. A medida que el tamaño de lote aumenta, el recall tiende a disminuir ligeramente, lo que sugiere una menor capacidad para detectar correctamente todas las clases, especialmente las minoritarias.

La evaluación de este modelo en el conjunto de datos 3 reveló un descenso en el rendimiento. La matriz de confusión mostró que los modelos tenían mayores dificultades para clasificar correctamente las clases menos representadas, particularmente la clase 0 como se puede ver en la Tabla II, lo que refleja que los LSTM con ventana pequeña son más sensibles a cambios cuando los datos de test son distintos al de entrenamiento.

Tabla II
RESULTADOS DEL MODELO LSTM CON TAMAÑO DE VENTANA 5 Y TAMAÑO DE BATCH DE 32 EN EL TEST CON EL DATASET 3.

Métrica	**Valor**
Porcentaje de acierto (%)	75,93
Porcentaje de fallo (%)	24,07
Precisión (Accuracy)	0,7593
F1-Score (weighted)	0,7368
Precision	0,7898
Recall	0,7593

Clase	**Clase 0**	**Clase 1**	**Clase 2**	**Clase 3**
Clase 0	**397**	2901	1368	3982
Clase 1	78	**1139**	66	1240
Clase 2	302	466	**34203**	668
Clase 3	70	991	25	**2617**

- **Tamaño de ventana igual a 10.** Cuando se aumenta el tamaño de la ventana a 10, los modelos LSTM muestran una ligera mejoría en la mayoría de las métricas. Esto es coherente con la hipótesis de que una secuencia temporal más larga permite al modelo capturar mejor la evolución del sueño a lo largo del tiempo, incluyendo patrones de transición entre etapas que podrían ser más difíciles de identificar con una ventana más pequeña.

  El mejor modelo en esta configuración también es el que tiene un tamaño de lote de 32 y alcanza un F1-score de 0,6968, el más alto registrado entre los LSTM entrenados con el conjunto de datos 1. Este modelo también mantiene un alto nivel de precisión (98,54 %) y recall (64,13 %), lo que sugiere un equilibrio adecuado entre sensibilidad y especificidad como se puede observar en la Figura 6. Cabe destacar que incluso con tamaños de lotes más grandes, el rendimiento sigue siendo elevado. Por ejemplo, con tamaño de lote 64 se alcanza un F1 de 0,6930, confirmando la robustez de esta configuración temporal.

  En conjunto, los resultados indican que las LSTM, cuando se entrenan con secuencias más largas, son capaces de extraer características temporales ligeramente más ricas del sueño, lo cual repercute en una leve mejoría del rendimiento en las métricas de evaluación como se puede ver en la Tabla III, especialmente en aquellas que penalizan los errores de clasificación de clases minoritarias.

Tabla III
RESULTADOS DEL MODELO LSTM CON TAMAÑO DE VENTANA 10 Y TAMAÑO DE BATCH DE 32 EN EL TEST CON EL DATASET 3.

Métrica	**Valor**
Porcentaje de acierto (%)	75,38
Porcentaje de fallo (%)	24,62
Precisión (Accuracy)	0,7538
F1-Score (weighted)	0,7317
Precisión	0,8502
Recall	0,7538

Clase	**Clase 0**	**Clase 1**	**Clase 2**	**Clase 3**
Clase 0	**220**	3176	947	4305
Clase 1	17	**1196**	75	1235
Clase 2	36	792	**34029**	777
Clase 3	7	1050	20	**2626**

- Modelos RNN, entrenados con Dataset 1.

- **Tamaño de ventana igual a 5.** En el caso de las redes RNN simples entrenadas con una ventana de tamaño 5, se observa un patrón similar al de las LSTM: los modelos con lotes más pequeños suelen obtener mejores resultados. El modelo con tamaño de lote 32 logra un F1-score ponderado de 0,7045, superior al mejor resultado LSTM con esta misma ventana como se puede ver en la Figura 7. Este modelo alcanza también una precisión de 98,65 % y un recall de 64,03 %, métricas destacables para una arquitectura menos compleja que la LSTM.

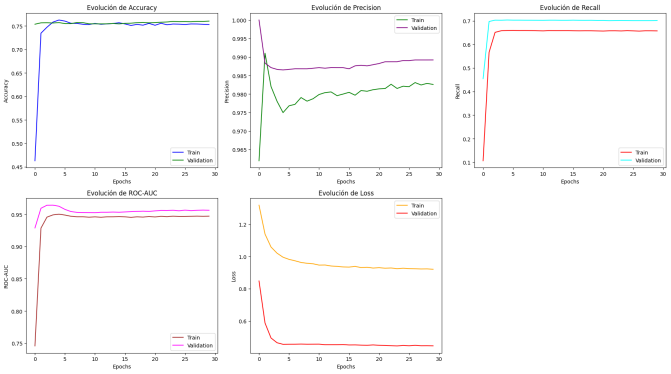

Figura 7. Métricas del entrenamiento y validación en el modelo RNN con un tamaño de ventana igual a 5 y un batchsize igual a 32 usando el conjunto de datos 1.

Sin embargo, el resto de configuraciones de tamaño de lote muestran resultados algo más dispersos. El modelo con tamaño de lote 256, por ejemplo, aunque mantiene una buena precisión, cae en términos de recall y F1-score, lo cual sugiere una posible pérdida de capacidad de generalización a causa del tamaño del lote y de la menor granularidad durante el aprendizaje.

En el test con el conjunto de datos 3, se puede observar que las clases minoritarias siguen siendo un problema para el modelo, aunque muestran un mayor número de aciertos que en los modelos anteriores en determinadas clases. En la Tabla IV se ve como la matriz de confusión revela que la arquitectura RNN tiene dificultades para distinguir las clases 0 y 3, probablemente debido a las similitudes de estas etapas del sueño.

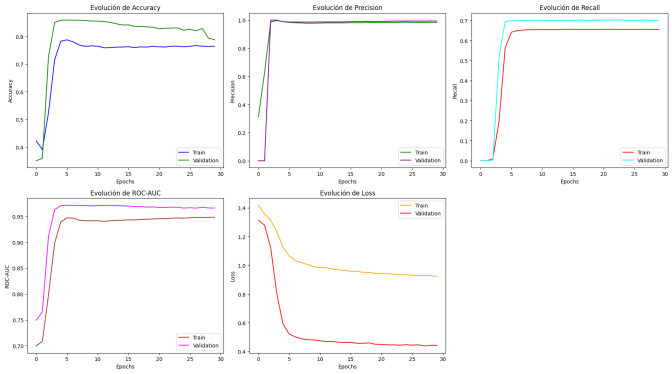

Figura 8. Métricas del entrenamiento y validación en el modelo RNN con un tamaño de ventana igual a 10 y un batchsize igual a 128 usando el conjunto de datos 1.

Tabla IV
RESULTADOS DEL MODELO RNN CON TAMAÑO DE VENTANA 5 Y TAMAÑO DE LOTE 32 EN EL TEST CON EL DATASET 3.

Métrica	Valor
Porcentaje de acierto (%)	76,53
Porcentaje de fallo (%)	23,47
Precisión (Accuracy)	0,7653
F1-Score (weighted)	0,7642
Precision	0,8294
Recall	0,7653

Clase	Clase 0	Clase 1	Clase 2	Clase 3
Clase 0	**1221**	4695	709	2023
Clase 1	105	**1940**	29	449
Clase 2	585	659	**34015**	380
Clase 3	103	2105	15	**1480**

- **Tamaño de ventana igual a 10.** Cuando se amplía el tamaño de ventana a 10 en las redes RNN, los resultados mejoran notablemente en comparación con las configuraciones anteriores. Esta mejora es especialmente visible en el modelo con tamaño de lote de 128 como se puede ver en la Figura 8, que alcanza el mejor F1-score ponderado de todo el conjunto de experimentos con un valor de 0,7935, junto con una precisión de 98,93 % y un accuracy del 78,92 %. Estos valores superan incluso a los mejores modelos LSTM, lo que resulta interesante desde el punto de vista de la eficiencia arquitectónica y computacional.

  Este comportamiento destaca la capacidad de las RNN para beneficiarse de ventanas temporales más amplias, lo cual puede deberse a que la arquitectura tiene más oportunidades para capturar patrones recurrentes en los datos secuenciales. También es destacable que el rendimiento en recall no decrece significativamente con respecto a otros modelos, manteniéndose en torno al 64 %, lo que indica una adecuada cobertura de las clases.

  Aunque al testear este modelo con el conjunto de datos 3, se puede ver en la Tabla V que mejora el clasificafión de las clases 0, 1 y 2 pero empeora el

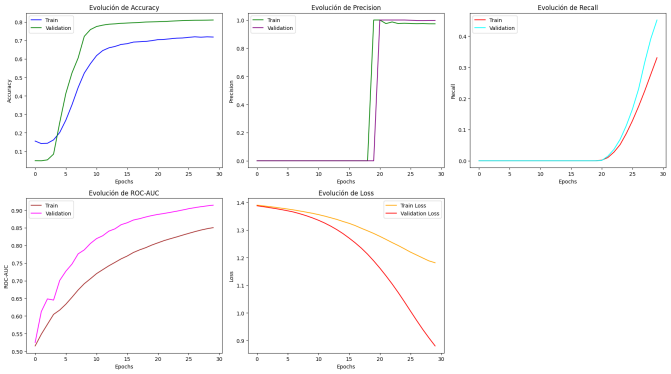

Figura 9. Métricas del entrenamiento y validación en el modelo LSTM con un tamaño de ventana igual a 5 y un batchsize igual a 256 usando el conjunto de datos 2.

de la clase 3. Este patrón de confusión sugiere que, aunque el modelo es competente para clasificar las fases de sueño predominantes, tiene dificultades en distinguir con precisión las clases menos frecuentes, lo que podría explicarse por el desbalanceo de clases presente en el conjunto de datos.

Tabla V
RESULTADOS DEL MODELO RNN CON TAMAÑO DE VENTANA 10 Y TAMAÑO DE LOTE 128 EN EL TEST CON EL DATASET 3.

Métrica	Valor
Porcentaje de acierto (%)	79,71
Porcentaje de fallo (%)	20,29
Precisión (Accuracy)	0,7971
F1-Score (weighted)	0,7899
Precision	0,7995
Recall	0,7971

Clase	Clase 0	Clase 1	Clase 2	Clase 3
Clase 0	**4491**	2970	922	265
Clase 1	885	**1569**	56	13
Clase 2	1017	384	**34101**	132
Clase 3	2380	1176	49	**98**

- Modelos LSTM, entrenados con Dataset 2.
  - **Tamaño de ventana igual a 5.** En el caso del dataset 2, los modelos LSTM con ventana de tamaño 5 muestran una tendencia contraria a lo observado anteriormente: los lotes grandes tienden a ofrecer un mejor rendimiento global. Concretamente, el modelo con tamaño de lote 256 alcanza el mejor F1-score ponderado dentro de esta configuración, con un valor de 0,7903, así como una precisión de 98,12 % y un ROC-AUC de 90,02 %, valores muy competitivos y estables que se pueden ver en la Figura 9.
    Aunque al evaluarlo en el Dataset 3 mostró un porcentaje de acierto del 81,56 % como se puede ver en la Tabla VI, destacando como uno de los mejores resultados obtenidos en términos de precisión global. La matriz de confusión revela que el modelo logra clasificar correctamente la mayoría de instancias de la clase 2, pero muestra un claro sesgo: prácticamente no predice ninguna instancia como clase 1. Este comportamiento puede deberse tanto al desbalanceo de clases como a posibles diferencias en la distribución temporal de los eventos entre los datasets de entrenamiento y prueba, reflejando limitaciones del modelo para generalizar en fases menos representadas como la clase 1.

Tabla VI
RESULTADOS DEL MODELO LSTM CON TAMAÑO DE VENTANA 5 Y TAMAÑO DE LOTE 256 EN EL TEST CON EL DATASET 3.

Métrica	Valor
Porcentaje de acierto (%)	81,56
Porcentaje de fallo (%)	18,44
Precisión (Accuracy)	0,8156
F1-Score (weighted)	0,7815
Precision	0,7639
Recall	0,8156

Clase	Clase 0	Clase 1	Clase 2	Clase 3
Clase 0	**7142**	0	1336	170
Clase 1	2410	**0**	76	37
Clase 2	1346	0	**34037**	256
Clase 3	3662	0	22	**19**

  - **Tamaño de ventana igual a 10.** Con el aumento del tamaño de ventana a 10, se observa un leve descenso en el rendimiento de los modelos, tanto en las métricas principales como en la estabilidad entre configuraciones. El modelo con tamaño de lote 128 alcanza el mejor F1-score ponderado en esta sección, con un valor de 0,7125. También destaca con un accuracy de 70,26 % y una precisión del 96,27 %, lo que pone de manifiesto su gran capacidad de predicción.
    Al evaluar este modelo en el conjunto de datos 3, vemos en la Tabla VII un empeoramiento de la clasificación de las clases minoritarias con respecto a la evaluación con tamaño de ventana 5. Este resultado sugiere que el aumento del tamaño de ventana no benefició al modelo, ya que podría haber introducido ruido o dificultado la captura de patrones temporales relevantes, afectando negativamente la precisión en fases que son más difíciles de distinguir.

Tabla VII
RESULTADOS DEL MODELO LSTM CON TAMAÑO DE VENTANA 10 Y TAMAÑO DE LOTE 128 EN EL TEST CON EL DATASET 3.

Métrica	Valor
Porcentaje de acierto (%)	72,99
Porcentaje de fallo (%)	27,01
Precisión (Accuracy)	0,7299
F1-Score (weighted)	0,7130
Precision	0,7724
Recall	0,7299

Clase	Clase 0	Clase 1	Clase 2	Clase 3
Clase 0	**758**	6415	1364	111
Clase 1	216	**2064**	198	45
Clase 2	334	1137	**34024**	139
Clase 3	195	3407	79	**22**

- Modelos RNN, entrenados con Dataset 2.
  - **Tamaño de ventana igual a 5.** En esta configuración, los modelos RNN se comportan de forma similar a lo observado con las redes LSTM, aunque con ligeras mejoras. El modelo con tamaño de lote 256 destaca con un F1-score de 0,7764, precisión del 97,13 % y un recall del 61,30 %. Estos valores demuestran que incluso sin mecanismos avanzados de memoria, como los usados en LSTM, las RNN simples pueden aprender patrones temporales útiles con una ventana de tiempo reducida.
    Al evaluarlo en el conjunto de datos 3 vemos un rendimiento general adecuado, aunque la matriz de confusión, que se puede ver en la Tabla VIII revela problemas notables: especialmente en la clase 0, donde 3978 instancias fueron clasificadas como clase 1, y en la clase 3, con 2532 instancias mal asignadas como clase 1. Estos errores indican que el modelo tiende a confundir estas clases minoritarias, sugiriendo que la RNN podría beneficiarse de técnicas adicionales como regularización o ajuste fino para mejorar la diferenciación entre clases.

Tabla VIII
RESULTADOS DEL MODELO RNN CON TAMAÑO DE VENTANA 5 Y TAMAÑO DE LOTE 256 EN EL TEST CON EL DATASET 3.

Métrica	Valor
Porcentaje de acierto (%)	76,20
Porcentaje de fallo (%)	23,80
Precisión (Accuracy)	0,7620
F1-Score (weighted)	0,7611
Precision	0,7761
Recall	0,7620

Clase	Clase 0	Clase 1	Clase 2	Clase 3
Clase 0	**3212**	3978	1338	120
Clase 1	1155	**1240**	92	36
Clase 2	702	663	**34019**	255
Clase 3	1122	2532	29	**20**

  - **Tamaño de ventana igual a 10.** Finalmente, en los modelos RNN entrenados con ventana de tamaño 10, se obtienen los mejores resultados en tamaños de lote intermedios. El modelo con tamaño de lote 128 sobresale con un F1-score ponderado de 0,7851, junto con un accuracy de 79,07 %, una precisión de 96,83 % y un ROC-AUC de 0,9130.
    Aunque al evaluarlo con el conjunto de datos 3, la matriz de confusión revela que al clasificar las clases 1 y 3, se presentan numerosos errores en comparación con las otras clases, como se puede ver en la Tabla IX

Tabla IX
RESULTADOS DEL MODELO RNN (VENTANA 10, TAMAÑO DE LOTE 128, ENTRENADO CON DATASET 2) EVALUADO EN EL DATASET 3.

Métrica	Valor
Porcentaje de acierto (%)	80,39
Porcentaje de fallo (%)	19,61
Precisión (Accuracy)	0,8039
F1-Score (weighted)	0,7783
Precision	0,7689
Recall	0,8039

Clase	Clase 0	Clase 1	Clase 2	Clase 3
Clase 0	**6726**	590	1174	158
Clase 1	2297	**64**	119	43
Clase 2	1498	109	**33784**	243
Clase 3	3476	155	43	**29**

En conjunto, esta sección demuestra que, cuando se utiliza un contexto temporal amplio (ventana 10) y se selecciona adecuadamente el tamaño de lote, las redes RNN pueden ser una solución muy competitiva para la clasificación de etapas del sueño.

V. DISCUSIÓN

En el campo del aprendizaje automático, la elección de modelos y la evaluación de su rendimiento son aspectos cruciales para una correcta implementación de soluciones basadas en datos. En este análisis, se compara el desempeño de redes neuronales recurrentes y redes neuronales de memoria a corto - largo plazo. Esta comparación ofrece una visión integral sobre cómo estos modelos manejan datos secuenciales y la efectividad de la validación cruzada en la evaluación del rendimiento.

V-A. Rendimiento comparativo y comportamiento por arquitectura

Los LSTM entrenados con el dataset 1 mostraron un rendimiento sólido en ambos tamaños de ventana, con accuracies cercanas al 74-76 % al evaluar en el dataset 3. No obstante, el LSTM con ventana 5 y tamaño de lote 32 destacó ligeramente por su estabilidad y mejor equilibrio entre precisión y recall en las clases minoritarias. El hecho de que tanto el modelo con ventana 5 como el de ventana 10 alcanzaran rendimientos similares sugiere que, para secuencias de sueño derivadas de registros relativamente homogéneos como el dataset 1, el tamaño de la ventana temporal no es el factor determinante,

siendo más relevante el tamaño de lote en conjunción con la arquitectura.

Sin embargo, al pasar a los RNN, se observa que el rendimiento mejora significativamente al aumentar la longitud de la ventana a 10 (tamaño de lote 128) respecto al tamaño de ventana 5 (tamaño de lote 32). Por ejemplo, el modelo RNN ventana 10 y tamaño de lote 128 entrenado con el dataset 1 alcanzó un accuracy del 80 % en el test con el dataset 3, frente a un 76 % para el modelo RNN ventana 5 y tamaño de lote 32. Esto evidencia que los RNN simples, carentes de mecanismos explícitos de memoria como los gates de LSTM, requieren ventanas temporales más largas para capturar la secuencia de etapas, compensando su menor capacidad intrínseca para retener información en pasos temporales prolongados.

Para los modelos entrenados con el dataset 2, el rendimiento fue aún más llamativo. El modelo LSTM con ventana 5 y tamaño de lote 256 consiguió accuracies superiores al 81.5 % en el test, convirtiéndose en el modelo con mejor desempeño absoluto. Su estabilidad se refleja en las métricas secundarias: F1-score 0,78 y precisión cercana al 0,76, mostrando balance entre la discriminación de clases y la reducción de falsos positivos. De forma complementaria, el modelo RNN ventana 10 y tamaño de lote 128 también alcanzó un rendimiento destacado (80,4 % accuracy), demostrando que, en secuencias más largas y con tamaños de lote que permiten un aprendizaje más estable al poder capturar una mejor diversidad de los patrones fisiológicos sin sobreajustarse, los RNN pueden acercarse al rendimiento de los LSTM, incluso en entornos de mayor heterogeneidad como el conjunto de datos 2.

V-B. Impacto del desbalanceo de clases

Uno de los mayores retos en este proyecto fue el desbalanceo entre las clases: las clases 1 y 3 (que se corresponden a sueño profundo y fase REM) se encontraban ampliamente subrepresentadas frente a la 2, que se corresponde con estar despierto. Este desbalanceo se tradujo en matrices de confusión con bajas tasas de recall para 1 y 3, incluso en los modelos que alcanzaron mejores resultados globales. A pesar de haber incorporado ponderaciones de clase en la función de pérdida —una técnica ampliamente empleada para mitigar el desbalanceo— los incrementos en el recall de clases minoritarias fueron moderados (mejoras del 3-5 % respecto a entrenamiento sin ponderación), evidenciando la limitación de este enfoque cuando el desbalanceo es extremo.

Este sesgo hacia la clase 2 es especialmente problemático desde la perspectiva clínica, donde identificar correctamente despertares y transiciones a sueño ligero (clase 0) resulta crítico para diagnosticar trastornos como insomnio, apnea o despertares frecuentes. El predominio de predicciones correctas en la clase despierto y sueño ligero, pero con confusión frecuente entre las otras clases, indica que los modelos no aprendieron características suficientes para discriminar eventos breves y poco frecuentes.

V-C. Calidad de los datos

La calidad de los registros desempeñó un papel fundamental en los resultados. En ambos conjuntos de datos se observaron artefactos como picos abruptos en la variable 'HEART_RATE' o errores de sincronización, que dificultan el aprendizaje estable. Las arquitecturas LSTM, aunque más robustas que los RNN a la pérdida de información en pasos individuales, tienden a amplificar la influencia de artefactos por el mecanismo de estados internos: un error grande en un paso puede propagarse y alterar las predicciones en varias ventanas consecutivas. Este efecto es especialmente notorio en el conjunto de datos 2, donde algunos registros incluían secciones problemáticas que coincidían con aumentos abruptos de la pérdida durante el entrenamiento.

A nivel práctico, estos problemas de calidad de datos señalan la necesidad de incorporar procedimientos de preprocesado avanzados como filtrado adaptativo [43] que podrían reducir la cantidad de información que confunde a las redes. Estos métodos solo son aplicables a los datos crudos, a los que no tenemos acceso ya que el dispositivo wearable los procesa.

Finalmente, se puede llegar a pensar que los problemas ocasionados en los modelos entrenados con el conjunto de datos 2 pueden ser porque en la etapa de preprocesado, se encontró un periodo de tiempo de una semana que se tuvo que eliminar por la mala monitorización del dispositivo wearable (marcaba que el usuario estaba despierto todo el rato y recogía que la frecuencia cardíaca por minuto era de 255). Al eliminar registros de datos extensos, generamos vacíos temporales que complican la clasificación debido a la falta de información temporal.

Además, también hay que tener en cuenta que la variable 'RAW_KIND' son números codificados. No se sabe a qué corresponden estos números, pudiendo ser un problema para generalizar lo aprendido en los modelos a datos nuevos.

VI. Conclusiones

Los modelos LSTM y RNN presentan diferencias notables en cuanto a su rendimiento en la predicción de patrones de sueño utilizando datos de dispositivos wearables. Los RNN han demostrado ser superiores en términos de precisión, consistencia y capacidad de generalización en comparación con los LSTM. Esta superioridad se refleja no solo en las métricas de precisión, sino también en la capacidad de los LSTM para manejar secuencias largas y complejas [44], lo que es crucial en la predicción de patrones de sueño, donde la temporalidad y las dependencias a largo plazo juegan un papel fundamental.

Por otro lado, los RNN, aunque útiles en ciertas aplicaciones, mostraron una mayor variabilidad en su rendimiento, lo que los hace menos confiables para aplicaciones donde la precisión y la consistencia son críticas.

Dada la limitada cantidad de usuarios en este estudio (tres sujetos), los modelos podrían estar sobreajustándose a las características individuales de los usuarios del conjunto de entrenamiento. Para mejorar la capacidad de generalización y evitar que el rendimiento caiga al evaluar en nuevos individuos, sería necesario incluir más usuarios con perfiles variados,

lo que permitiría capturar patrones generales del sueño en lugar de particularidades de unos pocos sujetos.

VI-A. Ventajas y limitaciones de la tecnología de dispositivos wearable en la monitorización del sueño

Los dispositivos wearables ofrecen varias ventajas en la monitorización del sueño. Son accesibles, cómodos de usar y proporcionan datos en tiempo real sobre patrones de sueño, lo que permite un seguimiento continuo y detallado de la calidad del sueño. Además, la facilidad de recopilación de datos en un entorno natural (es decir, mientras el usuario duerme en su propia cama) hace que los wearables sean una herramienta valiosa para el análisis del sueño sin la necesidad de intervenciones clínicas invasivas.

Sin embargo, estos dispositivos también tienen limitaciones significativas. La precisión de los datos recopilados por los wearables puede verse afectada por la calidad de los sensores y la metodología utilizada para la interpretación de los datos. Además, los wearables a menudo se enfrentan a desafíos en la clasificación de diferentes etapas del sueño, y pueden ser menos efectivos para detectar trastornos del sueño complejos [45]. La variabilidad entre dispositivos y la falta de estandarización también pueden conducir a inconsistencias en los datos, lo que puede afectar la fiabilidad de las predicciones.

VI-B. Aplicaciones prácticas y futuras tendencias en la predicción del sueño con dispositivos wearables

La capacidad de los dispositivos wearables para monitorear el sueño tiene importantes aplicaciones prácticas. En el ámbito de la salud, estos dispositivos pueden utilizarse para identificar patrones de sueño deficientes, lo que permite a los usuarios tomar medidas correctivas para mejorar su calidad de vida. En la investigación médica, los wearables pueden proporcionar grandes volúmenes de datos sobre el sueño, lo que puede ser utilizado para estudiar el impacto del sueño en diversas condiciones de salud [46].

En el futuro, se espera que los avances en la tecnología de sensores y en los algoritmos de aprendizaje automático, como los modelos LSTM, mejoren la precisión y la utilidad de los wearables en la monitorización del sueño. La integración de datos de múltiples fuentes (por ejemplo, actividad física, temperatura corporal y datos biométricos) podría ofrecer una visión más completa del estado de salud de un individuo. Además, el uso de inteligencia artificial avanzada podría permitir la predicción proactiva de problemas de sueño, lo que llevaría a una intervención temprana y personalizada [47].

VI-C. Comparación entre modelos de aprendizaje profundo y métodos clínicos tradicionales

Al comparar modelos de aprendizaje profundo, como LSTM y RNN, con la polisomnografía (PSG), que es el estándar clínico para la clasificación del sueño, surgen varias observaciones clave.

Los modelos de aprendizaje profundo, especialmente los LSTM, han mostrado una precisión considerable en la clasificación de patrones de sueño utilizando datos de dispositivos wearables. Sin embargo, a pesar de su capacidad para manejar grandes volúmenes de datos y procesar secuencias temporales complejas, la precisión de estos modelos aún no alcanza los niveles proporcionados por la PSG. La PSG ofrece una evaluación detallada y multifacética del sueño, midiendo múltiples parámetros fisiológicos como la actividad cerebral (electroencefalograma - EEG), los movimientos oculares (EOG), y la actividad muscular (EMG) [48], lo que permite una clasificación más precisa y detallada de las etapas del sueño. En contraste, los modelos de aprendizaje profundo se basan en datos limitados provenientes de wearables, que pueden no capturar con la misma exactitud las transiciones sutiles entre diferentes etapas del sueño.

Una de las ventajas más significativas de los modelos de aprendizaje profundo aplicados a datos de dispositivos wearables es su accesibilidad. A diferencia de la PSG, que requiere un entorno clínico especializado y es costosa y laboriosa [49], los dispositivos wearables permiten la monitorización del sueño de manera continua en el hogar del usuario. Esto no solo reduce los costos, sino que también permite la recolección de datos durante períodos prolongados, lo que puede proporcionar una visión más representativa de los patrones de sueño de un individuo. Esto es particularmente útil para el análisis longitudinal del sueño y la detección de trastornos del sueño que pueden no ser evidentes en una única noche de PSG.

A pesar de sus ventajas, los modelos de aprendizaje profundo enfrentan limitaciones importantes cuando se comparan con la PSG. La PSG es capaz de clasificar de manera precisa las diferentes etapas del sueño (NREM1, NREM2, NREM3 y REM), mientras que los modelos de aprendizaje profundo pueden tener dificultades para distinguir entre algunas de estas etapas debido a la naturaleza más limitada de los datos de los wearables. Además, los wearables pueden ser menos efectivos para detectar eventos específicos como apneas del sueño, que son fácilmente identificables mediante PSG. Esta diferencia en la capacidad de clasificación puede llevar a una subestimación o sobreestimación de ciertas etapas del sueño o de la presencia de trastornos del sueño, lo que es una limitación crítica cuando se trata de diagnósticos clínicos.

VI-D. Desafíos y oportunidades para el desarrollo futuro de esta tecnología

El uso de aprendizaje profundo para la clasificación del sueño está en constante evolución, y con los avances en tecnología de sensores y algoritmos, la brecha entre estos modelos y la PSG podría reducirse. Futuras tendencias incluyen la combinación de datos de múltiples sensores y la integración de señales adicionales, como el EEG portátil, para mejorar la precisión de los modelos. Además, el desarrollo de técnicas híbridas que utilicen aprendizaje profundo junto con métodos tradicionales podría ofrecer un compromiso entre la precisión de la PSG y la accesibilidad de los dispositivos wearables [50].

AGRADECIMIENTOS

A mi pareja por su esfuerzo y apoyo por acompañarme en este proyecto. A mis tutoras por su paciencia y por todo el

apoyo que me han brindado. Y a mis amigos y familiares, por el apoyo que siempre me dan.

REFERENCIAS

[1] Esteva, A., Robicquet, A., Ramsundar, B., Kuleshov, V., DePristo, M., Chou, K., Cui, C., Corrado, G., Thrun, S., & Dean, J. (2019). A guide to deep learning in healthcare. Nature medicine, 25(1), 24–29. https://doi.org/10.1038/s41591-018-0316-z

[2] Litjens, G., Kooi, T., Bejnordi, B. E., Setio, A. A. A., Ciompi, F., Ghafoorian, M., van der Laak, J. A. W. M., van Ginneken, B., & Sánchez, C. I. (2017). A survey on deep learning in medical image analysis. Medical image analysis, 42, 60–88. https://doi.org/10.1016/j.media.2017.07.005

[3] Adum Ruíz, J. H., Ruíz Ortega, M. G., Vera Ponce, H. J., & Álvarez Narváez, M. I. (2024). Inteligencia artificial en medicina: presente y futuro. RECIAMUC, 8(1), 166-177. https://doi.org/10.26820/reciamuc/8.(1).ene.2024.166-177

[4] Riccardo Miotto, Fei Wang, Shuang Wang, Xiaoqian Jiang, Joel T Dudley, Deep learning for healthcare: review, opportunities and challenges, Briefings in Bioinformatics, Volume 19, Issue 6, November 2018, Pages 1236–1246, https://doi.org/10.1093/bib/bbx044

[5] D. -E. -M. Nisar, R. Amin, N. -U. -H. Shah, M. A. A. Ghamdi, S. H. Almotiri and M. Alruily, "Healthcare Techniques Through Deep Learning: Issues, Challenges and Opportunities,ïn IEEE Access, vol. 9, pp. 98523-98541, 2021, doi: 10.1109/ACCESS.2021.3095312

[6] Yang, C. C., & Hsu, Y. L. (2010). A review of accelerometry-based wearable motion detectors for physical activity monitoring. Sensors, 10(8), 7772-7788.

[7] Arriba-Pérez, F D., Rodríguez, M C., & Santos-Gago, J M. (2016, September 21). Collection and Processing of Data from Wrist Wearable Devices in Heterogeneous and Multiple-User Scenarios. Multidisciplinary Digital Publishing Institute, 16(9), 1538-1538. https://doi.org/10.3390/s16091538

[8] Zhang, X., Kou, W., Eric, I., Chang, C., Gao, H., Fan, Y., & Xu, Y. (2018). Sleep stage classification based on multi-level feature learning and recurrent neural networks via wearable device. Computers in biology and medicine, 103, 71-81. https://doi.org/10.1016/j.compbiomed.2018.10.010

[9] Sun, C., Shrivastava, A., Singh, S., & Gupta, A. (2017). Revisiting unreasonable effectiveness of data in deep learning era. In Proceedings of the IEEE international conference on computer vision (pp. 843-852).

[10] Crivello, A., Barsocchi, P., Girolami, M., & Palumbo, F. (2019). The meaning of sleep quality: a survey of available technologies. IEEE access, 7, 167374-167390. DOI:10.1109/ACCESS.2019.2953835

[11] Goldstein, A. N., & Walker, M. P. (2014). The role of sleep in emotional brain function. Annual review of clinical psychology, 10, 679–708. https://doi.org/10.1146/annurev-clinpsy-032813-153716

[12] Irwin, M. R. (2019). Sleep and inflammation: partners in sickness and in health. Nature Reviews Immunology, 19(11), 702-715. doi:10.1038/s41577-019-0190-z

[13] Colten, H. R., Altevogt, B. M., & Institute of Medicine (US) Committee on Sleep Medicine and Research (Eds.). (2006). Sleep Disorders and Sleep Deprivation: An Unmet Public Health Problem. National Academies Press (US). DOI: 10.17226/11617

[14] Medic, G., Wille, M., & Hemels, M. E. (2017). Short- and long-term health consequences of sleep disruption. Nature and science of sleep, 9, 151–161. https://doi.org/10.2147/NSS.S134864

[15] Goldstein, A. N., & Walker, M. P. (2014). The Role of Sleep in Emotional Brain Function. Annu. Rev. Clin. Psychol, 10, 23-1.

[16] Sletten, T. L., Weaver, M. D., Foster, R. G., Gozal, D., Klerman, E. B., Rajaratnam, S. M., ... & Czeisler, C. A. (2023). The importance of sleep regularity: a consensus statement of the National Sleep Foundation sleep timing and variability panel. Sleep Health, 9(6), 801-820. https://doi.org/10.1016/j.sleh.2023.07.016

[17] Pino, R. (2024, 16 marzo). El sueño. FisioEspinal. https://fisioespinal.es/el-sueno/

[18] Worley S. L. (2018). The Extraordinary Importance of Sleep: The Detrimental Effects of Inadequate Sleep on Health and Public Safety Drive an Explosion of Sleep Research. P & T : a peer-reviewed journal for formulary management, 43(12), 758–763.

[19] Fabres, L., & Moya, P. (2021). Sueño: conceptos generales y su relación con la calidad de vida. Revista Médica Clínica Las Condes, 32(5), 527-534. https://doi.org/10.1016/j.rmclc.2021.09.001

[20] López de Castro, Francisco, Fernández Rodríguez, Olga, Fernández Agüero, Laura, Mareque Ortega, Mª Antonia, Alejandre Lázaro, Gemma, & Báez Montilla, Julia. (2011). Quality of life of people with insomnia in the Toledo health area. Revista Clínica de Medicina de Familia, 4(2), 92-99.

[21] Olivi, R. H. (2013). Apnea del sueño: cuadro clínico y estudio diagnóstico. Revista Médica Clínica Las Condes, 24(3), 359-373. https://doi.org/10.1016/S0716-8640(13)70173-1

[22] Medic, G., Wille, M., & Hemels, M. E. (2017). Short- and long-term health consequences of sleep disruption. Nature and science of sleep, 9, 151–161. https://doi.org/10.2147/NSS.S134864

[23] Nicte-Ha Tuz Castellanos, K., Lizcano Baños, A. J., Canche Garma, J. J., Juárez Sánchez, S. D., Domínguez Várguez, C. I., & Barrios de Tomasi, J. (2022). Síndrome de retraso de la fase del sueño: una revisión bibliográfica. Revista de la Facultad de Medicina (México), 65(1), 47-58. https://doi.org/10.22201/fm.24484865e.2022.65.1.08

[24] Lira, D., & Custodio, N. (2018). Los trastornos del sueño y su compleja relación con las funciones cognitivas. Revista de Neuro-Psiquiatría, 81(1), 20-28. http://dx.doi.org/https://doi.org/10.20453/rnp.v81i1.3270

[25] de León Arcila, R. (2018). Sueño, ciclos circadianos y obesidad. Archivos en medicina familiar, 20(3), 139-143.

[26] Aboalayon, K A I., Faezipour, M., Almuhammadi, W S., & Moslehpour, S. (2016, August 23). Sleep Stage Classification Using EEG Signal Analysis: A Comprehensive Survey and New Investigation. Multidisciplinary Digital Publishing Institute, 18(9), 272-272. https://doi.org/10.3390/e18090272

[27] Arora, A., Chakraborty, P. & Bhatia, M.P.S. Analysis of Data from Wearable Sensors for Sleep Quality Estimation and Prediction Using Deep Learning . Arab J Sci Eng 45, 10793–10812 (2020). https://doi.org/10.1007/s13369-020-04877-w

[28] Gadgetbridge contributors. Gadgetbridge, 2025. URL https://gadgetbridge.org/

[29] Pardo Otero, E., Fernández Garrido, I., Fernández Noriega Balseiro, S., Martínez Pérez, M., & Nieto-Riveiro, L. (2023). Solution for Capturing Data from Wearable Devices. En M. Lagos Rodríguez, Á. Leitao Rodríguez, & T. Varela Rodeiro (eds.), VI Congreso XoveTIC: impulsando el talento científico.

[30] Domingues, P., Francisco, J., & Frade, M. (2023). Post-mortem digital forensics analysis of the Zepp Life android application. Forensic Science International: Digital Investigation, 45, 301555. https://doi.org/10.1016/j.fsidi.2023.301555

[31] Kumar, A., Goel, S., Sinha, N., & Bhardwaj, A. (2022, May). A review on unbalanced data classification. In Proceedings of International Joint Conference on Advances in Computational Intelligence: IJCACI 2021 (pp. 197-208). Singapore: Springer Nature Singapore.

[32] Anand, A., Pugalenthi, G., Fogel, G.B. et al. An approach for classification of highly imbalanced data using weighting and undersampling. Amino Acids 39, 1385–1391 (2010). https://doi.org/10.1007/s00726-010-0595-2

[33] Pérez, E. R., & Medrano, L. A. (2010). Análisis factorial exploratorio: bases conceptuales y metodológicas. Revista Argentina de Ciencias del Comportamiento (RACC), 2(1), 58-66.

[34] Gómez, R. S., & Martínez, E. R. (2017). Métodos cuantitativos para un modelo de regresión lineal con multicolinealidad: Aplicación a rendimientos de letras del tesoro. Revista de Métodos Cuantitativos para la Economía y la Empresa, 24, 169-189.

[35] . (2024). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.SelectKBest.html

[36] Goodfellow, I. (2016). Deep learning (Vol. 196). MIT press.

[37] LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. Nature 521, 436–444 (2015). https://doi.org/10.1038/nature14539

[38] Lipton, Z. C., Berkowitz, J., & Elkan, C. (2015). A critical review of recurrent neural networks for sequence learning. arXiv preprint arXiv:1506.00019.

[39] Graves, A. (2012). Supervised Sequence Labelling. In: Supervised Sequence Labelling with Recurrent Neural Networks. Studies in Computational Intelligence, vol 385. Springer, Berlin, Heidelberg. https://doi.org/10.1007/978-3-642-24797-2_2

[40] IBM Developer. (2024). https://developer.ibm.com/articles/cc-machine-learning-deep-learning-architectures/?mhsrc=ibmsearch_a&mhq=LSTM

[41] Olah, C. (2015). Understanding lstm networks.

[42] compute_class_weight. (2025). Scikit-learn. https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html

[43] Becerra-Luna, B., Martínez-Memije, R., Cartas-Rosado, R., Infante-Vázquez, O. (2017). Aumento en la efectividad de la identificación de cimas y pies en el pulso fotopletismográfico al reconstruirlo mediante filtrado adaptativo. Archivos de cardiología de México, 87(1), 61-71.https://doi.org/10.1016/j.acmx.2016.10.005

[44] Pham, T., Tran, T., Phung, D., Venkatesh, S. (2016). DeepCare: A Deep Dynamic Memory Model for Predictive Medicine. In: Bailey, J., Khan, L., Washio, T., Dobbie, G., Huang, J., Wang, R. (eds) Advances in Knowledge Discovery and Data Mining. PAKDD 2016. Lecture Notes in Computer Science(), vol 9652. Springer, Cham. https://doi.org/10.1007/978-3-319-31750-2_3

[45] de Zambotti, M., Goldstein, C., Cook, J., Menghini, L., Altini, M., Cheng, P., & Robillard, R. (2024). State of the science and recommendations for using wearable technology in sleep and circadian research. Sleep, 47(4), zsad325. https://doi.org/10.1093/sleep/zsad325

[46] Berryhill, S., Morton, C. J., Dean, A., Berryhill, A., Provencio-Dean, N., Patel, S. I., Estep, L., Combs, D., Mashaqi, S., Gerald, L. B., Krishnan, J. A., & Parthasarathy, S. (2020). Effect of wearables on sleep in healthy individuals: a randomized crossover trial and validation study. Journal Of Clinical Sleep Medicine, 16(5), 775-783. https://doi.org/10.5664/jcsm.8356

[47] De Zambotti, M., Cellini, N., Goldstone, A., Colrain, I. M., & Baker, F. C. (2019). Wearable Sleep Technology in Clinical and Research Settings. Medicine & Science In Sports & Exercise, 51(7), 1538-1557. https://doi.org/10.1249/mss.0000000000001947

[48] Yildirim, O., Baloglu, U. B., & Acharya, U. R. (2019). A deep learning model for automated sleep stages classification using PSG signals. International journal of environmental research and public health, 16(4), 599.

[49] Tsinalis, O., Matthews, P.M. & Guo, Y. Automatic Sleep Stage Scoring Using Time-Frequency Analysis and Stacked Sparse Autoencoders. Ann Biomed Eng 44, 1587–1597 (2016). https://doi.org/10.1007/s10439-015-1444-y

[50] Urtnasan, E., Joo, E. Y., & Lee, K. H. (2021). AI-Enabled Algorithm for Automatic Classification of Sleep Disorders Based on Single-Lead Electrocardiogram. Diagnostics, 11(11), 2054. https://doi.org/10.3390/diagnostics11112054