Comparación de desempeño de modelos clásicos y cuánticos de redes neuronales convolucionales para la detección de patrones en señales electrocardiográficas.

Santiago Cadavid Gil

Trabajo de grado, ingeniero electrónico

Tutor
Gustavo Adolfo Patiño Alvarez, doctor en ingeniería eléctrica

Universidad de Antioquia
Facultad de ingeniería, depatamento de ingeniería electronica y telecomunicaciones
Ingeniería electrónica
Medellín
2023

Índice

1. Resumen **5**

2. Introducción **6**

3. Objetivos **8**

3.1. Objetivo General 8

3.2. Objetivos Específicos 8

4. Marco Teórico **9**

4.1. La Electrocardiografía 9

4.1.1. Definición 9

4.1.2. Electrocardiograma (ECG) 10

4.1.3. Utilidad del Aprendizaje de Máquina y la Analítica de Datos en la Electrocardiografía 11

4.1.4. Patrones Típicos en un ECG 12

4.1.5. PhysioNet 13

4.1.6. Base de Datos: ECG-ID 14

4.2. Aprendizaje de Máquina 15

4.2.1. Definición 15

4.2.2. Red Neuronal Artificial (RNA) 16

4.2.3. Función de Activación ReLU (Rectified Linear Unit) 18

4.2.4. Función de Activación Sigmoide 19

4.2.5. Función de Error Negative Log-Likelihood (NLL) 21

4.2.6. Optimizador Adam 22

4.2.7. Convolución 23

4.2.8. Jupyter Notebooks y Python 24

4.3. Computación Cuántica 25

4.3.1. Definición 26
4.3.2. Conceptos Básicos de Computación Cuántica 26
4.3.3. Aprendizaje de Máquina Cuántico (QML) 27
4.3.4. Convolución en el Dominio Cuántico 28
4.3.5. Métricas para la Evaluación de los Modelos Cuánticos 28
4.3.6. Qiskit y PennyLane 29
4.4. Métricas de Desempeño 30
4.4.1. Complejidad de Tiempo 30
4.4.2. Exactitud (Accuracy) 31

5. Metodología **33**
5.1. Generación del Conjunto de Datos 33
5.1.1. Elección de la Base de Datos 35
5.1.2. Selección de las Señales de Interés 35
5.1.3. Segmentación y Etiquetado 37
5.1.4. Selección de la Estructura de Datos 38
5.1.5. Normalización 39
5.2. Desarrollo del Modelo Clásico 41
5.2.1. Preparación del Conjunto de Datos 41
5.2.2. Diseño del Modelo Clásico 42
5.2.3. Entrenamiento del Modelo Clásico 46
5.2.4. Prueba del Modelo Clásico 48
5.3. Desarrollo del Modelo Híbrido con Capa de Salida Cuántica 50
5.3.1. Diseño del Modelo Híbrido con Capa de Salida Cuántica 51
5.3.2. Circuito Parametrizable 52
5.3.3. Mapa de Características: ZZFeatureMap 53
5.3.4. Ansatz: RealAmplitudes 54
5.4. Desarrollo del Modelo híbrido con convolución en el dominio cuántico 56

5.4.1. Transformación de los Datos al Dominio Cuántico 58

6. Resultados y Análisis **62**

6.1. Resultados 62

6.1.1. Resultados del Modelo Clásico 63

6.1.2. Resultados del Modelo Híbrido con Capa Cuántica a la Salida 65

6.1.3. Resultados del Modelo Híbrido con Convolución en el Dominio Cuántico 68

6.2. Análisis de Resultados 70

7. Conclusiones **72**

Referencias **74**

1. Resumen

En el presente trabajo de grado, se abordo el desafío de clasificar complejos QRS en señales de electrocardiograma (ECG) mediante la aplicación de tres enfoques de modelado diferentes. El problema de clasificación de complejos QRS reviste una gran importancia en el campo de la cardiología y el monitoreo de la salud, ya que permite identificar patrones cardíacos anormales que pueden requerir atención médica [1].

Se evaluó el uso de tres modelos distintos: un modelo clásico de red neuronal convolucional, un modelo híbrido que combina convolución en el dominio clásico con una capa cuántica de salida, y otro modelo híbrido que emplea convolución en el dominio cuántico. La evaluación de estos modelos se centró en su exactitud y su complejidad temporal.

Los resultados obtenidos indican que los tres modelos demostraron un desempeño excepcional en términos de exactitud. Tanto el modelo clásico como la simulación del modelo híbrido con convolución en el dominio cuántico alcanzaron altas tasas de precisión. Este alto nivel de exactitud los posiciona como opciones viables para tareas de clasificación binaria de complejos QRS en señales de ECG. La elección del modelo más adecuado dependerá de la criticidad de la aplicación y las necesidades específicas del contexto.

En lo que respecta al desempeño en función del tiempo, el modelo híbrido con convolución en el dominio cuántico sobresalió al exhibir una complejidad temporal significativamente mejorada en comparación con los otros dos modelos. Esta eficiencia temporal es especialmente relevante en aplicaciones que requieren un procesamiento rápido de señales. Si bien el modelo clásico y el modelo híbrido con capa de salida cuántica presentaron un rendimiento similar en términos de complejidad temporal, el modelo clásico mostró una ligera ventaja.

2. Introducción

La electrocardiografía es un pilar fundamental en el diagnóstico temprano de afecciones cardíacas, permitiendo identificar patrones anormales en la actividad eléctrica del corazón [1]. El papel crucial de la electrocardiografía en la atención médica ha llevado a un interés creciente en la aplicación de técnicas de aprendizaje automático y la ciencia de datos para mejorar la precisión de la clasificación de complejos QRS en señales electrocardiográficas [2]. Antecedentes sólidos respaldan la importancia de la electrocardiografía en la medicina [3], y la creciente relevancia del aprendizaje automático y la ciencia de datos ha abierto nuevas posibilidades en el análisis de señales biomédicas. Sin embargo, el uso de la computación cuántica en este campo es un área de estudio novedosa y prometedora que aún se encuentra en desarrollo [4].

El problema que aborda este estudio es la clasificación binaria de segmentos QRS en señales de ECG, y busca comparar el desempeño entre tres enfoques de modelado neuronal: redes neuronales convolucionales (CNN), redes neuronales híbridas con convolución en el dominio cuántico y redes neuronales híbridas con salida cuántica. El objetivo principal es determinar cuál de estos modelos ofrece la mejor exactitud y complejidad temporal en tareas de clasificación de patrones.

Los alcances de este proyecto incluyen la implementación de modelos de CNN convencionales, así como redes neuronales híbridas que incorporan conceptos de computación cuántica. Específicamente, se centra en la clasificación de un patrón particular, el complejo QRS en señales de ECG. La computación cuántica se aborda mediante simulaciones, y es importante mencionar que la convolución en el dominio cuántico se realizó con valores de filtro estáticos, sin entrenamiento.

Este estudio se enfrenta a ciertas limitaciones inherentes a la tecnología cuántica actual, como la necesidad de simulación de hardware cuántico, que puede requerir un tiempo significativo de cómputo. Además, se asume que los datos de entrada están limpios y bien preparados, una situación que puede no ser siempre realista en entornos del mundo real.

La metodología empleada se enfocó en mantener una estructura de datos consistente para todos los modelos y estructuras de red lo más similares posible, con el fin de evaluar su desempeño en igualdad de condiciones. Esto garantiza una comparación justa y precisa de los modelos.

Este estudio es relevante en el avance del campo de la electrocardiografía y la aplicación de técnicas de aprendizaje automático y computación cuántica en la medicina. Los resultados obtenidos sientan un precedente importante para futuras aplicaciones de esta tecnología emergente en el diagnóstico y seguimiento de enfermedades cardíacas. Además, el uso de técnicas de aprendizaje automático en la electrocardiografía es un tema de estudio actual [2], y este trabajo contribuye al conocimiento en constante expansión en esta área.

3. Objetivos

3.1. Objetivo General

Comparar el desempeño entre redes neuronales convolucionales (CNN), redes neuronales híbridas con convolución en el dominio cuántico y redes neuronales híbridas con salida cuántica en el estudio de electrocardiografía, más específicamente en la clasificación binaria de segmentos QRS, a fin de determinar cuál modelo neuronal ofrece mejor exactitud y complejidad temporal en este tipo de tareas de clasificación de patrones.

3.2. Objetivos Específicos

1. Implementar una CNN mediante la librería PyTorch de Python, capaz de clasificar los complejos QRS en señales electrocardiográficas.

2. Utilizar la librería Qiskit y la CNN obtenida en el objetivo anterior para desarrollar una red neuronal híbrida con salida cuántica con el propósito de clasificar el complejo QRS en señales electrocardiográficas.

3. Diseñar e implementar una red neuronal híbrida con convolución en el dominio cuántico mediante el uso de la librería PennyLane para realizar la tarea de clasificación del complejo QRS.

4. Comparar el desempeño de los tres modelos neuronales desarrollados en función de su exactitud y complejidad temporal para la clasificación del complejo QRS en las señales electrocardiográficas. Este análisis permitirá determinar la validez de estas herramientas en el estudio electrocardiográfico y cuál modelo presenta el mejor rendimiento.

4. Marco Teórico

En el marco teórico de este estudio, se explorarán tres áreas clave que desempeñan un papel fundamental en la investigación sobre la detección de patrones en señales electrocardiográficas (ECG) mediante redes neuronales convolucionales (CNN). En primer lugar, se abordarán los conceptos fundamentales de la electrocardiografía, proporcionando una sólida base para comprender la naturaleza y las características de las señales ECG, incluyendo el segmento QRS. Luego, se explorarán los principios de la inteligencia computacional, centrándose en el funcionamiento y la aplicación de las redes neuronales convolucionales en la clasificación de datos médicos. Por último, se adentrará en el campo de la computación cuántica. Estos tres pilares teóricos servirán como cimiento para comprender en profundidad las técnicas y metodologías utilizadas en la comparación de modelos clásicos y cuánticos en la detección de patrones en señales ECG.

4.1. La Electrocardiografía

La electrocardiografía es un campo de la medicina que desempeña un papel central en la evaluación de la actividad eléctrica del corazón. Mediante la utilización de electrodos colocados estratégicamente en la superficie del cuerpo, esta técnica permite registrar y visualizar las corrientes eléctricas generadas por el músculo cardíaco durante su ciclo de bombeo. Estas señales eléctricas se traducen en un electrocardiograma (ECG o EKG), un gráfico que muestra la variación del potencial eléctrico a lo largo del tiempo en diversas partes del corazón. El ECG es una herramienta fundamental en el diagnóstico de trastornos cardíacos, como arritmias, infartos de miocardio y enfermedades de la conducción eléctrica del corazón [1].

4.1.1. Definición

La electrocardiografía, es una rama de la medicina que se enfoca en el estudio y la representación gráfica de la actividad eléctrica del corazón a lo largo del tiempo. Esta técnica no invasiva permite obtener un electrocardiograma, comúnmente abreviado como ECG, un registro visual que refleja los cambios en el potencial eléctrico generado por el sistema de

conducción eléctrica del corazón durante su ciclo de contracción y relajación.

El procedimiento básico implica la colocación estratégica de electrodos en la superficie de la piel del paciente, que actúan como detectores de las corrientes eléctricas generadas por el músculo cardíaco. Estas corrientes son el resultado de la activación y despolarización de las células del corazón y se manifiestan como ondas y segmentos característicos en el electrocardiograma.

El ECG es una herramienta esencial en la práctica médica, utilizada para diagnosticar una amplia variedad de trastornos cardíacos, desde arritmias y bloqueos cardíacos hasta infartos de miocardio. Además, proporciona información crucial sobre la velocidad del ritmo cardíaco, la duración de los intervalos eléctricos y la coherencia de la actividad eléctrica en el corazón, lo que facilita la identificación de anormalidades y la toma de decisiones clínicas fundamentales en la atención médica cardiovascular[1].

4.1.2. Electrocardiograma (ECG)

El electrocardiograma (ECG) es una técnica no invasiva utilizada en medicina para registrar la actividad eléctrica del corazón. Proporciona información valiosa sobre la función cardíaca al medir y representar gráficamente las señales eléctricas generadas por el corazón durante su ciclo de bombeo. A través de la colocación de electrodos en puntos estratégicos en la superficie del cuerpo, se pueden capturar estas señales y convertirlas en un trazado gráfico conocido como un electrocardiograma[5].

El procedimiento para realizar un ECG implica la conexión de electrodos a la piel del paciente, generalmente en el pecho, las extremidades y a veces en las piernas. Estos electrodos detectan las corrientes eléctricas que fluyen a través del músculo cardíaco mientras se contrae y se relaja. Estas corrientes se registran como ondas en el ECG, que representan eventos específicos del ciclo cardíaco.

El principio de funcionamiento del ECG se basa en la idea de que la actividad eléctrica del corazón se propaga de manera predecible a través de las diferentes partes del órgano. Durante un ciclo cardíaco típico, se pueden observar varias ondas en el ECG, incluyendo la onda P,

el complejo QRS y la onda T. Cada una de estas ondas refleja un evento particular en el ciclo, como la despolarización de las aurículas (onda P), la despolarización de los ventrículos (complejo QRS) y la repolarización de los ventrículos (onda T)[5].

El ECG se utiliza ampliamente en el diagnóstico y monitoreo de trastornos cardíacos, como arritmias, infartos de miocardio, enfermedades de la conducción y otras afecciones cardíacas. Proporciona información esencial sobre la regularidad y la eficiencia de la actividad cardíaca, lo que ayuda a los médicos a evaluar la salud del corazón y tomar decisiones clínicas informadas[1].

4.1.3. Utilidad del Aprendizaje de Máquina y la Analítica de Datos en la Electrocardiografía

En el ámbito de la electrocardiografía, el aprendizaje de máquina y la analítica de datos han demostrado ser herramientas fundamentales para la interpretación y el análisis de las señales electrocardiográficas (ECG). Estas tecnologías ofrecen una valiosa capacidad para procesar y extraer información relevante de las ECG, lo que ha llevado a avances significativos en la detección de patrones y anomalías cardíacas[2].

La aplicabilidad de estas técnicas en el campo de la electrocardiografía es de gran relevancia, ya que permiten automatizar y mejorar la precisión en la identificación de patrones específicos en las señales ECG[6]. Esto incluye la detección de segmentos QRS, que es un componente crítico en el diagnóstico de trastornos cardíacos.

En el contexto de este trabajo de grado, que se enfoca en la comparación del uso de diferentes técnicas de aprendizaje de máquina para la detección de patrones, específicamente en la identificación del complejo QRS en electrocardiogramas, estas técnicas desempeñaron un papel crucial. La aplicación de algoritmos de aprendizaje de máquina permitió una deteccion automatizada del complejo QRS en un amplio conjunto de datos de ECG.

En adición, es importante destacar que la analítica de datos desempeñó un papel esencial en este trabajo. Aunque la limpieza de datos no fue realizada por el autor, la base de datos utilizada ya había sido preprocesada.

4.1.4. Patrones Típicos en un ECG

Un electrocardiograma (ECG) es una representación gráfica de la actividad eléctrica del corazón durante un ciclo cardíaco. En un ECG típico, observamos varias ondas que reflejan eventos específicos en el proceso de contracción y relajación del corazón:

- **Onda P:** La onda P es la primera en aparecer en un ECG y refleja la despolarización auricular, que es la activación de las aurículas del corazón. Esta activación precede a la contracción auricular, lo que permite que la sangre fluya de las aurículas a los ventrículos[5].

- **Complejo QRS:** El complejo QRS es la característica más distintiva de un ECG y representa la despolarización ventricular. Durante esta fase, los ventrículos del corazón se contraen, expulsando la sangre hacia las arterias principales. El complejo QRS consiste en una serie de ondas, incluyendo la onda Q (si está presente), la onda R (la más alta) y la onda S (si está presente). La morfología y duración del complejo QRS pueden variar y son cruciales para evaluar la salud del sistema de conducción cardíaca[5].

- **Onda T:** La onda T sigue al complejo QRS y representa la repolarización ventricular, es decir, la recuperación de los ventrículos después de la contracción. La forma y la amplitud de la onda T pueden proporcionar información sobre la función cardíaca y la presencia de trastornos[5].

- **Onda U (si está presente):** La onda U es menos común y suele ser pequeña. Su origen y significado no están completamente claros, pero se asocia con la repolarización de ciertas regiones del corazón[5].

En la Figura 1, se presenta una representación gráfica en la que se pueden visualizar claramente las diferentes ondas previamente mencionadas en el ciclo cardíaco. Esta imagen proporciona una representación visual de las ondas P, QRS y T en un electrocardiograma, lo que facilita la comprensión de la actividad eléctrica del corazón durante su funcionamiento.

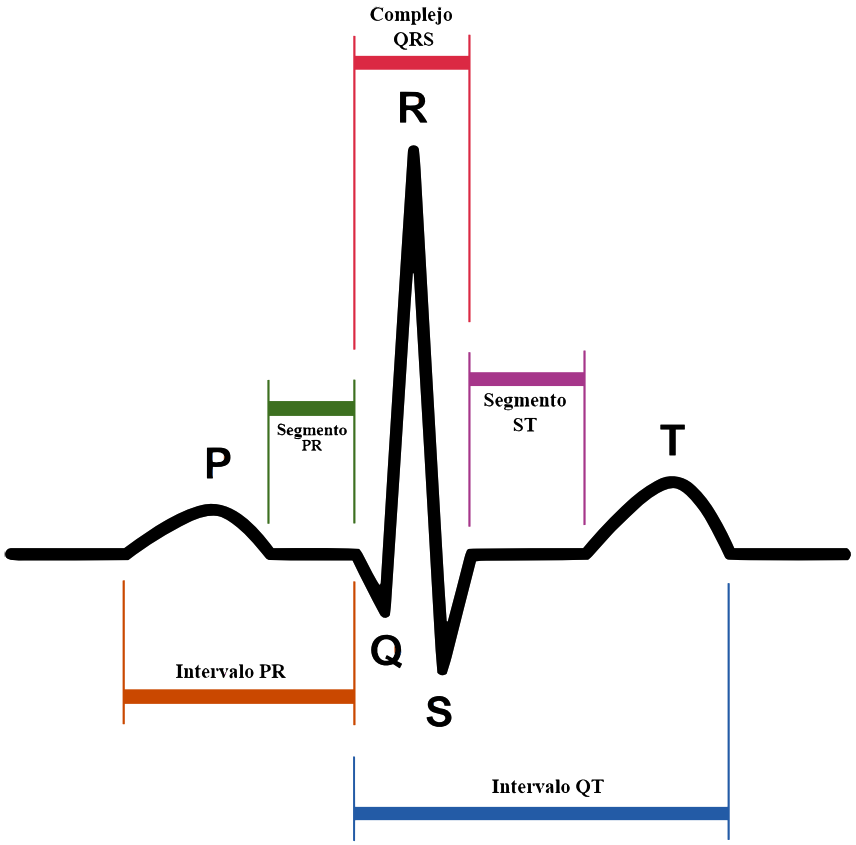

Figura 1

Representación de un segmento PQRST [7].

En el contexto de este trabajo de grado, se optó por la detección del complejo QRS debido a su forma característica y su importancia en la evaluación de la actividad eléctrica ventricular. Esta decisión se basa en la relevancia clínica del complejo QRS para el diagnóstico de trastornos cardíacos y su distinción en el ECG, lo que lo convierte en un punto focal para la investigación y la comparación de técnicas de detección de patrones.

4.1.5. PhysioNet

PhysioNet [8] es una plataforma en línea que se encuentra bajo la administración y supervisión del Instituto Nacional de Salud de Estados Unidos (NIH) y la Facultad de Medicina de la Universidad de Boston. Fue lanzada en el año 1999 y ha crecido hasta convertirse en una invaluable fuente de datos médicos y señales fisiológicas de libre acceso para la investigación y el desarrollo de algoritmos relacionados con la salud.

Esta plataforma ofrece una amplia gama de bases de datos clínicas y señales fisiológicas que abarcan diversas áreas médicas, desde electrocardiografía y registros de señales respiratorias hasta datos de monitoreo de la presión arterial. La disponibilidad pública de estos datos facilita la colaboración entre investigadores de todo el mundo y promueve avances significativos en la atención médica y la investigación biomédica[8].

En el presente trabajo de grado, todos los datos utilizados se obtuvieron de PhysioNet. Esta fuente confiable y accesible, respaldada por el NIH y la Universidad de Boston, proporcionó un conjunto de datos diverso y valioso que fue fundamental para el desarrollo de la investigación.

4.1.6. Base de Datos: ECG-ID

La base de datos ECG-ID utilizada en este trabajo de grado se originó en el estudio de Tatiana S. Lugovaya, titulado "Biometric human identification based on electrocardiogram"[9]. Esta base de datos tomada de PhysioNet es el resultado de investigaciones que exploraron la posibilidad de identificación biométrica de seres humanos basada en el electrocardiograma (ECG).

ECG-ID contiene registros de ECG de 90 voluntarios, sin restricciones en cuanto a la frecuencia cardíaca, estado físico o emocional. Para garantizar la usabilidad, se seleccionó la derivación D1 del plano frontal. La derivación D1 se obtiene colocando un electrodo en el brazo derecho y otro en el brazo izquierdo del paciente. La corriente eléctrica fluye entre estos dos electrodos y, como resultado, se registra la diferencia de potencial eléctrico entre ellos [5]. En dicho estudio, la elección de la derivación D1 se justifica por su facilidad de medición y su insensibilidad a variaciones menores en la ubicación de los electrodos. Los fragmentos de ECG que contenían complejos QRS, ondas P y T extraídos de los ECG se procesaron mediante Análisis de Componentes Principales (PCA) y se clasificaron utilizando Análisis Discriminante Lineal (LDA) y un Clasificador de Votación Mayoritaria [9].

Esta base de datos fue esencial para llevar a cabo investigaciones en el presente trabajo de grado y contribuyó significativamente al desarrollo de la investigación, permitiendo análisis

y experimentación que respaldaron la detección del complejo QRS en señales electrocardiográficas.

4.2. Aprendizaje de Máquina

Esta subsección explorará los conceptos clave del Aprendizaje de Máquina y su aplicación en la interpretación de señales ECG, resaltando su pertinencia en el contexto de este trabajo de grado. En particular, esta subsección se enfocará en explorar la concepción clásica de la inteligencia artificial, que se basa en la idea de desarrollar sistemas capaces de realizar tareas cognitivas que requieren inteligencia humana. Para la implementación de dichos sistemas en la interpretación de señales electrocardiográficas, se hará uso de redes neuronales convolucionales (CNN).

4.2.1. Definición

El Aprendizaje de Máquina (Machine Learning, en inglés) es una rama de la inteligencia artificial que se enfoca en desarrollar algoritmos y modelos computacionales capaces de aprender de manera autónoma a partir de datos y experiencias previas. Estos algoritmos permiten que las computadoras identifiquen patrones, realicen predicciones y tomen decisiones sin necesidad de una programación explícita[10]. En esencia, el Aprendizaje de Máquina busca que las máquinas mejoren su rendimiento en tareas específicas a medida que se les proporciona más información y datos.

En el contexto de este trabajo de grado, el Aprendizaje de Máquina desempeña un papel esencial en la detección del complejo QRS en señales electrocardiográficas (ECG). El complejo QRS es un patrón característico en el ECG que refleja la activación eléctrica de los ventrículos cardíacos. Su detección y análisis son cruciales para la identificación de trastornos cardíacos, ya que proporciona información valiosa sobre la salud cardíaca de un individuo[1]. La aplicación de algoritmos de Aprendizaje de Máquina permite automatizar este proceso, lo que facilita el diagnóstico temprano de afecciones cardíacas y mejora la atención médica. El Aprendizaje de Máquina es especialmente valioso en este contexto, ya que puede analizar grandes volúmenes de datos de ECG de manera eficiente y precisa, superando las limitaciones

de la revisión manual. Esto permite una atención médica más rápida y precisa, lo que puede salvar vidas y mejorar la calidad de vida de los pacientes con afecciones cardíacas[1].

4.2.2. Red Neuronal Artificial (RNA)

Una Red Neuronal Artificial (RNA) es un modelo computacional inspirado en la estructura y funcionamiento del cerebro humano. Se compone de una red interconectada de unidades de procesamiento llamadas neuronas artificiales. Cada neurona artificial opera como una unidad de cómputo simple que realiza cálculos en función de su entrada y produce una salida. La idea fundamental detrás de una RNA es emular la capacidad de aprendizaje y adaptación a partir de datos, similar a cómo las redes de neuronas en el cerebro humano procesan información[10].

Partes básicas de una RNA:

- **Entradas:** Las entradas son los datos que se proporcionan a la red neuronal para su procesamiento. Cada entrada está asociada con un valor numérico y representa una característica o atributo del problema que se está resolviendo.
- **Neurona:** Una neurona artificial es la unidad básica de procesamiento en una red neuronal. Realiza cálculos en función de las entradas que recibe y produce una salida. Cada neurona puede tener una función de activación que determina cómo procesa las entradas y genera la salida.
- **Pesos:** Los pesos son parámetros ajustables que se asocian con las conexiones entre las neuronas. Cada conexión tiene un peso que controla la influencia de la neurona de entrada en la neurona de salida. Durante el entrenamiento, estos pesos se ajustan para mejorar el rendimiento de la red.
- **Bias:** El bias, o sesgo, es un parámetro adicional asociado con cada neurona. Añade un término constante a la suma ponderada de las entradas antes de aplicar la función de activación. El bias permite que la red capture relaciones más complejas en los datos[11].

- **Función de Activación:** La función de activación determina la salida de una neurona en función de su entrada ponderada. Introduce no linealidad en la red y permite que la RNA aprenda relaciones más complejas en los datos[10].

- **Función de Error:** La función de error es una medida que evalúa cuán bien se desempeña la red neuronal en relación con los valores deseados o las etiquetas en un problema de aprendizaje supervisado. Cuantifica la discrepancia entre las salidas predichas por la red y las salidas reales, lo que permite ajustar los pesos y bias durante el entrenamiento.

- **Capas:** Las capas son grupos de neuronas organizadas en la red neuronal. La red consta de la capa de entrada, una o varias capas ocultas y la capa de salida. Cada capa tiene un propósito específico en el procesamiento de información.

- **Capas Ocultas:** Las capas ocultas son capas intermedias entre la capa de entrada y la capa de salida. Realizan transformaciones y cálculos complejos para aprender patrones en los datos. Cuantas más capas ocultas tenga una red, mayor será su capacidad para representar relaciones complejas[10].

- **Salidas:** Las salidas de la red neuronal son los resultados finales que produce después de procesar las entradas a través de sus capas y neuronas. En muchos casos, las salidas representan predicciones, clasificaciones o valores deseados en función del problema que se está resolviendo.

En la Figura 2, se presenta una representación gráfica que ilustra de manera básica la estructura de una neurona, el elemento fundamental de las redes neuronales. Esta imagen muestra los componentes esenciales de una neurona, incluyendo las conexiones de entrada, los pesos sinápticos y la función de activación. A partir de esta representación, es posible comprender cómo las neuronas individuales procesan la información y cómo se conectan en redes más grandes para realizar tareas más complejas.

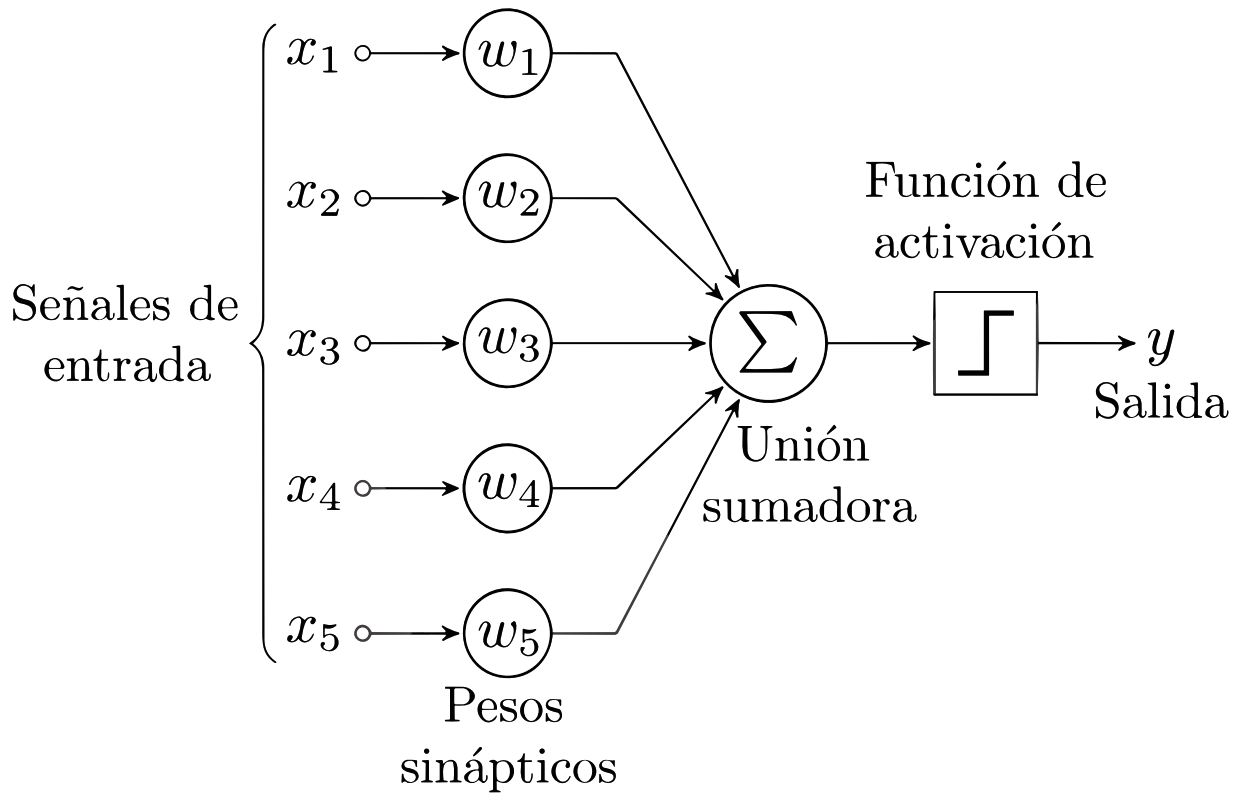

Figura 2

Representacion basica de una neurona [12].

Las RNA fueron la herramienta principal utilizada en este trabajo de grado para la detección del complejo QRS en señales electrocardiográficas.

4.2.3. Función de Activación ReLU (Rectified Linear Unit)

La función de activación ReLU [13], abreviatura de Rectified Linear Unit, es una función ampliamente utilizada en redes neuronales artificiales y se define matemáticamente como:

$$f(x) = \text{máx}(0, x)$$

Donde x es la entrada a la función. La operación de ReLU devuelve x si x es mayor o igual a cero y cero en caso contrario. En otras palabras, esta función rectifica"los valores negativos, convirtiéndolos en cero, y mantiene los valores positivos sin cambios.

La principal ventaja de la función de activación ReLU radica en su simplicidad y capaci-

dad para introducir no linealidad en la red neuronal. A pesar de su simplicidad, ReLU ha demostrado ser muy efectiva en la mayoría de las aplicaciones de aprendizaje profundo[13]. Algunas de las razones de su popularidad incluyen:

1. **Facilidad de cálculo**: La función ReLU es fácil de calcular computacionalmente, lo que la hace eficiente en términos de tiempo de entrenamiento y predicción.

2. **Prevención de desvanecimiento del gradiente**: A diferencia de algunas funciones de activación, ReLU no sufre del problema de desvanecimiento del gradiente, lo que significa que facilita el entrenamiento de redes neuronales profundas.

3. **No linealidad**: A pesar de su forma lineal en la región positiva, ReLU introduce no linealidad en la red, lo que le permite aprender y representar relaciones complejas en los datos.

4. **Efecto de selección**: ReLU actúa como un mecanismo de selección, permitiendo que solo las activaciones relevantes y fuertes se propaguen a través de la red, lo que ayuda a reducir la redundancia en la representación de características.

En la Figura 3, se presenta una representación gráfica de la función de activación Rectified Linear Unit (ReLU). En esta figura, se puede visualizar claramente cómo la función ReLU actúa como un interruptor que se activa de forma lineal con respecto a la entrada cuando ésta es positiva y se desactiva cuando es negativa.

4.2.4. Función de Activación Sigmoide

La función de activación sigmoide, también conocida como función logística, es una función matemática que transforma un valor real en un valor en el rango de 0 a 1 [15]. Se representa matemáticamente como:

$$\sigma(z) = \frac{1}{1 + e^{-z}}$$

Donde:

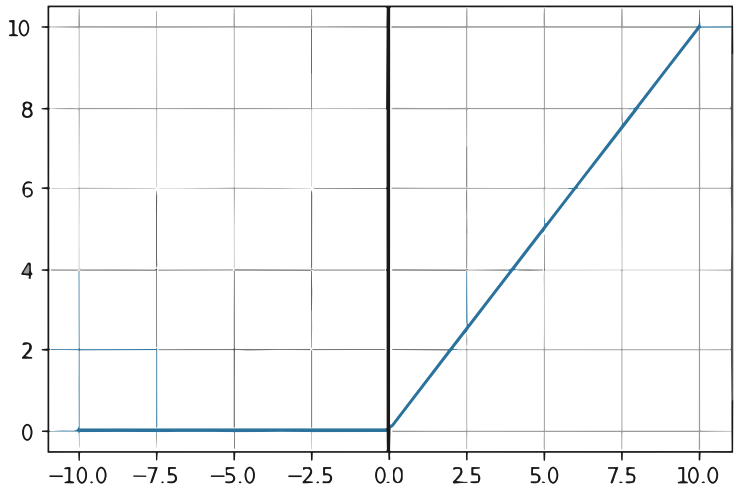

Figura 3

Función de activación ReLU [14].

- $\sigma(z)$ es el valor de activación transformado.
- z es la entrada a la función, que es la suma ponderada de las entradas y los pesos de una neurona en una red neuronal.

En la Figura 4, se presenta una representación gráfica de la función de activación sigmoide. En esta figura, se puede visualizar claramente cómo la función sigmoide actúa como una especie de interruptor suave que se activa gradualmente a medida que la entrada aumenta y se desactiva a medida que disminuye, transformando los valores de entrada en un rango de 0 a 1.

Utilidad en Problemas de Clasificación Binaria

La función sigmoide es ampliamente utilizada en problemas de clasificación binaria debido a su capacidad para representar probabilidades. Su rango de salida limitado entre 0 y 1 hace que sea adecuada para este propósito. En problemas de clasificación binaria, se busca asignar una observación a una de las dos clases posibles. La función sigmoide se interpreta como la probabilidad de que la observación pertenezca a la clase positiva.

Cuando se utiliza en una red neuronal, la función sigmoide se coloca típicamente en la capa de salida de la red. La salida de la función sigmoide se interpreta como la probabilidad de

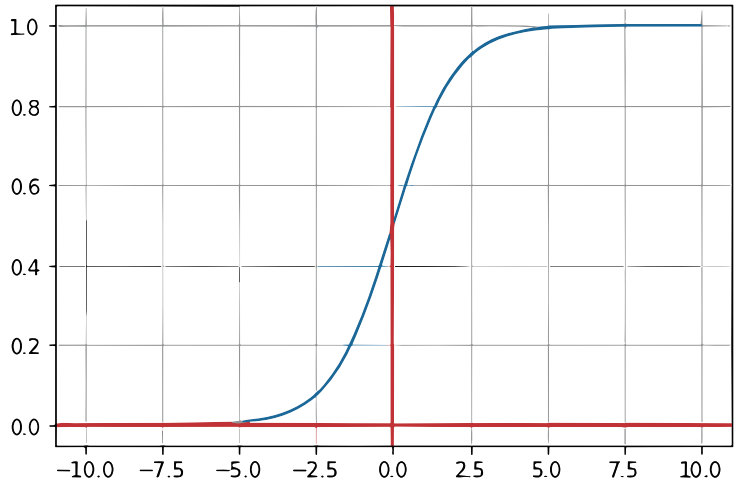

Figura 4

Función de activación simoide [14].

que una entrada dada pertenezca a una de las dos clases. Si la probabilidad es mayor o igual a 0.5, se clasifica como la clase positiva; de lo contrario, se clasifica como la clase negativa.

Utilidad en el Presente Trabajo de Grado

En el contexto del presente trabajo de grado, la función de activación sigmoide desempeñó un papel fundamental en la etapa de salida de la red neuronal utilizada para la detección de complejos QRS en señales electrocardiográficas. Dado que el problema se redujo a una clasificación binaria (determinar si una señal es o no un complejo QRS), la función sigmoide en la capa de salida permitió calcular la probabilidad de que una señal dada perteneciera a la clase de complejo QRS.

Esta utilización de la función sigmoide facilitó la interpretación de las salidas de la red como probabilidades y permitió tomar decisiones binarias basadas en umbral, lo que resultó ser crucial en la detección precisa de los complejos QRS en las señales electrocardiográficas.

4.2.5. Función de Error Negative Log-Likelihood (NLL)

La función de error Negative Log-Likelihood (NLL), también conocida como función de pérdida logarítmica negativa, es una función de error comúnmente utilizada en problemas de clasificación, especialmente en problemas de clasificación multiclase [16]. Su objetivo es cuan-

tificar la discrepancia entre las probabilidades predichas por el modelo y las probabilidades reales de las clases.

La fórmula matemática de la NLL se expresa como:

$$NLL(y, \hat{y}) = -\sum_{i=1}^{n} y_i \cdot \log(\hat{y}_i)$$

Donde:

- y es un vector one-hot encoding de las clases reales, donde y_i es 1 si la observación pertenece a la clase i y 0 en caso contrario.
- $\hat{y}$ es un vector de probabilidades predichas por el modelo para cada clase.
- $\log(\hat{y}_i)$ es el logaritmo natural de la probabilidad predicha para la clase i.

Funcionamiento y Utilidad

La función de error NLL mide la discrepancia entre las probabilidades predichas por el modelo y las probabilidades reales de las clases. Es especialmente útil en problemas de clasificación, donde se busca asignar una observación a una de las clases posibles. La NLL penaliza fuertemente las predicciones incorrectas, asignando un mayor valor de pérdida cuando la probabilidad predicha para la clase correcta es baja[16].

En el contexto de una red neuronal utilizada para la detección de patrones, como el presente trabajo de grado que se centra en la detección de complejos QRS en señales electrocardiográficas, la NLL se utiliza como función de error para entrenar el modelo. Durante el entrenamiento, la red neuronal ajusta sus pesos y bias para minimizar la NLL, lo que implica que el modelo se esfuerza por mejorar su capacidad de asignar probabilidades precisas a las clases de interés.

4.2.6. Optimizador Adam

El optimizador Adam (Adaptive Moment Estimation)[17] es un algoritmo de optimización ampliamente utilizado en el entrenamiento de modelos de aprendizaje automático, especialmente en redes neuronales. Fue desarrollado para abordar limitaciones presentes en otros

algoritmos de optimización, como el descenso de gradiente estocástico (SGD) [17]. La característica distintiva de Adam radica en su capacidad para ajustar de manera automática y eficiente la tasa de aprendizaje para cada parámetro del modelo, lo que lo hace especialmente efectivo en la optimización de modelos que involucran grandes conjuntos de datos y múltiples parámetros.

El funcionamiento de Adam se basa en el cálculo y seguimiento de estimaciones de primer y segundo momento del gradiente. Esto permite la adaptación de la tasa de aprendizaje de forma individualizada para cada parámetro, lo que conduce a una convergencia más rápida y una mayor estabilidad durante el entrenamiento.

Utilidad para el presente trabajo de grado

En el contexto del presente trabajo de grado, que se centra en la clasificación del complejo QRS en señales electrocardiográficas, el optimizador Adam desempeña un papel crucial en el entrenamiento de los modelos. Dado que se trabajará con redes neuronales y se buscará obtener una alta precisión en la clasificación, Adam se convierte en una elección ideal debido a su capacidad para ajustar dinámicamente la tasa de aprendizaje. El uso del optimizador Adam es esencial al lidiar con conjuntos de datos ruidosos, como las señales ECG. Facilita la rápida convergencia de los modelos, lo que garantiza un entrenamiento eficiente y resultados precisos en la clasificación de complejos QRS.

4.2.7. Convolución

La operación de convolución es una técnica fundamental en el procesamiento de imágenes y señales utilizada en redes neuronales convolucionales (CNN). En el contexto de las CNN, la convolución se utiliza para extraer características relevantes de una entrada, como una imagen o una señal, mediante la aplicación de filtros o kernels. Estos filtros son pequeñas matrices de pesos que se desplazan a lo largo de la entrada y realizan una operación de multiplicación y suma ponderada en cada posición. El resultado se almacena en una nueva matriz llamada "mapa de características"[18].

La función principal de la operación de convolución en una CNN es aprender características

locales y patrones en los datos de entrada. Los filtros actúan como detectores de características, como bordes, texturas o formas, y la convolución se utiliza para aplicar estos detectores de manera eficiente a todo el conjunto de datos de entrada. A medida que la red se entrena en tareas específicas, los filtros aprenden a reconocer patrones relevantes para la tarea, lo que permite a la CNN realizar tareas de clasificación, detección de objetos, entre otras.

La operación de convolución desempeña un papel crucial en este trabajo de grado, ya que se utiliza para extraer características relevantes de las señales electrocardiográficas con el objetivo de clasificar el complejo QRS. En este contexto, la convolución se aplica para detectar patrones específicos en las señales, como las características distintivas del complejo QRS, lo que facilita su identificación y clasificación.

4.2.8. Jupyter Notebooks y Python

Jupyter Notebooks es un entorno de desarrollo interactivo que permite la creación y compartición de documentos que contienen código, texto enriquecido, visualizaciones y explicaciones [19]. Los notebooks se organizan en celdas, donde cada celda puede contener código ejecutable o contenido markdown, lo que facilita la combinación de programación y documentación en un solo lugar.

Python, por otro lado, es un lenguaje de programación ampliamente utilizado en el campo del aprendizaje de máquina y la ciencia de datos debido a su simplicidad y versatilidad [20]. Python cuenta con una amplia variedad de bibliotecas y marcos de trabajo diseñados específicamente para tareas de análisis de datos y aprendizaje automático.

Utilidad en el Desarrollo del Trabajo de Grado

Jupyter Notebooks y Python desempeñaron un papel fundamental en el desarrollo del presente trabajo de grado. Su utilidad se puede resumir de la siguiente manera:

- **Programación Interactiva**: Jupyter Notebooks permitieron la programación interactiva y la ejecución de código paso a paso, lo que facilitó la experimentación y depuración de algoritmos de aprendizaje de máquina.

- **Documentación y Comunicación**: Los notebooks proporcionaron un medio eficaz para documentar el trabajo, incluyendo la descripción de métodos, resultados y hallazgos.

- **Exploración de Datos**: Python, junto con bibliotecas como NumPy, pandas y matplotlib, permitió la exploración y el análisis de datos de electrocardiogramas de manera eficiente. Esto fue esencial para comprender la naturaleza de los datos y prepararlos para su procesamiento.

- **Implementación de Algoritmos de Aprendizaje de Máquina**: Python cuenta con bibliotecas como Pytorch y TensorFlow que facilitaron la implementación de modelos de aprendizaje de máquina, incluyendo redes neuronales convolucionales. Esto permitió comparar diferentes técnicas y evaluar su rendimiento.

- **Visualización de Resultados**: Las capacidades de visualización de Python fueron cruciales para representar gráficamente los resultados, incluyendo gráficos de rendimiento y visualizaciones de señales electrocardiográficas.

4.3. Computación Cuántica

A continuación, se explorarán conceptos clave relacionados con la Computación Cuántica. Estos conceptos son fundamentales para la comprensión de algunas de las técnicas avanzadas que fueron utilizadas en el presente trabajo de grado. A medida que avanzamos en la era de la tecnología, la Computación Cuántica ha surgido como un campo de investigación y aplicación con el potencial de revolucionar la forma en que se abordan problemas computacionales complejos.

Enseguida, se definirán y explicarán los principios básicos de la Computación Cuántica, incluyendo los qubits, la superposición, el entrelazamiento y la convolución en el dominio cuántico. Estos conceptos son esenciales para comprender cómo las técnicas cuánticas pueden aplicarse en la interpretación y procesamiento de señales electrocardiográficas.

4.3.1. Definición

La Computación Cuántica es un paradigma computacional que utiliza principios de la mecánica cuántica, como la superposición y el entrelazamiento, para realizar cálculos de manera exponencialmente más eficiente que las computadoras clásicas. Estas propiedades cuánticas permiten abordar problemas complejos de manera más rápida y precisa[21], lo que puede mejorar y optimizar significativamente la detección de patrones en electrocardiogramas (ECG).

4.3.2. Conceptos Básicos de Computación Cuántica

A continuación, se describirán los conceptos fundamentales de la Computación Cuántica necesarios para comprender el trabajo realizado en este campo. Estos conceptos proporcionan una base sólida para comprender cómo se aplican las técnicas cuánticas en la detección de patrones en electrocardiogramas.

- **Qubit:** Un qubit, o bit cuántico, es la unidad básica de información cuántica. A diferencia de un bit clásico, que solo puede tomar valores 0 o 1, un qubit puede estar en una superposición de estados 0 y 1 simultáneamente, lo que permite un procesamiento de información más versátil y con la capacidad de representar información compleja de manera eficiente[22].

- **Compuerta Cuántica:** Las compuertas cuánticas, son operaciones que actúan sobre los qubits para realizar transformaciones cuánticas específicas. Estas operaciones son esenciales para realizar cálculos y manipulaciones en qubits, lo que afecta directamente la información representada en el sistema cuántico[22]. La rotación cuántica de los qubits implica la aplicación de ángulos de fase al estado de los mismo, lo que permite una mayor versatilidad en el procesamiento y manipulación de información en sistemas cuánticos. Estas rotaciones son una parte fundamental en la construcción de algoritmos cuánticos y la implementación de compuertas cuánticas para diversas aplicaciones en computación.

- **Superposición:** La superposición es un principio clave de la mecánica cuántica que

permite que un qubit esté en una combinación ponderada de sus estados base (0 y 1) al mismo tiempo. Esto permite realizar múltiples cálculos en paralelo y es esencial para el poder de procesamiento cuántico[21].

- **Entrelazamiento:** El entrelazamiento es otra propiedad cuántica que permite que dos o más qubits estén correlacionados de manera que el estado de uno de ellos dependa instantáneamente del estado de los otros, incluso a grandes distancias. Esta propiedad es fundamental en la Computación Cuántica y permite la implementación de algoritmos cuánticos eficientes[21].

4.3.3. Aprendizaje de Máquina Cuántico (QML)

A continuación, se explorará el concepto de Aprendizaje de Máquina Cuántico (QML), que combina los principios de la Computación Cuántica con el Aprendizaje de Máquina para resolver problemas de manera más eficiente y precisa. El QML se ha convertido en un área de investigación prometedora debido a su capacidad para aprovechar las ventajas cuánticas en la manipulación y el procesamiento de datos[23].

Un enfoque importante del QML es la construcción de redes neuronales cuánticas, que pueden contener capas convolucionales cuánticas y capas de salida cuánticas. Estas redes híbridas aprovechan la potencia del procesamiento cuántico para tareas específicas, como la detección de patrones en señales electrocardiográficas.

Las capas convolucionales cuánticas pueden realizar operaciones de convolución en el dominio cuántico, aprovechando el entrelazamiento cuántico. Esta propiedad es especialmente beneficiosa para analizar señales complejas como las provenientes de ECG, ya que permite una manipulación más eficiente y precisa de la información contenida en las señales electrocardiográficas. Por otro lado, las capas de salida cuánticas permiten realizar mediciones cuánticas que capturan información relevante para la tarea de clasificación o detección.

En el contexto de este trabajo de grado, se exploró la implementación de técnicas de QML para la detección del complejo QRS en señales electrocardiográficas, lo que permitió una

evaluación comparativa con enfoques clásicos de Machine Learning.

4.3.4. Convolución en el Dominio Cuántico

La convolución en el dominio cuántico es una operación matemática fundamental que se aplica a funciones cuánticas. Esta operación se basa en el principio de superposición cuántica y utiliza la propiedad del entrelazamiento cuántico para combinar el estado de dos o más qubits y obtener un nuevo estado cuántico resultante[24].

En el contexto del presente trabajo de grado, la convolución en el dominio cuántico se utiliza para procesar las señales electrocardiográficas (ECG) de manera eficiente y precisa. Permite identificar patrones específicos en las señales, como el complejo QRS característico, al realizar operaciones de convolución entre los qubits representativos de los patrones deseados y la señal ECG. Esto facilita la detección y el análisis de patrones cardíacos.

4.3.5. Métricas para la Evaluación de los Modelos Cuánticos

Para evaluar el desempeño de los modelos cuánticos desarrollados en este trabajo de grado, se emplearon métricas similares a las utilizadas en la evaluación de los modelos clásicos. Específicamente, se consideraron dos métricas clave: la complejidad de tiempo y la exactitud. El concepto de exactitud es el mismo tanto para los modelos cuánticos como para los modelos clásicos, ya que se refiere a la proporción de predicciones correctas respecto al total de predicciones realizadas por el modelo. Esta métrica mide qué tan acertado es el modelo en sus predicciones.

En cuanto a la complejidad de tiempo para los modelos cuánticos, se calcula de manera similar a los modelos clásicos, pero se tiene en cuenta la naturaleza cuántica de las operaciones. Dado que las redes neuronales cuánticas implementan capas cuánticas en su arquitectura, la complejidad de tiempo se relaciona con la cantidad de compuertas cuánticas secuenciales ejecutadas en función del tamaño de la entrada [25]. Esta métrica refleja cuánto tiempo se necesita para realizar cálculos cuánticos y es importante para medir la eficiencia de los modelos cuánticos en comparación con sus contrapartes clásicas.

4.3.6. Qiskit y PennyLane

Qiskit y PennyLane son dos entornos de programación cuántica ampliamente utilizados en la comunidad científica.

Qiskit es un framework de código abierto desarrollado por IBM que permite a los investigadores y desarrolladores trabajar con computadoras cuánticas reales y simuladas. Ofrece una amplia gama de herramientas y bibliotecas para construir, simular y ejecutar algoritmos cuánticos. Qiskit es altamente versátil y se ha convertido en una de las principales opciones para la programación cuántica debido a su accesibilidad y documentación extensa[26].

Por otro lado, PennyLane es un entorno de programación cuántica desarrollado por Xanadu AI que se enfoca en la programación cuántica diferenciable. Esto significa que PennyLane está diseñado para integrarse de manera fluida con bibliotecas de aprendizaje automático y permitir la optimización de circuitos cuánticos como parte de algoritmos de aprendizaje profundo. Esta característica es especialmente útil para aplicaciones de machine learning cuántico, donde se requiere la optimización de circuitos cuánticos como parte de algoritmos de aprendizaje[27].

Los entornos de desarrollo Qiskit y PennyLane jugaron un papel fundamental en la realización de este trabajo de grado, ya que su uso facilitó significativamente la implementación y experimentación con modelos cuánticos para la detección de patrones en electrocardiogramas.

4.4. Métricas de Desempeño

A continuación, se describen las métricas utilizadas para evaluar el desempeño de los modelos desarrollados en el presente trabajo de grado. Estas métricas son fundamentales para comprender y comparar la capacidad de los modelos en la clasificación del complejo QRS en señales electrocardiográficas. A continuación, se detallan dos métricas clave: la complejidad de tiempo y la exactitud (*accuracy*). La complejidad de tiempo se refiere al tiempo necesario para procesar una señal de ECG a través de un modelo, mientras que la exactitud evalúa la proporción de predicciones correctas realizadas por los modelos en relación con el total de predicciones. Ambas métricas proporcionan información crucial sobre el rendimiento de los modelos y su idoneidad para aplicaciones prácticas.

4.4.1. Complejidad de Tiempo

La complejidad de tiempo es una métrica que se utiliza para medir el tiempo necesario para que un algoritmo o una técnica resuelva un problema o realice una tarea específica y es usualmente teniendo en cuenta el numero de operaciones elementales [28]. En el contexto de las Redes Neuronales Artificiales (RNA), la complejidad de tiempo se refiere al tiempo que lleva procesar una entrada a través de la red y obtener una salida.

Cálculo de la Complejidad de Tiempo en una RNA

Calcular la complejidad de tiempo en una RNA implica evaluar cuánto tiempo se tarda en realizar una inferencia o predicción para una entrada dada. Esto depende de varios factores, incluyendo el tamaño de la red (número de neuronas y capas), el tipo de operaciones matriciales realizadas en cada capa y la cantidad de operaciones necesarias para propagar la señal a través de la red.

Una forma general de calcular la complejidad de tiempo en una RNA sería mediante la multiplicación de matrices y operaciones de activación realizadas en cada capa, junto con el tiempo requerido para la propagación hacia adelante y hacia atrás durante el entrenamiento si corresponde[29].

Importancia de la Complejidad de Tiempo

La complejidad de tiempo es una métrica fundamental al comparar diferentes técnicas o modelos en la resolución de un mismo problema. En el contexto de la detección de complejos QRS en señales electrocardiográficas, es esencial considerar el tiempo necesario para procesar una señal ECG y obtener un resultado, ya que puede tener implicaciones prácticas significativas.

Una RNA que requiere un tiempo de procesamiento excesivamente largo puede no ser adecuada para aplicaciones en tiempo real o para análisis de grandes conjuntos de datos. Por lo tanto, al comparar diferentes técnicas para la detección de complejos QRS, es esencial evaluar su complejidad de tiempo y determinar cuál ofrece un equilibrio adecuado entre precisión y velocidad de procesamiento.

4.4.2. Exactitud (Accuracy)

La exactitud (accuracy) es una métrica utilizada en la evaluación del desempeño de un modelo de clasificación. Representa la proporción de predicciones correctas realizadas por el modelo en relación con el total de predicciones realizadas[30].

Cálculo de la Exactitud

La exactitud se calcula mediante la fórmula:

$$\text{Exactitud} = \frac{\text{Predicciones Correctas}}{\text{Total de Predicciones}}$$

Donde:

- **Predicciones Correctas**: Son las predicciones que coinciden con los valores reales.
- **Total de Predicciones**: Es el número total de predicciones realizadas por el modelo.

Importancia de la Exactitud como Métrica

La exactitud es una métrica fundamental en la evaluación de modelos de clasificación, ya que proporciona una visión general de su rendimiento. Mide la capacidad del modelo para clasificar correctamente las instancias en general, sin distinguir entre clases positivas y negativas. Una alta exactitud indica que el modelo es efectivo en la mayoría de las predicciones.

Matriz de Confusión

La exactitud se basa en la información contenida en una **matriz de confusión** [30]. Esta matriz es una herramienta que muestra el rendimiento de un modelo de clasificación y compara las predicciones del modelo con los valores reales. Generalmente, consta de cuatro valores:

- **Verdaderos Positivos (VP)**: Casos positivos clasificados correctamente.
- **Falsos Positivos (FP)**: Casos negativos clasificados incorrectamente como positivos.
- **Verdaderos Negativos (VN)**: Casos negativos clasificados correctamente.
- **Falsos Negativos (FN)**: Casos positivos clasificados incorrectamente como negativos.

La Figura 5, muestra la estructura de una matriz de confusión. Esta matriz permite visualizar y cuantificar la exactitud de las predicciones de un modelo al compararlas con los valores reales. Proporciona una representación clara de las predicciones correctas e incorrectas, lo que facilita la evaluación de la capacidad del modelo para clasificar diferentes clases o categorías en un problema dado.

		Predicción	
		Positivos	**Negativos**
Observación	**Positivos**	Verdaderos Positivos (VP)	Falsos Negativos (FN)
	Negativos	Falsos Positivos (FP)	Verdaderos Negativos (VN)

Figura 5

Función de activación simoide [31].

La matriz de confusión desempeñó un papel fundamental en la evaluación del rendimiento de los modelos desarrollados en el presente trabajo de grado. Se utilizó para evaluar el desempeño

de los tres modelos en el problema de clasificación del complejo QRS y resultó especialmente útil para calcular métricas como la exactitud. A través de la matriz de confusión, no solo fue posible identificar el número de predicciones correctas, sino también los tipos de errores cometidos por los modelos, tales como falsos positivos y falsos negativos. Esto proporcionó una visión completa del comportamiento de los modelos y ayudó a determinar su efectividad en la tarea de clasificación del complejo QRS.

5. Metodología

En esta sección se detalla el proceso llevado a cabo para cumplir con los objetivos propuestos en el presente trabajo de grado. La metodología se divide en cuatro subsecciones, cada una enfocada en un aspecto clave del trabajo realizado. En primer lugar, se describe el proceso de generación del conjunto de datos que incluyó, la elección de la base de datos, la selección de las señales relevantes, la segmentación y etiquetado de los segmentos de interés. A continuación, se presenta el desarrollo del modelo clásico basado en una red neuronal convolucional, que sirvió como punto de referencia para las comparaciones subsiguientes. La tercera subsección se centra en la creación del primer modelo híbrido, que combina elementos clásicos y cuánticos, integrando una capa cuántica de salida en la estructura del modelo. Finalmente, se aborda el segundo modelo híbrido, donde se realizó la operación de convolución en el dominio cuántico. En esta sección se proporcionan detalles exhaustivos sobre los métodos y procesos utilizados en la implementación de estos modelos, incluyendo el contexto y la explicación de las decisiones tomadas en el proceso de diseño.

5.1. Generación del Conjunto de Datos

En esta sección, se describe el proceso de generación del conjunto de datos utilizado en el presente trabajo de grado. El objetivo principal es crear un conjunto de datos coherente y estandarizado que sirva como base para evaluar los diferentes modelos de detección de patrones. Para lograr esto, se abordan diversos aspectos del proceso, comenzando con la elección de la base de datos, seguida de la selección de las señales de interés. Posteriormente, se describe la segmentación y etiquetado de los segmentos relevantes de las señales, así como

la selección de una estructura de datos unificada para su almacenamiento y procesamiento. Finalmente, se aborda la normalización de los datos para garantizar la coherencia en las entradas de los modelos. Cada aspecto de este proceso se aborda en detalle, brindando contexto y justificación para las decisiones tomadas en el diseño de la metodología.

En la Figura 6, se muestra un diagrama de flujo que visualiza el proceso de generación del conjunto de datos utilizado en este trabajo de grado. Este proceso, como se mencionó anteriormente, se divide en varias etapas clave. Comienza con la elección de la base de datos y la selección de las señales de interés. Luego, se abordan las etapas de segmentación y etiquetado de los segmentos relevantes de las señales, seguidas de la selección de una estructura de datos unificada para su almacenamiento y procesamiento. Finalmente, se detalla el proceso de normalización. Este diagrama de flujo proporciona una visión general de la metodología utilizada en la generación del conjunto de datos, el cual será empleado en el entrenamiento y evaluación de cada uno de los modelos desarrollados en el presente trabajo de grado.

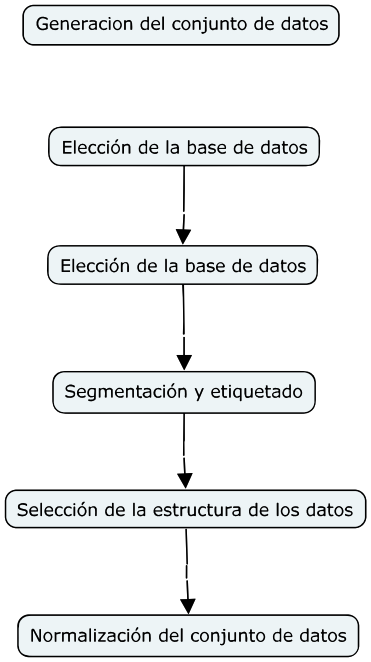

Figura 6

Diagrama de flujo: Generación del conjunto de datos.

5.1.1. Elección de la Base de Datos

La elección de la base de datos ECG-ID, como se describe en la Sección 4.1.6, se justifica por varios motivos. En primer lugar, esta base de datos se originó en el estudio de Tatiana S. Lugovaya, titulado "Biometric human identification based on electrocardiogram"[9], que tenía como objetivo la identificación biométrica basada en el electrocardiograma (ECG), lo que la hace relevante para el presente trabajo. ECG-ID contiene registros de ECG de 90 voluntarios, sin restricciones en cuanto a la frecuencia cardíaca, el estado físico o emocional, lo que proporciona una variedad de señales cardíacas para el análisis. Es importante destacar que, aunque esta base de datos ofrece una variedad de señales, en principio, los voluntarios no presentaban afecciones cardiacas, lo que sugiere que los datos son representativos de patrones cardíacos normales y no anómalos.

La elección de la derivación frontal D1 [5] es fundamental, ya que es comúnmente utilizada en estudios clínicos y proporciona señales de alta calidad para la detección de patrones cardíacos. Además, disponer de datos preprocesados facilita el análisis y la experimentación. En la Figura 7, se presenta una captura tomada de PhysioNet, donde se puede visualizar una de las grabaciones (señales) de la base de datos ECG-ID, tanto en su forma cruda como preprocesada.

5.1.2. Selección de las Señales de Interés

En esta etapa del proceso, se llevó a cabo la selección de las señales de interés de la base de datos ECG-ID. La elección de estas señales fue un paso crucial para asegurar la calidad y la idoneidad de los datos utilizados en el estudio. La base de datos original contenía una variedad de señales de electrocardiograma (ECG), pero no todas eran adecuadas para los propósitos de este trabajo. Se buscó identificar y seleccionar aquellas señales que presentaran un comportamiento normal, libre de anomalías y ruido excesivo. Para ilustrar este proceso, se presentarán tres tipos de señales: aquellas que representan un comportamiento deseado y están libres de ruido, las cuales serán seleccionadas para el presente trabajo de grado, como se muestra en la Figura 8; aquellas que muestran un comportamiento atípico, como

Figura 7

Representación de ECG crudo y preprocesado.

se observa en la Figura 9; y aquellas que contienen un nivel significativo de ruido, como se ilustra en la Figura 10, cada una con su respectiva explicación.

En la Figura 8, se presenta una captura de una señal ECG de la base de datos ECG-ID, tanto en su forma cruda como preprocesada. Este tipo de señales, que representan un comportamiento deseado y libre de ruido, se guardó para su utilización en el presente trabajo de grado.

Al igual que la figura anterior, la Figura 9 muestra una captura de una señal ECG de la base de datos ECG-ID, tanto en su forma cruda como preprocesada. Este tipo de señales, que muestra un comportamiento atípico en una señal ECG, incluyendo irregularidades o anormalidades, fue descartado.

Por otro lado, la Figura 10 también se basa en una captura de una señal ECG de la base de datos ECG-ID, tanto en su forma cruda como preprocesada. Este tipo de señales, que contiene un nivel significativo de ruido que dificultaría la clasificación precisa, también fue descartado.

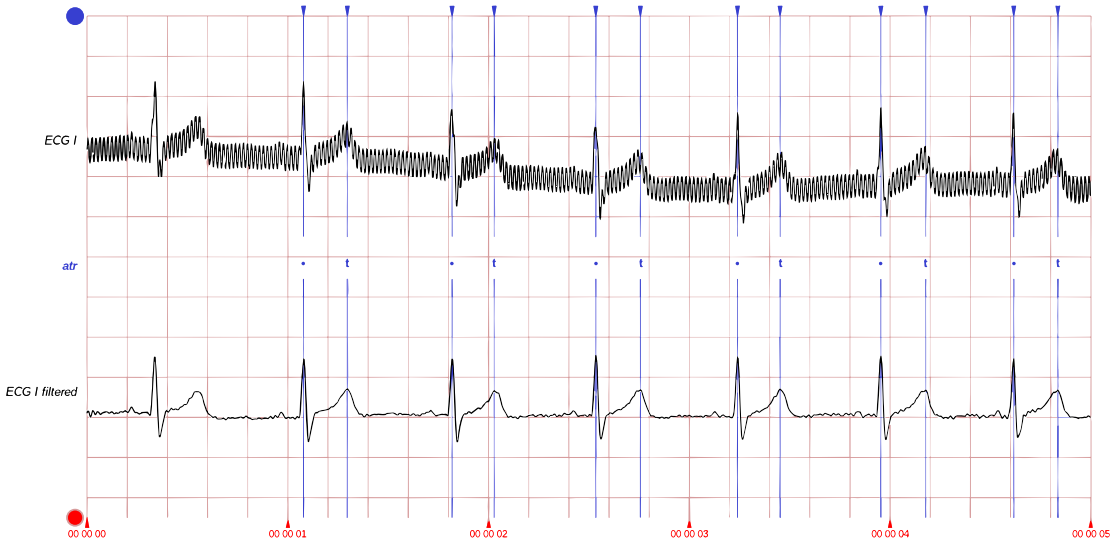

Figura 8

Representación de ECG con comportamiento normal.

5.1.3. Segmentación y Etiquetado

El proceso de segmentación y etiquetado se llevó a cabo de manera manual y constó de varios pasos esenciales. En primer lugar, se determinó la duración típica de un complejo QRS en el conjunto de datos ECG-ID, que resultó ser de 50 muestras de la señal. Dado que la frecuencia de muestreo de las señales fue de 500 Hz, esta duración equivale a 0.1 segundos (100 ms), lo que concuerda con la duración típica de un complejo QRS en señales electrocardiográficas.

Posteriormente, se procedió a realizar la segmentación de las señales. Esto implicó la extracción de fragmentos de señal de 0.1 segundos de duración (50 muestras) en puntos específicos de las grabaciones. Estos puntos incluyeron los complejos QRS, pero también se tomaron muestras de otros momentos en las señales seleccionadas en la sección anterior. Cada segmento se etiquetó de acuerdo con su contenido, especificando si contenía o no un complejo QRS.

Este proceso manual de segmentación y etiquetado desempeñó un papel fundamental en la preparación de los datos y permitió crear un conjunto de muestras correctamente etiquetadas que servirían como entrada para los modelos de detección de complejos QRS. La precisión

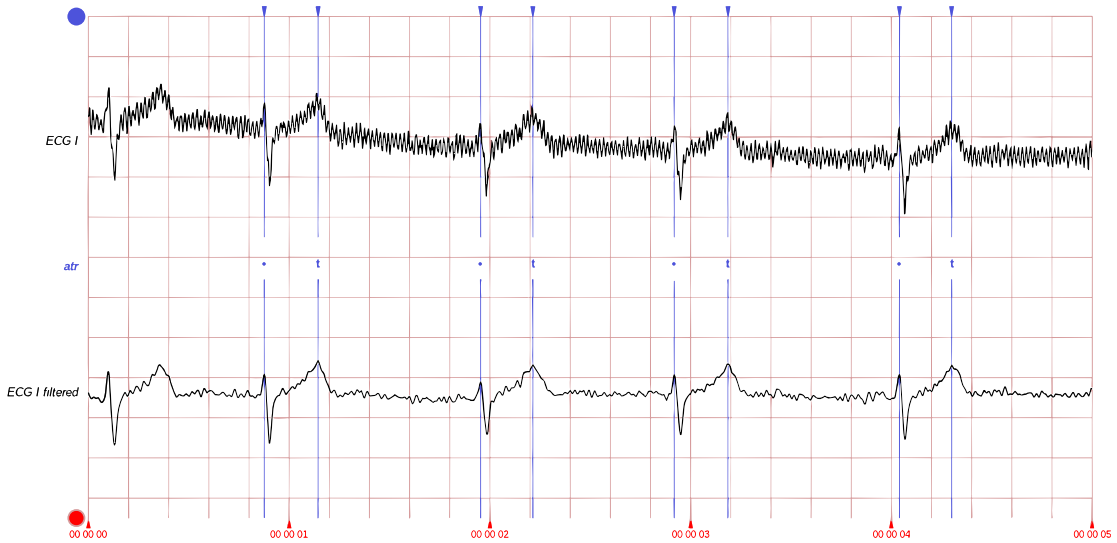

Figura 9

Representación de ECG con comportamiento atípico.

en la segmentación y etiquetado fue crucial para garantizar la calidad de los resultados y la capacidad de los modelos para identificar correctamente los complejos QRS en señales electrocardiográficas.

5.1.4. Selección de la Estructura de Datos

La selección de la estructura de datos es un aspecto crucial en el diseño de modelos de clasificación. En este trabajo, se tomó una decisión significativa con respecto a la estructura de datos que se utilizaría para la clasificación de señales electrocardiográficas. La elección se centró en si se debían procesar señales directamente o convertirlas en imágenes para su procesamiento posterior. Después de una cuidadosa evaluación, se optó por trabajar con señales en lugar de imágenes.

Esta decisión se basó en varias consideraciones importantes. En primer lugar, la elección de procesar señales en lugar de imágenes se relaciona con la capacidad de recursos de procesamiento necesarios para realizar la simulación de los modelos cuánticos. Procesar imágenes requeriría un número significativamente mayor de qubits, lo que aumentaría considerablemente la demanda de recursos de procesamiento y dificultaría la implementación práctica

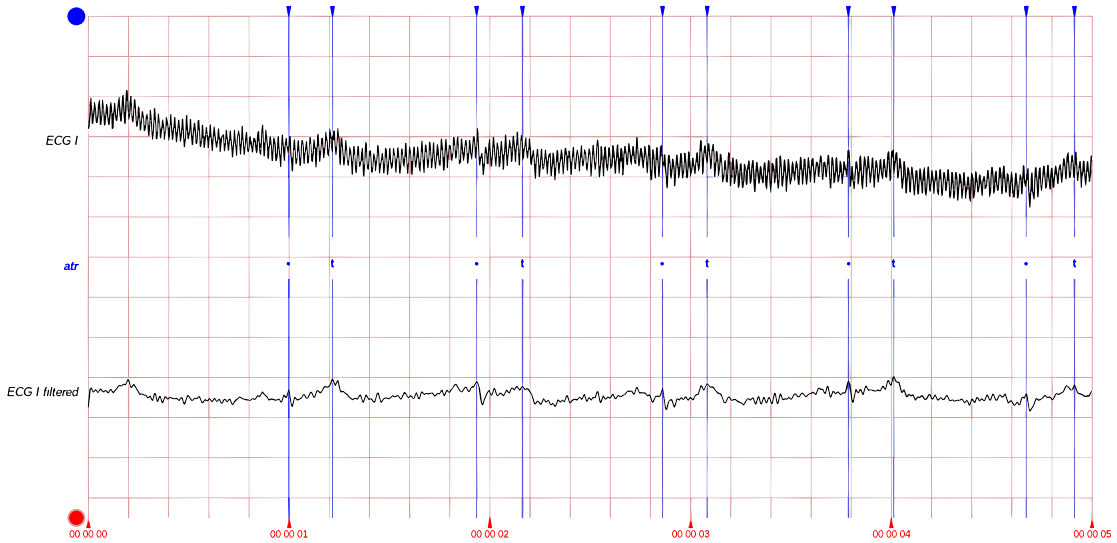

Figura 10

Representación de ECG con ruido significativo.

de los modelos cuánticos. Sin embargo, es importante tener en cuenta que este desafío es escalable y que en futuros trabajos se podría explorar la viabilidad de utilizar imágenes en un entorno cuántico.

Además, la selección de trabajar con señales se justifica desde el punto de vista de la utilidad de la información. Las imágenes generan una cantidad considerable de redundancia y contienen información que no es relevante para el proceso de clasificación, como las áreas en blanco alrededor de las señales. En contraste, trabajar directamente con señales permite una representación más eficiente de la información, centrándose en los aspectos cruciales de las señales electrocardiográficas.

5.1.5. Normalización

La normalización de los datos fue una etapa esencial en el proceso de preparación de los segmentos de señales electrocardiográficas para su posterior clasificación. Este proceso garantizó que todas las señales tuvieran una escala uniforme y comparativa, lo que facilitó la convergencia del modelo durante el entrenamiento y mejoró la capacidad del modelo para aprender patrones significativos en los datos.

La normalización se llevó a cabo en dos pasos principales. En la etapa inicial, se calcularon algunas medidas de tendencia central de los segmentos seleccionados, que incluyeron la media, la desviación estándar, así como los valores mínimo y máximo de todo el conjunto de datos. Estos cálculos proporcionaron una comprensión fundamental de la distribución de los datos y permitieron definir un rango de valores significativos.

Posteriormente, se aplicó la normalización restando a todos los segmentos el valor mínimo y dividiendo por la diferencia entre el valor máximo y el valor mínimo. Esta operación transformó todas las señales en un rango uniforme de valores entre 0 y 1, preservando las relaciones proporcionales entre los puntos de datos originales.

A continuación, en la Figura 11, se presenta una representación visual del proceso anteriormente descrito. Esta figura brinda una visión general del procedimiento utilizado para transformar las señales de ECG en un rango uniforme de valores entre 0 y 1.

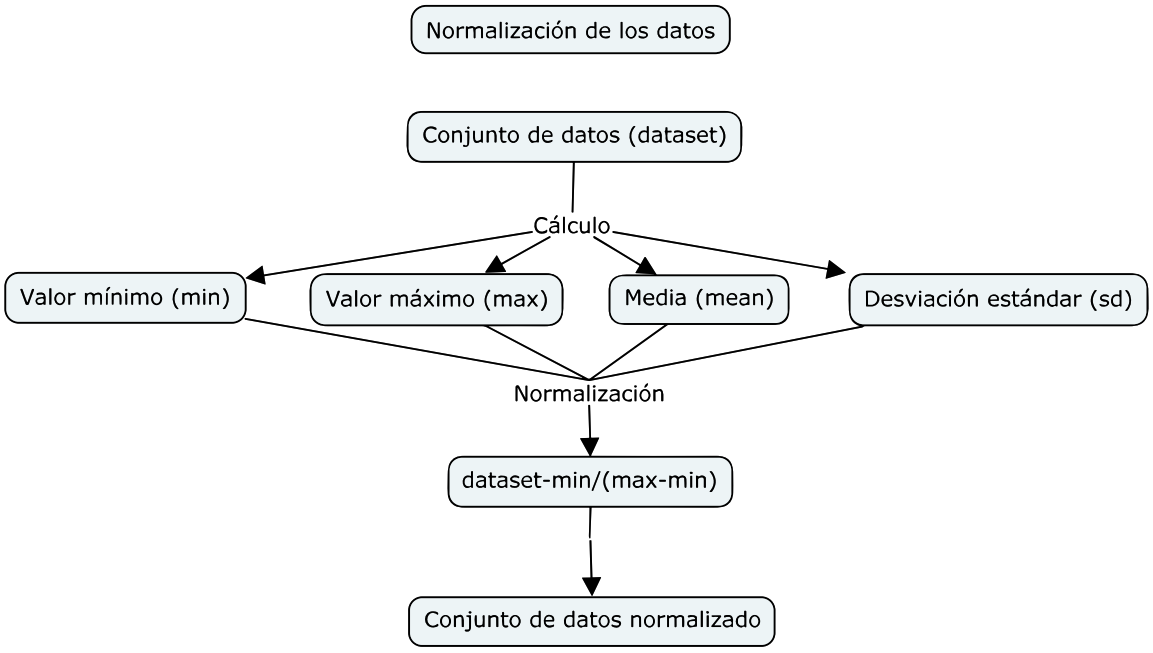

Figura 11

Diagrama de flujo: Normalización del conjunto de datos.

La importancia de trabajar con datos normalizados radicó en su capacidad para mejorar

la estabilidad y la eficiencia del proceso de entrenamiento de modelos de clasificación. La normalización elimina las diferencias en escala entre los datos, evitando que los modelos se vean dominados por características con valores más grandes, lo que podría haber llevado a un desempeño subóptimo o inestable. Además, la normalización facilitó la convergencia del modelo, lo que significa que el modelo pudo aprender más rápido y, en última instancia, tomar decisiones más precisas durante la clasificación.

5.2. Desarrollo del Modelo Clásico

En esta sección, se detallará el proceso de desarrollo del modelo clásico, que se estableció como la base para las comparaciones posteriores con los modelos híbridos cuánticos. La creación del modelo clásico involucró varias etapas cruciales, desde la preparación inicial del conjunto de datos hasta el diseño, entrenamiento y evaluación del modelo. Cada uno de estos pasos desempeñó un papel fundamental en la construcción de un modelo robusto y eficaz para la detección de complejos QRS en señales electrocardiográficas. A continuación, se describen en detalle las diversas fases del desarrollo del modelo clásico.

Adicionalmente, para una representación visual de todo el proceso de desarrollo del modelo clásico, se presenta el diagrama de flujo en la Figura 12. Este diagrama sintetiza e ilustra de manera esquemática las etapas clave involucradas en la creación y evaluación del modelo clásico para la detección de complejos QRS.

5.2.1. Preparación del Conjunto de Datos

La preparación del conjunto de datos fue una fase crítica en el desarrollo del modelo clásico para la detección de complejos QRS en señales electrocardiográficas. En esta etapa, se llevó a cabo la carga del conjunto de datos generado previamente, como se describió en la Sección 5.1. Luego de seleccionar las señales pertinentes de la base de datos ECG-ID y normalizarlas, se procedió a la división de este conjunto en dos partes: un conjunto de entrenamiento compuesto por el 80 por ciento de los datos y un conjunto de pruebas que comprendió el 20 por ciento restante. Este proceso estableció las bases para la construcción y evaluación del modelo clásico, que se detallará en las siguientes etapas.

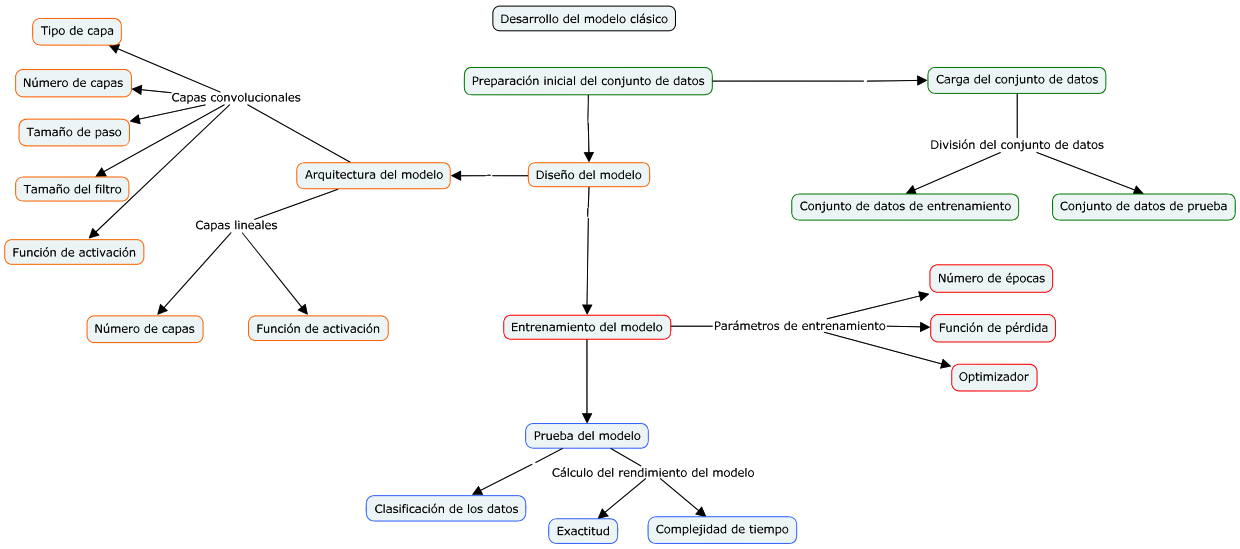

Figura 12

Diagrama de flujo: Desarrollo del modelo clásico.

A continuación, en la Figura 13, se presenta una captura que muestra una selección aleatoria de seis segmentos del conjunto de prueba, junto con sus etiquetas correspondientes. En esta representación visual, los segmentos etiquetados con 1 corresponden a aquellos que contienen el complejo QRS, mientras que los segmentos etiquetados con 0 son aquellos que no lo contienen. Cabe destacar que en cada una de estas muestras, el eje vertical corresponde al valor normalizado de voltaje de la señal electrocardiográfica, dentro de un rango de 0 a 1. Además, el eje horizontal representa las muestras, lo que resulta en un dominio de 0 a 50, dado que cada segmento está compuesto por 50 muestras, lo que equivale a 0.1 segundos de la señal electrocardiográfica.

5.2.2. Diseño del Modelo Clásico

En esta sección, se detalla la arquitectura del modelo clásico utilizado para la detección de complejos QRS en señales electrocardiográficas. Se explican las decisiones clave de diseño que han guiado la construcción de esta red neuronal, como el tamaño del kernel, el número de capas convolucionales y lineales, entre otros aspectos relevantes. En la Figura 14, se presenta una captura tomada del cuaderno de Jupyter que muestra la estructura del modelo.

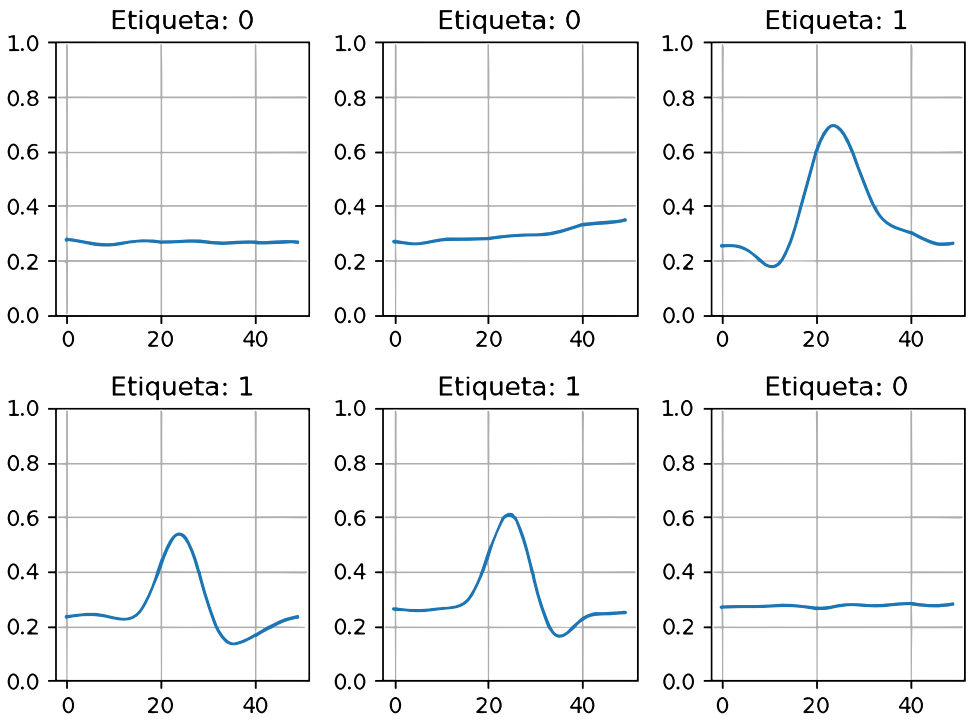

Figura 13

Muestra de algunos segmentos del conjunto de prueba con su etiqueta respectiva.

La estructura del modelo clásico se basa en una serie de capas, cada una con sus características específicas para el procesamiento de señales electrocardiográficas. A continuación, se enlistan las capas utilizadas en el diseño del modelo junto con sus características:

- Capa convolucional 1D (conv1): Esta capa cuenta con 16 canales de salida y utiliza un filtro (kernel) de tamaño 5 y un paso (stride) de 1 para realizar convoluciones en las señales de entrada. La elección del tamaño del filtro (kernel size) se fundamenta en la duración del pico R del complejo QRS en las señales electrocardiográficas, ya que un filtro de tamaño 5 captura eficazmente esta característica importante. La decisión de tener 16 canales de salida se basa en la capacidad del modelo para aprender múltiples representaciones de las señales. El paso de 1 se elige para asegurar que se convolucione toda la longitud de la señal de entrada sin pérdida de información. Además, se utiliza la función de activación ReLU (Rectified Linear Unit) después de la capa convolucional.

```
class Net(Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = Conv1d(in_channels=1, out_channels=16, kernel_size=5, stride=1)
        self.conv2 = Conv1d(in_channels=16, out_channels=1, kernel_size=5, stride=1)
        self.dropout = Dropout1d()
        self.num_features = self._calculate_num_features()
        self.fc1 = Linear(self.num_features, 64)
        self.fc2 = Linear(64, 8)
        self.fc3 = Linear(8,1)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.relu(self.conv2(x))
        x = self.dropout(x)
        x = x.view(x.shape[0], -1)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        x = self.fc3(x)
        x = torch.sigmoid(x)
        return cat((x, 1 - x), -1)
```

Figura 14

Sección del código en la que se define la estructura del modelo clásico.

La elección de ReLU se debe a su capacidad para introducir no linealidad en el modelo y su eficiencia computacional en comparación con otras funciones de activación. Esta no linealidad es esencial para que el modelo capture relaciones complejas en los datos. La convolución en 1D se realiza debido a la estructura unidimensional de los datos de entrada, que son señales electrocardiográficas registradas a lo largo del tiempo.

- Capa convolucional 1D (conv2): Similar a la capa anterior, esta capa continúa extrayendo características con 1 canal de salida, un filtro (kernel) de tamaño 5 y un paso (stride) de 1. La elección de tener solo 1 canal de salida se realizó con el objetivo de reducir el número de características y, por lo tanto, disminuir el costo computacional del modelo. Al igual que en la capa convolucional anterior, se utiliza la función de activación ReLU (Rectified Linear Unit) después de esta capa para introducir no linealidad y capturar relaciones complejas en los datos.

- Capa lineal (fc1): Una capa completamente conectada que procesa las características extraídas por las capas convolucionales. El objetivo principal de esta capa es reducir

aún más el número de características, manteniendo solo las más relevantes para la clasificación final. Este proceso de reducción de dimensiones contribuye a simplificar la representación de las señales electrocardiográficas, lo que puede mejorar la eficiencia y efectividad del modelo.

- Capa lineal (fc2): Esta capa completamente conectada refina las características extraídas previamente por las capas convolucionales. Su propósito es mejorar la representación de las características, permitiendo al modelo aprender patrones más complejos en las señales electrocardiográficas, lo que puede mejorar la detección de complejos QRS y la capacidad de generalización.

- Capa lineal (fc3): Esta capa de salida produce las predicciones finales del modelo. Utiliza una función de activación sigmoide en su salida para realizar la clasificación de los complejos QRS y no-QRS, asignando una probabilidad a cada clase.

La Figura 15, presenta una representación visual de la arquitectura del modelo clásico ya descrita, que consta de diversas capas. Esta arquitectura sienta las bases para los modelos híbridos y comparte muchas de las justificaciones de diseño ya mencionadas, que son fundamentales para lograr un rendimiento óptimo en la clasificación de complejos QRS.

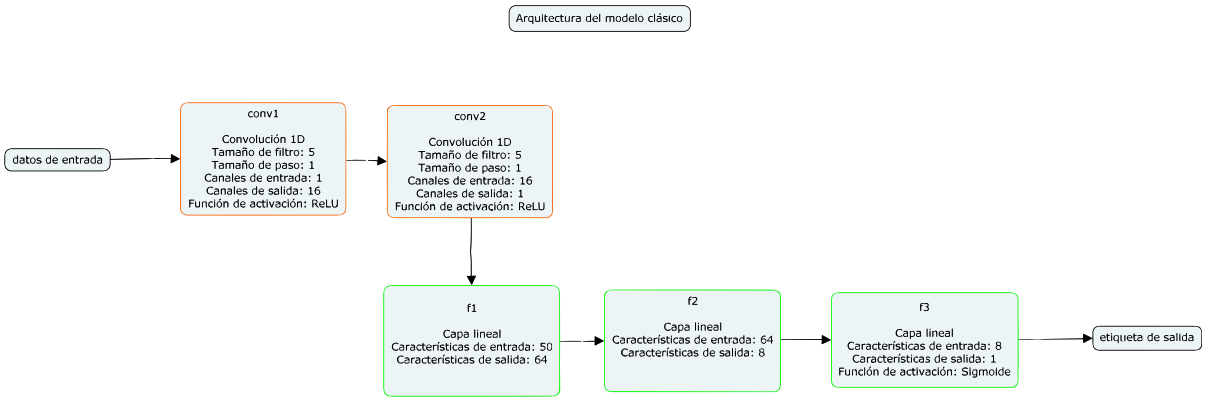

Figura 15

Representación gráfica de la arquitectura del modelo clásico.

5.2.3. Entrenamiento del Modelo Clásico

El entrenamiento desempeña un papel fundamental en la construcción de un sistema de detección de complejos QRS en señales electrocardiográficas. En esta sección, se profundizará en las decisiones críticas que dieron forma al proceso de entrenamiento, destacando los aspectos clave que se tuvieron en cuenta.

Uno de los aspectos cruciales al entrenar un modelo de clasificación es la elección de la función de pérdida. En este contexto, se optó por la función de pérdida Negative Log-Likelihood Loss (NLLLoss) [16]. Esta elección se fundamenta en la naturaleza de la tarea de clasificación binaria que implica distinguir entre la presencia o ausencia de complejos QRS en las señales. La NLLLoss es adecuada para problemas de clasificación y se encarga de medir la diferencia entre las probabilidades calculadas por el modelo y las etiquetas reales de los datos de entrenamiento. Al minimizar esta función de pérdida, el modelo se ajusta de manera óptima para realizar predicciones precisas.

En cuanto al optimizador, se eligió Adam [17] debido a su eficacia demostrada en la optimización de redes neuronales. Adam combina los beneficios de dos técnicas populares, el descenso de gradiente estocástico (SGD) con momentum y RMSprop [17]. Esta elección se basó en su capacidad para converger rápidamente y evitar los problemas de estancamiento en el entrenamiento. Adam ajusta automáticamente las tasas de aprendizaje para cada parámetro del modelo, lo que resulta en un proceso de entrenamiento más eficiente y estable.

Respecto al número de épocas de entrenamiento, se determinó que un total de 50 épocas sería adecuado para alcanzar un equilibrio entre el aprendizaje del modelo y el tiempo de entrenamiento. Si bien es posible continuar el entrenamiento durante más épocas, esto podría llevar a un sobreajuste del modelo a los datos de entrenamiento. Por otro lado, un número insuficiente de épocas podría resultar en un modelo subentrenado que no sea capaz de capturar patrones complejos. Las 50 épocas se consideraron suficientes para lograr un modelo efectivo sin incurrir en un alto costo computacional.

El entrenamiento del modelo clásico se llevó a cabo a través del código representado en la

Figura 16. En este código, se definió la función de pérdida (loss func) utilizando la pérdida logarítmica negativa (NLLLoss). Se configuró el modelo para entrenar durante 50 épocas (epochs), lo que determina el número de veces que el modelo recorre todo el conjunto de entrenamiento. Durante el proceso de entrenamiento, se utilizó el optimizador Adam para ajustar los pesos del modelo y minimizar la pérdida.

El bucle de entrenamiento consiste en iterar a través de las épocas y los lotes de datos (batch) con la ayuda de un bucle 'for'. En cada época, se realiza un pase hacia adelante (forward pass) al pasar los datos a través del modelo, lo que genera una salida. A continuación, se calcula la pérdida comparando la salida del modelo con las etiquetas reales del conjunto de entrenamiento. El proceso de retropropagación (backward pass) ajusta los pesos del modelo para reducir la pérdida. Luego, el optimizador optimiza los pesos del modelo para la siguiente iteración.

```
optimizer = optim.Adam(model.parameters(), lr=0.001)

# Start training
epochs = 50  # Set number of epochs
loss_list = []  # Store loss history
model.train()  # Set model to training mode

for epoch in range(epochs):
    total_loss = []
    for batch_idx, (data, target) in enumerate(train_loader):
        optimizer.zero_grad(set_to_none=True)  # Initialize gradient
        output = model(data)  # Forward pass
        loss = loss_func(output, target)  # Calculate loss
        loss.backward()  # Backward pass
        optimizer.step()  # Optimize weights
        total_loss.append(loss.item())  # Store loss
    loss_list.append(sum(total_loss) / len(total_loss))
    print("Training [{:.0f}%]\tLoss: {:.4f}".format(100.0 * (epoch + 1) / epochs, loss_list[-1]))
```

Figura 16

Sección del código en la que se realiza el entrenamiento del modelo clásico.

En la Figura 17, se presenta un gráfico que ilustra la evolución del error a lo largo de las épocas durante el entrenamiento del modelo clásico. El eje vertical representa la medición del NLLLoss, que es una medida de la pérdida de clasificación del modelo. Esta pérdida cuantifica qué tan cerca o lejos están las predicciones del modelo de las etiquetas reales du-

rante el entrenamiento. De esta manera, a medida que se obtienen mediciones de NLLLoss menores, se interpreta como que el modelo está mejorando en la tarea de clasificación. El gráfico es esencial para comprender cómo el modelo mejora su capacidad para hacer predicciones precisas a medida que se ajusta a los datos de entrenamiento. Se observa que el valor de NLLLoss disminuye significativamente en las primeras épocas y luego converge. Este fenómeno de convergencia es un indicador positivo de que el modelo está aprendiendo y ajustándose a los patrones presentes en los datos.

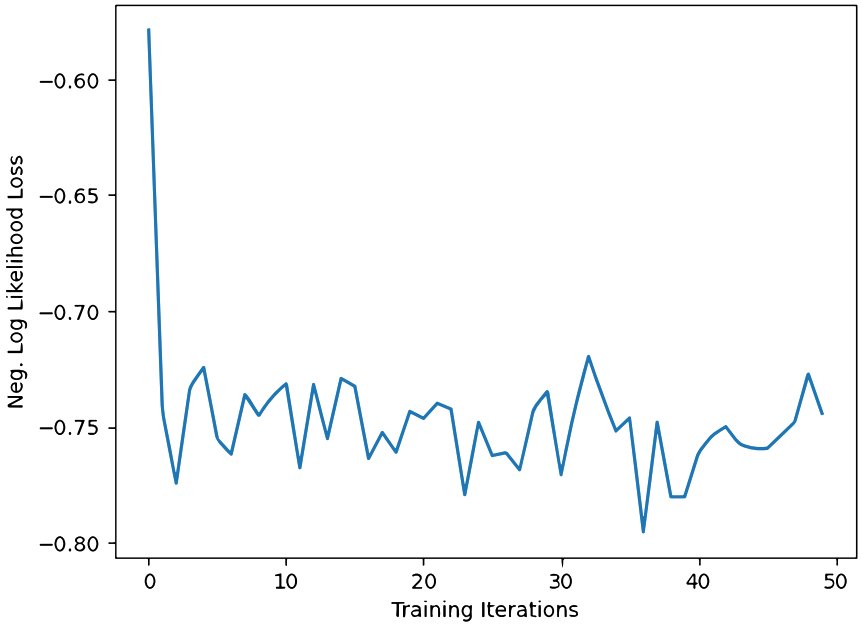

Figura 17

Grafico de epocas vs NLLL en el entrenamiento clasico.

5.2.4. Prueba del Modelo Clásico

En esta etapa, se llevó a cabo la evaluación del modelo clásico, la cual involucró la carga del modelo previamente entrenado, el procesamiento y clasificación de los datos del conjunto de prueba. Es importante destacar que, durante esta fase, se desactivó el cálculo del gradiente para asegurar que los pesos del modelo no se modificaran durante la evaluación. Luego, se procedió al procesamiento de los datos, que se logró al pasar cada segmento del conjunto de prueba a través del modelo ya entrenado. Finalmente, se compararon las etiquetas predichas

con las etiquetas reales para evaluar su correspondencia.

La evaluación del modelo clásico se llevó a cabo mediante el código presentado en la Figura 18. En esta etapa de evaluación, el modelo previamente entrenado se carga en la variable 'clasicLoadedModel'. Para garantizar una evaluación justa, el modelo se establece en modo de evaluación mediante 'clasicLoadedModel.eval()', y se desactiva el cálculo de gradientes. La evaluación se lleva a cabo en un bucle 'for', donde se procesan los segmentos del conjunto de prueba uno por uno. El modelo realiza predicciones en cada segmento y se compara la etiqueta predicha con la etiqueta real. La exactitud de las predicciones se registra en la variable 'correct', y se calcula el valor de la pérdida utilizando la función de pérdida NLLLoss. Los resultados se almacenan y, adicionalmente, se muestra una representación visual de seis segmentos aleatorios del conjunto de prueba junto con sus etiquetas reales y etiquetas predichas. Esto permite visualizar cómo el modelo realizó la tarea de clasificación.

En la Figura 19, se presentan seis ejemplos de clasificación tomados de manera aleatoria del conjunto de datos de prueba, donde se muestra la etiqueta predicha por el modelo junto con la etiqueta real correspondiente. Esta representación visual proporciona una clara visión del rendimiento del modelo en la clasificación de los complejos QRS en señales electrocardiográficas. En cada una de estas representaciones, el eje vertical refleja el valor normalizado de voltaje de la señal electrocardiográfica, dentro de un rango de 0 a 1, de la misma forma en la que se ilustró en la Figura 13. Mientras que el eje horizontal representa las muestras, con un dominio de 0 a 50, ya que cada segmento está compuesto por 50 muestras, equivalente a 0.1 segundos de la señal electrocardiográfica.

La correspondencia entre las etiquetas reales y las etiquetas predichas en estos ejemplos es fundamental, ya que cuando coinciden, indica que el modelo clásico está realizando predicciones correctas. Esto demuestra la eficacia del modelo en la tarea de detección de los complejos QRS en las señales ECG, lo que es esencial para garantizar resultados confiables en aplicaciones médicas y de monitoreo cardíaco.

```
clasicLoadedModel = Net()
clasicLoadedModel.load_state_dict(torch.load("clasicmodel.pt"))

clasicLoadedModel.eval()
with no_grad():

    correct = 0
    for batch_idx, (data, target) in enumerate(test_loader):
        output = clasicLoadedModel(data)
        if len(output.shape) == 1:
            output = output.reshape(1, *output.shape)

        pred = output.argmax(dim=1, keepdim=True)
        correct += pred.eq(target.view_as(pred)).sum().item()

        loss = loss_func(output, target)
        total_loss.append(loss.item())

        if (batch_idx<6):
            plt.subplot(2, 3, batch_idx+1)
            plt.grid()
            plt.title("Etiqueta: {}, Predicha: {}".format(target.numpy()[0], pred.numpy()[0][0]))
            plt.ylim(0,1)
            plt.plot(data.numpy()[0])
            plt.tight_layout()
    print(
        "Performance on test data:\n\tLoss: {:.4f}\n\tAccuracy: {:.1f}%".format(
            sum(total_loss) / len(total_loss), correct / len(test_loader) / 1 * 100
        )
    )
```

Figura 18

Sección del código en la que se realiza la prueba del modelo clásico.

5.3. Desarrollo del Modelo Híbrido con Capa de Salida Cuántica

En esta sección, se explorará el desarrollo de un modelo híbrido en el que se ha integrado una capa cuántica en la salida del modelo clásico previamente descrito mediante el uso de la clase 'TorchConnector' [32] de la librería qiskit. Aunque la preparación, el entrenamiento y la prueba mantienen similitudes con el modelo clásico, el enfoque principal estará centrado en el diseño del modelo híbrido y en cómo la capa cuántica ha sido incorporada en la arquitectura existente. Se explicará en detalle el concepto de mapa de características y ansatz, que son elementos esenciales de esta propuesta.

Adicionalmente, para obtener una representación visual del proceso de desarrollo del modelo híbrido con capa de salida cuántica, se ha incluido el diagrama de flujo presentado en la Figura 20. Este diagrama sintetiza y ilustra de manera esquemática las etapas clave involucradas

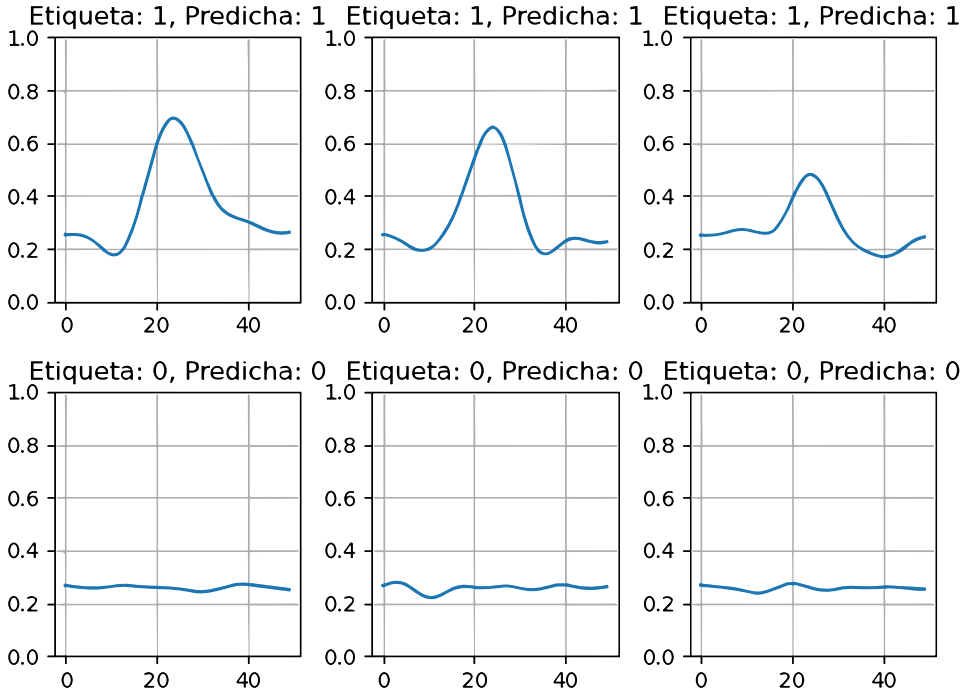

Figura 19

Muestra de segmentos clasificados por el modelo clasico.

en la creación y evaluación de este modelo híbrido. Es importante destacar la similitud de esta figura con la Figura 12, dado que este modelo se basa en el modelo clásico previamente descrito. Sin embargo, en el diagrama se evidencia claramente la integración de la capa de salida cuántica en el diseño del modelo

5.3.1. Diseño del Modelo Híbrido con Capa de Salida Cuántica

A continuación, se abordará el diseño del modelo híbrido, en el cual se incorpora una capa cuántica en la salida del modelo clásico previamente desarrollado, como se ilustra en la Figura 21. En esta figura, se presenta un fragmento de código que describe la arquitectura del modelo híbrido, destacándose la presencia de la capa 'TorchConnector' [32] en la salida de la red, la cual está compuesta por dos circuitos cuánticos parametrizables. El primero, conocido como 'mapa de características', recibe la información de salida del modelo clásico como parámetros de entrada. El segundo, denominado 'Ansatz', tiene como parámetros de entrada los pesos que serán modificados durante el entrenamiento para mejorar la tarea de

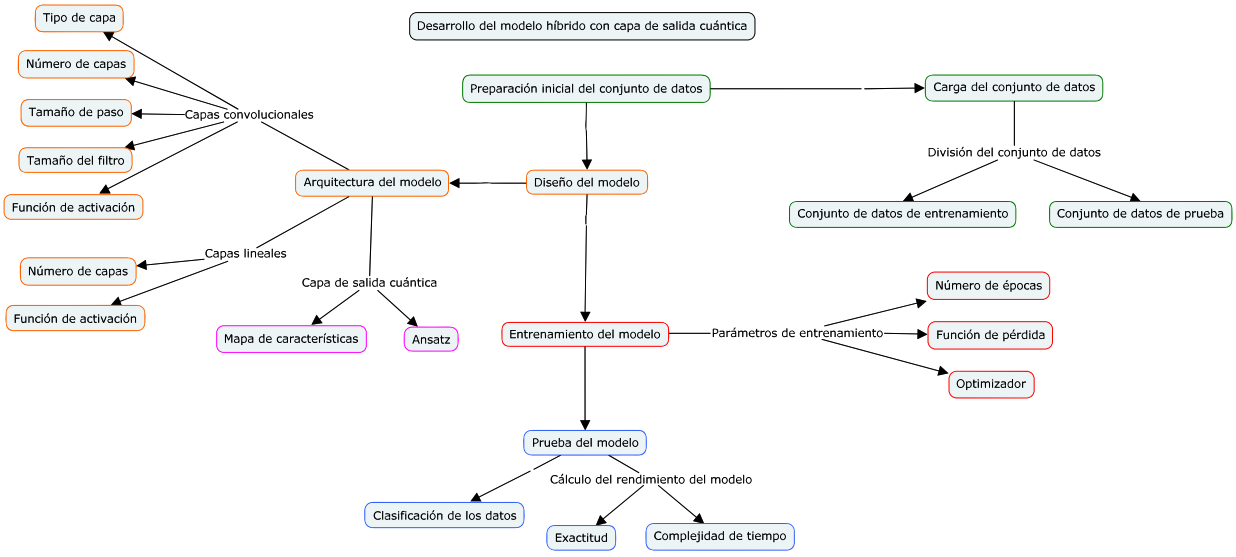

Figura 20

Diagrama de flujo: Desarrollo del modelo híbrido con capa de salida cuántica.

clasificación. Posteriormente, se explorará en detalle la funcionalidad y el propósito de esta capa en el contexto del modelo híbrido.

5.3.2. Circuito Parametrizable

El circuito cuántico parametrizable fue un componente esencial en el diseño de la red híbrida con capa de salida cuántica. Se caracteriza por su su capacidad única para ajustar dinámicamente sus parámetros, que alteran el estado de los qubits y, por lo tanto, la información que representan. Estos parámetros pueden desempeñar funciones específicas en la transformación de los datos, lo que a su vez influye en el proceso de clasificación de las señales electrocardiográficas. La adaptabilidad de este circuito es fundamental, ya que permite que el modelo aprenda y represente patrones complejos de manera efectiva.

En la Figura 22, se muestra un fragmento de código donde se define la "quantum neural network"(QNN), que constituye la capa de salida del modelo híbrido. Este QNN se encarga de transformar y procesar la información proporcionada por la red con el fin de realizar la clasificación final de los complejos QRS en las señales electrocardiográficas. El circuito cuántico se compone de dos partes: un primer circuito parametrizable llamado feature map

```
class Net(Module):
    def __init__(self):
        super(Net, self).__init__()
        self.conv1 = Conv1d(in_channels=1, out_channels=16, kernel_size=5, stride=1)
        self.conv2 = Conv1d(in_channels=16, out_channels=1, kernel_size=5, stride=1)
        self.dropout = Dropout1d()
        self.num_features = self._calculate_num_features()
        self.fc1 = Linear(self.num_features, 64)
        self.fc2 = Linear(64, 8)
        self.fc3 = Linear(8,2)
        self.qnn = TorchConnector(qnn)
        self.fc4 = Linear(1, 1)

    def forward(self, x):
        x = F.relu(self.conv1(x))
        x = F.relu(self.conv2(x))
        x = self.dropout(x)
        x = x.view(x.shape[0], -1)
        x = F.relu(self.fc1(x))
        x = self.fc2(x)
        x = self.fc3(x)
        x = self.qnn(x)
        x = self.fc4(x)
        x = torch.sigmoid(x)
        return cat((x, 1 - x), -1)
```

Figura 21

El fragmento de código donde se define la arquitectura de la red híbrida con capa de salida cuántica.

y un segundo circuito parametrizable denominado ansatz. A continuación, se explicará la función de cada uno de estos componentes en el proceso de clasificación.

5.3.3. Mapa de Características: ZZFeatureMap

El ZZFeatureMap, (mapa de características ZZ), fue un componente esencial en la arquitectura de la red híbrida con capa de salida cuántica. Su función principal consistió en transformar la información clásica proporcionada por la red neuronal clasica en un formato adecuado para su procesamiento en un circuito cuántico.

Este mapa de características utilizó una serie de transformaciones cuánticas, incluida una transformación Pauli-Z de segundo orden, entre los qubits que representaban las características de entrada de la información clásica. Estas transformaciones introdujeron interacciones cuánticas entre los qubits, lo que permitió codificar información compleja en la correlación de los estados cuánticos.

En la Figura 23, se presenta una representación visual del mapa de características denominado 'zzfeaturemap'. En esta figura, se puede observar el proceso de inicialización de los qubits mediante la aplicación de compuertas H que preparan los qubits en un estado adecuado.

```python
def create_qnn():
    feature_map = ZZFeatureMap(2)
    ansatz = RealAmplitudes(2, reps=1)

    qc = QuantumCircuit(2)
    qc.compose(feature_map, inplace=True)
    qc.compose(ansatz, inplace=True)

    qnn = EstimatorQNN(
        circuit=qc,
        input_params=feature_map.parameters,
        weight_params=ansatz.parameters,
        input_gradients=True,
    )
    return qnn

qnn = create_qnn()
```

Figura 22

El fragmento de código donde se define el circuito cuantico qnn.

Luego, se introduce la información de salida del modelo clásico utilizando compuertas P con parámetros de entrada x[0] y x[1]. A continuación, se realiza el entrelazamiento de ambos qubits, seguido de otra rotación P con los mismos parámetros de entrada x[0] y x[1], para finalizar con un último paso de entrelazamiento.

En esencia, el ZZFeatureMap facilitó la transición de la información desde el dominio clásico al cuántico, preparándola para su procesamiento posterior en el contexto de la red híbrida y permitiendo la clasificación efectiva de los complejos QRS en las señales electrocardiográficas.

5.3.4. Ansatz: RealAmplitudes

El Ansatz en este contexto desempeña un papel crucial en la transformación de la información cuántica proporcionada por el ZZFeatureMap en las salidas finales que se utilizan para la clasificación de los complejos QRS. Este componente parametrizable está compuesto por una serie de compuertas cuánticas que aplican rotaciones controladas por los pesos obtenidos durante el entrenamiento del modelo.

El Ansatz actúa como una capa intermedia entre el ZZFeatureMap y la capa de medición final del circuito cuántico, y su función es ajustar dinámicamente sus parámetros para realizar

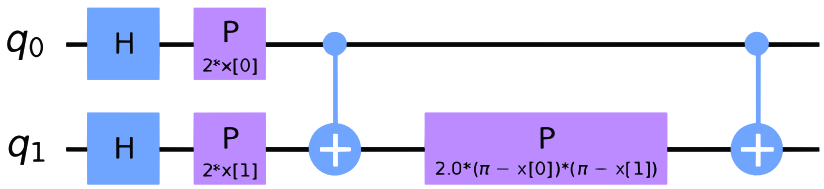

Figura 23

Representación visual del mapa de características: ZZFeatureMap.

las rotaciones necesarias en los qubits y, en última instancia, transformar la información cuántica en una representación adecuada para la tarea de clasificación.

El diseño del Ansatz se basó en consideraciones cuidadosas sobre la cantidad de parámetros necesarios para representar de manera efectiva las transformaciones requeridas, equilibrando la capacidad del modelo para capturar patrones complejos sin caer en la sobreparametrización. En consonancia con esta estrategia, el circuito 'RealAmplitudes' se configuró con una sola repetición del mismo, lo que se consideró suficiente para finalizar la tarea de clasificación del complejo QRS. Esto garantiza una representación eficaz sin una excesiva complejidad de parámetros.

En la Figura 24, se exhibe una representación visual del Ansatz 'RealAmplitudes'. Este circuito cuántico inicia su operación mediante la aplicación de compuertas de rotación Ry con parámetros de entrada w[0] y w[1], que corresponden a los pesos entrenables del modelo. Posteriormente, se efectúa un proceso de entrelazamiento entre los qubits, seguido por otra capa de compuertas de rotación Ry. En esta ocasión, las compuertas tienen parámetros de entrada w[2] y w[3], que también son pesos modificables durante el entrenamiento.

En conjunto, los circuitos 'ZZFeatureMap' y 'RealAmplitudes' conforman el componente cuántico esencial en la preparación y procesamiento de la información cuántica antes de la etapa final de clasificación. Estos circuitos se encuentran definidos dentro de la biblioteca 'QML' de qiskit.

La Figura 25 proporciona una representación visual de la arquitectura del modelo híbrido con capa de salida cuántica. Observarás que esta figura presenta una similitud notable con

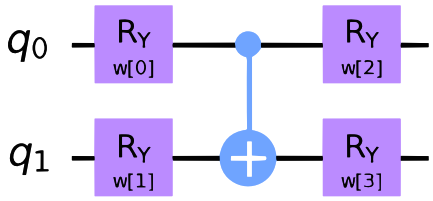

Figura 24

Representación visual del Ansatz: RealAmplitudes.

la Figura 15, que representa la arquitectura del modelo clásico. Sin embargo, la distinción principal en la Figura 25 es la inclusión de la capa de salida cuántica, que constituye un elemento crucial de este enfoque híbrido.

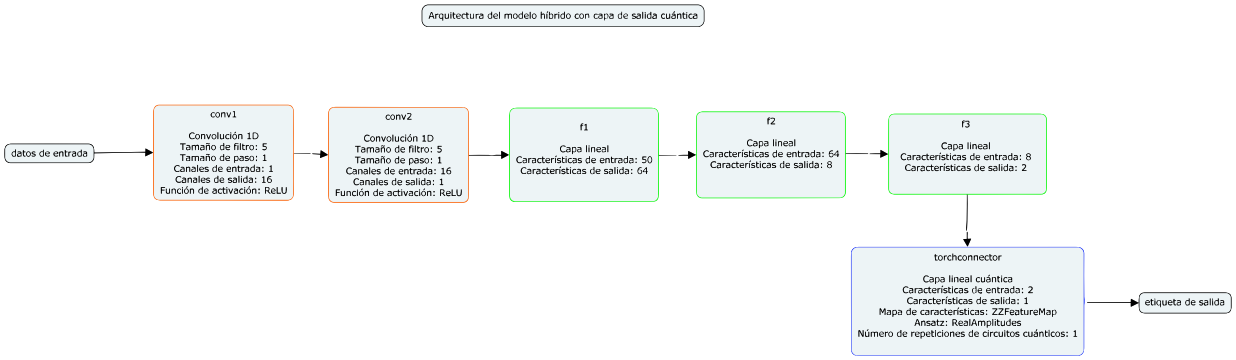

Figura 25

Representación gráfica de la arquitectura del modelo híbrido con capa de salida cuántica.

5.4. Desarrollo del Modelo híbrido con convolución en el dominio cuántico

En la presente sección, se describe el modelo híbrido con convolución en el dominio cuántico. En gran medida, el proceso de preparación de datos, entrenamiento y prueba se mantuvo sin cambios en comparación con el modelo clásico. Sin embargo, se incorporó una etapa inicial que implica la transformación de los datos a través de la operación de convolución en el dominio cuántico. Posteriormente, se ejecuta la parte lineal del modelo clásico dentro de esta arquitectura híbrida. Esto permite una integración más fluida de técnicas cuánticas en el

flujo de trabajo del modelo, lo que potencialmente puede mejorar su capacidad de detección de complejos QRS en las señales electrocardiográficas. A continuación, se profundizará en la operación de convolución en el dominio cuántico y cómo ha sido integrada en el diseño del modelo híbrido para la clasificación del complejo QRS en señales electrocardiográficas.

Adicionalmente, se introduce la Figura 26, que representa un diagrama de flujo que sintetiza el proceso de desarrollo del modelo híbrido con convolución en el dominio cuántico. Esta figura proporciona una visión clara de cómo el desarrollo del modelo híbrido difirió del desarrollo del modelo clásico, ilustrado en la Figura 12, destacando que en el primero, la etapa de convolución ya no forma parte del diseño del modelo. En cambio, la convolución en el dominio cuántico se lleva a cabo en la etapa de preparación inicial de los datos, antes del entrenamiento. Esto implica que la convolución se realiza sobre todo el conjunto de datos de manera previa al proceso de entrenamiento.

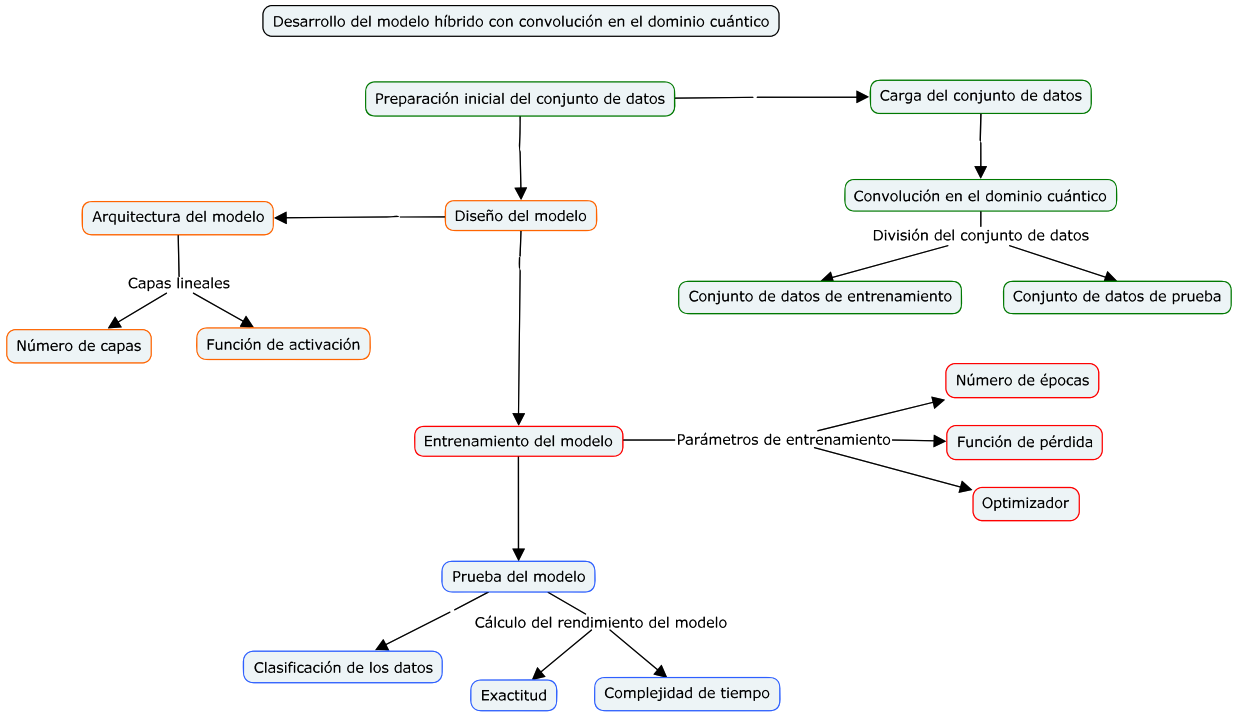

Figura 26

Diagrama de flujo: Desarrollo del modelo híbrido conconvolución en el dominio cuántico.

Finalmente, es importante destacar que en el desarrollo del modelo híbrido con convolución

en el dominio cuántico, se emplearon dos librerías fundamentales: TensorFlow y PennyLane. PennyLane se utilizó específicamente para llevar a cabo la convolución en el dominio cuántico. Esta elección se fundamenta en que PennyLane se encuentra más enfocada que la librería qiskit en el aprendizaje de máquina cuántico, proporcionando herramientas y funcionalidades específicas para esta área. Por otro lado, TensorFlow se utilizó para el entrenamiento del modelo. La razón detrás de esta elección radica en la facilidad de integración entre TensorFlow y PennyLane, lo que permitió un proceso de desarrollo eficiente y efectivo. La combinación de ambas librerías ha sido crucial para la implementación exitosa de un modelo híbrido capaz de aprovechar las ventajas de la computación cuántica en la detección de complejos QRS en señales electrocardiográficas.

5.4.1. Transformación de los Datos al Dominio Cuántico

Como se puede observar en la Figura 13, una vez que se ha cargado el conjunto de datos, se tienen a disposición segmentos de señales electrocardiográficas, cuyas amplitudes se han normalizado. En este punto, se procedió a diseñar el circuito de convolución cuántica y a realizar la transformación.

En la Figura 27, se puede observar el segmento de código en el que se definió la estructura del circuito cuántico de convolución, utilizando elementos de la librería PennyLane. Este circuito consta de cinco cables, lo cual corresponde a cinco qubits. En una etapa inicial, la información clásica se transmite mediante compuertas de rotación Rx, cuyos parámetros de rotación corresponden a la información clásica de la señal electrocardiográfica 'phi'. Posteriormente, la información de todos los qubits se transmite a sus vecinos mediante compuertas controladas CRz y CRx con parámetros de rotación fijos. Este proceso se repite hasta que la información de los cinco qubits se haya transmitido a uno solo de ellos, lo cual corresponde al proceso de convolución en el dominio cuántico.

En este punto, se introduce la Figura 28, la cual representa gráficamente la estructura descrita anteriormente. En la imagen, se puede observar claramente cómo el comportamiento de la convolución clásica es imitado por el circuito cuántico. Por ejemplo, el número de qubits

```python
def CONVCircuit(phi, wires, i=0):
    """
    quantum convolution Node
    """
    # parameter
    theta = np.pi / 2

    qml.RX(phi[0] * np.pi, wires=0)
    qml.RX(phi[1] * np.pi, wires=1)
    qml.RX(phi[2] * np.pi, wires=2)
    qml.RX(phi[3] * np.pi, wires=3)
    qml.RX(phi[4] * np.pi, wires=4)

    qml.CRZ(theta, wires=[1, 0])
    qml.CRZ(theta, wires=[4, 3])
    qml.CRX(theta, wires=[1, 0])
    qml.CRX(theta, wires=[4, 3])

    qml.CRZ(theta, wires=[0, 2])
    qml.CRX(theta, wires=[0, 2])
    qml.CRZ(theta, wires=[3, 2])
    qml.CRX(theta, wires=[3, 2])

    # Expectation value
    measurement = qml.expval(qml.PauliZ(wires=2))

    return measurement
```

Figura 27

El fragmento de código donde se define el circuito de convolucion cuantica.

en el circuito se considera equivalente al tamaño del filtro o kernel en la convolución clásica. Además, en este enfoque, los parámetros del filtro (denominados 'theta') se mantienen fijos, lo que significa que no son entrenados. No obstante, se contempla la posibilidad de integrar el circuito de convolución en el dominio cuántico en el proceso de entrenamiento del modelo híbrido, con el objetivo de optimizar los parámetros 'theta' para mejorar la tarea de clasificación de los complejos QRS.

En la Figura 29, se presentan algunas muestras de los datos después de haber sido sometidos a la convolución en el dominio cuántico, junto con sus respectivas etiquetas. En estas muestras transformadas, el eje vertical refleja los valores obtenidos de la convolución en el dominio cuántico del segmento de la señal electrocardiográfica normalizada, mientras que el eje horizontal corresponde a las muestras en las que se realizó la convolución en el dominio cuántico.

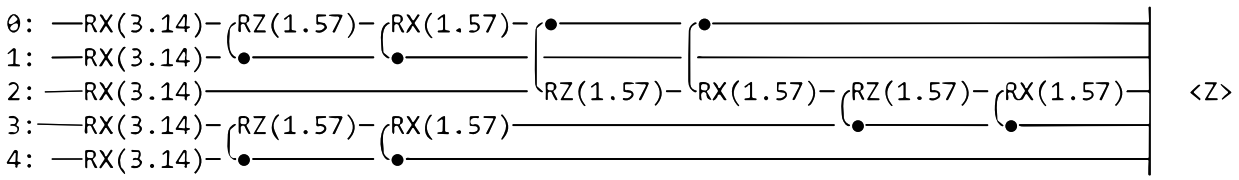

Figura 28

Representacion grafica del circuito convolucional cuantico.

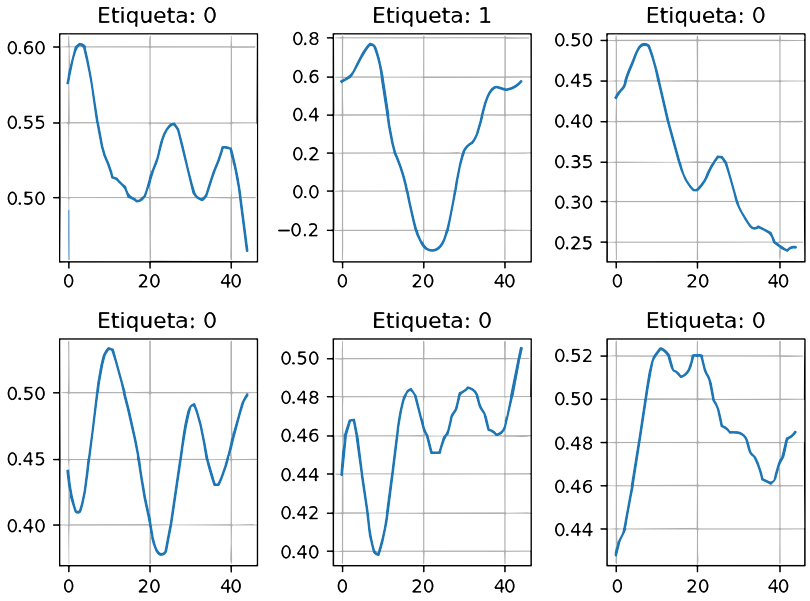

Figura 29

Muestra de los datos de entrada después de haber realizado la convolución en el dominio cuántico.

Para el diseño del modelo, se mantuvo la estructura de las capas lineales del modelo clásico y se siguió un proceso de entrenamiento y prueba similar al de las dos etapas anteriores, manteniendo consistencia en todo el proceso. A pesar de que la transformación de los segmentos podría sugerir que ya no son representativos de los complejos QRS, el destacado rendimiento del modelo demuestra lo contrario. Esto indica que la convolución en el dominio cuántico es una herramienta efectiva y valiosa para la clasificación precisa de los complejos QRS en señales electrocardiográficas.

En la Figura 30 se presenta una representación visual de la arquitectura de la red híbrida

con convolución en el dominio cuántico. Un aspecto destacado en esta figura es el proceso de convolución en el dominio cuántico, que se lleva a cabo antes de que los datos ingresen al modelo. Esta etapa de convolución cuántica transforma los datos, preparándolos para la posterior fase de clasificación.

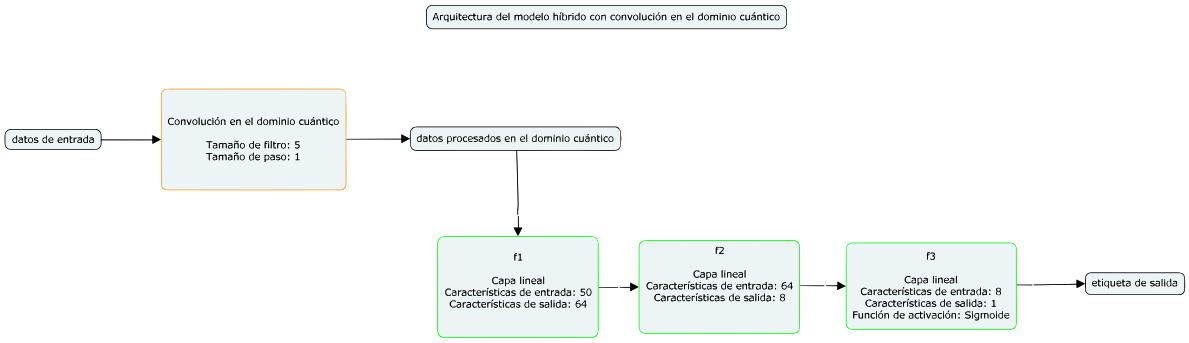

Figura 30

Representación gráfica de la arquitectura del modelo híbrido con convolución en el dominio cuántico.

6. Resultados y Análisis

A continuación, se presentan los resultados obtenidos de los tres modelos implementados en el presente trabajo de grado, así como el análisis respectivo a estos resultados. Estos resultados fueron evaluados principalmente en términos de su exactitud mediante el análisis de las matrices de confusión correspondientes. Además, se proporcionó una estimación de la complejidad temporal asociada a cada uno de los modelos implementados.

Posteriormente, se llevó a cabo un análisis comparativo de los resultados obtenidos en cada uno de los modelos implementados. Este análisis permitió identificar las fortalezas y debilidades de cada enfoque, contribuyendo así a una comprensión más profunda de su rendimiento en la tarea de clasificación de los complejos QRS en las señales electrocardiográficas.

6.1. Resultados

En esta sección, se presentan los resultados obtenidos de los tres modelos implementados en el presente trabajo. Estos resultados se basan en dos métricas fundamentales: la exactitud de la clasificación y la complejidad temporal estimada de cada modelo.

La exactitud de los modelos se evalúa a través del análisis de las matrices de confusión correspondientes. Estas matrices permiten determinar la capacidad de los modelos para proporcionar una clasificación confiable de los complejos QRS en las señales electrocardiográficas. Un alto nivel de exactitud es crucial para garantizar la precisión en aplicaciones médicas y de diagnóstico.

Además de la exactitud, se analiza la complejidad temporal estimada de cada modelo. La complejidad temporal es un indicador importante en la selección de un modelo para una aplicación específica, ya que el tiempo requerido para realizar la clasificación puede ser crítico en ciertos contextos. Comparar la complejidad temporal de los modelos proporciona información valiosa para la toma de decisiones en la implementación práctica.

6.1.1. Resultados del Modelo Clásico

Exactitud: En la Figura 31 se presentan los resultados de la evaluación del modelo clásico mediante una matriz de confusión. En esta representación gráfica, el eje vertical corresponde a las etiquetas reales, las cuales son Complejo QRS o No Complejo QRS, mientras que el eje horizontal corresponde a las etiquetas predichas por el modelo. Cada valor en la matriz indica el número de veces que se produjo la intersección de ambas etiquetas. Por ejemplo, el número en la intersección entre la etiqueta real Complejo QRS y la etiqueta predicha Complejo QRS representa la cantidad de veces que el modelo acertó al clasificar un segmento como Complejo QRS cuando realmente lo era.

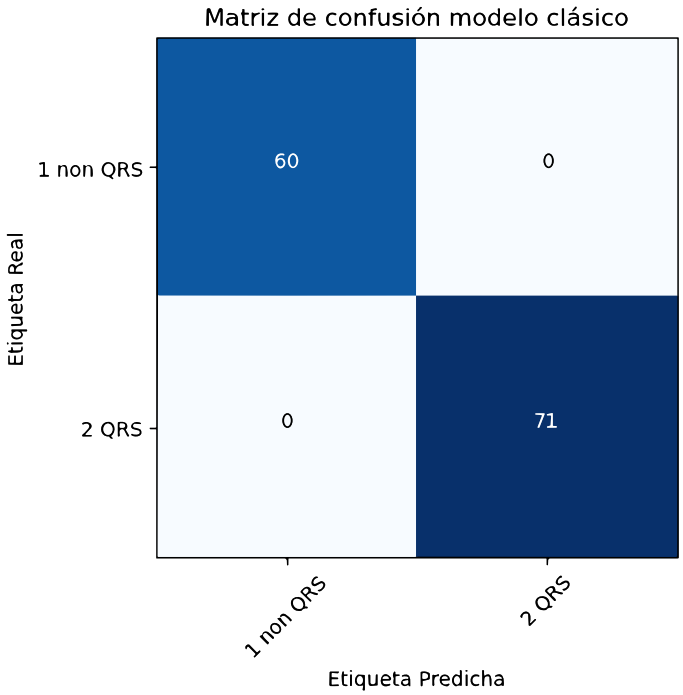

Figura 31

Matriz de confusión correspondiente a la evaluacion del modelo clásico.

La exactitud del modelo clásico se calcula como el cociente entre el número de predicciones correctas y el número total de predicciones realizadas por el modelo. En este contexto, una predicción se considera correcta cuando el modelo clasifica un segmento correctamente como Complejo QRS o como No Complejo QRS. Esto significa que se suman dos componentes:

el número de veces que el modelo predijo correctamente Complejo QRS cuando realmente lo era y el número de veces que predijo correctamente No Complejo QRS cuando realmente no lo era. El resultado se divide entre el número total de predicciones realizadas, lo que proporciona una medida de la proporción de clasificaciones correctas en relación con todas las clasificaciones realizadas por el modelo.

$$\text{Exactitud} = \frac{\text{Predicciones Correctas}}{\text{Total de Predicciones}}$$

$$\text{Exactitud del modelo clásico} = \frac{60 + 71}{131} = 1{,}00$$

Complejidad Temporal: La complejidad de tiempo total del modelo clásico de red neuronal convolucional 1D, en función del tamaño de entrada n, se puede estimar en base al número de operaciones elementales [33] realizadas en las diferentes capas del modelo. La arquitectura del modelo clásico se presenta en la Figura 15, y su complejidad temporal se calcula sumando las operaciones realizadas en las capas respectivas."

1. **Primera capa convolucional 1D:** Esta capa realiza operaciones de convolución en la señal de entrada. Con 1 canal de entrada, 16 canales de salida, un tamaño de kernel de 5 y un tamaño de paso de 1, el número total de operaciones de convolución es aproximadamente $16 \times 5 \times n$.

2. **Segunda capa convolucional 1D:** En esta capa, se realiza otra operación de convolución. Con 16 canales de entrada, 1 canal de salida, un tamaño de kernel de 5 y un tamaño de paso de 1, el número total de operaciones de convolución es aproximadamente $16 \times 5 \times n$.

3. **Primera capa lineal:** Esta capa realiza operaciones de multiplicación de matriz-vector. Con 64 neuronas de salida y n entradas, el número total de operaciones es $64 \times n$.

4. **Segunda capa lineal:** Similar a la capa anterior, esta capa también realiza operaciones de multiplicación de matriz-vector. Con 8 neuronas de salida y 64 entradas, el número total de operaciones es 8×64.

5. **Tercera capa lineal:** La última capa realiza operaciones de multiplicación de matriz-vector. Con 1 neurona de salida y 8 entradas, el número total de operaciones es 1×8.

Para obtener la complejidad temporal estimada en función de n, se calcula el número de operaciones elementales realizadas en todas las capas [28]:

$$\text{Complejidad Temporal del modelo clásico} = 16 \times 5 \times n + 16 \times 5 \times n + 64 \times n + 8 \times 64 + 1 \times 8$$

$$\text{Complejidad Temporal del modelo clásico} = 224 \times n + 520$$

Esto representa una estimación de la complejidad de tiempo del modelo clásico en función del tamaño de entrada n, donde en este caso $n = 50$ como se vio en la Sección 5.1.3.

6.1.2. Resultados del Modelo Híbrido con Capa Cuántica a la Salida

Exactitud: Los resultados de la evaluación del modelo híbrido con capa cuántica a la salida se presentan en la Figura 32. Esta matriz de confusión representa las etiquetas reales (Complejo QRS o No Complejo QRS) en el eje vertical y las etiquetas predichas por el modelo en el eje horizontal. Los valores en la matriz indican las veces que el modelo acertó al clasificar segmentos correctamente. La exactitud se calcula de manera similar al modelo clásico, sumando las predicciones correctas y dividiendo por el total de predicciones.

$$\text{Exactitud del modelo híbrido con capa cuántica a la salida} = \frac{\text{Predicciones Correctas}}{\text{Total de Predicciones}}$$

$$\text{Exactitud del modelo híbrido con capa cuántica a la salida} = \frac{124}{131} = 0{,}94$$

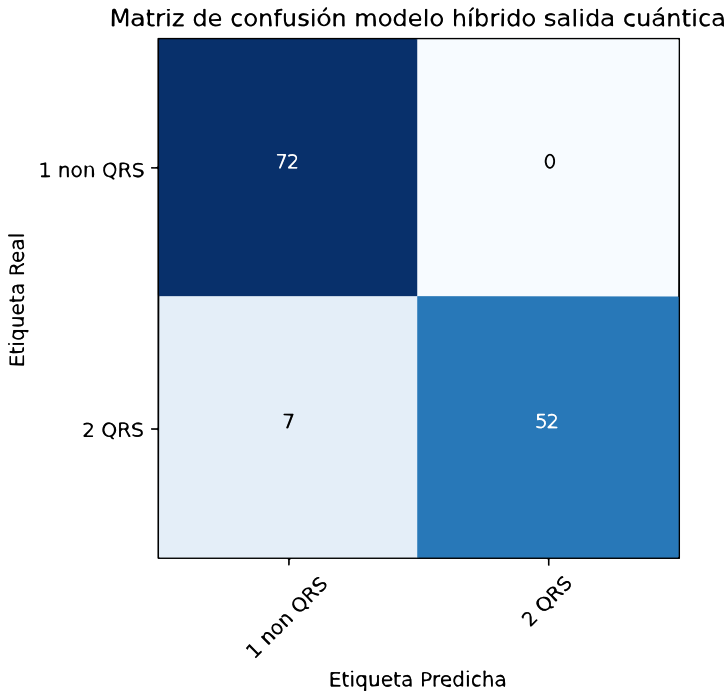

Figura 32

Matriz de confusión correspondiente a la evaluacion del modelo híbrido con capa cuántica a la salida.

Complejidad Temporal: La complejidad de tiempo estimada del modelo híbrido con capa cuántica a la salida, en función del tamaño de entrada n, se puede calcular considerando el número de operaciones fundamentales y el número de compuertas cuánticas unitarias secuenciales [25]. La arquitectura del modelo híbrido con capa cuántica se muestra en la Figura 25, y su complejidad temporal se estima teniendo en cuenta las operaciones realizadas en la red neuronal clásica y el número de compuertas cuánticas secuenciales en el circuito cuántico:

1. **Primera capa convolucional 1D:** Esta capa realiza operaciones de convolución en la señal de entrada. Con 1 canal de entrada, 16 canales de salida, un tamaño de kernel de 5 y un tamaño de paso de 1, el número total de operaciones de convolución es aproximadamente $16 \times 5 \times n$.

2. **Segunda capa convolucional 1D:** En esta capa, se realiza otra operación de convo-

lución. Con 16 canales de entrada, 1 canal de salida, un tamaño de kernel de 5 y un tamaño de paso de 1, el número total de operaciones de convolución es aproximadamente $16 \times 5 \times n$.

3. **Primera capa lineal:** Esta capa realiza operaciones de multiplicación de matriz-vector. Con 64 neuronas de salida y n entradas, el número total de operaciones es $64 \times n$.

4. **Segunda capa lineal:** Similar a la capa anterior, esta capa también realiza operaciones de multiplicación de matriz-vector. Con 8 neuronas de salida y 64 entradas, el número total de operaciones es 8×64.

5. **Tercera capa lineal:** La última capa realiza operaciones de multiplicación de matriz-vector. Con 2 neuronas de salida y 8 entradas, el número total de operaciones es 2×8.

6. **Circuito Cuántico, Mapa de Características:** Como se describe en la Sección 5.3.3, este circuito cuántico consta de 2 entradas y 2 qubits, junto con 5 compuertas cuánticas secuenciales, compuestas por 2 de rotación unitaria y 3 de rotación controlada. En total, esta capa contribuye con $(3 + 2)$ operaciones.

7. **Circuito Cuántico, Ansatz:** De manera similar, como se detalla en la Sección 5.3.4, este circuito cuántico se compone de 2 entradas y 2 qubits, y contiene 3 compuertas cuánticas secuenciales, que incluyen 2 de rotación unitaria y 1 de rotación controlada. En conjunto, esta capa suma $(2 + 1)$ operaciones.

La estimación de la complejidad temporal total en función de n se obtiene sumando el número de operaciones elementales de las capas:

$$\text{Complejidad Temporal del modelo híbrido con capa cuántica a la salida} = 224n + 536$$

Esto representa una estimación de la complejidad de tiempo del modelo híbrido con capa cuántica a la salida, en función del tamaño de entrada n, donde, al igual que en el modelo clásico, $n = 50$.

6.1.3. Resultados del Modelo Híbrido con Convolución en el Dominio Cuántico

Exactitud: La Figura 33 muestra los resultados de la evaluación del modelo híbrido con convolución en el dominio cuántico mediante una matriz de confusión. Esta representación gráfica ilustra cómo las etiquetas reales (Complejo QRS o No Complejo QRS) se comparan con las etiquetas predichas por el modelo. Cada valor en la matriz refleja cuántas veces el modelo acertó al clasificar un segmento correctamente, ya sea como Complejo QRS o No Complejo QRS.

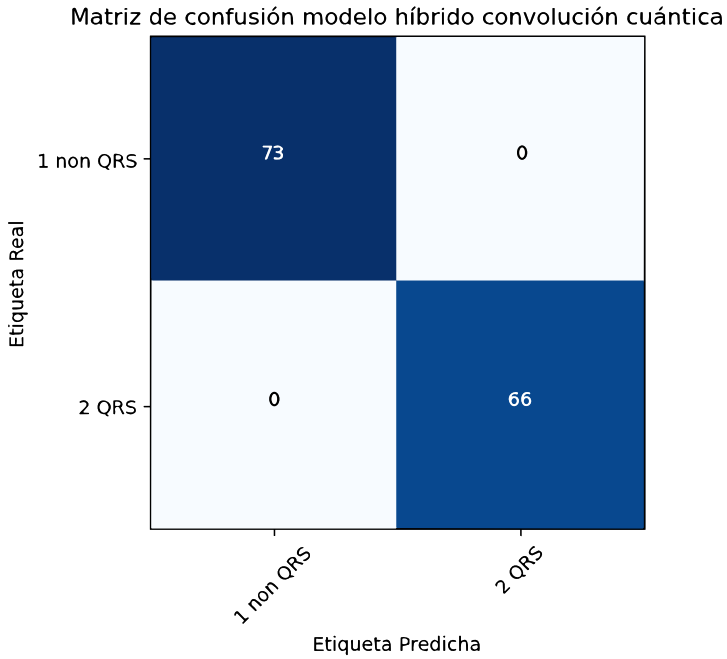

Figura 33

Matriz de confusión correspondiente a la evaluación del modelo híbrido con convolución en el dominio cuántico.

La exactitud del modelo híbrido con convolución en el dominio cuántico se calcula como el co-

ciente entre el número de predicciones correctas y el número total de predicciones realizadas por el modelo. Una predicción se considera correcta cuando el modelo clasifica adecuadamente un segmento como Complejo QRS o No Complejo QRS.

$$\text{Exactitud} = \frac{\text{Predicciones Correctas}}{\text{Total de Predicciones}}$$

$$\text{Exactitud del Modelo Híbrido con Convolución en el Dominio Cuántico} = \frac{73 + 66}{139} = 1{,}00$$

Complejidad Temporal: La complejidad de tiempo estimada del modelo híbrido con convolución en el dominio cuántico, en función del tamaño de entrada n, se calcula considerando el número de operaciones fundamentales y el número de compuertas cuánticas unitarias secuenciales, teniendo en cuenta el número de veces que se repite la ejecución del circuito. La arquitectura del modelo híbrido con convolución en el dominio cuántico se presenta en la Figura 30, y su complejidad temporal se estima teniendo en cuenta las operaciones realizadas tanto en la red neuronal clásica como en el circuito cuántico y las veces que este se repite:

1. **Circuito Cuántico de Convolución:** El circuito cuántico de convolución, descrito en la Sección 5.4.1, consta de 5 qubits y se ejecuta n veces. Cada ejecución del circuito incluye 7 compuertas cuánticas secuenciales, una de rotación unitaria y 6 de rotación controlada. Por lo tanto, el número total de operaciones cuánticas para el circuito de convolución es aproximadamente $7n$ operaciones.

2. **Primera capa lineal:** Esta capa realiza operaciones de multiplicación de matriz-vector. Con n neuronas de entrada y 32 neuronas de salida, el número total de operaciones es $32 \times n$ multiplicaciones.

3. **Segunda capa lineal:** Similar a la capa anterior, esta capa también realiza operaciones de multiplicación de matriz-vector. Con 32 neuronas de entrada y 8 neuronas de salida, el número total de operaciones es 32×8 multiplicaciones.

4. **Tercera capa lineal:** La última capa realiza operaciones de multiplicación de matriz-vector. Con 8 neuronas de entrada y 1 neurona de salida, el número total de operaciones es 8×1 multiplicaciones.

Para obtener la complejidad temporal estimada en función de n [28], se suman todas las operaciones de las capas:

Complejidad Temporal del modelo híbrido con convolución en el dominio cuántico = $39n$+264

Esto representa una estimación de la complejidad de tiempo del modelo híbrido con convolución en el dominio cuántico en función del tamaño de entrada n, donde en este caso $n = 50$.

6.2. Análisis de Resultados

A continuación, se realiza un análisis de los resultados obtenidos por los modelos implementados. El Cuadro 1 condensa los principales resultados de los modelos, incluyendo su exactitud y complejidad temporal, en función del tamaño de los datos de entrada n.

Cuadro 1

Resultados de los Modelos

Modelos	**Exactitud**	**Complejidad Temporal en función de n**
Modelo Clásico	1.00	$224 \times n + 520$
Modelo Híbrido con Salida Cuántica	0.94	$224 \times n + 536$
Modelo Híbrido con Convolución Cuántica	1.00	$39 \times n + 264$

Partiendo de la exactitud, se puede evidenciar que los tres modelos presentan una alta precisión, de lo cual se pueden inferir dos observaciones. En primer lugar, los tres modelos tienen un desempeño destacado en la tarea de clasificación del complejo QRS. En segundo lugar, la tarea de distinguir el modelo QRS parece ser relativamente sencilla, ya que tanto el modelo clásico como el modelo híbrido con convolución en el dominio cuántico muestran una exactitud del 100 por ciento, lo cual es atípico en tareas de aprendizaje automático. No

obstante, es importante destacar que el modelo híbrido con capa de salida cuántica presenta una ligera disminución en su exactitud en comparación con los otros modelos. Esto puede ser un factor clave al seleccionar un modelo, especialmente en tareas donde la clasificación sea crítica, y se busque la mayor exactitud posible.

Es relevante agregar que el modelo híbrido con capa de salida cuántica nunca detectó de forma incorrecta un complejo QRS. Siempre que predijo que un segmento era QRS, en efecto, lo era. Sin embargo, presentó errores al detectar segmentos que no eran QRS. En algunas ocasiones, predijo que el segmento no lo era, y en realidad, sí lo era. Este tipo de errores puede ser admisible en determinadas aplicaciones, dependiendo de la tolerancia al error y las necesidades específicas del contexto.

Por otro lado, al comparar la complejidad temporal, se evidencia que el modelo clásico y el modelo híbrido con capa cuántica de salida tienen un rendimiento bastante similar, con una leve ventaja para el modelo clásico. Esta ventaja puede resultar despreciable en señales de entrada de gran longitud, donde la diferencia en el tiempo de procesamiento entre los dos modelos es mínima.

En contraste, el modelo híbrido con convolución en el dominio cuántico presenta una mejora significativa en términos de complejidad temporal con respecto a los otros dos modelos. Esta mejora puede ser determinante a la hora de seleccionar uno de los tres modelos para una tarea donde la velocidad de procesamiento sea una consideración importante. En aplicaciones en tiempo real o en situaciones donde se requiere un procesamiento rápido de las señales, el modelo híbrido con convolución cuántica destaca como una opción favorable.

7. Conclusiones

En este estudio, se evaluó el desempeño de tres enfoques de modelado para la clasificación binaria de complejos QRS en señales de electrocardiograma. El presente análisis se centró en la exactitud de los modelos y su complejidad temporal. Además, se discuten algunas herramientas utilizadas en el trabajo, las posibles aplicaciones y consideraciones para futuras investigaciones.

En cuanto al desempeño en términos de exactitud, se observó un rendimiento sobresaliente en los tres modelos. Tanto el modelo clásico como la simulación del modelo híbrido con convolución en el dominio cuántico demostraron ser altamente precisos. Esta alta exactitud los convierte en opciones viables para tareas de clasificación binaria de complejos QRS. La elección entre estos modelos dependerá de la criticidad de la aplicación y de las necesidades específicas del contexto.

En lo que respecta al desempeño en términos de tiempo, el modelo híbrido con convolución en el dominio cuántico se destacó por su complejidad temporal estimada significativamente mejorada en comparación con los otros dos modelos. Esta ventaja en términos de eficiencia temporal es especialmente relevante en aplicaciones que requieren un procesamiento rápido de las señales. El modelo clásico y el modelo híbrido con capa de salida cuántica mostraron un rendimiento similar en términos de estimación de la complejidad temporal, siendo el modelo clásico ligeramente más eficiente.

En cuanto a las herramientas, en este trabajo de grado, destacamos el uso de Qiskit y PennyLane, dos librerías fundamentales para su desarrollo. Sin embargo, es importante mencionar que, debido al estado actual de la tecnología, la implementación de la computación cuántica en la clasificación de complejos QRS puede enfrentar desafíos adicionales inherentes a las tecnologías emergentes. Estos desafíos pueden incluir la decoherencia de los estados cuánticos de los qubits y la limitación de recursos computacionales cuánticos, lo que dificulta la ejecución de circuitos cuánticos en múltiples iteraciones. Por lo tanto, es necesario abordar los desafíos actuales en el estado del arte antes de considerar su implementación en aplica-

ciones del mundo real.

Además, cabe destacar la exigencia de procesamiento requerida para realizar dichas simulaciones, dado que su complejidad implica extensos tiempos de simulación. Esto hace inviable la utilización de simulaciones en aplicaciones prácticas en tiempo real. Dicho esto, en la actualidad, el modelo a elegir debería ser el clásico, debido a su alto desempeño en exactitud y una complejidad temporal aceptable.

Finalmente, a partir del presente trabajo, se proponen diferentes alternativas de estudio. La primera es la aplicación en tiempo real de modelos clásicos para la detección de patrones más complejos. Esto implicaría utilizar estos modelos en situaciones de monitoreo continuo, donde la detección precisa de patrones complejos en señales de electrocardiograma es fundamental. Por otro lado, aunque el modelo híbrido con convolución en el dominio cuántico mostró un buen desempeño, es importante destacar que los valores del filtro no fueron entrenados, sino que se mantuvieron estáticos. Es improbable que estos valores sean los más adecuados para la clasificación. Por lo tanto, se justifica extender el estudio investigando la forma de realizar un entrenamiento adecuado para estos parámetros y explorar su impacto en la precisión de la clasificación.

Referencias

[1] C. Thomas. "Electrocardiografía." (2021), dirección: https://www.msdmanuals.com/es-co/hogar/trastornos-del-coraz%C3%B3n-y-los-vasos-sangu%C3%ADneos/diagn%C3%B3stico-de-las-enfermedades-cardiovasculares/electrocardiograf%C3%ADa (visitado 16-09-2023).

[2] A. Mincholé, J. Camps, A. Lyon y B. Rodríguez, "Machine learning in the electrocardiogram," *Journal of Electrocardiology*, vol. 57, S61-S64, 2019, ISSN: 0022-0736. DOI: https://doi.org/10.1016/j.jelectrocard.2019.08.008. dirección: https://www.sciencedirect.com/science/article/pii/S0022073619304571.

[3] D. e. a. R., *Electrocardiografía*. Bogotá (Colombia): Sociedad Colombiana de Cardiología y Cirugía Cardiovascular, 2008, págs. 1-8.

[4] A. Mincholé, J. Camps, A. Lyon y B. Rodríguez, "Machine learning in the electrocardiogram," *Journal of Electrocardiology*, vol. 57, S61-S64, 2019, ISSN: 0022-0736. DOI: https://doi.org/10.1016/j.jelectrocard.2019.08.008. dirección: https://www.sciencedirect.com/science/article/pii/S0022073619304571.

[5] C. Castellano, M. P. De Juan y F. Attie, *Electrocardiografía*. Elsevier España, SA, 2007, págs. 9-18.

[6] M. Simjanoska, M. Gjoreski, M. Gams y A. Madevska Bogdanova, "Non-Invasive Blood Pressure Estimation from ECG Using Machine Learning Techniques," *Sensors*, vol. 18, n.º 4, 2018, ISSN: 1424-8220. DOI: 10.3390/s18041160. dirección: https://www.mdpi.com/1424-8220/18/4/1160.

[7] o. d. A. File:SinusRhythmLabels.svg: Created by Agateller (Anthony Atkielski) converted to svg by atom. "Representación esquemática de un electrocardiograma normal." (2017-05-29), dirección: https://es.wikipedia.org/wiki/Archivo:SinusRhythmLabels-es.svg.

[8] A. Goldberger, L. Amaral, L. Glass et al. “PhysioBank, PhysioToolkit, and PhysioNet: Components of a new research resource for complex physiologic signals.” (2000), dirección: https://www.physionet.org/static/pubs/circulation.pdf.

[9] T. Lugovaya, “Biometric Human Identification Based on Electrocardiogram,” Tesis de mtría., Faculty of Computing Technologies e Informatics, Electrotechnical University "LETI", Saint-Petersburg, Russian Federation, jun. de 2005.

[10] Z.-H. ZHOU, *Machine Learning*. SPRINGER VERLAG, SINGAPOR, 2022, págs. 103-128.

[11] “Neurons in Neural Networks.” (26 de oct. de 2022), dirección: https://www.baeldung.com/cs/neural-networks-neurons (visitado 19-10-2023).

[12] A. Cartas. “Diagrama de un perceptrón con cinco señales de entrada.” (2015-07-12), dirección: https://commons.wikimedia.org/wiki/File:Perceptr%C3%B3n_5_unidades.svg.

[13] J. Brownlee. “A Gentle Introduction to the Rectified Linear Unit (ReLU).” (20 de ago. de 2020), dirección: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/ (visitado 19-10-2023).

[14] “Funciones de activación de una neurona.” (), dirección: https://platzi.com/tutoriales/2155-calculo-data-science/10282-funciones-de-activacion-de-una-neurona/ (visitado 19-10-2023).

[15] J. Brownlee. “A Gentle Introduction To Sigmoid Function.” (17 de ago. de 2021), dirección: https://machinelearningmastery.com/a-gentle-introduction-to-sigmoid-function/ (visitado 19-10-2023).

[16] “Negative Log-Likelihood.” (), dirección: https://notesbylex.com/negative-log-likelihood#:~:text=Negative%20log%2Dlikelihood%20is%20a,all%20items%20in%20the%20batch. (visitado 19-10-2023).

[17] "Optimisation Algorithm - Adaptive Moment Estimation (Adam)." (21 de ene. de 2020), dirección: https://towardsdatascience.com/optimisation-algorithm-adaptive-moment-estimation-adam-92144d75e232 (visitado 19-10-2023).

[18] J. Brownlee. "How Do Convolutional Layers Work in Deep Learning Neural Networks?" (16 de abr. de 2020), dirección: https://machinelearningmastery.com/convolutional-layers-for-deep-learning-neural-networks/ (visitado 19-10-2023).

[19] "The Jupyter Notebook." (), dirección: https://jupyter-notebook.readthedocs.io/en/stable/notebook.html (visitado 19-10-2023).

[20] J. Brownlee. "Your First Machine Learning Project in Python Step-By-Step." (25 de sep. de 2023), dirección: https://machinelearningmastery.com/machine-learning-in-python-step-by-step/ (visitado 19-10-2023).

[21] R. Lee. "A Full Overview of Quantum Computing." (28 de sep. de 2021), dirección: https://medium.com/visionary-hub/a-full-overview-of-quantum-computing-6897ecb22004 (visitado 19-10-2023).

[22] *Elements of Quantum Computing: History, Theories and Engineering Applications.* Springer, 2015, págs. 68-82.

[23] H. Jaffali. "Quantum Machine Learning: a quick overview." (5 de oct. de 2021), dirección: https://medium.com/colibritd-quantum/quantum-machine-learning-a-quick-overview-f7437b368d43 (visitado 19-10-2023).

[24] I. Cong, S. Choi y M. D. Lukin, "Quantum convolutional neural networks," *Nature Physics*, vol. 15, n.º 12, págs. 1273-1278, 2019.

[25] P. Kim, D. Han y K. C. Jeong, "Time–space complexity of quantum search algorithms in symmetric cryptanalysis: applying to AES and SHA-2," *Quantum Information Processing*, vol. 17, n.º 12, pág. 339, oct. de 2018. DOI: 10.1007/s11128-018-2107-3.

[26] "Qiskit 0.44 documentation." (19 de oct. de 2023), dirección: https://qiskit.org/documentation/ (visitado 19-10-2023).

[27] "PennyLane Documentation." (), dirección: https://docs.pennylane.ai/en/stable/ (visitado 19-10-2023).

[28] G. L. Team. "What is Time Complexity And Why Is It Essential?" (24 de ago. de 2023), dirección: https://www.mygreatlearning.com/blog/why-is-time-complexity-essential/#how-to-calculate-time-complexity (visitado 19-10-2023).

[29] R. E. Korf, M. Reid y S. Edelkamp, "Time complexity of iterative-deepening-A," *Artificial Intelligence*, vol. 129, n.º 1, págs. 199-218, 2001, ISSN: 0004-3702. DOI: https://doi.org/10.1016/S0004-3702(01)00094-7. dirección: https://www.sciencedirect.com/science/article/pii/S0004370201000947.

[30] A. F. Gad. "Accuracy, Precision, and Recall in Deep Learning." (9 de abr. de 2021), dirección: https://blog.paperspace.com/deep-learning-metrics-precision-recall-accuracy/ (visitado 20-10-2023).

[31] "Evaluación de modelos de clasificación." (5 de oct. de 2017), dirección: https://rpubs.com/chzelada/275494 (visitado 19-10-2023).

[32] "Torch Connector and Hybrid QNNs." (), dirección: https://qiskit.org/ecosystem/machine-learning/tutorials/05_torch_connector.html (visitado 19-10-2023).

[33] B. Shah y H. Bhavsar, "Time Complexity in Deep Learning Models," *Procedia Computer Science*, vol. 215, págs. 202-210, 2022, 4th International Conference on Innovative Data Communication Technology and Application, ISSN: 1877-0509. DOI: https://doi.org/10.1016/j.procs.2022.12.023. dirección: https://www.sciencedirect.com/science/article/pii/S1877050922020944.