Universidad de Oviedo

ESCUELA POLITÉCNICA DE INGENIERÍA DE GIJÓN.

MÁSTER UNIVERSITARIO EN INGENIERÍA INFORMÁTICA

ÁREA DE CIENCIAS DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL

DIAGNÓSTICO DE LA FIBRILACIÓN AURICULAR MEDIANTE EL ANÁLISIS INTELIGENTE DE REGISTROS INTRACARDÍACOS

D. COSTA CORTEZ, Nahuel
TUTOR: D. RANILLA PASTOR, José

FECHA: Enero 2021

AGRADECIMIENTOS

La elaboración de este proyecto me ha otorgado la posibilidad de aprender y disfrutar a partes iguales. Me considero afortunado por ello, así como también por estar rodeado de las personas que me han acompañado durante este trayecto. Luciano, Vani, Mel, María, mamá y sobre todo tu, papá, os agradezco vuestro apoyo, este proyecto también es un poquito vuestro.

RESUMEN

Las enfermedades cardíacas se encuentran entre las principales causas de muerte a nivel mundial. La Fibrilación Auricular es el trastorno cardiovascular más común y su diagnóstico no es sencillo: habitualmente su tratamiento conlleva el uso de marcapasos para regular el ritmo cardiaco y no existe un procedimiento estándar para diagnosticar el estado de la enfermedad más allá de la revisión de los datos capturados por estos dispositivos por parte de los especialistas.

La Inteligencia Artificial está muy presente en nuestra sociedad y está irrumpiendo en campos diversos, incluyendo el diagnóstico de enfermedades. Sin embargo, su presencia en el diagnóstico de trastornos cardíacos es limitada, mientras que el uso de algoritmos eficientes puede apoyar en gran medida a la interpretación de los datos de los pacientes. Este proyecto tiene como objetivo ofrecer una herramienta de apoyo a los especialistas médicos para el diagnóstico de la Fibrilación Auricular a partir de los datos intracardíacos capturados por marcapasos de individuos que padecen la enfermedad. Especialmente, se incide en la predicción de la característica más crítica de la Fibrilación Auricular, el punto de cambio entre arritmias paroxísticas y permanentes, con el objetivo de anticiparse prematuramente a una posible complicación en la situación del paciente.

Se explora el uso de un tipo de red neuronal conocida como Autoencoder Variacional (VAE). Estas redes tienen como objetivo aprender una representación simplificada de los datos de entrada, reduciéndolos a unas menores dimensiones, para después reconstruirlos nuevamente a la dimensión original. Se aprovecha esta capacidad de compresión de datos para crear a partir de arritmias ya diagnosticadas, que reflejan diferentes etapas de la Fibrilación Auricular, un mapa gráfico bidimensional. Con esto se consigue una herramienta que cuando analice datos de pacientes, es decir, sus registros cardíacos, aprenderá a situar su representación comprimida en la zona del mapa que mejor se adecúe a sus características, ofreciendo así una forma sencilla de diagnosticar el estado de la enfermedad del individuo en función de su cercanía a arritmias cuyo diagnóstico es conocido.

PALABRAS CLAVE: Fibrilación Auricular, Registros intracardíacos, Redes neuronales, Autoencoder Variacional.

Contenidos

Lista de figuras

Lista de tablas

Lista de abreviaciones

OMS: Organización Mundial de la Salud

ECG: Electrocardiograma

IDC: Desfibrilador Cardioversor Implantable

CNN: Convolutional Neural Network

DNN: Deep Neural Network

RNN: Recurrent Neural Network

iECG: Electrocardiograma Intracardíaco

VAE: Autoencoder Variacional

GAN: Generative Adversarial Network

FA: Fibrilación Auricular

GRU: Gated recurrent unit

AMS: Automatic Mode Switching

LR: Learning Rate

1. Introducción

1.1.- MOTIVACIÓN

Las enfermedades del corazón suponen un problema que afecta a una parte importante de la población. Según el informe de Patrones de Mortalidad de España (2017) [1] las enfermedades cardiovasculares se sitúan como la segunda causa de muerte más común después del cáncer, y de acuerdo con la Organización Mundial de la salud (OMS) [2], es la principal causa de muerte a nivel global.

Además de ser un problema con alto nivel de incidencia, el incremento de la esperanza de vida en los países desarrollados previsiblemente ocasionará que el alcance de estas enfermedades sea aún mayor y por lo tanto también el número de muertes ocasionadas. Debido a estas razones, la prevención y el tratamiento de estas enfermedades es, a día de hoy, una tarea prioritaria en cualquier sistema sanitario.

Un patrón común entre este tipo de enfermedades son las arritmias. Una arritmia es una alteración de la frecuencia cardíaca, en el que el funcionamiento eléctrico del corazón se ve trastornado y puede aparecer una variación en los latidos hasta el punto en el que se produzcan demasiado lento (bradicardia), demasiado rápido (taquicardia) o de forma irregular.

La prueba más utilizada para analizar la actividad cardíaca en la medicina moderna es el electrocardiograma (ECG). Desde la perspectiva médica se necesita realizar este tipo de pruebas para, a partir de ellas, poder dar un diagnóstico sobre la situación del paciente a tratar.

El ECG es un método en el que se registra y representa gráficamente la actividad eléctrica del corazón en función del tiempo. Es una prueba sencilla y accesible que puede proporcionar información útil sobre la salud del corazón y otras patologías. Habitualmente los ECGs se obtienen superficialmente con un instrumento llamado electrocardiógrafo, el

cual dispone de unos electrodos que miden el potencial eléctrico en las zonas del pecho en los que se colocan. En la Figura 1.1 se puede ver un ejemplo de un ECG sinusal, que representa un ritmo normal. Por el contrario, en la Figura 1.2, en el que se refleja la presencia de una taquicardia, se puede apreciar un ECG en donde las ondas muestran un ritmo irregular.

Figura 1.1.- Ejemplo de ECG sinusal, registrando un ritmo cardíaco regular.

Figura 1.2.- Ejemplo de ECG durante un episodio de Fibrilación Auricular. El registro muestra ondas irregulares entre latidos, el ritmo es anómalo y errático.

Los especialistas médicos realizan una evaluación a partir de los datos recopilados por el ECG para determinar si existe algún riesgo en la situación del paciente que precise tomar alguna medida de prevención, como podría ser la implantación de un marcapasos para regular el pulso cardíaco. Las causas por las que aparecen las arritmias pueden ser muy diversas: mala alimentación, hipertiroidismo, niveles anormales de potasio en el cuerpo…. Además, también existe mucha variedad entre los ritmos cardíacos existentes, por lo que las evaluaciones han de ser precisas.

La implantación de mecanismos como los marcapasos es muy común. Desafortunadamente, portar un dispositivo de este tipo no supone la desaparición de la patología diagnosticada, sino que es un mediador que controla los efectos directos. Sin embargo, la evolución de la

enfermedad puede continuar e implicar un riesgo severo, por eso los pacientes que porten algún dispositivo invasivo suelen someterse a una monitorización periódica de los datos recopilados por el propio dispositivo, así como de otros aspectos como la batería o el estado de los electrodos. Un ejemplo común de lo que podría suceder sin un seguimiento médico continuo sería la progresión de la enfermedad hacia una taquicardia ventricular donde las arritmias producidas en el ventrículo producen una desincronización con las señales producidas en la aurícula y como consecuencia el corazón no puede bombear suficiente sangre al organismo.

Bajo este panorama, la evolución de la tecnología juega un papel importante. Además de la perfección de técnicas de recopilación de datos como los ECGs, actualmente también existen los medios suficientes para proporcionar a los especialistas de cualquier campo, en este caso del ámbito médico, herramientas que les permitan obtener una representación clara de los datos e incluso una interpretación de lo que suponen.

Siguiendo esta línea, el objetivo principal de este trabajo es proveer un soporte a los especialistas médicos en la representación, análisis e interpretación de registros intracardíacos con el fin de facilitar su trabajo y ayudar a que su efectividad se vea incrementada haciendo uso de innovadoras técnicas dentro del campo de la Inteligencia Artificial.

1.2.- PROCESAMIENTO DE REGISTROS CARDÍACOS

La medicina es una disciplina que precisa la evolución en otros campos para su propio desarrollo. A lo largo de la historia los descubrimientos en campos como la física o la química, entre otros, han beneficiado de forma directa a la medicina. De la misma manera, los progresos tecnológicos han supuesto un notable avance para el descubrimiento, tratamiento y curación de enfermedades.

La informática tiene un rol muy importante dentro de la medicina y es fácil encontrar aplicaciones en casi cualquier área de la misma: software de gestión hospitalaria, diagnóstico por imagen, telemedicina… En el caso que aborda este trabajo, el tratamiento de

enfermedades cardiovasculares, los marcapasos tienen un software instalado para su funcionamiento. En los últimos años, también ha ganado relevancia la presencia de los Desfibriladores Cardioversor Implantables (IDCs) (Figura 1.3), que son dispositivos que cumplen una función muy similar a los marcapasos, únicamente que, en lugar de regular el ritmo cardíaco continuamente, solamente actúan enviando una descarga eléctrica al corazón en el momento en el que se produce una arritmia. Recientemente, también han ido apareciendo nuevas maneras de conseguir registros de la actividad cardiaca mediante métodos no invasivos, como por ejemplo, mediante relojes o pulseras [3], [4]. Además, el análisis de los datos recopilados principalmente se apoya en novedosos algoritmos.

Las señales capturadas por los ECGs aportan información que los profesionales utilizan junto a otros datos personales como la edad o el historial clínico para tener un conocimiento más amplio y poder elaborar un diagnóstico en base a ello.

Figura 1.3.- Los IDCs (derecha) son muy similares a los marcapasos (izquierda). Los IDCs actúan únicamente cuando detectan un ritmo inusual para enviar un impulso eléctrico al corazón y así estabilizarlo.

Habitualmente, la forma más común de interpretar un ECG es mediante un examen visual. Los especialistas médicos estudian la morfología de las ondas, que son resultado de los potenciales recogidos durante la estimulación cardíaca. Como estas ondas (Figura 1.4) se repiten entre latidos es relativamente fácil observar la presencia de alguna arritmia, la cual alteraría esta repetición, como se ha visto en la Figura 1.2. En muchos casos el juicio visual es suficiente para determinar si existe algún problema, pero pueden existir alteraciones en las pruebas como un electrodo mal situado o un simple error humano que provoque un diagnóstico erróneo.

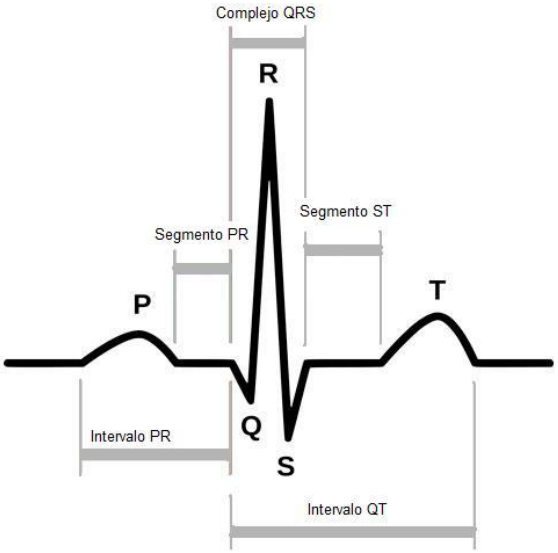


Figura 1.4.- Ejemplo de ondas electrocardiográfica.

En aquellos casos en los que la capacidad de evaluación humana es limitada, la tecnología ha de actuar como soporte. La aplicación más directa es la interpretación de las señales capturadas. Existen contribuciones [5] e incluso paquetes de software [6], [7] que analizan las ondas electrocardiográficas computacionalmente con el fin de obtener una mejor extracción de información y entendimiento de la actividad registrada.

Aparte de las técnicas convencionales, encontramos otros paradigmas que van un paso más allá buscando generar conocimiento a partir de los datos. Se trata de enfoques con soluciones basadas en Inteligencia Artificial, en las que se busca que los algoritmos desarrollados sean capaces de hacer tareas de forma más eficiente que un humano, o en su defecto, que puedan servir como apoyo a los especialistas. En esta rama, la cantidad de publicaciones es significante y se puede apreciar como la evolución en los propios algoritmos ha permitido el desarrollo de nuevas técnicas más eficientes.

Dentro de los algoritmos convencionales de la rama más visible de la Inteligencia Artificial, el Machine Learning, podemos encontrar técnicas aplicadas sobre datos de ECGs para extracción de características [8], [9], reducción de dimensionalidad de los datos [10] o para el análisis de componentes principales [11], [12], [13]. Estas técnicas mencionadas se aplican generalmente en aquellos casos en los que los datos que se pretende analizar tienen varias

dimensiones y se busca encontrar una representación simplificada. También se han empleado algoritmos ampliamente conocidos como SVM [14], K-means [15] o Random Forest [16] para la clasificación del ritmo cardiaco o de diferentes tipos de arritmias.

A pesar de ser trabajos prometedores, los resultados alcanzados no suponen un rendimiento que sea fiable para utilizar en entornos médicos reales. En los últimos años, sin embargo, el gran impacto derivado del avance en la capacidad de computación ha significado un nuevo incentivo para el Machine Learning. Un subcampo dentro de esta rama es el aprendizaje profundo o Deep Learning, que en el último lustro ha tomado un protagonismo muy significativo en la resolución de problemas de índoles muy dispares.

El enfoque principal dentro del Deep Learning son las redes neuronales, un paradigma de programación inspirado en el comportamiento biológico análogo que permite a una computadora aprender a realizar tareas a partir de datos. En la actualidad, su presencia está cada vez más arraigada en nuestro entorno. Sin ir muy lejos, la mayoría de smartphones actuales disponen de asistentes virtuales que reconocen y procesan instrucciones de voz o se pueden desbloquear mediante reconocimiento facial. Estas tareas son fruto de la aplicación de redes neuronales. En numerosos campos están destacando y como no puede ser de otra manera, en la medicina también. Importantes avances se están llevando a cabo, especialmente en sistemas de detección de enfermedades mortales como el cáncer [17], [18], [19] (Figura 1.5) o la Esclerosis Lateral Amiotrófica (ELA) [20], entre otros.

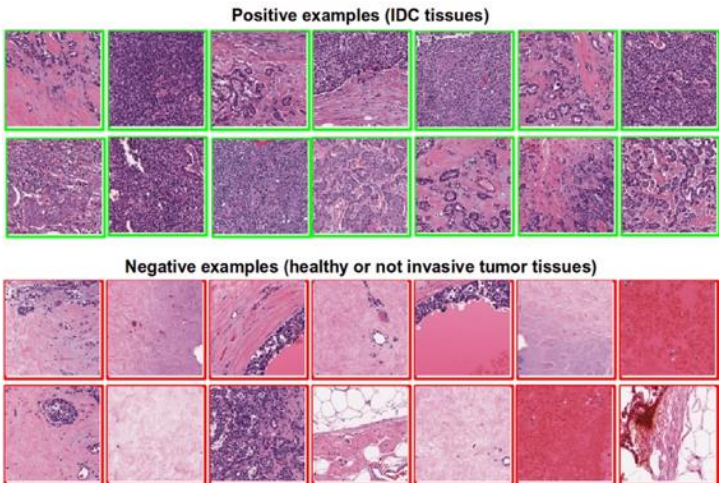


Figura 1.5.- Ejemplo de detección de células cancerígenas para el diagnóstico del cáncer de mama utilizando redes neuronales.

Entre los tipos de redes neuronales más comunes destacan las redes convolucionales o Convolutional Neural Networks (CNN). La importancia de estas redes se debe principalmente a que funcionan muy bien analizando imágenes y, de hecho, en el ámbito médico es común tener disponibles imágenes para analizar. En particular, aunque no se trate de datos asociados a imágenes, para el tratamiento de ECGs superficiales, la amplia mayoría de trabajos de investigación aplican este tipo de redes para tareas de clasificación y predicción [21], [22], [23], [24]. Existen también soluciones basadas en otros métodos como perceptrones multicapa [25], Deep Neural Networks (DNN) [26] o Recurrent Neural Networks (RNN) [27], pero aunque se llegue a conseguir buenos resultados, estas arquitecturas presentan algunas desventajas, como por ejemplo, el número de parámetros, que puede crecer exponencialmente y por consiguiente dificulta la implantación de los modelos desarrollados en dispositivos debido a la excesiva utilización de memoria. O también, es muy probable que se experimente un problema muy conocido en todo tipo de arquitecturas: el "overfitting", que ocurre cuando la red neuronal memoriza las respuestas a partir de los datos de entrenamiento en lugar de estar realmente aprendiendo.

La ventaja que aportan las CNN es la compartición de conexiones y pesos entre las neuronas, lo que permite reducir notablemente el número de parámetros. En la Figura 1.6 se puede apreciar este concepto: en cada capa se obtiene un nivel de complejidad semántica diferente, lo que permite conseguir representaciones de información más simples y de esta manera reducir el número de parámetros y por tanto también la complejidad computacional.

Las CNN principalmente se utilizan para datos en 2 dimensiones (imágenes), sin embargo, es destacable mencionar que en algunos de los trabajos citados se propone reemplazar las convoluciones 2D por convoluciones 1D para analizar las señales capturadas por los ECGs en las que únicamente se tiene en cuenta los datos temporales. En general, el rendimiento que reportan se puede considerar como bueno y en algunas ocasiones muy bueno al ser algoritmos que requieren una carga computacional reducida en inferencia. Esto permite que puedan ser implantadas en dispositivos con capacidad de cómputo limitada como podría ser una Raspberry o mismamente un marcapasos.

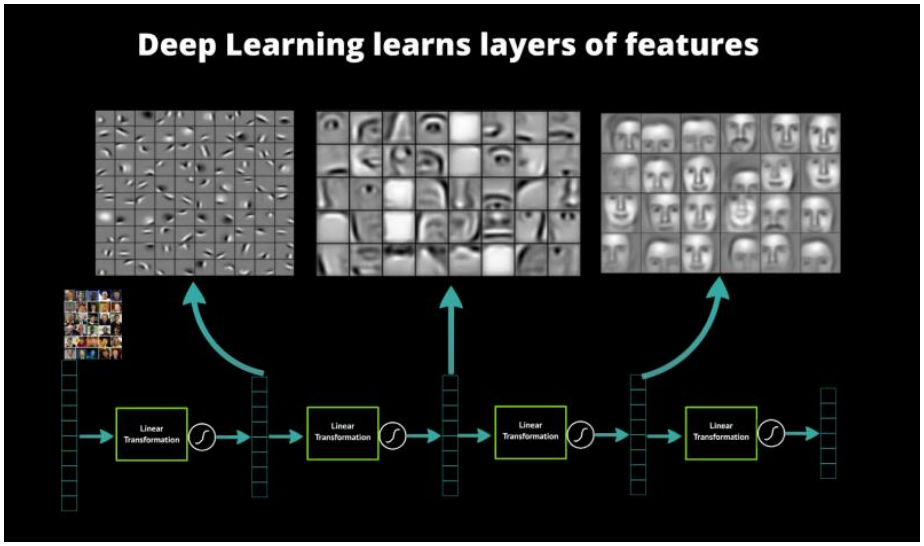


Figura 1.6.- Niveles de representación en distintas capas de una CNN.

Motivación

En este capítulo se ha elaborado una recapitulación de las soluciones existentes respecto al tratamiento de registros cardíacos, esencialmente ECGs. En este subapartado se argumentará las principales limitaciones que tienen estos trabajos para que se puedan tomar como referencia para la solución que se pretende dar en este proyecto.

En primer lugar, los trabajos citados están orientados a registros que se obtienen de forma no invasiva, bien sea mediante un electrocardiógrafo o mediante dispositivos de uso común como un reloj o un teléfono móvil. Con estos métodos, el ritmo cardíaco registrado se realiza en un período de tiempo muy concreto, que en ningún caso es suficientemente amplio como para captar la evolución de una arritmia.

En segundo lugar, estas pruebas podrían servir para enfermedades en etapas muy prematuras, no obstante, lo que se quiere cubrir en este trabajo son fases de la enfermedad más avanzadas y, a día de hoy, los dispositivos que pueden proporcionar información valiosa en los pacientes en estas etapas son los marcapasos.

Este proyecto formula una nueva perspectiva aplicada a un nicho en el que, a pesar de tener gran importancia, la investigación es bastante limitada. A diferencia de las soluciones citadas y que existen en la actualidad, el enfoque principal propuesto está orientado al análisis de

registros intracardíacos, es decir, registros capturados por dispositivos invasivos (implantados en el paciente), esencialmente marcapasos o IDCs.

Como se reporta en líneas anteriores, aunque los ECGs superficiales aportan información muy útil, no capturan la cantidad de información necesaria como para elaborar un diagnóstico prematuro de la evolución de la enfermedad. Esta necesidad se palia con los dispositivos implantables, ya que incluyen registros que reflejan exactamente los periodos de tiempo en los que un paciente ha sufrido un episodio de arritmia, la duración de la misma, así como el Electrocardiograma Intracardíaco (iECG) resultante. Con esta información se puede elaborar un registro con el historial de episodios de cada paciente.

Sería lícito pensar que si se dispone de un ECG intracardíaco, también sería posible aplicar alguna de las soluciones comentadas anteriormente a estos datos, sin embargo, esto no es posible porque a diferencia de los ECGs superficiales, los iECGs sólo incluyen las frecuencias instantáneas de la aurícula y el ventrículo porque la morfología de los latidos del corazón se pierde en el filtro de paso alto en el electrodo del dispositivo (Figura 1.7).

Un pilar importante de este trabajo es el tratamiento de la Fibrilación Auricular (FA). La FA es la arritmia cardíaca más común en la práctica clínica y habitualmente su tratamiento incluye la implantación de marcapasos o IDCs. La enfermedad evoluciona normalmente desde arritmias paroxísticas (arritmias que aparecen y desaparecen espontáneamente) a arritmias persistentes (episodios que duran más de 7 días y no terminan sin intervención externa, por ejemplo, con ayuda de medicamentos) o hacia arritmias permanentes (episodios ininterrumpidos). Más adelante se explicará en los objetivos del trabajo cómo se pretende ofrecer una solución para el tratamiento de este problema.

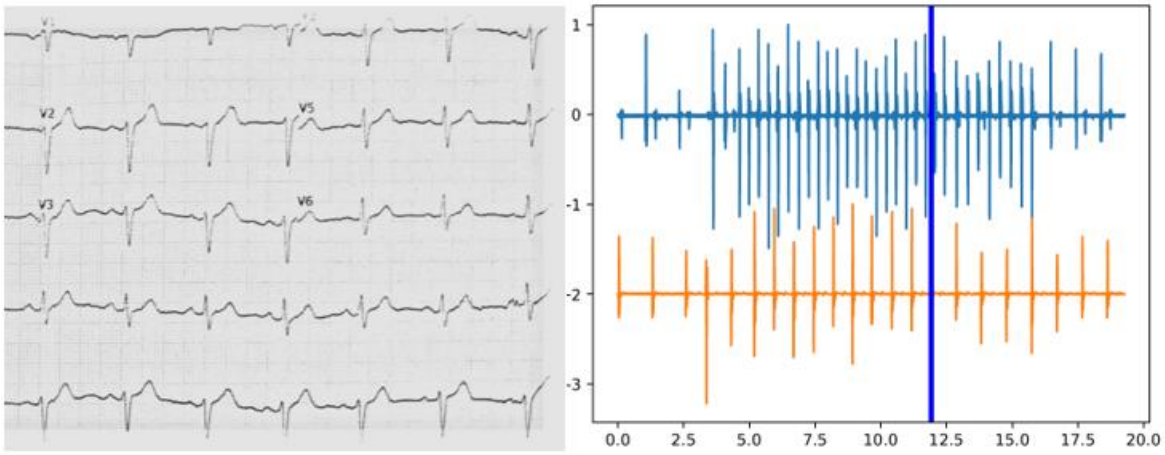


Figura 1.7.- La morfología del ECG superficial (izquierda) no se mantiene en el ECG intracardiaco (derecha), donde solo hay un pico por cada latido.

Teniendo en cuenta lo explicado en esta sección, el propósito de este proyecto no está orientado a la clasificación o detección de arritmias en un instante de tiempo determinado (como en los trabajos comentados), sino en poder proporcionar, a partir de los datos intracardíacos, un diagnóstico de la situación del paciente que posibilite mostrar cómo va a progresar la enfermedad en el futuro cercano con el fin de alertar a los especialistas médicos para que puedan prevenir cualquier complicación futura y así mejorar la salud, y por tanto, la calidad de vida del paciente.

La solución propuesta está totalmente enfocada a la aplicación de la Inteligencia Artificial a la medicina, concretamente mediante la utilización de redes neuronales. Aparte de la especial mención realizada a las CNN, es importante conocer que existen también otros tipos de redes que están irrumpiendo en otros campos de la ciencia. Como base para abordar el problema se propone una reciente arquitectura conocida como Autoencoder Variacional (VAE), que será explicada en detalle en el siguiente capítulo.

1.3.- MODELOS GENERATIVOS - VAE

VAE son las siglas de Variational Autoencoder (Autoencoder Variacional). Antes de explicar directamente qué son este tipo de redes es necesario conocer algunos antecedentes. Los autoencoders son una familia de redes neuronales que tiene como objetivo aprender una representación simplificada de los datos de entrada. Habitualmente se utilizan para datos que

tienen varias dimensiones, puesto que es útil para tener una perspectiva global que permita entender con qué datos se está trabajando.

La dimensionalidad de los datos es un factor muy importante a tener en cuenta. El número de dimensiones se puede entender como el número de variables o características que se utilizan para resolver un problema. Por ejemplo, en un problema en el que se quiera deducir si un animal es o no mamífero, se dispone por cada animal información sobre la altura, el peso, el número de patas y el tipo de pelo. En este caso serían 4 características que influirán a la hora de determinar la solución, es decir, 4 dimensiones.

El preprocesado de datos es uno de los pasos más importantes para resolver cualquier problema de Machine Learning y la reducción de dimensionalidad es una de las técnicas de preprocesado más comunes. Reducir la dimensionalidad se puede entender como "deshacerse" de información, puesto que es posible que se disponga de información que sea irrelevante para el aprendizaje del modelo o quizá se valore más el rendimiento computacional (menor cuantas menos variables se tengan) o simplemente para reducir complejidad y así facilitar la comprensión de los resultados del modelo.

Existen multitud de técnicas para realizar esta tarea, entre las cuáles merece la pena mencionar el Análisis de Componentes Principales (PCA) [28], debido al impacto que ha supuesto en el desarrollo de algoritmos posteriores como ICA, LLE o LDA.

Retomando la relevancia de los autoencoders, su funcionamiento se basa en la reconstrucción de los datos de entrada utilizando un modelo de red neuronal con un cuello de botella en el medio que captura una codificación comprimida de los datos. Esta representación de baja dimensión puede utilizarse para varias aplicaciones como la comprensión de los datos, crear nuevos modelos a partir de la representación aprendida o conocer las dependencias entre los ejemplos del conjunto de datos.

Autoencoder

Los autoencoders son una familia de redes neuronales que pertenecen al paradigma de aprendizaje no supervisado. En este paradigma la función objetivo no persigue ajustar pares entrada-salida (como sí lo hace el aprendizaje supervisado), sino que se busca aumentar el conocimiento estructural de los datos disponibles. La primera red que forma parte de esta familia posee el mismo nombre, Autoencoder. El Autoencoder [29] es una red diseñada para aprender una función que reconstruye la entrada original mientras que a la vez se comprimen los datos durante el proceso para descubrir una representación más eficiente para el entendimiento humano. Se compone de dos redes:

- Red codificadora o encoder: Se encarga de comprimir datos de alta dimensión a un espacio de baja dimensión, también conocido como espacio latente. Esta compresión de los datos se denomina representación latente.
- Red decodificadora o decoder: A partir de la representación latente aprendida por el encoder reconstruye los datos de entrada.

En la Figura 1.8 se ilustra un esquema de la arquitectura.

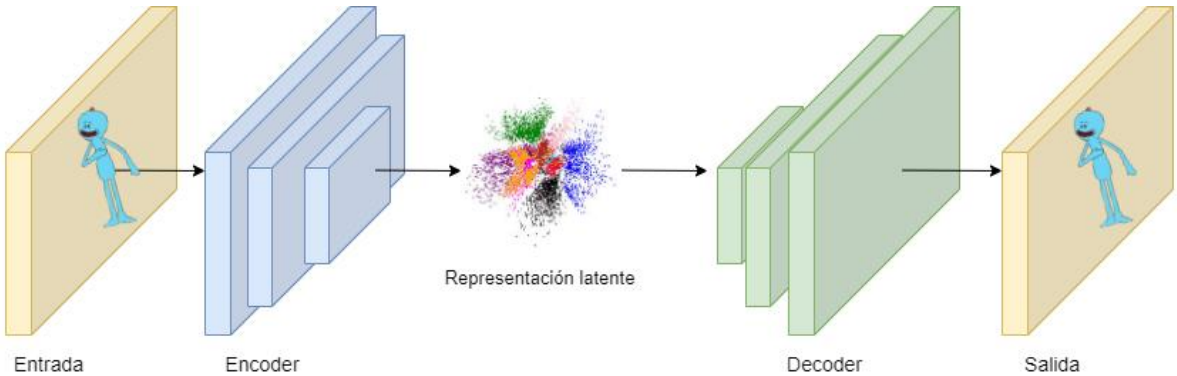


Figura 1.8.- Arquitectura de un autoencoder.

Como se puede apreciar, el encoder cumple con la reducción de dimensionalidad citada y además el autoencoder en su conjunto está explícitamente optimizado para la reconstrucción de los datos. Más adelante se verá para qué puede ser interesante aprovechar esta propiedad.

Con el paso de los años la idea del Autoencoder se ha ido asentando y se han ido proponiendo mejoras sobre el trabajo original. Una de las limitaciones más directas es el riesgo de overfitting cuando la red tiene más parámetros que el número de ejemplos del conjunto de datos. Para evitar esto, Denoising Autoencoder [30] propone una modificación en el que la entrada está parcialmente corrompida al enmascarar algunos valores del vector de entrada de manera estocástica. En Sparse Autoencoder [31] se aplica una restricción en la activación de algunas neuronas para evitar también el sobreajuste y mejorar la robustez. Otras contribuciones han ido apareciendo hasta la llegada del VAE [32].

El VAE supone una renovación del concepto de autoencoder. Es el método menos similar a los modelos anteriores, principalmente porque se basa en los conceptos de la inferencia bayesiana. Los autoencoders estándar aprenden a generar representaciones compactas y a reconstruir bien los datos de entrada, pero las aplicaciones son bastante limitadas. El principal problema es que el espacio latente al que convierten los datos no es un espacio continuo, es decir, traducen las entradas a un conjunto de vectores fijo y esto ocasiona que el espacio latente tenga discontinuidades. Un espacio latente discontinuo es apto para la reconstrucción de los datos de entrada, sin embargo, si se reconstruye una entrada a partir de una zona del espacio latente donde no existe representación alguna, la salida será poco realista, porque el decodificador no sabe cómo reconstruir una entrada a partir de una región del espacio latente vacía, durante el entrenamiento nunca vio vectores codificados procedentes de esa región.

Los VAE tienen una propiedad que los hace únicos respecto al resto de autoencoders: los espacios latentes son continuos. Esta propiedad permite al decodificador reconstruir datos a partir de una zona del espacio latente que no pertenezca a la compresión de ningún dato de entrada sino a una interpolación, generando por tanto un nuevo dato nunca antes visto. En la Figura 1.9 se puede ver un ejemplo de esto: la parte de la izquierda se corresponde con la representación latente de los datos de entrada, que en este caso son figuras geométricas, de un autoencoder convencional. Hay datos que no son similares que se codifican en zonas cercanas del espacio latente, por lo que si se elige algún punto intermedio, por ejemplo, entre el rectángulo y el triángulo redondeado, la decodificación resultante no tiene sentido (figura en rojo), en cambio en la parte de la derecha se ve que datos similares se codifican en zonas

cercanas, por tanto, si se interpola de nuevo entre el rectángulo y el triángulo redondeado, la codificación resultante será un rectángulo redondeado.

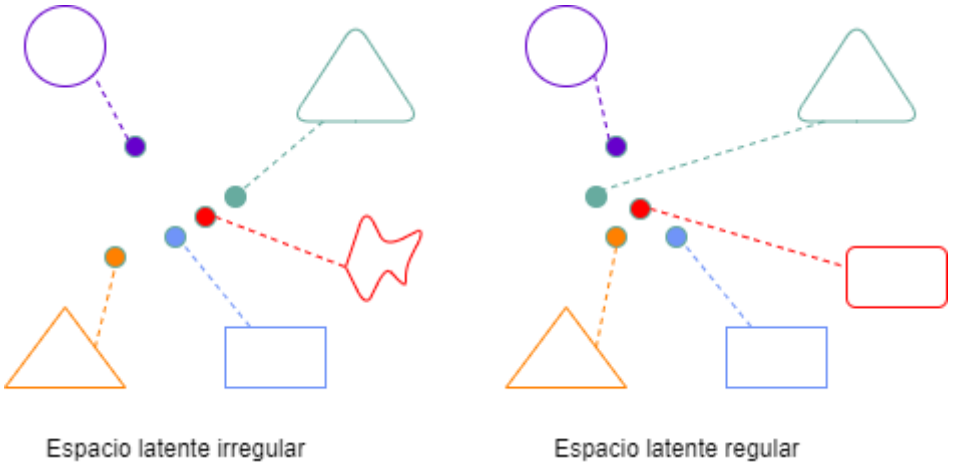


Figura 1.9.- Espacio latente de un autoencoder convencional (izquierda) vs un VAE (derecha).

Se puede establecer una analogía entre el funcionamiento del VAE y nuestra imaginación. Supongamos que queremos inventarnos un animal. Sabemos que tendrá 4 patas, respirará oxígeno, posiblemente tendrá pelo… Estas características se pueden entender como un vector de variables latentes que nuestra imaginación ha creado a partir de los animales que hemos conocido a lo largo de nuestra vida. Así, los animales que conocemos serían los datos de entrada, nuestra imaginación sería análoga al espacio latente producto del encoder, y el animal resultante generado por nuestra imaginación sería una posible salida del decoder, o si simplemente estamos recordando un animal, esto correspondería a una reconstrucción de los datos de entrada. Sin un espacio latente continuo, como ocurre en los autoencoders convencionales, la generación de los datos se haría a ciegas, es decir, como si no fuéramos capaces de imaginar animales que conocemos, lo que ocasionaría un resultado que podría no tener nada que ver con un animal.

La continuidad en el espacio latente se logra haciendo que el encoder no produzca un vector de tamaño fijo, sino dos vectores: un vector de medias μ y otro vector de desviaciones estándar σ, que definen una distribución de probabilidad. Al aprender una distribución, la reconstrucción de los datos se hace a partir de una muestra del espacio latente que siga esta distribución. Esto permite que el VAE sea un modelo generativo.

Los modelos generativos en los últimos años han supuesto un impacto extraordinario, en gran parte gracias a sus dos máximos exponentes: Los VAEs y las redes Generativas Adversarias (GAN). Las redes GAN [33] son otras de las redes neuronales que se utilizan para motivos generativos (Figura 1.10) y su aplicación hoy en día está ampliamente extendida. Este tipo de modelos tiene una importancia muy valiosa: la generación de datos. Son redes que aprenden la distribución de los datos con los que se entrenan y que pueden producir datos pertenecientes a esa distribución, pero diferentes a los ejemplos del conjunto de entrenamiento.

La posibilidad de generar nuevos datos da pie a muchas oportunidades. Concretamente en el mundo del Machine Learning una de las tareas más dificultosas es la obtención de un conjunto de datos que sea lo suficientemente grande y variado como para entrenar a un modelo eficientemente. Con esta nueva posibilidad esta insuficiencia se podría cubrir.

Figura 1.10.- Una de las aplicaciones más características de las redes GAN es la generación de caras realistas (imágenes generadas por la página https://thispersondoesnotexist.com).

Más allá de contribuir al aprendizaje de modelos, los VAEs son muy útiles para otros campos. Las aplicaciones de los autoencoders aumentan notablemente con las propiedades generativas que ofrecen los VAEs y previsiblemente seguirán aumentando a la par que se produzcan mejoras en las técnicas actuales. Las aplicaciones son extensibles a sectores de

diferentes índoles, aunque con un fuerte componente artístico en común. En la edición de imágenes se utilizan para la generación de caras humanas realistas [34] y transiciones de un dominio a otro [35], [36] (Figura 1.11). También se han utilizado para detección de anomalías [37], [38] o para generación de música y audio [39], [40], [41] en donde destaca WaveNet, de la empresa DeepMind (subempresa de Google), una herramienta para crear muestras de audio sintéticas o MuseNet, de OpenAI, que puede generar composiciones musicales de hasta 4 minutos con 10 instrumentos diferentes y también combinar diferentes estilos musicales.

Figura 1.11.- Traslación de imágenes de un dominio a otro, en este caso de la imagen original a la misma con diferentes condiciones atmosféricas y de luz.

Limitaciones

A pesar de que este tipo de modelos sea prometedor y haya demostrado funcionar eficientemente, tiene sus limitaciones, la más directa es el fuerte componte matemático. Aunque se puede entender intuitivamente, como se ha explicado en esta sección, con vocabulario básico y sin entrar un mucho detalle, los conceptos matemáticos que residen detrás del VAE son una de las barreras que impiden que sea accesible a un número mayor de personas. En contrapartida, no son el único modelo generativo, y las redes GAN han tomado la delantera, al menos en cuanto a mayor cantidad de contribuciones científicas, precisamente por la simplicidad en la complejidad teórica que rige estas redes, además de la alta calidad que demuestran cuando se trata de problemas con imágenes.

A su favor, en los VAEs existe una forma clara de evaluar la calidad de un modelo, es decir, hay una métrica que puede ser comparada entre soluciones. Este es uno de los grandes lastres de las redes GAN porque actualmente no hay un consenso en cuanto a cómo evaluar este tipo de redes. Además, las GAN tienden a ser mucho más pesadas de entrenar.

De cualquier manera, la utilización de cualquiera de estos modelos para fines generativos es una buena opción y depende mucho del tipo de problema.

VAEs en el ámbito médico

Las aplicaciones mencionadas son las más populares, sin embargo, existen campos en donde, a pesar de no tener tanta repercusión, estas redes tienen bastante protagonismo. En el ámbito médico se está empezando a asentar su aplicación, especialmente en temas relacionados con imágenes. Es importante destacar que los datos médicos tienen una sensibilidad característica porque son datos que pertenecen a personas. Este hecho provoca que la obtención de conjuntos de datos para construir modelos sea una tarea tediosa y en muchas ocasiones imposible debido a temas de privacidad.

La posibilidad que brindan los VAEs y en general los modelos generativos, implica un gran paso para la consecución de datos de carácter sensible como los datos médicos, a diferencia de otros que se pueden encontrar fácilmente como puede ser imágenes de coches, por ejemplo. En esta dirección existen contribuciones como [42] que presentan ambos modelos, GAN y VAE para la generación de datos médicos realistas.

Sin embargo, donde existe mayor interés respecto a la aplicación de VAEs en la medicina es para la detección de anomalías [43], [44] y para el aprendizaje de la naturaleza de los datos [45]. Como los VAEs tienen la capacidad de aprender la distribución de los datos con los que se entrenan, se pueden crear modelos capaces de detectar datos que no siguen específicamente la distribución aprendida, esto es, la detección de datos anómalos. Siguiendo la misma premisa, existen datos que, dispersos pueden ser difíciles de interpretar, pero conocer su representación latente puede ayudar a entender su naturaleza, y esto se alinea en gran parte con lo que se conoce como Representation Learning.

El rendimiento de los modelos de Machine Learning depende en gran medida de las representaciones aprendidas. Típicamente se requiere un algoritmo capaz de aprender las características que mejor representan la distribución subyacente de los datos, lo que facilita la realización de otras tareas como la clasificación o la predicción. Esto es lo que se conoce como Representation Learning. Este paradigma se ha empleado en varias áreas de la medicina con fines tales como la selección de factores de riesgo, el fenotipado de enfermedades y la predicción o clasificación de riesgos de enfermedades [46].

Esta línea de investigación es clave para desarrollar lo que se conoce como Explanable AI o IA explicativa, en donde los resultados de los algoritmos puedan ser interpretados por expertos humanos. Existe una falta de trabajo en esta dirección en lo que se refiere al diagnóstico de enfermedades cardiovasculares, donde los pocos trabajos que existen se centran en el procesamiento de imágenes [47] o se desarrollan simples clasificadores para datos de series temporales [48]. En el caso de la FA, la gran mayoría de contribuciones analizan datos de ECGs superficiales [23], o en otros casos como en [49], los autores estudian datos de dispositivos de muñeca con redes convolucionales, sin embargo, en cualquier caso, los datos provienen de dispositivos no invasivos, compatibles con pacientes en etapas tempranas de la enfermedad o sin patologías previas, por lo que están fuera del alcance de este trabajo. Los marcapasos siguen siendo los dispositivos que pueden proporcionar información valiosa en aquellos pacientes en etapas más avanzadas de la enfermedad.

Deficiencias en las soluciones existentes

Analizado el estado del arte en cuanto a las soluciones que se podrían llegar a considerar por su similitud con la temática de este proyecto, se concluye que en ninguno de los casos vistos se podría adaptar a nuestro problema. Una de las principales barreras de estas contribuciones es el amplio interés en el tratamiento de imágenes. Los datos con lo que se trabajará, los registros intracardíacos, conforman una sucesión de datos medidos cronológicamente, lo que se puede identificar como serie temporal y en cuanto a los trabajos citados únicamente en [48] se analiza este tipo de datos.

Si se establece el foco sobre la aplicación de VAEs para series temporales encontramos que en [50] los autores presentan un modelo de VAE que puede mapear datos de series temporales a una representación vectorial latente, pero el modelo ha quedado obsoleto debido a los avances recientes en arquitecturas recurrentes. Últimamente, han comenzado a surgir trabajos prometedores: para modelar la complexión temporal de los datos, en [51] se utilizan redes LSTM y en [52] se propone usar redes de ecoestado, ambas como componentes principales de un VAE. A pesar de que estos trabajos combinan arquitecturas recurrentes con VAEs, su objetivo difiere del nuestro ya que están orientadas a la detección de anomalías y poco tienen que ver con el ámbito médico.

Llegados a este punto, es evidente que existe un vacío en cuanto a la existencia de soluciones para tratar electrocardiogramas intracardíacos con VAEs, por eso la realización de este trabajo supone un reto adicional, pero que es necesario para estudiar su potencial en un problema tan importante como el que se aborda.

1.4.- OBJETIVOS DEL TRABAJO

La FA es el tipo más común de arritmia. Es causada por un problema en el sistema eléctrico del corazón que provoca un latido irregular en el que las cavidades superiores del corazón, las aurículas, fibrilan (como si temblaran). Como consecuencia de la pérdida de sincronía en el ritmo cardíaco es posible que aparezcan síntomas como dolores de pecho o mareos y en los peores escenarios se puede ocasionar la formación de coágulos en el interior del corazón debido a que la circulación sanguínea hacia los ventrículos se entorpece porque el vaciado de sangre en las aurículas puede no ser total. El riesgo de aparición de estos coágulos es muy alto porque si alcanzan el torrente sanguíneo pueden provocar graves problemas como la obstrucción de arterias en el cerebro (ictus cerebral).

El tratamiento más común para etapas más avanzadas de la enfermedad es la implantación de un marcapasos o IDC que controle la frecuencia cardíaca del paciente. Estos dispositivos monitorizan la actividad cardíaca de tal manera que son capaces de detectar cuando se produce un latido atípico, especialmente de alta frecuencia auricular, que corresponde habitualmente con un episodio de FA. Esta monitorización permite obtener datos de interés

que pueden ser utilizados para ofrecer un diagnostico prematuro, sin embargo, hay una serie de factores que dificultan esta tarea.

Hay que tener en cuenta que los registros que almacenan los marcapasos son limitados, de otra manera, si se tuviera un amplio historial de registros significaría que esa persona ya estaría en la fase final de la enfermedad. Esta limitación en los datos causa que sea dificultoso obtener un modelo que se ajuste a las características de estos para tener un conocimiento de estado del paciente. Además, el hecho de que la enfermedad evolucione con el paso del tiempo provoca que los datos no sean estacionarios y es precisamente el cambio en las propiedades de los datos lo que determina la progresión de la enfermedad, (paso de arritmias paroxísticas a permanentes), y por tanto es objeto de estudio.

Existen aún más dificultades, el algoritmo que utilizan los marcapasos para determinar la duración de los episodios de arritmia no es completamente fiable. Los parámetros del dispositivo se ajustan priorizando la seguridad, por lo que la tasa de falsos positivos es alta. Esto da lugar a largos episodios de FA que a veces se notifican erróneamente como conjuntos de episodios cortos, por lo que se necesita un preprocesamiento para tener en cuenta estos eventos espurios, que a su vez hace que el número de episodios disponibles se reduzca aún más.

Por otra parte, entre la información almacenada están los iECGs, que son representaciones de la diferencia de potencial entre dos puntos en contacto con el miocardio a lo largo del tiempo. Mantienen un registro de información segundos antes y después de la detección de cada episodio. Esto incluye sólo las frecuencias instantáneas de la aurícula y el ventrículo porque la morfología del latido del corazón se pierde en el filtro de paso alto, como se comentó previamente. Esta información no se utiliza para el diagnóstico, sino para ajustar los parámetros de funcionamiento del dispositivo. Conociendo esto, la fuente de información más fiable con la que trabajar son las fechas y las duraciones de los episodios registrados.

Expuestos estos hechos, es evidente que la progresión de la FA es un proceso complejo que depende de muchos factores diferentes. Lo ideal sería encontrar un modelo que fuera capaz de aprender, a partir de unas pocas docenas de registros, las propiedades específicas de la enfermedad y al mismo tiempo pudiera extrapolar el punto de cambio entre FA paroxística

y permanente. Desafortunadamente, tomar una aproximación de estas singularidades es una tarea prácticamente inviable.

En este proyecto se decide tomar un enfoque diferente. Los problemas explicados se toman como punto de partida para elaborar una solución que busca contribuir en la consecución de los objetivos citados. La meta principal que se busca conseguir es proveer un sistema capaz de detectar distintos tipos de arritmias, dotando al personal clínico de una herramienta objetiva que permita evaluar de forma precisa el riesgo en la evolución de un paciente con FA.

Uno de los obstáculos principales mencionados es la insuficiente cantidad de datos para elaborar un modelo que se adapte a sus características. Para mitigar esta primera barrera se buscará desarrollar un modelo capaz de capturar las propiedades distintivas de la progresión de la FA. Dicho de otra manera, se persigue la obtención de un modelo que simule el comportamiento real de la enfermedad para, a partir de él, generar episodios de arritmia simulados con diferentes características. Este modelo de simulación permitirá conseguir un conjunto de entrenamiento lo suficientemente grande para entrenar un VAE. Aprovechando el potencial de este tipo de redes, si se entrena a un VAE a partir de arritmias con características variadas, al final se tendrá un modelo capaz de proyectar en zonas diferentes del espacio latente los diferentes tipos de arritmia que se le pasen. Al mismo tiempo, esta representación en el espacio latente puede servir como punto de partida para construir un clasificador que cuantitativamente dicte a qué grupo pertenece una arritmia dada (Figura 1.12).

La clave para que esta idea sea exitosa está en la localización de cada arritmia. Se espera que grupos de arritmia similares se localicen en un mismo grupo, pero hay que tener en consideración que el factor variacional para propósitos generativos del autoencoder tenderá a agrupar todos los puntos en zonas cercanas, por tanto, existe la posibilidad que arritmias de diferente tipo se solapen. Como se verá más adelante, el entrenamiento del clasificador incide directamente en el espacio latente resultante.

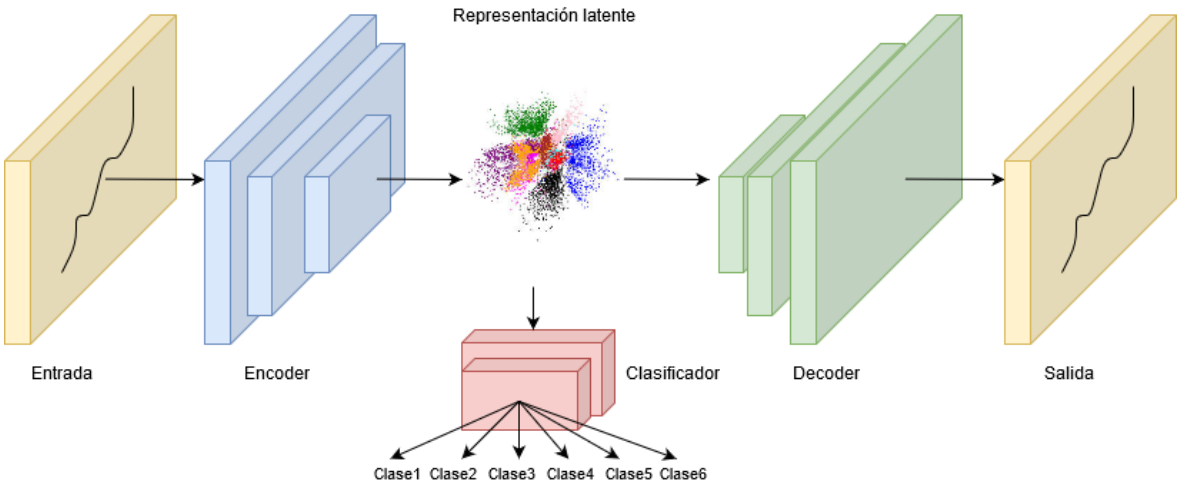


Figura 1.12. Estructura general de la solución.

El modelo simulador utiliza una serie de parámetros para caracterizar los distintos tipos de arritmia, así que la salida del encoder (el espacio latente), ofrecerá una estimación de los parámetros del modelo. Supongamos, por ejemplo, que se decide simular arritmias que suceden cada seis meses y cuya duración es de media hora, pero que en los próximos años evolucionarán a arritmias que ocurren cada mes y cuya duración es de cuatro horas. E l conjunto de datos que cumple con estas características se proyectará en una zona muy concreta del espacio latente formando un cluster. Se espera que la proyección para un caso real con las mismas características se sitúe también en ese cluster.

Se propone así, que el espacio latente resultante tras entrenar al VAE constituya un mapa (Figura 1.13) que sirva como guía para interpretar las características principales que tendrán las arritmias de cada paciente bajo estudio. Cuando se pase al modelo el caso de una persona real, se espera que la proyección del caso sobre el espacio latente caiga sobre un grupo concreto que corresponderá a aquellas arritmias del conjunto de entrenamiento con propiedades similares. De esta manera, la salida del mapa puede considerarse como una proyección de los parámetros del modelo que mejor se adapten a la criticidad del individuo, pudiendo así conocer su estado de salud.

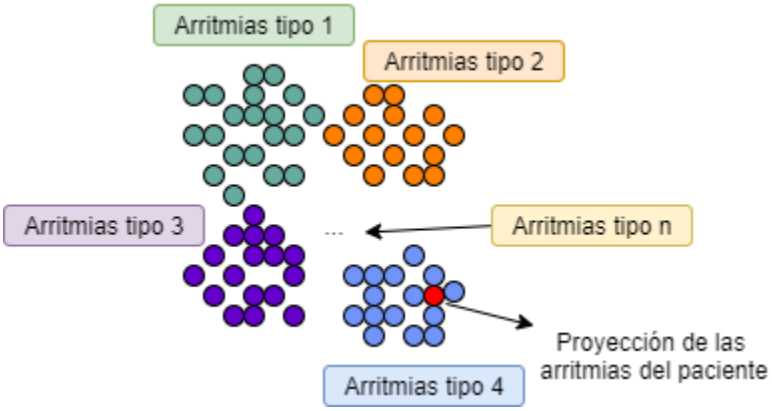


Figura 1.13.- En el espacio latente se agruparán en distintos conjuntos los distintos tipos de arritmia con los que se entrene, permitiendo conocer a qué grupo pertenecen las arritmias de un paciente (punto rojo), y por tanto los parámetros que mejor describen su situación.

Identificados los objetivos del trabajo, se pueden recoger de la siguiente manera:

- Construir un modelo capaz de simular el comportamiento real de los episodios de FA.
- Entrenar a un VAE utilizando como conjunto de datos distintos tipos de arritmia generados a partir del modelo del anterior punto.
- Entrenar a un clasificador para que, a partir de la proyección latente resultante, aprenda a dictaminar a qué grupo de arritmias pertenece la arritmia bajo estudio.
- Proyectar en el espacio latente la localización del caso de estudio para dar una idea específica de la situación del paciente.

Los dos últimos puntos, aunque de diferente manera, ofrecen el diagnóstico final que se busca. El clasificador dirá directamente a qué grupo de arritmias con las que se entrenó se parecen más las del sujeto y en el otro caso, la situación en el mapa es aún más informativa: a pesar de caer en un cluster concreto, la cercanía hacia un cluster u otro ofrecerá una visión de cómo está evolucionando la enfermedad en el paciente.

1.5.- ESTRUCTURA DEL PROYECTO

El proyecto está dividido en 5 secciones cuyo contenido se presenta acorde con la secuencia de objetivos expuestos previamente. Para mayor claridad, al principio de cada sección se mostrará un diagrama de flujo como el siguiente indicando qué tareas se abordarán para conseguir los objetivos propuestos.

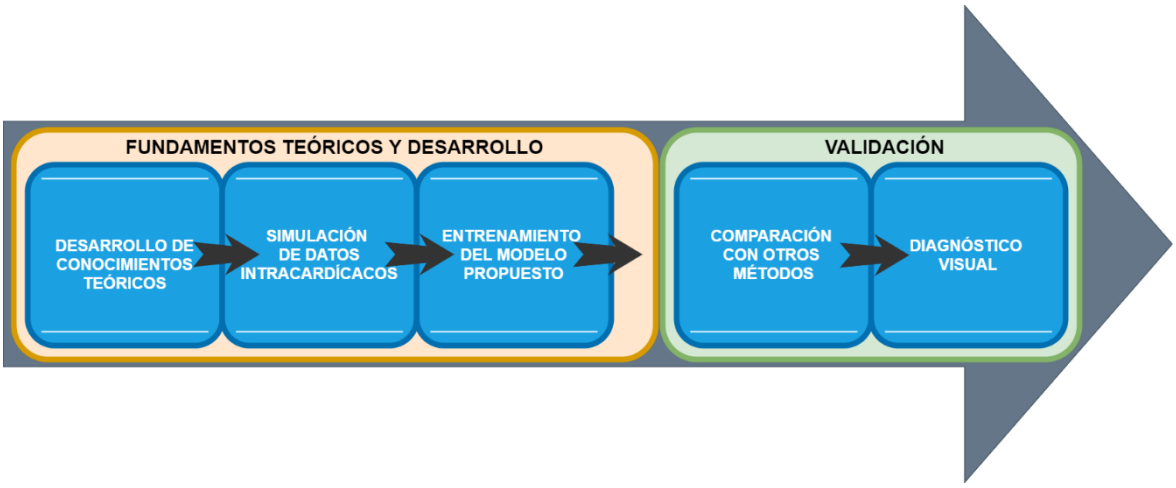
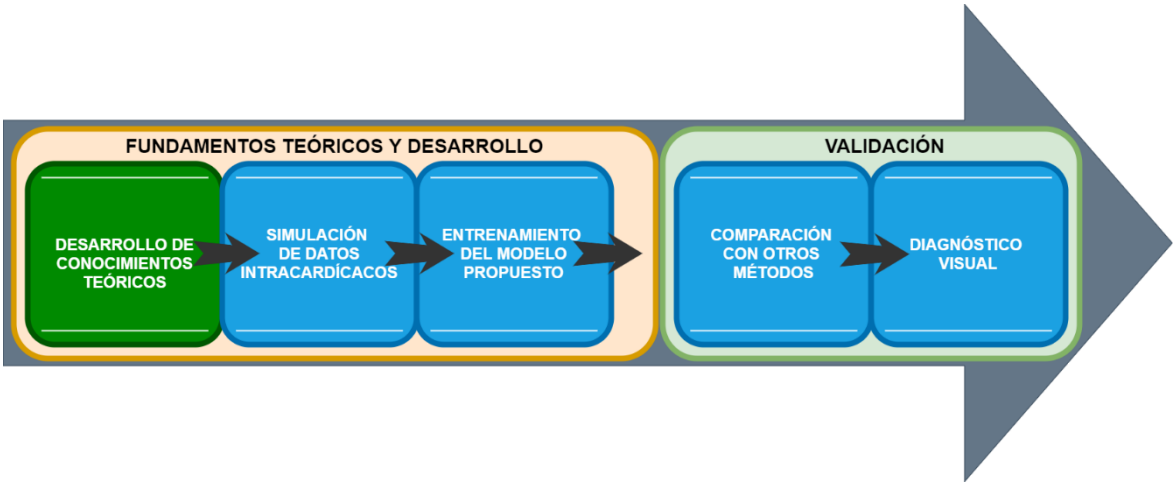


Figura 1.14.- Perspectiva general de los objetivos seguidos para la consecución del proyecto.

2. Desarrollo técnico

En este capítulo se expondrán conocimientos teóricos que permitirán entender conceptos técnicos que aparecen en el desarrollo del proyecto.



2.1.- FUNDAMENTOS TEÓRICOS

2.1.1.- Tratamiento de series temporales

Cuando se habla de redes neuronales didácticamente lo habitual es hacerlo sobre Feedforward Neural Networks. Son redes formadas por capas que están conectadas entre sí y que contienen neuronas (Figura 2.1). A cada neurona la llegan conexiones de neuronas anteriores y a la vez establece conexiones con neuronas posteriores.

Para elegir el tipo de red neuronal adecuado lo primero que hay que plantearse es qué tipo de dato se está tratando. Por ejemplo, las CNN funcionan bien para imágenes o vídeos y siguen el patrón de las Feedforward Neural Networks, teniendo conexiones hacia adelante, sin embargo, para los datos intracardíacos la aplicación de este patrón quizá no es la aproximación más adecuada.

Los datos disponibles de la actividad del corazón son datos que se almacenan en diferentes períodos de tiempo constituyendo un histórico, por lo que se identifican con un tipo de dato conocido como series temporales, es decir, secuencias de datos a lo largo del tiempo.

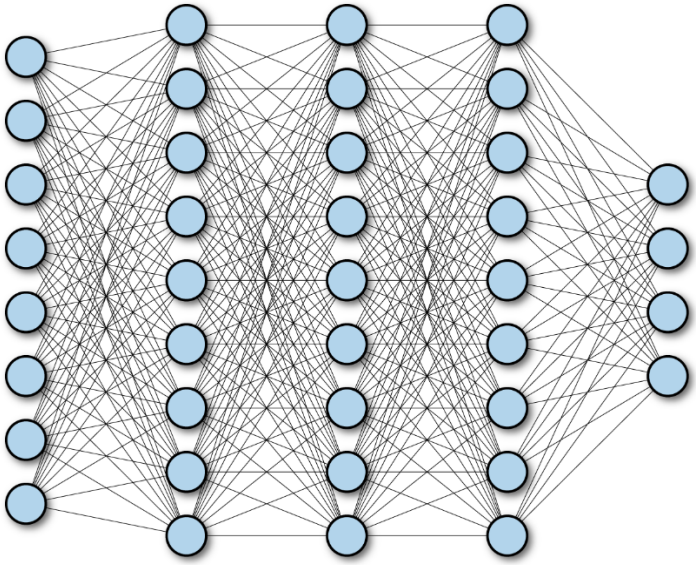


Figura 2.1.- Estructura interna básica de una red neuronal.

La presencia de series temporales es muy frecuente en casi cualquier ámbito, y cada vez su utilización está más extendida, por ejemplo, en campos como el financiero para la predicción de mercados de valores. Es por esto que existen multitud de técnicas para aplicar a problemas con este tipo de dato. Este proyecto se enfocará en el uso de redes neuronales para su tratamiento.

Retomando la importancia de elegir una arquitectura de red neuronal adecuada, hay que saber que en el análisis de series temporales es relevante conocer datos del pasado para comprender el presente y predecir el futuro. Este hecho hace que se centre la atención hacia un tipo de redes precisamente dedicadas a abordar este proceso, las Recurrent Neural Networks (RNN).

A diferencia de las Feedforward Neural Networks, las RNN tienen conexiones hacia delante, pero también hacia atrás. De esta manera se puede modelar el comportamiento dinámico de las secuencias, es decir, se puede calcular la salida de una neurona en base a lo que recibió en el pasado. Así, se tiene una estructura de red donde las salidas (o estados) de las neuronas

dependen no solo de la entrada que reciben, sino también de los estados que tuvieron previamente (Figura *2.2*).

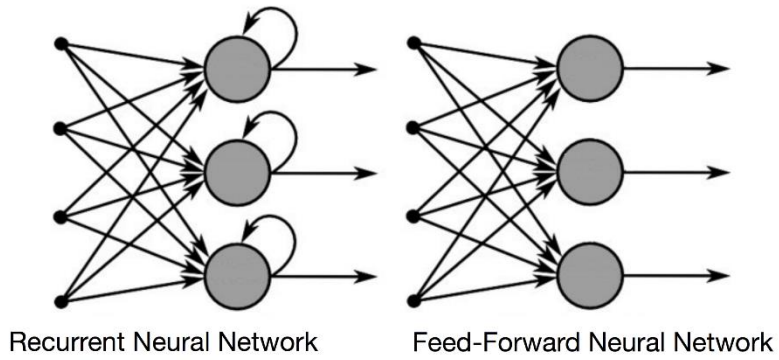


Figura 2.2.- Las neuronas reciben como entradas las salidas de estados de tiempo previos.

Llegados a este punto, se decide que el sistema de reconocimiento de diferentes tipos de arritmias, el objetivo final de este proyecto, estará constituido por un VAE que hará uso internamente de RNN, tanto el encoder como el decoder. La primera cuestión que cabe plantearse es qué tipo de RNN usar. Existen varias topologías basadas en el concepto de RNN. Hay redes como Elman and Jordan networks o Neural Touring Machines, pero su mención es prácticamente anecdótica, ya que han quedado obsoletas debido a la eficacia demostrada por las redes LSTM [53] y GRU [54], actualmente las topologías más conocidas dentro de las RNN.

Una de las razones por las que LSTM y GRU funcionan tan bien es que evitan un problema conocido como Vanishing Gradient [55]. Para entender por qué este fenómeno ocurre, primero hay que tener un conocimiento básico de cómo funciona una red neuronal.

Al igual que los seres humanos aprendemos a corregir nuestros errores en base a la experiencia para lograr una tarea en cuestión, las redes neuronales hacen lo mismo. Típicamente una red neuronal recibe unos datos de entrada y produce unas salidas, que determinarán la solución a la tarea que queramos resolver. Matemáticamente se modela la tarea como una función que hay que minimizar o maximizar, que se conoce como función de coste o error. El objetivo de la red es entonces encontrar la mejor configuración interna

que produzca que, evaluando la función de coste con las salidas obtenidas, se alcance el mínimo o el máximo de la función. Para encontrar la configuración óptima se utiliza la Retropropagación o Backpropagation, una técnica en la que se revisa la cadena de errores que propiciaron no alcanzar el objetivo propuesto, esto es, determinar el error cometido por cada neurona de la red. Conociendo estos errores se tiene una intuición de cuánto hay que modificar cada parámetro en cada neurona para que en la siguiente vez que se evalúe la red el error sea menor.

Explicado en términos un poco más técnicos, los errores de cada neurona se utilizan para calcular las derivadas parciales de cada parámetro de la red, que constituirán el gradiente o vector gradiente. Lo que se persigue es encontrar el mínimo (o máximo) de la función objetivo y el gradiente lo que expresa es cuánto crece la función en un punto específico (pendiente). Con esto sabemos que para encontrar el mínimo de la función hay que avanzar en dirección contraria y esto se hace modificando los parámetros erróneos. Seguir este proceso iterativamente hasta encontrar el mínimo de la función es lo que se conoce como Gradient Descent o Descenso del Gradiente.

En las RNN, cuanto más extensas sean las secuencias, más información hay y por tanto más compleja es la arquitectura o más bien, es más densa porque se añaden más capas y neuronas. Esto aumenta el número de parámetros y cuando se hace el Backpropagation, el Descenso del Gradiente implica calcular los gradientes de las funciones de activación de las neuronas, que son las funciones que definen la salida de cada neurona. Las funciones sigmoides fueron las más utilizadas durante muchos años, pero también son en parte responsables de que se produzca el Vanishing Gradient.

La función sigmoide produce como salida un número entre cero y uno (Figura 2.3 línea azul). Si n capas ocultas utilizan funciones de activación sigmoides, y el gradiente resultante es muy próximo a cero (Figura 2.3 línea naranja), n números próximos a cero se multiplicarán, ocasionando en el peor de los casos que la multiplicación sea 0, lo que significa que el valor del gradiente desaparece (Vanishing Gradient). Un gradiente pequeño o nulo implicará que los parámetros de la red no se actualicen correctamente y esto se hace notable porque las sesiones de entrenamiento son más duraderas o la evolución en la búsqueda del mínimo no mejora. El impacto de este problema se puede reducir utilizando

otras funciones de activación como la llamada Rectified Linear Unit (ReLU), que se comporta como una función lineal cuando los valores de entrada son positivos y como una función constante cuando los valores son negativos (Figura 2.4). Es una alternativa válida, pero el error no termina de desparecer.

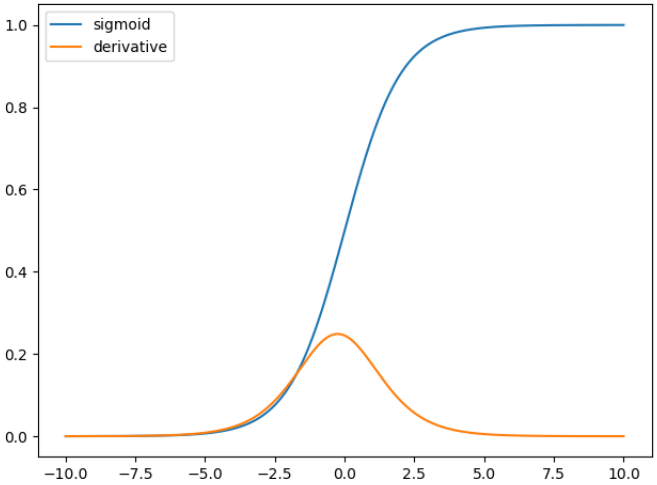


Figura 2.3.- Función sigmoide (línea azul) y el gradiente de la misma (línea naranja).

LSTM y GRU son dos topologías que evitan el Vanishing Gradient. GRU es relativamente nueva y supuestamente más eficiente, pero depende de cada problema, además cuando se tratan secuencias de longitud grande, LSTM rinde mejor porque GRU tiene menos parámetros de entrenamiento y, por tanto, menos memoria; es por esto que en un principio se escoge LSTM como RNN a utilizar.

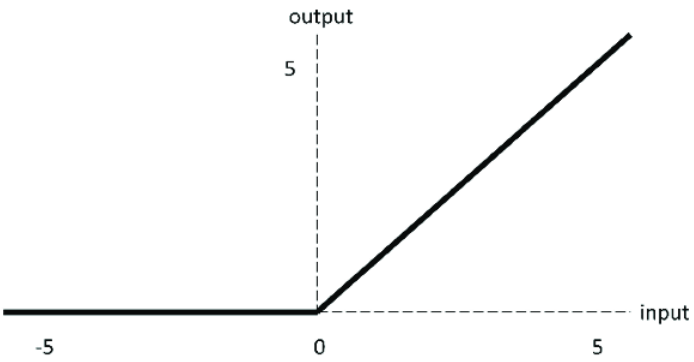


Figura 2.4.- Función ReLU, cuando el input es negativo es constante y cuando es positivo actúa como una función lineal.

Estructura LSTM

Los datos con los que se alimenta la red LSTM se organizan en vectores que contienen toda la información necesaria. En el caso que nos ocupa, cada dato corresponderá con un vector o secuencia que estará formada por x pasos de tiempo en el que en cada uno se mide el porcentaje de tiempo diario que el paciente estuvo en arritmia. Estas secuencias se distribuyen entre unos componentes llamados "células". Las células LSTM reciben típicamente 2 entradas: un vector o secuencia y también la salida de la célula anterior. Como efecto del procesamiento interno generan una salida, que generalmente se denomina "estado oculto" que será pasada a la siguiente célula (Figura 2.5).

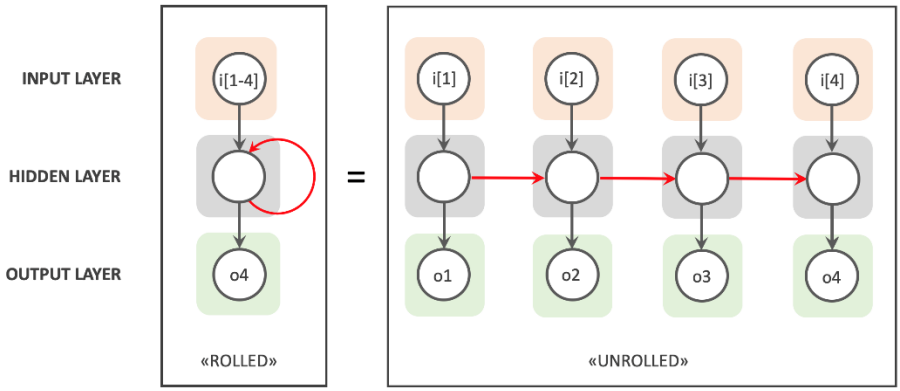


Figura 2.5.- Estructura de procesamiento de secuencias en una RNN (origen: *https://www.bouvet.no/bouvet-deler/explaining-recurrent-neural-networks*).

Las células preservan el valor de un estado a través del tiempo, dotando a la red de una memoria que permite recordar información relevante de la secuencia para realizar predicciones. Las operaciones internas de las células deciden la información que se recuerda o se olvida y es la esencia que diferencia las redes LSTM de otro tipo de RNN.

En la Figura 2.6 se puede ver un esquema de lo que ocurre dentro de una célula. Existen 3 puertas: puerta de entrada (input gate), puerta de olvido (forget gate) y puerta de salida (output gate).

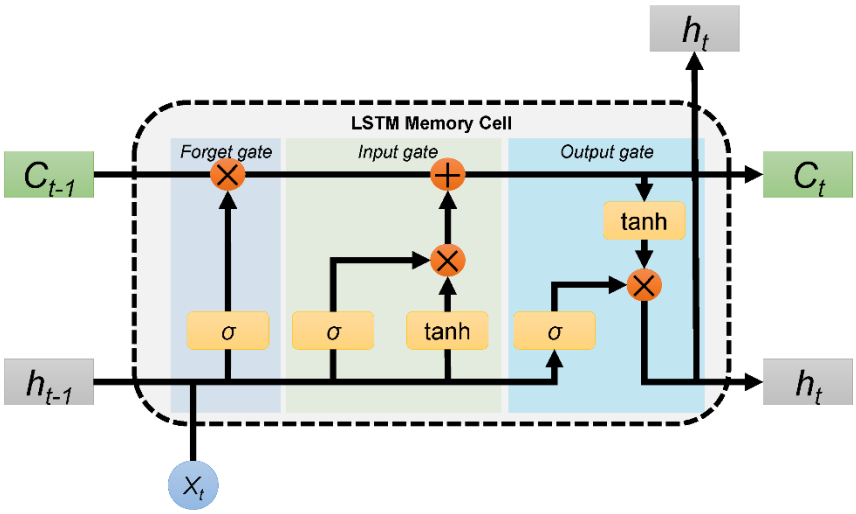


Figura 2.6.- Estructura interna de una célula LSTM (origen: https://www.mdpi.com/2073-4441/12/1/175/htm).

La *input gate* se utiliza para actualizar el estado oculto de las células. Los valores de entrada y el estado oculto anterior se pasan por una función sigmoide. Como se ha visto previamente, los valores de salida de la sigmoide son entre 0 y 1, lo que ocasiona que cuanto más cercanos sean a 0, más probabilidad hay de olvidar esa información y, al contrario, los valores cercanos a 1 propiciarán mantener esa información. La salida es multiplicada por la entrada pasada previamente por una función tanh para convertir los valores entre -1 y 1 y así regular la red. El resultado portará los valores de entrada que se mantienen activos.

La *forget gate* decide qué información debe ser despreciada o guardada. La información del estado oculto anterior y la información de entrada actual se pasa a través de una función sigmoide y se multiplica la salida por S_{t-1} (estado interno anterior de la célula). De nuevo, los valores resultantes cercanos a 1 serán más importantes y al contrario, los valores cercanos a 0 tenderán a olvidarse. Posteriormente, el resultado obtenido de la multiplicación se suma con el input recibido, esta suma es el nuevo estado de la célula.

Por último, el *output gate* decide cuál debe ser el siguiente estado oculto de la célula. Primero, se pasa el estado oculto anterior y la entrada actual por una función sigmoide. Luego se pasa el estado resultante por una función tanh. Se multiplica la salida de la función tanh con la salida de la sigmoide para decidir qué información debe llevar el estado oculto.

Las operaciones internas dentro de las células LSTM, especialmente en la puerta de olvido, permiten a la red actualizar los parámetros de tal manera que es menos probable que los gradientes de la información más relevante desaparezcan, y por tanto evitando que suceda el problema del Vanishing Gradient.

2.1.2.- Inferencia bayesiana variacional

Retomando el funcionamiento de los VAE, se puede plantear el esquema del mismo desde una perspectiva probabilista. En un VAE el entrenamiento se regulariza para asegurar que el espacio latente tenga buenas propiedades que permitan el proceso generativo. Esto se hace codificando los datos de entrada como una distribución sobre el espacio latente (tarea del encoder). En la Figura 2.7 se puede ver que el encoder aprende a partir de los datos de entrada, x, una distribución de probabilidad condicional (normalmente una gaussiana) $q_\theta(z|x)$ que se representa en un espacio latente. El decoder a partir de las variables latentes (z) reconstruye los datos de entrada aprendiendo otra distribución de probabilidad $p_\Phi(x|z)$.

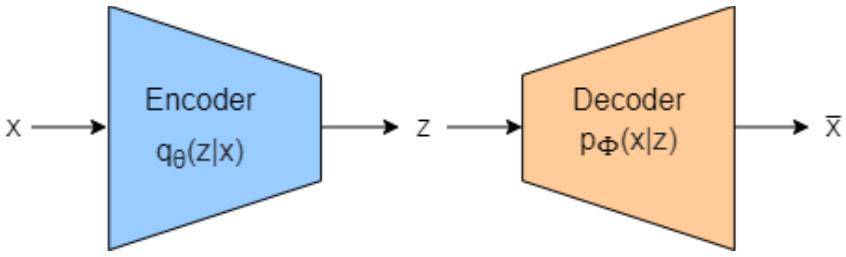


Figura 2.7.- Arquitectura de un VAE desde un enfoque probabilístico.

Adaptando estos términos a nuestro problema tendríamos:

- x: son los datos de entrada, los datos intracardiacos.
- $P(x)$: es la distribución de probabilidad que siguen los datos.
- z: son las representaciones de alto nivel que se aprenden de los datos (variables latentes).
- $P(z|x)$: describe la distribución de la variable codificada (z) dada por la decodificada (x), es decir, es la distribución de probabilidad simplificada de los datos.

- $P(x|z)$: describe la distribución de la variable decodificada (x) dada por la codificada (z), es decir, es la distribución de los datos reconstruidos a partir de las variables latentes.

El modelo probabilístico que abarca el VAE viene definido por:

$$P(x, z) = P(x|z)\, P(z) \tag{1}$$

Suponemos que $P(z)$ es una distribución gaussiana estándar y que $P(x|z)$ es una distribución también gaussiana cuya media está definida por una función determinista f de la variable z y cuya matriz de covarianza tiene la forma de una constante positiva c que multiplica la matriz de identidad I:

$$\begin{aligned} P(z) &\equiv N(0, I) \\ P(x|z) &\equiv N(f(z), cI) \end{aligned} \tag{2}$$

Al hacer estas suposiciones se parte de que se sabe $P(z)$ y $P(x|z)$, por tanto se puede utilizar el teorema de Bayes para calcular $P(z|x)$, que se puede expresar como:

$$P(z|x) = \frac{P(x|z)P(z)}{P(x)} \tag{3}$$

El problema reside en el denominador. Marginando las variables latentes se puede calcular $P(x)$ como:

$$P(x) = \int P(x|z)\, P(z) dz \tag{4}$$

Desafortunadamente, este cálculo es intratable por el coste computacional que supone debido a que requiere tiempo exponencial para su cálculo. Es por esto que se requieren técnicas de aproximación como la inferencia variacional.

La inferencia bayesiana es un tipo de inferencia estadística en el que las observaciones se utilizan para inferir la probabilidad de que una hipótesis pueda ser cierta. La inferencia

variacional es uno de los métodos más recurridos de la inferencia bayesiana que se utiliza para aproximar integrales intratables. En palabras más simples, es una técnica utilizada para aproximar distribuciones complejas.

En el caso que nos ocupa, la distribución compleja que se quiere aproximar es $P(z|x)$. Para aproximarla mediante inferencia variacional la idea es presuponer una familia de distribuciones (distribución gaussiana) q_θ tratables y simples y plantear la aproximación como un problema de optimización: buscar dentro de la familia de distribuciones q_θ la distribución q que más se aproxime a la distribución objetivo $P(z|x)$ para poder utilizar q en lugar de $P(z|x)$. El parámetro variacional θ denota los parámetros de la familia de distribuciones elegida, por ejemplo, si asumimos que va a ser una gaussiana, θ serían la media y la varianza de las variables latentes.

Para medir la similitud entre q y $P(z|x)$ se utiliza la divergencia de Kullback-Leibler (KL):

$$D_{KL}\big(q_\theta(z|x)||P(z|x)\big) = E_q[\log q_\theta(z|x)] - E_q[\log P(x,z)] + \log \mathrm{P}(x), \quad (5)$$

que mide cuánto difieren dos distribuciones de probabilidad. Con esta medida se sabrá como de bien aproxima q a $P(z|x)$.

Nuestro objetivo será encontrar los parámetros variacionales θ que minimicen el valor de la divergencia KL. Esto se puede expresar como:

$$q_\theta(z|x) = \arg\min D_{KL}\left(q_\theta(z|x)||P(z|x)\right) \quad (6)$$

Aquí aparece un nuevo problema: no es posible calcular la ecuación directamente. Otra vez aparece el término $P(x)$ en (5) y como vimos anteriormente su cálculo es intratable. Se necesita aún un paso más conseguir una inferencia variacional tratable. Considérese la siguiente función:

$$ELBO(\theta) = E_q[\log P(x,z)] - E_q[\log q_\theta(z|x)] \quad (7)$$

Corresponde con el Evidence Lower BOund, que nos permitirá aproximar $P(x)$. Se puede combinar (7) con la divergencia KL expresada en (5) y reescribir la evidencia como:

$$\log \mathrm{P}(x) = ELBO(\theta) + D_{KL}\big(q_\theta(z|x)||P(z|x)\big), \tag{8}$$

Según la desigualdad de Jensen, la divergencia KL es siempre mayor o igual a cero. Sabiendo esto y con la expresión de (8) se puede inferir que minimizar la divergencia KL es equivalente a maximizar el ELBO. En resumen, en lugar de calcular la divergencia KL (no tratable), lo que hacemos es maximizar el ELBO, objetivo que sí es tratable. De esta manera maximizando el ELBO (mediante el Descenso del Gradiente sobre los parámetros de θ) tendremos una aproximación a $\mathrm{P}(x)$, con la que conseguiremos que la diferencia entre las dos distribuciones q y $P(z|x)$ sea mínima. Con esto ya se tendría $P(z|x)$ y, por tanto, todos los elementos necesarios.

Recuérdese la Figura 2.7. El encoder lo que hace es aproximar $P(z|x)$ (ahora ya sabemos cómo) mediante $q_\theta(z|x)$. Toma como entrada un conjunto de datos x y como salida tendrá los parámetros que define θ, es decir la media y la varianza de las variables latentes. Por otra parte, se parametriza $P(x|z)$ con una red generativa (el decoder) que toma las variables latentes y las reconstruye en el espacio original de datos a la distribución aproximada $p_\phi(x|z)$.

2.1.3.- Simulación de registros intracardíacos.

Cuando se realiza un ECG se colocan unos electrodos en el pecho del paciente para registrar las señales eléctricas del corazón. Se registra todo tipo de actividad eléctrica que es enviada al electrocardiógrafo, quien trascribe la información capturada a una representación gráfica (Figura 1.7, izquierda) que se muestra en un monitor. Los electrocardiogramas intracardiacos, en contraposición, simplemente representan la diferencia de potencial entre dos puntos en contacto con el tejido muscular del corazón, el miocardio (Figura 1.7, derecha). Como ya se comentó en la anterior sección, la morfología de la actividad cardíaca se pierde, por tanto, su utilidad queda relegada a un segundo plano. Sin embargo, los marcapasos también registran las fechas y duraciones de cada episodio de arritmia.

La captura de datos de los marcapasos no es constante, sino que hay ciertos eventos que suscitan su registro, concretamente los episodios de alta frecuencia auricular. Cuando entran en acción emiten una descarga eléctrica que activa las células del corazón para favorecer la contracción cardíaca. El algoritmo del marcapasos tiene diferentes modos de operación, que se van turnando en función de la actividad registrada. Cuando se produce un episodio de arritmia con frecuencia auricular más alta de lo debido, es un indicador que el paciente está sufriendo un episodio de FA, por tanto, el marcapasos cambia su modo de operación de reposo a emisión de descarga eléctrica para propiciar la excitación del ventrículo. El cambio de modo para actuar ante la presencia de una arritmia es parte de un proceso conocido como Automatic Mode Switching (AMS) y cuando se produce se activa la captura de información, que será la fecha en la que se produce el episodio, su duración y el iECG resultante (Figura 2.8).

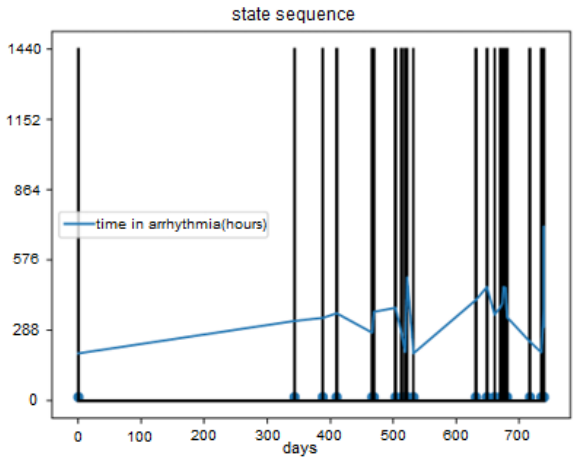


Figura 2.8.- Fechas en las que se produjo un cambio de modo o inicio de episodio de FA y duraciones de cada episodio.

Para que el funcionamiento del algoritmo sea eficiente, es necesario configurar los parámetros del marcapasos, que son específicos de cada paciente. Generalmente se utilizan los iECG para ajustar estos parámetros. El problema del proceso AMS es que, como se priorizan los falsos positivos para no generar la excitación del ventrículo en momentos inadecuados, hay episodios de arritmia largos que se reportan como varios cortos. Es decir, es común que el algoritmo detecte la presencia de una arritmia, por tanto, genere un evento AMS y luego la de por finalizada, para descubrir segundos después que el episodio aún está ocurriendo, por lo que genera otro evento AMS para atenuar la arritmia y posteriormente

cuando se consigue, el modo vuelve al estado inicial. Estos casos no suponen ninguna consecuencia trascendental ni para el paciente ni para el dispositivo, pero sí influye en la fidelidad de los datos registrados, motivando la captura de información inexacta.

Como se explicó en la introducción, una de las partes del proyecto es la consecución de un modelo que sea capaz de simular registros intracardíacos, por tanto, hay que tener en cuenta lo recientemente explicado a la hora de desarrollar un modelo que obedezca al comportamiento que siguen los marcapasos para capturar los eventos de FA.

Modelo de Markov

El modelo propuesto para simular el funcionamiento de un marcapasos o IDC se representa en la Figura 2.9 como un modelo de Markov continuo donde existen tres estados: "Normal", "Arritmia" y “Falso Normal". Un paciente está en estado "Normal" hasta que se detecta una arritmia y el dispositivo emite un evento AMS, entonces el paciente cambia al estado "Arritmia". Hay dos caminos posibles para salir de este estado: volver a "Normal" cuando el episodio termina, o una transición a "Falso Normal" cuando se emite un falso final de episodio. En este segundo caso, el paciente permanece en el estado "Falso Normal" hasta que se despacha un nuevo episodio de AMS y vuelve al estado "Arritmia". De esta manera, los eventos AMS marcan el comienzo de un verdadero episodio de FA o el final de un estado de "Falso Normal". Esta segunda clase de eventos de AMS son anormales y deberían ser eliminados, pero no existe un procedimiento sencillo para eliminarlos de los datos del marcapasos, por lo que estos eventos estarán presentes en los pacientes reales, y, por tanto, el modelo generativo debe producir también estos eventos espurios.

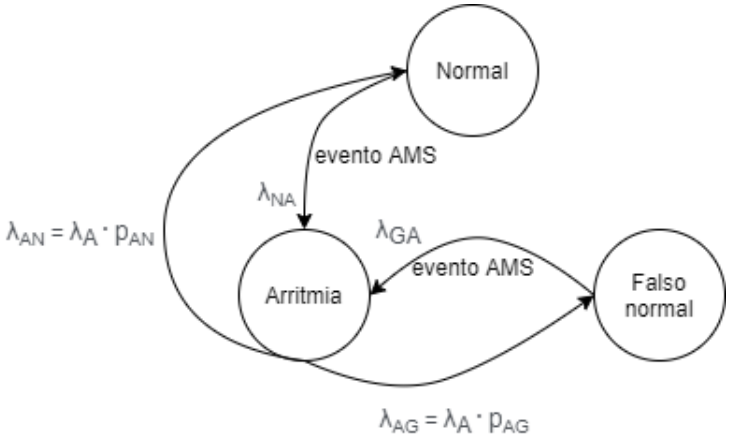


Figura 2.9.- Diagrama de estados del modelo de los inicios de episodios de FA.

Se asume que el tiempo entre dos episodios sigue una distribución exponencial con el parámetro λ_{NA}. La duración de un episodio también sigue una distribución exponencial con el parámetro λ_A. La progresión de la FA paroxística a permanente se mide por la velocidad de cambio de estos dos parámetros: a medida que la condición cardíaca empeora, el tiempo entre los episodios es más corto y los episodios son más largos. La velocidad de la progresión está modelada por un parámetro $\alpha \in [0,1]$,

$$\lambda NA(t) = \lambda NA(0) \cdot \alpha^{t}, \quad (9)$$

$$\lambda A(t) = \lambda A(0) \cdot \alpha^{-t}, \quad (10)$$

donde $\alpha = 1$ es un paciente estable y los valores inferiores a 1 son pacientes con una rápida progresión a arritmia permanente. También se supone que la transición del estado "Arritmia" a "Normal" puede ocurrir con una probabilidad p_{AN}. La probabilidad de la transición de "Arritmia" a "Falso Normal " es por lo tanto $p_{AN} = 1- p_{AG}$. p_{AG} es la fracción de falsos positivos, que es la probabilidad de que el algoritmo de detección de FA en el marcapasos señale el final de un episodio demasiado pronto.

En resumen, el modelo generativo propuesto es un modelo de Markov en tiempo continuo caracterizado por 5 parámetros (λ_{NA}, λ_{GA}, λ_A, p_{AG} y α). Con este modelo es posible generar una lista de eventos de forma aleatoria que se pueden interpretar como un paciente hipotético

cuyo tipo de FA está definido por los parámetros anteriores. Estos datos se utilizarán como conjunto de entrenamiento para el VAE propuesto.

2.2.- ARQUITECTURA DEL MODELO

Se divide el problema en dos pasos: un primer paso de entrenamiento no supervisado seguido de un paso de aprendizaje supervisado. El VAE se entrena con el conjunto de datos generado por el modelo de Markov descrito en la sección anterior para aprender la representación latente de los datos (aprendizaje no supervisado) y el encoder resultante se utiliza como extractor de características sobre el cual se entrena un clasificador lineal (aprendizaje supervisado).

2.2.1.- Encoder

En un VAE el entrenamiento prioriza que el espacio latente tenga buenas propiedades que permitan el proceso generativo. Precisamente estas propiedades contribuyen a que los datos de entrada se mapeen en el espacio latente de tal manera que los datos que sean similares se localicen en zonas cercanas (recordar Figura *1.9*). El papel del encoder se puede interpretar como un extractor de características, un proceso por el cual un conjunto de datos se reduce a grupos más manejables para su procesamiento, en nuestro caso, los distintos tipos de arritmia.

Un VAE, al recibir un dato de entrada, trata de encontrar un vector latente que sea capaz de describirlo y al mismo tiempo tenga las propiedades necesarias para generarlo de nuevo. Aludiendo a la explicación de 2.1.2.- pero aplicado a redes neuronales se tiene: El encoder toma como entrada un conjunto de datos y como salida aproxima $P(z|x)$ mediante $q_\theta(z|x)$; el decoder toma las variables latentes y reconstruye o genera datos de la distribución $p_\phi(x|z)$. Los parámetros θ y ϕ se corresponden con los pesos y los sesgos de cada red (encoder y decoder) que se optimizan para maximizar el ELBO (recodar (7)) usando el Descenso del Gradiente. Se puede escribir el ELBO para incluir los parámetros de la red generativa como:

$$L_{VAE} = E_{q\Phi(z|x)}[\log p_\theta(x|z)] - D_{KL}(q_\Phi(z|x)||p_\theta(z)), \quad (11)$$

La denominación L_{VAE} se debe a "Lower Bound". El primer término de la resta es la reconstrucción de x que tiende a hacer que el esquema de codificación-decodificación sea lo más eficiente posible al maximizar el logaritmo de verosimilitud $\log p\theta(x|z)$ con el muestreo de $q_{\Phi(z|x)}$, modelado por una red neuronal (el encoder) cuyo resultado son los parámetros de una Gaussiana: una matriz de covarianzas diagonal y una matriz de medias. El segundo término tiende a regularizar la organización del espacio latente haciendo que las distribuciones devueltas por el codificador se aproximen a una normal estándar. Se regulariza la variable latente z minimizando la divergencia KL entre la aproximación variacional y la distribución previa de la variable latente.

El codificador, representado por $q_\Phi(z|x)$ es la parte que se utilizará como extractor de características, ya que su objetivo es mapear los datos de entrada a un espacio latente de dimensiones reducidas. Para utilizar la dependencia temporal de las series intracardiacas, se sustituye el encoder convolucional de un VAE tradicional por un encoder recurrente. De esta manera, el codificador se aproxima a la distribución gaussiana $p_\theta(z)$ alimentando la salida de un LSTM en dos módulos lineales para estimar su media y covarianza. La compresión de los datos de entrada da como resultado un espacio latente bidimensional dominado por los ejes representados por la media y la varianza de la distribución aproximada. Se espera que las arritmias se agrupen en diferentes conjuntos según sus características, simbolizando una representación más simple de su naturaleza.

Basándose en la representación aprendida por el encoder, los datos x se muestrean a partir de la distribución de probabilidad condicional p(x|z). A efectos generativos, esta regularización en el espacio latente es muy eficaz para facilitar el muestreo aleatorio y la interpolación para la creación de nuevos datos. Este es el objetivo del decodificador y es la aplicación más utilizada de los VAEs en la literatura. Sin embargo, se decide descartar esta parte después del entrenamiento porque el principal interés está en el diagnóstico de los datos de entrada en base a las representaciones latentes aprendidas en lugar de la generación de nuevos tipos de arritmia.

2.2.2.- Mapa de diagnóstico

La herramienta de diagnóstico introducida en este estudio es un mapa codificado con colores que muestra el estado real del paciente y la velocidad de cambio de su condición de FA paroxística a permanente. Una vez que el VAE se entrena con la simulación de los eventos AMS (modelo de Markov), se obtiene un mapa topológico en el espacio latente a partir del cual se identifican los grupos correspondientes a los diferentes tipos de arritmias, como se puede ver en la Figura 2.10. A este respecto, el mapa puede considerarse como una proyección de los datos intracardiacos en un espacio cuyas coordenadas son los valores de λ_{NA}, λ_{GA}, λ_{A}, p_{AG} y α. Los valores de λ_{NA}, λ_{A} y α miden la condición del paciente y la progresión de la FA. λ_{GA} y p_{AG} miden la posibilidad de que un evento de AMS en el marcapasos sea espurio, son parámetros propios de cada dispositivo y en principio no son críticos para obtener un diagnóstico preciso.

Cuando se utilicen datos reales como entrada del modelo, es decir, los datos intracardiacos de un paciente, el encoder los situará de acuerdo a sus características, en una zona del espacio latente que proporcionará información acerca del tipo de arritmias que sufre el paciente: en primer lugar, se conocerán los parámetros que más se ajusten al estado de la enfermedad en función del grupo en el que caigan los datos de la actividad intracardíaca del paciente y en segundo lugar, acorde con la cercanía a otros grupos de arritmias entrenados con otros parámetros diferentes, se conocerá cuál es la evolución más probable hacia la que tiende la enfermedad, procurando a los especialistas médicos una visión de cómo podría desarrollarse la enfermedad si no se actuara prematuramente.

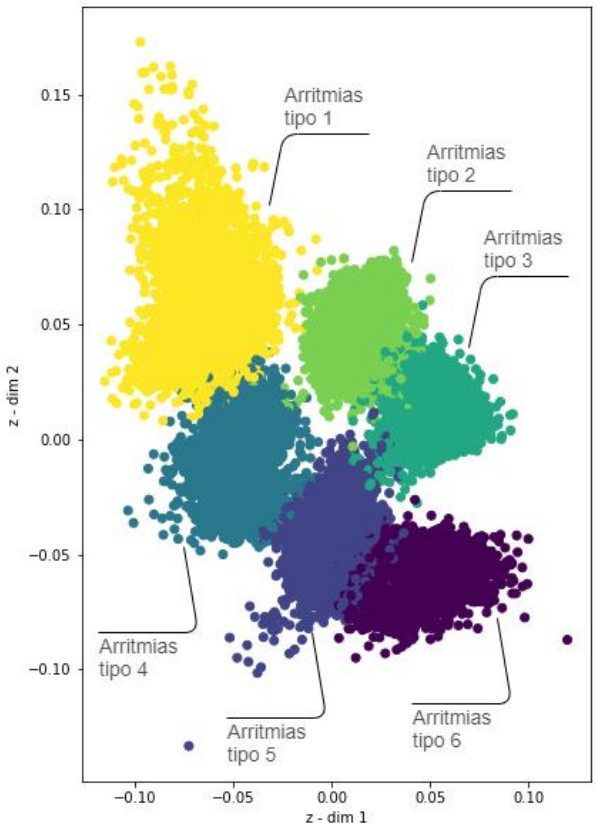


Figura 2.10.- Representación aprendida de los eventos AMS simulados.

2.2.3.- Clasificador

Existen trabajos de aprendizaje supervisado aplicado a VAEs, entre el que destaca el propuesto por los autores del VAE en colaboración con investigadores de DeepMind [56], donde primero se entrena un extractor de características (encoder) y sobre ese modelo ya entrenado se construye un clasificador. La principal diferencia entre su modelo y el propuesto en este trabajo es que la optimización del extractor de características y del clasificador se hacen simultáneamente. La razón para hacer esto es que si se entrena un VAE sin ninguna restricción, el espacio latente resultante priorizará las propiedades generativas del modelo, y esto se traduce en la localización de los datos de entrada en zonas donde arritmias que no pertenecen al mismo grupo de parámetros con el que fueron entrenadas se sitúan demasiado cerca incluso en algunos casos se superponen (ver Figura 2.11).

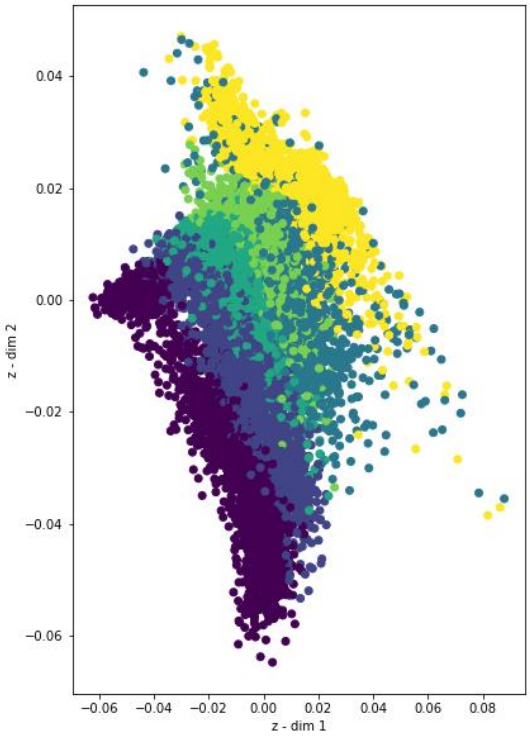


Figura 2.11.- Representación latente aprendida sin incluir el clasificador en el entrenamiento.

Como complemento al diagnóstico visual que se puede ofrecer en el mapa latente obtenido, el clasificador resultante puede explicar cuantitativamente qué parámetros modelados por el modelo de Markov son los que mejor representan cada arritmia que se alimenta. Además, el hecho de construir un clasificador que prediga explícitamente la clase a la que pertenecen las arritmias de un paciente dado, permitirá hacer comparaciones de nuestro modelo con otros clasificadores del estado del arte, como se verá en la siguiente sección.

2.2.4.- Ajuste de parámetros

Una vez definidas las partes del modelo, el siguiente paso a realizar es el entrenamiento de la red neuronal. El entrenamiento está condicionado por el número de “epochs”, que son el número de iteraciones necesarias para que la red vea todos los datos de entrenamiento. En cada epoch los datos se analizan por batches, de tal manera que una vez se evalúan todos los batches, se completa una epoch y se obtienen las métricas que se deseen para evaluar el rendimiento del modelo. En base a las métricas resultantes se actualizan los pesos de la red con el Descenso del Gradiente.

El Descenso del Gradiente minimiza la función objetivo optimizando los parámetros de las neuronas en dirección contraria al gradiente de la función de coste para encontrar, en este caso, el mínimo. Hay un parámetro, la ratio de aprendizaje o Learning Rate (LR), que define cuánto se avanza en cada iteración para encontrar el mínimo de la función, esto es cuánto afecta el gradiente a la actualización de los parámetros de la red.

Como entre los objetivos propuestos está la obtención de un VAE y un clasificador, en realidad la función objetivo a minimizar serán dos: una función para conseguir un VAE que aprenda una representación eficiente de los datos de entrada a la vez que permita el proceso generativo y otra función para optimizar el porcentaje de clasificación de los distintos tipos de arritmias. Estas funciones se unifican mediante la suma de ambas para su optimización conjunta.

3. Trabajo realizado y resultados

Todos los modelos y experimentos han sido implementados en Python. El código fuente para reproducir los resultados experimentales está disponible públicamente en el repositorio de Github https://github.com/NahuelCostaCortez/RVAE.

Esta sección contiene las tareas realizadas para la simulación de distintos tipos de arritmia mediante el modelo de Markov propuesto.

3.1.- SIMULACIÓN DE REGISTROS INTRACARDÍACOS

Se busca tener un conjunto de datos representativo que incluya un rango amplio de diferentes tipos de arritmias que puedan sufrir pacientes de FA. Reunir un conjunto de datos de estas características con registros de personas reales es prácticamente imposible, primero por la sensibilidad de los datos y después por la escasez de datos de pacientes reales a los que se tiene acceso. En particular, los datos reales con los que se ha podido trabajar en este proyecto han sido proporcionados por Medtronic y en ningún caso se ha proporcionado información personal del paciente. Debido a estos motivos, se intuye necesaria la creación del modelo explicado en el apartado 2.1.3. Los datos con los que se entrenará el VAE propuesto se obtienen con este modelo, cuya única misión es la simulación de registros intracardíacos.

Las simulaciones imitan la progresión de la FA en un hipotético paciente. Es posible modificar la evolución de la enfermedad variando los parámetros que rigen el modelo. Se simula un número de días determinado, entre los cuáles la máquina de estados decidirá qué eventos son los que ocurren en el paciente, por tanto, se conoce exactamente en qué instante el individuo entró o salió de cada uno de los tres estados: Normal, Arritmia y Falso Normal. Tras la simulación, hay un postprocesado para extraer información que permite construir una serie temporal: se consultan las fechas de inicio y de fin de cada episodio de arritmia para conocer los momentos en los que el sujeto entra y sale del estado "Arritmia". Posteriormente, se calcula para cada día simulado el tiempo que pasó el paciente en ese estado y finalmente se obtiene el porcentaje de tiempo en arritmia diario. Con esta información se puede construir una gráfica para cada paciente semejante a la de la Figura 3.1.

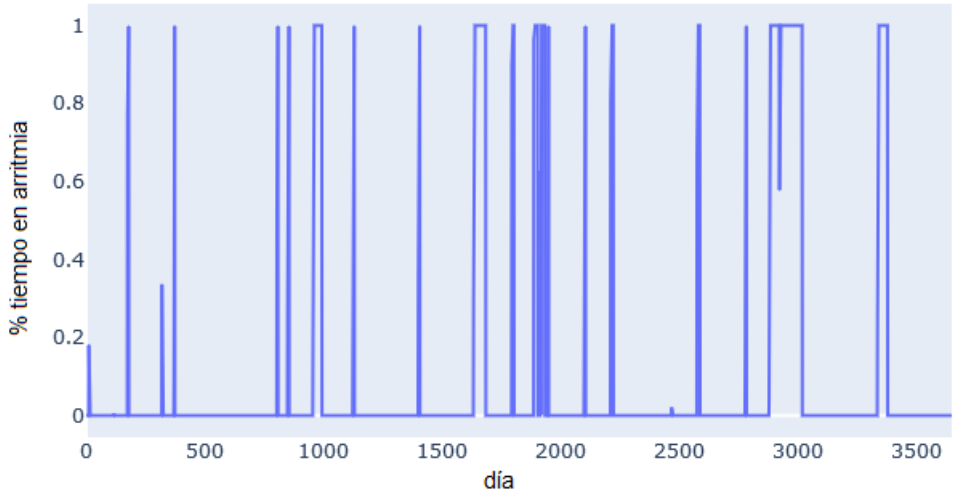


Figura 3.1.- Porcentaje de tiempo en arritmia de un paciente.

Alcanzado este punto, podría plantearse emplear estas series como entrada para el VAE, en las que se refleja el porcentaje diario de tiempo en arritmia de cada paciente. Sin embargo, a simple vista se puede percibir que la morfología de la secuencia no se adecúa a la de una serie temporal clásica como las de la Figura 3.2. Esto se debe a que la serie tiene muy pocos pasos de tiempo que aporten información importante. Además, esto se agravaría en aquellos pacientes que no están en estado crítico porque el porcentaje de tiempo diario que pasan en

arritmia es muy bajo o directamente cero. Esto ocasionaría, si se decidiera utilizar estos datos como conjunto de entrenamiento, que el rendimiento de la red sería insignificante.

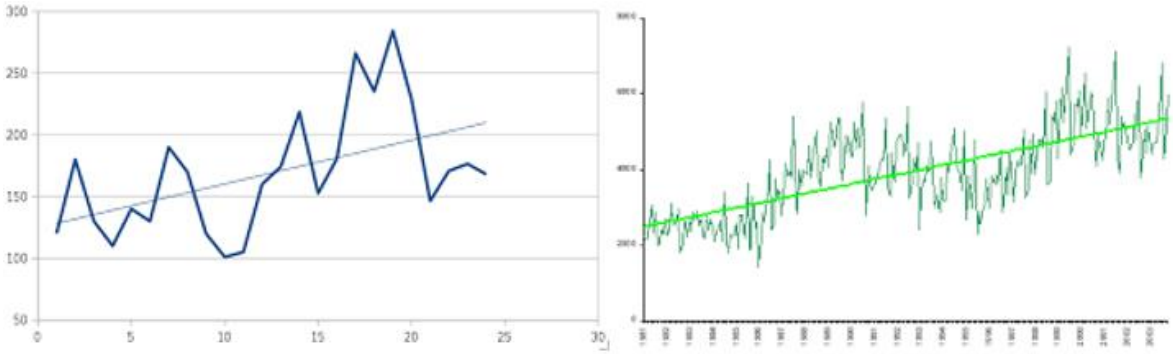


Figura 3.2.- Ejemplos de series temporales.

Se decide, por tanto, cambiar la perspectiva en la que se representan los datos intracardiacos en la serie temporal: la información se representa en un gráfico de dispersión suavizado que muestra la progresión de las arritmias que sufre un paciente: A partir de las secuencias obtenidas, se aplica un suavizado con kernel Gaussiano tomando como ancho de la ventana unos pocos días. Con esta técnica la morfología de la secuencia obtenida anteriormente se transforma de tal manera que se puede apreciar la evolución de las arritmias paroxísticas a permanentes. En la Figura 3.3 se muestra un ejemplo del resultado conseguido: el inicio de la gráfica corresponde con arritmias espontáneas, que a medida que pasa el tiempo se producen con más frecuencia y son de mayor duración, coincidiendo el final de la gráfica con el paso a arritmias crónicas, donde el porcentaje de tiempo de arritmia diario es muy alto.

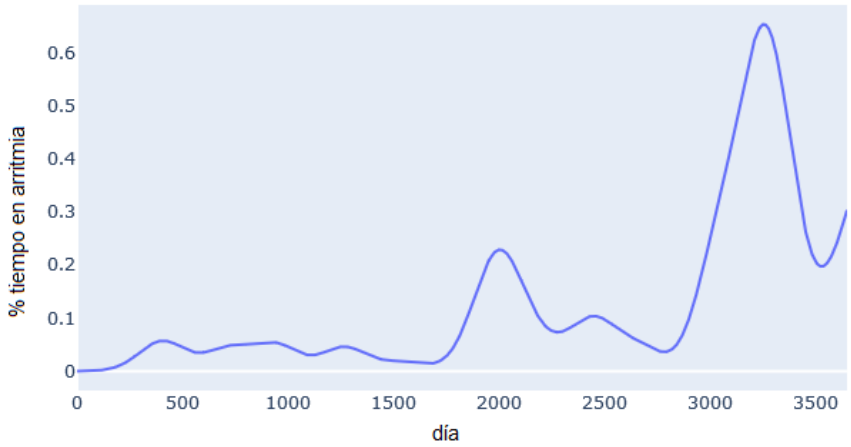


Figura 3.3.- Suavizado aplicado al porcentaje de tiempo en arritmia de un paciente.

Con esta transformación en la representación de los datos, se está más cerca de obtener unas series temporales con las que se pueda trabajar, pero sigue habiendo un inconveniente: el tiempo de simulación. En las secuencias obtenidas a partir de la simulación, cada punto en la gráfica corresponde con un día, por tanto, si se simulan datos de 10 años, la secuencia estará compuesta por 3650 puntos (10 años por 365 días que tiene un año) lo que supone una longitud intratable para cualquier red LSTM. En su lugar, se reduce la cantidad de puntos que conforman la gráfica a 146, que es una longitud más manejable. Se escoge el valor de cada punto como la media de los más cercanos, en el caso concreto de 3650, serían los 25 puntos más próximos. A pesar de reducir drásticamente la cantidad de puntos, no se pierde información porque la morfología de la gráfica se sigue manteniendo (Figura 3.4).

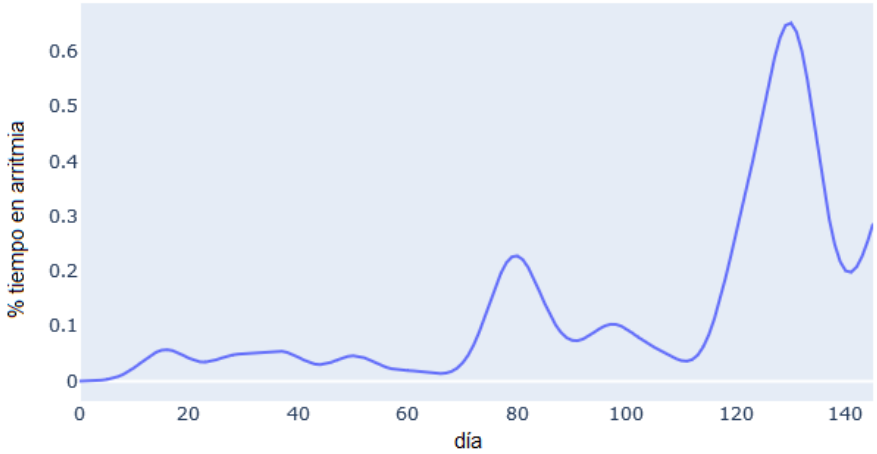


Figura 3.4.- Representación del paciente de la Figura 3.3 con menos puntos.

Esta última figura representa la secuencia que se pretende obtener para que pueda ser procesada por el VAE. Cada simulación realizada por el modelo deberá seguir el proceso descrito para finalmente obtener un conjunto de datos, que sea lo suficientemente representativo para reflejar distintas evoluciones de la FA.

En la siguiente sección se explican los pasos seguidos para entrenar la red neuronal propuesta de forma eficiente.

3.2.- EXPERIMENTACIÓN

Una vez diseñado el modelo de VAE y con los datos de entrenamiento listos, se procede a realizar los experimentos necesarios para conseguir una metodología de entrenamiento que conduzca hacia la solución que mejor se ajuste al problema a resolver. En esta sección se recoge un resumen de la evolución en el proceso de experimentación hasta la obtención de un modelo eficiente.

Para el desarrollo de los experimentos se utilizaron los siguientes frameworks:

- Jupyter Notebooks: es una aplicación de código abierto ampliamente utilizada en investigación que permite crear “cuadernos” que contienen código.

- Google Colaboratory: es un entorno de ejecución gratuito de Google basado en Jypyter Notebooks (Figura 3.5). La principal ventaja es que proporciona servicios de

computación en la nube que dan acceso a la utilización de GPUs (Graphic Processing Units). Estas unidades son esenciales para el entrenamiento de redes neuronales porque pueden optimizar la carga computacional de estos algoritmos. Concretamente las experimentaciones fueron llevadas a cabo en una GPU Tesla k40.

- Keras: es una biblioteca para redes neuronales de código abierto escrita en Python. Es un framework de alto nivel (más adaptable al lenguaje humano) que puede ejecutarse sobre otras bibliotecas de más bajo nivel como Theano o TensorFlow. Para las experimentaciones se utilizó TensorFlow como backend.

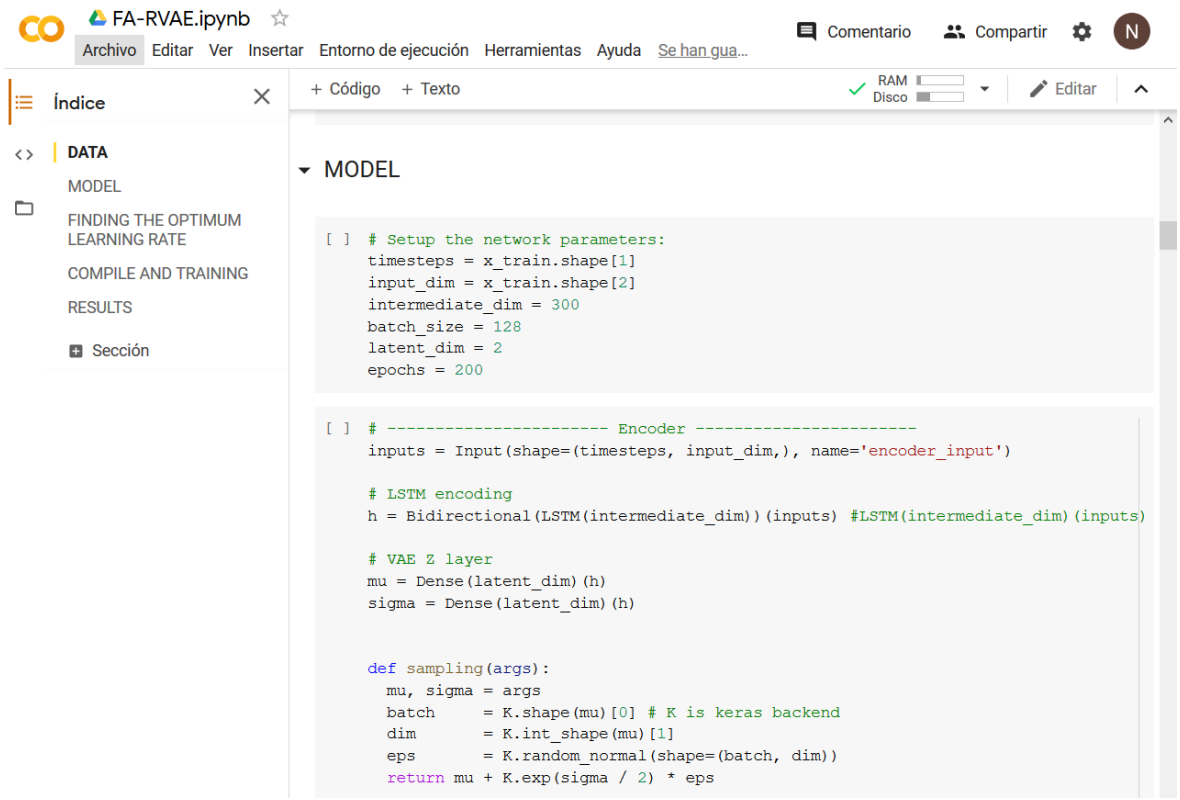


Figura 3.5.- Extracto de un cuaderno Jupyter creado para este proyecto en la plataforma Google Colaboratory.

Primera aproximación

Se parte de la base que el tipo de dato con el que se va a trabajar son series temporales y la arquitectura que mejor se adecua a este tipo de dato son las RNN. A pesar de esto, los experimentos van a seguir un proceso incremental, comenzando con arquitecturas sencillas y también con datos más simples para poder avanzar con firmeza hacia la obtención de la mejor arquitectura posible.

La primera aproximación se basa en una arquitectura básica donde todas las neuronas de cada capa están completamente conectadas (Fully Conected Layer) con las demás. Además, por el momento solo estarán presentes el encoder y el decoder (sin el clasificador) para evaluar únicamente el funcionamiento del VAE.

El tipo de dato utilizado para estos primeros experimentos son ondas senoidales. Esta elección se debe a la distribución de la que proceden estos datos, a priori más fácil de aprender que la del conjunto de datos final, ya que su naturaleza tiene un factor periódico que depende sólo de tres parámetros: frecuencia, amplitud y fase. Se generan ondas con frecuencias en el rango [1.0, 4.0], amplitudes en [0.125, 0.5], y fases aleatorias entre [-π, π] (Figura 3.6).

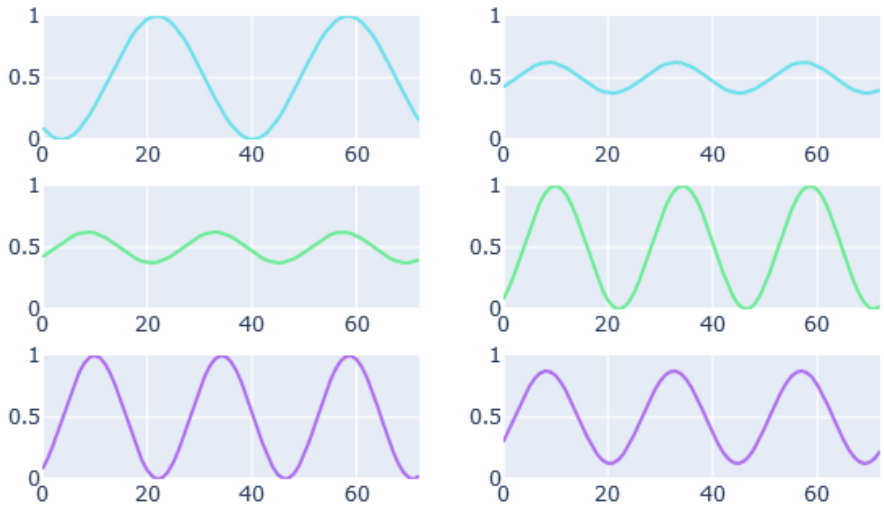


Figura 3.6.- Muestras del conjunto de datos construido con secuencias senoidales.

El conjunto de datos se divide en:

- train: es el conjunto de datos utilizado para entrenar el modelo.
- validation: conjunto de datos utilizado para dar una estimación de la habilidad del modelo durante el entrenamiento (validar el modelo).
- test: conjunto de datos utilizado para proporcionar una evaluación no sesgada de un modelo ya entrenado sobre el conjunto de entrenamiento.

Un punto importante en el entrenamiento del modelo es la elección de hiperparámetros. Estos son parámetros propios del modelo que gobiernan la forma en la que el proceso de entrenamiento se efectúa y se deben ajustar para poder conseguir un entrenamiento eficiente que proporcione como resultado un buen modelo. Inicialmente, se comienza el entrenamiento con estos hiperparámetros:

Hiperparámetros	
Dimensión intermedia	300
Batch Size	128
Nº de epochs	200
Optimizador	RMSprop

Tabla 3.1.- Hiperparámetros con los que se entrena el VAE en la primera aproximación.

- Dimensión intermedia: Es la cantidad de neuronas que tienen las capas implementadas.
- Batch Size: Define cada cuantas muestras vistas del conjunto de datos se ajustan los parámetros de la red (backpropagation). Por ejemplo, si se tienen 2400 ejemplos de entrenamiento y el tamaño del batch es 100, significa que la red se entrena primero con 100 muestras, se actualizan los parámetros y luego se vuelve a entrenar con las 100 muestras siguientes hasta llegar a las 2400 muestras.
- Nº de epochs: como ya se mencionó en el apartado 2.2.4.-, el número de epochs define el número de iteraciones en las que el modelo examina todos los datos del conjunto de entrenamiento. Cada epoch está dividido en batches (Batch Size). Si

hubiera 14080 muestras de entrenamiento y el tamaño del batch es 128, la red actualiza sus parámetros cada 14080/128 muestras, es decir, cada 110. Cuando se examinen las 14080 muestras se habrá completado una epoch, así que se repetirá el proceso 200 veces (según la Tabla 3.1).

- Optimizador: Los optimizadores son algoritmos que cambian parámetros del modelo para reducir la función de pérdida. El optimizador más conocido cuando se habla didácticamente de redes neuronales es el Descenso del Gradiente (explicado en el apartado 2.1.1.-). El resto de optimizadores (salvo alguna excepción) son derivaciones de este algoritmo. En este caso se utiliza RMSprop, que combina la idea de utilizar el signo del gradiente, propio del Descenso del Gradiente, con la idea de adaptar cuánto se avanza en esa dirección para encontrar el mínimo de la función. Se recuerda que el Learning Rate (apartado 2.2.4.-) es precisamente el parámetro que define cuánto se avanza, por eso RMSprop es un método de Learning Rate adaptativo, porque modifica su valor a medida que avanza el entrenamiento.

Para valorar la bondad de los modelos se dispone de las evaluaciones de la función de pérdida. Esta función es una combinación de dos sumandos: la pérdida en reconstrucción de los datos y la pérdida de la divergencia KL para asegurar que el espacio latente sea continuo (dos puntos cercanos en el espacio latente deberían dar contenidos similares una vez reconstruidos) y completo (un punto reconstruido desde el espacio latente debería dar un contenido significativo).

Se utilizará la pérdida sobre el conjunto de test para evaluar el rendimiento del modelo. En este primer experimento los resultados conseguidos con la arquitectura mencionada y con estos parámetros, no son prometedores (como era de esperar). Se dispone del valor de la función de pérdida sobre el conjunto de test, que es de 0.5059, aunque por el momento no es un valor significativo porque no se dispone de la misma métrica de otros modelos con los que se pueda comparar.

Otra forma de evaluar el modelo es observando el comportamiento del encoder, mediante la visualización del espacio latente y la del decoder mediante la reconstrucción de los datos de entrada. En la Figura 3.7 y en la Figura 3.8 se observa que para este experimento el espacio

latente no es coherente porque hay datos con diferentes parámetros que se solapan en un espacio muy reducido y la reconstrucción de los datos no es del todo eficiente.

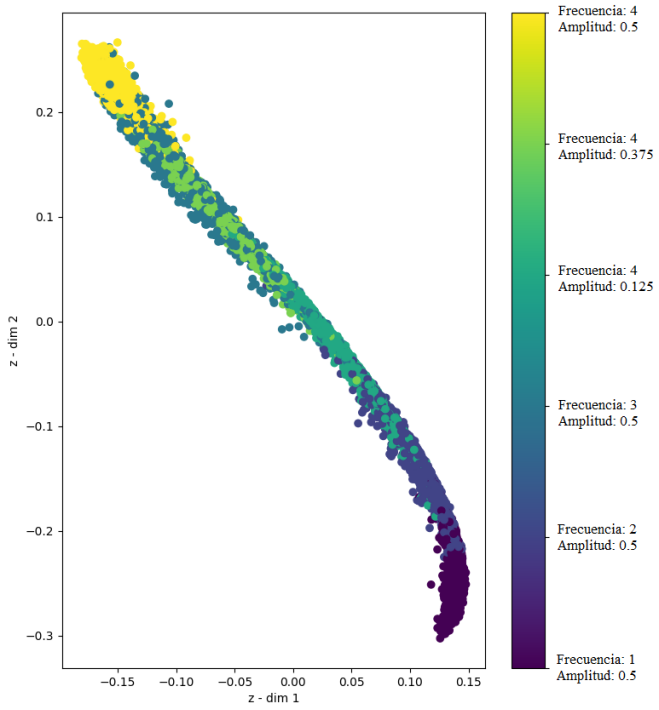


Figura 3.7.- Espacio latente con una arquitectura con capas FC.

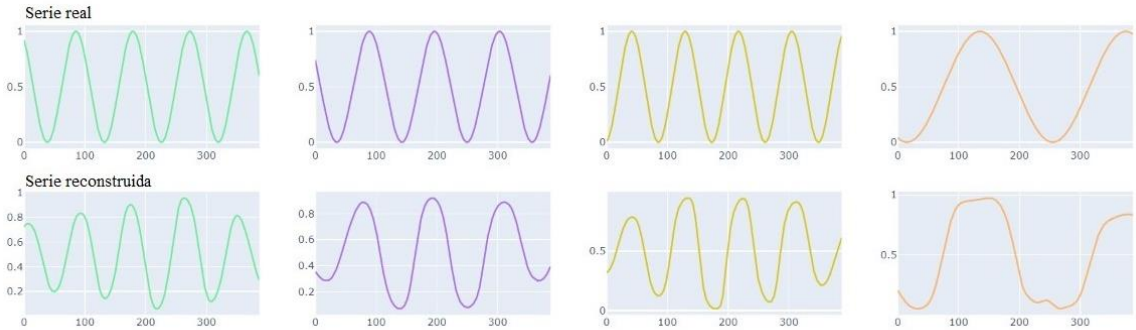


Figura 3.8.- Reconstrucción de los datos de entrada con una arquitectura con capas FC.

Uno de los principales motivos por los que el rendimiento de un modelo puede no ser eficiente es una mala elección de hiperparámetros. Los tiempos de entrenamiento de una red neuronal pueden ser muy elevados llegando incluso a necesitar varios días. Sabiendo esto, se evidencia que el ajuste de hiperparámetros es una tarea muy tediosa debido a que hay que esperar a que termine el entrenamiento para conocer si la configuración de hiperparámetros

elegida fue la correcta y si no es el caso, hay que modificarlos y volver a lanzar el experimento. Para evitar este proceso existen librerías que lo hacen de forma más eficiente. En ese proyecto se utilizó la librería Hyperopt [57], que permite explorar el espacio de búsqueda de los hiperparámetros para encontrar la configuración que mejor se ajuste al modelo. Utiliza internamente un algoritmo conocido como Tree Parzen Estimator (TPE) que utiliza el razonamiento bayesiano para construir un modelo subrogado para elegir los mejores hiperparámetros con menor coste temporal y computacional.

Haciendo uso de esta librería para este experimento se consigue reducir la pérdida en test a 0.4494, utilizando los siguientes hiperparámetros:

Hiperparámetros	
Dimensión intermedia	685
Batch Size	32
Nº de epochs	250
Optimizador	Adam (algoritmo similar a RMSprop que incluye en su ecuación un par de parámetros más)

Tabla 3.2.- Hiperparámetros obtenidos en la primera aproximación utilizando Hyperopt.

Sin embargo, la mejora no es lo suficientemente notable como para considerarlo buena solución (Figura 3.9). A pesar de que las clases se diferencian mejor, la coherencia en el espacio latente no se mantiene y la distribución de los datos no es equitativa, toma una forma alargada que da pie a confusión ya que salvo la clase amarilla y la clase violeta las demás clases se superponen resultando imposible distinguir con claridad la diferencia entre ellas.

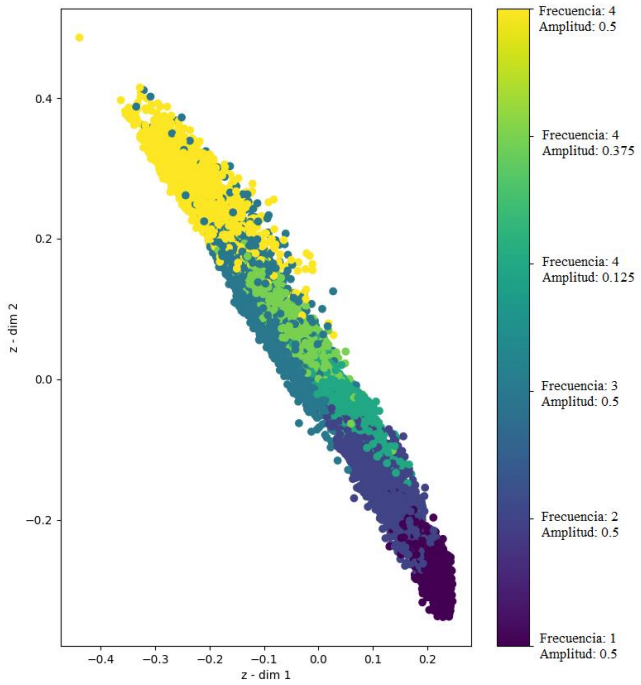


Figura 3.9.- Espacio latente con arquitectura FC optimizando hiperparámetros.

Segunda aproximación

Tras comprobar que una arquitectura no recurrente no modela correctamente el comportamiento temporal de los datos, se enfrenta esta segunda aproximación haciendo uso de las redes introducidas en 2.1.1.-. Ahora tanto el encoder como el decoder son redes LSTM. Además, también se introducen algunos cambios en el proceso de entrenamiento, aparte de Hyperopt para la optimización de hiperparámetros. Antes se utilizaba un número fijo de epochs, lo que suponía que cuando se alcanzara el número prefijado, el entrenamiento terminaba, siendo este por tanto el criterio de parada. Esto puede conducir a interpretaciones erróneas por suspender el entrenamiento demasiado pronto, cuando el modelo tiene aún margen para aprender, o demasiado tarde, pudiendo provocar overfitting. Teniendo esto en cuenta es más acertado establecer otro tipo de criterio para terminar el entrenamiento. En este caso se implementa un callback para que el entrenamiento finalice cuando, después de encontrar el mejor modelo en el entrenamiento (aquel que mejore la pérdida sobre el conjunto de validación), pasa un determinado número de epochs sin que se encuentre un modelo mejor. Así, se conocerá el modelo óptimo que se puede obtener con la configuración con la que se lanza el experimento. Además, según avanza el entrenamiento se va guardando

el mejor modelo hasta el momento en lugar de guardar el modelo obtenido tras finalizar el entrenamiento.

La evaluación en test disminuye a 0.1042, lo que resulta significativo teniendo en cuenta que con la anterior aproximación la métrica obtenida había sido de 0.4494. En la Figura 3.10.- se presenta el espacio latente. Los 3 grupos más externos pertenecen a ejemplos que tienen la misma amplitud: 0.5, pero diferente frecuencia: 1, 2 y 3 de más externo a más interno. Los 3 grupos más internos tienen la misma frecuencia: 4, pero diferente amplitud: 0.125, 0.25 y 0.375 de más interno a más externo. Esta disposición muestra que las clases más similares son adyacentes en el mapa, lo que puede permitir que un nuevo punto se ubique en el área del mapa que mejor se ajuste a sus parámetros, que es precisamente lo que se busca. En la Figura 3.11 se aprecia que la reconstrucción de los datos es mucho mejor que antes, sin embargo, también se refleja que las reconstrucciones no son completamente fieles.

La mejoría con esta aproximación es notable, aunque a priori parece limitada. Para comprobarlo se repite la misma experimentación, pero con el conjunto de datos final que representa los datos intracardíacos. Se utilizan seis clases que se generan variando los parámetros principales del modelo de Markov (explicado en 2.1.3.-): α y $\lambda_{NA.}$ α mide la velocidad de progresión de la FA y se utiliza el inverso de λ_{NA} para modelar el tiempo medio (medido en días) entre dos episodios de FA. Los valores de α que se utilizan son 999, que denota una progresión lenta y 998, que denota una progresión rápida. Para referirse al inverso de λ_{NA} se utilizará la denominación β por simplicidad y los valores elegidos son 10, 30 y 180 días.

El valor de la pérdida en validación es de 0.4250. En la Figura 3.12 se muestra el espacio latente, en donde la coherencia en la proximidad de los datos existe, pero el solapamiento de nuevo provoca que no se pueda diferenciar con claridad los diferentes grupos de arritmias. En la Figura 3.13 está la reconstrucción de los datos de entrada y, aunque se capta la tendencia de la serie temporal, la reconstrucción no se puede considerar buena.

Los mejores hiperparámetros encontrados fueron los siguientes:

Hiperparámetros	
Dimensión intermedia	425
Batch Size	64
Optimizador	Adam

Tabla 3.3.- Hiperparámetros obtenidos en la segunda aproximación utilizando Hyperopt.

Nótese que el número de epochs ya no se incluye como hiperparámetro porque a priori no se conoce cuántas se van a necesitar, sino que se relega a la red la cantidad de iteraciones que necesite para encontrar el modelo óptimo.

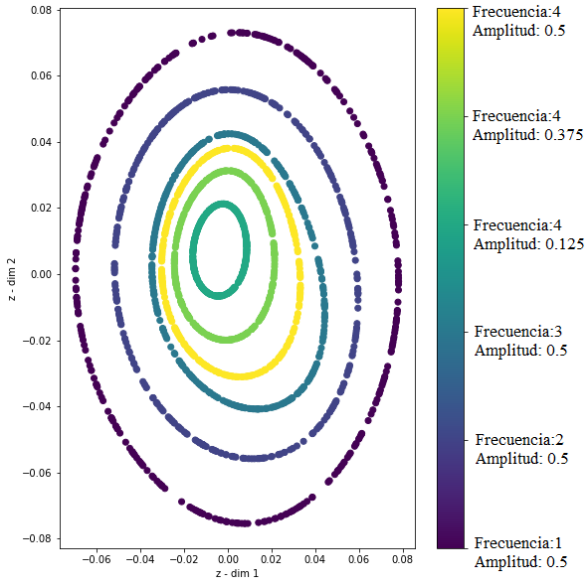


Figura 3.10.- Espacio latente resultante con una arquitectura basada en redes LSTM para datos senoidales.

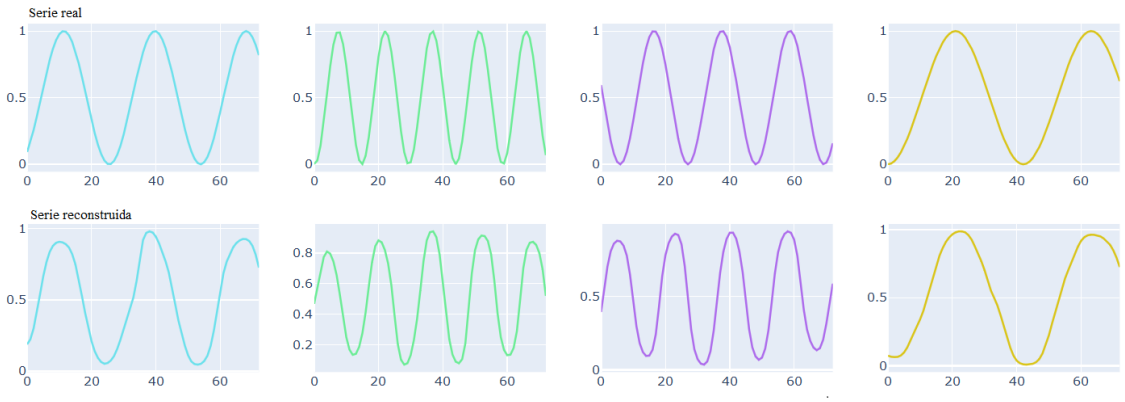


Figura 3.11.- Reconstrucción de los datos de entrada con una arquitectura basada en redes LSTM para datos senoidales.

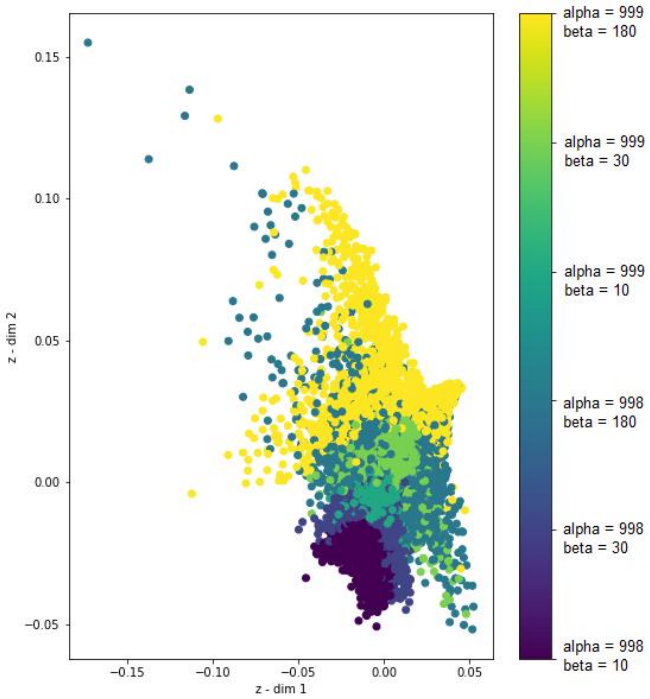


Figura 3.12.- Espacio latente resultante con una arquitectura basada en redes LSTM para datos intracardíacos.

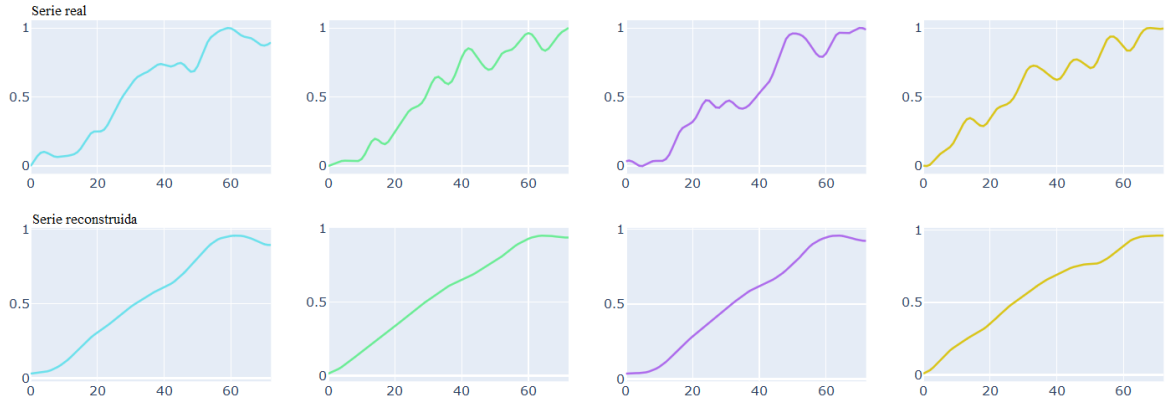


Figura 3.13.- Reconstrucción de los datos de entrada con una arquitectura basada en redes LSTM para datos intracardíacos.

Tercera aproximación

Desde que se comenzó a entrenar el modelo se ha ido mejorando el entrenamiento modificando la arquitectura, optimizando los hiperparámetros, cambiando el criterio de parada… No obstante, la solución óptima aún no se ha encontrado.

Las redes LSTM analizan los datos de atrás hacia adelante preservando la información del pasado a través de los estados ocultos. Por otra parte, también es posible preservar la información del futuro, tratando los datos de entrada de adelante hacia atrás. Precisamente este es el principio de funcionamiento de las redes LSTM Bidireccionales (Bidirectional LSTM). Hacen un recorrido en dos direcciones, primero de pasado a futuro y luego de futuro a pasado, conservando información de ambos períodos.

Las redes LSTM son muy utilizadas también en el reconocimiento del lenguaje natural (Natural Language Processing). Por simplicidad se presenta un ejemplo de Bidirectional LSTM aplicado a este campo: Supongamos que se quiere predecir la palabra siguiente en la frase "Ciencia es creer en …". Una red LSTM simple predecirá la siguiente palabra conociendo sólo este contexto, sin embargo, el contexto de una red LSTM Bidireccional es más amplio porque también ha visto en la pasada de adelante hacia atrás cómo seguía la frase: "… en la ignorancia de los científicos". De esta manera, utilizar información del futuro puede ayudar a la red comprender qué tipo de información predecir.

Hay otra modificación que se puede añadir en el entrenamiento del modelo. Hasta el momento se han utilizado LRs adaptativos, factor que mejora el entrenamiento, pero si no se escoge un punto de partida bueno, el entrenamiento puede tardar excesivamente en converger. Para evitar iteraciones con saltos innecesarios hacia la búsqueda del mínimo de la función se utilizan Learning Rates Cíclicos [58]. En la Figura 3.14 se reflejan los efectos derivados de la elección del LR: si el LR es muy pequeño (parte de la izquierda) la búsqueda del mínimo de la función requiere muchas actualizaciones, lo que significa un mayor tiempo de entrenamiento. Por el contrario, si es muy grande (parte de la derecha), las actualizaciones de la red son muy drásticas, pudiendo ocasionar saltos demasiado largos que nunca encuentran el mínimo. Lo ideal es ir dando los pasos adecuados en el momento oportuno como la figura del centro.

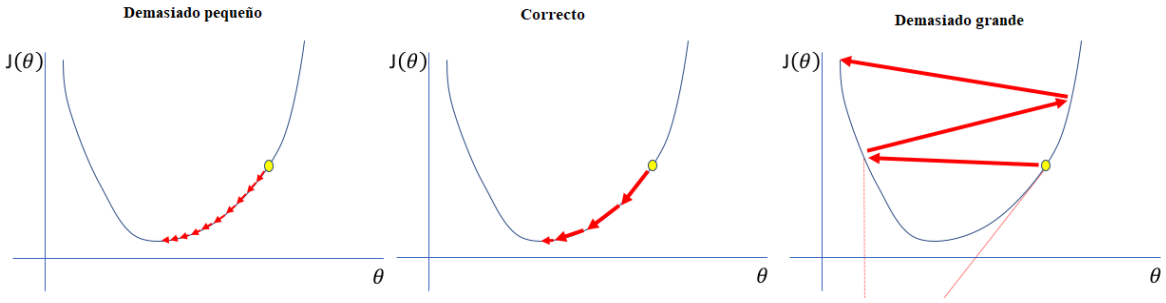


Figura 3.14.- Tipos de LR (fuente: https://www.jeremyjordan.me/nn-learning-rate/).

Para conseguir esto se utilizan los Learning Rates Cíclicos. La idea es la siguiente: se definen unos límites para el LR, uno muy pequeño y otro muy grande. Se comienza entrenando la red con el límite inferior y a medida que se actualizan los pesos al final del entrenamiento en cada batch se incrementa el LR. Al terminar el entrenamiento se puede obtener una gráfica como la de la Figura 3.15. En ella, se puede ver cuáles son los valores óptimos del LR para el problema que se quiere abordar. La evolución de la gráfica debería ser similar para cualquier problema: cuando se aplican LR bajos (parte izquierda de la gráfica), los pasos hacia el mínimo de la función son pequeños y la pérdida disminuye a un ritmo lento, pero cuando se aplica un LR óptimo (parte central de la gráfica), se observa una rápida caída en la función de pérdida, y por último, si se aumenta demasiado el LR (parte derecha de la gráfica), la pérdida volverá a subir. El interés radica en la pendiente de la gráfica porque conociendo los valores que la delimitan se podrá saber con qué LR comenzar el aprendizaje. Por ejemplo, véase la gráfica obtenida para nuestro problema en la Figura 3.16. La pendiente se encuentra entre los valores de 10e-5 y 10e-4, por tanto, se tendrá que comenzar el entrenamiento con un LR entorno a 10e-4, al principio grande para avanzar rápidamente hacia el mínimo y después, una vez cerca del mínimo, avanzar con pasos más pequeños (recordar imagen central de la Figura 3.14), es decir, con valores del LR en torno a 10e-5.

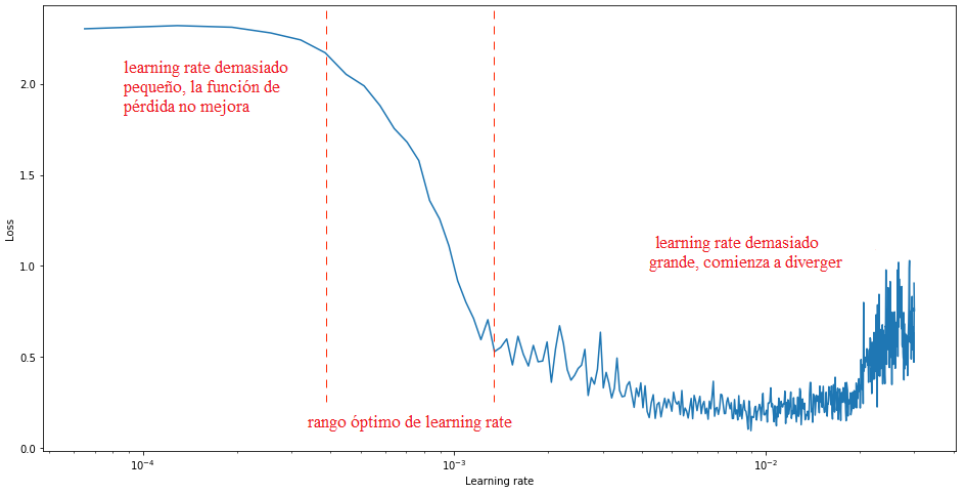


Figura 3.15.- Análisis del error en función del LR.

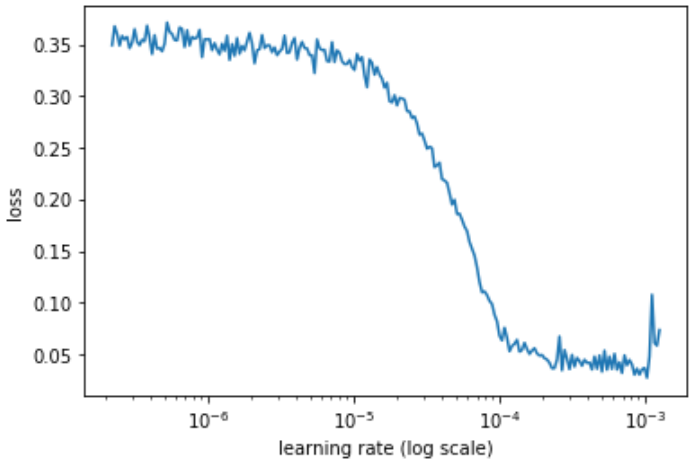


Figura 3.16.- Análisis del LR para nuestro problema.

Para este experimento se sustituyen las redes LSTM anteriores por Bidirectional LSTM y se añade también la búsqueda del LR óptimo para los datos intracardíacos. Los hiperparámetros finales son los siguientes:

Hiperparámetros	
Dimensión intermedia	300
Batch Size	128
Optimizador	Adam comenzando con un valor de LR de 10e-4

Tabla 3.4.- Hiperparámetros obtenidos en la tercera aproximación utilizando Hyperopt y LR Cíclicos.

Los resultados son: 0.0136 de pérdida en validación, como espacio latente la Figura 3.17 y como reconstrucción la Figura 3.18. Poniendo en perspectiva el valor de pérdida, se ha conseguido reducirlo desde 0.4250 a 0.0136. El espacio latente ahora tiene unas propiedades que se asemejan más al objetivo prefijado, los datos similares están próximos y evidencian más claridad en la interpretación de los distintos grupos de arritmia. La reconstrucción no es del todo perfecta ni se podría considerar buena. Por una parte, se puede deber a la naturaleza de los datos, ya que, aunque las tendencias se capturan bien, las variaciones en la serie temporal son difíciles de capturar, a diferencia que en datos más sencillos como se puede ver en la Figura 3.19, donde se muestra una reconstrucción perfecta para datos senoidales, también con el mismo modelo. Por otra parte, y como ya se comentó en el apartado 2.2.1.- el objetivo de este modelo no es la generación de nuevos tipos de datos. A pesar de que se utilizara la reconstrucción como indicador de rendimiento del modelo, esto sirvió como soporte visual adicional en las primeras fases, pero ahora lo principal es la obtención de un espacio latente coherente, que en este caso es bastante prometedor y será lo que se priorizará.

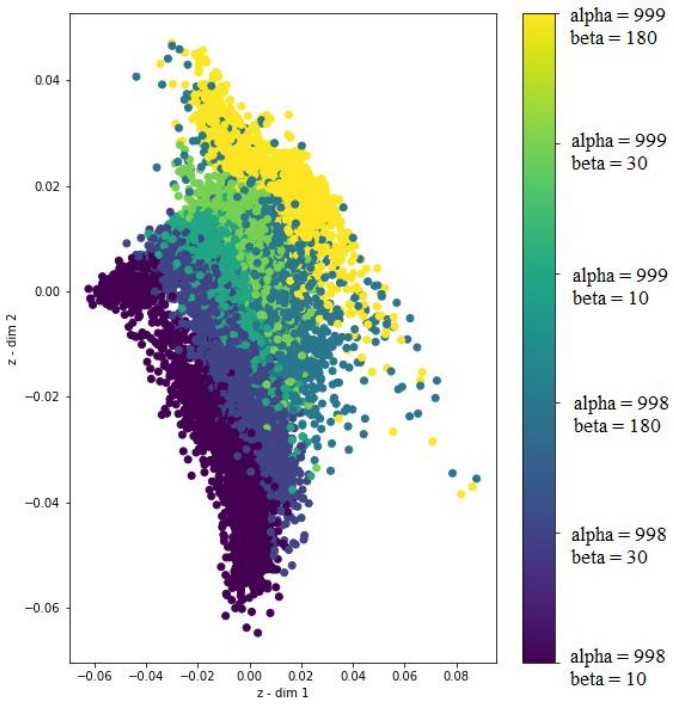


Figura 3.17.- Espacio latente resultante con la tercera aproximación para datos intracardíacos.

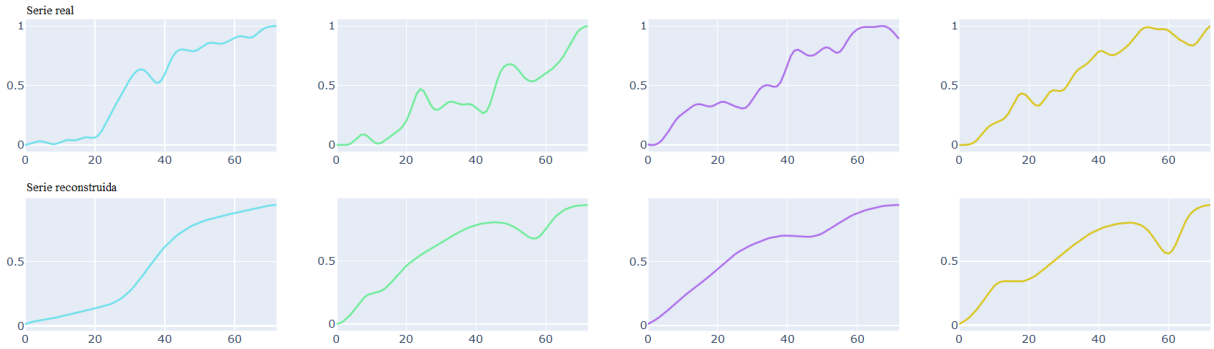


Figura 3.18.- Reconstrucción de los datos de entrada con la tercera aproximación para datos intracardíacos.

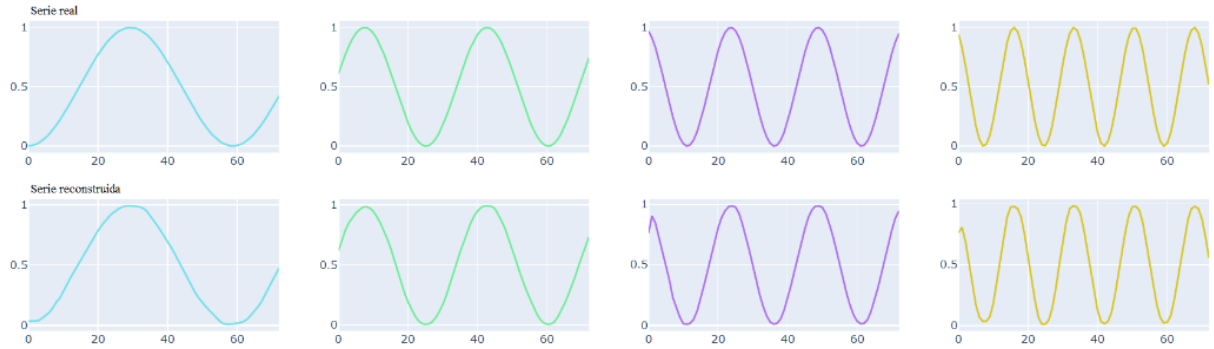


Figura 3.19.- Reconstrucción de los datos de entrada con la tercera aproximación para datos senoidales.

Aproximación final

En la tercera aproximación nos hemos acercado ampliamente a la solución que se busca, pero es necesario recordar que el framework que se propone en este trabajo está compuesto además de por un VAE, por un clasificador. Hasta ahora, en lo experimentos anteriores se había dejado de lado la implementación del clasificador, pero se verá que su incorporación al modelo es totalmente necesaria.

La mayoría de soluciones que combinan VAEs con clasificadores lo hacen construyendo un clasificador sobre el VAE ya entrenado. Para estudiar esta alternativa se construyó un clasificador simple sobre los pesos congelados del encoder. La arquitectura utilizada está constituida por una capa completamente conectada (Fully Conected) y por una capa softmax para determinar sobre la salida de la anterior capa a qué clase pertenece cada muestra. El

rendimiento conseguido, con un porcentaje del 72.67% de precisión sobre los datos de test, se considera bastante exiguo. Se puede percibir en la Figura 3.17 que esto seguramente se deba a que el clasificador es incapaz de distinguir algunas arritmias que pertenecen a otras clases similares. En la parte central de la figura hay tres clases cuyos puntos se solapan, por tanto, la limitación no está en el clasificador, ya que es una tarea que escapa de sus límites, por lo que hay que buscar otro enfoque.

En esta última aproximación se incluye el clasificador al entrenamiento para optimizar también el porcentaje de clasificación. En un primer experimento, haciendo uso de todos los métodos que se han ido citando a lo largo de este apartado, el resultado conseguido es de un 83.35% de precisión en test. Este porcentaje, aunque significativamente mayor al anterior, sigue siendo insuficiente. Si se atiende a la evolución del entrenamiento hay una particularidad a tener en cuenta: ahora se están optimizando 2 objetivos, los del VAE y el de clasificación. En la evolución de los errores se estima que las funciones de pérdida están en diferentes escalas y esto ocasiona que la optimización de la red priorice la disminución de un error sobre otro. La solución para evitar esto es asignar diferentes pesos a cada error para que el proceso de optimización se pueda regular otorgando la misma importancia a los dos objetivos. En la Figura 3.20 se puede ver como las pérdidas del clasificador y del VAE, gráficas de la izquierda y del centro respectivamente, disminuyen a la par y, por tanto, la pérdida final (combinación de ambas), la gráfica de la derecha, también.

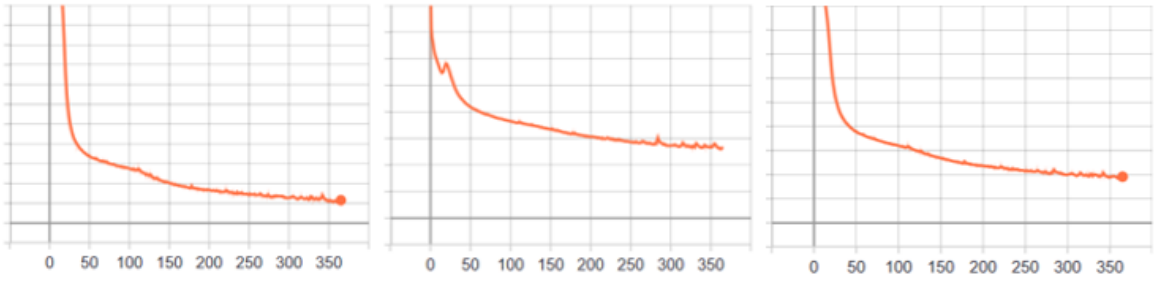


Figura 3.20.- Evolución de las funciones de error.

El conjunto de hiperparámetros utilizado se mantiene respecto al de la Tabla 3.4. También es destacable mencionar que se asigna 10 veces más peso al objetivo del clasificador para que la optimización no priorice los objetivos del VAE. El mejor modelo se encuentra en la epoch 356.

Con este último entrenamiento el porcentaje de clasificación asciende hasta 97.11% y se obtiene un nuevo espacio latente (Figura 3.21). A primera vista, la diferencia respecto a los anteriores espacios latentes expuestos es indudable. Las seis clases con las que se entrena el modelo se representan de forma clara, distinguiéndose seis grupos distintos en los que el solapamiento es minúsculo y la coherencia se sigue manteniendo, donde clases con parámetros similares se sitúan en zonas próximas. Con un espacio latente de estas características se reúnen todas las condiciones para elaborar un mapa de diagnóstico cuando se reciba una nueva arritmia de un paciente real.

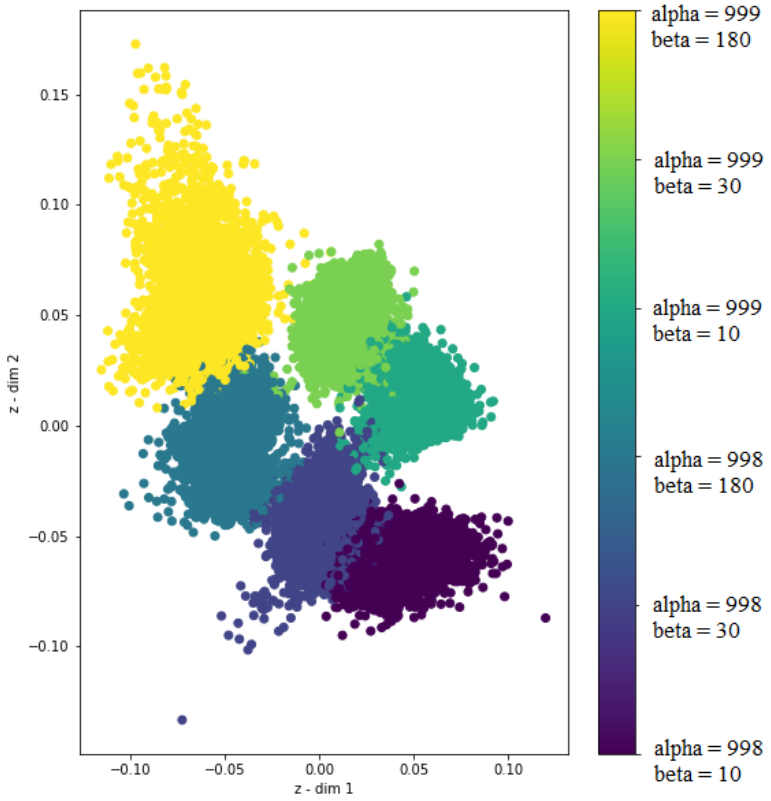


Figura 3.21.- Espacio latente conseguido con la solución final.

En esta sección se pone en perspectiva los resultados obtenidos comparando el rendimiento del framework propuesto con otras técnicas recientes del estado del arte.

3.3.- COMPARACIÓN CON OTROS MÉTODOS

En el apartado anterior la explicación se ha focalizado sobre la optimización del VAE para conseguir la solución que mejor se ajuste a los datos. En este punto las tareas planteadas se han alcanzado, pero se requiere elaborar una evaluación de la calidad de la solución para determinar si los objetivos establecidos se han logrado satisfactoriamente.

La meta final es el diagnóstico de la situación del paciente y en este proyecto se aborda principalmente mediante la proyección latente de los datos intracardíacos. Se podría establecer una discusión acerca de la capacidad del modelo en este sentido, pero no sería justa porque no se argumenta con una comparativa que demuestre que el método propuesto es mejor que otras técnicas ya existentes que se podrían haber usado, y tampoco es una tarea trivial. No existen métricas consensuadas acerca de cómo confrontar métodos gráficos, por eso entra de nuevo en juego la importancia del clasificador. Como se constató en los objetivos del trabajo, tanto la proyección latente como el clasificador proporcionan un diagnóstico y el último caso se puede emplear para equiparar el rendimiento de la solución propuesta a otras soluciones.

En [59] se elabora una revisión del estado actual (hasta 2019) de los algoritmos de Deep Learning para la clasificación de series temporales y se proporciona una herramienta de

código abierto disponible en (https://github.com/hfawaz/dl-4-tsc) que se utilizará para realizar la comparativa con los mejores métodos que se citan en el estudio, concretamente los siguientes:

- Resnet: una red profunda propuesta por [60] compuesta por tres bloques residuales seguidos de una capa GAP y una capa softmax cuyo número de neuronas es igual al número de clases del conjunto de datos.
- FCN: una red neuronal completamente convolucional (Fully Convolutional Neural Network) elaborada con la misma arquitectura propuesta en [60], que consiste en tres bloques convolucionales cuyo resultado se promedia sobre toda la dimensión temporal que corresponde a la capa GAP. A la salida de esta capa se conecta un clasificador softmax.
- Encoder: Originalmente propuesto por [61], Encoder es una CNN híbrida profunda cuya arquitectura se inspira en FCN, con una diferencia principal donde la capa GAP es reemplazada por una capa de atención [62].
- TWIESN: Time Warping Invariant Echo State Network es una variante de la Red de Estados de Eco (ESN) propuesta por [63] en la que cada paso de tiempo se proyecta en un espacio cuyas dimensiones se infieren del tamaño de un depósito. Después, para cada elemento, se entrena un clasificador Ridge [64] para predecir la clase de cada elemento en la serie temporal.

Cabe destacar que en la sección 2.1.1.- se resaltó la importancia de las RNN para tratar series temporales mientras que en este estudio solo se presenta TWIESN como única RNN a comparar. Las RNN se aplican generalmente para la predicción en la evolución de las series temporales, sin embargo, en el caso particular de la clasificación es diferente debido a los siguientes motivos:

- Este tipo de arquitecturas está diseñado principalmente para predecir una salida para cada elemento en la serie temporal [65].
- Las RNN suelen sufrir el problema del Vanishing Gradient debido al entrenamiento en series temporales largas [66].
- Las RNN se consideran difíciles de entrenar y paralelizar, lo que lleva a evitar utilizarlas por razones computacionales [67].

En el caso de nuestra solución el principal objetivo no es el de la clasificación si no el del tratamiento de la evolución en las series que representan las arritmias, por eso no se considera la aplicación de otras arquitecturas.

En cuanto a las experimentaciones llevadas a cabo para estas comparativas, debido a las restricciones de cómputo de Google Colaboratory, se hicieron en una GPU NVIDIA Tesla T4 del departamento de Metrología y Modelos.

La Tabla 3.5 recoge el rendimiento de los diferentes modelos para cada clase en términos de precisión. Cada entrada de la tabla es el número de veces que una arritmia de una clase fue reconocida por cada modelo por la clase adecuada. Además, para ilustrar el rendimiento de cada método se añade la clasificación calculada por el método de Friedman (clasificación por rango) para cada conjunto de datos y la clasificación resultante promediada.

Se observa que el mejor clasificador es Resnet, seguido por nuestra solución, etiquetada como RVAE. Para ampliar la comparativa entre los diferentes métodos se procede a realizar pruebas post-hoc para detectar diferencias significativas por pares entre todos los clasificadores según se aconseja en [68]. La Tabla 3.6 muestra la familia de hipótesis formuladas para comparar los clasificadores ordenadas por los p-valores correspondientes. Si la prueba de significación arroja un p-valor menor que un umbral predefinido (generalmente 0.05), entonces la diferencia se considera significativa, por lo tanto, un modelo se declara superior a otro. En este caso únicamente Resnet es significativamente superior a otros modelos, que son FCN, Encoder y TWIESN si se considera un nivel de significación del 0.05 ya que los p-valores están por debajo de este umbral. La única solución a la que no supera significativamente es la nuestra. Si se considera la corrección de Bonferroni, en la que se tienen en cuenta la cantidad de comparaciones, el umbral que se tendría que fijar es 0.05 dividido entre el número de comparaciones, es decir 0.05/6 = 0.0083. Tomando este valor, Resnet solo sería significativamente superior a TWIESN. Este dato es importante destacarlo porque solo TWIESN y nuestra solución emplean RNNs, por lo que se podría deducir que nuestra solución supera a la mejor RNN del estado del arte en términos de clasificación.

Como conclusión de este estudio comparativo, se puede afirmar que el modelo final desarrollado es capaz de competir con los mejores clasificadores para series temporales reportados hasta el año 2019.

	Resnet	FCN	Encoder	TWIESN	RVAE
998na10	0.9681(2)	0.9867 (1)	0.9361(5)	0.9517(4)	0.9543(3)
998na30	0.9664(1)	0.8788(5)	0.9456(3)	0.9438(4)	0.9553(2)
998na180	0.9846(1)	0.9719(4)	0.9729(3)	0.9611(5)	0.9779(2)
999na10	0.9849(1)	0.9849(2)	0.9364(5)	0.9505(4)	0.9770(3)
999na30	0.9879(1)	0.9778(3)	0.9826(2)	0.9791(4)	0.9733(5)
999na180	0.9904(1)	0.9886(4)	0.9895(3)	0.9786(5)	0.9895(2)
Resumen de los resultados					
Precisión	0.9803	0.9647	0.9603	0.9607	0.9712
Rango promedio	1.166	3.166	3.500	4.333	2.833

Tabla 3.5.- Precisión de los distintos clasificadores para los 6 tipos de FA.

i	hipótesis	$Z = (R_0 - R_i)/SE$	p
1	Resnet vs TWIESN	3.465	0.0005
2	Resnet vs Encoder	2.556	0.0106
3	Resnet vs FCN	2.191	0.0285
4	Resnet vs RVAE	1.826	0.0679
5	RVAE vs TWIESN	1.640	0.1010
6	FCN vs TWIESN	1.275	0.2023
7	Encoder vs TWIESN	0.912	0.3618
8	RVAE vs Encoder	0.731	0.4648
9	FCN vs Encoder	0.366	0.7144
10	FCN vs RVAE	0.365	0.7151

Tabla 3.6.- Familia de hipótesis ordenadas por el p-valor.

En esta última sección se aborda el diagnóstico de la situación del paciente mediante la proyección visual latente.

3.4.- PROYECCIÓN DE LA SITUACIÓN DEL PACIENTE

Hasta esta sección se ha elaborado una revisión del recorrido del proyecto para superar los objetivos del trabajo, sin embargo, la meta principal, el diagnóstico de un individuo que padece de FA, aún no se ha cubierto.

En el entrenamiento del modelo propuesto se priorizaron las propiedades latentes del VAE para obtener un espacio cuyas características fueran adecuadas para una representación simplificada de los datos. La Figura 3.21 es la materialización final de las representaciones latentes obtenidas por el encoder para los datos de entrenamiento que puede entenderse como una proyección de los 5 parámetros que rigen el modelo que simula las arritmias. Se distinguen seis agrupaciones que corresponden con las seis clases con las que se entrenó el VAE etiquetadas según los 2 parámetros más relevantes de las arritmias simuladas: α y β. Las representaciones se organizan acorde a la criticidad de estos dos parámetros.

El interés clínico reside en poder proyectar los datos de un paciente real sobre el espacio latente para conocer a qué parámetros del modelo se ajustan más. En primera instancia se validará este comportamiento con datos simulados por el modelo de Markov para ilustrar el funcionamiento del método y después se repetirá el mismo proceso con registros intracardíacos de pacientes reales.

Para el caso de las arritmias sintéticas se eligen dos instancias aleatorias del conjunto de test: una de ellas tiene parámetros $\alpha = 0.998$ y $\beta = 1/\lambda_{NA} = 30$ y la otra $\alpha = 0.999$ y $\beta = 1/\lambda_{NA} = 10$. El procedimiento es bastante intuitivo, estas arritmias se pasan al encoder, que predice una salida que será la media y varianza de cada una de las arritmias adaptadas a la distribución que aprendió durante el entrenamiento. Estos dos parámetros son los ejes que rigen el espacio latente, por tanto, su codificación en este espacio, al igual que el conjunto de entrenamiento, corresponde a un punto con coordenadas x=media e y=varianza. Al proyectar estas dimensiones sobre el mapa aprendido, su localización sobre los grupos de arritmias que hay indicará si la solución final es capaz de ubicar datos nunca antes vistos en la clase correspondiente. La Figura 3.22 es el resultado de este procedimiento: La parte de la izquierda se corresponde a la proyección de la primera arritmia (punto rojo), con parámetros $\alpha = 0.998$ y $\beta = 30$, que justo se sitúa en el grupo correspondiente a las arritmias del conjunto de entrenamiento que tienen los mismos parámetros. La parte de la derecha es la proyección de la arritmia con parámetros $\alpha = 0.999$ y $\beta = 10$, de nuevo situada en el grupo al que le corresponde.

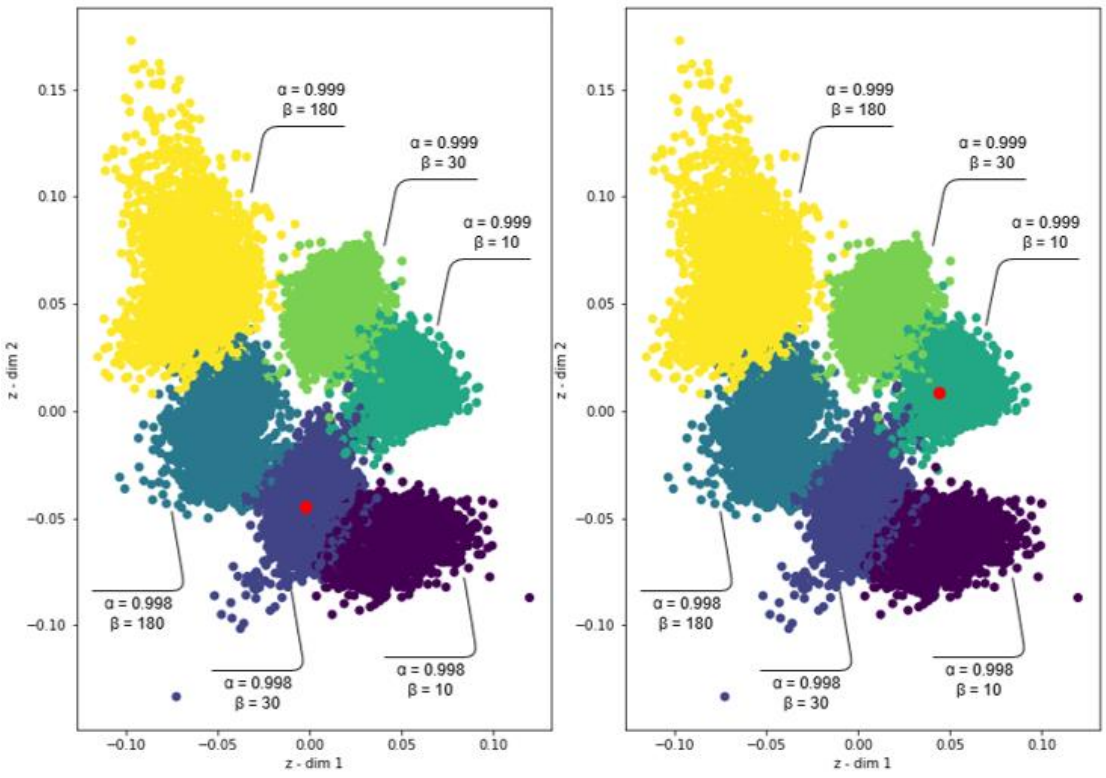


Figura 3.22.- Proyección de secuencias de eventos AMS generados con el modelo de Markov.

Acorde con estas estimaciones, es inmediato que cuando se presente al VAE datos de un paciente real, su localización en el mapa nos otorgará una visión de los parámetros más probables que se adapten a su situación. En la Figura 3.23 se ofrece una proyección de dos pacientes elegidos al azar. Los datos pertenecen a marcapasos, que tuvieron que procesarse a partir de datos crudos para que se pudieran analizar con el VAE.

En la parte de la izquierda se observa que la proyección del paciente cae en el grupo perteneciente a arritmias que tienen parámetros $\alpha = 0.999$ y $\beta = 180$. Se recuerda que el parámetro α mide la velocidad de progresión de las arritmias y que valores de α cercanos a 0.999 indican una progresión lenta. β indica el tiempo medio entre arritmias, en este caso es más probable que las de este paciente tengan lugar, al menos cada 180 días, por lo que se estima que es un paciente que evoluciona positivamente sin entrañar mucho riesgo.

Tal y como está organizado el mapa se evidencia que los valores de β se localizan de izquierda a derecha de mayor a menor (180, 30, 10), lo que es igual a una organización de menor a mayor criticidad ya que valores bajos de β indican tiempos breves entre arritmias. Por otro lado, los valores de α se organizan de arriba abajo (999, 998), de menor a mayor criticidad. Se puede hacer uso de esta información para elaborar una mejor interpretación del mapa. La zona superior derecha denota las arritmias menos críticas, mientras que la zona inferior derecha manifiesta aquellas arritmias que suponen una etapa muy avanzada de la enfermedad. A la par, el resto de parámetros del modelo de Markov durante la generación del conjunto de entrenamiento se ha variado aleatoriamente, lo que influye ligeramente sobre la condición de las arritmias, por lo que esta propiedad puede dar pie a interpretar arritmias entre dos grupos como una interpolación entre los parámetros de dos clases. Esto se puede entender mediante el caso del paciente de la parte derecha. El segundo caso expuesto se localiza en el grupo de arritmias con parámetros $\alpha = 0.998$ y $\beta = 30$. En primer lugar, el parámetro β se acerca más a 30, pero debido a su cercanía al grupo inferior izquierdo ($\beta = 180$) se puede entender que su evolución está en camino a llegar a 30 y en segundo lugar, el parámetro más crítico, α, se corresponde con un valor de 0.998, lo que significa que la evolución se acerca hacia una arritmia permanente. No es el caso más crítico, pero sí que puede suponer un caso que necesite intervención médica para evitar una futura complicación.

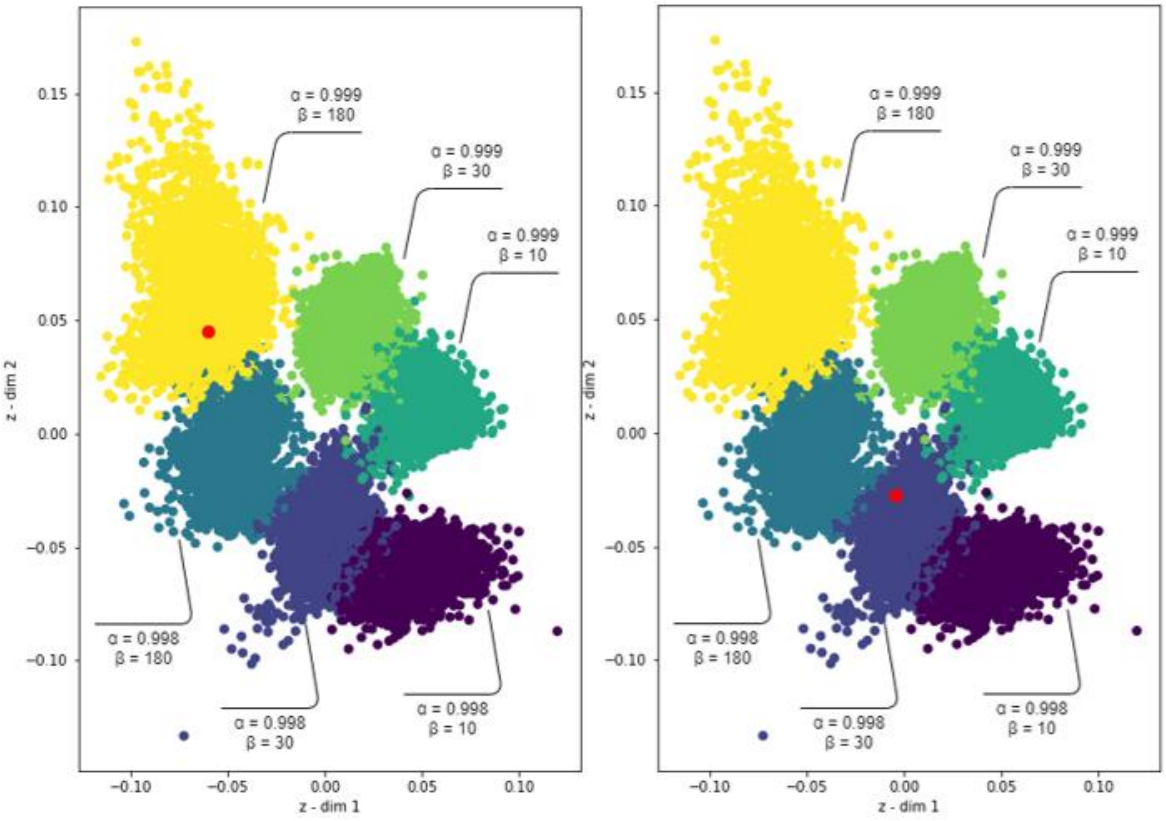


Figura 3.23.- Proyección de secuencias de eventos AMS a partir de un marcapasos real.

4. Discusión de resultados

Expuestos los resultados, se puede discutir el rendimiento de la solución propuesta. En primer lugar, se presentó la relevancia del clasificador haciendo un estudio en el que se equiparaba el rendimiento de varios clasificadores del estado del arte. Se demostró que nuestro método es capaz de competir con soluciones dedicadas exclusivamente a la clasificación de series temporales, superando a tres de los cuatro métodos presentados en términos de porcentaje de aciertos. Por otra parte, la sensibilidad del clasificador es esencial para desarrollar un modelo confiable. Los errores en la predicción de la clase a la que pertenece cada arritmia se corresponden con arritmias que por sus características se sitúan entre clases similares a la que pertenece realmente por lo que estos fallos se pueden

interpretar como una estimación de la clase de arritmia que más se asemeja a sus parámetros o incluso como la posible evolución futura que tendrán.

En segundo lugar, se presentó el mapa de diagnóstico utilizado para juzgar de forma visual la situación de un paciente. La organización del espacio latente demuestra que el modelo es capaz de diferenciar bien los distintos valores de α y β, permitiendo conocer si el estado de una arritmia determinada avanza peligrosamente hacia FA permanente, siendo esta la finalidad principal de trabajo. Se validó primero la proyección con datos simulados con propiedades conocidas y posteriormente con datos intracardíacos obtenidos de marcapasos.

Es importante remarcar la organización latente obtenida y su interpretabilidad. Como se mencionó en el anterior apartado, las arritmias más peligrosas se sitúan en la parte inferior derecha y las que no evidencian demasiado peligro en la parte superior izquierda. Esta evolución desde un extremo a otro en el mapa da pie a interpretar el espacio latente como una interpolación de los parámetros utilizados para ofrecer un diagnóstico: α en el eje Y y β en el eje X. La Figura 4.1 refuerza esta idea: se proyectan sobre el espacio latente nuevas arritmias simuladas variando los parámetros del modelo de Markov, pero a diferencia del proceso seguido para generar los datos de entrenamiento donde los parámetros λ_{GA}, λ_{A} y p_{AG} se alteraron aletoriamente dentro de un rango determinado, en esta ocasión se dejaron fijos. Como resultado, se representan las instancias como cruces y se etiquetan en función del parámetro β con los que fueron generados. El parámetro α se omite ya que la pertenencia hacia 0.998 o a 0.999 es evidente. La proyección en el mapa demuestra que la interpretación de los parámetros de una arritmia determinada se puede establecer acorde con la cercanía a un cluster concreto, es decir, a pesar de que el primer grupo caracteriza aquellas arritmias con parámetros α=0.999 y β=10, si una arritmia se sitúa en el límite entre este grupo y el de su izquierda es muy probable que tenga un parámetro β intermedio entre ambos, por ejemplo, 20, o si una arritmia se situara entre un grupo superior y otro inferior querría decir que el parámetro α evoluciona peligrosamente hacia valores de 0.998. De esta manera, se puede conocer cómo podría evolucionar la progresión de una arritmia determinada.

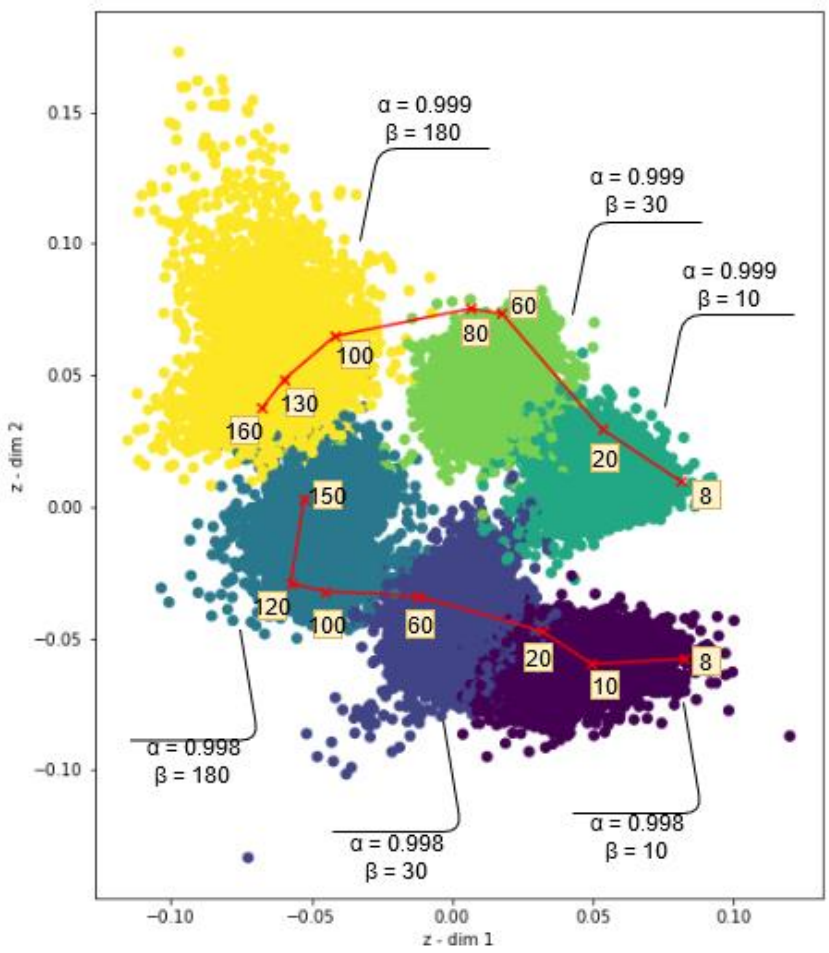


Figura 4.1.- Proyección de casos simulados con parámetros conocidos.

Para terminar, es importante remarcar que se ha conseguido poder establecer un diagnóstico fiel a la realidad disponiendo de una cantidad de registros limitada para cada paciente y esta es quizá la mayor dificultad del problema. Un diagnóstico temprano no sería compatible con un conjunto grande de episodios de FA, el cuál determinaría que un paciente está en una etapa muy avanzada de la enfermedad. El interés radica en poder deducir el estado de un paciente de forma que se pueda actuar de forma prematura y así evitar potenciales intervenciones futuras. Por esta razón, la valoración final de lo conseguido es muy positiva.

5. Conclusiones y trabajo futuro

Se ha demostrado que los registros intracardíacos de los marcapasos e ICDs se pueden utilizar para estimar el cambio de una FA paroxística a una permanente. La principal dificultad estriba en la corta longitud de los registros, que se ha abordado mediante una proyección gráfica de la secuencia de eventos registrados por los marcapasos. La proyección ha sido resultado del entrenamiento de un VAE con secuencias simuladas que representan a un rango amplio de arritmias que un paciente puede sufrir, proporcionando una herramienta de apoyo a los especialistas para el diagnóstico de la FA.

A la vista de lo conseguido se pueden cuestionar los límites del proyecto. A continuación, se cita una serie de metas que quedaron fuera del ámbito abarcado pero que pueden aportar una extensión útil al recorrido del proyecto:

Aspecto	Descripción
Explorar otras aplicaciones posibles del VAE	Utilización del decoder o de alguna variación del VAE para otras posibles aplicaciones.
Perfeccionamiento del modelo de Markov	Validación y perfeccionamiento del modelado dinámico de los eventos AMS.
Explotación del modelo para otros dominios	Utilización del modelo propuesto para otros problemas de similar naturaleza.

Explorar otras aplicaciones posibles del VAE

En primer lugar, hay que destacar que para la solución propuesta se ha descartado uno de los elementos del modelo, el decoder. A lo largo del entrenamiento se han utilizado sus reconstrucciones como métrica para evaluar el rendimiento del modelo, pero no forma parte activa de la solución. Se podría aprovechar el potencial de este elemento para la detección de anomalías en datos intracardíacos a partir de reconstrucciones de los datos o también para, a partir de los datos de un paciente, predecir los siguientes pasos de tiempo, es decir, predecir cómo evolucionará la arritmia en el tiempo futuro y combinar esta extrapolación con la estimación de α y β.

Por otra parte, los avances en el campo del Machine Learning llevan un ritmo extraordinariamente alto y es algo que puede aportar nuevas mejoras al proyecto. Desde la aparición del VAE han surgido otros modelos que proponen modificaciones sobre el VAE como β-VAE [69] que hace especial énfasis en el descubrimiento de factores latentes "desenredados", esto son representaciones latentes que son sensibles a un factor muy generalista. Un ejemplo sería que para un modelo entrenado sobre caras humanas fuera capaz de aprender detalles concretos tales como la longitud del pelo, emociones, el color de la piel u otras propiedades y esto podría ser beneficioso para explicar con más detalle las causas en la evolución de la enfermedad. También hay otros avances interesantes como VQ-VAE [70] que aprende representaciones discretas que pueden ser beneficiosas para problemas como el lenguaje, el habla o el razonamiento o TD-VAE [71], que trabaja con la misma idea pero aplicada a datos secuenciales.

Perfeccionamiento del modelo de Markov

El modelo de Markov se ha utilizado para modelar la secuencia de eventos AMS de un marcapasos. Al mismo tiempo que se producen avances en Machine Learning, también se producen avances en el perfeccionamiento de estos mecanismos y como tal, los cambios en los marcapasos pueden afectar al modelado de los registros intracardiacos. Un posible camino a seguir sería la elaboración de un modelo generalista que se pudiera adaptar a los mecanismos de registro de cualquier dispositivo invasivo y también sería interesante elaborar algún proceso de validación clínica que determinara la precisión con la que el modelo es capaz de reflejar la variedad de arritmias que un paciente puede sufrir.

Explotación del modelo para otros dominios

El marco propuesto constituye una solución personalizada para un tipo de problema muy concreto, pero como resultado se obtiene un modelo que puede ofrecer soluciones a problemas, no del mismo ámbito, pero sí de naturaleza similar. Se dispone de un modelo que, dados unos datos de entrada aprende una representación comprimida de estos y a la vez un clasificador es capaz de determinar las diferentes clases que hay en entre esas representaciones. Sería interesante aplicar la misma solución a otros problemas y de hecho no sería necesario repetir el proceso de entrenamiento completo desde el principio, se podría aplicar lo que se conoce como Transfer Learning, que consiste en transferir el conocimiento adquirido en este problema para aplicarlo a otro problema diferente utilizando parte de la

configuración del modelo final. De esta manera el entrenamiento que se necesitaría requeriría mucho menos esfuerzo.

6. Planificación y Presupuesto.

En esta sección se expone la planificación temporal seguida para llevar a cabo el proyecto y un resumen del presupuesto estimado.

6.1.- PLANIFICACIÓN

La planificación del proyecto se dividió en las siguientes fases:

FASE	DESCRIPCIÓN
Familiarización con la temática del proyecto	Conocer la base del problema y asentar conceptos básicos sobre el funcionamiento del corazón y los problemas derivados de la FA.
Búsqueda del estado del arte	Establecer el alcance y objetivos del proyecto para investigar las soluciones existentes, ventajas, limitaciones y extraer posibles ideas para aplicar.
Análisis y valoración de alternativas	Estudiar las posibles soluciones y determinar la mejor alternativa.
Formación en conceptos técnicos	Consolidar el conocimiento necesario para abordar el problema, tanto en el ámbito médico como en el informático.
Desarrollo	Elaboración de los objetivos planteados.
Evaluación y optimización	Evaluación de los resultados obtenidos y optimización de los sistemas construidos para conseguir la solución óptima.
Explotación de resultados	Análisis de los resultados finales para extraer conclusiones sobre la elaboración del proyecto y proposición de objetivos futuros.

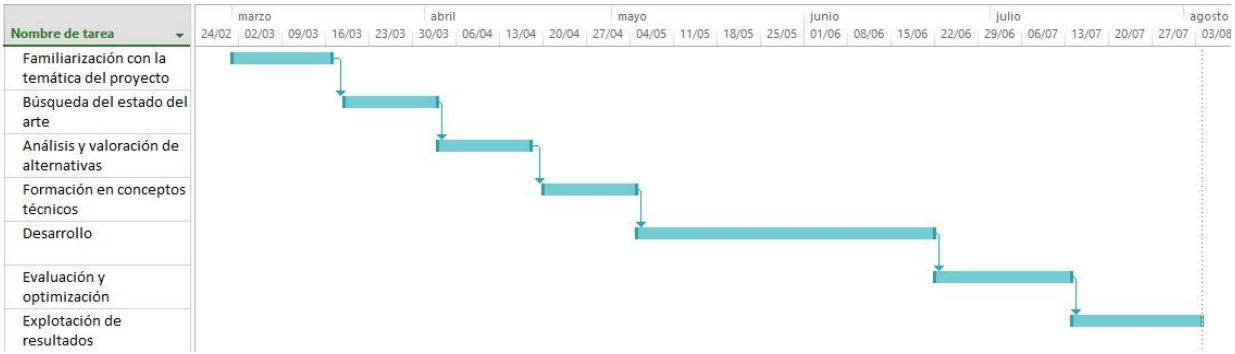


Figura 6.1.- Desarrollo temporal de las fases del proyecto.

6.2.- PRESUPUESTO

Gastos de personal

Concepto	Número de personas	Meses	Coste (€)
Contratado personal de investigación	1	5	1036,64
Total			5183,20

Materiales

Concepto	Cantidad	Coste (€)
NVIDIA Tesla T4	1	1855,92
Total		1855,92

Total costes

Concepto	Coste (€)
Gastos de personal	5183.20
Materiales	1855,92
Total (+ IVA)	7039.12€

REFERENCIAS

[1] "Patrones de Mortalidad en España, 2017." [Online]. Available: https://www.mscbs.gob.es/estadEstudios/estadisticas/estadisticas/estMinisterio/mortalidad/docs/Patrones_Mortalidad_2017.pdf.

[2] "WHO The top 10 causes of death." [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/the-top-10-causes-of-death.

[3] S. Kang, S. Lee, H. Cho, H. P.-I. S. Processing, and undefined 2016, "ECG authentication system design based on signal analysis in mobile and wearable devices," *ieeexplore.ieee.org*.

[4] C. J. Deepu, X. Y. Xu, X. D. Zou, L. B. Yao, and Y. Lian, "An ECG-on-Chip for Wearable Cardiac Monitoring Devices," *ieeexplore.ieee.org*, 2010, doi: 10.1109/DELTA.2010.43.

[5] A. Gacek and W. Pedrycz, "ECG signal analysis, classification, and interpretation: A framework of computational intelligence," in *ECG Signal Processing, Classification and Interpretation: A Comprehensive Framework of Computational Intelligence*, vol. 9780857298683, Springer-Verlag London Ltd, 2014, pp. 47–77.

[6] "Software para electrocardiografía - Pathfinder SL."

[7] "ECG Software Development." [Online]. Available: http://www.ecg-soft.com/.

[8] S. Mahmoodabadi, … A. A.-… I. E. in, and undefined 2006, "ECG feature extraction based on multiresolution wavelet transform," *ieeexplore.ieee.org*.

[9] A. Ahmadian, S. Z. Mahmoodabadi, A. Ahmadian, and M. D. Abolhasani, "ECG feature extraction using Daubechies wavelets Study of impact of Parsiss Navigation system in surgeries View project ECG FEATURE EXTRACTION USING DAUBECHIES WAVELETS," 2005.

[10] S. Fatemian, D. H.-2009 16th international, and undefined 2009, "A new ECG feature extractor for biometric recognition," *ieeexplore.ieee.org*.

[11] R. Martis, U. Acharya, L. M.-B. S. P. and Control, and undefined 2013, "ECG beat classification using PCA, LDA, ICA and discrete wavelet transform," *Elsevier*.

[12] P. Langley, E. Bowers, A. M.-I. transactions on, and undefined 2009, "Principal component analysis as a tool for analyzing beat-to-beat changes in ECG features: application to ECG-derived respiration," *ieeexplore.ieee.org*.

[13] F. Castells, P. Laguna, A. Bollmann, and J. Millet Roig, "Principal Component Analysis in ECG Signal Processing," *EURASIP J. Adv. Signal Process.*, vol. 74580, 2007, doi: 10.1155/2007/74580.

[14] S. Faziludeen, P. S.-2013 I. C. on, and undefined 2013, "ECG beat classification using wavelets and SVM," *ieeexplore.ieee.org*.

[15] S. Mehta, D. Shete, N. Lingayat, V. C.- Irbm, and undefined 2010, "K-means algorithm for the detection and delineation of QRS-complexes in Electrocardiogram," *Elsevier*.

[16] A. Naït-Ali, R. Fournier, N. Belgacem, A. Nait-Ali, and F. Bereksi-Reguig, "ECG Based Human Authentication using Wavelets and Random Forests," *Int. J. Cryptogr. Inf. Secur.*, vol. 2, no. 2, 2012, doi: 10.5121/ijcis.2012.2201.

[17] N. Wahab, A. Khan, Y. L.-C. in biology and medicine, and undefined 2017, "Two-phase deep convolutional neural network for reducing class skewness in histopathological images based breast cancer detection," *Elsevier*.

[18] B. E. Bejnordi *et al.*, "Deep learning-based assessment of tumor-associated stroma for diagnosing breast cancer in histopathology images," Feb. 2017.

[19] "Detecting Breast Cancer with Deep Learning."

[20] A. Sengur, Y. Akbulut, Y. Guo, and V. Bajaj, "Classification of amyotrophic lateral sclerosis disease based on convolutional neural network and reinforcement sample learning algorithm," *Heal. Inf Sci Syst. 2017 Dec;5(1) 9. doi10.1007/s13755-017-0029-6. PMID 29142739; PMCID PMC5662529.*, 2017.

[21] S. Kiranyaz, T. Ince, … R. H.-2015 37th A., and undefined 2015, "Convolutional Neural Networks for patient-specific ECG classification," *ieeexplore.ieee.org*.

[22] S. Kiranyaz, T. Ince, M. G.-I. T. on, and undefined 2015, "Real-time patient-specific ECG classification by 1-D convolutional neural networks," *ieeexplore.ieee.org*.

[23] B. Pourbabaee, M. J. Roshtkhari, and K. Khorasani, "Deep Convolutional Neural Networks and Learning ECG Features for Screening Paroxysmal Atrial Fibrillation Patients," *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 48, no. 12, pp. 2095–2104, Dec. 2018, doi: 10.1109/TSMC.2017.2705582.

[24] U. Acharya, H. Fujita, O. Lih, M. Adam, … J. T.-K.-B., and undefined 2017, "Automated detection of coronary artery disease using different durations of ECG segments with convolutional neural network," *Elsevier*.

[25] M. K. Das and S. Ari, "ECG Beats Classification Using Mixture of Features," *hindawi.com*, 2014, doi: 10.1155/2014/178436.

[26] G. Sannino and G. De Pietro, "A deep learning approach for ECG-based heartbeat classification for arrhythmia detection," *Futur. Gener. Comput. Syst. 10.1016/j.future.2018.03.057.*, 2018.

[27] R. Salloum, C. K.-2017 I. I. C. on, and undefined 2017, "ECG-based biometrics using recurrent

neural networks," *ieeexplore.ieee.org*.

[28] J. Shlens, "A Tutorial on Principal Component Analysis," Apr. 2014.

[29] J. B. Pendry *et al.*, "References and Notes Supporting Online Material Reducing the Dimensionality of Data with Neural Networks," *IEEE Trans. Microw. Theory Tech*, vol. 47, no. 9, p. 653, 1999, doi: 10.1126/science.1129198.

[30] P. Vincent, H. Larochelle, Y. Bengio, and P. A. Manzagol, "Extracting and composing robust features with denoising autoencoders," in *Proceedings of the 25th International Conference on Machine Learning*, 2008, pp. 1096–1103, doi: 10.1145/1390156.1390294.

[31] A. Makhzani and B. Frey, "k-Sparse autoencoders," in *2nd International Conference on Learning Representations, ICLR 2014 - Conference Track Proceedings*, 2014.

[32] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," Dec. 2013.

[33] I. J. Goodfellow *et al.*, "Generative Adversarial Networks," Jun. 2014.

[34] X. Yan, J. Yang, K. Sohn, and H. Lee, "Attribute2Image: Conditional image generation from visual attributes," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 2016, vol. 9908 LNCS, pp. 776–791, doi: 10.1007/978-3-319-46493-0_47.

[35] A. Gonzalez-Garcia, J. Van De Weijer, and Y. Bengio, "Image-to-image translation for cross-domain disentanglement."

[36] M.-Y. Liu, T. Breuel, and J. Kautz, "Unsupervised Image-to-Image Translation Networks."

[37] H. Xu *et al.*, "Unsupervised Anomaly Detection via Variational Auto-Encoder for Seasonal KPIs in Web Applications," in *Proceedings of the 2018 World Wide Web Conference on World Wide Web - WWW '18*, 2018, pp. 187–196, doi: 10.1145/3178876.3185996.

[38] J. An and S. Cho, "SNU Data Mining Center 2015-2 Special Lecture on IE Variational Autoencoder based Anomaly Detection using Reconstruction Probability," 2015.

[39] N. Mor, L. Wolf, A. Polyak, and Y. Taigman, "A Universal Music Translation Network."

[40] J. Engel *et al.*, "Neural Audio Synthesis of Musical Notes with WaveNet Autoencoders," 2017.

[41] I. V. Serban *et al.*, "A Hierarchical Latent Variable Encoder-Decoder Model for Generating Dialogues," May 2016.

[42] C. Esteban, S. L. Hyland, and G. Rätsch, "Real-valued (Medical) Time Series Generation with Recurrent Conditional GANs," Jun. 2017.

[43] X. Chen and E. Konukoglu, "Unsupervised Detection of Lesions in Brain MRI using constrained adversarial auto-encoders," Jun. 2018, doi: 10.3929/ethz-b-000321650.

[44] Q. Wei, Y. Ren, R. Hou, B. Shi, … J. L.-M. I. 2018, and undefined 2018, "Anomaly detection for medical images based on a one-class classification," *spiedigitallibrary.org*.

[45] R. Zhang, P. Isola, A. A. Efros, and B. A. Research, "Split-Brain Autoencoders: Unsupervised Learning by Cross-Channel Prediction."

[46] R. Miotto, F. Wang, S. Wang, … X. J.-B. in, and undefined 2018, "Deep learning for healthcare: review, opportunities and challenges," *academic.oup.com*.

[47] M. R. Avendi, A. Kheradvar, and H. Jafarkhani, "A combined deep-learning and deformable-model approach to fully automatic segmentation of the left ventricle in cardiac MRI," *Med. Image Anal.*, vol. 30, pp. 108–119, May 2016, doi: 10.1016/j.media.2016.01.005.

[48] J. EOM, S. KIM, and B. ZHANG, "AptaCDSS-E: A classifier ensemble-based clinical decision support system for cardiovascular disease level prediction," *Expert Syst. Appl.*, vol. 34, no. 4, pp. 2465–2479, May 2008, doi: 10.1016/j.eswa.2007.04.015.

[49] S. P. Shashikumar, A. J. Shah, Q. Li, G. D. Clifford, and S. Nemati, "A deep learning approach to monitoring and detecting atrial fibrillation using wearable technology," in *2017 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, 2017, pp. 141–144, doi: 10.1109/BHI.2017.7897225.

[50] O. Fabius and J. R. van Amersfoort, "Variational Recurrent Auto-Encoders," Dec. 2014, doi: 1412.6581.

[51] D. Park, Y. Hoshi, C. K.-I. R. and Automation, and undefined 2018, "A multimodal anomaly detector for robot-assisted feeding using an lstm-based variational autoencoder," *ieeexplore.ieee.org*.

[52] S. Suh, D. H. Chae, H.-G. Kang, and S. Choi, "Echo-state conditional variational autoencoder for anomaly detection," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1015–1022, doi: 10.1109/IJCNN.2016.7727309.

[53] S. Hochreiter and J. Schmidhuber, "Long Short Term Memory," *Neural Comput. 9*, 1997.

[54] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," Dec. 2014.

[55] Y. Bengio, P. Simard, and P. Frasconi, "Learning long-term dependencies with gradient descent is difficult," *IEEE Trans. Neural Networks*, vol. 5(2), 1994.

[56] D. P. Kingma, D. J. Rezende, S. Mohamed, and M. Welling, "Semi-supervised Learning with Deep Generative Models."

[57] J. Bergstra, D. Yamins, and D. D. Cox, "Hyperopt: A Python Library for Optimizing the Hyperparameters of Machine Learning Algorithms."

[58] L. S.-2017 I. W. C. on A. of and undefined 2017, "Cyclical learning rates for training neural networks," *ieeexplore.ieee.org*.

[59] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for

time series classification: a review," *Data Min. Knowl. Discov.*, vol. 33, no. 4, pp. 917–963, Jul. 2019, doi: 10.1007/s10618-019-00619-1.

[60] Z. Wang, W. Yan, T. O.-2017 I. joint conference, and undefined 2017, "Time series classification from scratch with deep neural networks: A strong baseline," *ieeexplore.ieee.org*.

[61] J. Serrà, S. Pascual, and A. Karatzoglou, "Towards a universal neural network encoder for time series," May 2018, doi: 1805.03908.

[62] A. Vaswani *et al.*, "Attention Is All You Need."

[63] P. Tanisaro, G. H.-2016 15th I. International, and undefined 2016, "Time series classification using time warping invariant echo state networks," *ieeexplore.ieee.org*.

[64] A. E. Hoerl and R. W. Kennard, "Ridge Regression: Applications to Nonorthogonal Problems," *Technometrics*, vol. 12, no. 1, pp. 69–82, 1970, doi: 10.1080/00401706.1970.10488635.

[65] M. Längkvist, L. Karlsson, A. L.-P. R. Letters, and undefined 2014, "A review of unsupervised feature learning and deep learning for time-series modeling," *Elsevier*.

[66] R. Pascanu, T. Mikolov, Y. B.- CoRR, undefined abs/1211.5063, and undefined 2012, "Understanding the exploding gradient problem."

[67] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks," 2013.

[68] S. García, F. Herrera, and H. U. Es, "An Extension on 'Statistical Comparisons of Classifiers over Multiple Data Sets' for all Pairwise Comparisons," 2008.

[69] I. Higgins, L. Matthey, A. Pal, C. Burgess, and X. Glorot, "beta-vae: Learning basic visual concepts with a constrained variational framework," 2016.

[70] A. van den Oord DeepMind, O. Vinyals DeepMind, and K. Kavukcuoglu DeepMind, "Neural Discrete Representation Learning."

[71] K. Gregor, G. Papamakarios, F. Besse, L. Buesing, and T. Weber, "Temporal Difference Variational Auto-Encoder," *7th Int. Conf. Learn. Represent. ICLR 2019*, Jun. 2018.