



UNIVERSIDAD AUTÓNOMA DEL ESTADO
DE HIDALGO

INSTITUTO DE CIENCIAS BÁSICAS E
INGENIERÍA

DOCTORADO EN CIENCIAS EN INGENIERÍA,
CON ÉNFASIS EN ANÁLISIS Y MODELACIÓN
DE SISTEMAS

T E S I S

Optimización de Hiperparámetros de una Red
Neuronal Convolutiva a través de un
Algoritmo Metaheurístico para mejorar la
Clasificación de Arritmias Cardíacas.

PARA OBTENER EL TÍTULO DE:

Doctor en Ciencias en Ingeniería, con Énfasis
en Análisis y Modelación de Sistemas

P R E S E N T A :

M. en C. Fredy Santander Baños

DIRECTOR DE TESIS:

Dr. Norberto Hernández Romero

CODIRECTOR DE TESIS:

Dr. Gustavo Erick Anaya Fuentes

COMITÉ TUTORIAL:

Dr. Irving Barragán Vite

Dr. Juan Carlos Seck Tuoh Mora

Pachuca de Soto, Hgo., México., noviembre de 2023



UNIVERSIDAD AUTÓNOMA DEL ESTADO DE HIDALGO

Instituto de Ciencias Básicas e Ingeniería

School of Engineering and Basic Sciences

Área Académica de Ingeniería y Arquitectura

Department of Engineering and Architecture

Número de control: ICBI-AAIyA/3183/2023

Asunto: Autorización de impresión

Mtra. Ojuky del Rocío Islas Maldonado
Directora de Administración Escolar de la UAEH
Presente.

El Comité Tutorial de Tesis del programa educativo de posgrado titulado “**Optimización de Hiperparámetros de una Red Neuronal Convolutiva a través de un Algoritmo Metaheurístico para mejorar la Clasificación de Arritmias Cardíacas**”, realizado por la sustentante **Fredy Santander Baños** con número de cuenta **429458** perteneciente al programa de **Doctorado en Ciencias en Ingeniería, con Énfasis en Análisis y Modelación de Sistemas**; una vez que ha revisado, analizado y evaluado el documento recepcional de acuerdo a lo estipulado en el Artículo 110 del Reglamento de Estudios de Posgrado, tiene a bien extender la presente:

AUTORIZACIÓN DE IMPRESIÓN

Por lo que la sustentante deberá cumplir los requisitos del Reglamento de Estudios de Posgrado y con lo establecido en el proceso de grado vigente.

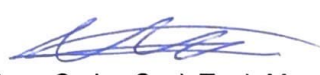
Atentamente
“Amor, Orden y Progreso”
Pachuca, Hidalgo a 17 de noviembre de 2023

El Comité Tutorial


Dr. Norberto Hernández Romero
Director


Dr. Gustavo Erick Anaya Fuentes
Codirector


Dr. Irving Barragán Vite
Miembro del comité


Dr. Juan Carlos Seck Tuoh Mora
Miembro del comité



JCSTM/MABC

Ciudad del Conocimiento
Carretera Pachuca-Tulancingo km 4.5 Colonia
Carboneras, Mineral de la Reforma, Hidalgo,
México. C.P. 42184
Teléfono: +52 (771) 71 720 00 ext. 4000, 4001
Fax 2109
aai_icbi@uaeh.edu.mx



www.uaeh.edu.mx

Agradecimientos

Expreso mi primer agradecimiento a la persona que a lo largo de mi trayectoria de formación como investigador, ha estado apoyándome, motivándome para no abandonar y retirarme de este camino; madre, gracias por ser un ejemplo de fortaleza y perseverancia, por tus palabras de aliento y por estar ahí en los momentos más estresantes.

Mi segundo agradecimiento es para mi novia quién sin duda ha sido un pilar y un motivante para no desistir pese a las adversidades, este trabajo se logró, gracias infinitas por tu paciencia y perseverancia.

A mi núcleo familiar que siempre ha creído en mí y me motivan a culminar cada reto nuevo que me propongo. Principalmente a mi abue y tíos, quiénes jamás dejaron de apoyarme y alentarme a continuar con este gran reto.

Dr. Norberto Hernández Romero, mil gracias por su invaluable apoyo y constante seguimiento que me brindó durante la realización del presente trabajo, pero sobre todo agradezco la comprensión, paciencia y su forma de motivarme a no declinar pese a la situación adversa y complicada que se suscitó al final del trabajo de investigación.

Agradezco a mi codirector de tesis y comité tutorial, por sus valiosas aportaciones y críticas que permitieron enriquecer y mejorar este trabajo de tesis. A los Doctores que me impartieron clase durante mi formación como doctorante.

Y, al Consejo Nacional de Humanidades, Ciencias y Tecnologías (CONAHCyT), por haberme otorgado la beca como estudiante de posgrado, lo que propició que este trabajo de investigación fuese posible y se vea reflejado en la presente tesis.

Resumen

Aunque el rendimiento por si solo de las redes neuronales artificiales (RNAs) suele ser satisfactorio, diversos trabajos de investigación han demostrado que la propuesta de hibridaciones entre RNAs y algoritmos metaheurísticos (AMH), pueden incrementar dicho rendimiento en sus tareas de estimación o clasificación; según corresponda el contexto del problema y la naturaleza de la red neuronal. El presente trabajo de tesis, aborda la clasificación de arritmias cardíacas (ACs) a través del uso de algoritmos de cómputo inteligente; el primero de ellos corresponde a la red neuronal convolucional (CNN, por sus siglas en inglés) y el segundo al de optimización por enjambre de partículas (PSO, por sus siglas en inglés). La metodología implementada para el desarrollo de este trabajo consta de tres procedimientos esenciales. El primero consiste en la preparación de la fuente de información que corresponde a las ACs, de tal forma que le permita a la CNN llevar a cabo la tarea de clasificación. En el segundo procedimiento, se hace uso del PSO, para la búsqueda de una solución satisfactoria compuesta por una arquitectura de capas con sus correspondientes hiperparámetros que le permita a la CNN obtener un rendimiento satisfactorio; la optimización se realiza mediante la minimización del error de entropía cruzada categórica (CE), dicho error es tomado por el PSO como una función objetivo. El último procedimiento consiste en tomar los datos de la mejor partícula y evaluar el rendimiento de la CNN, lo que permite describir los resultados experimentales obtenidos. Los resultados del trabajo se traducen como el diseño de un modelo computacional con un desempeño satisfactorio en la clasificación de ACs.

Abstract

Although the performance of artificial neural networks (ANNs) alone is usually satisfactory, several research works have shown that the proposal of hybridizations between ANNs and metaheuristic algorithms (MHA) can increase this performance in their estimation or classification tasks, depending on the context of the problem and the nature of the neural network. The present thesis work addresses the classification of cardiac arrhythmias (CAs) by using intelligent computing algorithms; the first one corresponds to convolutional neural network (CNN) and the second one to particle swarm optimization (PSO). The methodology implemented for the development of this work consists of three essential procedures. The first one consists of the preparation of the information source corresponding to the CAs, in such a way that it allows the CNN to perform the classification task. In the second procedure, the PSO is used to search for a satisfactory solution composed of a layered architecture with its corresponding hyperparameters that allows the CNN to obtain a satisfactory performance; the optimization is performed by minimizing the categorical cross-entropy (CE) error, which is taken by the PSO as an objective function. The last procedure consists of taking the data of the best particle and evaluating the performance of the CNN, which allows describing the experimental results obtained. The results of the work are translated as the design of a computational model with a satisfactory performance in the classification of ACs.

Contenido

Introducción	1
Planteamiento del Problema	7
Justificación	7
Propuesta de Solución	9
Hipótesis	10
Objetivos de la Tesis	10
Estado del Arte	12
Aportaciones	17
Organización de la Tesis	18
1 Arritmias Cardíacas	20
1.1 Antecedentes Históricos	22
1.1.1 Antigüedad	22
1.1.2 Edad Media y Renacimiento	23
1.1.3 Siglos XVII al XIX	23
1.1.4 Siglo XX	24
1.1.5 Siglo XXI	25
1.2 Causas Principales	27
1.3 Tipos de Arritmias	28
1.3.1 Taquicardias	28
1.3.2 Bradicardias	29
1.3.3 Otras Arritmias	30
1.4 Síntomas, Diagnóstico y Tratamiento	31
1.4.1 Síntomas	31
1.4.2 Diagnóstico	31
1.4.3 Tratamiento	32
1.5 Morfología Básica del ECG	32

1.6	Diagnóstico de arritmias cardíacas con inteligencia artificial	34
1.6.1	Metodología a través de IA	35
1.6.2	Ventajas y Desafíos	37
1.6.3	Modelos de IA en la Detección de Arritmias	38
1.6.4	Futuro de la detección de arritmias con IA	39
2	Redes Neuronales Convolucionales	40
2.1	Antecedentes Históricos	41
2.1.1	Inspiración Biológica	41
2.1.2	Primeros Modelos	42
2.1.3	Auge de la Retropropagación	42
2.1.4	Avances en la Década de 2000	43
2.1.5	Integración y Adopción	43
2.2	Arquitectura y Funcionamiento	44
2.2.1	Capa Convolutiva	47
2.2.2	Capa Pooling	49
2.2.3	Capa de Aplanado	49
2.2.4	Capa Totalmente Conectada	50
2.3	Ventajas y Desafíos	50
2.3.1	Ventajas	50
2.3.2	Desafíos	53
2.4	Las CNNs en el Diagnóstico de Arritmias	54
2.5	Programación de la CNN en Python	56
3	Algoritmo de Enjambre de Partículas	58
3.1	Antecedentes Históricos del PSO	60
3.1.1	Década de 1950 y 1960	60
3.1.2	1995 - Origen del PSO	60
3.1.3	Finales de la década de 1990	61
3.1.4	Década de 2000	61
3.1.5	Década de 2010 en adelante	61
3.2	Fundamentos del PSO	62
3.2.1	Partículas	62
3.2.2	Movimiento y Actualización	63
3.2.3	Ecuación de Actualización	63
3.2.4	Criterio de Parada	66
3.3	Ventajas y Desafíos del PSO	68

3.3.1	Ventajas	69
3.3.2	Desafíos	70
3.4	El PSO en el Diagnóstico de Arritmias	71
3.4.1	Selección y Extracción de Características	71
3.4.2	Optimización de Parámetros del Modelo	71
3.4.3	Detección de Puntos Específicos en Señales de ECG	72
3.4.4	Diseño y Ajuste de Filtros	72
3.4.5	Detección de Arritmias en Datos Multimodales	72
3.4.6	Optimización de Estrategias de Diagnóstico	72
3.5	Programación del PSO en Python	73
3.5.1	Adquisición de Datos	73
3.5.2	Preprocesamiento	74
3.5.3	Extracción de Características	74
3.5.4	Diseño del modelo	74
3.5.5	Optimización con PSO	74
3.5.6	Evaluación	75
3.5.7	Diagnóstico	75
3.5.8	Consideraciones adicionales	75
4	Algoritmo H-PSO-CNN	77
4.1	Esencia del H-PSO-CNN	79
4.2	Arquitectura del H-PSO-CNN	81
4.2.1	Inicialización de población numérica	83
4.2.2	Transformación de la Población: Arquitectura Numérica a Capas	86
4.2.3	Entrenamiento de la CNN y Estimación del CE	90
4.2.4	Proceso de Actualización	91
5	Pruebas y Resultados	97
5.1	Banco de Datos de ECG	97
5.2	Prueba y Obtención de Métricas de Clasificación	100
5.2.1	Matriz de Confusión	100
5.2.2	Precision	100
5.2.3	Recall	100
5.2.4	F1-Score	101
5.2.5	Accuracy	101
5.3	Resultados de Experimentación y Discusión.	101
5.3.1	Entorno de Desarrollo	101

5.3.2	Parámetros para el H-PSO-CNN	102
5.3.3	Fuente de Datos	102
5.3.4	Métricas	106
6	Conclusiones y Trabajos Futuros	112
6.1	Estudio Comparativo	112
6.2	Conclusiones	114
6.3	Trabajos Futuros	115

Lista de Figuras

1.1	Representación sistemática de la onda PQRST	34
2.1	Arquitectura general de una CNN.	46
3.1	Flujograma General del PSO.	65
4.1	Diagrama de Flujo del modelo propuesto H-PSO-CNN.	78
4.2	Transformación de datos numéricos de las partículas en datos de capas de arquitectura de partículas.	89
4.3	Actualización de L mediante la optimización del PSO y a través del uso de Algorithm 3.	92
5.1	Conjunto de datos Arrithimya del MIT-BIH agrupado en cinco clases según el estándar AAMI EC57.	98
5.2	Balanceo para el MITDB con cinco clases de arritmias cardíacas. . . .	104
5.3	Optimización del CE con H-PSO-CNN.	106
5.4	Precisión para la clasificación de arritmia cardíaca con H-PSO-CNN.	107

5.5	Valor de pérdida a través de la clasificación de arritmia cardíaca con H-PSO-CNN.	108
5.6	Matriz de confusión para la clasificación de arritmias cardíacas con H-PSO-CNN.	110

Lista de Tablas

5.1	Mapeo entre diferentes clases y descripciones y clases AAMI EC57. . .	99
5.2	Datos totales de prueba y entrenamiento de CNN.	103
5.3	Parámetros de configuración de ajustes para CNN y PSO.	105
5.4	La mejor arquitectura de capas para CNN.	109
5.5	Métricas de rendimiento para CNN Train con H-PSO-CNN.	109
5.6	Métricas de rendimiento para la prueba CNN con H-PSO-CNN. . . .	111
6.1	Resumen de las comparaciones entre el modelo propuesto y los métodos existentes.	113

Introducción

Una red neuronal artificial (RNA) es un modelo computacional que pertenece a la familia de los algoritmos de inteligencia artificial (IA), y que tiene como principales tareas: la predicción, la clasificación o el reconocimiento de patrones [1, 2]. Toda RNA tiene como fuente natural de inspiración el funcionamiento y estructura orgánica del cerebro y sus propias neuronas interconectadas a través de nodos, permitiendo así; el aprendizaje, el recuerdo y/o reconocimiento de sucesos o formas [3].

Las RNAs suelen clasificarse en algoritmos de aprendizaje profundo (AAP) o automático (AAA), según corresponda la profundidad en su número de capas [4] y la configuración de sus correspondientes parámetros o hiperparámetros. Redes neuronales con más de dos capas se catalogan como AAP, mientras que las restantes como AAA.

Los AAP son capaces de reconocer patrones complejos en imágenes, textos, sonidos y otros datos; a fin de generar información y predicciones precisas. Es posible utilizar los AAP para automatizar tareas que habitualmente requieren inteligencia humana, como la descripción de imágenes, la transcripción a texto de un archivo de sonido, la realización de resúmenes de documentos o el reconocimiento de rostros; con mayor

precisión [5].

Aun cuando la teoría que fundamenta el funcionamiento de los AAP se originó en la década de los 50, no fue hasta hace aproximadamente 15 años que se pudo comprobar de forma experimental el potencial que tienen estos algoritmos, esto fue posible gracias a la mejora considerable en el hardware de las computadoras. La puesta en marcha de estos modelos ha despertado el interés en diversos investigadores de diferentes campos de la ciencia, quiénes en su área, buscan obtener el mejor rendimiento de estas técnicas y que esto les permita garantizar una mayor precisión en la tarea encomendada.

Aunque por si solos, la mayoría de los AAP obtienen buenos resultados en sus tareas encomendadas, en algunos casos no ha sido suficiente para cumplir con el objetivo deseado. Por ello, algunos trabajos propuestos por investigadores incluyen el uso de otro tipo de algoritmia computacional, estas técnicas están más centradas a la optimización de un determinado resultado, a través de conceptos como lo son la maximización o caso contrario minimización. Este tipo de modelado computacional hace uso de los AMH, y al igual que los AAP o AAA, existe una amplia gama de ellos [6]. Los AMH son métodos aproximados diseñados para resolver problemas de optimización combinatoria, en los que los heurísticos clásicos no son efectivos [7]. Los AMH proporcionan un marco general para crear nuevos algoritmos híbridos, combinando diferentes conceptos derivados de la IA, la evolución biológica y los mecanismos estadísticos [8, 9].

La reciente propuesta de modelos computacionales híbridos que incluyen tanto a los AAP, AAA y AMH, ha demostrado que se pueden alcanzar resultados mayormente

satisfactorios a los que, por si solos, cada uno de los algoritmos en cuestión son capaces de obtener. La gama de aplicaciones de estas nuevas propuestas son muy amplias, por mencionar algunos: ámbito industrial, comercial, aeroespacial, militar, deportivo y/o médico [10, 11, 12, 13, 14, 15]; siendo esta última un área de interés para el presente trabajo de tesis.

La mayoría de los modelos híbridos que se han propuesto hasta ahora en el campo de la medicina, tienen como objetivo en común mejorar la precisión al máximo posible en las tareas de predicción de enfermedades, diagnóstico oportuno de las mismas e incluso mejorar el control y seguimiento de pacientes crónicos a través de dispositivos electrónicos médicos de monitoreo [16]. El producto final resultante y que aprovecha el potencial de estos algoritmos, se resume en una tecnología sanitaria que incluye en su programación lógica; el funcionamiento de estos modelos computacionales. Esta tecnología sanitaria es aprobada, regulada y vigilada por agencias internacionales como la administración de alimentos y medicamentos (FDA, por sus siglas en inglés) [17], agencias que tienen estrictos estándares y protocolos para la aprobación y/o su futura comercialización de este tipo de productos. Por lo anterior, es fundamental que estos modelos híbridos computacionales garanticen una precisión máxima.

No obstante, para que cualquier AAA o AAP funcione de manera correcta, es de suma importancia contar con fuentes de datos (Datasets) que contengan la información mínima necesaria para que estos algoritmos puedan llevar a cabo sus tareas encomendadas. Estos Datasets pueden ser gratuitos o tener un licenciamiento comercial, y su presentación suele representarse como imágenes o vídeos digitales, texto plano, series temporales u ondas sonoras. Es copiosa la cantidad de Datasets que es-

tán disponibles para la comunidad de investigadores o usuarios que así lo decidan, y hagan uso de esta información con el simple objetivo de la puesta en marcha de algoritmos de IA. En consecuencia, estos datos son ampliamente utilizados por diferentes campos de la ciencia; la medicina no es la excepción [18, 19, 20].

Algunos de los Datasets pertenecientes al campo de la salud, permiten al equipo de investigadores interesados en la aplicación de IA en este rubro; proponer modelos computacionales cuyo objetivo en común, independientemente de la metodología o naturaleza de los algoritmos, consiste en mejorar la precisión en el diagnóstico o predicción de enfermedades. Indudablemente al hablar de estos padecimientos, es referirnos a un número amplio y heterogéneo que abarcan distintos sistemas orgánicos, esto ha permitido la amplia propuesta del uso de algoritmos de IA en la medicina, debido a la capacidad de los mismos en trabajar con la naturaleza de estos datos. Por lo cual, el número de trabajos de investigación que se han publicado en al menos los últimos 10 años en este campo científico, es basto, existen trabajos de investigación que abordan el uso de únicamente AAP cómo lo son las RNAs, trabajos que hacen uso sólo de AMH y aquellos que proponen hibridaciones entre la combinación de RNAs o entre AMH con AAA-AAP, algunos de estos trabajos se describen más adelante en el Estado del Arte.

Resulta sumamente extensa la cantidad de enfermedades que existen hoy en día cuya etiología y fisiología suele ser variada. En este trabajo de tesis, nos concentramos en las enfermedades que se encuentran directamente relacionadas con las ACs, y que regularmente son ocasionadas por un mal comportamiento eléctrico del corazón. Las ACs son un grupo de anomalías que pueden ser mortales si no son detectadas a tiempo,

su descubrimiento es posible mediante la realización de un electrocardiograma (ECG). Un ECG no es más que un trazo bidimensional que refleja el comportamiento eléctrico del corazón, comúnmente su representación suele ser como una serie temporal [21, 22].

Datasets en línea como lo es [20], contiene una basta cantidad de electrocardiogramas (ECGs) con distintos tipos de ACs, estos datos son cruciales para la propuesta de algoritmos de IA. Sin embargo, cómo ya se mencionó con anterioridad, el rendimiento que han demostrado estos modelos híbridos, no solo en esta área de la medicina, sino también en otros campos de la ciencia, despierta el interés de investigadores para la propuesta continua de estos modelos, y hablando específicamente en el área de la salud; se busca el modelo con el mejor rendimiento para que agencias reguladoras internacionales lo adopten como una opción capaz de ser adherida a tecnología sanitaria en desarrollo.

Por consiguiente, en el presente trabajo de tesis se propone un modelo híbrido computacional compuesto por dos algoritmos de IA: el primero de ellos consiste en una CNN; que pertenece a la familia de los AAP, y el segundo; al PSO de la familia de los AMH. El modelo computacional desarrollado tiene como objetivo la clasificación satisfactoria de cinco tipos de clases de ACs, que fueron obtenidas de [20]. Esencialmente nuestro modelo propuesto hace uso del potencial que tiene la CNN como algoritmo clasificador y toma ventaja de la metodología de búsqueda de la mejor partícula a cargo del PSO. Aunque la funcionalidad del modelo se detalla en el capítulo 4, es crucial que el lector conozca desde este momento, que nuestro modelo computacional tiene como fin encontrar la mejor arquitectura de capas con sus correspondientes hiperparámetros (número y tipo de capas, configuración para cada

tipo de capa, número de neuronas, número de épocas, entre otros), que le permita a la CNN clasificar de forma satisfactoria las cinco clases distintas de ACs. Dejando así, el trabajo de búsqueda de la mejor partícula al PSO, cada partícula del AMH se encuentra definida por una determinada arquitectura de capas y sus correspondientes hiperparámetros (H) de la propia CNN, la generación de la población es aleatoria.

La clasificación de las ACs, es un tema de investigación al que se le han planteado diversas soluciones, que van desde las técnicas clásicas, la implementación de los AAA o AAP, los AMH, y recientemente la propuesta de modelos híbridos compuestos por cualquiera de los anteriormente mencionados. El objetivo que tienen en común todos los investigadores que han desarrollado estos modelos computacionales, consiste en mejorar el rendimiento del algoritmo clasificador, obteniendo precisiones por arriba del 98% de precisión.

El modelo propuesto en el presente trabajo, denominado modelo computacional híbrido de una Red Neuronal Convolutiva con sintonización en sus hiperparámetros por un algoritmo de Optimización de Enjambre de Partículas (H-PSO-CNN), no solo obtiene un rendimiento satisfactorio en la clasificación de cinco clases de ACs, sino también, es capaz de obtener una arquitectura de capas para la CNN de forma automática. En capítulos posteriores se describe el fundamento teórico de cada algoritmo, así como el fundamento metodológico del H-PSO-CNN.

Planteamiento del Problema

Aun cuando la implementación de los AAP-AAA demuestra obtener mejor rendimiento que el aprendizaje humano, en algunos algoritmos computacionales la selección de parámetros o hiperparámetros sigue siendo una tarea que está a cargo de una entidad humana, lo que lleva a un proceso de selección del modelo más tardado y con un mayor margen de error, debido a la experimentación de prueba y error hasta lograr el mejor ajuste de dichos parámetros. Por lo tanto, el planteamiento de este trabajo es la determinación óptima de hiperparámetros en la CNN mediante un algoritmo computacional metaheurístico como el PSO que minimice los errores en la predicción de las ACs. Finalmente, la heterogeneidad de los datos provoca un bajo rendimiento en los AAP o AAA debido a que el dominio de información de clases mayores vs otras menores, provoca que la mayoría de los modelos de IA discriminen las clases minoritarias si la configuración de los parámetros no es la óptima.

Justificación

La propuesta de modelos de aprendizaje computacional compuestos por algoritmos de IA; específicamente hablando de los AAA, AAP y AMH o la hibridación entre ellos, ha incrementado considerablemente en al menos los últimos 10 años. El objetivo de cada trabajo de investigación desarrollado sin importar el campo de aplicación científico, médico o industrial, siempre es el mismo, y consiste en obtener el mejor rendimiento posible del algoritmo en cuestión, y con ello garantizar un alto porcentaje

de precisión según corresponda la tarea de predicción encomendada.

Lo expuesto anteriormente, permite garantizar la correcta adhesión de este tipo de algoritmia en tecnologías sanitarias de última generación, contando con el aval y autorización en lo que respecta a su comercialización, de agencias internacionales como la FDA, encargadas de la vigilancia médica y/o sanitaria.

El uso de algoritmos de IA, cada vez es más frecuente en diferentes campos científicos, en la salud se aprovecha el potencial de estos modelos para llevar a cabo tareas de clasificación o estimación. Por ejemplo: el oportuno diagnóstico de enfermedades como lo son el cáncer, seguimiento y monitoreo continuo de señales biológicas para la notificación de alguna anomalía, reconocimiento de patrones en imágenes médicas, entre otros. La electrónica de las tecnologías sanitarias comienza a equiparse con este tipo de algoritmia, aprovechando al máximo sus capacidades computacionales y alcance de información. Situación que motiva el desarrollo de nuevos modelos computacionales que permitan obtener resultados satisfactorios, y con ello asegurar la aprobación de dicha tecnología por parte de agencias internacionales como lo es la FDA, quienes realizan estrictas pruebas a cada nueva propuesta, con el fin de garantizar una tecnología sanitaria confiable, precisa y segura; tanto para el personal de salud cómo para los pacientes.

La identificación de las ACs, en la actualidad es un proceso que se realiza mediante la inspección del trazo de la señal eléctrica del corazón plasmada en un ECG, dicha inspección es realizada por un especialista médico quién regularmente es un cardiólogo, y es el encargado de identificar la existencia de características particulares de la señal que le permitan diagnosticar el tipo de arritmia cardíaca (AC). Lo anterior,

implica un alcance limitado en relación a la cantidad de pacientes vs. especialistas, y al ser un proceso humano, es propenso a un margen de error mucho mayor que el de un algoritmo de IA; debido a la cantidad de datos que puede aprender el médico especialista vs. AAA-AAP-AMH. Además, hoy en día la mayoría de los dispositivos electrónicos (electrocardiógrafos) son los encargados de obtener un ECG, si bien es cierto que cuentan con la identificación de las ACs, la precisión y el número de ACs que puede diagnosticar sigue siendo insuficiente para las demandas médicas actuales.

A lo anteriormente descrito, se agrega el desbalance que existe en los Datasets relacionados con el campo de la salud, es decir, puede existir un mayor número de muestreos de arritmias sanas y un mínimo de algún tipo de arritmia poco frecuente en los pacientes, a esto se le conoce como la heterogeneidad en los datos. Este acontecimiento sucede a menudo cuando se trabajan con datos en materia de salud.

Propuesta de Solución

Tomando en consideración que la clasificación de las ACs es un proceso que en la actualidad es posible a través de la observación y aprendizaje por parte de un especialista médico. Lo anterior, motiva a la propuesta de soluciones automatizadas que realicen la misma tarea, en el menor tiempo y con una mayor precisión. Una estrategia computacional híbrida es una opción ante el problema planteado, la naturaleza de estos algoritmos permite que sean capaces de realizar tareas de clasificación, predicción y optimización, por lo que, son capaces de llevar el mismo proceso de observación y aprendizaje, incrementando la velocidad y la cantidad de datos; algo que para un

especialista médico es complicado.

La propuesta de solución consiste en el diseño de una estrategia computacional híbrida, compuesta por una CNN y un algoritmo PSO para la obtención de un rendimiento satisfactorio en la clasificación de ACs. El PSO es el encargado de aumentar dicho rendimiento a través de la minimización del error calculado por parte de la CNN en cada una de las clasificaciones, cada partícula de la población del PSO se encuentra definida por una configuración de arquitectura de capas con sus correspondientes H que le permiten a la CNN llevar a cabo su tarea principal en este caso y que consiste en la clasificación de cinco categorías distintas de ACs. Cada partícula es actualizada conforme a los operadores propios del PSO, con el objetivo de minimizar el error en la identificación de las ACs.

Hipótesis

Una CNN por naturaleza es un algoritmo clasificador con un buen rendimiento en su tarea, sin embargo, su limitante es la definición de algunos de sus hiperparámetros de forma manual. Por otra parte, un PSO resulta ser un algoritmo capaz de encontrar una solución óptima global en un determinado espacio de búsqueda. Entonces, una hibridación en la cual el PSO se encargue de buscar una solución satisfactoria que contenga hiperparámetros óptimos de la CNN, permitirá obtener un rendimiento satisfactorio en la clasificación de arritmias cardíacas.

Objetivos de la Tesis

Los objetivos de este trabajo de investigación son los siguientes:

Objetivo General

Diseñar y evaluar un modelo computacional que obtenga una solución satisfactoria en la clasificación de arritmias cardíacas a través de la optimización de los hiperparámetros de una red neuronal convolucional mediante el uso del algoritmo optimización por enjambre de partículas.

Objetivos Específicos

1. Obtener un banco de datos certificado y avalado por organismos internacionales, que contenga al menos cinco distintos tipos de arritmias cardíacas.
2. Normalizar el banco de datos de tal manera que se representen como entradas correctas para la red neuronal convolucional.
3. Definir los parámetros iniciales y estructura de la población del PSO que permitan llevar a cabo una correcta hibridación.
4. Desarrollar el núcleo del modelo propuesto, considerando las funcionalidades y restricciones de los algoritmos utilizados.
5. Implementar la hibridación del PSO en la optimización de la CNN, para reducir la función de coste de la CNN en su tarea de clasificación y con ello

encontrar a la mejor partícula con su correspondiente configuración de capas e hiperparámetros.

6. Obtener el rendimiento de la CNN con los datos de la mejor partícula obtenida por el PSO previamente y con el banco de datos de entrenamiento.
7. Comparar los resultados con otros modelos computacionales que utilicen a la CNN como algoritmo clasificador.

Estado del Arte

Antes de la aplicación de los AAA, AAP o AMH para tareas de clasificación o predicción, la clasificación de las ACs se fundamentaba principalmente en el uso de algoritmos tradicionales que dependen de la extracción y clasificación de características [23, 24]. El proceso en estos algoritmos solía ser más complejo y con una serie de procedimientos adicionales, mismos que son incluidos y de cierta forma automatizada en los AAP. Entre los algoritmos tradicionales destacan las técnicas estadísticas, como lo es el modelo de Markov [25], hasta el uso de los AAA que incluyen a las máquinas de vectores de soporte (SVM)[26, 27] , algoritmo de k-vecinos más cercanos (KNN)[28, 29], bosques aleatorios (RF) [30], análisis de componentes principales (PCA)[27], entre otros [31, 32]. El uso que se le da a cada uno de los anteriores algoritmos, principalmente se enfoca en tareas de clasificación o predicción, y no significa que sean los únicos disponibles en la literatura. Sin embargo, una de las desventajas que presentan los AAA en este tipo de tareas es su improductividad para

la mayoría de las demandas que exigen las aplicaciones modernas, y si a esto le agregamos el aumento en la cantidad de información digital que se produce día a día; trae como consecuencia que los AAA requieran de una gran cantidad de observaciones para lograr la generalización.

Los AAP resultan ser más completos y sofisticados a diferencia de los AAA, al permitir el incremento en la profundidad de capas internas que posee la red neuronal. Lo anterior, se traduce en la eliminación de ciertas necesidades de preprocesamiento que los AAA requieren para su correcto funcionamiento, ayudando al aprendizaje y a la obtención de resultados con una mayor precisión [33].

Existen un gran número de trabajos de investigación que han abordado la clasificación de las ACs a través de técnicas de aprendizaje profundo, y la mayoría de estos han sido publicados en los últimos diez años. Entre este universo, algunos de los DLA con mayor participación e interés en su aplicación para la clasificación de ACs son los siguientes: perceptrón multicapa (MLP) [34], red neuronal recurrente (RNN) [35], red neuronal de memoria a corto plazo (LSTM) [36] y la CNN[37][38][39][40]; esta última con una mayor frecuencia de aparición.

En [36] propusieron el uso del MLP para la clasificación de cinco clases de ACs en un ECG, utilizando como fuente de información a MIT-BIH Arrhythmia Dataset (MITDB), logrando obtener un rendimiento del 98.72%. Otro ejemplo del uso de MLP, es el propuesto por [41], quienes trabajaron con la clasificación de dos diferentes clases de fibrilación atrial, su fuente de datos la obtuvieron del MIT-BIH Atrial Fibrillation database (MITAFD), reportando un rendimiento del 80.67%.

Cómo se mencionó con anterioridad, otro tipo de red neuronal preferida en la

clasificación de ACs, es la RNN y también la LSTM. Por ejemplo, en [39] realizaron un estudio comparativo que les permitió evaluar el rendimiento de dos RNN y una RNN en combinación con una LSTM, usando dos clases del MITDB; este trabajo de investigación reportó un rendimiento del 85.4%, 82.5% y 88.1%, respectivamente. En [38] trabajaron con la clasificación de la fibrilación atrial utilizando un sistema embebido para la adquisición de las señales ECG y diferentes tipos de redes neuronales recurrentes, esta propuesta logró un rendimiento mínimo del 95%. En cambio, en [37] utilizaron 8 clases de arritmias que fueron tomadas del MITDB, y mediante el uso de una LSTM lograron un rendimiento del 99.26%. En [42] proponen el uso de una LSTM basada en una RNN profunda para una clasificación de 3 a 9 clases distintas de ACs, tomadas del MIT-BIH Normal Sinus Rhythm y del MITDB, logrando obtener un rendimiento del 99.73% para tres clases y 99.80% para 9 clases distintas de ACs. Finalmente, en [40] se propone una LSTM bidireccional que trabaja con los datos obtenidos tanto del MITAFD y del MITDB, esta propuesta les permitió obtener un rendimiento del 99.1% y 98.4%.

Aunque el uso de los anteriores AAP en la clasificación de ACs han demostrado ser algoritmos confiables en dicha tarea, no significa que sean los AAP con mayor preferencia. La CNN es el algoritmo AAP que cuenta con mayor número de publicaciones en la clasificación de ACs. Dentro de estas publicaciones recientes destaca el trabajo desarrollado por [43], la cual consiste en usar una CNN de dos dimensiones, con una arquitectura de capas definida de forma manual para la clasificación de 2, 7 y 8 distintas clases de ACs, los resultados de esta investigación obtuvieron un rendimiento de 98.91% para 2 clases, 98.10% para 7 clases y 98.45% para 8 clases. Otro trabajo

que incluye a la CNN, pertenece a [44], quienes hicieron uso de una CNN de una dimensión y clasificaron 16 arritmias distintas provenientes del MITDB, reportando un rendimiento del 98% durante el entrenamiento y 95% en las pruebas. A diferencia de [44], en [45] trabajaron también con una CNN para la clasificación de 5 distintos tipos de clases de ACs, obtenidas del MITDB, logrando obtener un rendimiento del 99.06%. En [46] también utilizaron a la CNN, trabajaron con seis distintas clases de ACs, sus resultados experimentales reportan un rendimiento de prueba cuyos valores varían del 95.3% al 94.41%. Sin embargo, tanto en [45] y en [46], la arquitectura de capas con sus correspondientes hiperparámetros se realizó de forma artesanal.

También se han propuesto modelos híbridos compuestos entre al menos dos tipos de redes neuronales. En [47] proponen un modelo compuesto por CNN-LSTM para la clasificación de seis clases de CAs, dicho modelo fue capaz de obtener un rendimiento del 99.89%. De igual manera, en [48] proponen el uso de CNN-LSTM aplicado en la clasificación de la fibrilación atrial cuyos datos fueron tomados del MITAFD, reportando que su modelo fue capaz de obtener un rendimiento del 97.87%.

Se puede observar que la red neuronal con mayor presencia en estos trabajos de investigación, cómo ya se mencionó con anterioridad, es la CNN. Esta red neuronal por si sola cuenta con un 52% de participación, un 10.6% en modalidad híbrida (CNN+LSTM) y un 6.6% utilizando (CNN+RNN) [49].

Por otra parte, algunos investigadores han encontrado que la aplicación de los AMH en la optimización de redes neuronales, aumenta la precisión en la clasificación de ACs. Sin embargo, la cantidad de publicaciones disponibles usando este tipo de estrategia es muy mínima comparada con las que hacen uso de cualquier técnica

de aprendizaje automático o profundo. En [48] proponen la hibridación entre un AMH denominado depredadores marinos (MPA) en conjunto con la CNN para la clasificación de seis clases de CAs tomadas del MITDB, European ST-T y el ST Petersburg Incart Dataset; su modelo propuesto logró un rendimiento del 99.32%, 99.76% y 99.47% respectivamente. Otro ejemplo de hibridación, es el propuesto por [50], quienes trabajaron con el MA Bat-Rider Optimization (BaROA) en conjunto con la CNN, utilizando el MITDB como fuente de datos, su modelo fue capaz de obtener un 93.19% de precisión. Otra combinación pertenece a [51], quienes usaron el MLP y el PSO en 5 clases de CAs del MITDB, obteniendo un rendimiento del 98.56%. De manera similar, en [52] trabajaron con una versión modificada del PSO y también con el MLP, de igual forma en 5 clases de CAs del MITDB, logrando obtener una precisión del 99.44%. Y, en [53] optaron por implementar el modified Pigeon Inspired Optimizer (MPIO) para optimizar las características del ECG y la red neuronal profunda (DNN), obteniendo un rendimiento de 99.10%.

A diferencia de los trabajos que se han mencionado con anterioridad, en [54] hicieron uso de tres estrategias de aprendizaje automático y profundo (SVM, la red neuronal retroalimentada de propagación hacia adelante y la transformada discreta wavelet) en conjunto con el algoritmo de optimización de búsqueda por cuckoo (COA), obteniendo una precisión del 98.53%. También se han considerado hibridaciones entre técnicas clásicas de aprendizaje automático. Por ejemplo, en [55] desarrollaron un modelo híbrido basado en el algoritmo genético para la optimización de características de un árbol de decisiones, reportando un rendimiento del 86.96%. Por último, otro ejemplo es el propuesto por [46], quienes utilizan el algoritmo de máquinas de

vectores de soporte gemelos combinados (MVSGC) con un modelo híbrido de AMH compuesto por el PSO y el algoritmo de búsqueda gravitacional; estos últimos dos para la optimización de los parámetros del MVSGC, su modelo fue capaz de obtener un rendimiento del 99.44%.

Existen las propuestas de otros modelos híbridos entre los MA y las redes neuronales, pero su aplicación es distinta a la clasificación de CAs. En particular solo mencionaremos dos: el desarrollado por [56], que se fundamenta en el uso del PSO-CNN-MVS, con el objetivo de clasificar enfermedades renales crónicas, y la propuesta de [57], basada en un modelo híbrido compuesto por el algoritmo optimizador de apareamiento de percebes con MVS, para la selección de genes en la clasificación de cáncer.

La mayoría de los trabajos mencionados con anterioridad se centra en el uso de técnicas de aprendizaje profundo. La aparición de modelos híbridos que incluye a los AMH y los AAP, es un suceso que se está volviendo más frecuente en distintos campos de la ciencia, y la medicina no es la excepción.

Aportaciones

Las contribuciones del presente trabajo de tesis se resumen en:

- El desarrollo de un algoritmo híbrido que es capaz de realizar una búsqueda automática de los hiperparámetros de la arquitectura que una CNN requiere para realizar su tarea de clasificación.

- La reducción de tiempo y costo computacional en la búsqueda de los hiperparámetros adecuados para la arquitectura de una CNN, dejando al PSO esta tarea de forma automatizada, encontrando la arquitectura en un tiempo de 17.68 horas; logrando así prescindir del trabajo artesanal que hoy en día se realiza en la selección de estos hiperparámetros.
- Un intérprete que traduce la población generada para el funcionamiento del PSO, a una población de partículas cuyo contenido es entendible por la CNN, con la finalidad de obtener para cada partícula su correspondiente función costo, que corresponde al valor del error. También, dicho intérprete es capaz de actualizar la población tanto del PSO como para la CNN, según los nuevos valores y cambios calculados por parte de los operadores del metaheurístico. Este intérprete es el núcleo del H-PSO-CNN, sin él, la interacción entre el MA y la CNN, no sería posible.
- El modelo H-PSO-CNN, almacena la mejor arquitectura de capas encontrada por el algoritmo metaheurístico, junto con sus correspondientes hiperparámetros, lo que permite obtener el rendimiento de dicha configuración con datos de prueba que no fueron aprendidos por la CNN.
- Un marco computacional confiable y universal, que puede ser probado en otros campos de la ciencia, siempre y cuando correspondan a tareas de clasificación.

Organización de la Tesis

El presente trabajo de tesis se ha organizado de la siguiente manera:

Capítulo 1. Arritmias Cardíacas, se describen los fundamentos teóricos de las arritmias cardíacas: concepto, tipos, fisiología, medios de diagnóstico, entre otros aspectos relevantes que le permitirán al lector comprender la etiología de este tipo de patología.

Capítulo 2. Redes Neuronales Convolucionales, explica el marco teórico de la red neuronal convolucional, sus componentes y el principio de funcionamiento de la red.

Capítulo 3. Algoritmo Metaheurístico PSO, explica el fundamento teórico del algoritmo meta-heurístico junto con sus operadores, la fuente de bioinspiración y la metodología de funcionamiento del algoritmo.

Capítulo 4. Algoritmo H-PSO-CNN, se detalla la propuesta de modelo computacional propuesto en este trabajo de tesis, dónde se especifica la funcionalidad que tiene el PSO y la CNN en este modelo híbrido.

Capítulo 5. Pruebas y Resultados, se muestran las pruebas experimentales con el conjunto de datos utilizado y los resultados obtenidos por parte del modelo propuesto en este trabajo de investigación.

Capítulo 6. Conclusiones y Trabajos Futuros, se describen las conclusiones y se proponen trabajos futuros que se pueden llevar a cabo con el modelo computacional desarrollado.

Capítulo 1

Arritmias Cardíacas

El corazón humano es un órgano muscular hueco, del tamaño aproximado de un puño, situado en el mediastino (la región central del tórax, entre los pulmones). Su función principal es bombear la sangre a través del sistema circulatorio, suministrando oxígeno y nutrientes a los tejidos del cuerpo y eliminando dióxido de carbono y otros desechos [58]. El corazón se compone de cuatro cámaras: dos aurículas (izquierda y derecha) que reciben la sangre, y dos ventrículos (izquierdo y derecho) que expulsan la sangre hacia el cuerpo. La contracción rítmica y coordinada de estas cámaras permite la circulación eficiente de la sangre, mantenida a través de un sistema eléctrico intrínseco que regula la frecuencia y el ritmo cardíacos [59].

El corazón humano, como órgano encargado de bombear sangre a todo el cuerpo, funciona gracias a impulsos eléctricos específicos que determinan su ritmo. Una arritmia cardíaca se refiere a cualquier irregularidad o anomalía en la secuencia y/o frecuencia de estos impulsos eléctricos. La relación del corazón con las arritmias se

puede describir en términos de cómo y por qué estas irregularidades ocurren y cómo afectan la función cardíaca.

Las arritmias cardíacas son trastornos del ritmo cardíaco que pueden afectar a personas de todas las edades y condiciones de salud. Desde latidos acelerados hasta pausas inquietantes que dejan al corazón en silencio por un momento, las arritmias pueden manifestarse de diversas formas, con consecuencias que van desde leves molestias hasta amenazas graves para la vida. Una arritmia cardíaca es una anomalía en el ritmo o la secuencia de los latidos del corazón, lo que significa que el corazón late de manera irregular, ya sea demasiado rápido (taquicardia), demasiado lento (bradicardia) o de manera irregular (arritmia)[59, 60].

Estas alteraciones pueden ser causadas por diversos factores, como problemas eléctricos en el corazón, enfermedades cardíacas subyacentes, trastornos del sistema nervioso o desequilibrios químicos en el organismo [61]. Las arritmias cardíacas pueden variar en gravedad y pueden manifestarse con una amplia gama de síntomas, desde palpitaciones leves hasta desmayos o eventos potencialmente mortales[62]. El diagnóstico oportuno y el tratamiento adecuados son esenciales para gestionar las arritmias cardíacas y prevenir complicaciones graves en la salud.

En este capítulo se abordan conceptos relacionados con las ACs incluyendo sus antecedentes históricos, exploraremos las causas detrás de estos latidos fuera de ritmo, los diferentes tipos de arritmias, sus síntomas, cómo se diagnostican y tratan, y las estrategias para prevenir su aparición. A medida que adentramos en este tema, comprenderemos la importancia de la función cardíaca normal y cómo los avances médicos y tecnológicos nos permiten enfrentar estos desafíos con mayor conocimiento y pre-

cisión.

1.1 Antecedentes Históricos

Los antecedentes históricos de las arritmias cardíacas se remontan a la antigüedad, aunque en aquel entonces no se comprendían completamente ni se trataban de manera efectiva. En los siguientes párrafos se describen de manera resumida cada una de las etapas históricas relevantes que han aportado en el desarrollo de este concepto.

1.1.1 Antigüedad

En la antigüedad, las arritmias cardíacas se entendían de manera limitada y se atribuían a causas misteriosas o sobrenaturales. Los primeros textos médicos, como el Papiro de Ebers (c. 1550 a.C.), describen algunas anomalías del pulso y el corazón, aunque la comprensión de estas era extremadamente básica y fundamentada en la magia y la religión [63].

La antigua Grecia aportó un enfoque más sistemático con médicos como Hipócrates (c. 460-370 a.C.) que, aunque aún limitados por la comprensión científica de su época, comenzaron a clasificar y documentar diferentes tipos de pulsos, reconociendo variabilidades y anomalías [64].

1.1.2 Edad Media y Renacimiento

En la Edad Media, la medicina islámica, particularmente en la figura de Avicena (Ibn Sina), en su "Canon de Medicina" (1025), introdujo una descripción más detallada de las alteraciones del pulso y señaló la relevancia de observar las irregularidades del ritmo cardíaco [65].

En el Renacimiento, la obra de médicos y anatomistas como Andreas Vesalio (1514-1564) impulsó un aumento en la comprensión de la anatomía humana, aunque el entendimiento de las arritmias seguía siendo limitado y basado en observaciones superficiales y especulativas [65].

1.1.3 Siglos XVII al XIX

William Harvey, en el siglo XVII, realizó descubrimientos cruciales respecto a la circulación de la sangre, aunque el mecanismo exacto de la activación eléctrica del corazón permanecía desconocido. En el siglo XVIII y XIX, la medicina comenzó a adoptar un enfoque más científico hacia las arritmias.

La invención del primer electrocardiograma (ECG) en 1786 por Luigi Galvani y posteriormente por Alessandro Volta sentó las bases para el estudio moderno de las arritmias. El ECG permitió registrar y analizar la actividad eléctrica del corazón, lo que revolucionó la comprensión y el diagnóstico de las arritmias.

En la década de 1840, Carl Ludwig y Augustus Waller realizaron investigaciones significativas en electrofisiología cardíaca, sentando las bases para la comprensión de los ritmos cardíacos anormales. Jean-Baptiste Bouillaud describió por primera vez la

fibrilación auricular en 1847. En 1842-1843, Carlo Matteucci fue uno de los primeros en observar que cada latido cardíaco estaba asociado con una corriente eléctrica al experimentar con palomas. Para 1856, Rudolf Albert von Kölliker y Heinrich Müller, lograron registrar la actividad eléctrica del corazón de un gato y un conejo usando un electrodo de electrofisiología. Finalmente, en 1887, Augustus Waller logró registrar el primer ECG humano utilizando un electrómetro capilar.

1.1.4 Siglo XX

1903, Willem Einthoven desarrolló el primer ECG práctico, utilizando un instrumento llamado "galvanómetro de cadena", mucho más preciso y fiable que los electrómetros capilares. Por este avance, Einthoven fue galardonado con el Premio Nobel de Fisiología o Medicina en 1924 [65, 64].

Willem Einthoven acuñó la terminología P, Q, R, S y T para describir las diferentes fases del ECG, que se sigue utilizando hoy en día, y permite la detección de las arritmias cardíacas.

De 1920 a 1930, el ECG empezó a ser utilizado de manera regular en hospitales, contribuyendo en el diagnóstico y manejo de múltiples enfermedades cardíacas. En los 1950s, se inició el desarrollo y la popularización de los monitores cardíacos en la atención de pacientes.

En la década de 1950, Norman Holter desarrolló el monitor Holter, un dispositivo portátil que permitía la monitorización continua del ritmo cardíaco durante 24 horas, lo que mejoró significativamente la detección de arritmias. Por otra parte en esa

misma época, la tecnología de marcapasos cardíacos comenzó a desarrollarse y ha sido fundamental en el tratamiento de bradicardias. Para la época de 1960, el avance tecnológico de los microprocesadores y la aparición del concepto de digitalización, permitieron que los electrocardiogramas, que son dispositivos para la obtención de ECGs, se hicieran más pequeños, portátiles, y accesibles.

De 1980 a 1990, la telemedicina comenzó a adoptar la tecnología ECG, permitiendo el monitoreo remoto y la transferencia de datos cardíacos a través de líneas telefónicas, lo que sentó las bases para el desarrollo de nuevas tecnologías mucho más equipadas y que aparecieron en el siguiente siglo.

Se puede observar que a lo largo del siglo XX, el desarrollo de marcapasos, desfibriladores y otras intervenciones tecnológicas mejoraron drásticamente la gestión y el tratamiento de las arritmias.

1.1.5 Siglo XXI

La tecnología del siglo XXI ha demostrado ser fundamental en la detección y gestión de las arritmias cardíacas debido a diversos factores que han mejorado tanto la accesibilidad como la precisión en la evaluación del ritmo cardíaco.

La disponibilidad de wearables y dispositivos portátiles permite monitorear el ritmo cardíaco en tiempo real y en situaciones cotidianas, ofreciendo una perspectiva más completa y continua de la salud cardíaca del individuo. Regularmente este tipo de dispositivos modernos pueden proporcionar alertas inmediatas tanto a los pacientes como a los profesionales de la salud en caso de detectar arritmias o eventos cardíacos

adversos.

Hoy en día, la tecnología permite recoger y analizar datos específicos del paciente, facilitando un enfoque de tratamiento más personalizado y basado en evidencias específicas. Esta recolección de datos hace posible la capacidad de compartir datos cardíacos con profesionales de la salud remotamente amplía el acceso a la atención médica especializada, especialmente en regiones donde esta puede ser limitada.

Por otra parte la propuesta de algoritmos de IA ha incrementado considerablemente, debido a la aptitud que les permite analizar grandes volúmenes de datos para identificar patrones y anomalías en el ritmo cardíaco, incluso aquellos que podrían ser sutiles o esporádicos y, por ende, difíciles de detectar en un ECG estándar en el consultorio.

La comprensión y el tratamiento de las arritmias cardíacas han evolucionado significativamente a lo largo de la historia, pasando de explicaciones místicas a un entendimiento científico y tecnológico más profundo. El desarrollo del ECG y la innovación continua en tecnología médica han sido hitos cruciales en esta evolución, permitiendo diagnósticos más precisos y tratamientos más efectivos.

La intersección de la tecnología y la cardiología en el siglo XXI ha facilitado avances significativos en la detección y manejo de las arritmias cardíacas, creando un paradigma donde la atención es más accesible, personalizada y eficaz. Este progreso no solo mejora los resultados clínicos sino que también potencia la autonomía del paciente y amplía las posibilidades de investigación y desarrollo en el campo de la salud cardíaca.

1.2 Causas Principales

Las arritmias cardíacas pueden ser causadas por una variedad de factores y condiciones médicas, que van desde los problemas de tiroides, trastornos electroquímicos, consumo o abuso de ciertas sustancias, el estrés y ansiedad, lesiones cardíacas traumáticas, factores genéticos y propiamente el envejecimiento. Sin embargo, también pueden ser causadas por enfermedades o trastornos relacionados con la propia fisiología del corazón, y éstas son de interés para el presente trabajo de tesis debido a que comúnmente son detectadas a través de un ECG, estas enfermedades o trastornos son:

Enfermedades Cardíacas Subyacentes

- **Cardiopatía Isquémica:** La enfermedad de las arterias coronarias, que causa la reducción del flujo sanguíneo al corazón, puede llevar a arritmias, como la fibrilación ventricular.
- **Insuficiencia Cardíaca:** La incapacidad del corazón para bombear sangre de manera efectiva puede alterar su ritmo normal.
- **Miocardopatías:** Las enfermedades que afectan el músculo cardíaco pueden provocar arritmias, como la taquicardia ventricular.
- **Valvulopatías Cardíacas:** Anomalías en las válvulas cardíacas pueden desencadenar arritmias.

Trastornos del Sistema Eléctrico del Corazón

- Enfermedad del Nodo Sinusal: Cuando el nodo sinusal, que controla la frecuencia cardíaca, no funciona correctamente, pueden ocurrir bradicardias.
- Bloqueo Auriculoventricular: Un fallo en la conducción eléctrica entre las aurículas y los ventrículos puede causar bradicardias.

Es importante señalar que las causas de las arritmias pueden variar ampliamente y, en muchos casos, pueden ser multifactoriales. El diagnóstico adecuado es fundamental para determinar la causa subyacente y guiar el tratamiento apropiado.

1.3 Tipos de Arritmias

Existen varios tipos de arritmias cardíacas, que pueden clasificarse en tres categorías principales: taquicardias, bradicardias y otras arritmias. Cada una de estas categorías incluye una variedad de subtipos y trastornos específicos. Comprender los distintos tipos de arritmias es fundamental para su diagnóstico y tratamiento adecuados.

1.3.1 Taquicardias

Las taquicardias son un grupo de arritmias caracterizadas por latidos cardíacos rápidos, generalmente con una frecuencia cardíaca superior a 100 latidos por minuto. Algunos de los tipos de taquicardias más comunes son:

- Fibrilación Auricular (FA): La FA es una de las arritmias más comunes y se caracteriza por contracciones caóticas y descoordinadas de las aurículas. Esto

puede resultar en una frecuencia cardíaca irregular y rápida. Los pacientes con FA a menudo experimentan palpitaciones, mareos y un mayor riesgo de formación de coágulos sanguíneos.

- Taquicardia Supraventricular (TSV): Las TSV son taquicardias que se originan por encima de los ventrículos y pueden involucrar estructuras como el nodo auriculoventricular (NAV). Pueden ser paroxísticas, lo que significa que ocurren súbita y temporalmente, y a menudo se detienen por sí solas.
- Taquicardia Ventricular (TV): La TV se produce cuando se generan impulsos eléctricos anormales en los ventrículos en lugar de en las aurículas. Esto puede resultar en una frecuencia cardíaca extremadamente rápida y, en casos graves, puede ser potencialmente mortal.

1.3.2 Bradicardias

Las bradicardias son arritmias caracterizadas por una frecuencia cardíaca lenta, generalmente por debajo de 60 latidos por minuto. Algunos ejemplos de bradicardias incluyen:

- Síndrome del Nodo Sinusal Enfermo (SNSE): En este trastorno, el nodo sinusal, que normalmente genera el impulso eléctrico inicial del corazón, no funciona correctamente. Esto puede llevar a episodios de bradicardia y, en ocasiones, taquicardia.
- Bloqueo AV (Bloqueo Auriculoventricular): El bloqueo AV es una interrupción

en la conducción del impulso eléctrico entre las aurículas y los ventrículos. Se clasifica en primer, segundo y tercer grado, según la gravedad de la obstrucción.

1.3.3 Otras Arritmias

Además de las taquicardias y bradicardias, existen otras arritmias menos comunes, como:

- Flutter Auricular: Similar a la fibrilación auricular, pero con contracciones auriculares más regulares y rápidas.
- Taquicardia Ventricular Polimórfica (TVP): Una forma rara de TV que puede ser desencadenada por un trastorno genético.
- Taquicardia Ventricular Sostenida (TVS): Una forma continua de TV que puede ser potencialmente mortal si no se trata.
- Taquicardia sinusal inapropiada: Una arritmia que se produce cuando el nodo sinusal del corazón genera impulsos eléctricos a una frecuencia demasiado alta en reposo.

Estos son solo algunos ejemplos de los tipos de arritmias cardíacas. Cada tipo de arritmia puede tener múltiples subtipos y variaciones, y el diagnóstico preciso es fundamental para determinar el tratamiento adecuado. Las arritmias pueden ser benignas o potencialmente mortales, por lo que el diagnóstico oportuno de las mismas es crucial para preservar la vida de las personas.

1.4 Síntomas, Diagnóstico y Tratamiento

1.4.1 Síntomas

Los síntomas de las arritmias cardíacas pueden variar ampliamente según el tipo de arritmia, su gravedad y la salud general del individuo. Algunas personas pueden no experimentar ningún síntoma, mientras que otras pueden experimentar molestias significativas. Entre los síntomas más comunes se encuentran: palpitaciones, mareo o desmayo (síncope), dificultad para respirar (disnea), dolor en el pecho (angina de pecho), fatiga, sensación de latido irregular, ansiedad o sensación de pánico, hinchazón en los tobillos o edema

Es importante señalar que no todas las arritmias causan síntomas evidentes, y algunas pueden detectarse solo durante un examen médico de rutina o en un electrocardiograma (ECG) [62]. La gravedad de los síntomas no siempre se correlaciona con la gravedad de la arritmia; algunas arritmias benignas pueden causar síntomas notables, mientras que otras graves pueden ser asintomáticas.

1.4.2 Diagnóstico

El diagnóstico de las arritmias cardíacas implica identificar y caracterizar irregularidades en el ritmo del corazón. El diagnóstico preciso es esencial para guiar el tratamiento adecuado. Se utilizan varios métodos y herramientas para ayudar en este proceso: ECG, monitoreo holter, monitoreo de eventos, prueba de esfuerzo, estudio electrofisiológico, ecocardiograma, monitores implantables, prueba de inclinación

(tilt test), pruebas de laboratorio e imagen por resonancia magnética. La mayoría de los métodos mencionados con anterioridad basan su principio de funcionamiento a través de la lectura e interpretación de la señal que producen los impulsos eléctricos del corazón.

Estos son algunos de los métodos principales, y la elección del método dependerá de la naturaleza de los síntomas, la sospecha clínica y la disponibilidad de recursos.

1.4.3 Tratamiento

El tratamiento de las arritmias cardíacas depende de la naturaleza y la gravedad de la arritmia, así como de la condición subyacente del corazón y la presencia de otros problemas médicos. Los objetivos del tratamiento son controlar el ritmo cardíaco, reducir los síntomas y las complicaciones, y tratar cualquier enfermedad cardíaca subyacente que pueda estar causando o contribuyendo a la arritmia. Entre las principales estrategias de tratamiento se encuentran: medicamentos antiarrítmicos, cardioversión, ablación con catéter, marcapasos cirugía de derivación coronaria y modificación del estilo de vida.

1.5 Morfología Básica del ECG

El ECG es una herramienta poderosa para detectar arritmias y otras enfermedades cardíacas. Sin embargo, interpretar un ECG requiere experiencia y formación. Si se sospecha una arritmia o cualquier otra anomalía cardíaca basada en un ECG, es fundamental que un cardiólogo o un médico especializado realice una evaluación

clínica completa.

Existen distintos tipos de ondas dentro de una señal ECG, que representan el comportamiento del corazón en diferentes perspectivas. La corriente eléctrica en las cavidades superiores del corazón está representada por la onda P (auricular). El impulso eléctrico de esta onda comienza desde el nódulo sinusoidal (SA), y se propaga por toda la aurícula. Por otra parte, el complejo QRS representa a la corriente eléctrica en las cavidades inferiores del corazón (ventricular), y consiste en la despolarización eléctrica ventricular. Este complejo consta de una curva descendente (onda negativa o Q), una curva ascendente (onda positiva o R) y una pequeña curva descendente (onda S) [66]. Finalmente, la onda T representa el período de descanso a corto plazo del corazón, que indica la repolarización ventricular (retorno a la actividad eléctrica neutra). La señal normal del ECG consta de ondas P, QRS y T. El intervalo QRS es una medida de la duración total de la despolarización del tejido ventricular.

La detección de QRS proporciona la referencia fundamental para casi todos los algoritmos de análisis de ECG automatizados. Antes de realizar la detección de QRS, se requiere la eliminación o supresión del ruido [66].

Una arritmia puede ocurrir en cualquier instante de tiempo de la señal ECG, su morfología ya sea regular o irregular es la que permite identificar el tipo de arritmia. Comúnmente estos datos son expresados en un plano de coordenadas (x, y) ; el eje x corresponde al tiempo transcurrido, mientras que los datos en la coordenada y a los valores de los impulsos eléctricos.

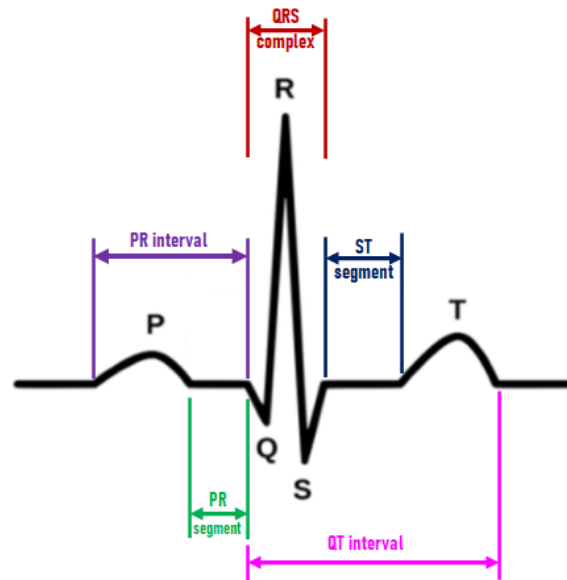


Figura 1.1: Representación sistemática de la onda PQRST

1.6 Diagnóstico de arritmias cardíacas con inteligencia artificial

Las arritmias cardíacas son una causa significativa de morbilidad y mortalidad en todo el mundo. Detectar y diagnosticar estas irregularidades del ritmo cardíaco de manera oportuna es esencial para el manejo clínico y la prevención de complicaciones graves. En este contexto, la inteligencia artificial (IA) ha emergido como una herramienta revolucionaria en la detección de arritmias cardíacas debido a su capacidad para analizar grandes volúmenes de datos de electrocardiogramas de manera precisa y eficiente.

El diagnóstico preciso de las arritmias cardíacas es esencial para su manejo ade-

cuado. Los métodos tradicionales incluyen el ECG, monitoreo Holter y pruebas de estrés. Sin embargo, la creciente disponibilidad de datos de monitoreo cardíaco continuo ha impulsado la aplicación de la inteligencia artificial (IA) en el diagnóstico.

1.6.1 Metodología a través de IA

La capacidad de la IA para analizar grandes conjuntos de datos y detectar patrones ha demostrado ser particularmente útil en el ámbito de la cardiología, especialmente en el diagnóstico de arritmias a partir de electrocardiogramas. La metodología implementada consta de los siguientes pasos:

- **Captura y Procesamiento de Datos:** El primer paso en el proceso es la captura de datos del ECG. Estos datos pueden ser recopilados utilizando monitores Holter, dispositivos portátiles de ECG, o sistemas de telemetría en un entorno hospitalario.
- **Preprocesamiento y Transformación:** Los datos del ECG a menudo requieren limpieza y transformación para eliminar el ruido y destacar las características relevantes del ritmo cardíaco. Los algoritmos de IA utilizan técnicas de procesamiento de señales para aislar y mejorar las ondas P, QRS y T en el ECG. La Figura 1.1 muestra un ejemplo de una señal ECG con segmento PQRST.
- **Extracción de Características:** Una vez que los datos están limpios, la IA identifica y extrae características clave que son relevantes para la detección de arritmias, como la duración de las ondas, la amplitud, la variabilidad del ritmo

cardíaco, entre otros.

- **Entrenamiento del Modelo:** Con un conjunto de datos etiquetado (donde las arritmias ya han sido identificadas por expertos), se entrena un modelo de IA, como una red neuronal, máquina de vectores de soporte o algoritmos de aprendizaje profundo. Estos modelos "aprenden" de los ejemplos proporcionados y ajustan sus parámetros para hacer predicciones precisas en nuevos datos.
- **Evaluación y Validación:** El modelo entrenado se prueba en un conjunto de datos separado que no participó en el entrenamiento. El rendimiento del modelo se mide en términos de precisión, sensibilidad, especificidad, entre otros indicadores.
- **Evaluación y Validación:** El modelo entrenado se prueba en un conjunto de datos separado que no participó en el entrenamiento. El rendimiento del modelo se mide en términos de precisión, sensibilidad, especificidad, entre otros indicadores.
- **Implementación en la Práctica Clínica:** Una vez validados, estos algoritmos pueden integrarse en sistemas de ECG o dispositivos portátiles para ofrecer diagnósticos en tiempo real o alertas sobre posibles arritmias.

1.6.2 Ventajas y Desafíos

Ventajas

La principal ventaja de usar IA para diagnosticar arritmias es la capacidad de procesar y analizar rápidamente grandes conjuntos de datos, proporcionando diagnósticos precisos y rápidos. Sin embargo, los algoritmos de IA son tan buenos como los datos en los que se entrenan, por lo que es esencial que cuenten con datos de alta calidad y diversidad. Además, siempre es crucial que un especialista revise los resultados de la IA para confirmar el diagnóstico.

La IA ofrece varias ventajas cruciales en la detección de arritmias cardíacas:

- **Precisión y Consistencia:** Los modelos de IA pueden analizar ECG de manera sistemática y precisa, incluso en casos difíciles de detectar para los médicos. La IA puede mantener una alta precisión incluso con grandes volúmenes de datos.
- **Diagnóstico Temprano:** La IA puede detectar arritmias cardíacas en etapas tempranas, lo que permite la intervención médica oportuna y la prevención de complicaciones graves.
- **Monitoreo Continuo:** Los sistemas de IA pueden monitorear de forma continua a los pacientes, lo que es especialmente beneficioso para aquellos en riesgo de arritmias recurrentes.
- **Reducción de la Carga de Trabajo:** Al automatizar la detección inicial, la IA puede liberar a los profesionales de la salud para que se centren en tareas más complejas y de toma de decisiones.

Desafíos

La detección de arritmias cardíacas presenta varios desafíos. Los patrones anómalos en los ECG pueden ser sutiles, y la variabilidad entre pacientes complica la identificación manual. Además, la interpretación precisa de ECG a menudo requiere experiencia clínica significativa. Estos desafíos pueden retrasar el diagnóstico y aumentar el riesgo para los pacientes.

1.6.3 Modelos de IA en la Detección de Arritmias

Existen varios tipos de modelos de IA utilizados en la detección de arritmias:

arritmias cardíacas:

- **Redes Neuronales Convolucionales (CNN):** Estos modelos son efectivos en la extracción de características de los ECG y pueden detectar patrones anómalos con alta precisión.
- **Redes Neuronales Recurrentes (RNN):** Las RNN son adecuadas para modelar secuencias de datos de ECG a lo largo del tiempo, lo que es importante para detectar arritmias intermitentes.
- **Redes Neuronales Profundas:** Las redes neuronales profundas, como las redes neuronales recurrentes largas (LSTM) y las redes neuronales convolucionales 1D (1D-CNN), han demostrado un alto rendimiento en la detección de arritmias.
- **Aprendizaje por Transferencia:** La transferencia de conocimiento de modelos

previamente entrenados en grandes conjuntos de datos puede mejorar la precisión de los modelos de IA.

1.6.4 Futuro de la detección de arritmias con IA

El uso de la IA en la detección de arritmias cardíacas es un campo en rápido crecimiento. Se espera que en el futuro, la IA sea aún más eficiente y precisa en la identificación de arritmias, lo que podría revolucionar la atención médica cardiovascular. Además, la integración de la IA en dispositivos médicos portátiles y sistemas de telemedicina podría permitir el monitoreo continuo y la detección temprana de arritmias en entornos no clínicos.

La integración de algoritmos de inteligencia artificial en el diagnóstico de arritmias promete mejorar la eficiencia y precisión en la detección y tratamiento de estas condiciones. Sin embargo, es esencial considerar la IA como una herramienta complementaria, no reemplazante, del juicio clínico experto.

La importancia de la inteligencia artificial en la detección de arritmias cardíacas es innegable. La capacidad de la IA para mejorar la precisión y la eficiencia del diagnóstico puede tener un impacto significativo en la atención médica, permitiendo intervenciones más tempranas y efectivas en pacientes con riesgo de arritmias cardíacas.

En los siguientes capítulos abordaremos los algoritmos de IA que se utilizaron para la generación del modelo propuesto en el presente trabajo de tesis.

Capítulo 2

Redes Neuronales Convolucionales

Las Redes Neuronales Convolucionales (CNN, por sus siglas en inglés, Convolutional Neural Networks) representan uno de los avances más significativos en el ámbito del aprendizaje profundo, particularmente en tareas relacionadas con el procesamiento y reconocimiento de imágenes. Inspiradas en la comprensión biológica del sistema visual humano, las CNNs son especialmente potentes para tareas que involucran patrones visuales debido a su capacidad para extraer automáticamente características relevantes de las imágenes.

Una CNN es un algoritmo que se inspira en la fisiología de los sistemas neuronales biológicos, su arquitectura es análoga al patrón de conectividad de las neuronas en el cerebro humano con el córtex visual [67]. Las neuronas responden de manera individual a los estímulos que se producen en la región del campo visual, conocida como campo receptivo, una colección de tales campos se superpone para cubrir toda el área visual [67]. Toda CNN se distingue de otro tipo de red neuronal, por su

rendimiento superior con entradas de señales que corresponden a: imágenes, voz o audio, y series temporales como lo es un ECG [68, 67].

2.1 Antecedentes Históricos

El concepto detrás de las CNN se basa en la forma en que el área visual del cerebro humano procesa la información. En la década de 1960, los científicos David H. Hubel y Torsten Wiesel realizaron investigaciones pioneras sobre cómo las neuronas en el córtex visual responden a estímulos visuales, lo que llevó al concepto de un "campo receptivo" y a la idea de que ciertas neuronas responden a características específicas en la imagen [69].

La evolución de las CNNs ha sido un proceso gradual que ha tomado inspiración de la biología, la matemática y la informática. Su historia se encuentra entrelazada con el desarrollo general de las redes neuronales y la visión por computadora.

2.1.1 Inspiración Biológica

El comienzo conceptual de las CNNs proviene de la comprensión de cómo funciona el sistema visual en los seres vivos. En la década de 1950 y 1960, los neurofisiólogos David H. Hubel y Torsten Wiesel llevaron a cabo experimentos en el córtex visual de gatos y monos. Descubrieron que ciertas neuronas respondían a estímulos visuales específicos, como líneas o bordes en orientaciones particulares. A estas neuronas se les denominó "neuronas de detección de características". Su trabajo les valió el Premio Nobel en 1981 y estableció la idea de que el procesamiento visual ocurre en etapas

jerárquicas.

2.1.2 Primeros Modelos

El perceptrón, introducido por Frank Rosenblatt en 1957, fue uno de los primeros modelos de red neuronal. Aunque no era una CNN, el perceptrón sentó las bases para la creación de redes neuronales más complejas en el futuro.

En los años 80, Kunihiko Fukushima presentó el "Neocognitron", un modelo jerárquico inspirado en las investigaciones de Hubel y Wiesel. El Neocognitron tenía células "S" para la detección de características y células "C" para el agrupamiento espacial (similar al pooling en CNNs modernas). Aunque este modelo no se entrenaba mediante retropropagación, introdujo el concepto de convolución en redes neuronales [69].

2.1.3 Auge de la Retropropagación

El método de retropropagación, popularizado por Rumelhart, Hinton y Williams en la década de 1980, permitió el entrenamiento eficiente de redes neuronales multicapa. Yann LeCun, en 1989, integró esta técnica en un modelo de red convolucional llamado LeNet-5, diseñado para reconocer dígitos escritos a mano. Este fue uno de los primeros ejemplos exitosos de una CNN moderna.

2.1.4 Avances en la Década de 2000

Con el aumento de la potencia computacional y la disponibilidad de grandes conjuntos de datos, las CNNs comenzaron a ganar popularidad en la década de 2000. Sin embargo, fue en 2012 cuando las CNNs capturaron realmente la atención del mundo de la investigación [69]. En ese año, Alex Krizhevsky, Ilya Sutskever y Geoffrey Hinton presentaron un modelo llamado "AlexNet" en el desafío de ImageNet, estableciendo un nuevo estado del arte en el reconocimiento de imágenes y superando a otros métodos tradicionales por un margen significativo.

Este éxito impulsó una oleada de interés en las CNNs, lo que llevó a la creación de arquitecturas más profundas y sofisticadas, como VGG, GoogLeNet, y ResNet.

2.1.5 Integración y Adopción

Desde entonces, las CNNs han sido adoptadas y adaptadas en una amplia gama de aplicaciones, desde sistemas de recomendación hasta vehículos autónomos. Han demostrado ser excepcionalmente efectivas en una variedad de tareas y han desplazado a muchos métodos tradicionales en visión por computadora.

En resumen, las redes neuronales convolucionales son el resultado de décadas de investigación en múltiples campos, desde la biología hasta la informática. Han evolucionado de modelos conceptuales basados en el sistema visual a potentes herramientas computacionales que están redefiniendo lo que es posible en la visión por computadora y el aprendizaje automático.

2.2 Arquitectura y Funcionamiento

Las CNNs son una categoría especializada de redes neuronales diseñadas para procesar datos con una estructura similar a una cuadrícula, como una imagen. Las imágenes son intrínsecamente cuadrículas de píxeles, y las CNNs son especialmente eficientes para encontrar patrones en ellas.

Una CNN es capaz de aprender paso a paso la jerarquía espacial de los datos memorizando patrones de alto y bajo nivel. Por lo general, su estructura matemática está conformada por una arquitectura que consta de tres tipos de capas alineadas: capas convolucionales (*Conv*), de agrupación (*Pooling*) y totalmente conectadas (*FC*). Tanto las capas *Conv* como las *Pooling*, son las encargadas de realizar la extracción de características (cómo colores y bordes) y también de reducir la dimensionalidad. Si bien, el orden de aparición entre capas *Conv* y *Pooling* no tiene relevancia por su naturaleza, es decir, puede existir una combinación en la aparición de éstas, si es indispensable que la primer capa en la arquitectura sea de tipo *Conv* y la última sea de tipo *Pooling*, en cambio las *FC* se deben de ubicar al final de la arquitectura de capas de toda CNN [70]. Sin embargo, antes de las *FC* debe existir una capa de aplanado (*Flatten*), misma que se encarga de ajustar las salidas de la última capa de tipo *Pooling*, a un formato que las *FC* logren comprender para realizar su tarea correctamente. Las *FC* son capas que se encargan de mapear las características extraídas con el objetivo de encontrar la predicción correcta según el número de clases establecidas [68]. En la Figura 2.1, se puede observar la arquitectura general de capas para la CNN, cuya entrada es bidimensional (2-D), en dónde se toma del

Dataset una entrada que es procesada por cada una de las capas propias de la CNN, cada tipo de capa se encuentra definido por sus hiperparámetros, su naturaleza de comportamiento, función de activación y la salida que produce [68, 67].

Como se muestra en la Figura 2.1, la salida de cada capa es entrada de la siguiente, y su complejidad aumenta al incrementar el número total de capas definido por **NC**.

Este número actualmente es definido de forma manual, regularmente su valor está relacionado con el número de clases que se busca predecir y a la naturaleza del Dataset, es decir, si se trata de imágenes, audio o series temporales de una dimensión (1-D). El orden de aparición según el tipo de capa es el mismo que se describió con anterioridad, comenzando con una *Conv* y colocando una *Pooling* antes de la capa *Flatten*. Dentro de las capas *Pooling* existen dos subtipos *Average* o *Max*, mismas que se describen más adelante. La capa *Flatten* debe colocarse como intermedia entre el conjunto de capas *Pooling* y las *FC*, esta capa tiene la función de convertir los datos a un formato vectorial, de tal forma que las *FC* puedan procesar para llevar a cabo su tarea de predicción.

Los hiperparámetros que definen a las capas *Conv* son: padding, kernel y stride, mientras que para las *Pooling* sólo son kernel y stride. Finalmente, para las capas *FC* son el dropout y número de neuronas (*units*), estos hiperparámetros se explican en los siguientes párrafos. Las funciones de activación que suelen utilizarse son: unidad lineal rectificadora (ReLU) y la multinomial logistic regression (Softmax), la primera para las capas *Conv*, y la última para las capas *FC*.

La arquitectura que se observa en la Figura 2.1, es de índole general y el número total de capas al igual que los valores de sus hiperparámetros depende de la comple-

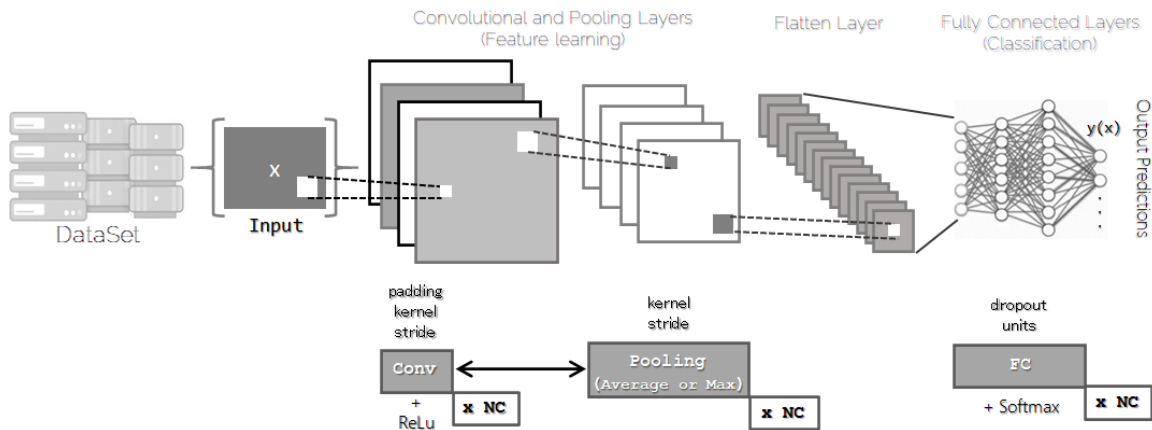


Figura 2.1: Arquitectura general de una CNN.

En la tarea de clasificación que se requiera llevar a cabo. La selección de estos hiperparámetros es un proceso manual que se ha realizado en los últimos años. En el estado del arte se describieron algunos de los trabajos de investigación en donde presentan arquitecturas finales definidas de tal forma. Estos ajustes realizados de forma artesanal permiten evaluar el performance de la CNN, y con ello decidir cuál es la mejor arquitectura para determinada tarea de clasificación. El error de una CNN se determina mediante la diferencia entre el valor estimado y el valor real, y entre los más comunes para tareas de clasificación se encuentran: error cuadrático medio, entropía cruzada binaria y entropía cruzada categórica o CE. Este error es calculado por la CNN, y depende del algoritmo de optimización en los pesos de la red neuronal para tal fin, entre los algoritmos optimizadores de pesos para una CNN se encuentran: Descenso Estocástico del Gradiente, Adam, Adagrad y Adedelta [69, 68]. En este trabajo de investigación, el error calculado mediante la optimización de los pesos, no es más que la función costo para el PSO.

Cuando se introducen los datos a una CNN, es común aplicar un preprocesamiento con el objetivo de que el comportamiento de estos tenga una distribución normal, es decir, con una media cero y una varianza unitaria, evitando así la saturación temprana de las funciones de activación. Este problema suele aparecer en las capas intermedias (*Conv* or *Pooling*), debido al comportamiento del proceso de entrenamiento, lo que vuelve lento el aprendizaje a la adaptación de nuevas distribuciones. La forma de evitar este problema consiste en aplicar la normalización por lotes [67]. Otro mecanismo para evitar el sobre ajuste, consiste en aplicar una técnica de regularización conocida como Dropout en las *FC*, la cual evita la co-adaptación de neuronas a patrones repetitivos, garantizando así una mejor generalización del modelo. En el Dropout se activan y se desactivan neuronas en las capas ocultas, a través de una malla de máscaras, aquellas neuronas inactivas no contribuyen en ninguna operación para la salida [69].

Cada tipo de capa presente en la CNN, tiene una tarea en particular, que en conjunto hacen posible el aprendizaje y clasificación de clases, la funcionalidad de cada una de ellas se describe a continuación.

2.2.1 Capa Convolutiva

La capa convolutiva representa el bloque de construcción principal de una CNN. Dentro de una *Conv* existen un conjunto de filtros (o núcleos), cuyos parámetros deben aprenderse a lo largo del entrenamiento. Regularmente, el tamaño de estos filtros suele ser de una dimensión menor al Input del Dataset. La tarea de cada filtro,

consiste en convolucionar con el Input y así crear un mapa de características. El proceso de convolución consiste en mover los filtros a través de los campos receptivos de la Input, con el objetivo de verificar si la característica está presente.

El detector de características comúnmente puede estar representado por una matriz 2-D de pesos, al igual que un filtro, lo que permite determinar el tamaño del campo receptivo. Este filtro se aplica a un área de la Input y se procede a calcular un producto punto entre los datos del Input y los datos del filtro. El producto escalar resultante se introduce a una matriz de salida, logrando que el filtro cambie, y se repite el proceso hasta que el núcleo del kernel recorre todos los datos de la entrada. El resultado final de la serie de productos escalares entre la entrada y el filtro, se conoce como mapa de características.

Los pesos en el detector de características permanecen fijos a medida que se mueve por la entrada, lo que también se conoce como uso compartido de parámetros. Algunos parámetros, como los valores de peso, se ajustan durante el entrenamiento a través del proceso de backpropagation y gradiente descendente, este procedimiento es realizado en las capas *FC*. Estos pesos, son actualizados mediante los algoritmos optimizadores que fueron mencionados con anterioridad. Sin embargo, hay tres hiperparámetros que afectan el tamaño del volumen de la salida que deben configurarse antes de que comience el entrenamiento de la red neuronal: kernel de los filtros, stride y padding.

Después de cada operación de convolución, una CNN aplica una ReLU al mapa de características, introduciendo así la no linealidad en el modelo. Si la capa siguiente se trata de una *Conv*, la estructura de la CNN se vuelve jerárquica.

2.2.2 Capa Pooling

La agrupación de capas es la encargada de reducir la dimensionalidad de la entrada, lo que tiene efecto en la disminución de parámetros durante el entrenamiento. El proceso de agrupación es similar al de convolución, barriendo con un filtro toda la entrada, pero la diferencia es que este filtro no tiene ningún peso. En cambio, el kernel aplica una función de agregación a los valores dentro del campo receptivo, poblando la matriz de salida. Existen dos subtipos principales de agrupación: *Avg-Pooling* y *Max-Pooling*. Las *Avg-Pooling* calculan el promedio de los elementos presentes en la región del mapa de características cubierto por el filtro, mientras que *Max-Pooling* selecciona el elemento máximo de la región del mapa de características cubierto por el filtro.

Las capas de agrupación ayudan a reducir la complejidad, mejorar la eficiencia y limitar el riesgo de sobre adaptación.

2.2.3 Capa de Aplanado

La capa *Flatten* se utiliza para redes neuronales convolucionales cuya Input sean de naturaleza 2-D, su función consiste en convertir todas las matrices bidimensionales resultantes de los mapas de características agrupados, en un solo vector lineal largo y continuo. La matriz aplanada se alimenta como entrada a la capa *FC* para clasificar los datos de entrada.

2.2.4 Capa Totalmente Conectada

Esta capa realiza la tarea de clasificación en base a las características extraídas, a través de las capas anteriores y sus diferentes filtros. Mientras que, las capas *Conv* y *Pooling* tienden a usar funciones ReLu, las capas *FC* generalmente aprovechan una función de activación Softmax para clasificar las entradas de manera adecuada, produciendo una probabilidad de 0 a 1. Las *FC* usan del mismo principio de funcionamiento que un MLP, y para evitar el problema del sobre ajuste, es en estas capas donde se configura una tasa de abandono o Dropout. La última capa de toda CNN siempre será de tipo *FC* y su número de neuronas dependerá de la cantidad de clases que se desee predecir.

Las CNNs están diseñadas para detectar automáticamente y de forma jerárquica las características en imágenes, lo que elimina la necesidad de extracción manual de características. Su estructura se basa en la repetición de capas convolucionales y de pooling, seguidas de capas totalmente conectadas que llevan a una decisión final. Su eficacia en tareas de visión por computadora las ha consolidado como una herramienta esencial en el campo del aprendizaje profundo.

2.3 Ventajas y Desafíos

2.3.1 Ventajas

Las CNNs han transformado la visión por computadora y otros campos relacionados con el procesamiento de información estructurada en cuadrículas. A continuación, se

detallan sus principales ventajas:

- **Aprendizaje Automático de Características:** Tradicionalmente, la extracción de características de imágenes requería un conocimiento experto y una ingeniería manual intensiva. Las CNNs, en cambio, aprenden automáticamente las características relevantes de los datos de entrenamiento, eliminando la necesidad de definir y ajustar manualmente estas características.
- **Jerarquía de Características:** Las CNNs tienen la capacidad de aprender características a diferentes niveles de abstracción. Las primeras capas pueden detectar bordes y texturas, mientras que las capas más profundas pueden reconocer patrones más complejos y abstractos, como formas y objetos.
- **Robustez frente a Variaciones:** Gracias a las operaciones de convolución y pooling, las CNNs son robustas frente a variaciones menores en la posición y orientación de las características en la imagen. Esto significa que pueden reconocer objetos y patrones aunque estén ligeramente rotados, escalados o desplazados.
- **Eficiencia en Parámetros:** Aunque las CNNs pueden tener millones de parámetros, la compartición de pesos en la convolución reduce significativamente el número de parámetros en comparación con las redes neuronales totalmente conectadas. Esto las hace más eficientes en términos de memoria y menos propensas al sobreajuste.
- **Adaptabilidad:** Las CNNs se pueden adaptar y transferir a diferentes tareas. Por ejemplo, los modelos entrenados en un gran conjunto de datos, como Im-

ageNet, pueden ser adaptados a tareas más específicas mediante técnicas de transferencia de aprendizaje, aprovechando las características aprendidas previamente.

- **Procesamiento Invariante a la Posición:** La estructura de las CNNs permite que un patrón o característica pueda ser reconocido independientemente de su posición en la imagen, gracias a la naturaleza de las operaciones de convolución y pooling.
- **Capacidad para Manejar Imágenes de Gran Tamaño:** Las operaciones de pooling reducen progresivamente el tamaño espacial de los mapas de características, permitiendo a las CNNs manejar imágenes de entrada de gran tamaño sin un aumento exponencial en la complejidad computacional.
- **Integración con Otras Arquitecturas:** Las CNNs se pueden combinar fácilmente con otros tipos de redes neuronales, como las redes neuronales recurrentes (RNNs), para abordar tareas que involucran tanto imágenes como secuencias, como la descripción automática de imágenes.
- **Desempeño de Vanguardia:** En muchas tareas de visión por computadora, desde la clasificación de imágenes hasta la detección de objetos y la segmentación semántica, las CNNs han establecido nuevos estándares de desempeño, superando ampliamente los métodos tradicionales.
- **Expansión a Otros Dominios:** Aunque inicialmente diseñadas para imágenes, las CNNs se han adaptado para trabajar con otros tipos de datos estructurados

en cuadrículas, como espectrogramas en el procesamiento de voz o datos de series temporales en ciertos contextos.

Las CNNs representan una de las arquitecturas más influyentes en el aprendizaje profundo, y su versatilidad y eficacia en una amplia gama de tareas las convierten en una herramienta esencial en la inteligencia artificial moderna.

2.3.2 Desafíos

A pesar de sus notables ventajas, las CNNs también tienen desafíos y limitaciones. A continuación, se detallan sus principales desventajas:

- **Necesidad de Grandes Cantidad de Datos:** Las CNNs, especialmente las arquitecturas más profundas, requieren grandes conjuntos de datos etiquetados para entrenarse adecuadamente y evitar el sobreajuste. Obtener y etiquetar estos datos puede ser costoso y laborioso.
- **Complejidad Computacional:** Aunque las CNNs son más eficientes en términos de parámetros que las redes completamente conectadas, siguen siendo computacionalmente intensivas, especialmente durante el entrenamiento. Esto puede requerir hardware especializado, como GPUs o TPUs.
- **Interpretabilidad:** Las CNNs, como muchas otras redes neuronales profundas, son a menudo consideradas como "cajas negras". Aunque pueden lograr un alto rendimiento en tareas específicas, entender exactamente qué características han aprendido y cómo toman decisiones puede ser complicado.

- **Sensibilidad a Perturbaciones Adversas:** Se ha demostrado que las CNNs pueden ser susceptibles a ataques adversos, donde pequeñas perturbaciones indetectables para el ojo humano en una imagen pueden llevar a predicciones erróneas.

Si bien las CNNs han demostrado ser herramientas poderosas en el campo del aprendizaje profundo y la visión por computadora, es esencial ser consciente de sus limitaciones y desafíos. Al igual que con cualquier herramienta o método, es crucial comprender sus desventajas para usarlas de manera efectiva y ética.

2.4 Las CNNs en el Diagnóstico de Arritmias

El diagnóstico preciso y temprano de arritmias cardíacas es crucial para la prevención de morbilidad y mortalidad relacionadas con enfermedades cardiovasculares. Aunque las arritmias se diagnostican tradicionalmente utilizando el ECG, la interpretación manual de estas señales puede ser compleja y sujeta a errores. Aquí es donde las CNNs juegan un papel transformador.

- **Automatización y Precisión:** Las CNNs pueden ser entrenadas para reconocer patrones en los trazados del ECG, identificando características específicas de las arritmias. Una vez entrenadas, pueden analizar automáticamente grandes cantidades de datos y detectar arritmias con una precisión que a menudo supera la de los especialistas humanos, especialmente en casos sutiles o atípicos.
- **Análisis en Tiempo Real:** Dado que las CNNs pueden procesar datos rápidamente, ofrecen la posibilidad de diagnósticos en tiempo real. Esto es particu-

larmente valioso en situaciones de emergencia donde el tiempo es esencial para el tratamiento.

- **Reducción de Falsos Positivos y Negativos:** Al mejorar la precisión del diagnóstico, las CNNs pueden reducir el número de falsos positivos y negativos. Esto no sólo mejora la atención al paciente sino que también puede reducir costos al evitar tratamientos innecesarios o pruebas adicionales.
- **Escalabilidad:** Una vez entrenadas, las CNNs pueden analizar rápidamente grandes conjuntos de datos. Esto es esencial en entornos hospitalarios donde se generan constantemente grandes volúmenes de datos de ECG.
- **Adaptabilidad a Nuevos Patrones:** A medida que se recopilan más datos, las CNNs pueden ser reentrenadas o afinadas para reconocer nuevos patrones o variantes de arritmias, lo que las hace adaptables a la evolución del conocimiento médico.
- **Integración con Sistemas de Monitoreo:** Las CNNs se pueden integrar con dispositivos de monitoreo cardíaco, como los wearables, permitiendo la detección continua y en tiempo real de arritmias en pacientes ambulatorios. Esto es particularmente útil para detectar arritmias que pueden no manifestarse durante un ECG estándar en un hospital.
- **Facilitación del Aprendizaje Médico:** Las CNNs no sólo pueden ser herramientas de diagnóstico sino también herramientas educativas. Al proporcionar análisis

detallados y etiquetados de trazados de ECG, pueden ayudar a médicos y estudiantes a aprender y reconocer patrones asociados con diferentes arritmias.

El uso de las CNNs en el diagnóstico de arritmias cardíacas representa un avance significativo en la cardiología y la medicina en general. Al combinar la capacidad de las CNNs para detectar patrones complejos con la riqueza de datos presentes en los ECG, se abre una vía prometedora para mejorar la precisión, la eficiencia y la atención al paciente en el campo de las enfermedades cardiovasculares. Sin embargo, es esencial abordar con precaución, validando siempre las herramientas basadas en IA con estudios clínicos rigurosos y asegurando su integración ética y efectiva en la práctica clínica.

2.5 Programación de la CNN en Python

El uso de CNNs con Python para diagnosticar arritmias cardíacas generalmente implica procesar señales de ECGs. A continuación, se presenta un esquema básico de cómo esto podría ser abordado utilizando TensorFlow y Keras, dos bibliotecas populares de aprendizaje profundo en Python.

1. **Preparación de Datos:** Primero, necesitarás un conjunto de datos de trazados ECG etiquetados con diferentes tipos de arritmias.
2. **Construcción del Modelo CNN:** Diseñar una arquitectura CNN. Los trazados ECG son series temporales 1D, por lo que se utilizarán capas Conv1D.
3. **Entrenamiento del Modelo:** Entrena el modelo con los datos de entrenamiento.

4. **Evaluación:** Una vez entrenado, puedes evaluar el rendimiento del modelo en el conjunto de datos de prueba.
5. **Predicciones:** Realiza predicciones en nuevos trazados ECG.

La implementación en Python de cada uno de los pasos descritos con anterioridad se pueden apreciar en el Anexo 1.

Este es un esquema simplificado, asumiendo que los datos ECG ya fueron adquiridos, de lo contrario se debe considerar los siguientes aspectos:

- Preprocesamiento de las señales ECG: filtrado, normalización, segmentación, etc.
- Mejoras en la arquitectura del modelo: regularización, ajuste de hiperparámetros, capas adicionales, etc.
- Uso de métricas adicionales como F1-score, matriz de confusión, entre otras, para evaluar el desempeño en un contexto clínico.
- Validación cruzada o uso de un conjunto de validación aparte para afinar el modelo.

Tener en cuenta los desafíos éticos y clínicos asociados con el uso de modelos de IA en el diagnóstico médico. Es esencial que cualquier modelo destinado a la aplicación clínica sea rigurosamente validado en múltiples conjuntos de datos y en entornos clínicos antes de su uso real.

Capítulo 3

Algoritmo de Enjambre de Partículas

El algoritmo de enjambre de partículas (Particle Swarm Optimization, PSO por sus siglas en inglés) es una técnica de optimización metaheurística inspirada en el comportamiento colectivo de organismos, como bancos de peces o bandadas de aves. Aunque a primera vista puede parecer extraño que un algoritmo informático se inspire en tales fenómenos, la naturaleza ha demostrado ser una fuente inagotable de inspiración para la creación de algoritmos eficientes que resuelven problemas complejos[71]. Así como las aves buscan el lugar óptimo para posarse o los peces intentan moverse al mejor sitio dentro de un banco, el PSO busca el punto óptimo en un espacio de soluciones.

Su propósito principal es encontrar una solución óptima en un espacio de búsqueda multidimensional, y ha demostrado ser efectivo en una variedad de aplicaciones, desde la optimización de funciones hasta la solución de problemas complejos en ingeniería, ciencias de la computación y finanzas [72].

La esencia del PSO reside en la idea de que múltiples soluciones candidatas (lla-

madras partículas) se desplazan en el espacio de búsqueda teniendo en cuenta tanto su memoria individual (dirección, velocidad, aceleración) como el conocimiento compartido con sus vecinos. Estas partículas ajustan su posición en función de su experiencia previa y la de sus vecinos, lo que les permite converger hacia las zonas óptimas del espacio de búsqueda [71].

Cada partícula en el enjambre tiene una posición que representa una solución candidata para el problema en cuestión. Además, cada partícula tiene una velocidad que determina la dirección y el tamaño del paso que tomará en la siguiente iteración.

Durante el proceso de optimización, las partículas comparten información sobre sus posiciones actuales y las mejores posiciones que han encontrado, y se actualizan en función de esta información.

La eficiencia del PSO radica en su simplicidad y adaptabilidad. A diferencia de otros métodos de optimización que requieren derivadas o supuestos específicos sobre el problema, el PSO es un método basado en poblaciones que no hace suposiciones específicas y puede aplicarse a una amplia variedad de problemas, desde la optimización de funciones continuas hasta problemas combinatorios discretos[73].

Su funcionamiento está inspirado en el comportamiento que tienen las bandadas de pájaros o bancos de peces en los que, el movimiento de cada individuo , es el resultado de combinar las decisiones individuales de cada uno con el comportamiento del resto [72].

3.1 Antecedentes Históricos del PSO

El PSO tiene raíces que se remontan a la observación y modelado del comportamiento social en grupos de animales, como aves y peces [71]. La idea central es que el comportamiento colectivo de estos organismos puede ser emulado para resolver problemas de optimización. A continuación, se presenta una breve cronología de los eventos y desarrollos clave que llevaron a la creación y popularización del PSO.

3.1.1 Década de 1950 y 1960

Los primeros estudios en el modelado de comportamientos de enjambre y bandada comenzaron a surgir. Estos estudios se centraron en entender cómo los individuos en un grupo interactúan entre sí y con su entorno.

3.1.2 1995 - Origen del PSO

El PSO fue introducido por James Kennedy (psicólogo social) y Russell Eberhart (ingeniero eléctrico) en 1995[73]. Originalmente, su interés no estaba en la optimización sino en el modelado del comportamiento social. El PSO surgió como un algoritmo basado en la simulación de comportamientos sociales simplificados. Kennedy y Eberhart presentaron el PSO en el Congreso Internacional de Redes Neuronales (ICNN'95). Lo propusieron como una técnica para optimizar redes neuronales, pero pronto se dieron cuenta de que tenía aplicaciones mucho más amplias.

3.1.3 Finales de la década de 1990

El PSO comenzó a ganar reconocimiento y aceptación en la comunidad científica. Investigadores de diversas disciplinas comenzaron a explorar y adaptar el PSO para una variedad de problemas de optimización. Durante este período, se propusieron muchas variantes del PSO original para mejorar su eficiencia y adaptabilidad.

3.1.4 Década de 2000

La popularidad del PSO continuó creciendo, llevando a la creación de conferencias y simposios específicos sobre el tema. Se realizaron investigaciones adicionales para entender teóricamente el comportamiento del PSO, incluyendo la convergencia y la estabilidad del algoritmo. Emergieron aplicaciones en áreas tan diversas como la robótica, las finanzas, la medicina y la ingeniería.

3.1.5 Década de 2010 en adelante

El PSO se ha consolidado como una de las principales técnicas de optimización meta-heurística. Se han propuesto numerosas extensiones y variantes del PSO para abordar problemas específicos, como la optimización multiobjetivo, problemas con restricciones, y problemas en espacios discretos, entre otros.

El PSO, desde su concepción en 1995, ha crecido y evolucionado significativamente, transformándose de un modelo de comportamiento social a una herramienta poderosa y versátil de optimización que se utiliza en una amplia variedad de aplicaciones prácticas.

3.2 Fundamentos del PSO

El algoritmo PSO se basa en la simulación del comportamiento social de las aves y peces cuando buscan alimentos o evaden depredadores. Por lo tanto, existen una serie de fundamentos clave que definen el comportamiento y las características del PSO, los cuales son:

3.2.1 Partículas

Representación:

En el contexto de PSO, una "partícula" es una solución potencial al problema de optimización. En términos simples, cada partícula es un punto en el espacio de solución definido por un conjunto de parámetros o coordenadas.

Atributos:

Cada partícula tiene asociados varios atributos clave:

- Posición: p_i Representa la solución actual de la partícula en el espacio de búsqueda.
- Velocidad: v_i Define la dirección y magnitud con la que una partícula cambiará su posición.
- Mejor posición personal: p_{best} Es la mejor solución (posición) que la partícula ha encontrado desde el inicio de la búsqueda.

3.2.2 Movimiento y Actualización

Las partículas se mueven a través del espacio de solución basadas en su experiencia previa y la experiencia de sus vecinas. La velocidad y posición de cada partícula se actualizan en cada iteración basadas en dos componentes principales:

- Componente cognitivo: Representa el conocimiento de la propia partícula y se basa en su mejor posición personal encontrada hasta el momento.
- Componente social: Representa el conocimiento del enjambre y se basa en la mejor posición encontrada por todas las partículas del enjambre (denominado g_{best}).

3.2.3 Ecuación de Actualización

Un algoritmo PSO trabaja con una población (llamada enjambre) de soluciones candidatas (llamadas partículas). Dichas partículas se desplazan a lo largo del espacio de búsqueda conforme unas reglas matemáticas. El movimiento de cada partícula depende de su mejor posición obtenida, así como de la mejor posición global hallada en todo el espacio de búsqueda. A medida que se descubren nuevas y mejores posiciones, éstas pasan a orientar los movimientos de las partículas. El proceso se repite con el objetivo, no garantizado, de hallar en algún momento una solución lo suficientemente satisfactoria [74, 75]. En la Figura 3.1 se puede apreciar el flujograma general de funcionamiento del PSO.

Las nuevas y mejores posiciones son obtenidas mediante una velocidad que se

calcula tomando en cuenta la mejor posición global y la mejor posición actual de cada partícula. La velocidad de cada partícula del enjambre se actualiza empleando:

$$v_i(t + 1) = w \cdot v_i(t) + c_1 \cdot r_1[p_i^{best} - p_i(t)] + c_2 \cdot r_2[p_i^{gbest} - p_i(t)] \quad (3.1)$$

Dónde:

- $v_i(t + 1)$: velocidad de la partícula i en el momento $t + 1$.
- $v_i(t)$: velocidad de la partícula i en el momento t .
- w : coeficiente de inercia, reduce o aumenta a la velocidad de la partícula.
- c_1, c_2 : coeficiente cognitivo y social.
- r_1, r_2 : vector de valores aleatorios entre 0 y 1 de longitud igual a la del vector velocidad.
- p_i^{best} : mejor posición en la que ha estado la partícula i hasta el momento.
- $p_i(t)$: posición de la partícula i en el momento t .
- p_i^{gbest} : es la mejor posición global de todo el enjambre.

Una vez calculada la nueva velocidad, se puede actualizar la posición de la partícula a través de:

$$p_i(t + 1) = p_i(t) + v_i(t + 1) \quad (3.2)$$

Dónde $p_i(t + 1)$ es el vector que contiene las nuevas posiciones obtenidas en la iteración $t + 1$.

$p_i(t)$ corresponde a las posiciones previas de las partículas que se calcularon en la iteración t . Finalmente, $v_i(t + 1)$ es el vector de velocidad que se obtuvo usando la Ecuación (3.1).

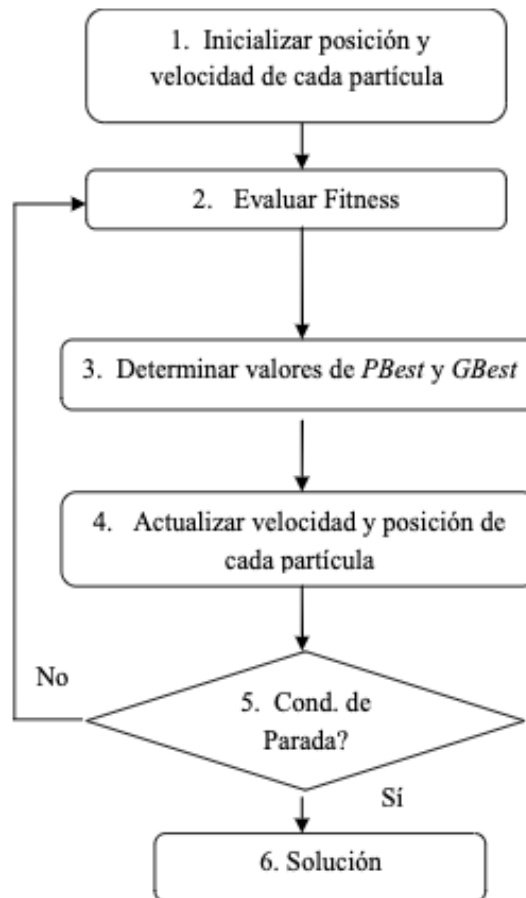


Figura 3.1: Flujograma General del PSO.

En cada iteración, los elementos de la población comparten información, la cual

les permite acercarse con mayor o menor rapidez a la solución global del problema. Para determinar la calidad de población se requiere de una función objetivo, donde se evalúan los individuos cada vez que toman una nueva posición, esta función objetivo viene definida por el problema que se desee optimizar, en nuestro caso se define mediante el valor correspondiente a CE que la CNN obtiene a partir de su optimizador en cada entrenamiento, este proceso se explica con mayor detalle en el siguiente capítulo.

3.2.4 Criterio de Parada

El criterio de parada el algoritmo PSO determina cuándo el algoritmo debe detenerse, es decir, cuándo debe dejar de buscar una mejor solución. Estos criterios son esenciales para garantizar que el algoritmo no se ejecute indefinidamente y para definir un punto en el que se considera que el algoritmo ha convergido o ha hecho suficiente esfuerzo en la búsqueda. Existe una amplia gama de criterios a considerar entre ellos se encuentran:

Número máximo de iteraciones

Este es uno de los criterios de parada más simples y comunes. Se define un número máximo de iteraciones (o generaciones) que el algoritmo debe ejecutar. Una vez que se alcanza este número, el algoritmo se detiene. Este criterio garantiza que el PSO se ejecutará durante un tiempo limitado.

Estabilidad de la solución

Si la mejor solución (o el valor de la función objetivo de la mejor solución) no ha cambiado o ha cambiado muy poco durante un número predefinido de iteraciones consecutivas, el algoritmo se detiene. Este criterio se basa en la idea de que si el enjambre no ha encontrado una mejor solución durante un cierto período, es probable que haya convergido.

Umbral de tolerancia

Se detiene el algoritmo si la diferencia entre la mejor solución encontrada y un valor objetivo predefinido es menor que un cierto umbral. Este criterio es útil cuando se tiene una idea de qué tan buena debe ser la solución.

Estabilidad de la diversidad del enjambre

Si la diversidad de las partículas en el enjambre (es decir, cuánto difieren las soluciones entre sí) cae por debajo de un cierto umbral, el algoritmo puede detenerse, ya que esto podría indicar que todas las partículas se han agrupado en una región del espacio de búsqueda y la exploración se ha vuelto estancada.

Tiempo de ejecución

En algunos casos, especialmente en aplicaciones en tiempo real, el algoritmo puede tener un límite de tiempo. Si el algoritmo no encuentra una solución aceptable dentro de este límite, se detiene.

Condiciones específicas del problema

Dependiendo del problema específico que se esté abordando, pueden existir otros criterios de parada. Por ejemplo, en problemas donde se busca una solución que satisfaga ciertas restricciones, el algoritmo puede detenerse tan pronto como encuentre una solución que cumpla con esas restricciones.

Combinación de criterios

Es común combinar varios de los criterios anteriores para definir cuándo detener el algoritmo. Por ejemplo, se podría utilizar un número máximo de iteraciones combinado con la estabilidad de la solución para asegurar que el algoritmo no se ejecute indefinidamente, pero también que tenga la oportunidad de refinar la solución si hay potencial de mejora.

En el presente trabajo de tesis se hace uso del umbral de tolerancia que se encuentra establecido por el valor de la función objetivo definido por el valor del error de la CNN, dicho valor se minimiza a través del ajuste de la población de partículas por parte del PSO, esto se explicará con mayor detalle en el siguiente capítulo.

3.3 Ventajas y Desafíos del PSO

El algoritmo PSO es una técnica de optimización heurística popular que ha sido aplicada a una amplia variedad de problemas. Como cualquier algoritmo, tiene sus ventajas y desventajas.

3.3.1 Ventajas

1. **Simplicidad:** En comparación con otros algoritmos de optimización, PSO tiene una estructura simple con reglas claras, lo que facilita su implementación y comprensión.
2. **Flexibilidad:** PSO no hace suposiciones estrictas sobre el problema de optimización, lo que lo hace adecuado para una amplia gama de aplicaciones.
3. **No necesita gradientes:** A diferencia de los métodos de optimización basados en gradientes, PSO no requiere que la función objetivo sea diferenciable. Esto es útil para problemas con espacios de búsqueda discontinuos o ruidosos.
4. **Paralelismo:** Las partículas pueden evaluarse simultáneamente, lo que hace que el PSO sea intrínsecamente paralelo y pueda beneficiarse de hardware paralelo.
5. **Adaptabilidad:** Los parámetros del PSO, como el factor de inercia, se pueden ajustar para adaptarse a diferentes problemas, permitiendo un equilibrio entre la exploración del espacio de búsqueda y la explotación de soluciones prometedoras.
6. **Robustez:** PSO ha demostrado ser eficaz en encontrar soluciones aceptables en una variedad de problemas, incluso en presencia de ruido y perturbaciones.

3.3.2 Desafíos

1. **Convergencia prematura:** En ciertas circunstancias, el enjambre puede converger prematuramente a un óptimo local en lugar de encontrar el óptimo global, especialmente si la capacidad de exploración no está bien equilibrada.
2. **Ajuste de parámetros:** Aunque el PSO es adaptable, la necesidad de ajustar parámetros (como el factor de inercia, coeficientes cognitivos y sociales) puede ser tediosa y depende del problema específico.
3. **No garantiza la optimalidad:** Al ser un método heurístico, PSO no proporciona garantías de encontrar la solución óptima global, aunque a menudo encuentra soluciones aproximadas de alta calidad.
4. **Puede ser lento en converger:** En problemas con espacios de búsqueda extensos o complejos, el PSO puede requerir muchas iteraciones para converger a una solución aceptable.
5. **Riesgo de oscilación:** Si no se configura adecuadamente, las partículas pueden oscilar alrededor de un punto sin converger realmente.
6. **Dependencia de la inicialización:** La inicialización del enjambre puede influir en la calidad de la solución encontrada. Una mala inicialización puede llevar al algoritmo a áreas menos prometedoras del espacio de búsqueda.

Mientras que el PSO ofrece una serie de ventajas que lo hacen adecuado para una variedad de problemas de optimización, también presenta desafíos que requieren una

consideración cuidadosa y a menudo un ajuste específico del problema. La elección de usar PSO (o cualquier otra técnica) debe basarse en la naturaleza del problema, los recursos disponibles y las prioridades del equipo de investigadores que lo utilicen.

3.4 El PSO en el Diagnóstico de Arritmias

El PSO ha demostrado ser una herramienta eficaz en diversas áreas de la ingeniería y la ciencia, y el diagnóstico de arritmias cardíacas no es una excepción. Algunas de las principales aportaciones del PSO a este contexto son:

3.4.1 Selección y Extracción de Características

El análisis de electrocardiogramas (ECG) genera una gran cantidad de datos. PSO puede ser utilizado para seleccionar las características más relevantes del ECG que son indicativas de arritmias cardíacas, optimizando el proceso de diagnóstico. Además de la selección, PSO también puede ayudar en la extracción de características, donde se derivan nuevos atributos o se combinan atributos existentes para mejorar la discriminación entre ritmos normales y anormales.

3.4.2 Optimización de Parámetros del Modelo

Los modelos de aprendizaje automático y las redes neuronales son comunes en la clasificación de arritmias. PSO puede ser usado para optimizar los parámetros de estos modelos, como tasas de aprendizaje, número de neuronas en capas ocultas o

hiperparámetros en máquinas de soporte vectorial.

3.4.3 Detección de Puntos Específicos en Señales de ECG

Detectar puntos clave en una señal de ECG, como el pico R, es esencial para el análisis posterior. PSO puede ser utilizado para optimizar algoritmos que identifican y localizan estos puntos en el ECG, mejorando la precisión de la detección.

3.4.4 Diseño y Ajuste de Filtros

Antes de analizar una señal de ECG, es esencial filtrarla para eliminar el ruido y otras interferencias. PSO puede ser utilizado para diseñar filtros óptimos o ajustar los parámetros de filtros existentes para esta tarea.

3.4.5 Detección de Arritmias en Datos Multimodales

Además del ECG, otras fuentes de datos (como la variabilidad de la frecuencia cardíaca o las imágenes cardíacas) pueden proporcionar información relevante sobre arritmias. PSO puede ayudar en la integración y fusión de datos de múltiples fuentes para mejorar el diagnóstico.

3.4.6 Optimización de Estrategias de Diagnóstico

En situaciones en las que se deben tomar decisiones secuenciales (por ejemplo, si se debe realizar una prueba adicional después de una inicial), PSO puede ser usado para

encontrar la estrategia óptima que minimice los costos y maximice la precisión del diagnóstico.

Estos usos destacan el papel de PSO como herramienta de optimización en el ámbito del diagnóstico de arritmias cardíacas. Al aprovechar la capacidad del PSO para explorar y explotar eficientemente grandes espacios de búsqueda, es posible mejorar la precisión y eficiencia de los sistemas de diagnóstico cardíaco. Para el caso del presente trabajo de tesis el uso que se le dio al PSO fue para la optimización de hiper parámetros del modelo CNN.

3.5 Programación del PSO en Python

Implementar PSO para el diagnóstico de arritmias cardíacas requiere varios pasos, incluyendo la adquisición de datos, la selección o extracción de características, la definición de la función de fitness, la implementación del algoritmo PSO y, finalmente, el uso de PSO para optimizar algún aspecto del proceso de diagnóstico.

3.5.1 Adquisición de Datos

Electrocardiogramas: Es necesario obtener registros de ECG que contengan ejemplos tanto de ritmos cardíacos normales como de diferentes tipos de arritmias. Estos registros pueden provenir de bases de datos públicas, como la base de datos MIT-BIH, o de conjuntos de datos propios.

3.5.2 Preprocesamiento

Filtrado: Las señales de ECG suelen estar contaminadas con ruido (por ejemplo, ruido muscular o de línea eléctrica). Es esencial filtrar este ruido para obtener una señal clara.

3.5.3 Extracción de Características

Utilizando PSO, se pueden seleccionar y extraer las características más relevantes de las señales de ECG. Estas características podrían incluir intervalos entre picos R, amplitudes, áreas bajo segmentos específicos de la onda, entre otros.

3.5.4 Diseño del modelo

El siguiente paso es decidir qué tipo de modelo se utilizará para la clasificación. Puede ser una máquina de soporte vectorial, una red neuronal, una regresión logística, entre otros.

3.5.5 Optimización con PSO

Aquí es donde PSO entra en juego de manera más prominente:

- **Inicialización:** Se crea un enjambre de partículas, donde cada partícula representa una posible solución al problema (por ejemplo, un conjunto de parámetros para un modelo de clasificación).

- **Evaluación:** Cada partícula se evalúa usando un conjunto de entrenamiento. Esto podría implicar entrenar un modelo con los parámetros de la partícula y luego evaluar su rendimiento en un conjunto de validación.
- **Actualización:** Basándose en el rendimiento de las partículas, se actualizan sus posiciones y velocidades, considerando tanto su rendimiento personal (pbest) como el rendimiento del mejor miembro del enjambre (gbest).
- **Iteración:** Los pasos b y c se repiten hasta que se cumple un criterio de parada, como un número máximo de iteraciones o una mejora mínima en el rendimiento.

3.5.6 Evaluación

Una vez que se ha identificado la mejor solución (o partícula), se evalúa su rendimiento en un conjunto de datos de prueba separado para determinar su precisión, sensibilidad, especificidad, entre otros indicadores de rendimiento.

3.5.7 Diagnóstico

Finalmente, el modelo optimizado se puede utilizar en la práctica clínica o en dispositivos portátiles para diagnosticar arritmias en tiempo real.

3.5.8 Consideraciones adicionales

Es crucial tener un conjunto de datos lo suficientemente grande y diverso para entrenar y validar el modelo adecuadamente.

PSO es solo una herramienta en el proceso. Dependiendo de la naturaleza y calidad de los datos, puede ser necesario realizar más pasos de preprocesamiento o posprocesamiento.

La optimización mediante PSO no garantiza una solución global óptima. Sin embargo, suele encontrar soluciones que son lo suficientemente buenas para muchas aplicaciones prácticas. En el Anexo 2 se puede apreciar el código de programación necesario para lograr una implementación general del PSO en el lenguaje de programación de Python.

Capítulo 4

Algoritmo H-PSO-CNN

El H-PSO-CNN es un modelo híbrido enfocado en aprovechar las cualidades de cada uno de los algoritmos implicados. En el trabajo relacionado se aprecia que la CNN es el algoritmo predilecto para la tarea de clasificación de las CAs, en cada uno de estos trabajos se reportan resultados satisfactorios. Una de las principales razones por las que se elige a la CNN como algoritmo favorito en esta tarea de clasificación, se debe a su habilidad de realizar la extracción de características y clasificación de los ítems de entrada, en un sólo procedimiento. Por otra parte, el PSO es una técnica metaheurística, que al menos en los últimos diez años, ha demostrado ser confiable en la búsqueda de mejores soluciones en una determinada población de partículas, que permitan satisfacer la función objetivo de forma eficaz. La naturaleza de la población, las soluciones encontradas y la función objetivo, dependerán del contexto que se busque resolver con la aplicación del PSO. En cada uno de los casos, el PSO busca de todas las mejores soluciones encontradas, a la mejor global (p_{gbest}), y los

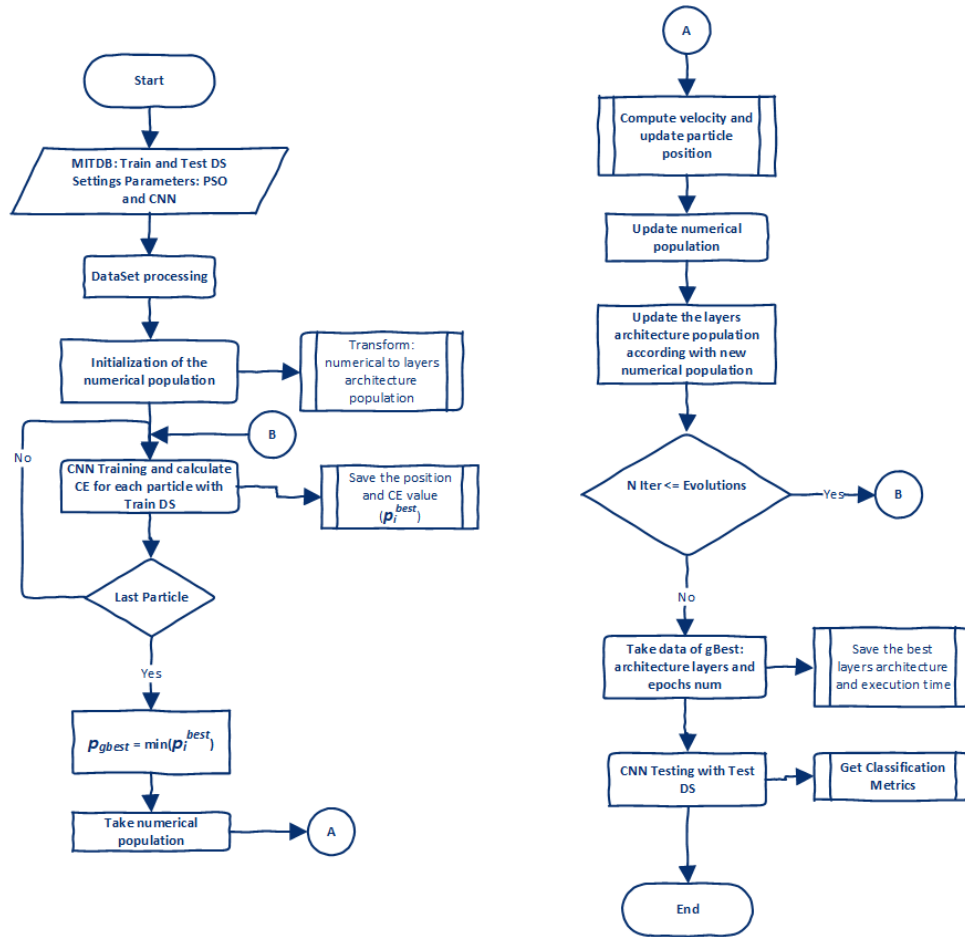


Figura 4.1: Diagrama de Flujo del modelo propuesto H-PSO-CNN.

valores dentro de esta partícula, son los que mejor satisfacen a la función objetivo planteada.

4.1 Esencia del H-PSO-CNN

El H-PSO-CNN utiliza a la CNN como el algoritmo que se encarga de aprender y clasificar las cinco clases de CAs que se le proporcionan, mientras que hace uso del PSO para realizar una búsqueda en un enjambre de partículas, que se encuentran definidas cada una de ellas por una determinada configuración distinta de arquitectura de capas con sus correspondientes hiperparámetros, y un número determinado de épocas. El PSO realiza la búsqueda de la partícula que le permita a la CNN obtener resultados satisfactorios en su tarea de clasificación.

Esta búsqueda global es realizada mediante la minimización de la función objetivo que se define mediante la Ecuación (4.9), es decir, a través de la minimización del CE, el PSO se encarga de buscar la mejor arquitectura de esa población que le permita a la CNN obtener un performance satisfactorio. El valor del CE es calculado por medio del optimizador propio de la red neuronal, y sólo es considerado para clasificaciones que incluyen al menos tres clases distintas, tal es el caso de los datos que se encuentran en el MITDB y son utilizados en nuestro trabajo de investigación.

El H-PSO-CNN comienza con el primer paso que corresponde al algoritmo meta-heurístico, es decir, la inicialización de una población conformada por N partículas, cada una de ellas definida en un espacio con cuatro dimensiones. La primer dimensión corresponde al número de capas *Conv*, la segunda al número de capas *Pooling*, la tercer dimensión al número de capas *FC* y finalmente la cuarta al número de épocas. Para cada tipo de capa se configuran sus correspondientes hiperparámetros, y se realiza un proceso de conversión de la población inicial, de tal forma que la CNN logre

comprender la arquitectura de capas, este procedimiento se explica en las siguientes subsecciones.

Cada dimensión de la población inicial del PSO, almacena un valor numérico entero distinto, los datos que corresponden a las primeras tres dimensiones permiten una diversidad en la profundidad de la arquitectura de capas para que entrene la CNN, mientras que el dato numérico de la cuarta dimensión se encarga de encontrar un número adecuado de épocas para esa configuración de capas de la CNN.

La CNN toma las CAs de entrenamiento provenientes del MITDB (Train DS) y en conjunto con la configuración de cada una de las partículas que conforman a toda la población comienza su entrenamiento, la red neuronal calcula para cada partícula su correspondiente CE, y este valor es almacenado como un mejor local (p_i^{best}), mismo que es utilizado por el PSO en su proceso de optimización. Este proceso se repite hasta concluir con la última partícula de la población inicial.

Al finalizar este ciclo iterativo, el PSO determina al p_{gbest} a través de la partícula i cuyo valor p_i^{best} sea el menor de todos los CE calculados para toda la población inicial. Tomando en cuenta la posición y el valor que representan a p_{gbest} , se hace uso de los operadores del PSO que se definieron previamente en las Ecuaciones (3.1) y (3.2), permitiendo el cálculo de la velocidad y la actualización de la posición de cada una de las partículas que definen a la población inicial del PSO, este proceso permite la actualización de la población. La búsqueda del p_{gbest} se repite hasta que se cumpla con el número total de evoluciones, y en cada iteración el PSO busca minimizar el valor del CE. El p_{gbest} representa a la mejor partícula de la población, es decir, la que obtuvo una solución satisfactoria durante el entrenamiento de la CNN para la

clasificación de las cinco distintas clases de CAs. Una vez, concluidas las evoluciones, se toma la configuración de capas que está representada por el p_{gbest} , para evaluar el rendimiento de la CNN, mediante el uso de CAs de prueba (Test DS) provenientes del MITDB. Lo anterior, permite obtener las métricas relacionadas con la clasificación categórica y con ello conocer el rendimiento de la red neuronal, estos valores son explicados en el capítulo 6.

En la Fig. 4.1 se puede apreciar el diagrama de flujo del modelo H-PSO-CNN, mientras que en el Algoritmo 1 su correspondiente pseudocódigo, ambos describen la metodología del modelo propuesto en este trabajo de investigación.

4.2 Arquitectura del H-PSO-CNN

La mayoría de los algoritmos metaheurísticos, como lo es el PSO, requieren de una población cuya naturaleza corresponda de preferencia a todos los números racionales, es decir, no están preparados para optimizar poblaciones compuestas por valores de tipo alfanuméricos como lo es el caso de una arquitectura de capas de una CNN. Por lo tanto, el H-PSO-CNN ofrece una solución de conversión entre la población numérica (P) para el PSO y la población compuesta por distintas configuraciones de arquitecturas de capas y número de épocas (L), que son entendibles para la tarea de clasificación por parte de la CNN. Este procedimiento define el núcleo del H-PSO-CNN, debido a que permite una funcionalidad correcta de los algoritmos implicados. Con P el PSO es capaz de utilizar sus operadores para llevar a cabo su tarea de optimización, mientras que con L la CNN es capaz de realizar su encomienda en la

clasificación categórica y calcular la función objetivo del PSO. La forma de obtener tanto a P y L , así como su actualización, se explica en las siguientes sub-secciones.

Algorithm 1 Enfoque propuesto H-PSO-CNN

Input: MITDB, PSO CNN Settings parameters

Output: p_{gbest} , *metrics* and *time*

- 1: MITDB Processing to get Train and Test DS
 - 2: Initialize numerical population (P) with Equation (3)
 - 3: Get P and Transform in L with Algorithm 2
 - 4: **while** $iter \leq evolutions$ **do**
 - 5: **for** $i = 1$ to $size(P)$ **do**
 - 6: Use Train DS for Train CNN with $L_{(i)}$ and Calculate $CE_{(i)}$ using Equation (5)
 - 7: Local particle is p_i^{best} with $CE_{(i)}$ value
 - 8: **end for**
 - 9: The global particle p_{gbest} is $min(p^{best})$
 - 10: Compute velocity in P with Equation (3.1)
 - 11: Update position in P with Equation (3.2)
 - 12: Update L with new P using Algorithm 3
 - 13: **end while**
 - 14: Save time execution in *time*
 - 15: With p_{gbest} and using Test DS:
 - 16: Test CNN and Get *metrics*
 - 17: **return** p_{gbest} , *metrics* and *time*
-

4.2.1 Inicialización de población numérica

Algoritmos metaheurísticos como el PSO, requieren de un espacio de búsqueda de posibles soluciones, mismo que se encuentra definido por una determinada población con un número determinado de individuos. Estas posibles soluciones buscan satisfacer una función objetivo, cuya finalidad radica en maximizar o minimizar su valor. En el caso particular del PSO en el H-PSO-CNN, los individuos que conforman a las partículas del metaheurístico, se encuentran definidos por una matriz P , que está conformada por $(N \times M)$ dimensiones:

$$P = \begin{pmatrix} X_{1,1} & X_{1,2} & \cdots & X_{1,M} \\ X_{2,1} & X_{2,2} & \cdots & X_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ X_{N,1} & X_{N,2} & X_{N,3} & X_{N,M} \end{pmatrix} \quad (4.1)$$

Dónde N corresponde al número total de partículas que conforman a P , mientras que M al número de dimensiones o variables independientes que son actualizadas por los operadores del PSO. En conjunto las dimensiones de P , definen el espacio de búsqueda de soluciones, siendo $M = 4$ para este modelo. Las primeras tres dimensiones se encuentran directamente relacionadas con el número de capas de la CNN, es decir, para $M = 1$ el valor numérico representa al número de capas *Conv*, en $M = 2$ se almacena el número de capas *Pooling* y para $M = 3$ el número de capas *FC*. La última dimensión $M = 4$, corresponde a un número de épocas distinto para cada partícula, con un número de capas acorde a los valores generados en $M = 1, 2, 3$. Lo

anterior, hace posible contar con una población heterogénea de partículas que hasta ahora definen el número de capas y épocas con las que la CNN realizará su tarea de clasificación.

Por lo tanto, $X_{N,M}$ representa a los valores que se definen para cada una de las dimensiones que conforman a la población, su contenido se obtiene según corresponda el valor de M . Para $M = 1, 2, 3$, el contenido de cada partícula N depende de la profundidad $depth$ y de coeficientes de peso porcentual. La $depth$ que se calcula mediante:

$$depth = randint(min_L, max_L) \quad (4.2)$$

Dónde $min_L = 2$ para todas las partículas y representa el valor mínimo de capas para cada partícula, para el H-PSO-CNN es crucial contar con al menos dos capas en la arquitectura, la primera que debe ser de tipo *Conv* y la última *FC* con un número de neuronas igual al de las clases que se busca predecir. Por otra parte, max_L es un número entero que depende de la complejidad en la entrada y el número de clasificaciones objetivo.

La Ecuación (4.2) hace uso de una función *randint* que obtiene un valor aleatorio entero entre min_L y max_L , obteniendo así a $depth$, misma que es diferente para cada partícula N .

Los coeficientes de peso porcentual, al igual que max_L dependen de la complejidad de los datos de entrada y el número de clasificaciones que se busque predecir. Para las capas *Conv* su coeficiente es $coef_{Conv}$, para las *Pooling* es $coef_{Pool}$ y para las *FC* es

$coef_{FC}$, la suma de todos estos coeficientes tiene que ser igual a uno, como se muestra a continuación:

$$coef_{Conv} + coef_{Pool} + coef_{FC} = 1 \quad (4.3)$$

Toda vez, que ya se cuenta con $depth$ y cada uno de los valores de los coeficientes, el cálculo para $X_{N,1}$, $X_{N,2}$, $X_{N,3}$ se realiza mediante:

$$X_{N,1} = coef_{Conv} * depth \quad (4.4)$$

$$X_{N,2} = coef_{Pool} * depth \quad (4.5)$$

$$X_{N,3} = coef_{FC} * depth \quad (4.6)$$

Lo anterior, permite que se obtengan valores heterogéneos para cada una de las partículas, es importante recordar que $depth$ es distinta para cada partícula N , mientras que $coef_{Conv}$, $coef_{Pool}$ y $coef_{FC}$ es el mismo toda la población P .

Finalmente, $X_{N,4}$ es calculado mediante una función $randint$ que se define por medio de un min_E y un max_E , tal y como se aprecia en la Ecuación (4.7).

$$X_{N,4} = randint(min_E, max_E) \quad (4.7)$$

Dónde min_E y max_E dependen de la complejidad en los datos de entrada y el

número de clases objetivo a predecir.

El procedimiento explicado con anterioridad, hace posible la existencia de P como una matriz bidimensional de N partículas, conformada por valores numéricos generados de forma aleatoria en cada una de sus dimensiones. Hasta ahora P sólo puede ser funcional para el proceso de optimización por parte del PSO . Sin embargo, es necesario realizar el cálculo de cada uno de los valores que definen a la función objetivo para cada partícula en cuestión y con ello comenzar con el proceso de optimización, es decir, se busca minimizar ese valor.

Por lo anterior, es necesario transformar la población de P a una estructura de datos que la CNN pueda comprender, en dónde para cada partícula se presente una configuración distinta de arquitectura de capas con sus correspondientes hiperparámetros y por supuesto su número de épocas. Es entonces, que con estos datos la CNN puede iniciar su proceso de aprendizaje y clasificación, calculando así el valor de la función objetivo definida por su respectivo CE, para cada una de las partículas de la población. Este proceso de transformación también es un aporte del modelo H-PSO-CNN y se explica en la siguiente subsección.

4.2.2 Transformación de la Población: Arquitectura Numérica a Capas

La transformación de P a una estructura de datos llamada L , es un paso esencial para el correcto funcionamiento del H-PSO-CNN. Esta transformación tiene como objetivo llevar una población de partículas cuya naturaleza es numérica, a una totalmente

distinta cuya naturaleza permite el correcto funcionamiento de la CNN como un algoritmo clasificador. Para nuestro modelo propuesto, L se define como:

$$L = \begin{pmatrix} Y_{1,1} & Y_{1,2} \\ Y_{2,1} & Y_{2,2} \\ \vdots & \vdots \\ Y_{N,1} & Y_{N,2} \end{pmatrix} \quad (4.8)$$

La estructura L está compuesta por dos dimensiones, en donde N sigue representando a la misma partícula de P . Sin embargo, para cada $Y_{N,1}$ se almacena una arquitectura de capas distinta con sus respectivos hiperparámetros. El proceso de transformación de los valores presentes en: $X_{N,1}$, $X_{N,2}$ y $X_{N,3}$ a su respectivo $Y_{N,1}$, se logra mediante la implementación del Algorithm 2. Para el caso de $Y_{N,2}$ sólo es necesario trasladar cada uno de los valores contenidos en $X_{N,4}$.

Algorithm 2 Algoritmo para transformar población: arquitectura numérica a capas

Input: P

Output: L

```

1: for each  $X_{N,M}$  do
2:   Generate random hiperparam values
3:   if ( $M = 1$ ) then
4:     First Layer is Conv with hiperparam values
5:     Create  $X_{N,1} - 1$  Conv layers with hiperparam values on random positions
6:   end if
7:   if ( $M = 2$ ) then
8:     Create  $X_{N,2}$  Pooling layers with hiperparam values on random positions
9:   end if
10:  if ( $M = 3$ ) then
11:    Create  $X_{N,3} - 1$  FC layers with random units on random positions
12:    Last layer is FC with units number of classes
13:  end if
14:  Save the results of 3  $\rightarrow$  11 in  $Y_{N,1}$  like architecture list
15:   $Y_{N,2} = X_{N,4}$ 
16: end for
17: return  $L$ 

```

El Algorithm 2 recibe como entrada los valores contenidos en P . Según el valor de M , se crean las capas que corresponden a dicha dimensión. Cada tipo de capa en

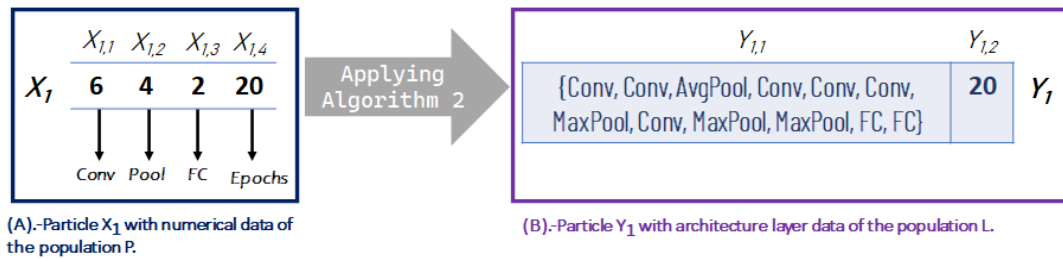


Figura 4.2: Transformación de datos numéricos de las partículas en datos de capas de arquitectura de partículas.

cuestión recibe sus correspondientes hiperparámetros, valores que se definen dentro de settings parameters of CNN en el Algorithm 1, y son generados de forma aleatoria. Cada *hiperparam* es distinto según corresponda el tipo de capa a crear para la partícula N .

La primera capa de toda partícula siempre es de tipo *Conv* y sus *hiperparam* son aleatorios, mientras que la última capa en la arquitectura es de tipo *FC*, y su número de neuronas (*units*) corresponde al número total de clases que se busque clasificar; en este caso su valor es igual a cinco. La posición entre las capas *Conv* y *Pooling* puede variar, siempre y cuando no se coloquen dentro del grupo de capas *FC*; para asignar una posición a las capas se utiliza una función *random* que coloca las capas según su valor calculado. Otra restricción corresponde a que las capas *FC* siempre deben estar al final de la arquitectura.

Por ejemplo, en la Fig. 4.2-A se muestra el contenido de una partícula con un *depth* = 12 y con veinte épocas; los datos de esta partícula corresponden a la población de P . Aplicando el Algorithm 2 se transforma X_1 a una partícula Y_1 (Ver Fig. 4.2-B), dónde $Y_{1,2}$ mantiene el mismo valor que está en $X_{1,4}$ y corresponde al número de

épocas. Por otra parte, en $Y_{1,1}$ se almacena la arquitectura creada para esa partícula, el número de capas *Conv* coincide con el valor almacenado en $X_{1,1}$, y su posición en la arquitectura se define de forma aleatoria, colocándose únicamente capas de este tipo entre las *Pooling* y *Conv*. El valor presente en $X_{1,2}$ es consistente con el número de capas de tipo *Pooling*, su posición se asigna de igual forma que las *Conv*, el subtipo de *Pooling* (*Avg* or *Max*) se define de forma aleatoria. Finalmente, $X_{1,3}$ coincide con el número total de capas de tipo *FC* y su posición corresponde al final de la arquitectura de capas de la partícula $Y_{1,1}$. Los parámetros que requiere cada tipo de capa, se encuentran definidos dentro de los Settings Parameters of CNN que fueron introducidos en el Algorithm 1; mismos que corresponden a los que se describen en la Fig. 2.1, cada uno de estos valores es configurable, por lo que el rango es variable.

Una vez que se concluye el proceso de transformación de P a L , el H-PSO-CNN, específicamente la CNN del modelo, toma cada una de las partículas en L y comienza con su proceso de entrenamiento y clasificación, obteniendo para cada partícula un valor de CE, mismo que es calculado propiamente por la CNN, al tratarse de una clasificación categórica multiclase.

4.2.3 Entrenamiento de la CNN y Estimación del CE

La función objetivo del PSO se define con el valor del error de entropía cruzada categórica o CE, y se calcula mediante:

$$CE = - \sum_{c=1}^M y_{o,c} \log(p_{o,c}) \quad (4.9)$$

Dónde: M es el número de clases, \log el logaritmo natural, y indicador binario (0 or 1) si la etiqueta de la clase es c es la clasificación correcta para la observación o , y p observación de probabilidad predicha o de la clase c .

Cada valor calculado de CE por parte de la CNN, para cada partícula de L , es tomado como un p_i^{best} , recuperando su valor y posición. Al calcular el CE para cada partícula de toda la población contenida en L , se genera un vector con estos valores y sus correspondientes posiciones. Dicho vector, es funcional para que el PSO comience con su tarea de optimización. Por lo que, el PSO toma como la mejor partícula global p_{gbest} al valor mínimo de todos los CE calculados, recuperando propiamente el valor y la posición de dicha partícula. A partir de este resultado el PSO realiza el cálculo de la velocidad y la nueva posición, a través de sus operadores. El proceso de actualización se realiza en la matriz P , por lo que toda vez que el PSO concluye con el cálculo de los nuevos valores y posiciones de las partículas, se vuelve indispensable actualizar también la estructura de datos L con los nuevos valores calculados por el PSO. Este procedimiento de actualización se explica con mayor detalle en la siguiente subsección.

4.2.4 Proceso de Actualización

El CE calculado mediante la Ecuación (4.9) para cada una de las partículas que conforman a la población de L , permite al modelo H-PSO-CNN, principalmente al PSO, realizar el proceso de optimización en P a través de sus operadores descritos en las Ecuaciones (3.1) y (3.2). Sin embargo, la actualización se realiza en P de forma

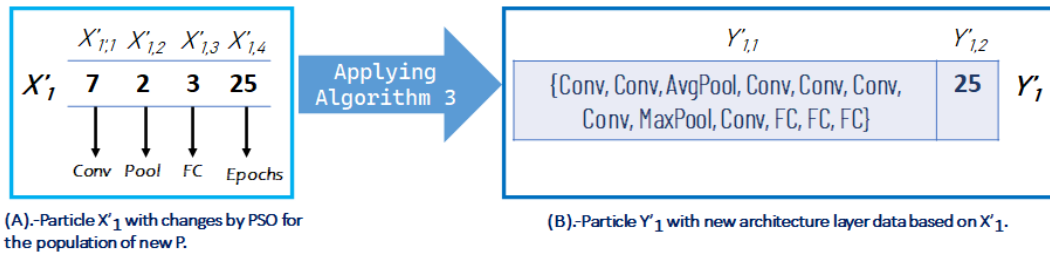


Figura 4.3: Actualización de L mediante la optimización del PSO y a través del uso de Algorithm 3.

directa, debido a que estas ecuaciones no permiten trabajar con una estructura de datos como lo es L , por lo que se vuelve necesaria la implementación de Algorithm 3, y con ello se logra aplicar los cambios de P a L . Resultando así, este procedimiento de actualización, un aporte más del modelo propuesto H-PSO-CNN.

El primer paso a realizar para lograr el proceso de actualización en la estructura de datos L , implica realizar una copia de la matriz P , y almacenarla como P_{old} . P_{old} sirve para detectar los cambios que los operadores del PSO realizan sobre P , toda vez que ya se haya concluido con el cálculo de los CE para cada partícula en L . Al realizar cambios en las partículas de P ; tanto en su contenido como en sus posiciones, es posible para el PSO obtener los p_i^{best} y de igual forma p_{gbest} , mismos que se actualizan en cada evolución del PSO, dicho procedimiento se vuelve repetitivo y concluirá al terminar ese número total de evoluciones.

La comparación entre P y P_{old} se realiza mediante el uso de la siguiente Ecuación:

$$P' = P - P_{old} \tag{4.10}$$

En cada evolución del PSO, se hace uso de las Ecuaciones (3.1) y (3.2), obteniendo así nuevos valores para P , mismos que tienen que ser considerados para la actualización de L . El resultado de la Ecuación (4.10), define las modificaciones que se deben realizar para cada una de las partículas. Ante esta situación, se pueden presentar tres posibles escenarios:

1. **Incremento:** Cuando el valor de P es mayor a P_{old} en cualquiera de sus dimensiones de cada partícula, se debe incrementar el número de capas o épocas acorde al nuevo valor obtenido por el PSO. Con esta actualización, la CNN comienza su entrenamiento y obtiene un nuevo CE en la próxima evolución del PSO.
2. **Decremento:** Cuando el valor de P es menor a P_{old} en cualquiera de sus dimensiones de cada partícula, se debe decrementar el número de capas o épocas acorde al nuevo valor obtenido por el PSO.
3. **Same:** No existe ningún cambio para la partícula, el entrenamiento de la CNN en la próxima evolución del PSO, permanece igual.

La actualización de los datos contenidos en L se realiza conforme al Algorithm 3, donde el número de capas según su tipo, cambiará conforme a la diferencia calculada entre P y P_{old} . El proceso de agregación de nuevas capas, al igual que en la población inicial es un proceso aleatorio, esto mismo sucede con la eliminación de capas; se decide la posición de las capas a eliminar o agregar de forma aleatoria. En el caso del número de épocas, por ser un valor numérico, sólo se cambia acorde a la diferencia

calculada. Si la diferencia no arroja ningún cambio entre las poblaciones, cada una de sus partículas se mantiene igual. En el caso de las nuevas posiciones de cada una de las partículas, sólo se cambian de su lugar mediante el índice anterior versus el actual. Por ejemplo, en la Fig. 4.3 se aprecian los cambios que el PSO realiza a la partícula de la Fig. 4.2. La optimización del PSO obtiene nuevos valores para X_1 expresados como X'_1 , donde se observa que las capas *Conv* incrementaron una unidad, por el contrario las capas *Pool* decrementaron dos unidades, las capas *FC* aumentaron una unidad y el número de épocas incrementó cinco unidades (Ver Fig. 4.3-A). Algorithm 3, permite la actualización de los datos que contiene la partícula Y_1 por los valores de Y'_1 , donde se puede apreciar de igual forma las modificaciones tanto en la arquitectura de capas como en el número de épocas (Ver Fig. 4.3-B).

Algorithm 3 Algoritmo para la actualización de L

Input: P, L **Output:** L

- 1: $P_{old} = P$
 - 2: Use Equation (4.10)
 - 3: **for** *each* particle of L **do**
 - 4: Update *Conv* layers with $X'_{N,1}$
 - 5: Update *Pool* layers with $X'_{N,2}$
 - 6: Update *FC* layers with $X'_{N,3}$
 - 7: All updates on *random* positions and *hiperparm*
 - 8: All updates save in $Y'_{N,1}$
 - 9: $Y'_{N,2} = X'_{N,4}$
 - 10: **end for**
 - 11: **return** L
-

Estos cambios son repetitivos hasta que se concluya con el número total de evoluciones del PSO. Al concluir este bucle, el H-PSO-CNN almacena la arquitectura de capas y número de épocas que corresponden a la mejor partícula global. Esta configuración de capas es utilizada en el siguiente subproceso, el cual consiste en evaluar el rendimiento de la CNN por medio de sus correspondientes métricas que se describen en el capítulo 6.

En el transcurso de este capítulo, hemos explorado las vastas posibilidades que se despliegan con el fusionamiento de dos paradigmas de IA importantes: el PSO y

la CNN. El modelo híbrido propuesto no solo representa la convergencia de diversas tecnologías, sino también la síntesis de décadas de investigación y desarrollo en el campo de la IA.

La ventaja inherente de un modelo híbrido radica en su capacidad para compensar las debilidades de una técnica individual con las fortalezas de otra. Así, cuando un enfoque puede fallar o ser insuficiente, otro entra en juego, asegurando robustez, precisión y adaptabilidad en un rango más amplio de escenarios y desafíos.

Es esencial reconocer que la verdadera innovación no se encuentra únicamente en el diseño de nuevos algoritmos o en la construcción de modelos más complejos, sino en la habilidad para orquestar, con maestría, lo mejor de cada herramienta y técnica existente. De esta manera, la propuesta de un modelo híbrido es un testimonio de cómo la colaboración y la integración pueden llevarnos más allá de lo que cualquier método individual podría lograr por sí solo.

Capítulo 5

Pruebas y Resultados

5.1 Banco de Datos de ECG

El MITDB ha representado en al menos los últimos 5 años, una referencia para el desarrollo y propuesta de algoritmos que incluyen a MLA-DLA o MA, con el fin de encontrar el mejor performance en la tarea de clasificación de ACs. La información contenida en el MITDB permite que estos modelos trabajen con una gran cantidad de datos, tanto en su fase de aprendizaje como en la de pruebas. El origen de estos datos proviene de 47 pacientes que fueron estudiados en el hospital Beth Israel de Boston [68], estas grabaciones se digitalizaron y fueron etiquetadas por dos o más cardiólogos, quienes obtuvieron un aproximado de 110, 000 anotaciones o CAs [76, 77].

A través de Physionet ATM Bank[20, 77], es posible obtener gratuitamente todo el conjunto de datos correspondientes al MITDB. Sin embargo, la información disponible en este recurso requiere de un procesamiento previo para que pueda ser utilizada

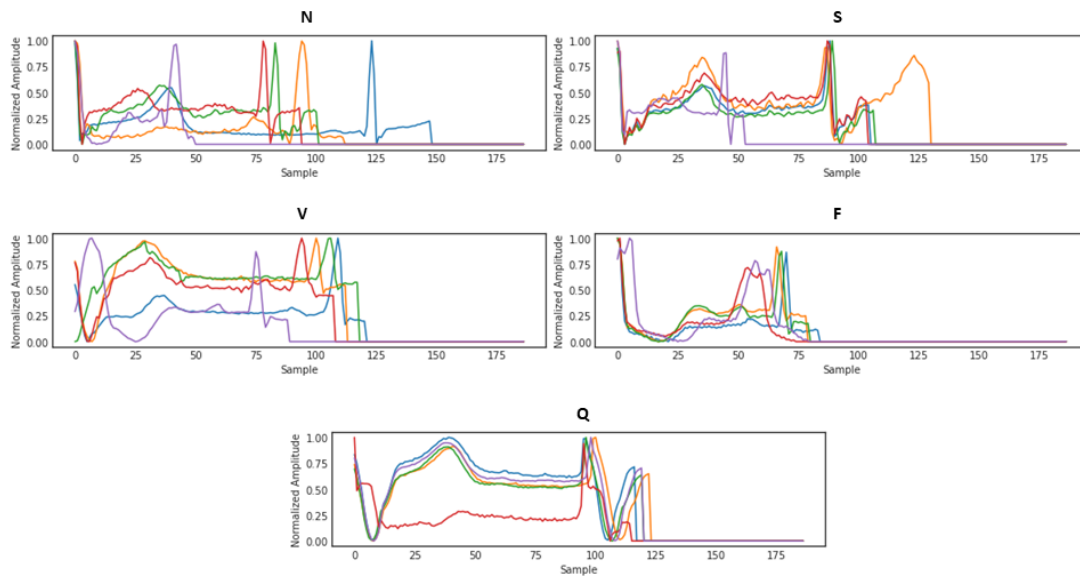


Figura 5.1: Conjunto de datos Arrhythmia del MIT-BIH agrupado en cinco clases según el estándar AAMI EC57.

con cualquier algoritmo de inteligencia artificial. En [20] se procesó la información disponible en Physionet ATM Bank y se organizó de forma estructurada. Su procesamiento le permitió obtener cinco clases de CAs diferentes (**N**, **S**, **V**, **F** y **Q**), y almacenar un total de 109,446 filas: 87,554 de ellas corresponden a los datos de entrenamiento (80%), mientras que 21,892 a los datos de pruebas (20%). Cada fila representa una anotación y se encuentra compuesta por 187 columnas, en donde la última contiene el valor de la clase a la que pertenece. Las otras 186 columnas restantes contienen valores numéricos normalizados con un rango de cero a uno y que pertenecen al trazo del ECG. Aquellas anotaciones que no completen las 186 columnas, son rellenadas con valores igual a cero[20].

Las 109,446 filas representan a las anotaciones realizadas por parte del equipo

Tabla 5.1: Mapeo entre diferentes clases y descripciones y clases AAMI EC57.

Clase	Descripción	Anotaciones
N	Nonectopic beat	Normal (NOR) Left bundle branch block (LBBB) Right bundle branch block beat (RBBB) Atrial escape (AE) Nodal (junctional) escape (NE)
S	Supraventricular ectopic beat	Atrial premature (AP) Aberrated atrial premature (aAP) Nodal premature (NP) Supraventricular premature (SP)
V	Ventricular ectopic beat	Premature ventricular contraction (PVC) Ventricular escape (VE)
F	Fusion beat	Fusion of ventricular and normal (fVN)
Q	Unknown beat	Paced (P) Fusion of paced and normal (fPN) Unclassifiable (U)

de cardiólogos del hospital Beth, cada una de estas anotaciones son agrupadas en cinco clases distintas conforme lo establece el estándar EC57 de la Asociación para el Avance de la Instrumentación Médica (AAMI). En la Tabla 5.1 se enlista al conjunto de anotaciones que pertenecen a cada una de las cinco clases: **N** contiene un total de cinco anotaciones, **S** a cuatro, **V** dos, **Q** a tres y **F** sólo una anotación.

La Figura 5.1 describe la morfología característica de cada una de las clases que fueron obtenidas por [71], en cada uno de los ejemplos se observa lo siguiente: la amplitud de los ECG es expresada en miliVolts y el número de muestras para toda anotación debe ser igual a ciento ochenta y seis, para cada clase se muestran cinco ejemplos de anotaciones.

5.2 Prueba y Obtención de Métricas de Clasificación

Este último procedimiento consiste en evaluar el rendimiento de la CNN, a través de la configuración de la mejor partícula obtenida por el PSO. El rendimiento de la CNN se obtiene a través de indicadores de evaluación cuantitativos: Matriz de Confusión, Precision, Recall, F1-Score and Accuracy.

5.2.1 Matriz de Confusión

Una matriz de confusión hace uso de: verdadero positivo (TP), falso positivo (FP), verdadero negativo (TN) y falso negativo (FN), para describir la relación estadística entre los valores reales y las clases previstas, con el fin de evaluar el rendimiento de la clasificación de la CNN.

5.2.2 Precision

Con esta métrica, es posible medir la calidad de la CNN en su tarea de clasificación, a través de:

$$precision = \frac{TP}{TP + FP} \quad (5.1)$$

5.2.3 Recall

La métrica de exhaustividad nos informa sobre la cantidad de ítems que la CNN es capaz de identificar correctamente. Su Ecuación es:

$$recall = \frac{TP}{TP + FN} \quad (5.2)$$

5.2.4 F1-Score

El valor F1 se utiliza para combinar las medidas de *precision* y *recall* en un sólo valor. Esto es práctico porque hace más fácil el poder comparar el rendimiento combinado de la precisión y la exhaustividad entre varias soluciones.

F1 se calcula haciendo la media armónica entre la precisión y la exhaustividad:

$$F1 = \frac{2TP}{2TP + FP + FN} \quad (5.3)$$

5.2.5 Accuracy

Accuracy mide el porcentaje de casos que el modelo ha acertado. Su cálculo se logra por medio de:

$$accuracy = \frac{TP + TN}{TP + TN + FP + FN} \quad (5.4)$$

5.3 Resultados de Experimentación y Discusión.

5.3.1 Entorno de Desarrollo

El proceso de experimentación y prueba del modelo H-PSO-CNN se realizó utilizando dos entornos de desarrollo, los cuales son:

- Software: Python 3.8.10 como lenguaje de programación, Keras 2.6.0 y Tensorflow 2.6.0 como bibliotecas para algoritmos Deep Learning y que permiten el uso de la tarjeta gráfica NVIDIA®️, Anaconda Navigator como un entorno de desarrollo integrado (IDE), y finalmente Visual Studio Code como editor de código.
- Hardware: NVIDIA®️ GeForce RTX 2060 6GB, Intel®️ Core™ i7-9750H CPU @ 2.60GHz, 31.92 GB RAM.

5.3.2 Parámetros para el H-PSO-CNN

Anteriormente en la explicación del H-PSO-CNN, se menciona que dicho modelo requiere algunos parámetros para su funcionamiento, estos se encuentran definidos por: la fuente de datos del MITDB y los parámetros para el funcionamiento tanto del PSO como la CNN. Por ejemplo, número de evoluciones, tamaño de la población, profundidad de la red neuronal, rango del tamaño de kernel, rango de número de neuronas, etc.

5.3.3 Fuente de Datos

La fuente de datos proviene del MITDB. Sin embargo, esta fuente requiere de un pre-procesamiento para que pueda ser entendida por una CNN. En la Fig. 5.1 se observa el comportamiento característico de cada una de las clases a clasificar. Recordemos, que el MITB contiene un total de 109,446 filas, cada fila corresponde a una arritmia que pertenece a una determinada clase. De las 109,446 filas: 87,554 corresponden a los

Tabla 5.2: Datos totales de prueba y entrenamiento de CNN.

Class	Train	Test	Total Data
N	72,471	18,118	90,589
S	2,223	556	2,779
V	5,788	1,448	7,236
F	641	162	803
Q	6,431	1,608	8,039
<i>Total</i>	87,554	21,892	109,446

datos de entrenamiento, mientras que las 21, 892 restantes a los datos de prueba. En la Tabla 5.2 se observa el número total de filas para cada una de las clases, siendo la clase **N** con mayor número de arritmias tanto para el entrenamiento como la prueba, y la clase **F** con el menor número de arritmias. La Fig. 5.2 nos describe en porcentaje la presencia de datos para cada clase, este porcentaje es el mismo para los dos conjuntos de datos descritos en la Tabla 5.2. Se observa que el 82.8% de los 87, 554 y 21,892, pertenecen a la clase **N**, un 7.3% para **Q**, **V** con un 6.6%, mientras que **S** contiene un 2.3%, y finalmente **F** con sólo un 0.7%. Lo anterior, nos describe un desbalanceo en el Dataset, dónde la cantidad de datos para todas las clases carece de homogeneidad.

PSO

El algoritmo metaheurístico PSO, además de requerir los coeficientes y vectores descritos en la Ecuación (3.1) y (3.2) respectivamente. También es necesario proporcionarle un número total de partículas o tamaño de la población y un número total de evoluciones. Los datos con los que se trabajó en la experimentación del modelo H-PSO-CNN, se describen en la Tabla 5.3. El número total de capas y su tipo se

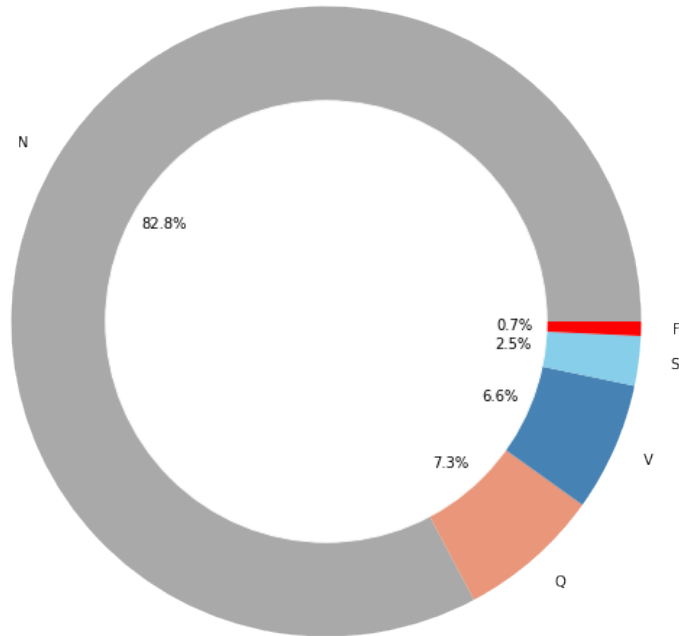


Figura 5.2: Balanceo para el MITDB con cinco clases de arritmias cardíacas.

define mediante el uso de las Ecuaciones (4.4), (4.5) y (4.6), dónde la profundidad de la CNN varía entre un mínimo de 2 y un máximo de 12 (Ver Tabla 5.3), diferente para cada partícula de P . Por lo que, para la experimentación se definen los valores de $coef_{Conv}$, $coef_{Pool}$, y $coef_{FC}$, en 50%, 30% y 20%, respectivamente. Lo anterior, demuestra que se pide al modelo H-PSO-CNN que cree un mayor número de capas tipo *Conv*, seguidas de las *Pool* y finalmente se les da un menor porcentaje a las capas *FC*. En [75], explican esta recomendación acorde a la complejidad de los datos de entrada, el número de clases objetivo y la complejidad en la clasificación.

Tabla 5.3: Parámetros de configuración de ajustes para CNN y PSO.

Modelo	Parámetro	Valor
PSO	<i>population</i>	20
	<i>evolutions</i>	25
	<i>w</i>	0.95
	<i>c₁, c₂</i>	2.0
	<i>r₁, r₂</i>	[0 - 1]
CNN	<i>depth</i>	[2-12]
	<i>epochs</i>	[1-25]
	<i>kernel</i>	[2 x 2] - [5 x 5]
	<i>filters</i>	[3 -256]
	<i>batch - size</i>	64
	<i>dropout</i>	0.5
	<i>units</i>	[5 -512]

CNN

Los parámetros mínimos que requiere una CNN para realizar su tarea de clasificación, también se describen en la Tabla 5.3, dónde se aprecia que se definen rangos con un valor mínimo y un máximo; lo que implica que estos son elegidos de forma aleatoria conforme a una función *random*, con el objetivo de que sea distinto para cada partícula. A diferencia, *batch - size* y *dropout*, que por naturaleza propia de la CNN se mantienen fijos y es el mismo para todas las partículas conformadas por arquitecturas de capas que pertenecen a L .

Entonces, es con estos datos que se configuró el modelo H-PSO-CNN, que permitieron obtener los resultados que a continuación se describen.

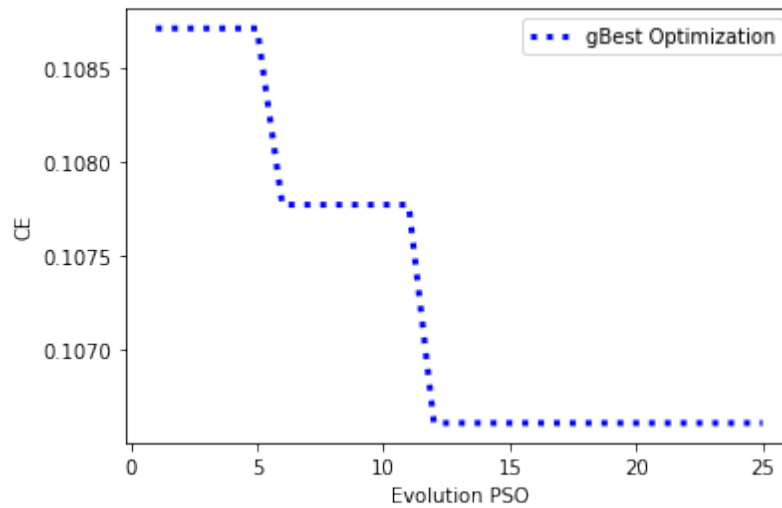


Figura 5.3: Optimización del CE con H-PSO-CNN.

5.3.4 Métricas

El modelo H-PSO-CNN propuesto es evaluado mediante las siguientes métricas cuantitativas: (1) Matriz de Confusión (2) Exactitud, (3) Precisión, (4) Especificidad, (5) Sensibilidad, y (6) Puntaje F1 (ver sección IV-E). Cada una de estas métricas se definen en las Ecuaciones (5.1), (5.2), (5.3), (5.4); y en conjunto evalúan el rendimiento que tiene la CNN en su tarea de clasificación, para las cinco clases de arritmias cardíacas. Estos valores se obtienen tanto en el entrenamiento como en la prueba de la arquitectura de capas representada por p_{gbest} .

El proceso de optimización por parte del PSO, de la función costo representada por el CE en la CNN, se observa en la Fig.5.3; en dónde se aprecia que el CE alcanzó su valor mínimo a partir de la iteración 12. El número total de evoluciones es igual a 25, y le llevó 17.68 horas al H-PSO-CNN encontrar esta arquitectura.

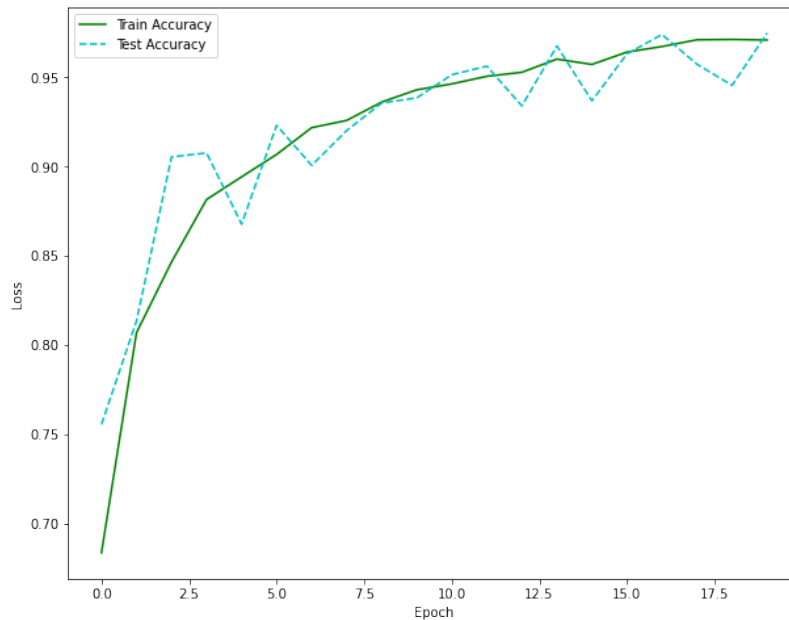


Figura 5.4: Precisión para la clasificación de arritmia cardíaca con H-PSO-CNN.

La arquitectura de capas que representa a la partícula p_{gbest} se describe en la Tabla 5.4. Esta arquitectura está configurada con 20 épocas y una profundidad de siete capas: 5 *Conv*, 1 *Avg-Pool* y 1 *FC*; cada una con sus correspondientes *hiperparam*.

Esta arquitectura obtuvo los valores de *accuracy* y *loss* que se muestran en las Fig. 5.4 y 5.5, durante el entrenamiento y prueba. Para ambas gráficas la línea continua representa el rendimiento de la CNN con la arquitectura de p_{gbest} durante la fase de entrenamiento, mientras que la línea punteada corresponde al performance durante la prueba de la arquitectura. En el entrenamiento se alcanzó un $CE = 0.044968$ con un *accuracy* = 0.98, mientras que durante la prueba un $CE = 0.084768$ y un *accuracy* = 0.97. La matriz de confusión se observa en la Fig. 5.6 y en ella se pueden observar los valores estimados de forma correcta por el modelo computacional

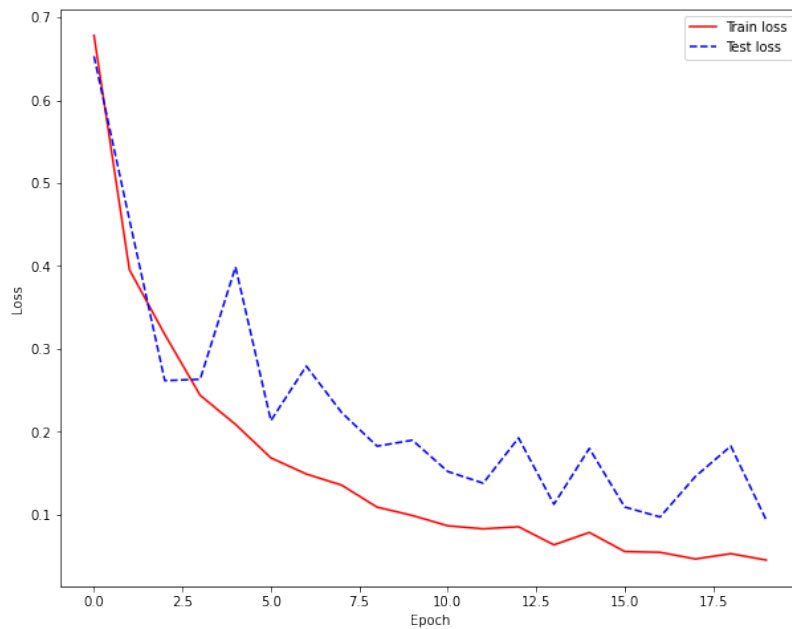


Figura 5.5: Valor de pérdida a través de la clasificación de arritmia cardíaca con H-PSO-CNN.

implementado.

El conjunto de métricas descritas en la sección IV-E para ambas fases, se describen en las Tablas 5.5 y 5.6.

El Train Accuracy logrado por el modelo asciende al 98%. Sin embargo, en la categoría **S** a pesar de tener una buena precisión, su recall es muy bajo, lo que significa que el modelo no detecta la clase muy bien, pero cuando lo hace es altamente confiable. Por otra parte, para la clase **F** el modelo obtuvo una precisión y una sensibilidad baja, es decir, no logra clasificar la clase correctamente, y se debe a que la cantidad de datos para esta clase es muy mínima, lo que implica que el Dataset está desbalanceado. En el campo de la salud, esta circunstancia es particularmente

Tabla 5.4: La mejor arquitectura de capas para CNN.

Capa	Parámetro	Valor
<i>Conv</i>	<i>kernel</i> <i>filters</i>	[5 x 5] 101
<i>Conv</i>	<i>kernel</i> <i>filters</i>	[3 x 3] 130
<i>Conv</i>	<i>kernel</i> <i>filters</i>	[5 x 5] 62
<i>Conv</i>	<i>kernel</i> <i>filters</i>	[4 x 4] 233
<i>Conv</i>	<i>kernel</i> <i>filters</i>	[4 x 4] 146
<i>Avg-Pool</i>	<i>kernel</i>	[2 x 2]
<i>FC</i>	<i>units</i>	5

Tabla 5.5: Métricas de rendimiento para CNN Train con H-PSO-CNN.

Clase	Precision	Recall	F1-Score
N	0.98	0.99	0.99
S	0.91	0.63	0.75
V	0.94	0.93	0.94
F	0.80	0.69	0.74
Q	0.99	0.98	0.99

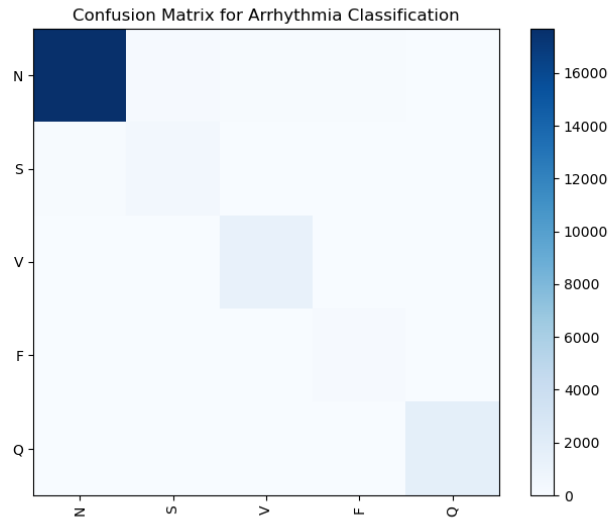


Figura 5.6: Matriz de confusión para la clasificación de arritmias cardíacas con H-PSO-CNN.

frecuente y por ello tenemos que recurrir al balanceo de clases. Cabe mencionar que el alcance de este trabajo no incluye el balanceo de la fuente de datos.

Por otra parte, el Test Accuracy asciende al 97%, en este performance la precisión para la clase **S** disminuyó, pero el recall incrementó de 0.63 a 0.83, lo que significa un aumento en la clasificación correcta de esta clase, pero un decremento en su confiabilidad determinada por su precisión. Una situación similar sucede de igual forma con la categoría **F**. Note que para las demás clases sus métricas son aceptables debido a que sus valores se encuentran por arriba del 0.90.

Tabla 5.6: Métricas de rendimiento para la prueba CNN con H-PSO-CNN.

Clase	Precision	Recall	F1-Score
N	0.99	0.98	0.98
S	0.69	0.83	0.75
V	0.92	0.95	0.94
F	0.48	0.88	0.62
Q	0.99	0.99	0.99

Capítulo 6

Conclusiones y Trabajos Futuros

6.1 Estudio Comparativo

La tarea de clasificar diferentes clases de arritmias, como hemos visto, se ha abordado mediante el uso de distintos tipos de algoritmos que van desde las técnicas clásicas de clasificación hasta el uso de modelos computacionales híbridos de inteligencia artificial. En la tabla 6.1 se describen algunos de los trabajos que se han desarrollado para esta tarea al menos en los últimos 5 años. La tabla 6.1 describe el año del trabajo, la(s) base(s) de datos utilizada(s), el número de clases de arritmias, si estos datos estaban balanceados, el tamaño de los datos, el algoritmo(s) utilizado(s), la precisión obtenida y finalmente si cada propuesta es capaz de obtener la arquitectura por capas de forma automática.

Los resultados que se reportan en la tabla 6.1 son igualmente satisfactorios o incluso superiores a los descritos en el presente trabajo de tesis. Sin embargo, si

Tabla 6.1: Resumen de las comparaciones entre el modelo propuesto y los métodos existentes.

Referencia.	Año	Dataset	Clases	Balanceo	Dimensión	Método	Accuracy	Automat.
[35]	2018	MITDB	2	No	1D	RNN-LSTM	82.5% - 85.4%	No
[33]	2019	MITDB	5	Si	1D	MLP	98.72%	No
[48]	2019	MITDB	5	No	1D	MLP-PSO	98.56%	No
[39]	2019	MITDB	5	Si	1D	CNN	99.06%	No
[47]	2020	MITDB	2	No	1D	BaROA-CNN	93.19%	No
[38]	2020	MITDB	15	Si	2D	CNN	99.05%	No
[40]	2021	MITDB, MITAFD, CUDB, VFDB	6	No	2D	CNN	95.3% - 98.41%	No
[43]	2021	MITDB & PTBDB	6	Si	1D	CNN-LSTM	99.82%	No
[46]	2021	MITDB, EDB & IN-CART	4	Si	1D	MPA-CNN	99.76%	No
[53]	2022	MITDB	3	Si	2D	CNN-LSTM	98.7%	No
Propuesta [78]	2023	MITDB	5	No	1D	H-PSO-CNN	98%	Yes

observamos a detalle la información de dicha tabla podemos observar que los autores de estas investigaciones trabajaron fuentes de datos balanceadas, o que la dimensión de la fuente de datos es distinta, o en su defecto el algoritmo clasificador es diferente a la CNN. Por las situaciones descritas con anterioridad, la propuesta del H-PSO-CNN, la hace única en su tipo y novedosa, al contar con la capacidad para construir y optimizar una arquitectura de capas adecuada, a partir de una población de partículas previamente definida. Esta capacidad se presenta de forma automatizada, algo que no tienen los otros modelos listados en la Tabla 6.1, en estos la arquitectura con la que trabajaban se definía de forma tradicional, algunos de ellos mediante el método de prueba y error.

6.2 Conclusiones

El modelo H-PSO-CNN demuestra ser un método híbrido capaz de reducir el tiempo de la búsqueda de una arquitectura de capas y número de épocas óptimo para obtener un rendimiento satisfactorio en la clasificación de arritmias cardíacas. Proceso que actualmente se realiza de manera artesanal en la mayoría de los modelos. Lo anterior, aumenta la efectividad de los algoritmos de aprendizaje profundo al encontrar los mejores parámetros que la red requiere en un determinado dominio de problema.

De un total de 109,446 diferentes muestras de arritmias cardíacas; 87,554 se consideraron para el entrenamiento y 21,892 para las pruebas. Se obtuvo un Accuracy del 98% y 97%, siendo **N**, **S**, **V** y **Q**, las clases con un mejor rendimiento.

Se concluye que, el modelo implementado es un enfoque simple, eficaz y útil que

puede ser utilizado por los expertos, para identificar de forma rápida y automática problemas cardiovasculares asociados a arritmias cardíacas. Sin embargo, para aumentar la efectividad del modelo, se requiere trabajar en el balanceo de los datos que pertenecen a cada una de las clases de arritmias cardíacas, en las que los valores de precisión y sensibilidad no resultaron satisfactorios.

La realización del presente trabajo de tesis permitió desarrollar dos aportaciones en formato de publicación de artículos, en [79] se publicaron los resultados que se obtuvieron con la propuesta del modelo H-PSO-CNN, mientras que en [78] se desarrolló un modelo que implementa AMH para la optimización global a través de un autómata celular que toma diferentes adaptaciones de la regla de la mayoría y su contra parte, la regla de la minoría, para establecer diferentes reglas que modifican los vectores de valores reales con el fin de lograr un buen equilibrio en las tareas de exploración y explotación para tareas de optimización.

6.3 Trabajos Futuros

En estudios futuros, el H-PSO-CNN puede ser utilizado con otras bases de datos e incluir más clases de arritmias cardíacas, mismas que también tendrían que ser balanceadas para llegar a un rendimiento mayormente satisfactorio que el obtenido en este trabajo. Además de que al ser un modelo computacional cuya tarea principal consiste en clasificar un cierto número determinado de clases, el H-PSO-CNN, bien puede utilizarse en otras áreas de la ciencia que busque mejorar la clasificación según sus necesidades, por lo que la presente propuesta no es exclusiva para el ámbito

médico. Finalmente, el H-PSO-CNN puede ser mejorado al incluir más dimensiones que definan a cada una de las partículas de la población, lo que implica la posibilidad de incluir más hiperparámetros de la CNN.

Referencias

- [1] J. J. Hopfield, “Artificial neural networks,” *IEEE Circuits and Devices Magazine*, vol. 4, no. 5, pp. 3–10, 1988.
- [2] S.-C. Wang and S.-C. Wang, “Artificial neural network,” *Interdisciplinary computing in java programming*, pp. 81–100, 2003.
- [3] A. Krogh, “What are artificial neural networks?,” *Nature biotechnology*, vol. 26, no. 2, pp. 195–197, 2008.
- [4] A. Dongare, R. Kharde, A. D. Kachare, *et al.*, “Introduction to artificial neural network,” *International Journal of Engineering and Innovative Technology (IJEIT)*, vol. 2, no. 1, pp. 189–194, 2012.
- [5] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, “State-of-the-art in artificial neural network applications: A survey,” *Heliyon*, vol. 4, no. 11, 2018.
- [6] X.-S. Yang, *Nature-inspired metaheuristic algorithms*. Luniver press, 2010.

-
- [7] A. H. Gandomi, X.-S. Yang, S. Talatahari, and A. H. Alavi, "Metaheuristic algorithms in modeling and optimization," *Metaheuristic applications in structures and infrastructures*, vol. 1, pp. 1–24, 2013.
- [8] M. Abdel-Basset, L. Abdel-Fatah, and A. K. Sangaiah, "Metaheuristic algorithms: A comprehensive review," *Computational intelligence for multimedia big data on the cloud with engineering applications*, pp. 185–231, 2018.
- [9] T. Dokeroglu, E. Sevinc, T. Kucukyilmaz, and A. Cosar, "A survey on new generation metaheuristic algorithms," *Computers & Industrial Engineering*, vol. 137, p. 106040, 2019.
- [10] H. Khachnaoui, R. Guetari, and N. Khelifa, "A review on deep learning in thyroid ultrasound computer-assisted diagnosis systems," in *2018 IEEE International Conference on Image Processing, Applications and Systems (IPAS)*, pp. 291–297, IEEE, 2018.
- [11] P. R. Jeyaraj and E. R. Samuel Nadar, "Computer-assisted medical image classification for early diagnosis of oral cancer employing deep learning algorithm," *Journal of cancer research and clinical oncology*, vol. 145, pp. 829–837, 2019.
- [12] B. Liu, W. Chi, X. Li, P. Li, W. Liang, H. Liu, W. Wang, and J. He, "Evolving the pulmonary nodules diagnosis from classical approaches to deep learning-aided decision support: three decades' development course and future prospect," *Journal of cancer research and clinical oncology*, vol. 146, pp. 153–185, 2020.

-
- [13] H. Y. Chong, H. J. Yap, S. C. Tan, K. S. Yap, and S. Y. Wong, “Advances of metaheuristic algorithms in training neural networks for industrial applications,” *Soft Computing*, vol. 25, no. 16, pp. 11209–11233, 2021.
- [14] M. Kaveh and M. S. Mesgari, “Application of meta-heuristic algorithms for training neural networks and deep learning architectures: A comprehensive review,” *Neural Processing Letters*, vol. 55, no. 4, pp. 4519–4622, 2023.
- [15] K. Rajwar, K. Deep, and S. Das, “An exhaustive review of the metaheuristic algorithms for search and optimization: taxonomy, applications, and open challenges,” *Artificial Intelligence Review*, pp. 1–71, 2023.
- [16] A. Akay and H. Hess, “Deep learning: current and emerging applications in medicine and technology,” *IEEE journal of biomedical and health informatics*, vol. 23, no. 3, pp. 906–920, 2019.
- [17] N. Aisu, M. Miyake, K. Takeshita, M. Akiyama, R. Kawasaki, K. Kashiwagi, T. Sakamoto, T. Oshika, and A. Tsujikawa, “Regulatory-approved deep learning/machine learning-based medical devices in japan as of 2020: A systematic review,” *PLOS Digital Health*, vol. 1, no. 1, p. e0000001, 2022.
- [18] D. T. Marc, J. Beattie, V. Herasevich, L. Gatewood, and R. Zhang, “Assessing metadata quality of a federally sponsored health data repository,” in *AMIA Annual Symposium Proceedings*, vol. 2016, p. 864, American Medical Informatics Association, 2016.

- [19] J. Parker, M. Talih, D. J. Malec, V. Beresovsky, M. D. Carroll, J. F. Gonzalez, B. E. Hamilton, D. D. Ingram, K. D. Kochanek, F. McCarty, *et al.*, “National center for health statistics data presentation standards for proportions,” 2017.
- [20] G. B. Moody, “Physionet,” in *Encyclopedia of Computational Neuroscience*, pp. 2806–2808, Springer, 2022.
- [21] T. A. Manolio, C. D. Furberg, P. M. Rautaharju, D. Siscovick, A. B. Newman, N. O. Borhani, J. M. Gardin, and B. Tabatznik, “Cardiac arrhythmias on 24-h ambulatory electrocardiography in older women and men: the cardiovascular health study,” *Journal of the American College of Cardiology*, vol. 23, no. 4, pp. 916–925, 1994.
- [22] M. Gertsch, *The ECG: a two-step approach to diagnosis*. Springer Science & Business Media, 2003.
- [23] P. De Chazal, M. O’Dwyer, and R. B. Reilly, “Automatic classification of heartbeats using ecg morphology and heartbeat interval features,” *IEEE transactions on biomedical engineering*, vol. 51, no. 7, pp. 1196–1206, 2004.
- [24] P.-C. Chang, J.-J. Lin, J.-C. Hsieh, and J. Weng, “Myocardial infarction classification with multi-lead ecg using hidden markov models and gaussian mixture models,” *Applied Soft Computing*, vol. 12, no. 10, pp. 3165–3175, 2012.
- [25] S.-T. Pan, T.-P. Hong, and H.-C. Chen, “Ecg signal analysis by using hidden markov model,” in *2012 International conference on Fuzzy Theory and Its Applications (iFUZZY2012)*, pp. 288–293, IEEE, 2012.

- [26] K. N. Rajesh and R. Dhuli, "Classification of ecg heartbeats using nonlinear decomposition methods and support vector machine," *Computers in biology and medicine*, vol. 87, pp. 271–284, 2017.
- [27] X. Chen, Y. Wang, L. Wang, *et al.*, "Arrhythmia recognition and classification using ecg morphology and segment feature analysis," *IEEE/ACM transactions on computational biology and bioinformatics*, vol. 16, no. 1, pp. 131–138, 2018.
- [28] Mohebbanaaz, L. Rajani Kumari, and Y. Padma Sai, "Classification of arrhythmia beats using optimized k-nearest neighbor classifier," in *Intelligent Systems: Proceedings of ICMIB 2020*, pp. 349–359, Springer, 2021.
- [29] Y. Touluni, T. Belhoussine Drissi, and B. Nsiri, "Ecg signal diagnosis using discrete wavelet transform and k-nearest neighbor classifier.," in *Proceedings of the 4th International Conference on Networking, Information Systems & Security*, pp. 1–6, 2021.
- [30] M. Kropf, D. Hayn, and G. Schreier, "Ecg classification based on time and frequency domain features using random forests," in *2017 Computing in Cardiology (CinC)*, pp. 1–4, IEEE, 2017.
- [31] A. Mincholé, J. Camps, A. Lyon, and B. Rodríguez, "Machine learning in the electrocardiogram," *Journal of electrocardiology*, vol. 57, pp. S61–S64, 2019.
- [32] X. Liu, H. Wang, Z. Li, and L. Qin, "Deep learning in ecg diagnosis: A review," *Knowledge-Based Systems*, vol. 227, p. 107187, 2021.

- [33] G. Kumar, U. Pawar, and R. O'Reilly, "Arrhythmia detection in ecg signals using a multilayer perceptron network.," *AICS*, vol. 2563, pp. 353–364, 2019.
- [34] P. H. Borghi, R. C. Borges, and J. P. Teixeira, "Atrial fibrillation classification based on mlp networks by extracting jitter and shimmer parameters," *Procedia Computer Science*, vol. 181, pp. 931–939, 2021.
- [35] S. Singh, S. K. Pandey, U. Pawar, and R. R. Janghel, "Classification of ecg arrhythmia using recurrent neural networks," *Procedia computer science*, vol. 132, pp. 1290–1297, 2018.
- [36] J. Gao, H. Zhang, P. Lu, Z. Wang, *et al.*, "An effective lstm recurrent network to detect arrhythmia on imbalanced ecg dataset," *Journal of healthcare engineering*, vol. 2019, 2019.
- [37] J. Ferretti, V. Randazzo, G. Cirrincione, and E. Pasero, "1-d convolutional neural network for ecg arrhythmia classification," *Progresses in artificial intelligence and neural systems*, pp. 269–279, 2021.
- [38] R. Rohmantri and N. Surantha, "Arrhythmia classification using 2d convolutional neural network," *International Journal of Advanced Computer Science and Applications*, vol. 11, no. 4, 2020.
- [39] H. Wang, H. Shi, X. Chen, L. Zhao, Y. Huang, and C. Liu, "An improved convolutional neural network based approach for automated heartbeat classification," *Journal of medical systems*, vol. 44, pp. 1–9, 2020.

- [40] B. M. Mathunjwa, Y.-T. Lin, C.-H. Lin, M. F. Abbod, and J.-S. Shieh, "Ecg arrhythmia classification by using a recurrence plot and convolutional neural network," *Biomedical Signal Processing and Control*, vol. 64, p. 102262, 2021.
- [41] B.-H. Kim and J.-Y. Pyun, "Ecg identification for personal authentication using lstm-based deep recurrent neural networks," *Sensors*, vol. 20, no. 11, p. 3069, 2020.
- [42] J. Wang, "An intelligent computer-aided approach for atrial fibrillation and atrial flutter signals classification using modified bidirectional lstm network," *Information Sciences*, vol. 574, pp. 320–332, 2021.
- [43] H. M. Rai and K. Chatterjee, "Hybrid cnn-lstm deep learning model and ensemble technique for automatic detection of myocardial infarction using big ecg data," *Applied Intelligence*, vol. 52, no. 5, pp. 5366–5384, 2022.
- [44] G. Petmezas, K. Haris, L. Stefanopoulos, V. Kilintzis, A. Tzavelis, J. A. Rogers, A. K. Katsaggelos, and N. Maglaveras, "Automated atrial fibrillation detection using a hybrid cnn-lstm network on imbalanced ecg datasets," *Biomedical Signal Processing and Control*, vol. 63, p. 102194, 2021.
- [45] N. Lopac, F. Hržić, I. P. Vuksanović, and J. Lerga, "Detection of non-stationary gw signals in high noise from cohen's class of time–frequency representations using deep learning," *Ieee Access*, vol. 10, pp. 2408–2428, 2021.
- [46] E. H. Houssein, D. S. Abdelminaam, I. E. Ibrahim, M. Hassaballah, and Y. M. Wazery, "A hybrid heartbeats classification approach based on marine predators

- algorithm and convolution neural networks,” *IEEE Access*, vol. 9, pp. 86194–86206, 2021.
- [47] D. K. Atal and M. Singh, “Arrhythmia classification with ecg signals based on the optimization-enabled deep convolutional neural network,” *Computer Methods and Programs in Biomedicine*, vol. 196, p. 105607, 2020.
- [48] F. Bouaziz and D. Boutana, “Automated ecg heartbeat classification by combining a multilayer perceptron neural network with enhanced particle swarm optimization algorithm,” *Research on Biomedical Engineering*, vol. 35, pp. 143–155, 2019.
- [49] J. E. S. Simanjuntak, M. L. Khodra, and M. C. T. Manullang, “Design methods of detecting atrial fibrillation using the recurrent neural network algorithm on the arduino ad8232 ecg module,” in *Iop conference series: Earth and environmental science*, vol. 537, p. 012022, IOP Publishing, 2020.
- [50] A. Nainwal, Y. Kumar, and B. Jha, “An ecg classification using dnn classifier with modified pigeon inspired optimizer,” *Multimedia Tools and Applications*, vol. 81, no. 7, pp. 9131–9150, 2022.
- [51] A. A. Abdelhamid, E.-S. M. El-Kenawy, B. Alotaibi, G. M. Amer, M. Y. Abdelkader, A. Ibrahim, and M. M. Eid, “Robust speech emotion recognition using cnn+ lstm based on stochastic fractal search optimization algorithm,” *IEEE Access*, vol. 10, pp. 49265–49284, 2022.

- [52] P. Sharma, S. K. Dinkar, and D. Gupta, "A novel hybrid deep learning method with cuckoo search algorithm for classification of arrhythmia disease using ecg signals," *Neural computing and Applications*, vol. 33, pp. 13123–13143, 2021.
- [53] P. Madan, V. Singh, D. P. Singh, M. Diwakar, B. Pant, and A. Kishor, "A hybrid deep learning approach for ecg-based arrhythmia classification," *Bioengineering*, vol. 9, no. 4, p. 152, 2022.
- [54] E. H. Houssein, A. A. Ewees, and M. A. ElAziz, "Improving twin support vector machine based on hybrid swarm optimizer for heartbeat classification," *Pattern Recognition and Image Analysis*, vol. 28, pp. 243–253, 2018.
- [55] B. Navaneeth and M. Suchetha, "Pso optimized 1-d cnn-svm architecture for real-time detection and classification applications," *Computers in biology and medicine*, vol. 108, pp. 85–92, 2019.
- [56] E. J. d. S. Luz, W. R. Schwartz, G. Cámara-Chávez, and D. Menotti, "Ecg-based heartbeat classification for arrhythmia detection: A survey," *Computer methods and programs in biomedicine*, vol. 127, pp. 144–164, 2016.
- [57] U. R. Acharya, H. Fujita, M. Adam, O. S. Lih, V. K. Sudarshan, T. J. Hong, J. E. Koh, Y. Hagiwara, C. K. Chua, C. K. Poo, *et al.*, "Automated characterization and classification of coronary artery disease and myocardial infarction by decomposition of ecg signals: A comparative study," *Information Sciences*, vol. 377, pp. 17–29, 2017.

- [58] L. Gaztañaga, F. E. Marchlinski, and B. P. Betensky, “Mecanismos de las arritmias cardíacas,” *Revista española de cardiología*, vol. 65, no. 2, pp. 174–185, 2012.
- [59] A. Porta-Sánchez, R. Casado, O. Salvador, C. Sánchez-Enrique, S. Bayona-Horta, P. Sánchez-Borque, J. R. Campal, and J. Cabrera, “Arritmias cardíacas,” *Medicine-Programa de Formación Médica Continuada Acreditado*, vol. 13, no. 44, pp. 2568–2576, 2021.
- [60] J. A. Lozano, “Arritmias cardíacas y su tratamiento,” *Offarm*, vol. 20, no. 11, pp. 96–105, 2001.
- [61] J. Gómez Flores, “Arritmias cardíacas,” *Archivos de cardiología de México*, vol. 79, pp. 2–2, 2009.
- [62] D.-g. Fu, “Cardiac arrhythmias: diagnosis, symptoms, and treatments,” *Cell biochemistry and biophysics*, vol. 73, no. 2, pp. 291–296, 2015.
- [63] R. T. Giffoni and R. M. Torres, “Breve história da eletrocardiografia,” *Revista Médica de Minas Gerais*, vol. 20, no. 2, pp. 263–270, 2010.
- [64] H. J. Wellens, “Cardiac arrhythmias: the quest for a cure: a historical perspective,” *Journal of the American College of Cardiology*, vol. 44, no. 6, pp. 1155–1163, 2004.
- [65] B. Lüderitz, “History of cardiac rhythm disorders,” *Clinical Research in Cardiology*, vol. 91, pp. iv50–iv55, 2002.

- [66] G. B. Moody and R. G. Mark, “The mit-bih arrhythmia database on cd-rom and software for use with it,” in *[1990] Proceedings Computers in Cardiology*, pp. 185–188, IEEE, 1990.
- [67] V. Thakkar, S. Tewary, and C. Chakraborty, “Batch normalization in convolutional neural networks—a comparative study with cifar-10 data,” in *2018 fifth international conference on emerging applications of information technology (EAIT)*, pp. 1–5, IEEE, 2018.
- [68] S. Albawi, T. A. Mohammed, and S. Al-Zawi, “Understanding of a convolutional neural network,” in *2017 international conference on engineering and technology (ICET)*, pp. 1–6, Ieee, 2017.
- [69] E. Cetinic, T. Lipic, and S. Grgic, “Learning the principles of art history with convolutional neural networks,” *Pattern Recognition Letters*, vol. 129, pp. 56–62, 2020.
- [70] B. Macukow, “Neural networks—state of art, brief history, basic models and architecture,” in *Computer Information Systems and Industrial Management: 15th IFIP TC8 International Conference, CISIM 2016, Vilnius, Lithuania, September 14-16, 2016, Proceedings 15*, pp. 3–14, Springer, 2016.
- [71] E. Garcia-Gonzalo and J. L. Fernandez-Martinez, “A brief historical review of particle swarm optimization (pso),” *Journal of Bioinformatics and Intelligent Control*, vol. 1, no. 1, pp. 3–16, 2012.

- [72] M. N. K. Kulkarni, M. S. Patekar, M. T. Bhoskar, M. O. Kulkarni, G. Kakandikar, and V. Nandedkar, "Particle swarm optimization applications to mechanical engineering-a review," *Materials Today: Proceedings*, vol. 2, no. 4-5, pp. 2631–2639, 2015.
- [73] M. R. AlRashidi and M. E. El-Hawary, "A survey of particle swarm optimization applications in electric power systems," *IEEE transactions on evolutionary computation*, vol. 13, no. 4, pp. 913–918, 2008.
- [74] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95-international conference on neural networks*, vol. 4, pp. 1942–1948, IEEE, 1995.
- [75] T. M. Shami, A. A. El-Saleh, M. Alswaitti, Q. Al-Tashi, M. A. Summakieh, and S. Mirjalili, "Particle swarm optimization: A comprehensive survey," *IEEE Access*, vol. 10, pp. 10031–10061, 2022.
- [76] Z. F. M. Apandi, R. Ikeura, and S. Hayakawa, "Arrhythmia detection using mit-bih dataset: A review," in *2018 International Conference on Computational Approach in Smart Systems Design and Applications (ICASSDA)*, pp. 1–5, IEEE, 2018.
- [77] A. L. Goldberger, L. A. Amaral, L. Glass, J. M. Hausdorff, P. C. Ivanov, R. G. Mark, J. E. Mietus, G. B. Moody, C.-K. Peng, and H. E. Stanley, "Physiobank, physiotoolkit, and physionet: components of a new research resource for complex physiologic signals," *circulation*, vol. 101, no. 23, pp. e215–e220, 2000.

-
- [78] J. C. Seck-Tuoh-Mora, N. Hernandez-Romero, F. Santander-Baños, V. Volpi-Leon, J. Medina-Marin, and P. Lagos-Eulogio, “A majority–minority cellular automata algorithm for global optimization,” *Expert Systems with Applications*, vol. 203, p. 117379, 2022.
- [79] F. S. Baños, N. H. Romero, J. C. S. T. Mora, J. M. Marín, I. B. Vite, and G. E. A. Fuentes, “A novel hybrid model based on convolutional neural network with particle swarm optimization algorithm for classification of cardiac arrhythmias,” *IEEE Access*, 2023.