

**DECANATURA DE INGENIERÍA INDUSTRIAL
DECANATURA DE INGENIERÍA DE SISTEMAS
DECANATURA DE MATEMÁTICAS
MAESTRÍA EN CIENCIA DE DATOS
FORMATO DE ENTREGA TRABAJO DE GRADO**

Fecha de entrega:

Estudiante: Juan Manuel Liscano Fierro - juan.liscano-f@mail.escuelaing.edu.co

Director: Hector Javier Hortua Orjuela - hector.hortua@escuelaing.edu.co

Codirector: NA

El presente documento avala la entrega del trabajo de grado por parte del director y codirector.
Documentos anexos: copia digital del Trabajo de Grado (1).

Firma Director

Firma Estudiante

Cuantificación de la incertidumbre en la clasificación de imágenes y evaluación del impacto de las redes neuronales bayesianas en la clasificación de pacientes con neumonía por COVID-19 en tomografías computarizadas 3D y 2D: un análisis comparativo de rendimiento

Juan Manuel Liscano Fierro

**Escuela Colombiana de Ingeniería Julio Garavito
Decanatura de Ingeniería Industrial
Decanatura de Ingeniería de Sistemas
Decanatura de Matemáticas
Maestría en Ciencia de Datos
Bogotá D.C., Colombia
2024**

Cuantificación de la incertidumbre en la clasificación de imágenes y evaluación del impacto de las redes neuronales bayesianas en la clasificación de pacientes con neumonía por COVID-19 en tomografías computarizadas 3D y 2D: un análisis comparativo de rendimiento

Juan Manuel Liscano Fierro

Trabajo de grado para optar al título de
Magíster en Ciencia de Datos

Director
Hector Javier Hortúa Orjuela
Doctor

Escuela Colombiana de Ingeniería Julio Garavito
Decanatura de Ingeniería Industrial
Decanatura de Ingeniería de Sistemas
Decanatura de Matemáticas
Maestría en Ciencia de Datos
Bogotá D.C., Colombia
2024

©Únicamente se puede usar el contenido de las publicaciones para propósitos de información. No se debe copiar, enviar, recortar, transmitir o redistribuir este material para propósitos comerciales sin la autorización de la Escuela Colombiana de Ingeniería. Cuando se use el material de la Escuela se debe incluir la siguiente nota “Derechos reservados a Escuela Colombiana de Ingeniería” en cualquier copia en un lugar visible. Y el material no se debe notificar sin el permiso de la Escuela.

Publicado en 2013 por la Escuela Colombiana de Ingeniería Julio Garavito. Avenida 13 No
205-59 Bogotá. Colombia
TEL: +57 – 1 668 36 00

Agradecimientos

A mi tutor, Hector Hortúa, quiero expresar mi más sincero agradecimiento por su dedicación y orientación a lo largo de este camino de investigación. Su profundo conocimiento en el campo y su disposición para compartirlo han sido la brújula que ha guiado cada paso de este proyecto. Sin sus valiosos consejos este trabajo no habría alcanzado su plenitud.

A mi compañera de vida, Kata, por alentarme constantemente. Su confianza en mí fue el motor que me impulsó en este proyecto. En cada altibajo, en cada momento de duda, su apoyo incondicional y su fe en mi capacidad fueron la luz que iluminó el camino. No solo ha sido mi apoyo emocional, sino también mi cómplice en esta aventura académica.

A mi amigo Luis, por su constante interés y apoyo durante todo el proceso.

A mi familia, por su inquebrantable apoyo y amor durante este viaje académico.

Resumen

La clasificación precisa de la neumonía por COVID-19 en tomografías computarizadas 3D y 2D sigue siendo un desafío en el análisis de imágenes médicas. Aunque las redes neuronales, especialmente las determinísticas, han mostrado promisorios resultados en esta área, carecen de la capacidad para proporcionar una medida de incertidumbre en sus predicciones, un aspecto crucial en la toma de decisiones clínicas. En contraste, las redes neuronales bayesianas ofrecen una interpretación probabilística de sus predicciones, permitiendo cuantificar la incertidumbre y mejorar la toma de decisiones.

Este proyecto busca comparar la eficiencia de las redes neuronales bayesianas frente a las determinísticas en la clasificación de la neumonía por COVID-19 en tomografías 3D y 2D. Se utilizará un conjunto de datos de 'MosMedData: Chest CT Scans with COVID-19 Related Findings' [Morozov et al., 2020]. Ambos tipos de redes se desarrollarán y entrenarán bajo una arquitectura adecuada, evaluando su rendimiento en precisión, sensibilidad, especificidad y AUC-ROC.

Los resultados proporcionarán información valiosa sobre los beneficios potenciales de las redes neuronales bayesianas en imágenes médicas, especialmente en la clasificación de neumonía COVID-19. Al cuantificar la incertidumbre, estas redes pueden mejorar las decisiones clínicas, contribuyendo a los esfuerzos continuos para mejorar el diagnóstico y tratamiento de la neumonía por COVID-19.

Abstract

Accurately classifying COVID-19 pneumonia in 3D and 2D CT scans remains a significant challenge in the field of medical image analysis. Although neural networks, especially deterministic ones, have shown promising results in this area, they lack the ability to provide a measure of uncertainty in their predictions, a crucial aspect in clinical decision making. In contrast, Bayesian neural networks offer a probabilistic interpretation of their predictions, allowing to quantify uncertainty and improve decision making.

This research project seeks to compare the efficiency of Bayesian versus deterministic neural networks in the classification of COVID-19 pneumonia in 3D and 2D CT scans. A dataset from 'MosMedData: Chest CT Scans with COVID-19 Related Findings' will be used [Morozov et al., 2020]. Both types of networks will be developed and trained under a suitable architecture, evaluating their performance in terms of accuracy, sensitivity, specificity and AUC-ROC.

The results will provide valuable information on the potential benefits of Bayesian neural networks in medical imaging, especially in COVID-19 pneumonia classification. By quantifying uncertainty, these networks can improve clinical decisions, contributing to ongoing efforts to improve the diagnosis and treatment of COVID-19 pneumonia.

Índice

1. Introducción	12
1.1. Planteamiento del problema	12
1.2. Objetivos	14
1.2.1. General	14
1.2.2. Específicos	14
1.3. Alcance y Limitaciones	14
1.3.1. Alcance	14
1.3.2. Limitaciones	15
2. Fundamentación Teórica	16
2.1. Redes Neuronales Profundas	16
2.1.1. Redes Neuronales Convolucionales	17
2.1.2. Capas de Normalización y Pooling en CNN	18
2.2. Redes Neuronales Bayesianas	19
2.2.1. Distribución de pesos w a posteriori	21
2.2.2. Inferencia variacional	22
2.2.2.1. Divergencias Alternativas	23
2.2.2.2. Distribuciones Variacionales	23
2.2.2.3. Técnicas de perturbación a los pesos	24
2.2.2.3.1. Flipout	24
2.2.2.3.2. Reparametrización	25
2.2.3. Flujos de normalización multiplicativos (MNF)	25
2.2.4. Flujos de normalización multiplicativos en una representación de cúbica de véxeles	26
2.3. Calibración en Redes Neuronales Profundas	27
2.3.1. Diagramas de Confiabilidad:	28
2.3.2. Error de Calibración Esperado (ECE):	29
2.4. Métricas para evaluar el rendimiento del modelo	29
3. Marco Metodológico	32
3.1. Configuración del Entorno	32
3.2. Datos	32
3.3. Metodología	33
3.3.1. Tomografías 3D:	33
3.3.2. Tomografías 2D:	36
4. Resultados:	37
4.1. Ventana HU:	37
4.2. Data Augmentation:	40
4.3. Tomografías 3D:	42
4.3.1. Modelos:	42
4.3.1.1. Redes Deterministas	42
4.3.1.2. Redes Bayesianas	47
4.3.2. Calibración:	50
4.3.3. Incertidumbre:	51
4.3.3.1. Análisis de imágenes individuales:	51

4.4. Tomografías 2D:	57
4.4.1. Modelos:	57
5. Conclusiones	59
6. ABREVIACIONES	65
A. ANEXOS	66
A.1. Descripción modelos evaluados	66
A.2. Cambios Iniciales Transición Determinista - Bayesiana	68
A.3. Métricas Primeras Alternativas Redes Bayesianas	69
A.4. Monte Carlo Dropout	69
A.5. Coordenadas Paralelas - Ajuste Capa Sigmoid	70
A.6. Coordenadas Paralelas - Ajuste Umbral	74
A.7. Diagramas Confiability	76

Índice de figuras

1.	Arquitectura de una RNN compuesta por capas convolucionales. Tomada de developers-breach	18
2.	Ilustración de la generación de una predicción a partir de una red neuronal bayesiana mediante el muestreo de Monte Carlo. Una red neuronal estándar (A, arriba a la izquierda) tiene un peso para cada una de sus conexiones (w_*), ajustadas a través del conjunto de entrenamiento y utilizada para generar una predicción para un ejemplo de prueba. Una red neuronal bayesiana (B, abajo a la izquierda) tiene, en cambio, una distribución posterior para cada peso, parametrizada por θ ($q_\theta(w)$). El proceso de entrenamiento comienza con una distribución previa asignada para cada peso y devuelve una distribución posterior aproximada. En el momento de la prueba (C, derecha), se extrae una muestra w_1 (rojo) de la distribución a posteriori de los pesos y la red resultante se usa para generar una predicción $p(y x, w_1)$ para un ejemplo x . Se puede hacer lo mismo con las muestras w_2 (azul) y w_3 (verde), lo que arroja predicciones $p(y x, w_2)$ y $p(y x, w_3)$, respectivamente. Las tres redes se tratan como un conjunto y se promedian sus predicciones. Tomada de [McClure et al., 2019]	20
3.	Matriz de Confusión	30
4.	Curva ROC	31
5.	Métricas evaluadas en el primer grupo de modelos - Set de Test	39
6.	Métricas evaluadas en una red neuronal con ventana W1 vs su contraparte con Data Augmentation del paquete volumetractions-3D. V1 corresponde al modelo inicial, V2 al modelo en donde se implementa <i>Data Augmentation</i>	41
7.	Contaste de métricas evaluadas en los modelos deterministas antes y después de optimizar hiper-parámetros - Set de Test	44
8.	Contaste de métricas evaluadas en todos los modelos deterministas - Set de Test	46
9.	Contaste de métricas evaluadas en los modelos Bayesianos - Set de Test	49
10.	Diagrama de Confiabilidad e Histograma - Set de Test	51
11.	Análisis Incertidumbre - Imagen 1: CT-1	52
12.	Análisis Incertidumbre - Imagen 2: CT-1	53
13.	Análisis Incertidumbre - Imagen 3: CT-4	54
14.	Análisis Incertidumbre - Imagen 4: CT-4	55
15.	Análisis Incertidumbre - Imagen 5: CT-0	56
16.	Análisis Incertidumbre - Imagen 6: CT-0	57
17.	Contaste de métricas evaluadas en los modelos para tomografías 2D - Set de Test	58
18.	Evolución Red Neuronal Bayesiana - Opción 1A	69
19.	Evolución Red Neuronal Bayesiana - Opción 1B	69
20.	Evolución Red Neuronal Bayesiana - Opción 2A	69
21.	Evolución Red Neuronal Bayesiana - Opción 2B	69
22.	Contaste de métricas evaluadas en los modelos Bayesianos - Set de Test	71
23.	Contaste de métricas evaluadas en los modelos Bayesianos - Set de Test	72
24.	Contaste de métricas evaluadas en los modelos Bayesianos - Set de Test	73
25.	Contaste de métricas evaluadas en los modelos Bayesianos - Set de Test	75
26.	Diagramas Confiabilidad - Set de Test	77

Índice de cuadros

1.	Ventanas HU para procesamiento de Tomografías Computarizadas	34
----	--	----

2.	Métricas Red Neuronal Determinista - Ventana W4	43
3.	Métricas Red Neuronal Determinista - Ventana W4 - Configuración hiperparámetros según Keras-Tuner	43
4.	Descripción Modelos 3D	67
5.	Descripción Modelos 2D	68

Información General del Proyecto

- Modalidad:

Profundización

- Duración del proyecto (meses): 8 Meses

- Nombre del estudiante: Juan Manuel Liscano Fierro

Programa de maestría al que pertenece: Maestría en Ciencia de Datos

- Nombre del director del trabajo de grado: Hector Javier Hortua Orjuela

Dedicación del director (horas-semana): 2.5 horas por semana.

1. Introducción

En el panorama dinámico de la investigación médica y la inteligencia artificial (IA), la intersección del aprendizaje profundo y las imágenes clínicas se ha convertido en un ámbito fundamental para mejorar las capacidades de diagnóstico. Este proyecto de investigación busca aprovechar las bondades de las redes neuronales tradicionales y de las redes neuronales bayesianas en la clasificación de la neumonía por COVID-19 en tomografías computarizadas 3D y 2D. El objetivo no es sólo mejorar la precisión del diagnóstico sino también explorar el papel vital de la cuantificación de la incertidumbre en el perfeccionamiento de los procesos de toma de decisiones clínicas.

Las redes neuronales, con su capacidad para aprender patrones complejos a partir de vastos conjuntos de datos, han demostrado un éxito notable en diversos ámbitos, incluido el análisis de imágenes médicas. Sin embargo, el paradigma convencional de las redes neuronales a menudo carece de una comprensión matizada de la incertidumbre, un elemento crucial en el diagnóstico médico. Por el contrario, las redes neuronales bayesianas introducen un marco probabilístico que no sólo proporciona predicciones sino que también cuantifica la incertidumbre asociada con esas predicciones, un elemento crítico que a menudo se pasa por alto en los enfoques tradicionales de aprendizaje profundo. Este proyecto busca explorar la eficacia comparativa de las redes neuronales tradicionales y las redes neuronales bayesianas en el contexto de la clasificación de la neumonía COVID-19.

El conjunto de datos es una recopilación de tomografías computarizadas de pulmón humano anonimizadas, que entrelaza la complejidad de las manifestaciones del COVID-19 y el potencial de la tecnología de inteligencia artificial de vanguardia. Estas exploraciones, obtenidas en hospitales médicos de Moscú, Rusia, durante el período comprendido entre el 1 de marzo y el 25 de abril de 2020, capturan una instantánea del impacto de la pandemia en la salud pulmonar.

Este conjunto de datos, que comprende 424 estudios, cada uno de los cuales corresponde a un paciente único, constituye un ejercicio de clasificación binaria. La tarea que se desarrolla es distinguir entre tejido pulmonar normal (CT-0) y aquellos con diversos grados de afectación de la neumonía por COVID-19 (CT-2 y CT-3).

La decisión de emplear redes neuronales bayesianas está motivada por el reconocimiento de que la incertidumbre es una faceta inherente de los diagnósticos médicos. No se trata simplemente de hacer predicciones; se trata de comprender la confianza y confiabilidad de esas predicciones. Este proyecto busca destacar la importancia de la incertidumbre en el ámbito de la clasificación de la neumonía COVID-19. Al comparar el enfoque bayesiano con las redes neuronales tradicionales, pretendemos resaltar los beneficios potenciales de un modelo más interpretable, uno que no solo ofrezca predicciones sino que también proporcione información sobre la certeza y las limitaciones de esas predicciones.

Además, el proyecto incluye un meticuloso proceso de preprocesamiento, que contiene la estandarización de cortes de imágenes, la normalización del valor de los píxeles y la aplicación de técnicas de aumento de datos. Estos pasos son esenciales no sólo para optimizar el entrenamiento del modelo sino también para garantizar la coherencia y confiabilidad del análisis.

1.1. Planteamiento del problema

¿Cómo se compara la implementación de las redes neuronales bayesianas con las redes neuronales determinísticas en su capacidad para clasificar con precisión la neumonía por COVID-19 en tomografías computarizadas 3D y 2D, y cuál es la importancia de cuantificar la incertidumbre asociada a través del enfoque bayesiano para la toma de decisiones clínicas?

informadas?

La pandemia de COVID-19 atrajo nuevamente la atención de la ciencia sobre el campo de análisis de imágenes médicas, particularmente en el contexto de imágenes pulmonares. Las tomografías computarizadas de tórax se han utilizado ampliamente para ayudar en el diagnóstico y tratamiento de la neumonía por COVID-19, que es una complicación clave de la enfermedad. Sin embargo, clasificar con precisión la neumonía por COVID-19 en las tomografías computarizadas sigue siendo un desafío, particularmente cuando se distingue de otros tipos de neumonía o patologías pulmonares.

Las técnicas de aprendizaje profundo (Deep Learning), incluidas las redes neuronales, se han mostrado muy prometedoras en el análisis de imágenes médicas, tales como la detección de hemorragias intracerebrales como en [Sharrock et al., 2021] y [Chang et al., 2018], incluso en el diagnóstico de la neumonía y afecciones pulmonares por COVID-19 [Chen et al., 2021]. Las redes neuronales determinísticas se han utilizado para clasificar la neumonía por COVID-19 en tomografías computarizadas con buenos resultados, pero no tienen en cuenta la incertidumbre inherente a las imágenes médicas y el diagnóstico. Las redes neuronales bayesianas, por otro lado, pueden proporcionar una interpretación probabilística de sus predicciones, lo que puede ser útil para los médicos en los procesos de toma de decisiones.

El uso de técnicas de aprendizaje profundo en imágenes médicas se ha vuelto cada vez más importante en los últimos años, ya que tienen el potencial de mejorar la precisión del diagnóstico, reducir la variabilidad entre observadores y mejorar la eficiencia de los flujos de trabajo clínicos como en [Lundervold and Lundervold, 2019] y [Shen et al., 2017]. Por ejemplo, un estudio de [McKinney et al., 2020] demostró que los algoritmos de aprendizaje profundo podían detectar con precisión el cáncer de seno a través de imágenes, con un rendimiento similar al de los radiólogos experimentados. En otro estudio, [Ardila et al., 2019] demostró que los algoritmos de aprendizaje profundo pueden detectar con precisión el cáncer de pulmón en tomografías computarizadas, donde habla del potencial en los modelos de aprendizaje profundo para aumentar la precisión y la adopción de las pruebas de detección del cáncer de pulmón en todo el mundo. Creando una oportunidad para optimizar el proceso de selección a través de la asistencia informática y la automatización.

Existen varias técnicas diferentes que se pueden utilizar en el análisis de imágenes médicas, incluidas las redes neuronales convolucionales, las redes neuronales recurrentes y los modelos generativos. Cada una de estas técnicas tiene sus propias fortalezas y debilidades, y la elección de la técnica dependerá de la aplicación y el conjunto de datos específicos. Por ejemplo, las redes neuronales convolucionales son particularmente efectivas en la extracción de características en imágenes 2D o 3D, mientras que las redes neuronales recurrentes son útiles para analizar datos de series temporales.

Las redes neuronales bayesianas representan un enfoque particularmente prometedor en este campo, ya que pueden proporcionar una medida de incertidumbre que falta en las redes neuronales determinísticas. La cuantificación de la incertidumbre es esencial en la toma de decisiones clínicas, ya que permite a los médicos emitir juicios informados sobre la fiabilidad de las predicciones diagnósticas y ajustar los planes de tratamiento en consecuencia.

Medir y calibrar las incertidumbres del modelo es fundamental en la toma de decisiones clínicas, ya que estas proporcionan información valiosa sobre la fiabilidad y confianza de las predicciones, identificando áreas en las que las predicciones del modelo pueden ser poco fiables o inconsistentes, y así mismo permitiendo comprender mejor las limitaciones del modelo y los posibles errores. Esto es especialmente importante en el caso de la neumonía por COVID-19, donde la clasificación errónea puede tener graves consecuencias para el tratamiento del paciente. Al mejorar la precisión y confiabilidad de los modelos de clasificación, podemos mejorar la calidad de la atención y respaldar la toma de decisiones clínicas. Esta información es

crucial para que los médicos y radiólogos tomen decisiones informadas sobre el diagnóstico del paciente.

Comprender la incertidumbre asociada con las predicciones ayuda a identificar casos en los que el modelo sea más incierto y que pueda ser necesaria una mayor investigación o una segunda opinión, asegurando que los pacientes reciban la atención adecuada y minimizando el riesgo de diagnóstico erróneo, promoviendo un enfoque más transparente y responsable para utilizar modelos de aprendizaje profundo en entornos clínicos.

Al comparar las redes neuronales bayesianas con las redes neuronales tradicionales o determinísticas, podemos valorar los avances y los beneficios potenciales que aportan los enfoques bayesianos al campo del análisis de imágenes médicas. Esta investigación tiene el potencial de contribuir al desarrollo de modelos de diagnóstico más confiables.

En conclusión, el uso de redes neuronales bayesianas en imágenes médicas representa una vía prometedora para mejorar la precisión y confiabilidad de las predicciones diagnósticas, particularmente en el contexto de la clasificación de neumonía por COVID-19 en tomografías computarizadas 3D y 2D. Al proporcionar una medida de incertidumbre, las redes neuronales bayesianas pueden mejorar la toma de decisiones clínicas y facilitar el desarrollo de planes de tratamiento más personalizados. Por ello se necesita más investigación en esta área para evaluar el rendimiento de las redes neuronales bayesianas en comparación con las redes neuronales determinísticas y para evaluar su impacto potencial en la práctica clínica.

1.2. Objetivos

1.2.1. General

- Evaluar el impacto de la cuantificación de la incertidumbre proporcionada por las redes neuronales bayesianas en la clasificación de la neumonía por COVID-19 en tomografías computarizadas 3D y 2D.

1.2.2. Específicos

- Entrenar redes neuronales determinísticas y bayesianas optimizadas para detectar neumonía por COVID-19 en tomografías computarizadas 2D y 3D a través del paquete TensorFlow.
- Incorporar una metodología adecuada para la calibración y cuantificación de la incertidumbre aleatoria y epistémica en diferentes redes neuronales bayesianas asociadas a la tarea de clasificación.
- Estudiar y evaluar el desempeño de las redes neuronales determinísticas y bayesianas en términos de precisión, sensibilidad, especificidad y área bajo la curva ROC (AUC-ROC).
- Implementar métodos para la calibración de las incertidumbres encontradas en cada uno de los modelos estocásticos.

1.3. Alcance y Limitaciones

1.3.1. Alcance

- Este proyecto de investigación se centra en la implementación y comparación de redes neuronales tradicionales y redes neuronales bayesianas para la clasificación de la neumonía por COVID-19 en tomografías computarizadas 3D y 2D. El modelo está diseñado

específicamente para clasificar anomalías CT-0 como indicativas de ausencia de neumonía, y combina las etiquetas CT-2 y CT-3 como indicativas de la presencia de manifestaciones características de neumonía por COVID-19. Este enfoque específico tiene como objetivo optimizar el rendimiento del modelo para escenarios de clasificación clínicamente relevantes. El alcance incluye además el preprocesamiento del conjunto de datos, el desarrollo de modelos, el entrenamiento y la evaluación. Por último, el proyecto busca explorar la importancia de la cuantificación de la incertidumbre proporcionada por los enfoques bayesianos en la toma de decisiones clínicas.

1.3.2. Limitaciones

- Capacidad de procesamiento: Los recursos computacionales disponibles, particularmente la capacidad de procesamiento en la máquina virtual, imponen limitaciones a la escalabilidad de ciertos aspectos del proyecto. Específicamente, las limitaciones de capacidad restringen la expansión de los tamaños de lote (*batches*) durante el entrenamiento, lo que podría afectar la eficiencia de la convergencia del modelo y la exploración de conjuntos de datos más grandes. Esto a su vez influyó en la decisión de centrarse en un subconjunto de clasificaciones de tomografías computarizadas para garantizar un entrenamiento y evaluación eficientes del modelo.
- Complejidad del modelo: debido a limitaciones de recursos, la complejidad de las arquitecturas de redes neuronales puede ser limitada. Esta limitación podría afectar la profundidad y amplitud de las características aprendidas por los modelos, afectando potencialmente su rendimiento en comparación con arquitecturas más complejas.
- Especificidad del conjunto de datos: el conjunto de datos utilizado en este proyecto de investigación se limita a tomografías computarizadas de pulmón humano anonimizadas con hallazgos relacionados con COVID-19, obtenidos de hospitales médicos en Moscú, Rusia, durante un período de tiempo específico. Si bien este conjunto de datos proporciona información valiosa sobre la clasificación de la neumonía por COVID-19, es posible que su alcance no capture completamente la diversidad de casos observados a nivel mundial o en diferentes entornos de atención médica.
- Relevancia clínica: La decisión de priorizar las clasificaciones CT-0, CT-2 y CT-3 se alinea con la importancia clínica de distinguir entre ausencia y presencia de manifestaciones características de neumonía en pacientes con COVID-19. Las clasificaciones CT-1 y CT-4, si bien son importantes, pueden representar escenarios relativamente raros o atípicos que son menos prevalentes en el conjunto de datos o menos relevantes clínicamente para la tarea de diagnóstico específica en cuestión.
- Generalizabilidad: Los hallazgos y conclusiones extraídas de esta investigación pueden ser específicas del conjunto de datos y la configuración experimental utilizada. La generalización de los resultados a poblaciones más amplias o conjuntos de datos con diferentes características debe hacerse con cautela, considerando posibles variaciones en la demografía de los pacientes, los protocolos de imágenes y las manifestaciones de la enfermedad.
- Interpretabilidad: si bien las redes neuronales bayesianas ofrecen una cuantificación mejorada de la incertidumbre, la interpretabilidad de las predicciones de los modelos aún puede plantear desafíos. Comprender los factores subyacentes que impulsan las decisiones sobre modelos, particularmente en tareas complejas de imágenes médicas, sigue siendo un área de investigación en curso.

- Factores externos: los factores externos, como las variaciones en la calidad de la imagen, los artefactos y las anotaciones, pueden introducir ruido o sesgos en el análisis, lo que podría influir en el rendimiento y la interpretabilidad de los modelos.
- Consideraciones éticas: Durante todo el proceso de investigación se deben reconocer y abordar las consideraciones éticas relacionadas con el uso responsable de la IA en la atención sanitaria, incluidas las cuestiones de sesgo, equidad y transparencia.

2. Fundamentación Teórica

2.1. Redes Neuronales Profundas

En los últimos años, las redes neuronales profundas (DNNs, por sus siglas en inglés) se han convertido en modelos de alto rendimiento. Estas redes se componen de múltiples capas de nodos de procesamiento o neuronas interconectadas. Con esa arquitectura son capaces de aprender representaciones complejas de datos de entrada a través de un proceso conocido como entrenamiento, que implica ajustar los pesos de las conexiones entre las neuronas para minimizar el error entre la salida predicha y la salida real. Estas neuronas procesan los datos de entrada de una manera particular; la primera capa de una DNN es la capa de entrada, que toma los datos de entrada sin procesar y los transforma en un conjunto de características que luego procesan las capas posteriores. La salida de la capa final es la salida prevista de la red, es decir, la cantidad de neuronas va a depender del objetivo. El proceso de entrenamiento generalmente se lleva a cabo utilizando el algoritmo de descenso de gradiente estocástico (SGD, por sus siglas en inglés), donde los pesos se actualizan en pequeños incrementos en función del gradiente de error calculado con respecto a un conjunto de entrenamiento. Este proceso se repite para varias sub-muestras durante el entrenamiento hasta que el promedio de la función objetivo deja de decrecer y se le llama estocástico porque cada una de estas sub-muestras da una estimación ruidosa del promedio del gradiente de todos los ejemplos [LeCun et al., 2015]. Luego del entrenamiento, el modelo se evalúa en un set de datos diferente que comúnmente se denomina set de validación o testeo, y así finalmente se evalúa la habilidad del modelo para generalizar las predicciones.

Las DNN se han convertido en la técnica de vanguardia para una amplia gama de aplicaciones, incluido el reconocimiento de voz, detección de objetos en imágenes, entre otros casos como lo mencionan en [LeCun et al., 2015]. Uno de los factores clave que han contribuido a su éxito es la disponibilidad de conjuntos de datos a gran escala y potentes recursos informáticos que permiten el entrenamiento de arquitecturas profundas con millones de parámetros.

Matemáticamente, en un vector de entrada $x = [x_1, x_2, x_3, \dots, x_d]$ de dimensión d , la operación realizada por cada neurona se representa por

$$f\left(\sum_{i=1}^d W_i x_i + b\right), \quad (1)$$

donde W_i es el vector que contiene los pesos de cada neurona y multiplica con cada una de las entradas en x , b corresponde al término de sesgo, un parámetro único que permite ajustar la salida junto con la suma ponderada, estos se conocen como *learnable parameters*, y por último $f()$ corresponde a la función de activación.

En resumen, las DNN son una clase potente y versátil de modelos de aprendizaje automático que han revolucionado muchos campos de la investigación y la industria. Su éxito se debe a

su capacidad para aprender representaciones complejas de datos de entrada y su escalabilidad a grandes conjuntos de datos y espacios de entrada de gran dimensión.

2.1.1. Redes Neuronales Convolucionales

Las redes neuronales convolucionales (CNN, por sus siglas en inglés) son un tipo especializado de DNN que han sido particularmente exitosas en tareas de reconocimiento de imágenes dentro del campo de la salud, como en la detección de cáncer de hueso conocido como osteosarcoma [Gawade et al., 2023] y otros estudios importantes como la detección de la enfermedad de Alzheimer trabajado en [de Silva and Kunz, 2023]. La innovación clave de las CNN es el uso de capas convolucionales, que aplican un conjunto de filtros de aprendizaje a la imagen de entrada para extraer características locales como bordes y texturas. Estos filtros son generalmente de un tamaño pequeño (ej. 3x3 o 5x5) y se aplican a la imagen de entrada en una forma de ventana que se desliza por toda la imagen como lo ilustra la sección de extracción de características (*Feature Extraction*) en la Fig.(1). La salida de la capa convolucional luego pasa a través de una función de activación no lineal, como por ejemplo la unidad lineal rectificada (ReLU), para introducir no linealidades en el modelo.

Estas capas convolucionales tienen varias ventajas importantes sobre las capas totalmente conectadas, que se utilizan en las redes neuronales. En primer lugar, las capas convolucionales tienen menos parámetros que las capas completamente conectadas, lo que reduce el riesgo de sobreajuste y permite el entrenamiento de arquitecturas más profundas. En segundo lugar, las capas convolucionales aprovechan la estructura espacial de la imagen de entrada, lo que permite la detección de características invariantes a la traslación. Esto significa que un filtro que detecta una característica particular en una parte de la imagen también puede detectar la misma característica en otra parte de la imagen, incluso si se ha trasladado o girado. Además de las capas convolucionales, las CNN también suelen incluir capas de agrupación (*pooling*), que se utilizan para reducir la dimensionalidad espacial de los mapas de características producidos por las capas convolucionales y son útiles para aumentar la eficiencia de la red. Las capas de agrupación agregan las activaciones de los elementos vecinos en el mapa de características, lo que reduce la cantidad de parámetros y ayuda a reducir el sobreajuste. Los tipos más comunes de agrupación son la agrupación máxima y promedio, que toma el valor de activación máximo o promedio en una vecindad local del mapa de características.

En resumen, las capas convolucionales son un bloque de construcción fundamental de las redes neuronales convolucionales y son esenciales para la extracción efectiva de características en las tareas de reconocimiento de imágenes. Aprovechan la estructura espacial de la imagen de entrada y tienen menos parámetros que las capas completamente conectadas, lo que las hace muy adecuadas para arquitecturas profundas y reduce el riesgo de sobreajuste.

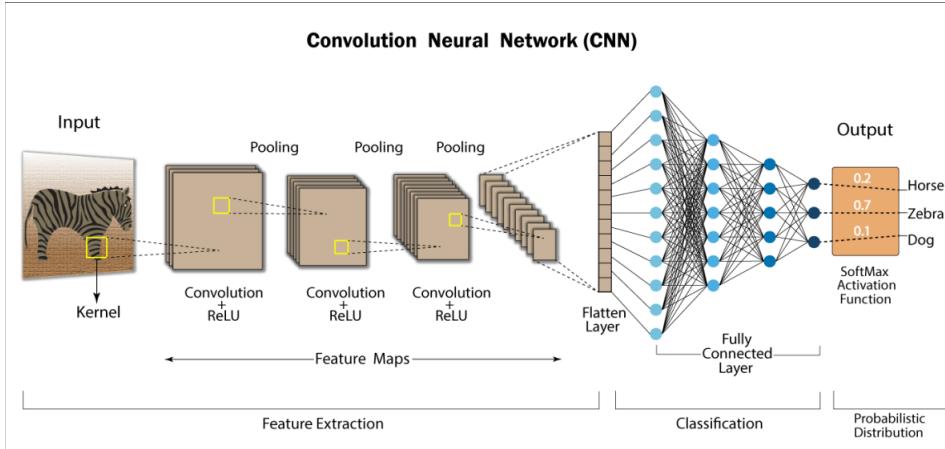


Figura 1: Arquitectura de una RNN compuesta por capas convolucionales. Tomada de developersbreach

Estas arquitecturas convolucionales son muy comunes en CNN como AlexNet [Krizhevsky, 2014], VGG Net [Simonyan and Zisserman, 2014], y ResNet [He et al., 2016].

2.1.2. Capas de Normalización y Pooling en CNN

En las redes neuronales convolucionales (CNN), las capas pooling desempeñan un papel crucial en la reducción de las dimensiones espaciales de los mapas de características¹ generados por las capas convolucionales, reduciendo así la complejidad computacional y controlando el sobreajuste. Además, con frecuencia se emplean capas de normalización por lotes para acelerar el proceso de entrenamiento y mejorar la estabilidad de la red.

- Capas GlobalAveragePooling3D y 2D: La agrupación promedio global es una técnica que se utiliza para reducir la muestra de los mapas de características obtenidos de capas convolucionales. En esta capa, cada mapa de características se promedia en todas sus dimensiones espaciales, lo que da como resultado un valor único por mapa de características. Estas capas ayudan a reducir las dimensiones espaciales de los mapas de características y al mismo tiempo retiene información importante. Es particularmente útil para reducir el sobreajuste al resumir todo el mapa de características en un solo valor.
- GlobalMaxPooling3D y 2D: De manera similar a la agrupación promedio global, la agrupación máxima global es otra técnica de reducción de resolución utilizada en las CNN. En esta capa, el valor máximo dentro de cada mapa de características se extrae en todas las dimensiones espaciales. La agrupación máxima global resalta las características más importantes dentro de cada mapa de características y descarta información irrelevante.
- MaxPool3D y 2D: La agrupación máxima es una técnica de reducción de resolución local que divide los mapas de características de entrada en regiones que no se superponen y retiene solo el valor máximo dentro de cada región. Esta capa ayuda a reducir las dimensiones espaciales de los mapas de características y al mismo tiempo preserva las características más dominantes.

¹Los mapas de características, también conocidos como mapas de activación, son la salida de capas convolucionales en una red neuronal convolucional (CNN).

- BatchNormalization: La normalización por lotes es una técnica utilizada para normalizar las activaciones de cada capa en una red neuronal ajustando y escalando las activaciones para que tengan media cero y varianza unitaria. Esta capa ayuda a estabilizar y acelerar el proceso de entrenamiento al reducir el cambio de covariables interno. Al normalizar las activaciones, la normalización por lotes permite una convergencia más fluida y rápida durante el entrenamiento, lo que conduce a un mejor rendimiento y a la generalización del modelo.

2.2. Redes Neuronales Bayesianas

Las redes neuronales bayesianas (BNN, por sus siglas en inglés) son un tipo de red neuronal que incorpora la teoría de la probabilidad bayesiana en el proceso de entrenamiento e inferencia. A diferencia de las redes neuronales deterministas, que generan una predicción fija dado un conjunto de entradas, las BNN generan una distribución de probabilidad sobre posibles predicciones (Fig(2) ilustra de forma general cómo se produce una estimación bajo una red neuronal bayesiana). Esto permite modelar la incertidumbre epistémica, que es la incertidumbre asociada a los parámetros del modelo (esto puede imaginarse como la dispersión de la distribución a posteriori de los pesos $p(w|D)$, en la que una distribución posterior más plana refleja una mayor incertidumbre epistémica, mientras que una distribución posterior con picos refleja una menor incertidumbre epistémica), y la incertidumbre aleatoria, que es la aleatoriedad inherente a los datos, en donde dados unos datos de entrada y los parámetros de pesos fijos, una incertidumbre elevada significa que tenemos una estimación volátil o ruidosa de su predicción (para la regresión) o no sabemos a qué clase pertenece (para la clasificación). Una incertidumbre aleatoria alta sugiere que no tenemos suficiente información para predecir el valor de salida para una entrada con parámetros de peso fijos, debido a variables no observadas o latentes que el modelo no puede capturar [Chai, 2018].

[Chai, 2018] habla sobre la importancia de descomponer la incertidumbre predictiva argumentando que las incertidumbres aleatorias y epistémicas nos hablan sobre las diferentes facetas de un valor de entrada (input). Una incertidumbre epistémica alta sugiere que el valor de entrada es un valor atípico en relación con la distribución de entrenamiento, y se puede reducir recomiendo más datos de entrenamiento cerca de la región de evaluación, de tal forma que en el caso de contar con datos infinitos, esta incertidumbre se reduzca a cero (es decir, si conocemos todos los datos del universo, confiamos en la función que asigna la entrada a la salida, y por tanto la distribución de pesos se convierte en un pico), pero más datos no ayudan a la incertidumbre aleatoria; para reducirla, se necesitan conocimientos sobre variables no observadas a través de características adicionales o mediciones más refinadas. En la práctica, estas mediciones no suelen estar disponibles, por lo que no siempre es posible reducir la incertidumbre aleatoria.

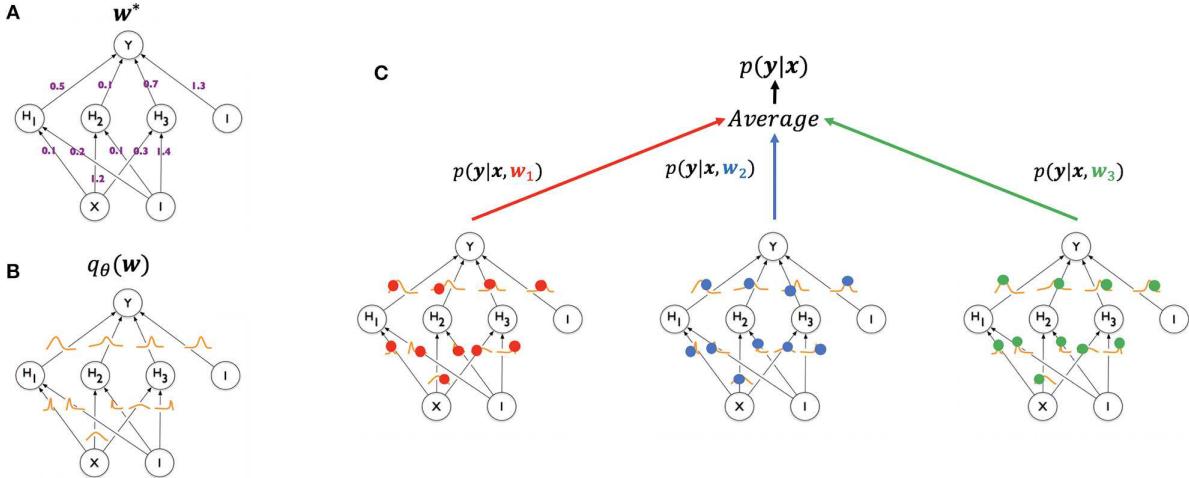


Figura 2: Ilustración de la generación de una predicción a partir de una red neuronal bayesiana mediante el muestreo de Monte Carlo. Una red neuronal estándar (A, arriba a la izquierda) tiene un peso para cada una de sus conexiones (w^*), ajustadas a través del conjunto de entrenamiento y utilizadas para generar una predicción para un ejemplo de prueba. Una red neuronal bayesiana (B, abajo a la izquierda) tiene, en cambio, una distribución posterior para cada peso, parametrizada por θ ($q_\theta(w)$). El proceso de entrenamiento comienza con una distribución previa asignada para cada peso y devuelve una distribución posterior aproximada. En el momento de la prueba (C, derecha), se extrae una muestra w_1 (rojo) de la distribución a posteriori de los pesos y la red resultante se usa para generar una predicción $p(y|x, w_1)$ para un ejemplo x . Se puede hacer lo mismo con las muestras w_2 (azul) y w_3 (verde), lo que arroja predicciones $p(y|x, w_2)$ y $p(y|x, w_3)$, respectivamente. Las tres redes se tratan como un conjunto y se promedian sus predicciones. Tomada de [McClure et al., 2019]

Los BNN usan distribuciones a priori sobre los pesos de la red y aprenden distribuciones a posteriori $p(w|D)$ usando la regla de Bayes: $p(w|D) \sim p(D|w)p(w)$, donde $p(D|w)$ denota la función de verosimilitud que representa la probabilidad de los datos observados D dado los pesos w , y $p(w)$ que representa la distribución a priori de los pesos. La distribución a posteriori representa la incertidumbre en los pesos dados los datos observados $D = (X, Y)$ normalmente se aproxima mediante métodos de inferencia variacional [Chai, 2018, Heek, 2018] o cadena de Markov Monte Carlo (MCMC) [Neal, 2012]. Luego, la distribución a posteriori se usa para hacer predicciones tomando la esperanza del output sobre la distribución. Una vez realizado el cálculo de la distribución a posteriori, la distribución de probabilidad de un ejemplo nuevo de prueba x^* puede determinarse por

$$p(y^*|x^*, D) = \int_w p(y^*|x^*, w)p(w|D)dw, \quad (2)$$

donde $p(y^*|x^*, w)$ es la distribución predictiva posterior correspondiente al set de pesos. Una ventaja de las BNN es que pueden proporcionar predicciones más sólidas y confiables al tener en cuenta la incertidumbre en los datos y el modelo. Esto es particularmente útil en aplicaciones donde las predicciones incorrectas o inciertas pueden tener consecuencias graves, como diagnósticos médicos o pronósticos financieros. Otra ventaja es que las BNN pueden proporcionar una medida de incertidumbre para cada predicción, que se puede usar para guiar la toma de decisiones o para identificar áreas donde se necesitan más datos o refinamiento del modelo.

Sin embargo, también existen algunos desafíos asociados con el uso de BNN. Un desafío es que pueden ser computacionalmente costosos para entrenar y evaluar. Otro desafío es que pueden ser más difíciles de interpretar en términos matemáticos que las redes neuronales determinísticas, ya que el resultado es una distribución de probabilidad en lugar de una sola predicción. Además, elegir distribuciones previas y métodos de inferencia apropiados puede ser un desafío y puede tener un impacto significativo en el rendimiento del modelo.

En resumen, las redes neuronales bayesianas son un enfoque prometedor para modelar la incertidumbre en las redes neuronales y pueden brindar predicciones más sólidas y confiables en ciertas aplicaciones. Sin embargo, también presentan algunos desafíos y requieren una cuidadosa consideración de las distribuciones a priori y los métodos de inferencia apropiados.

2.2.1. Distribución de pesos w a posteriori

Dado un set de datos de entrenamiento con tamaño N con entradas x y salidas y denominado $D = \{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(N)}, y^{(N)})\}$, una red neuronal bayesiana (BNN) puede ajustar la distribución a posteriori sobre los pesos de la red, $p(w|D)$. Como se ha mencionado, w consolida todos los pesos sobre las L capas en la red: $w = \{w_l\}_{l=1}^L$. Esta distribución representa qué tan probable es un determinado set de pesos después de ver los datos de entrenamiento, en vez de una estimación puntual de los pesos en las DNNs. Una aplicación directa de la regla de bayes produce

$$p(w|D) = \frac{p(D|w)p(w)}{p(D)}, \quad (3)$$

en el numerador, el primer término $p(D|w)$ refleja la verosimilitud de los datos de entrenamiento dado un determinado set de pesos w . Asumiendo que cada punto de datos de entrenamiento es independiente, esta cantidad se convierte en el producto de verosimilitudes para cada punto de entrenamiento individual

$$p(D|w) = \prod_{n=1}^N p(y^{(n)}|w, x^{(n)}). \quad (4)$$

Las tareas de clasificación pueden usar las predicciones softmax de la red neuronal directamente como la verosimilitud: $p(y = c|w, x) = f^w(x)^c$, donde c es la verdadera clase del input, y las predicciones softmax para cada clase $f^w(x)^c$ están normalizadas para ser mayores a 0 y sumar 1 [Chai, 2018].

El segundo término del numerador, $p(w)$ es la distribución a priori sobre los pesos. Refleja una creencia sobre la distribución de los pesos w sin ver ningún dato. Un ejemplo de distribución a priori es la distribución normal. Los dos términos del numerador son tratables al momento de calcular un determinado set de w . Sin embargo, el problema de lograr la distribución posterior es el denominador $p(D)$. Calcular $p(D)$ implica marginar todas las configuraciones de los pesos

$$p(D) = \int p(D|w)p(w)dw. \quad (5)$$

En muchos modelos, esta integral es intratable, motivando la necesidad de aproximaciones de la a posteriori. Métodos de muestreo, como Metropolis Hastings o Hamiltonian Monte Carlo, producen estimaciones no sesgadas de la verdadera a posteriori, pero pueden tardar en converger. Un enfoque alternativo es la inferencia variacional, en la que la verdadera distribución a posteriori se aproxima con una distribución variacional más simple [Blei et al., 2017, Jordan et al., 1999]. Esta aproximación está sesgada, pero suele ser más

rápida que los métodos de muestreo. En [Chai, 2018], usan una distribución variacional para aproximar la distribución a posteriori de los pesos de las BNN.

2.2.2. Inferencia variacional

La inferencia variacional se aproxima a la compleja distribución a posteriori $p(w|D)$ con una distribución manejable más simple sobre los pesos del modelo, $q(w)$, con parámetros variacionales v . Estos parámetros variacionales v son ajustados de modo que $q(w)$ se aproxime a la distribución posterior deseada $p(w|D)$. Esta distribución variacional ajustada se usa para realizar las predicciones del modelo en lugar la verdadera distribución.

Una forma de medir la distancia entre las dos funciones de probabilidad $q(x)$ y $p(x)$ es usando la divergencia de Kullback-Leibler o divergencia-KL. Definida como

$$KL(q(x)||p(x)) \equiv \mathbb{E}_{q(x)} \left[\log \frac{q(x)}{p(x)} \right] = \int q(x) \log \frac{q(x)}{p(x)} dx. \quad (6)$$

Para obtener la distribución variacional $q(w)$ cercana a la distribución posterior $p(w|D)$, se requiere minimizar la divergencia-KL entre estas dos distribuciones

$$\begin{aligned} KL(q(w)||p(w|D)) &= \mathbb{E}_{q(w)} \left[\log \frac{q(w)}{p(w|D)} \right] \\ &= \int q(w) \log \frac{q(w)}{p(w|D)} dw \\ &= \int q(w) \log \frac{q(w)p(D)}{p(D|w)p(w)} dw \\ &= \int q(w) \log \frac{q(w)}{p(w)} dw + \int q(w) \log p(D) dw - \int q(w) \log p(D|w) dw \\ &= KL(q(w)||p(w)) + \log p(D) - \mathbb{E}_{q(w)}[\log p(D|w)]. \end{aligned} \quad (7)$$

Debido al término $\log p(D)$, la divergencia-KL no se puede calcular directamente. Sin embargo, reordenando los términos obtenemos

$$\log p(D) = KL(q(w)||p(w|D)) + \mathbb{E}_{q(w)}[\log p(D|w)] - KL(q(w)||p(w)). \quad (8)$$

Dado que $\log p(D)$ es una constante (aunque intratable), minimizar $KL(q(w)||p(w|D))$ es equivalente a maximizar $\mathbb{E}_{q(w)}[\log p(D|w)] - KL(q(w)||p(w))$. Esta última expresión se denomina límite inferior de la evidencia (ELBO, por sus siglas en inglés) y también puede obtenerse a partir del logaritmo de la probabilidad de los datos

$$\begin{aligned} \log p(D) &= \int p(D|w)p(w)dw \\ &= \log \int \frac{q(w)}{q(w)} p(D|w)p(w)dw \\ &\text{aplicando la desigualdad de Jensen:} \\ &\geq \int q(w) \log \frac{p(D|w)p(w)}{q(w)} dw \\ &= - \int q(w) \log \frac{q(w)}{p(w)} dw + \int q(w) \log p(D|w) dw \\ &= KL(q(w)||p(w)) + \mathbb{E}_{q(w)}[\log p(D|w)]. \end{aligned} \quad (9)$$

Con esto tenemos que

$$\log p(D) \geq KL(q(w)||p(w)) + \mathbb{E}_{q(w)}[\log p(D|w)]. \quad (10)$$

A diferencia de Ec.(8), la desigualdad(10) carece del término $KL(q(w)||p(w|D))$. Sin embargo, debido a las propiedades no negativas de la divergencia-KL, se puede concluir que ELBO es menor o igual que el logaritmo de la probabilidad de los datos.

Por lo tanto, encontrar los parámetros variacionales v para minimizar la divergencia-KL es lo mismo que maximizar ELBO. En efecto, la inferencia variacional traduce el problema de la inferencia sobre la distribución de pesos en el problema de optimización de la maximización de la ELBO. Una vez definido el objetivo Una vez definido el objetivo ELBO, podemos tomar muestras de $q(w)$ y utilizar la retropropagación, como en las DNN, para encontrar valores óptimos. DNNs, para encontrar los valores óptimos de los parámetros variacionales que maximizan la ELBO.

2.2.2.1 Divergencias Alternativas

La divergencia-KL es una de las formas que existen para calcular la distancia entre distribuciones. Otra métrica alternativa es la divergencia- α [Amari, 2012, Zhu and Rohwer, 1995, Hortua et al., 2020]

$$D_\alpha(p(x)||q(x)) = \frac{1}{\alpha(1-\alpha)} \left(1 - \int p(x)^\alpha q(x)^{1-\alpha} dx \right). \quad (11)$$

Usando las mismas distribuciones $q(w)$ y $p(w|D)$ que antes, como $\lim_{\alpha \rightarrow 0}$ recuperamos $KL(q(w)||p(w|D))$ usado en la inferencia variacional, pero esta divergencia-KL lleva a que $q(w)$ sea 0 en todas partes donde $p(w|D)$ es 0. En consecuencia, una distribución variacional ajustada con los criterios de divergencia-KL tiende a subestimar la incertidumbre porque se ajusta a un modo local de la posterior [Li and Gal, 2017, Hernandez-Lobato et al., 2016].

Una de las motivaciones para usar la divergencia- α es para obtener mejores estimaciones de la incertidumbre relajando esta restricción. Un valor pequeño de α favorece que la distribución variacional se ajuste al modo más alto de la distribución posterior, y un valor grande de α favorece que la distribución variacional cubra toda la masa de la distribución a posteriori [Hernandez-Lobato et al., 2016]. Cuando $\alpha = 0.5$, la divergencia es simétrica tal que $D_{0.5}(p||q) = D_{0.5}(q||p)$. La divergencia- α se ha aplicado a BNNs tanto en contextos de regresión como de clasificación, observándose que el valor intermedio de $\alpha = 0.5$ produce mejores predicciones que $\alpha = 0$ utilizado en inferencia variacional y $\alpha = 1$ utilizado en propagación de expectativas [Li and Gal, 2017, Hernandez-Lobato et al., 2016, Depeweg et al., 2016].

2.2.2.2 Distribuciones Variacionales

La esencia de la distribución variacional es que es lo suficientemente similar a la distribución de pesos a posteriori después de minimizar la divergencia, pero es más sencilla de extraer muestras. Una forma común de distribución variacional es la aproximación de campo medio (mean-field), en la que asumimos que la distribución variacional se factoriza en el producto de distribuciones al tratar los pesos como variables independientes. Además, podemos utilizar una distribución gaussiana como distribución variacional, lo que permite muestrear más fácilmente a partir de una distribución normal en lugar del peso exacto a posteriori [Chai, 2018, Garcia-Farieta et al., 2024, Louizos and Welling, 2017a, Hortúa et al., 2023]. En este caso, la distribución variacional se convierte en

$$q(w|\theta) = \prod_{ij} \mathcal{N}(w; \mu_{ij}, \sigma_{ij}^2), \quad (12)$$

donde i y j son los índices de las neuronas de las capas anterior y actual, respectivamente. Aplicando el truco de reparametrización, obtuvimos $w_{ij} = \mu_{ij} + \sigma_{ij} * \epsilon_{ij}$, donde ϵ_{ij} se extrae de la distribución normal. Además, cuando la prior es una composición de distribuciones gaussianas independientes, la divergencia KL entre el prior y el posterior variacional se puede calcular analíticamente. Esta característica mejora la eficiencia informática de este enfoque [Garcia-Farieta et al., 2024].

2.2.2.3 Técnicas de perturbación a los pesos

Las perturbaciones de los pesos se refieren al proceso de introducir pequeñas variaciones o perturbaciones en los pesos de una red neuronal durante el entrenamiento. Estas perturbaciones suelen aplicarse a los pesos de la red para introducir estocasticidad y aleatoriedad en el proceso de aprendizaje. Al perturbar los pesos, la red se ve obligada a explorar diferentes regiones del espacio de parámetros, lo que puede ayudar a evitar el sobreajuste y mejorar la generalización.

Existen varias formas de implementar las perturbaciones de los pesos explicadas por [Wen et al., 2018], entre las que se incluyen:

- Perturbaciones gaussianas: Añadir ruido a los pesos extraídos de una distribución gaussiana.
 - Perturbaciones aditivas: Añadiendo pequeños valores aleatorios a los pesos de la red.
 - Perturbaciones multiplicativas: Escalando los pesos por pequeños factores aleatorios.
- DropConnect: Poner a cero aleatoriamente un subconjunto de pesos durante el entrenamiento.

Las perturbaciones de peso se utilizan habitualmente en las técnicas de regularización para evitar que la red memorice los datos de entrenamiento y fomentar la robustez de las representaciones aprendidas. Al introducir la aleatoriedad a través de las perturbaciones de peso, la red se expone a diferentes escenarios de entrenamiento, lo que puede conducir a una mejor generalización y un mejor rendimiento en datos no vistos.

2.2.2.3.1 Flipout

En casos donde el muestreo de $q(w|\theta)$ no sea totalmente independiente para los distintos ejemplos de un minilote, se obtendrán estimaciones del gradiente con una varianza elevada. [Wen et al., 2018] introduce Flipout como un método eficiente para decorrelacionar los gradientes dentro de un minilote mediante el muestreo implícito de perturbaciones de peso pseudoindpendientes para cada ejemplo, reduciendo efectivamente la correlación entre las actualizaciones de peso para diferentes ejemplos. Este enfoque ayuda a conseguir una estimación más precisa del gradiente y reduce la varianza en las estimaciones del gradiente durante el entrenamiento.

Flipout se aplica a cualquier distribución de perturbación que se factorice por peso y sea simétrica alrededor de 0 [Wen et al., 2018, Hortúa et al., 2023] -incluyendo DropConnect, perturbaciones gaussianas multiplicativas, estrategias de evolución y redes neuronales bayesianas variacionales- y a muchas arquitecturas, incluyendo redes totalmente conectadas, redes convolucionales y RNNs.

2.2.2.3.2 Reparametrización

La técnica de reparametrización local [Kingma et al., 2015] ayuda a reducir la varianza de los gradientes estocásticos para la inferencia bayesiana variacional traduciendo la incertidumbre sobre los parámetros globales en ruido local que es independiente de los puntos de datos del minilote. Esta técnica permite transformar la incertidumbre global de los pesos en una forma de incertidumbre local más fácil de muestrear y eficiente desde el punto de vista computacional.

2.2.3. Flujos de normalización multiplicativos (MNF)

Las distribuciones gaussianas de campo medio descritas en la ecuación Ec.(12) son la familia más utilizada para la variación posterior en BNN. Desafortunadamente, esta distribución carece de la capacidad de representar adecuadamente la compleja naturaleza del verdadero posterior. Por lo tanto, se anticipa que mejorar la complejidad de la posterior variacional producirá mejoras sustanciales en el rendimiento. Esto se atribuye a la capacidad de tomar muestras de una distribución más confiable, que se aproxima mucho a la verdadera distribución posterior. El proceso de mejora de la posterior variacional exige métodos computacionales eficientes al tiempo que se garantiza su viabilidad numérica. Se han propuesto flujos de normalización multiplicativos (MNF) para adaptar eficientemente las distribuciones posteriores mediante el uso de variables aleatorias auxiliares y los flujos de normalización. Los flujos de normalización de mezclas (MNF) sugieren que la posterior variacional se puede representar matemáticamente como una mezcla infinita de distribuciones [Garcia-Farieta et al., 2024]

$$q(w|\theta) = \int q(w|z, \theta)q(z|\theta)dz, \quad (13)$$

siendo θ el parámetro posterior estimable, y $z \sim q(z|\theta) \equiv q(z)$ el vector con la misma distribución de la capa de entrada, que desempeña el papel de variable auxiliar latente. Además, al permitir reparametrizaciones locales, la variación posterior para capas completamente conectadas se vuelve

$$w \sim q(w|z) = \prod_{ij} \mathcal{N}(w; z_{ij}\mu_{ij}, \sigma_{ij}^2). \quad (14)$$

Se puede aumentar la flexibilidad del posterior variacional mejorando la complejidad de $q(z)$. Esto se puede hacer utilizando flujos de normalización ya que la dimensionalidad de z es mucho menor que los pesos. A partir de muestras $z_0 \sim q(z_0)$ de gaussianas totalmente factorizadas (ver Ec.(12)), se puede obtener una distribución $q(z_K)$ aplicando sucesivamente transformaciones f_k invertibles,

$$z_k = NF(z_0) = f_k \circ \dots \circ f_1(z_0), \quad (15)$$

$$\log q(z_K) = \log q(z_0) - \sum_{k=1}^K \log \left| \det \frac{\partial f_k}{\partial z_{k-1}} \right| \quad (16)$$

Para manejar la intratabilidad de la posterior, [Louizos and Welling, 2017b] sugirieron utilizar nuevamente la ley de Bayes $q(z_K)q(w|z_K) = q(w)q(z_K|w)$ e introducir una nueva distribución auxiliar $r(z_K|w, \phi)$ parametrizado por ϕ , con el propósito de aproximar la distribución a posteriori de los parámetros variacionales originales $q(z_K|w)$ para reducir aún más el límite del término de divergencia KL. En consecuencia, el término de divergencia KL se puede reescribir de la siguiente manera

$$-KL[q(w) \parallel p(w)] \geq \mathbb{E}_{q(w,z_k)}[-KL[q(w|z_k) \parallel p(w)] + \log q(z_k) + \log r(z_k|w, \phi)]. \quad (17)$$

El primer término se puede calcular analíticamente ya que será la divergencia KL entre dos distribuciones gaussianas, mientras que el segundo término se calcula mediante el flujo normalizador generado por f_K (ver Ec.(16)). Además, el término a posteriori auxiliar está parametrizado mediante flujos de normalización inversa de la siguiente manera

$$z_0 = NF^{-1}(z_k) = g_1^{-1} \circ \cdots \circ g_k^{-1}(z_k), \quad (18)$$

$$\log r(z_K|w, \phi) = \log r(z_0|w, \phi) + \sum_{k=1}^K \log \left| \det \frac{\partial g_k^{-1}}{\partial z_k} \right|, \quad (19)$$

en donde g_k^{-1} puede ser parametrizado como otro flujo normalizador. Un parámetro flexible de parametrización de la posterior auxiliar puede darse por

$$z_0 \sim r(z_k|w, \phi) = \prod_i \mathcal{N}(z_0; \tilde{\mu}_i(w, \phi), \tilde{\sigma}_i^2(w, \phi)), \quad (20)$$

en donde la parametrización de la media $\tilde{\mu}$ y la varianza $\tilde{\sigma}^2$ lo lleva a cabo la máscara RealNVP como la elección de los flujos normalizadores.

2.2.4. Flujos de normalización multiplicativos en una representación de cúbica de vértices

[Garcia-Farieta et al., 2024] presentan el resultado de la generalización de la Ec.(14) hacia capas convolucionales 3D. Iniciando con la extensión de la posterior variacional como

$$w \sim q(w|z) = \prod_i^{D_d} \prod_j^{D_h} \prod_k^{D_w} \prod_l^{D_f} \mathcal{N}(w; z_l \mu_{ijkl}, \sigma_{ijkl}^2), \quad (21)$$

donde D_h , D_w y D_d son las 3 dimensiones espaciales de la cajas, y D_f es el número de filtros para cada kernel. El objetivo consiste en abordar el desafío de mejorar la adaptabilidad de la distribución a posteriori aproximada de los pesos provenientes de una capa convolucional 3D. El algoritmo(1) describe el procedimiento para la propagación hacia adelante de cada capa convolucional 3D [Garcia-Farieta et al., 2024]. De manera similar al caso de las redes totalmente conectadas, el parámetro auxiliar solo afecta la media con el propósito de evitar una gran variación, y mantuvieron un mapeo lineal para parametrizar los flujos de normalización inversa en lugar de aplicar transformaciones de características tanh.

Algorithm 1 Propagación hacia adelante para cada capa convolucional 3D. M_w , Σ_w son las medias y las varianzas para cada capa. H es la capa de entrada y $NF(\cdot)$ es la máscara RealNVP de los flujos normalizados aplicados sobre las muestras inicialmente extraídas de una distribución Gaussiana q . D_f es el número de filtros para cada kernel. \odot corresponde a la multiplicación por elementos.

Input: vector de características de la capa anterior (minibatch)

```

 $H \leftarrow$  Input de la capa convolucional 3D (minibatch)
 $z_0 \sim q(z_0)$ 
 $z_{T_f} = NF(z_0)$ 
 $M_h = H * (M_w \odot reshape(z_{T_f}, [1, 1, 1, D_f]))$ 
 $V_h = H^2 * \Sigma_w$ 
 $E \sim \mathcal{N}(0, 1)$ 
return  $M_h + \sqrt{V_h} \odot E$ 
```

Output: muestra de vector de características según la Ec.(21)

2.3. Calibración en Redes Neuronales Profundas

[Guo et al., 2017] afirma que las redes neuronales profundas modernas a menudo no están calibradas. Como resultado, interpretar los números predichos como probabilidades no es correcto. A menudo, los problemas del mundo real requieren modelos que produzcan no sólo una predicción correcta sino también una medida fiable de confianza en ella. La confiabilidad se refiere a la probabilidad estimada de que el pronóstico sea correcto. Por ejemplo, como lo aclara [Vasilev and D'yakonov, 2023], si un algoritmo predice que una muestra determinada de pacientes está sana con una confianza de 0,9, esperamos que el 90 % de ellos esté realmente sano. Un modelo con una estimación de confianza confiable se llama calibrado. Junto con la interpretación de las predicciones de las redes neuronales, la calibración confiable es importante cuando las estimaciones de probabilidad se introducen en pasos posteriores del algoritmo.

Como se ha dicho, muchas aplicaciones de la vida real requieren algo más que un modelo que prediga el resultado más probable. Esto resulta evidente en ámbitos críticos como los coches con conducción autónoma o los diagnósticos médicos, en los que unas predicciones deficientes pueden provocar pérdidas significativas. Estas aplicaciones también requieren predicciones sólidas y fiables con garantías sobre la incertidumbre del modelo [Kendall and Gal, 2017].

La importancia del uso de las probabilidades para expresar la incertidumbre de una predicción ha sido ampliamente reconocida tanto en la estadística clásica [Dawid, 1982, Murphy and Winkler, 1977] y en los ambientes de Machine Learning [Gal, 2016, Guo et al., 2017, Hernández-Lobato and Adams, 2015, Li and Gal, 2017]. No obstante, la investigación contemporánea carece de un punto de referencia ampliamente reconocido para la calidad de la incertidumbre predictiva.

En su lugar, la calidad de las estimaciones de la incertidumbre se determina basándose en fundamentos teóricos [Hernandez-Lobato et al., 2016, Keskar et al., 2016, Li et al., 2017]; la capacidad del modelo para generalizar datos de prueba o ejemplos fuera de la distribución [Blundell et al., 2015, Gal and Ghahramani, 2016, Graves, 2011, Korattikara Balan et al., 2015]; y la asignación de baja probabilidad a ejemplos adversos [Li and Gal, 2017]. Aunque todas estas justificaciones son importantes, ninguno de estos enfoques proporciona evidencia directa de la calidad de las estimaciones de la incertidumbre.

En [Heek, 2018] se formaliza la calidad de las estimaciones de la incertidumbre utilizando el marco de calibración [Dawid, 1982]. Se dice que las predicciones de un modelo están bien

calibradas si la esperanza sobre cualquier variable aleatoria derivada de estas predicciones coincide con la media observada a largo plazo.

Los métodos de calibración ofrecen una poderosa herramienta para evaluar la calidad de la estimación de las incertidumbres, ya que las predicciones bien calibradas pueden ser interpretadas como probabilidades objetivas. En el marco bayesiano, una predicción debe considerarse subjetiva en el sentido de que las predicciones dependen de suposiciones previas sobre el comportamiento de un proceso aleatorio. La probabilidad objetiva sigue la interpretación frecuentista de la probabilidad, según la cual la probabilidad de un suceso aleatorio corresponde a su frecuencia a largo plazo. [Heek, 2018] demuestra cómo las probabilidades subjetivas (bayesianas) y objetivas (frecuentistas) pueden relacionarse mediante el teorema de calibración, que establece que cualquier modelo de probabilidad subjetivo debe considerarse bien calibrado.

Estos métodos de calibración están muy relacionados con otros tópicos en el aprendizaje automático. Investigaciones recientes sobre ejemplos adversos (o controversiales)² han revelado que las redes neuronales de clasificación tienden a realizar predicciones falsas con alta confianza en un subconjunto significativo del espacio de datos de entrada [Goodfellow et al., 2014].

El hecho de que estos errores se cometan con alta confiabilidad puede verse como un tipo de descalibración. Otro subcampo relacionado es el de la imparcialidad en IA, cuyo objetivo es reducir los sesgos contra las subpoblaciones en los datos. La calibración puede servir como definición de equidad [Kleinberg et al., 2016] y también ser una condición para una clasificación justa en un algoritmo de clasificación [Pleiss et al., 2017]. Además, un modelo con buenas estimaciones de incertidumbre puede utilizarse para determinar un buen equilibrio entre la exploración y la explotación en los algoritmos de refuerzo [Blundell et al., 2015]. Los métodos actuales de inferencia variacional y muestreo no son capaces de superar a los enfoques lineales simples en el aprendizaje por refuerzo basado en el muestreo de Thompson [Riquelme et al., 2018]. Esto sugiere que las estimaciones de incertidumbre no son de calidad suficiente.

[Vasilev and D'yakonov, 2023, Guo et al., 2017] relacionan algunas técnicas y métricas para evaluar qué tan calibrados están los modelos.

Iniciando con la definición de la perfecta calibración como

$$\mathbb{P}(\hat{Y} = Y | \hat{P} = p) = p, \forall p \in [0, 1]. \quad (22)$$

2.3.1. Diagramas de Confiabilidad:

Son una representación visual de la calibración del modelo. Estos diagramas trazan la precisión esperada de la muestra en función de la confianza. Si el modelo está perfectamente calibrado, es decir, si Ec.(22) se cumple, entonces el diagrama debería representar la función de identidad. Cualquier desviación de una diagonal perfecta representa una mala calibración. Para estimar la exactitud esperada de muestras finitas, se agrupan las predicciones en M contenedores o *bins* de intervalos (cada uno de tamaño $1/M$) y se calcula la precisión de cada contenedor. Definiendo B_m como el set de índices de muestras cuya confianza de predicción cae en el intervalo $I_m = (\frac{m-1}{M}, \frac{m}{M}]$. Entonces, la exactitud de B_m es

$$\text{Exactitud o acc}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} 1(\hat{y}_i = y_i), \quad (23)$$

donde \hat{y}_i y y_i son los valores predichos y reales de las clases para la muestra i . La probabilidad clásica dice que si $\text{acc}(B_m)$ es un estimador insesgado y consistente de $\mathbb{P}(\hat{Y} = Y | \hat{P} \in I_m)$. Se definiría la confianza promedio dentro del contenedor B_m como

²Ejemplos Adversos

$$\text{conf}(B_m) = \frac{1}{|B_m|} \sum_{i \in B_m} \hat{p}_i, \quad (24)$$

donde \hat{p}_i es la confianza en la muestra i . $\text{acc}(B_m)$ y $\text{conf}(B_m)$ aproximan a los lados izquierdo y derecho de Ec.(22) respectivamente para cada lote B_m . Por lo tanto, un modelo perfectamente calibrado tendrá $\text{acc}(B_m) = \text{conf}(B_m)$ para todo $m \in 1, \dots, M$.

2.3.2. Error de Calibración Esperado (ECE):

Si bien los diagramas de calibración son herramientas visuales muy poderosas, es más conveniente tener una estadística que resuma la evaluación de la calibración. Un indicativo mala calibración es la diferencia de expectativas entre confianza y exactitud

$$\mathbb{E}_{\hat{P}} \left[|\mathbb{P}(\hat{Y} = Y | \hat{P} = p) - p| \right]. \quad (25)$$

ECE aproxima Ec.(25) particionando las predicciones en M igualmente-espaciados contenedores o *bins* (similar a la definición de la sección 2.3.1) y toma el promedio ponderado de la diferencia entre exactitud / confianza de los contenedores. Más precisamente

$$\text{ECE} = \sum_{m=1}^M \frac{|B_m|}{n} |\text{acc}(B_m) - \text{conf}(B_m)|, \quad (26)$$

donde n es el número de muestra. La diferencia entre acc y conf para un contenedor dado representa el *gap* de calibración (generalmente se visualizan en las barras rojas dentro del diagrama de confiabilidad).

2.4. Métricas para evaluar el rendimiento del modelo

Al evaluar la eficacia de los modelos de clasificación, es esencial emplear una gama diversa de métricas que ofrezcan información matizada sobre su desempeño. Estas métricas sirven como medidas cuantitativas para medir la precisión, confiabilidad y capacidad de generalización del modelo en diferentes tareas de clasificación. Desde métricas fundamentales como la exactitud y la precisión hasta medidas más matizadas como el área bajo la curva ROC y la intersección sobre la unión, cada métrica proporciona perspectivas únicas sobre las fortalezas y debilidades del modelo. En esta sección, profundizamos en un examen exhaustivo de varias métricas comúnmente utilizadas en la evaluación de modelos de clasificación, aclarando su importancia, interpretación y fórmulas. Al evaluar exhaustivamente los modelos utilizando una combinación de estas métricas, los investigadores pueden obtener una comprensión integral de su desempeño y tomar decisiones informadas con respecto a la selección, optimización e implementación del modelo.

- Matriz de confusión: La matriz de confusión proporciona un resumen completo de las predicciones del modelo en comparación con las etiquetas reales en un formato tabular. Consta de cuatro cuadrantes: verdaderos positivos (TP, por sus siglas en inglés), verdaderos negativos (TN, por sus siglas en inglés), falsos positivos (FP, por sus siglas en inglés) y falsos negativos (FN, por sus siglas en inglés). Cada celda de la matriz representa el recuento de instancias de una combinación particular de etiquetas reales y previstas. La matriz de confusión es invaluable para comprender los tipos y frecuencias de errores de clasificación cometidos por el modelo, lo que permite una comprensión más profunda de

su desempeño en diferentes clases y facilita el análisis de errores y el refinamiento del modelo.

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

Figura 3: Matriz de Confusión

- Exactitud (Acc) [Müller et al., 2022]: La exactitud mide la proporción de muestras clasificadas correctamente del total de muestras. Es una métrica fundamental para evaluar el rendimiento general de un modelo de clasificación. La fórmula para la precisión es

$$\text{Exactitud} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \quad (27)$$

- Precisión (Prec) : La precisión mide la proporción de predicciones positivas verdaderas entre todas las predicciones positivas realizadas por el modelo. Indica la capacidad del modelo para evitar falsos positivos. La fórmula para la precisión es

$$\text{Precisión} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \quad (28)$$

- Recall [Müller et al., 2022]: También conocido como sensibilidad, mide la proporción de predicciones positivas verdaderas entre todas las instancias positivas reales en los datos. Indica la capacidad del modelo para identificar todos los casos relevantes. La fórmula es

$$\text{Recall (Sensibilidad)} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \quad (29)$$

- Puntaje F1 (F1) [Müller et al., 2022]: El puntaje F1 es la media armónica de precisión y recuperación. Proporciona una medida equilibrada que considera tanto los falsos positivos como los falsos negativos. La fórmula para el puntaje F1 es

$$\text{F1} = \frac{2 \times \text{TP}}{2 \times \text{TP} + \text{FP} + \text{FN}}. \quad (30)$$

- Kappa de Cohen (Kap) [Müller et al., 2022]: El Kappa de Cohen mide la concordancia entre las clasificaciones previstas y reales, teniendo en cuenta la posibilidad de que el acuerdo se produzca por casualidad. Es particularmente útil cuando se trata de conjuntos de datos desequilibrados. La fórmula del Kappa de Cohen varía según el contexto, pero normalmente implica cálculos de concordancia observada y esperada.

$$Kap = \frac{\text{Acuerdo Observado} - \text{Acuerdo Esperado}}{1 - \text{Acuerdo Esperado}}. \quad (31)$$

- Área bajo la curva ROC (AUC ROC) [Müller et al., 2022]: La curva ROC es una representación gráfica del equilibrio entre la tasa de verdaderos positivos (sensibilidad) y la tasa de falsos positivos (1-especificidad) para diferentes valores de umbral. AUC ROC cuantifica el rendimiento general de un modelo de clasificación en todos los umbrales posibles. Cuanto mayor sea el AUC ROC, mejor será la capacidad del modelo para discriminar entre instancias positivas y negativas.

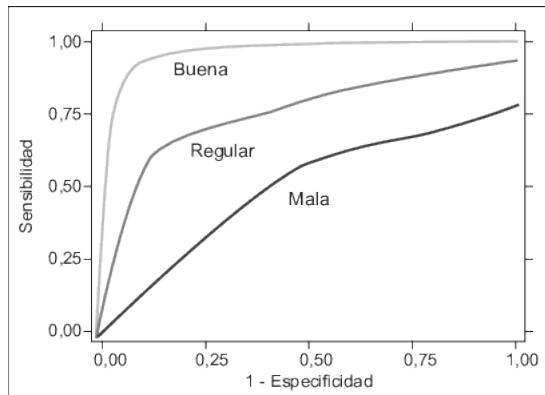


Figura 4: Curva ROC

- Intersección sobre unión (IoU) [Müller et al., 2022]: También conocido como la Similitud de Jaccard, mide la superposición entre los cuadros delimitadores o las máscaras de segmentación de la verdad fundamental y la predicha. Se utiliza comúnmente en tareas de detección de objetos y segmentación de imágenes, y se diferencia del F1-Score porque penaliza más la subsegmentación y la sobresegmentación. La fórmula para IoU es

$$IoU = \frac{\text{Área de Intersección}}{\text{Área de Unión}}. \quad (32)$$

$$IoU = \frac{TP}{TP + FP + FN}. \quad (33)$$

- FBeta-Score: El FBeta-Score es una forma generalizada de puntaje F1 que permite ajustar el énfasis en la precisión y la recuperación utilizando el parámetro beta. Cuando beta es 1, equivale a la puntuación F1. La fórmula para FBeta-Score es

$$FBeta-Score = (1 + \beta^2) \times \frac{\text{Precisión} \times \text{Recall}}{(\beta^2 \times \text{Precisión}) + \text{Recall}} \quad (34)$$

El uso de métricas defectuosas y estándares de evaluación faltantes en la comunidad científica para la evaluación del desempeño de modelos en procedimientos sensibles a la salud es una gran amenaza para la calidad y confiabilidad de los sistemas CDS (Clinical Decision

Support) [Müller et al., 2022]. Estas métricas aplicadas, junto con la matriz de confusión, desempeñan un papel crucial en la evaluación del rendimiento de los modelos de clasificación, incluida la precisión, la confiabilidad y la capacidad de generalización. Al considerar múltiples métricas, los investigadores obtienen una comprensión integral de las fortalezas y debilidades de un modelo, lo que permite una toma de decisiones informada y la optimización del modelo.

3. Marco Metodológico

3.1. Configuración del Entorno

Para el entorno computacional, se usó una máquina virtual alojada en Google Colab, equipada con una GPU T4 para acelerar los cálculos de aprendizaje profundo. Para garantizar un rendimiento óptimo y adaptarse a las demandas computacionales de nuestro proyecto, ampliamos la asignación de RAM. Nuestro entorno Python se configuró con versiones específicas de bibliotecas clave:

- tensorflow: 2.15.0
- tensorflow-probability: 0.23.0
- volumentations-3D: 1.0.4
- keras_tuner: 1.4.5
- classification_models_3D: 1.0.7

Estas versiones proporcionaron funcionalidades esenciales para construir y entrenar redes neuronales con componentes bayesianos, facilitando el aumento de datos para imágenes volumétricas 3D, la exploración eficiente del espacio de hiperparámetros para la optimización del modelo y el acceso a arquitecturas de redes neuronales convolucionales 3D previamente entrenadas para tareas de clasificación. Este entorno proporcionó las herramientas y recursos necesarios para realizar experimentos y análisis integrales para el proyecto.

3.2. Datos

El conjunto de datos [Morozov et al., 2020] utilizado en este proyecto consiste en tomografías computarizadas (TC) anónimas de pulmón humano con hallazgos relacionados con COVID-19, así como también sin hallazgos. Las tomografías se obtuvieron entre el 1 de marzo de 2020 y el 25 de abril de 2020, en hospitales médicos de Moscú, Rusia. Estos datos según la licencia de MosMed sirve para múltiples propósitos, incluyendo material educativo para especialistas en imágenes médicas, desarrollo y prueba de servicios basados en IA, y como fuente de información para especialistas médicos y el público en general. También es de libre acceso y puede compartirse, copiarse y redistribuirse bajo condiciones específicas.

Las tomografías del conjunto de datos muestran signos radiológicos intrínsecos de la infección por COVID-19. Un subconjunto de los estudios ha sido anotado con máscaras binarias de píxeles, indicando regiones de interés como opacificaciones en vidrio deslustrado y consolidaciones. Las dimensiones de las imágenes son uniformes para los ejes X e Y, pero los cortes de las tomografías presentan diferencias, variando desde 33 a 65 cortes.

Aunque no se facilita información sobre el sexo de los pacientes, cada estudio corresponde a un único paciente, y cada estudio está representado por una serie de imágenes reconstruidas de la ventana mediastínica de tejidos blandos. Los estudios se clasifican en cinco grupos en

función de los signos de neumonía viral. El conjunto de datos comprende un total de 1110 estudios, dividiéndose en las siguientes 5 categorías:

1. CT-0: Representa tejido pulmonar normal sin signos TC de neumonía viral (254 muestras).
2. CT-1: Indica varias opacificaciones en vidrio deslustrado, con menos del 25 % de afectación del parénquima pulmonar (684 muestras).
3. TC-2: Muestra opacificaciones en vidrio deslustrado con afectación del 25-50 % del parénquima pulmonar (125 muestras).
4. TC-3: Muestra opacificaciones en vidrio deslustrado con consolidación parcial, con afectación del 50-75 % del parénquima pulmonar (45 muestras).
5. TC-4: Muestra opacificaciones en vidrio deslustrado con consolidación parcial, con afectación de más del 75 % del parénquima pulmonar (4 muestras).

Es importante señalar que la distribución de los estudios en estas categorías se basó únicamente en los hallazgos radiológicos y no en los resultados de la prueba de reacción en cadena de la polimerasa (PCR) ni en la verificación clínica.

Adicional, dada la relevancia clínica de distinguir entre ausencia y presencia de manifestaciones características de neumonía por COVID-19, este proyecto se ha centrado estratégicamente en las clasificaciones CT-0, CT-2 y CT-3 para el desarrollo y la evaluación de modelos. Al priorizar estas clasificaciones, el proyecto pretende abordar los principales desafíos de diagnóstico que enfrentan los médicos y al mismo tiempo garantizar la relevancia clínica y la aplicabilidad de los hallazgos. Además, considerando la complejidad de la tarea de clasificación y los limitados recursos computacionales disponibles, centrarse en un subconjunto de clasificaciones permite una arquitectura de modelo más manejable e interpretable, minimizando el riesgo de sobreajuste e inestabilidad del modelo.

3.3. Metodología

La fase inicial de la metodología se centra principalmente en procesar y analizar imágenes de TC en 3D para identificar enfoques óptimos para la clasificación de la neumonía por COVID-19. Una vez que se obtienen resultados satisfactorios con los modelos 3D, pasamos a una fase posterior donde el enfoque cambia a transformar las tomografías computarizadas volumétricas en representaciones 2D. Esta transformación nos permite extraer características y patrones significativos de las imágenes al tiempo que reduce la complejidad computacional. Posteriormente, implementamos las arquitecturas de mejor rendimiento identificadas durante la fase de modelado 3D pero en sus contrapartes 2D. Al aprovechar las representaciones 3D y 2D, nuestro objetivo es explorar el rendimiento comparativo de las arquitecturas de redes neuronales en diferentes modalidades de datos de imágenes de TC, lo que en última instancia mejorará nuestra comprensión de la tarea de clasificación y optimizará el rendimiento del modelo para aplicaciones del mundo real.

Para ejecutar el proyecto, establecimos el siguiente flujo de trabajo estructurado:

3.3.1. Tomografías 3D:

1. **Estandarización del Estudio:**

- Antes de iniciar el procesamiento de imágenes, desarrollamos un script (0_Organizacion_Rutas.ipynb) para organizar las rutas de todas las imágenes y asignarlas aleatoriamente en tres conjuntos de datos: entrenamiento (70 %), prueba (20 %) y validación (10 %). Esto aseguró la coherencia en el entrenamiento del modelo y permitió realizar pruebas comparables.

2. Procesamiento de imágenes:

- Investigamos exhaustivamente métodos estándar para procesar imágenes de tomografía computarizada (TC), centrándonos particularmente en imágenes torácicas. Se observó que las imágenes de TC normalmente se procesan utilizando valores de Unidad Hounsfield (HU) en términos de ancho y centro de la ventana. Posteriormente, definimos diferentes ventanas HU para realizar pruebas, referenciadas en el Cuadro(1), limitando los valores de píxeles de las imágenes de acuerdo con el ancho de ventana HU especificado. Luego normalizamos los datos utilizando una escala mínima-máxima para restringir los valores entre 0 y 1, lo que facilita el entrenamiento de la red neuronal.
- Sobre la estandarización de cortes: se utilizará la proyección del paquete SciPy [Virtanen et al., 2020] en Python.

ID Ventana	Límite Inferior Píxeles	Límite Superior Píxeles	Origen	Fuente
W1	-1000	400	Ejercicio Keras	https://keras.io/examples/vision/3D_image_classification/
W2	-1100	500	Investigación	https://www.youtube.com/watch?v=totknaoZ-2o&t=334s
W3	-950	550	Investigación	https://www.unsam.edu.ar/escuelas/ciencia/alumnos/PUBLIC.1999-2006-%20Alumnos%20P.F.I/(TAC)%20GUERREIRO%20MARTINS%20MARIANO.pdf
W4	-1000	0	Investigación	https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7120362/

Cuadro 1: Ventanas HU para procesamiento de Tomografías Computarizadas

- Realizamos una lectura de imágenes de archivos Nifti (extensión .nii), seguida de una rotación de 90 grados para garantizar una orientación consistente. Se estandarizó el tamaño de las imágenes a unas dimensiones de 128 x 128 x 64.

3. Aumento de datos:

- Utilizando la biblioteca Volumentations 3D [Solovyev et al., 2022], se implementaron varias transformaciones para las tomografías computarizadas, incluido invertir los ejes X, Y y Z, agregar ruido gaussiano y ajustar la fluctuación del color para mejorar la solidez y la generalización del modelo.

4. Entrenamiento modelo:

- El entrenamiento inicial de los modelos implicó la creación de modelos individuales para cada ventana HU. Se adoptó como punto de partida la arquitectura descrita en la documentación de Keras para la clasificación de imágenes 3D. Posteriormente, aplicamos técnicas de aumento de datos a estos modelos y comparamos sus métricas de rendimiento.

- En la fase de entrenamiento posterior, se experimentó con variantes de arquitecturas de redes neuronales convolucionales populares disponibles en el paquete de modelos de clasificación 3D [Solovyev et al., 2022], incluidas:
 - ResNet18
 - ResNet34
 - SeresNet18
 - SeresNet34
 - EfficientNetB0
 - DenseNet121

Estos modelos se entrenaron utilizando la ventana HU de mejor rendimiento identificada en el paso anterior.

- Para explorar la cuantificación de la incertidumbre, reemplazamos las capas deterministas con contrapartes bayesianas en el modelo seleccionado, aprovechando las bibliotecas TensorFlow [Abadi et al., 2015] y TensorFlow Probability de Python. Esto implicó incorporar capas de inferencia variacional (tanto densas como convolucionales) y capas de flujos de normalización multiplicativos (MNF).
- Toda la parte práctica del proyecto se implementó utilizando una membresía de Google Colab, lo que mejoró las capacidades de la máquina virtual, permitiendo la experimentación y el entrenamiento eficiente del modelo.

5. Calibración e Incertidumbre:

- Análisis de calibración ³:

El objetivo principal del análisis de calibración es determinar en qué medida las probabilidades predichas se alinean con la exactitud real de las predicciones del modelo.

 - Diagramas de confiabilidad: Se usarán los diagramas de confiabilidad para visualizar la relación entre las probabilidades predichas y la precisión esperada. Para cada *bin*, la confianza y la exactitud promedio se calculan y se grafican entre sí. Las desviaciones de la línea diagonal indicarán errores de calibración, y las brechas entre confianza y precisión revelan un exceso o falta de confianza en las predicciones del modelo.
 - Error de calibración esperado (ECE): El ECE es la estadística que permitirá cuantificar el rendimiento general de la calibración del modelo. Mide la diferencia de expectativas entre confianza y exactitud, en donde los valores más bajos indican una mejor calibración.
 - Análisis de umbrales: Se realizan diagramas de confiabilidad y cálculos de ECE para diferentes puntos de corte (umbrales), incluidos 0.4, 0.5, 0.6, 0.7 y 0.8. Este análisis proporciona información sobre cómo varía la calibración del modelo entre diferentes umbrales de decisión.
- Evaluación de incertidumbre ⁴:

La evaluación de la incertidumbre tiene como objetivo cuantificar la incertidumbre asociada con las predicciones del modelo y proporcionar información sobre la variabilidad y solidez de las predicciones del modelo.

³Adaptado de este repositorio

⁴Adaptado de este repositorio

- Enfoque basado en simulación: La evaluación de la incertidumbre se realiza mediante un enfoque basado en simulación, donde se realizan múltiples simulaciones para generar intervalos de confianza para cada etiqueta. Cada simulación genera un conjunto de predicciones, lo que permite estimar intervalos de confianza que capturan la variabilidad e incertidumbre en las predicciones del modelo.
- Intervalos de confianza: Los intervalos de confianza proporcionan un rango de valores plausibles para cada etiqueta, lo que refleja la incertidumbre inherente a las predicciones del modelo. Al analizar la amplitud y la variabilidad de estos intervalos, se pueden obtener conocimientos sobre los niveles de confianza e incertidumbre del modelo.

6. Finalización:

- Con base en los resultados obtenidos, así como los conocimientos adquiridos a lo largo del proyecto, consolidamos los aprendizajes teóricos y prácticos en este documento. Resumiendo los conceptos, metodologías, resultados computacionales y conocimientos adquiridos durante el proyecto de profundización.

3.3.2. Tomografías 2D:

En este capítulo, profundizamos sobre el proceso de proyección de tomografías 3D en sus contrapartes 2D, explorando diversas alternativas de proyección y metodologías para el preprocesamiento. Para la proyección de tomografías 3D en imágenes 2D, se examinaron tres alternativas a lo largo de la tercera dimensión (*slices*). Estas alternativas permiten transformar datos volumétricos en una representación bidimensional.

- **Optimización de ventanas:** similar al enfoque adoptado para las tomografías 3D, el preprocesamiento y la estandarización de imágenes 2D giran en torno a identificar la configuración de ventana óptima para el procesamiento de imágenes. Para este caso, se aprovechará la ventana con mayor rendimiento en tomografía 3D, se aplican los mismos pasos de rotación y cambio de dimensiones a lo largo de los ejes X e Y, lo que da como resultado imágenes redimensionadas a 128 x 128.
- **Métodos de aplanamiento:** para obtener imágenes bidimensionales a partir de tomografías 3D, se emplean tres métodos de aplanamiento distintos: proyección promedio, proyección suma y proyección máxima a lo largo de la tercera dimensión. Cada método ofrece una representación simplificada de los datos volumétricos originales.
 - Proyección promedio: en este método, los valores de píxeles a lo largo de la tercera dimensión (cortes) de la tomografía 3D se promedian para generar una única imagen 2D. Cada píxel de la imagen resultante representa la intensidad promedio de los píxeles correspondientes en todos los cortes. La proyección promedio ofrece una representación simplificada de los datos volumétricos, donde cada valor de píxel refleja la intensidad media de la estructura 3D subyacente.
 - Proyección de suma: la técnica de proyección de suma implica sumar los valores de píxeles a lo largo de la tercera dimensión del tomograma 3D para producir una imagen 2D. Cada píxel de la imagen resultante representa la intensidad acumulada de los píxeles correspondientes en todos los cortes. La proyección de suma proporciona una visualización de la contribución acumulativa de cada voxel a lo largo de la tercera dimensión.

- Proyección máxima: en la proyección máxima, se evalúan los valores de píxeles a lo largo de la tercera dimensión del tomograma 3D y se selecciona el valor de intensidad máxima para cada posición de píxel en la imagen 2D resultante. Este método enfatiza áreas de máxima intensidad dentro de los datos volumétricos, ofreciendo información sobre la distribución espacial de regiones de alta intensidad en la imagen.
- **Modelamiento:** Aprovechando los conocimientos obtenidos de las tomografías 3D, se seleccionan arquitecturas que muestran un rendimiento superior para una evaluación adicional en imágenes 2D.

Al explorar meticulosamente alternativas de proyección, metodologías de preprocesamiento y técnicas de evaluación de modelos, el capítulo de imágenes 2D tiene como objetivo proporcionar un marco integral para aprovechar las representaciones bidimensionales de tomografías 3D en tareas de clasificación de imágenes médicas.

En resumen, la metodología propuesta para desarrollar y entrenar redes neuronales bayesianas y determinísticas optimizadas para clasificar la neumonía por COVID-19 en tomografías computarizadas 2D y 3D implica cuatro pasos principales: (1) preprocesar el conjunto de datos para garantizar la coherencia en la calidad de la imagen, (2) desarrollar y entrenar los modelos usando capas convolucionales, (3) evaluar el desempeño de los modelos usando métricas estándar y (4) analizar las incertidumbres encontradas en cada uno de los modelos estocásticos. Todo el proceso se implementará utilizando Python y el marco TensorFlow, para la arquitectura de red neuronal bayesiana nos apoyaremos en el paquete TensorFlow Probability. Todos los modelos se procesarán en una máquina virtual alojada en Google Cloud Platform con GPU T4 y RAM ampliada.

4. Resultados:

La exploración de las etapas iniciales del proyecto arrojó información importante sobre el rendimiento del modelo y las estrategias de optimización. Aquí presentamos una descripción completa de los resultados obtenidos, destacando los hallazgos clave y las metodologías empleadas. Esta investigación se embarcó en una exploración multifacética de varias metodologías y arquitecturas de modelos para discernir enfoques óptimos para clasificar la neumonía por COVID-19 en tomografías computarizadas en 3D y 2D. Este esfuerzo abarcó definir la ventana de la Unidad Hounsfield (HU) más adecuada, evaluar el impacto de las técnicas de aumento de datos, examinar la eficacia de diversas arquitecturas de redes neuronales y profundizar en la calibración y estimación de la incertidumbre de los modelos con mayor desempeño. A través de un análisis meticuloso, el objetivo es proporcionar información valiosa sobre las complejidades del rendimiento de los modelos y la cuantificación de la incertidumbre en el contexto de la clasificación de la neumonía COVID-19.

4.1. Ventana HU:

Uno de los primeros hallazgos del proyecto fue la determinación de la ventana HU óptima para el procesamiento de imágenes por TC. A través de varias experimentaciones, se encontró que la ventana HU, denominada W4 en el Cuadro(1), proporcionaba métricas de rendimiento superiores en comparación con otras ventanas. En particular, el modelo entrenado con la ventana W4 mostró una precisión notable en el set de datos de testeо:

- Precisión - ACC (Keras_Arch_3D_W1_V1 - Ventana W1):

- Test: 0.84
- Precisión - ACC (Keras_Arch_3D_W2_V1 - Ventana W2):
 - Test: 0.89
- Precisión - ACC (Keras_Arch_3D_W3_V1 - Ventana W3):
 - Test: 0.85
- **Precisión - ACC (Keras_Arch_3D_W4_V1 - Ventana W4):**
 - **Test: 0.91**

El siguiente gráfico es una representación visual de las métricas de rendimiento obtenidas por varios modelos en el conjuntos de datos de testeo. Cada gráfico de este tipo representa un trazado de coordenadas paralelas, donde el eje X denota las métricas de rendimiento individuales, y el eje Y representa los valores de las métricas correspondientes que van de 0 a 1. Dentro de cada gráfico, cada línea corresponde a un modelo específico, mostrando su rendimiento a través de múltiples métricas simultáneamente.

En la parte superior de cada métrica evaluada, el nombre del modelo que alcanza el valor más alto para esa métrica en particular se muestra de forma destacada, ofreciendo una referencia rápida para identificar los modelos de mayor rendimiento. Además, una leyenda adjunta proporciona información sobre el tipo de arquitectura de cada modelo, lo que facilita la comparación y la interpretación.

Estos gráficos de coordenadas paralelas son una herramienta de visualización eficaz que permite a los investigadores discernir tendencias, patrones y discrepancias en el rendimiento de los modelos en diferentes conjuntos de datos y variaciones en la arquitectura de los modelos.

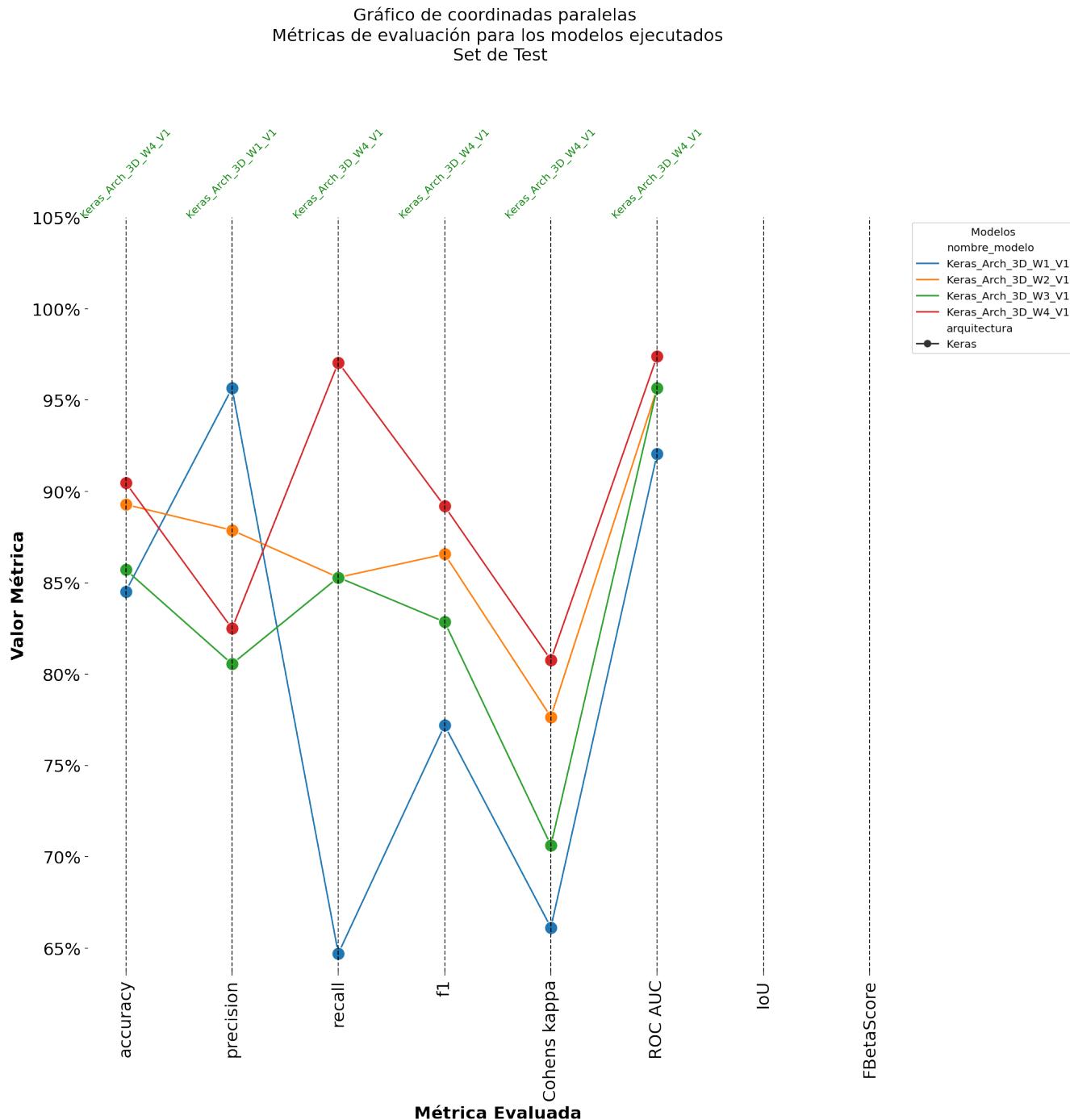


Figura 5: Métricas evaluadas en el primer grupo de modelos - Set de Test

La Fig.(5) muestra claramente que el modelo que sobresale en la mayoría de las métricas evaluadas es el Keras_Arch_3D_W4_V1.

4.2. Data Augmentation:

Después de implementar varias transformaciones del paquete volumentations-3D mencionadas en la sección 3.3, se observó una disminución notable en las métricas de casi 20 puntos en comparación con el modelo de referencia sin aumento como lo ilustra la Fig.(6). Por esto, se tomó la decisión de aplicar solo una transformación específica a los datos de entrenamiento. Adoptamos una transformación de rotación en la que cada volumen de tomografía computarizada 3D se rotaba en un ángulo seleccionado al azar dentro de un rango predefinido (-20° a +20°). Esta rotación ayudó a aumentar el conjunto de datos de entrenamiento al introducir variaciones en la orientación de las tomografías computarizadas, mejorando así la solidez del modelo ante diferentes perspectivas de imágenes.

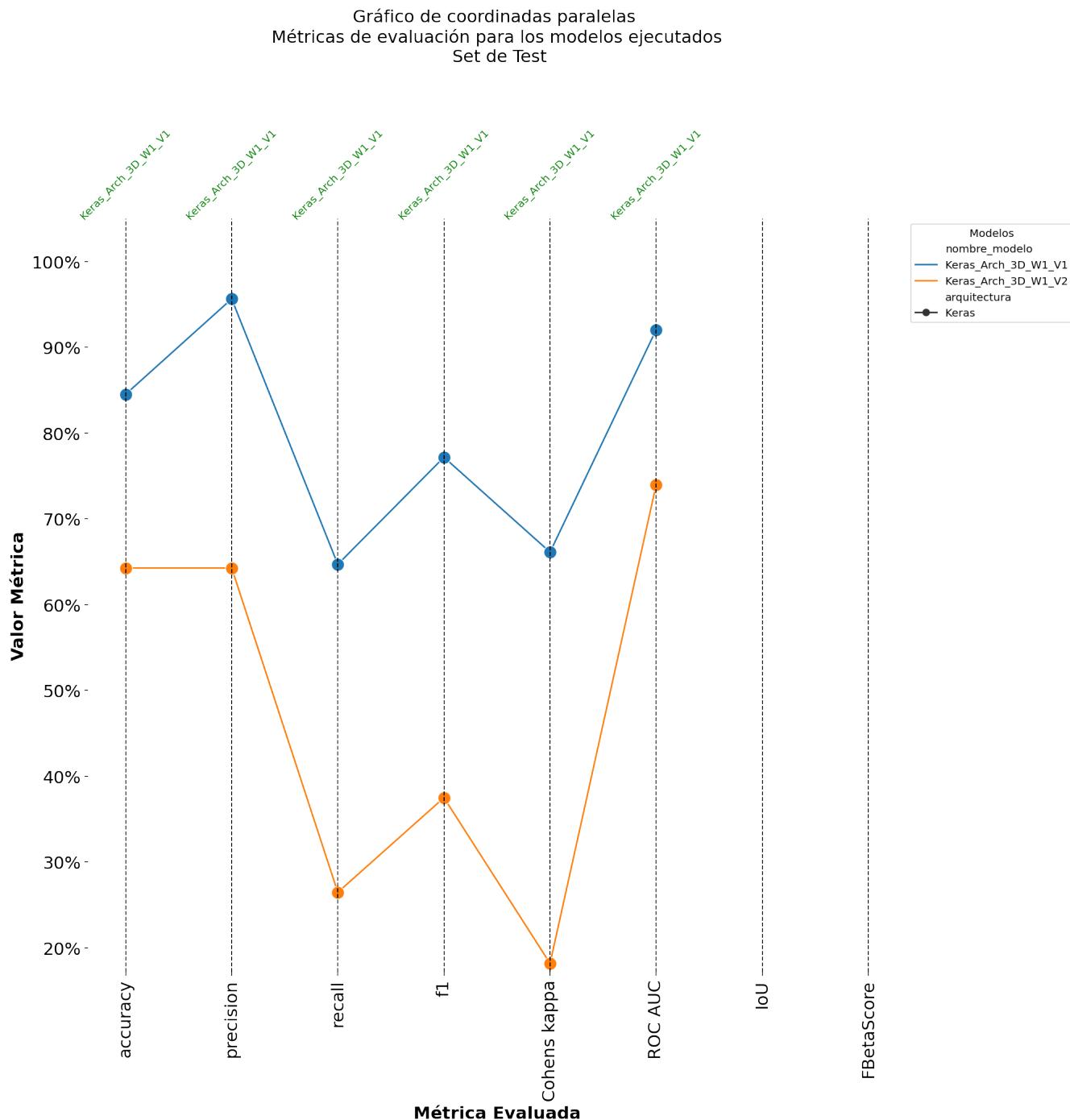


Figura 6: Métricas evaluadas en una red neuronal con ventana W1 vs su contraparte con Data Augmentation del paquete volumentations-3D. V1 corresponde al modelo inicial, V2 al modelo en donde se implementa *Data Augmentation*

4.3. Tomografías 3D:

4.3.1. Modelos:

4.3.1.1 Redes Deterministas

Una vez definida la ventana HU con mayor desempeño para nuestro caso de estudio y haciendo uso de la librería classification-models-3D, se exploraron arquitecturas de redes neuronales populares en su forma nativa y con modificaciones menores, como agregar capas GlobalAveragePooling3D y GlobalMaxPooling3D, así como también ajustar el número de filtros en las capas convolucionales. Sin embargo, los resultados de estas arquitecturas no superaron el modelo de arquitectura inicial de Keras utilizando la ventana W4 mencionado en la sección 4.1.

Por lo que los siguientes ejercicios estuvieron centrados en optimizar los hiperparámetros del modelo de red neuronal determinista seleccionado del grupo inicial (Keras_Arch_3D_W4_V1). Aprovechando el paquete keras-tuner [O'Malley et al., 2019] con el tuneador Hyperband, se realizó una búsqueda exhaustiva en varias configuraciones de hiperparámetros. El tuneador Hyperband optimiza el proceso de búsqueda descartando iterativamente configuraciones de hiperparámetros con bajo rendimiento, lo que permite una exploración eficiente del espacio de hiperparámetros. La cuadrícula de búsqueda de hiperparámetros incluía los siguientes parámetros:

- Número de bloques ⁵: hp.Int('num_blocks', min_value=1, max_value=3)
- Número de neuronas iniciales: hp.Choice('start_neuron', valores=[16, 32, 64, 128])
- Tipo de capa de agrupación: hp.Choice('pooling_type', valores=['global_avg', 'global_max'])
- Tasa de aprendizaje: hp.Choice('learning_rate', valores=[1e-2, 1e-3, 1e-4])
- Tasa de abandono: hp.Float('dropout_rate', min_value=0.0, max_value=0.5, step=0.1)
- Número de unidades en la última capa densa: hp.Int('units_dense', min_value=64, max_value=512, sampling='log', step=2)

Los hiperparámetros óptimos son los siguientes:

- Número de bloques ⁵: 3
- Número de neuronas de inicio: 128
- Tipo de capa de agrupación: agrupación máxima global
- Tasa de aprendizaje: 0.001
- Tasa de abandono: 0.2
- Número de unidades en la última capa densa: 256

⁵*En este proyecto, un bloque se refiere a un grupo de capas compuesto por una capa convolucional, una capa MaxPool y una capa BatchNormalization.

Con estos hiperparámetros, logramos el modelo determinista de mejor rendimiento dentro de las opciones consideradas.

nombre.modelo	tipo.set	accuracy	precision	recall	f1	Cohens kappa	ROC AUC
Keras_Arch_3D_W4_V1	Test	90 %	83 %	97 %	89 %	81 %	97 %

Cuadro 2: Métricas Red Neuronal Determinista - Ventana W4

nombre.modelo	tipo.set	accuracy	precision	recall	f1	Cohens kappa	ROC AUC
Keras_Arch_3D_W4_V1.Optimizado	Test	96 %	92 %	100 %	96 %	93 %	99 %

Cuadro 3: Métricas Red Neuronal Determinista - Ventana W4 - Configuración hiperparámetros según Keras-Tuner

Las métricas de rendimiento del modelo Keras_Arch_3D_W4_V1_Optimizado relacionadas en el Cuadro(3) mostraron una mejora significativa en comparación con su versión original: Keras_Arch_3D_W4_V1 (Cuadro(2)). En particular, el modelo optimizado demostró capacidades de clasificación mejoradas en varias métricas, particularmente evidentes en el conjunto de datos de Test. Por ejemplo, el área bajo la curva (AUC) aumentó del 97 % al 99 %, la exactitud (accuracy) aumentó del 90 % al 96 %, ilustrando un mayor poder discriminatorio del modelo. Como un último ejercicio, se aplicaron técnicas de regularización al modelo determinista optimizado, sin embargo, no se lograron resultados satisfactorios como lo ilustra la Fig.(7), en donde el modelo Keras_Arch_3D_W4_V1_Hyper_Reg reduce el desempeño del modelo optimizado. Como resultado, la arquitectura optimizada (Keras_Arch_3D_W4_V1_Optimizado) fue designada como el modelo representativo entre sus contrapartes deterministas.

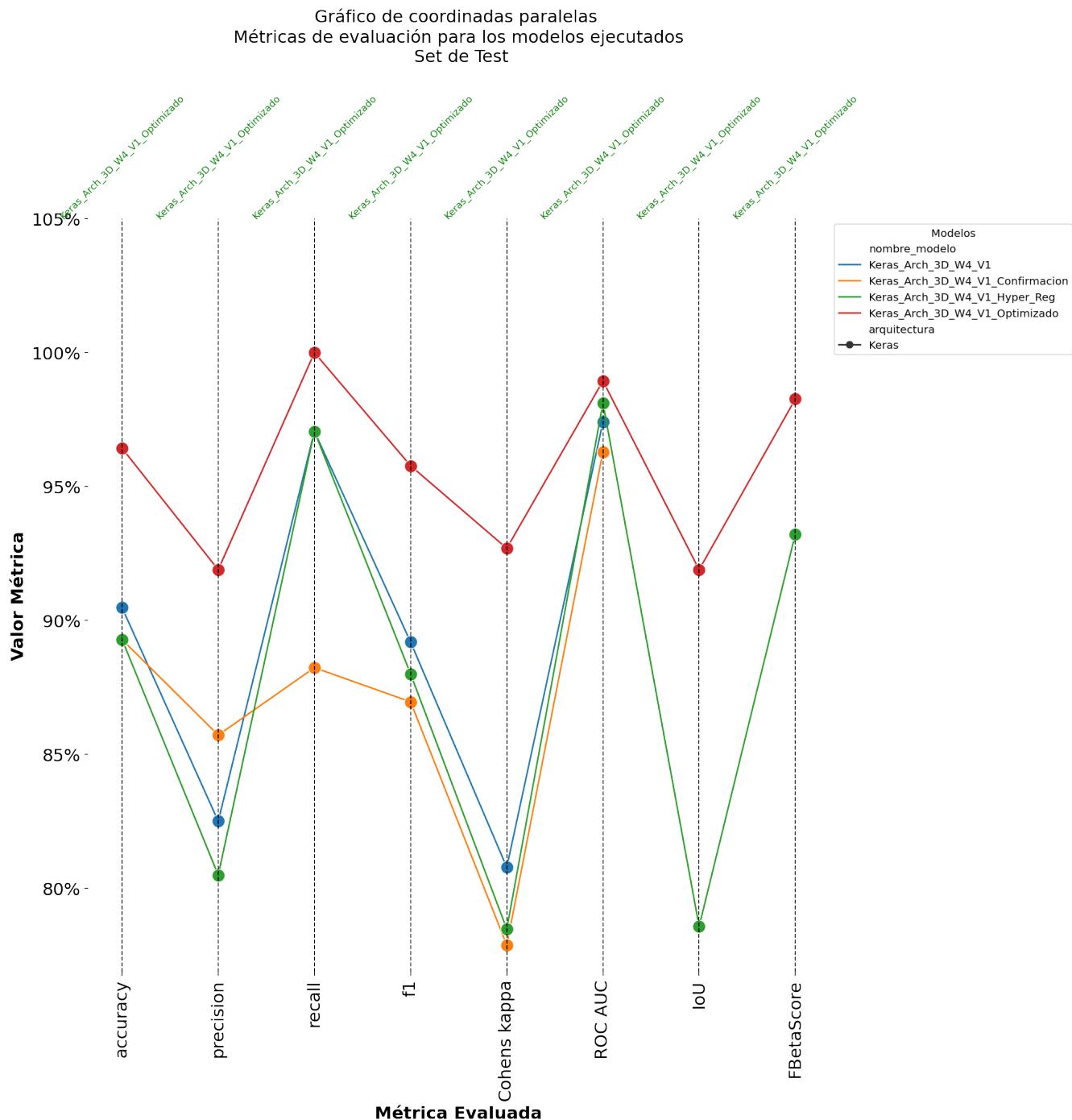


Figura 7: Contaste de métricas evaluadas en los modelos deterministas antes y después de optimizar hiper-parámetros - Set de Test

En conclusión, las redes neuronales deterministas exploradas en este estudio han propor-

cionado información valiosa sobre la clasificación de la neumonía por COVID-19 en tomografías computarizadas 3D. A través de rigurosas experimentaciones y optimizaciones, se han identificado configuraciones de arquitecturas claves e hiperparámetros que contribuyen a mejorar el rendimiento del modelo. A pesar de los desafíos encontrados, como la eficacia limitada de la regularización, la arquitectura determinista optimizada emerge como una solución prometedora para tareas de clasificación precisas. A medida que hacemos la transición a las redes neuronales bayesianas para una mayor exploración, las lecciones aprendidas de los modelos deterministas sirven como base para el desarrollo y refinamiento futuros de modelos. En el gráfico de coordenadas paralelas Fig.(8), se resumen todos los resultados de los modelos deterministas evaluados ⁶, proporcionando una descripción general de su desempeño en varias métricas.

⁶La descripción de cada modelo la pueden encontrar en A.1

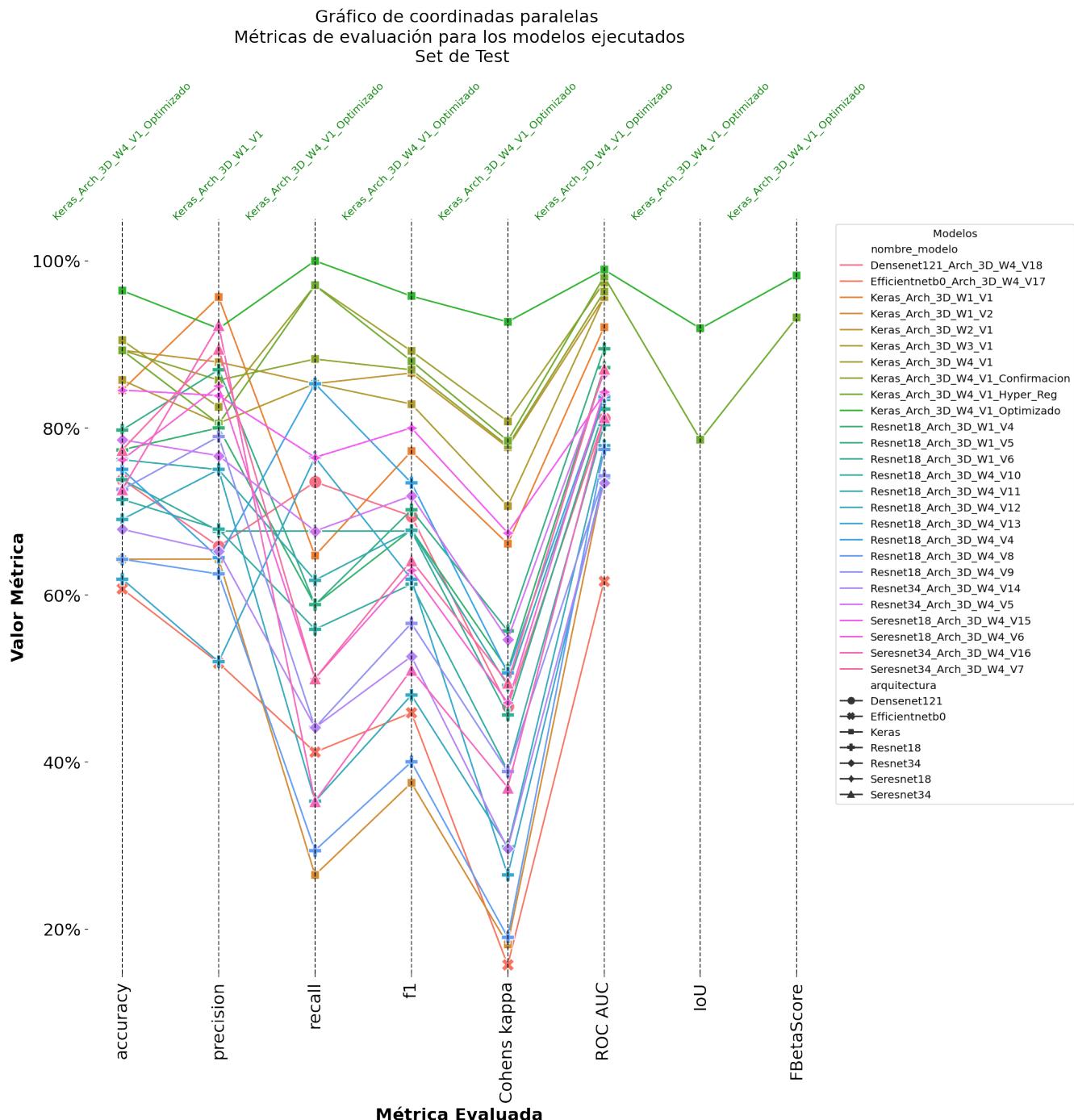


Figura 8: Contaste de métricas evaluadas en todos los modelos deterministas - Set de Test

4.3.1.2 Redes Bayesianas

El siguiente paso de exploración profundizó en el entrenamiento de redes neuronales bayesianas, comenzando con modificaciones menores en la arquitectura del modelo determinista optimizado. La principal alteración implicó reemplazar la última capa densa del modelo determinista con sus contrapartes bayesianas, (fragmentos de código Python en A.2)

Estos cambios introducen estocasticidad en la salida del modelo, permitiéndole capturar la incertidumbre inherente a los datos y las predicciones. Al utilizar capas bayesianas, el modelo gana la capacidad de expresar incertidumbre en sus predicciones, ofreciendo así resultados más informativos en comparación con sus homólogos deterministas. Específicamente, las opciones 1A y 1B utilizan capas de Bernoulli independientes para modelar la distribución de salida binaria, mientras que las opciones 2A y 2B emplean capas MNFDense junto con capas de Bernoulli independientes para incorporar un flujo de normalización multiplicativo. Estas variaciones permiten diferentes enfoques para la cuantificación de la incertidumbre y la representación del modelo, lo que proporciona flexibilidad para modelar relaciones complejas dentro de los datos.

Si bien las exploraciones iniciales no implicaron cálculos de las métricas para los conjuntos de datos, se llevó a cabo un análisis de la evolución del modelo época por época, centrándose en la precisión y la función de pérdida A.3. Las observaciones revelaron que las redes neuronales bayesianas iniciales no lograron igualar el rendimiento del modelo determinista optimizado.

Además de las variaciones mencionadas anteriormente en la última capa, la experimentación adicional implicó ajustes a los parámetros de entrenamiento, específicamente aumentando el número de épocas y alterando la tasa de aprendizaje. Finalmente, se exploraron capas alternativas para evaluar su impacto en el rendimiento del modelo. Estos incluyeron la integración de las siguientes capas:

- Dense Flipout: Esta capa introduce estocasticidad aplicando la técnica "flipout" durante el entrenamiento, que perturba los pesos de la red. Al incorporar incertidumbre a través de perturbaciones de peso, Dense Flipout permite que el modelo capture y represente la incertidumbre aleatoria inherente a los datos.
- Reparametrización local densa: en esta capa se emplea el truco de reparametrización local (LRT), cuyo objetivo es aproximar la distribución de salida de la capa reparametrizando los pesos localmente. La reparametrización local densa mejora la capacidad del modelo para capturar y propagar la incertidumbre a través de la red.
- Reparametrización densa: similar a la reparametrización local densa, esta capa utiliza el truco de reparametrización para incorporar estocasticidad al modelo. Al repararmetrizar los pesos de la capa, facilitando la propagación de la incertidumbre a través de la red, permitiendo así un modelado más robusto y expresivo de distribuciones de datos complejas.

Tras la exploración de las variaciones en la última capa, el paso siguiente implicó reemplazar las capas convolucionales deterministas con sus contrapartes bayesianas en tres escenarios distintos:

- Enfoque 1: Capas convolucionales MNF 3D: en este enfoque, las capas convolucionales deterministas se sustituyeron por capas convolucionales bayesianas de flujos de normalización multiplicativa (MNF) 3D, aprovechando los resultados de la investigación en [Garcia-Farieta et al., 2024]. Las capas MNF ofrecen un marco flexible para capturar distribuciones complejas de pesos, lo que permite representaciones más expresivas y conscientes de la incertidumbre dentro del modelo.

- Enfoque 2: Capas convolucionales Convolution3DFlipout: este enfoque implicó reemplazar las capas convolucionales deterministas con capas convolucionales Flipout. Las capas Flipout introducen estocasticidad en las operaciones convolucionales mediante la aplicación de la técnica "flipout", que perturba los pesos de los filtros convolucionales durante el entrenamiento.
- Enfoque 3: Capas convolucionales Convolution3DReparameterization: en este enfoque, las capas convolucionales deterministas se reemplazaron con capas convolucionales de reparametrización. Las capas de Reparametrización Convolution3D, aprovechan la técnica de Reparametrización para el muestreo de pesos.

Además de los enfoques de capas convolucionales bayesianas mencionados anteriormente, otra alternativa explorada fue la implementación de una red neuronal Monte Carlo Dropout. En este enfoque, se mantuvo la arquitectura del modelo optimizado determinista, con modificaciones realizadas en la capa dropout descritas en A.4

En la capa dropout modificada, la inclusión de "training=True" desempeña un papel importante a la hora de especificar el comportamiento de la misma durante la fase de entrenamiento de la red neuronal. Este argumento indica a la capa dropout que debe aplicar la regularización de abandono estableciendo aleatoriamente una fracción de unidades de entrada en cero durante cada iteración de entrenamiento. Al activar el abandono durante el entrenamiento, el modelo aprende a adaptarse y generalizar de manera más efectiva, ya que está expuesto a diferentes patrones de abandono en cada época de entrenamiento. Esta estocasticidad introducida por la regularización de abandonos ayuda a prevenir el sobreajuste y mejora la capacidad del modelo para generalizar a datos nuevos.

Después de evaluar varias alternativas de arquitectura bayesiana, los hallazgos de la Fig.(9) indican que la red neuronal bayesiana que comprende capas convolucionales de flujo de normalización multiplicativos (MNF) y una capa de salida densa de MNF demuestra un rendimiento superior en las métricas evaluadas en el conjunto de datos de testeo. Esta arquitectura bayesiana logra un rendimiento comparable al de la red neuronal determinista seleccionada, demostrando su eficacia a la hora de clasificar la neumonía por COVID-19 en tomografías computarizadas en 3D.

Al aprovechar las capas convolucionales de MNF y la capa de salida densa de MNF, esta red neuronal bayesiana captura de manera efectiva la incertidumbre subyacente inherente a los datos, mejorando así sus capacidades predictivas. Estos resultados subrayan la importancia de los enfoques bayesianos en las tareas de imágenes médicas, particularmente para proporcionar predicciones sólidas y confiables al tiempo que cuantifican la incertidumbre.

El siguiente gráfico de coordenadas paralelas resume el rendimiento de todos los modelos bayesianos evaluados y ofrece información valiosa sobre su rendimiento comparativo y destaca la eficacia de la red neuronal bayesiana basada en MNF (Modelo: Keras_Arch_3D_W4_V1_Uncertainty_V4_3C) ⁷.

⁷Más detalle sobre los nombres de los modelos en A.1

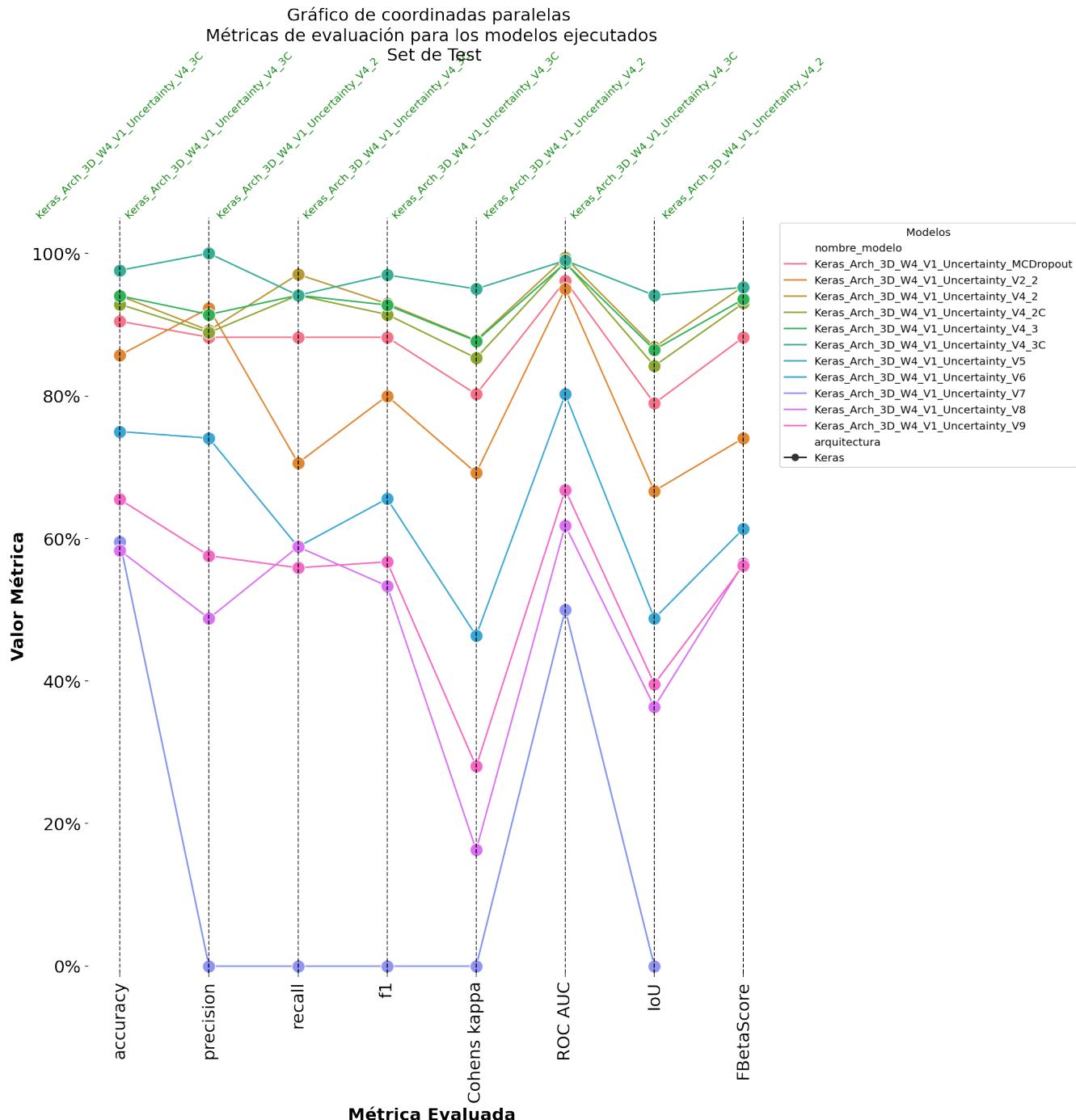


Figura 9: Contaste de métricas evaluadas en los modelos Bayesianos - Set de Test

4.3.2. Calibración:

El análisis de calibración de los modelos implicó la visualización de diagramas de confiabilidad y el cálculo del Error de Calibración Esperado (ECE). Estos análisis se realizaron para una variedad de modelos⁸ con el fin de evaluar su desempeño de calibración e identificar configuraciones óptimas.

Los modelos considerados para el análisis de calibración incluyeron:

- Keras_Arch_3D_W4_V1_Confirmacion
- Keras_Arch_3D_W4_V1_Optimizado
- Keras_Arch_3D_W4_V1_Uncertainty_V4_2C
- Keras_Arch_3D_W4_V1_Uncertainty_V4_3C
- Keras_Arch_3D_W4_V1_Uncertainty_V6
- Keras_Arch_3D_W4_V1_Uncertainty_MCDropout

Además de evaluar diferentes modelos, se realizaron experimentos para evaluar el impacto de cambiar la capa probabilística de Bernoulli de los modelos bayesianos a una capa de activación *Sigmoid*.

Los resultados del análisis de calibración revelaron varias ideas clave:

- Efecto de la capa de activación *Sigmoid*: la sustitución de la capa probabilística de Bernoulli por una capa de activación *Sigmoid* en las redes neuronales bayesianas no parece afectar significativamente la calidad de las predicciones de clase. Las comparaciones de gráficos de coordenadas paralelas entre modelos con y sin la capa de activación *Sigmoid* ilustraron un rendimiento consistente en la mayoría de las métricas de clasificación A.5.
- Punto de corte óptimo: Un hallazgo notable fue la identificación de un punto de corte óptimo (umbral) para las predicciones de clase 1. Si bien el umbral predeterminado se había establecido en 0.5 en experimentos anteriores, el análisis reveló un mejor rendimiento de las métricas cuando el umbral se ajustó a 0.4. Este ajuste resultó en una mayor precisión de clasificación y otras métricas relacionadas A.6.
- Confiabilidad y métricas ECE: entre los modelos evaluados, dos emergieron como los de mejor desempeño en términos de confiabilidad y métricas ECE. Para los modelos deterministas, Keras_Arch_3D_W4_V1_Optimizado (Fig.(10a)) demostró consistentemente una sólida calibración en diferentes configuraciones de umbral. Su histograma de confiabilidad exhibió líneas verticales estrechamente alineadas y un alto nivel de confianza, indicativo de predicciones confiables. De manera similar, para los modelos bayesianos, Keras_Arch_3D_W4_V1_Uncertainty_V4_3C (Fig.(10b)) con la capa *Sigmoid* y un umbral de 0.4 mostró excelentes características de calibración, con líneas verticales muy agrupadas y una desviación mínima de la diagonal A.7.

⁸La descripción de cada modelo la pueden encontrar en A.1

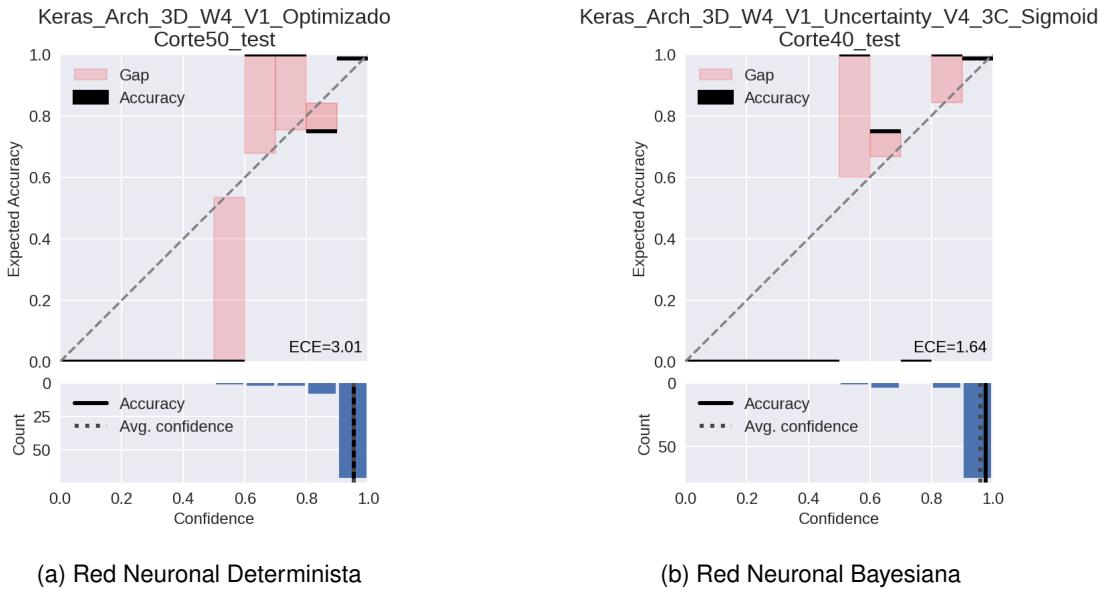


Figura 10: Diagrama de Confiabilidad e Histograma - Set de Test

Los diagramas de confiabilidad en la Fig.(10) de estos modelos de alto rendimiento reforzaron aún más la calidad de su calibración, y los histogramas representan predicciones bien calibradas y discrepancias mínimas entre confianza y precisión. Estos hallazgos subrayan la importancia del análisis de calibración para evaluar la confiabilidad de los modelos de aprendizaje automático.

4.3.3. Incertidumbre:

El análisis de incertidumbre de los modelos implicó el análisis de intervalos de predicción derivados de ejercicios de simulación realizados en imágenes de TC seleccionadas mencionado en la sección 3.3.1. Estas imágenes abarcaban una variedad de clases y se utilizaron para evaluar el rendimiento predictivo y las capacidades de estimación de incertidumbre de los modelos deterministas y bayesianos.

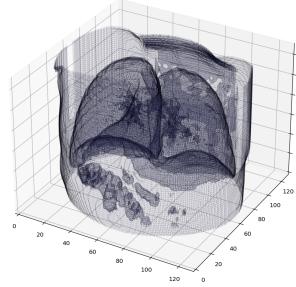
De forma aleatoria, se seleccionaron seis imágenes para el análisis, comprendidas por cuatro imágenes que representan clases con distintos grados de afectación de neumonía (CT-1 y CT-4) desconocidas por el modelo y dos imágenes del conjunto de datos de testeo etiquetadas como no afectadas por neumonía (CT-0).

El análisis se centró en contrastar las predicciones y la incertidumbre (en el escenario bayesiano) de los dos mejores modelos elegidos en etapas anteriores: el modelo determinista (Keras_Arch_3D_W4_V1_Optimized) y el modelo bayesiano (Keras_Arch_3D_W4_V1_Uncertainty_V4_3C + Sigmoid). Estos modelos fueron seleccionados en función de su desempeño y representatividad en el contexto del estudio.

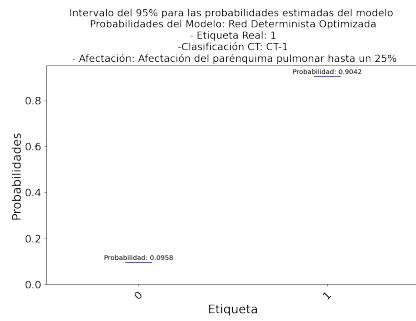
4.3.3.1 Análisis de imágenes individuales:

- Imágenes de clase CT-1:

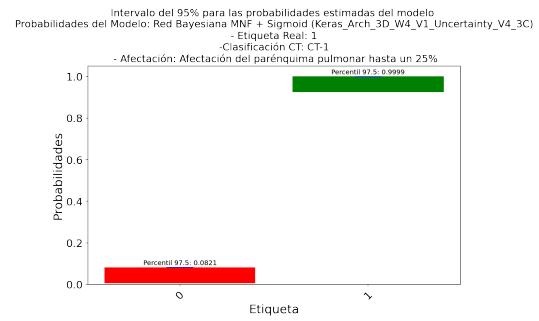
- Imagen 1 (Fig.(11)): Tanto el modelo determinista como el bayesiano identificaron correctamente deficiencias en la tomografía, clasificándola como clase 1 (presencia de afectación). Sin embargo, el modelo bayesiano destacó al ofrecer intervalos de predicción estrechos, indicando alta confianza en sus predicciones. Esta capacidad de cuantificar la incertidumbre de manera precisa es esencial en aplicaciones médicas, ya que permite una toma de decisiones más informada y precisa. Estos resultados resaltan la utilidad de los modelos bayesianos en el diagnóstico médico. Los intervalos de confianza para cada clase por el modelo bayesiano son:
 - Clase 0: [0.0001179 - 0.08210155]
 - Clase 1: [0.91789845 - 0.9998821]



(a) Representación 3D Tomografía



(b) Red Neuronal Determinista



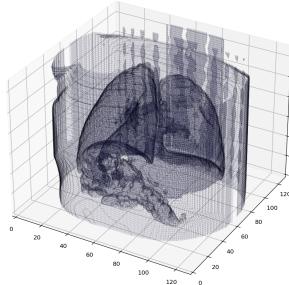
(c) Red Neuronal Bayesiana

Figura 11: Análisis Incertidumbre - Imagen 1: CT-1

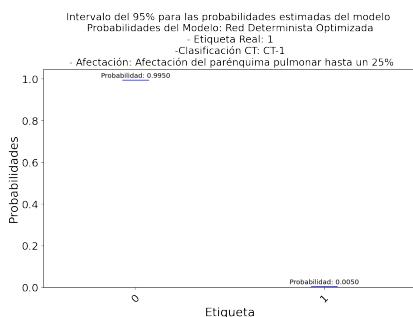
- Imagen 2 (Fig.(12)): Ambos modelos concuerdan en que la tomografía analizada no presenta ninguna afectación (es decir, clase 0), no obstante esta afirmación no es precisa, ya que la imagen pertenece a una clase con un compromiso del parénquima pulmonar del 25 %. A pesar de este error en la predicción, la red bayesiana ofrece intervalos de confianza notablemente amplios, lo que sugiere la presencia de incertidumbre. Este resultado resalta la importancia de considerar la incertidumbre en las predicciones, incluso cuando estas son incorrectas, proporcionando así una evaluación más cautelosa del diagnóstico médico. Los intervalos de confianza para cada clase por el modelo bayesiano son:

- Clase 0: [0.85678702 - 0.99927345]

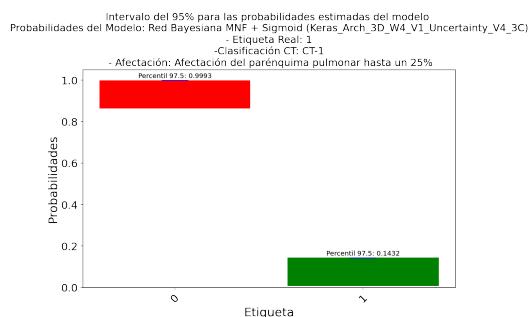
- Clase 1: [0.00072655 - 0.14321298]



(a) Representación 3D Tomografía



(b) Red Neuronal Determinista



(c) Red Neuronal Bayesiana

Figura 12: Análisis Incertidumbre - Imagen 2: CT-1

■ Imágenes de clase CT-4:

- Imagen 3 (Fig.(13)): Ambos modelos identificaron correctamente alteraciones en la tomografía analizada, clasificándola como perteneciente a la clase 1. Sin embargo, los intervalos de predicción del modelo bayesiano tienen una amplitud bastante considerable, indicando una mayor incertidumbre en sus predicciones. Esta amplia incertidumbre podría ser considerada como una señal para generar alertas o solicitar una revisión adicional por parte de los profesionales médicos, con el fin de asegurar una evaluación más precisa y evitar posibles fallos diagnósticos. Los intervalos de confianza para cada clase por el modelo bayesiano son:

- Clase 0: [0.00087272 - 0.36155368]
- Clase 1: [0.63844632 - 0.99912728]

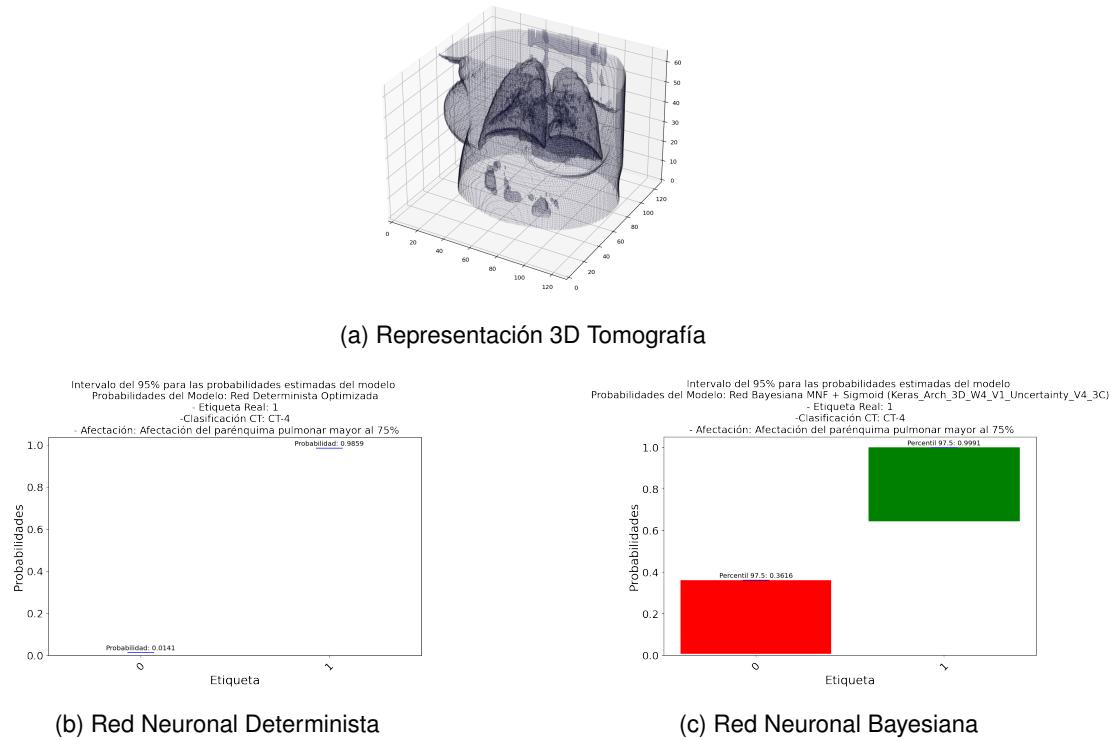


Figura 13: Análisis Incertidumbre - Imagen 3: CT-4

- Imagen 4 (Fig. 14): De manera similar a la imagen de clase CT-4 anterior (Fig. 13), ambos modelos identificaron correctamente las deficiencias en la tomografía como clase 1. Sin embargo, resulta interesante observar que los intervalos de predicción del modelo bayesiano fueron notablemente estrechos, lo que sugiere una incertidumbre extremadamente baja en sus predicciones. Esta estrechez en los intervalos de predicción del modelo bayesiano apunta a una confiabilidad excepcional en sus resultados, remarcando su capacidad para realizar predicciones precisas y fiables incluso en situaciones complejas. Los intervalos de confianza para cada clase por el modelo bayesiano son:
 - Clase 0: [0.0 - 0.00002203]
 - Clase 1: [0.99997797 - 1.0]

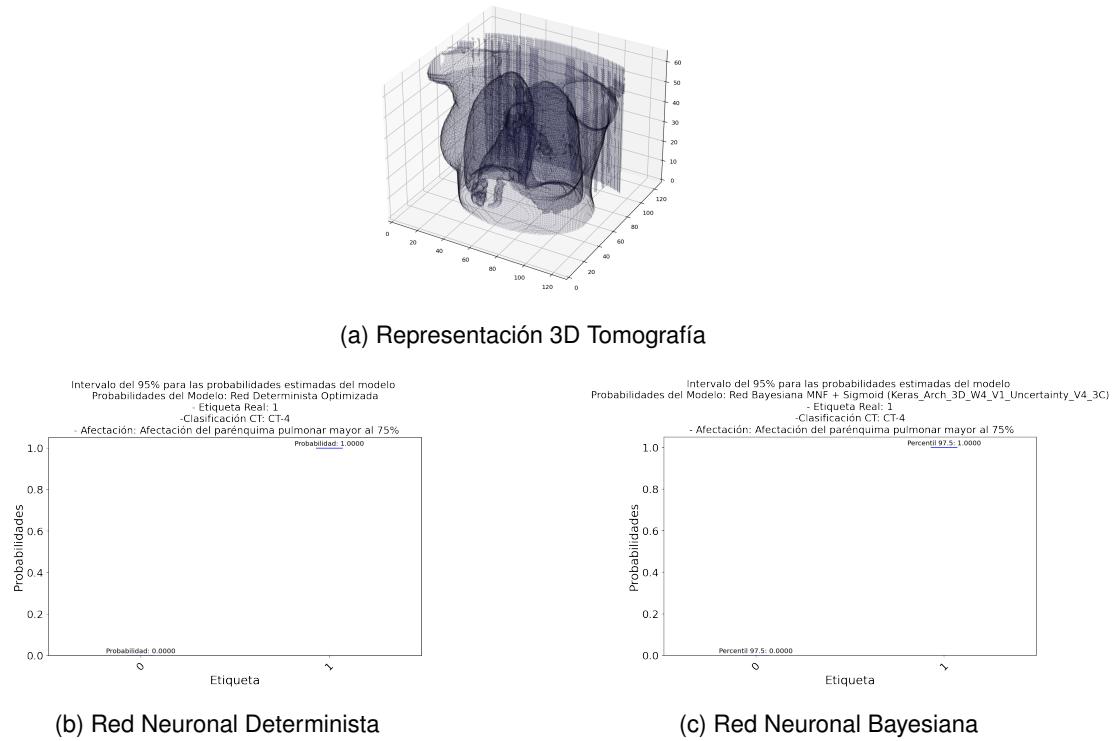


Figura 14: Análisis Incertidumbre - Imagen 4: CT-4

■ Imágenes de clase CT-0:

- Imagen 5 (Fig. 15)): El modelo determinista exhibió un sesgo hacia la etiqueta 1, mientras que el modelo bayesiano se inclinó hacia la etiqueta 0. A pesar de que la etiqueta real era 0 (ausencia de anomalías), los amplios intervalos de predicción del modelo bayesiano indican una incertidumbre sustancial. La presencia de estos amplios intervalos de predicción en el modelo bayesiano sugiere que el modelo no está completamente seguro de su predicción, abriendo una alerta para un diagnóstico clínico más detallado. Esto resalta la importancia de las redes neuronales bayesianas en el campo de la medicina, ya que proporcionan información crucial sobre la incertidumbre asociada con las predicciones, lo que puede ayudar a los profesionales médicos a tomar decisiones más informadas y precisas. Los intervalos de confianza para cada clase por el modelo bayesiano son:
 - Clase 0: [0.27592939 - 0.98892005]
 - Clase 1: [0.01107995 - 0.72407061]

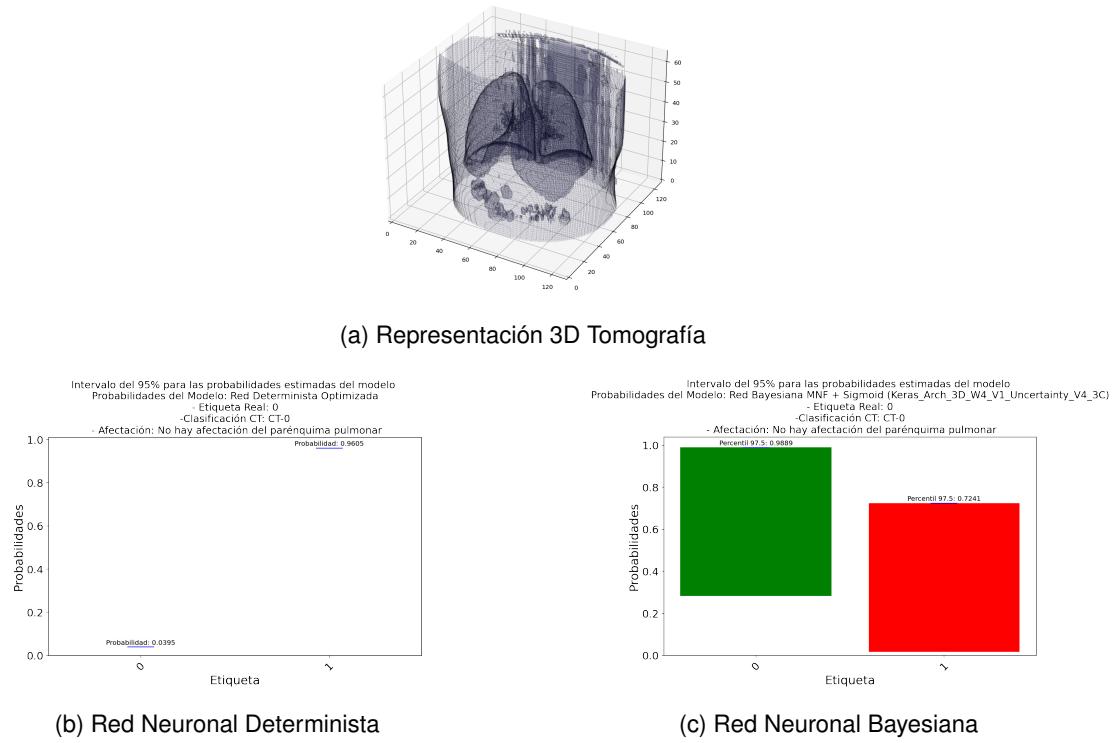


Figura 15: Análisis Incertidumbre - Imagen 5: CT-0

- Imagen 6 (Fig. 16): Ambos modelos identificaron con precisión la ausencia de anomalías en la tomografía analizada, clasificándola como clase 0. Sin embargo, cabe destacar que el modelo bayesiano muestra intervalos de confianza angostos, indicador de una baja incertidumbre en sus predicciones. Esta característica del modelo bayesiano sugiere una mayor confiabilidad en sus resultados. Los intervalos de confianza para cada clase por el modelo bayesiano son:
 - Clase 0: [0.93343621 - 0.99933766]
 - Clase 1: [0.00066234 - 0.06656379]

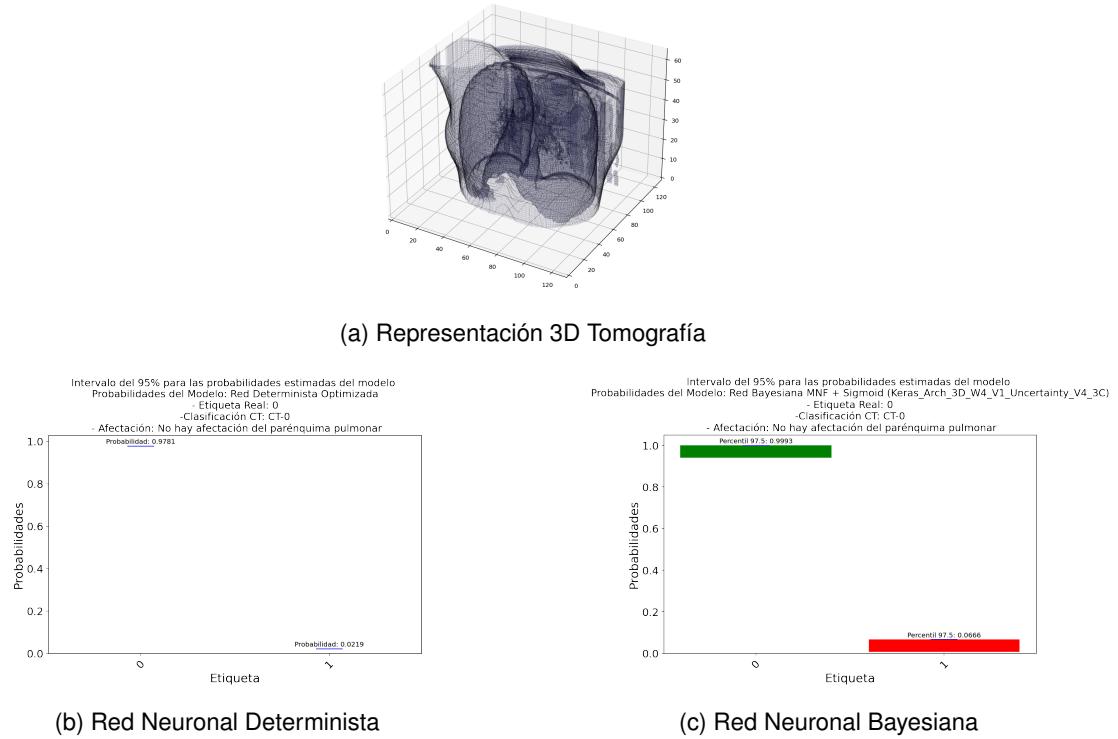


Figura 16: Análisis Incertidumbre - Imagen 6: CT-0

El análisis de imágenes individuales proporciona información sobre el rendimiento predictivo y las capacidades de estimación de la incertidumbre de los modelos, destacando la importancia de la conciencia de la incertidumbre en los procesos de toma de decisiones.

El análisis destaca la importancia de la estimación de la incertidumbre en las predicciones de los modelos. A pesar de las imprecisiones del modelo, los amplios intervalos de predicción proporcionados por la red bayesiana en algunas imágenes resaltaron la presencia de incertidumbre en las predicciones. Este reconocimiento de la incertidumbre es crucial para los procesos de toma de decisiones, ya que alerta a las partes interesadas sobre las "dudas" del modelo sobre ciertas predicciones y fomenta una interpretación cautelosa de los resultados.

Es importante destacar que intervalos excepcionalmente anchos en las predicciones de la red bayesiana pueden servir como señal para despertar alertas y promover una evaluación más detallada por parte de los profesionales médicos.

4.4. Tomografías 2D:

4.4.1. Modelos:

En este capítulo profundizamos en la evaluación de varios modelos entrenados, abarcando enfoques tanto deterministas como bayesianos. Después de ejecutar varias alternativas y entrenar diferentes modelos, incluidos enfoques deterministas y bayesianos, los resultados obtenidos no fueron satisfactorios. Parece que cambiar el tamaño de las tomografías de 3D a 2D puede haber provocado la pérdida de información crucial, afectando negativamente el rendimiento del modelo. En consecuencia, decidimos no continuar con el análisis de calibración e

incertidumbre para estos modelos. El gráfico de coordenadas paralelas en la Fig.(17) proporciona una representación visual del rendimiento de todos los modelos entrenados:

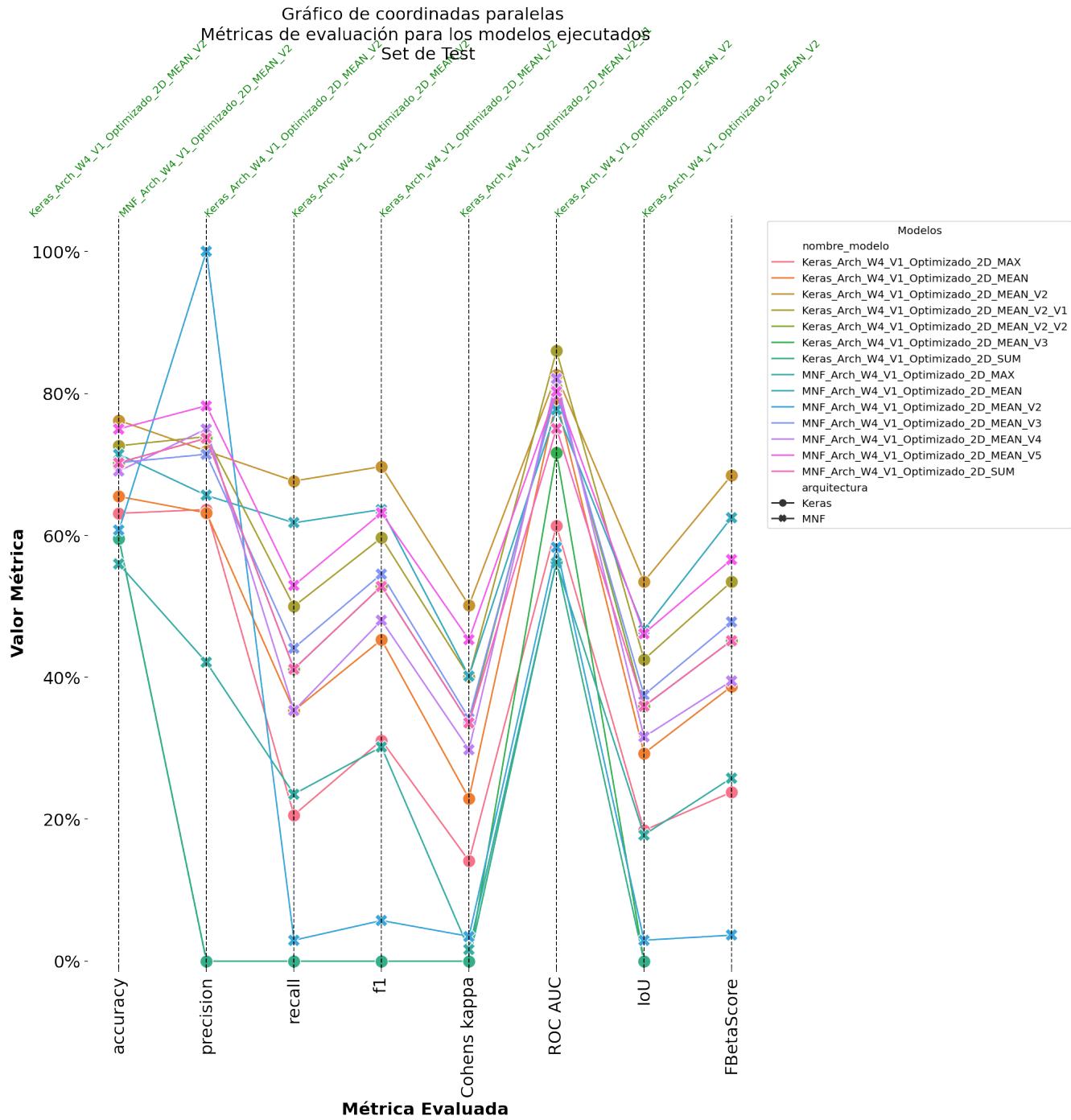


Figura 17: Contaste de métricas evaluadas en los modelos para tomografías 2D - Set de Test

Al analizar el gráfico de coordenadas paralelas, resulta evidente que todos los modelos, independientemente de su arquitectura o metodología de entrenamiento, exhiben patrones de rendimiento similares. Esta uniformidad en los resultados denota la necesidad de un enfoque más matizado en el manejo de la etapa de preprocesamiento de datos, particularmente para mitigar la pérdida de información asociada con la conversión de tomografías 3D a 2D.

5. Conclusiones

- La investigación inicial de las ventanas de la Unidad Hounsfield (HU) proporcionó información valiosa sobre la selección de alternativas de píxeles óptimas para mejorar la discernibilidad de características relevantes durante el entrenamiento de modelos de clasificación. Este paso inicial sentó una base sólida para el posterior desarrollo y optimización del modelo.
- La optimización de los hiperparámetros dentro de redes deterministas contribuyó significativamente a mejorar las métricas evaluadas, lo que en última instancia nos guió hacia la identificación de una arquitectura óptima para los ejercicios de modelado bayesiano. Este proceso iterativo resaltó la importancia de ajustar los parámetros del modelo para lograr un rendimiento superior.
- La implementación de arquitecturas bayesianas, en particular aprovechando MNF, demostró ser fundamental para identificar un modelo óptimo capaz de capturar y cuantificar la incertidumbre. Este aspecto es crucial para mejorar la interpretabilidad y confiabilidad de las predicciones de los modelos, especialmente en aplicaciones de atención médica.
- El análisis de calibración proporcionó información valiosa para comprender la calibración de nuestros modelos, arrojando conocimiento sobre la alineación entre las probabilidades previstas y los resultados reales. Este análisis profundizó nuestra comprensión de la confianza y confiabilidad del modelo.
- La exploración de la incertidumbre, particularmente a través de los intervalos de confianza generados por las predicciones de los modelos, surgió como un aspecto crítico en la evaluación de la confiabilidad de las predicciones de los modelos. Esta idea es particularmente pertinente en contextos sanitarios, donde las consecuencias de los errores de predicción pueden tener implicaciones importantes.
- El rendimiento deficiente de los modelos en diversas métricas subraya las limitaciones de la proyección 2D en comparación con la tomografía 3D. El rendimiento inferior sugiere que las representaciones 3D preservan información vital crítica para predicciones precisas, destacando la superioridad de la tomografía 3D sobre las proyecciones 2D para nuestra tarea de clasificación.
- Las metodologías y los conocimientos obtenidos de este estudio ejemplifican la utilidad de las técnicas de aprendizaje profundo en aplicaciones sanitarias. Al ofrecer enfoques matizados para el análisis de datos y la cuantificación de la incertidumbre, estos hallazgos resaltan el potencial transformador del aprendizaje profundo en el avance de las imágenes médicas y la mejora de los procesos de toma de decisiones clínicas. Estos avances sirven como testimonio de las invaluables contribuciones de las metodologías de aprendizaje profundo para abordar desafíos complejos de atención médica y mejorar los resultados de la atención al paciente.

En conclusión, este proyecto se adentró en el ámbito de la clasificación de la neumonía COVID-19 en tomografías computarizadas 3D utilizando arquitecturas de redes neuronales tanto bayesianas como deterministas. Mediante una experimentación meticulosa, la exploración de hiperparámetros y un análisis de la calibración y la incertidumbre del modelo, se obtuvieron valiosos conocimientos sobre el rendimiento y la fiabilidad de los modelos implementados.

Mientras que las tomografías tridimensionales arrojaron resultados prometedores, el rendimiento de los modelos entrenados en proyecciones bidimensionales fue menos satisfactorio, lo que indica que las representaciones tridimensionales pueden captar más información crítica para una clasificación precisa.

Las conclusiones y resultados de este proyecto, junto con los modelos entrenados, están disponibles en el repositorio GitHub [ct-scan-modeling-thesis-bayesian-deterministic](https://github.com/ct-scan-modeling-thesis-bayesian-deterministic).

Referencias

- [Abadi et al., 2015] Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., and Zheng, X. (2015). TensorFlow: Large-scale machine learning on heterogeneous systems. Software available from tensorflow.org.
- [Amari, 2012] Amari, S.-i. (2012). *Differential-geometrical methods in statistics*, volume 28. Springer Science & Business Media.
- [Ardila et al., 2019] Ardila, D., Kiraly, A. P., Bharadwaj, S., Choi, B., Reicher, J. J., Peng, L., Tse, D., Etemadi, M., Ye, W., Corrado, G., Naidich, D. P., and Shetty, S. (2019). End-to-end lung cancer screening with three-dimensional deep learning on low-dose chest computed tomography. *Nature Medicine*, 25(6):954–961.
- [Blei et al., 2017] Blei, D. M., Kucukelbir, A., and McAuliffe, J. D. (2017). Variational inference: A review for statisticians. *Journal of the American statistical Association*, 112(518):859–877.
- [Blundell et al., 2015] Blundell, C., Cornebise, J., Kavukcuoglu, K., and Wierstra, D. (2015). Weight uncertainty in neural network. In *International conference on machine learning*, pages 1613–1622. PMLR.
- [Chai, 2018] Chai, L. R. (2018). Uncertainty estimation in bayesian neural networks and links to interpretability. *Master's Thesis, Massachusetts Institute of Technology*.
- [Chang et al., 2018] Chang, P., Kuoy, E., Grinband, J., Weinberg, B., Thompson, M., Homo, R., Chen, J., Abcede, H., Shafie, M., Sugrue, L., Filippi, C., Su, M.-Y., Yu, W., Hess, C., and Chow, D. (2018). Hybrid 3d/2d convolutional neural network for hemorrhage evaluation on head ct. *American Journal of Neuroradiology*, 39(9):1609–1616.
- [Chen et al., 2021] Chen, C., Zhou, K., Zha, M., Qu, X., Guo, X., Chen, H., Wang, Z., and Xiao, R. (2021). An effective deep neural network for lung lesions segmentation from covid-19 ct images. *IEEE Transactions on Industrial Informatics*, 17(9):6528–6538.
- [Dawid, 1982] Dawid, A. P. (1982). The well-calibrated bayesian. *Journal of the American Statistical Association*, 77(379):605–610.
- [de Silva and Kunz, 2023] de Silva, K. and Kunz, H. (2023). Prediction of alzheimer's disease from magnetic resonance imaging using a convolutional neural network. *Intelligence-Based Medicine*, 7:100091.
- [Depeweg et al., 2016] Depeweg, S., Hernández-Lobato, J. M., Doshi-Velez, F., and Udluft, S. (2016). Learning and policy search in stochastic dynamical systems with bayesian neural networks. *arXiv preprint arXiv:1605.07127*.
- [Gal, 2016] Gal, Y. (2016). Uncertainty in deep learning.
- [Gal and Ghahramani, 2016] Gal, Y. and Ghahramani, Z. (2016). Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

- [Garcia-Farieta et al., 2024] Garcia-Farieta, J. E., Hortua, H. J., and Kitaura, F.-S. (2024). Bayesian deep learning for cosmic volumes with modified gravity. *Astronomy and Astrophysics*, 684:A100.
- [Gawade et al., 2023] Gawade, S., Bhansali, A., Patil, K., and Shaikh, D. (2023). Application of the convolutional neural networks and supervised deep-learning methods for osteosarcoma bone cancer detection. *Healthcare Analytics*, 3:100153.
- [Goodfellow et al., 2014] Goodfellow, I. J., Shlens, J., and Szegedy, C. (2014). Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.
- [Graves, 2011] Graves, A. (2011). Practical variational inference for neural networks. *Advances in neural information processing systems*, 24.
- [Guo et al., 2017] Guo, C., Pleiss, G., Sun, Y., and Weinberger, K. Q. (2017). On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.
- [He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.
- [Heek, 2018] Heek, J. (2018). Well-calibrated bayesian neural networks. *University of Cambridge*.
- [Hernandez-Lobato et al., 2016] Hernandez-Lobato, J., Li, Y., Rowland, M., Bui, T., Hernandez-Lobato, D., and Turner, R. (2016). Black-box alpha divergence minimization. In Balcan, M. F. and Weinberger, K. Q., editors, *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 1511–1520, New York, New York, USA. PMLR.
- [Hernández-Lobato and Adams, 2015] Hernández-Lobato, J. M. and Adams, R. (2015). Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International conference on machine learning*, pages 1861–1869. PMLR.
- [Hortúa et al., 2023] Hortúa, H. J., García, L. n., and Castañeda C., L. (2023). Constraining cosmological parameters from n-body simulations with variational bayesian neural networks. *Frontiers in Astronomy and Space Sciences*, 10.
- [Hortua et al., 2020] Hortua, H. J., Malago, L., and Volpi, R. (2020). Reliable uncertainties for bayesian neural networks using alpha-divergences. *arXiv preprint arXiv:2008.06729*.
- [Jordan et al., 1999] Jordan, M. I., Ghahramani, Z., Jaakkola, T. S., and Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine learning*, 37:183–233.
- [Kendall and Gal, 2017] Kendall, A. and Gal, Y. (2017). What uncertainties do we need in bayesian deep learning for computer vision? *Advances in neural information processing systems*, 30.
- [Keskar et al., 2016] Keskar, N. S., Mudigere, D., Nocedal, J., Smelyanskiy, M., and Tang, P. T. P. (2016). On large-batch training for deep learning: Generalization gap and sharp minima. *arXiv preprint arXiv:1609.04836*.
- [Kingma et al., 2015] Kingma, D. P., Salimans, T., and Welling, M. (2015). Variational dropout and the local reparameterization trick.

- [Kleinberg et al., 2016] Kleinberg, J., Mullainathan, S., and Raghavan, M. (2016). Inherent trade-offs in the fair determination of risk scores. *arXiv preprint arXiv:1609.05807*.
- [Korattikara Balan et al., 2015] Korattikara Balan, A., Rathod, V., Murphy, K. P., and Welling, M. (2015). Bayesian dark knowledge. *Advances in neural information processing systems*, 28.
- [Krizhevsky, 2014] Krizhevsky, A. (2014). One weird trick for parallelizing convolutional neural networks. *CoRR*, abs/1404.5997.
- [LeCun et al., 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521(7553):436–444.
- [Li and Gal, 2017] Li, Y. and Gal, Y. (2017). Dropout inference in bayesian neural networks with alpha-divergences. In *International conference on machine learning*, pages 2052–2061. PMLR.
- [Li et al., 2017] Li, Y., Turner, R. E., and Liu, Q. (2017). Approximate inference with amortised mcmc. *arXiv preprint arXiv:1702.08343*.
- [Louizos and Welling, 2017a] Louizos, C. and Welling, M. (2017a). Multiplicative normalizing flows for variational bayesian neural networks.
- [Louizos and Welling, 2017b] Louizos, C. and Welling, M. (2017b). Multiplicative normalizing flows for variational bayesian neural networks. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 2218–2227. PMLR.
- [Lundervold and Lundervold, 2019] Lundervold, A. S. and Lundervold, A. (2019). An overview of deep learning in medical imaging focusing on mri. *Zeitschrift für Medizinische Physik*, 29(2):102–127. Special Issue: Deep Learning in Medical Physics.
- [Mcclure et al., 2019] Mcclure, P., Rho, N., Lee, J., Kaczmarzyk, J., Zheng, C., Ghosh, S., Nielson, D., Thomas, A., Bandettini, P., and Pereira, F. (2019). Knowing what you know in brain segmentation using bayesian deep neural networks. *Frontiers in Neuroinformatics*, 13:67.
- [McKinney et al., 2020] McKinney, S. M., Sieniek, M., Godbole, V., Godwin, J., Antropova, N., Ashrafiyan, H., Back, T., Chesus, M., Corrado, G. S., Darzi, A., Etemadi, M., Garcia-Vicente, F., Gilbert, F. J., Halling-Brown, M., Hassabis, D., Jansen, S., Karthikesalingam, A., Kelly, C. J., King, D., Ledsam, J. R., Melnick, D., Mostofi, H., Peng, L., Reicher, J. J., Romera-Paredes, B., Sidebottom, R., Suleyman, M., Tse, D., Young, K. C., De Fauw, J., and Shetty, S. (2020). International evaluation of an ai system for breast cancer screening. *Nature*, 577(7788):89–94.
- [Müller et al., 2022] Müller, D., Soto-Rey, I., and Kramer, F. (2022). Towards a guideline for evaluation metrics in medical image segmentation.
- [Morozov et al., 2020] Morozov, S., Andreychenko, A., Pavlov, N., Vladzimyrskyy, A., Ledikhova, N., Gombolevskiy, V., Blokhin, I., Gelezhe, P., Gonchar, A., and Chernina, V. (2020). Mosmeddata: Chest ct scans with covid-19 related findings dataset. *arXiv preprint arXiv:2005.06465*, <https://mosmed.ai/datasets/covid191110/>.
- [Murphy and Winkler, 1977] Murphy, A. H. and Winkler, R. L. (1977). Reliability of subjective probability forecasts of precipitation and temperature. *Journal of the Royal Statistical Society Series C: Applied Statistics*, 26(1):41–47.

- [Neal, 2012] Neal, R. M. (2012). *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.
- [O'Malley et al., 2019] O'Malley, T., Bursztein, E., Long, J., Chollet, F., Jin, H., Invernizzi, L., et al. (2019). Keras Tuner. <https://github.com/keras-team/keras-tuner>.
- [Pleiss et al., 2017] Pleiss, G., Raghavan, M., Wu, F., Kleinberg, J., and Weinberger, K. Q. (2017). On fairness and calibration. *Advances in neural information processing systems*, 30.
- [Riquelme et al., 2018] Riquelme, C., Tucker, G., and Snoek, J. (2018). Deep bayesian bandits showdown: An empirical comparison of bayesian deep networks for thompson sampling. *arXiv preprint arXiv:1802.09127*.
- [Sharrock et al., 2021] Sharrock, M. F., Mould, W. A., Ali, H., Hildreth, M., Awad, I. A., Hanley, D. F., and Muschelli, J. (2021). 3d deep neural network segmentation of intracerebral hemorrhage: Development and validation for clinical trials. *Neuroinformatics*, 19(3):403–415.
- [Shen et al., 2017] Shen, D., Wu, G., and Suk, H.-I. (2017). Deep learning in medical image analysis. *Annual Review of Biomedical Engineering*, 19(1):221–248. PMID: 28301734.
- [Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.
- [Solovyev et al., 2022] Solovyev, R., Kalinin, A. A., and Gabruseva, T. (2022). 3d convolutional neural networks for stalled brain capillary detection. *Computers in Biology and Medicine*, 141:105089.
- [Vasilev and D'yakonov, 2023] Vasilev, R. and D'yakonov, A. (2023). Calibration of neural networks.
- [Virtanen et al., 2020] Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Millman, K. J., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C. J., Polat, İ., Feng, Y., Moore, E. W., VanderPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and SciPy 1.0 Contributors (2020). SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272.
- [Wen et al., 2018] Wen, Y., Vicol, P., Ba, J., Tran, D., and Grosse, R. (2018). Flipout: Efficient pseudo-independent weight perturbations on mini-batches.
- [Zhu and Rohwer, 1995] Zhu, H. and Rohwer, R. (1995). Information geometric measurements of generalisation. Workingpaper, Aston University.

6. ABREVIACIONES

- NN: Red Neuronal (NN; Neural Network, por sus siglas en inglés)
- DNN: Red Neuronal Profunda (DNN; Deep Neural Network, por sus siglas en inglés)
- BNN: Red Neuronal Bayesiana (BNN, Bayesian Neural Network, por sus siglas en inglés)
- MNF: Flujos de normalización multiplicativos (MNF; Multiplicative Normalizing Flows, por sus siglas en inglés.)
- CNN: Red Neuronal Convolucional (CNN, Convolutional Neural Network, por sus siglas en inglés)

A. ANEXOS

A.1. Descripción modelos evaluados

Las tablas relacionadas en este anexo describen brevemente el detalle de cada modelo implementado durante el proyecto.

A.1 Descripción modelos evaluados

A ANEXOS

Nombre Modelo	Nombre del notebook	Dimensión	Tipo	Descripción	Ventana	Tasa de Aprendizaje	Epochs	Batch Size
Keras_Arch_3D_W1_V1	2_W1_Modelo_V1.ipynb	3D	Determinista	Modelo con arquitectura tomada del ejercicio de Keras para redes neuronales 3D.	W1	0,001	100	2
Keras_Arch_3D_W1_V2	2_W1_Modelo_V2.ipynb	3D	Determinista	Modelo con arquitectura tomada del ejercicio de Keras para redes neuronales 3D + Implementación de técnicas de Data Augmentation	W1	0,0001	100	2
Resnet18_Arch_3D_W1_V4	2_W1_Modelo_V4.ipynb	3D	Determinista	Implementación del modelo Resnet18 del paquete classification.models_3D y añadiendo una capa GlobalMaxPooling3D	W1	0,0001	100	2
Resnet18_Arch_3D_W1_V5	2_W1_Modelo_V5.ipynb	3D	Determinista	Implementación del modelo Resnet18 del paquete classification.models_3D y añadiendo una capa GlobalMaxPooling3D	W1	0,001	100	2
Resnet18_Arch_3D_W1_V6	2_W1_Modelo_V6.ipynb	3D	Determinista	Implementación del modelo Resnet18 del paquete classification.models_3D y añadiendo una capa GlobalAveragePooling3D	W1	0,0001	100	2
Keras_Arch_3D_W2_V1	2_W2_Modelo_V1.ipynb	3D	Determinista	Modelo con arquitectura tomada del ejercicio de Keras para redes neuronales 3D.	W2	0,0001	100	2
Keras_Arch_3D_W3_V1	2_W3_Modelo_V1.ipynb	3D	Determinista	Modelo con arquitectura tomada del ejercicio de Keras para redes neuronales 3D.	W3	0,0001	100	2
Keras_Arch_3D_W4_V1	2_W4_Modelo_V1.ipynb	3D	Determinista	Modelo con arquitectura tomada del ejercicio de Keras para redes neuronales 3D.	W4	0,0001	100	2
Keras_Arch_3D_W4_V1_Confirmacion	2_W4_Modelo_V1_Confirmado.ipynb	3D	Determinista	Modelo con arquitectura tomada del ejercicio de Keras para redes neuronales 3D. (Se ejecuta por 2da vez para confirmar valores)	W4	0,0001	100	2
Keras_Arch_3D_W4_V1_Optimizado	2_W4_Modelo_V1_Optimizado.ipynb	3D	Determinista	Modelo con arquitectura tomada del ejercicio de Keras para redes neuronales 3D. + Optimización con Keras-Tuner	W4	0,001	60	2
Keras_Arch_3D_W4_V1_Hyper_Reg	2_W4_Modelo_V1_Hiperparametros.ipynb	3D	Determinista	Modelo Optimizado Determinista + Optimización con Keras-Tuner en función de Regularizadores	W4	0,001	60	2
Resnet18_Arch_3D_W4_V4	2_W4_Modelo_V4.ipynb	3D	Determinista	Resnet18	W4	0,0001	100	2
Resnet34_Arch_3D_W4_V5	2_W4_Modelo_V5.ipynb	3D	Determinista	Resnet34	W4	0,0001	100	2
Seresnet18_Arch_3D_W4_V6	2_W4_Modelo_V6.ipynb	3D	Determinista	Seresnet18	W4	0,0001	100	2
Seresnet34_Arch_3D_W4_V7	2_W4_Modelo_V7.ipynb	3D	Determinista	Seresnet34	W4	0,0001	100	2
Resnet18_Arch_3D_W4_V8	2_W4_Modelo_V8.ipynb	3D	Determinista	Resnet18 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D	W4	0,0001	100	2
Resnet18_Arch_3D_W4_V9	2_W4_Modelo_V9.ipynb	3D	Determinista	Resnet18 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D + Filtros Modelo 3D: 8	W4	0,0001	100	2
Resnet18_Arch_3D_W4_V10	2_W4_Modelo_V10.ipynb	3D	Determinista	Resnet18 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D + Filtros Modelo 3D: 16	W4	0,0001	100	2
Resnet18_Arch_3D_W4_V11	2_W4_Modelo_V11.ipynb	3D	Determinista	Resnet18 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D + Filtros Modelo 3D: 32	W4	0,0001	100	2
Resnet18_Arch_3D_W4_V12	2_W4_Modelo_V12.ipynb	3D	Determinista	Resnet18 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D + Filtros Modelo 3D: 8	W4	0,0001	100	6
Resnet18_Arch_3D_W4_V13	2_W4_Modelo_V13.ipynb	3D	Determinista	Resnet18 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D	W4	0,0001	100	4
Resnet34_Arch_3D_W4_V14	2_W4_Modelo_V14.ipynb	3D	Determinista	Resnet34 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D	W4	0,0001	100	4
Seresnet18_Arch_3D_W4_V15	2_W4_Modelo_V15.ipynb	3D	Determinista	Seresnet18 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D	W4	0,0001	100	4
Seresnet34_Arch_3D_W4_V16	2_W4_Modelo_V16.ipynb	3D	Determinista	Seresnet34 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D	W4	0,0001	100	2
EfficientnetB0_Arch_3D_W4_V17	2_W4_Modelo_V17.ipynb	3D	Determinista	EfficientnetB0 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D	W4	0,0001	100	2
Densenet121_Arch_3D_W4_V18	2_W4_Modelo_V18.ipynb	3D	Determinista	Densenet121 + Penúltima capa de la arquitectura 3D será GlobalMaxPooling3D	W4	0,0001	100	2
Keras_Arch_3D_W4_V1_Uncertainty_V1	2_W4_Modelo_V1_Uncertainty_V1.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa Bernoulli — NO TIENE METRICAS	W4	0,001	150	2
Keras_Arch_3D_W4_V1_Uncertainty_V2	2_W4_Modelo_V1_Uncertainty_V2.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa Bernoulli (MEAN) — NO TIENE METRICAS	W4	0,001	150	2
Keras_Arch_3D_W4_V1_Uncertainty_V2_2	2_W4_Modelo_V1_Uncertainty_V2_2.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa Bernoulli (MEAN)	W4	0,0001	300	2
Keras_Arch_3D_W4_V1_Uncertainty_V3	2_W4_Modelo_V1_Uncertainty_V3.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa MNFDense Bernoulli — NO TIENE METRICAS	W4	0,001	150	2
Keras_Arch_3D_W4_V1_Uncertainty_V4	2_W4_Modelo_V1_Uncertainty_V4.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa MNFDense Bernoulli (MEAN) — NO TIENE METRICAS	W4	0,001	150	2
Keras_Arch_3D_W4_V1_Uncertainty_V4_2	2_W4_Modelo_V1_Uncertainty_V4_2.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa MNFDense Bernoulli (MEAN)	W4	0,0001	800	2
Keras_Arch_3D_W4_V1_Uncertainty_V4_2C	2_W4_Modelo_V1_Uncertainty_V4_2C.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa MNFDense Bernoulli (MEAN) — Es el mismo modelo V4_2, solo que se ejecutó nuevamente porque los pesos guardados del original estaban corruptos	W4	0,0001	800	2
Keras_Arch_3D_W4_V1_Uncertainty_V5	2_W4_Modelo_V1_Uncertainty_V5.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa DenseLocalParameterization Bernoulli (MEAN)	W4	0,0001	300	2
Keras_Arch_3D_W4_V1_Uncertainty_V6	2_W4_Modelo_V1_Uncertainty_V6.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa DenseLocalParameterization Bernoulli (MEAN)	W4	0,0001	300	2
Keras_Arch_3D_W4_V1_Uncertainty_V7	2_W4_Modelo_V1_Uncertainty_V7.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa DenseReparameterization Bernoulli (MEAN)	W4	0,0001	300	2
Keras_Arch_3D_W4_V1_Uncertainty_V4_3	2_W4_Modelo_V1_Uncertainty_V4_3.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa MNFDense Bernoulli (MEAN) + Capas MNFConv3D	W4	0,0001	300	2
Keras_Arch_3D_W4_V1_Uncertainty_V4_3C	2_W4_Modelo_V1_Uncertainty_V4_3C.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa MNFDense Bernoulli (MEAN) + Capas MNFConv3D	W4	0,0001	300	2
Keras_Arch_3D_W4_V1_Uncertainty_MCDropout	2_W4_Modelo_V1_Optimizado_MCDropout.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Dropout MC	W4	0,0001	100	2
Keras_Arch_3D_W4_V1_Uncertainty_V8	2_W4_Modelo_V1_Uncertainty_V8.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa DenseFlipout Bernoulli (MEAN) + Capas ConvFlipout	W4	0,0001	300	2
Keras_Arch_3D_W4_V1_Uncertainty_V9	2_W4_Modelo_V1_Uncertainty_V9.ipynb	3D	Bayesiano	Keras (Optimizado Tuner) + Capa DenseReparameterization Bernoulli (MEAN) + Capas Conv Reparametrization	W4	0,0001	300	2

67
Cuadro 4: Descripción Modelos 3D

Nombre Modelo	Nombre del notebook	Dimensión	Tipo	Descripción	Ventana	Tasa de Aprendizaje	Epochs	Batch Size
Keras_Arch_W4_V1_Optimizado_2D_MEAN	2_W4_Modelo_Keras_2D_MEAN.ipynb	2D	Determinista	Modelo determinista con arquitectura optimizada + Conversión imágenes 3D a 2D a través del promedio de los pixeles de los slices (3ra dimensión).	W4	0,0001	150	2
Keras_Arch_W4_V1_Optimizado_2D_MEAN_V2	2_W4_Modelo_Keras_2D_MEAN_V2.ipynb	2D	Determinista	Modelo determinista con arquitectura optimizada + Conversión imágenes 3D a 2D a través del promedio de los pixeles de los slices (3ra dimensión).	W4	0,001	150	2
Keras_Arch_W4_V1_Optimizado_2D_MEAN_V3	2_W4_Modelo_Keras_2D_MEAN_V3.ipynb	2D	Determinista	Modelo determinista con arquitectura optimizada + Conversión imágenes 3D a 2D a través del promedio de los pixeles de los slices (3ra dimensión).	W4	0,00001	150	2
Keras_Arch_W4_V1_Optimizado_2D_MEAN_V2_V1	2_W4_Modelo_Keras_2D_MEAN_V2_V1.ipynb	2D	Determinista	Modelo determinista con arquitectura optimizada + Conversión imágenes 3D a 2D a través del promedio de los pixeles de los slices (3ra dimensión).	W4	0,001	150	4
Keras_Arch_W4_V1_Optimizado_2D_MEAN_V2_V2	2_W4_Modelo_Keras_2D_MEAN_V2_V2.ipynb	2D	Determinista	Modelo determinista con arquitectura optimizada + Conversión imágenes 3D a 2D a través del promedio de los pixeles de los slices (3ra dimensión).	W4	0,001	150	8
Keras_Arch_W4_V1_Optimizado_2D_SUM	2_W4_Modelo_Keras_2D_SUM.ipynb	2D	Determinista	Modelo determinista con arquitectura optimizada + Conversión imágenes 3D a 2D a través de la suma de los pixeles de los slices (3ra dimensión).	W4	0,0001	150	2
Keras_Arch_W4_V1_Optimizado_2D_MAX	2_W4_Modelo_Keras_2D_MAX.ipynb	2D	Determinista	Modelo determinista con arquitectura optimizada + Conversión imágenes 3D a 2D a través del máximo de los pixeles de los slices (3ra dimensión).	W4	0,0001	150	2
MNF_Arch_W4_V1_Optimizado_2D_MEAN	2_W4_Modelo_MNF_2D_MEAN.ipynb	2D	Bayesiano	Modelo bayesiano MNF + Conversión imágenes 3D a 2D a través del promedio de los pixeles de los slices (3ra dimensión).	W4	0,0001	300	2
MNF_Arch_W4_V1_Optimizado_2D_MEAN_V2	2_W4_Modelo_MNF_2D_MEAN_V2.ipynb	2D	Bayesiano	Modelo bayesiano MNF + Conversión imágenes 3D a 2D a través del promedio de los pixeles de los slices (3ra dimensión).	W4	0,001	300	2
MNF_Arch_W4_V1_Optimizado_2D_MEAN_V3	2_W4_Modelo_MNF_2D_MEAN_V3.ipynb	2D	Bayesiano	Modelo bayesiano MNF + Conversión imágenes 3D a 2D a través del promedio de los pixeles de los slices (3ra dimensión).	W4	0,00001	300	2
MNF_Arch_W4_V1_Optimizado_2D_MEAN_V4	2_W4_Modelo_MNF_2D_MEAN_V4.ipynb	2D	Bayesiano	Modelo bayesiano MNF + Conversión imágenes 3D a 2D a través del promedio de los pixeles de los slices (3ra dimensión).	W4	0,0001	300	4
MNF_Arch_W4_V1_Optimizado_2D_MEAN_V5	2_W4_Modelo_MNF_2D_MEAN_V5.ipynb	2D	Bayesiano	Modelo bayesiano MNF + Conversión imágenes 3D a 2D a través del promedio de los pixeles de los slices (3ra dimensión).	W4	0,0001	300	8
MNF_Arch_W4_V1_Optimizado_2D_SUM	2_W4_Modelo_MNF_2D_SUM.ipynb	2D	Bayesiano	Modelo bayesiano MNF + Conversión imágenes 3D a 2D a través de la suma de los pixeles de los slices (3ra dimensión).	W4	0,0001	300	2
MNF_Arch_W4_V1_Optimizado_2D_MAX	2_W4_Modelo_MNF_2D_MAX.ipynb	2D	Bayesiano	Modelo bayesiano MNF + Conversión imágenes 3D a 2D a través del máximo de los pixeles de los slices (3ra dimensión).	W4	0,0001	300	2

Cuadro 5: Descripción Modelos 2D

A.2. Cambios Iniciales Transición Determinista - Bayesiana

El código relacionado en este anexo comprende los cambios realizados a la arquitectura del modelo determinista para iniciar la transición hacia el campo bayesiano.

Listing 1: Capa Final - Modelo determinista:

```
outputs = layers.Dense(units=1, activation="sigmoid")(x)
```

Listing 2: Alternativa para Modelo Bayesiano:

```
# Opcion 1 - A
x = layers.Dense(units=tfp.layers.IndependentBernoulli.params_size(1))(x)
outputs = tfp.layers.IndependentBernoulli(1)(x)

# Opcion 1 - B
x = layers.Dense(units=tfp.layers.IndependentBernoulli.params_size(1))(x)
outputs = tfp.layers.IndependentBernoulli(1, tfd.Bernoulli.mean)(x)

# Opcion 2 - A
x = MNFDense(tfp.layers.IndependentBernoulli.params_size(1))(x)
outputs = tfp.layers.IndependentBernoulli(1)(x)

# Opcion 2 - B
x = MNFDense(tfp.layers.IndependentBernoulli.params_size(1))(x)
outputs = tfp.layers.IndependentBernoulli(1, tfd.Bernoulli.mean)(x)
```

A.3. Métricas Primeras Alternativas Redes Bayesianas

Las gráficas relacionadas en este anexo comprenden la evolución época a época del entrenamiento de los primeros modelos bayesianos.

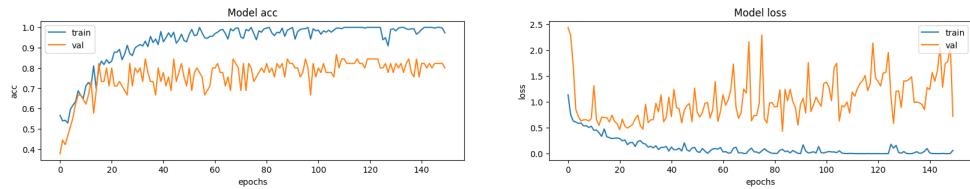


Figura 18: Evolución Red Neuronal Bayesiana - Opción 1A

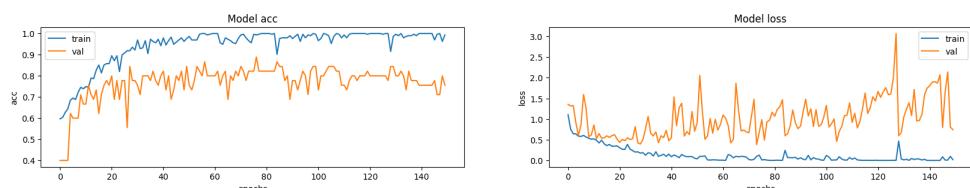


Figura 19: Evolución Red Neuronal Bayesiana - Opción 1B

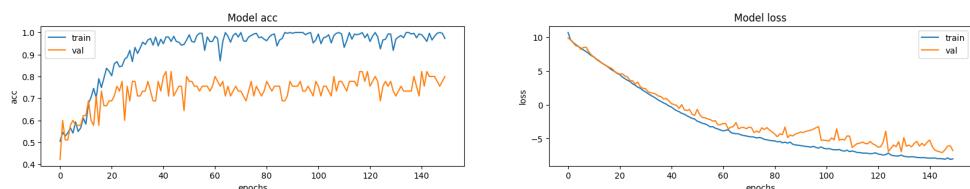


Figura 20: Evolución Red Neuronal Bayesiana - Opción 2A

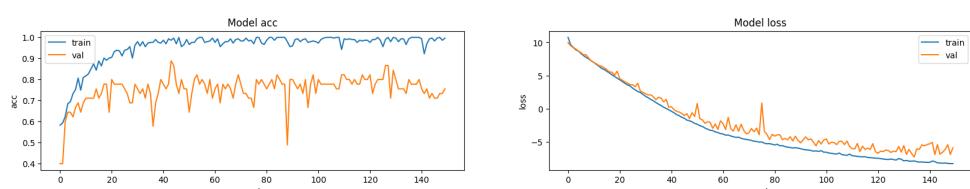


Figura 21: Evolución Red Neuronal Bayesiana - Opción 2B

A.4. Monte Carlo Dropout

El código relacionado en este anexo comprende los cambios realizados a la arquitectura del modelo determinista para entrenar la red bayesiana Monte Carlo Dropout.

Listing 3: Arquitectura determinista

```
def get_model(width=128, height=128, depth=64):
    """Build a 3D convolutional neural network model."""
    inputs = keras.Input((width, height, depth, 1))

    x = layers.Conv3D(filters=128, kernel_size=3, activation="relu")(inputs)
    x = layers.MaxPool3D(pool_size=2)(x)
    x = layers.BatchNormalization()(x)

    x = layers.Conv3D(filters=128, kernel_size=3, activation="relu")(x)
    x = layers.MaxPool3D(pool_size=2)(x)
    x = layers.BatchNormalization()(x)

    x = layers.Conv3D(filters=256, kernel_size=3, activation="relu")(x)
    x = layers.MaxPool3D(pool_size=2)(x)
    x = layers.BatchNormalization()(x)

    x = layers.Conv3D(filters=512, kernel_size=3, activation="relu")(x)
    x = layers.MaxPool3D(pool_size=2)(x)
    x = layers.BatchNormalization()(x)

    x = layers.GlobalMaxPooling3D()(x)
    x = layers.Dense(units=256, activation="relu")(x)
    x = layers.Dropout(0.2)(x)

    outputs = layers.Dense(units=1, activation="sigmoid")(x)

    # Define the model.
    model = keras.Model(inputs, outputs, name="3dcnn")
    return model

# Build model.
model = get_model(width=128, height=128, depth=64)
model.summary()
```

Listing 4: Modificamos la capa dropout:

```
x = layers.Dropout(0.2)(x)
```

Listing 5: Capa modificada:

```
x = layers.Dropout(0.2)(x, training = True)
```

A.5. Coordenadas Paralelas - Ajuste Capa Sigmoid

Las gráficas relacionadas en este anexo comprenden:

Fig.(22): Impacto en las métricas para la red bayesiana en cuestión luego de cambiar la capa de salida probabilística por una capa *Sigmoid*.

Fig.(23): Impacto en las métricas para la red bayesiana en cuestión luego de cambiar la capa de salida probabilística por una capa *Sigmoid*.

Fig.(24): Impacto en las métricas para la red bayesiana en cuestión luego de cambiar la capa de salida probabilística por una capa *Sigmoid*.

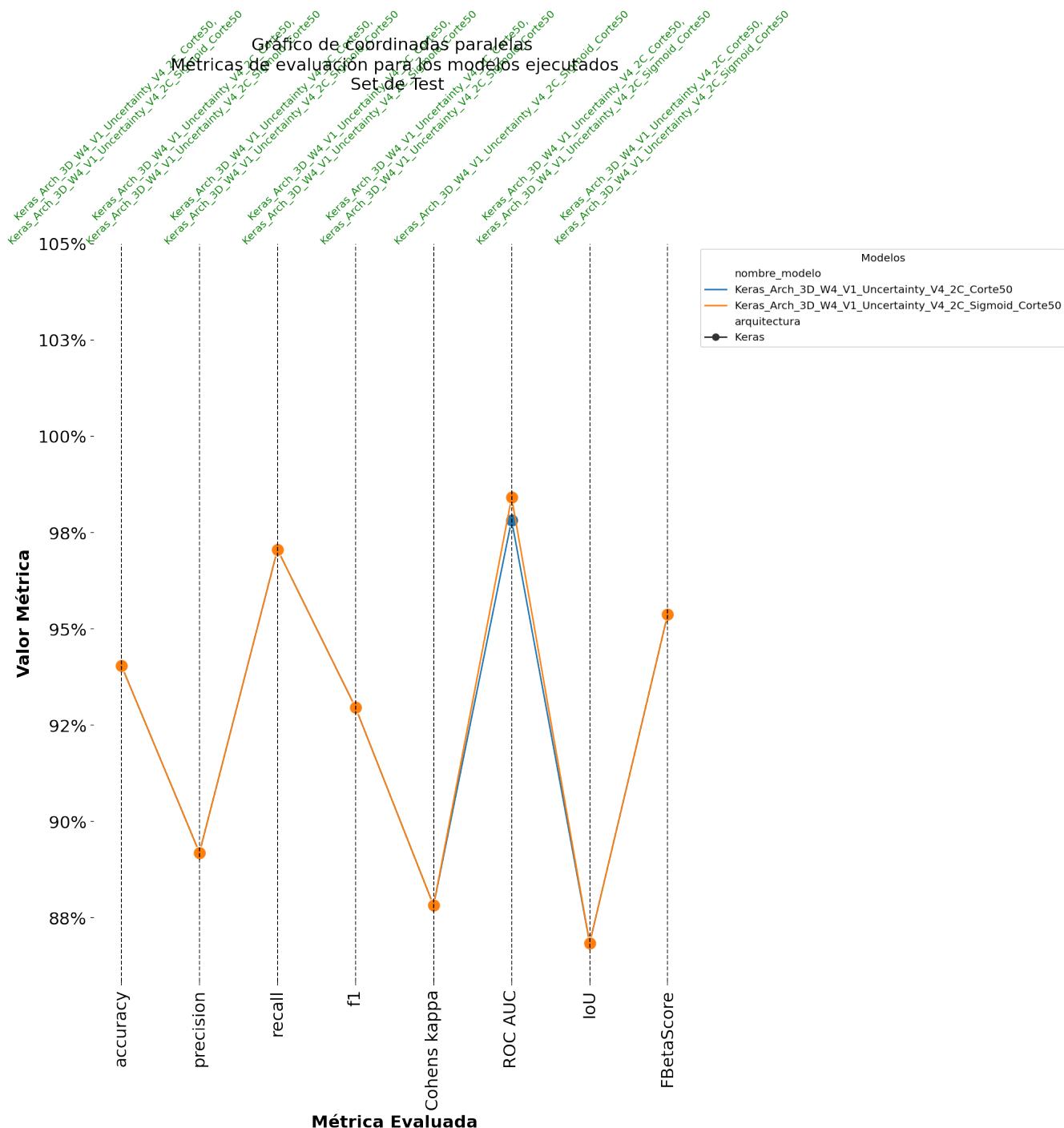


Figura 22: Contaste de métricas evaluadas en los modelos Bayesianos - Set de Test

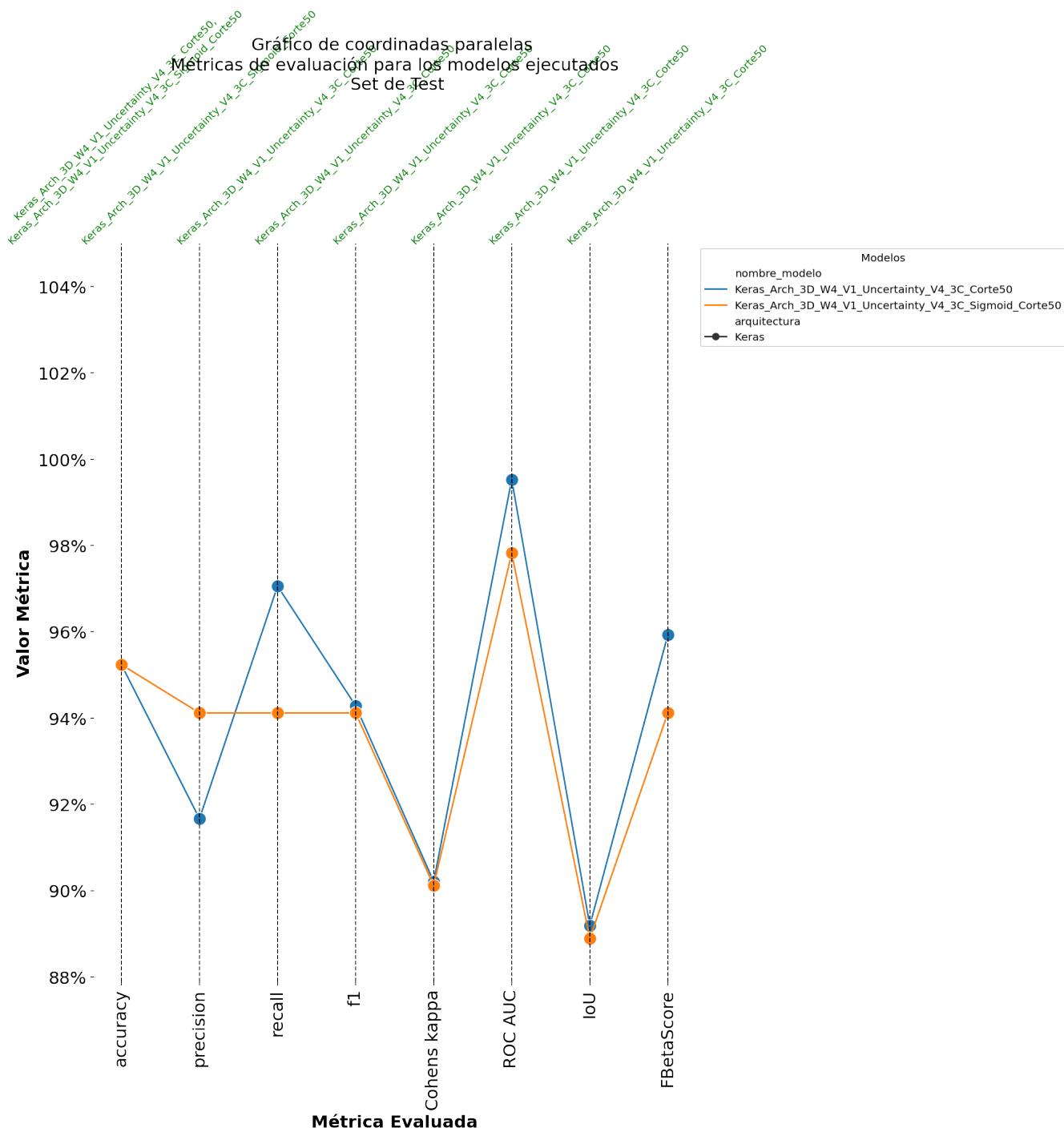


Figura 23: Contaste de métricas evaluadas en los modelos Bayesianos - Set de Test

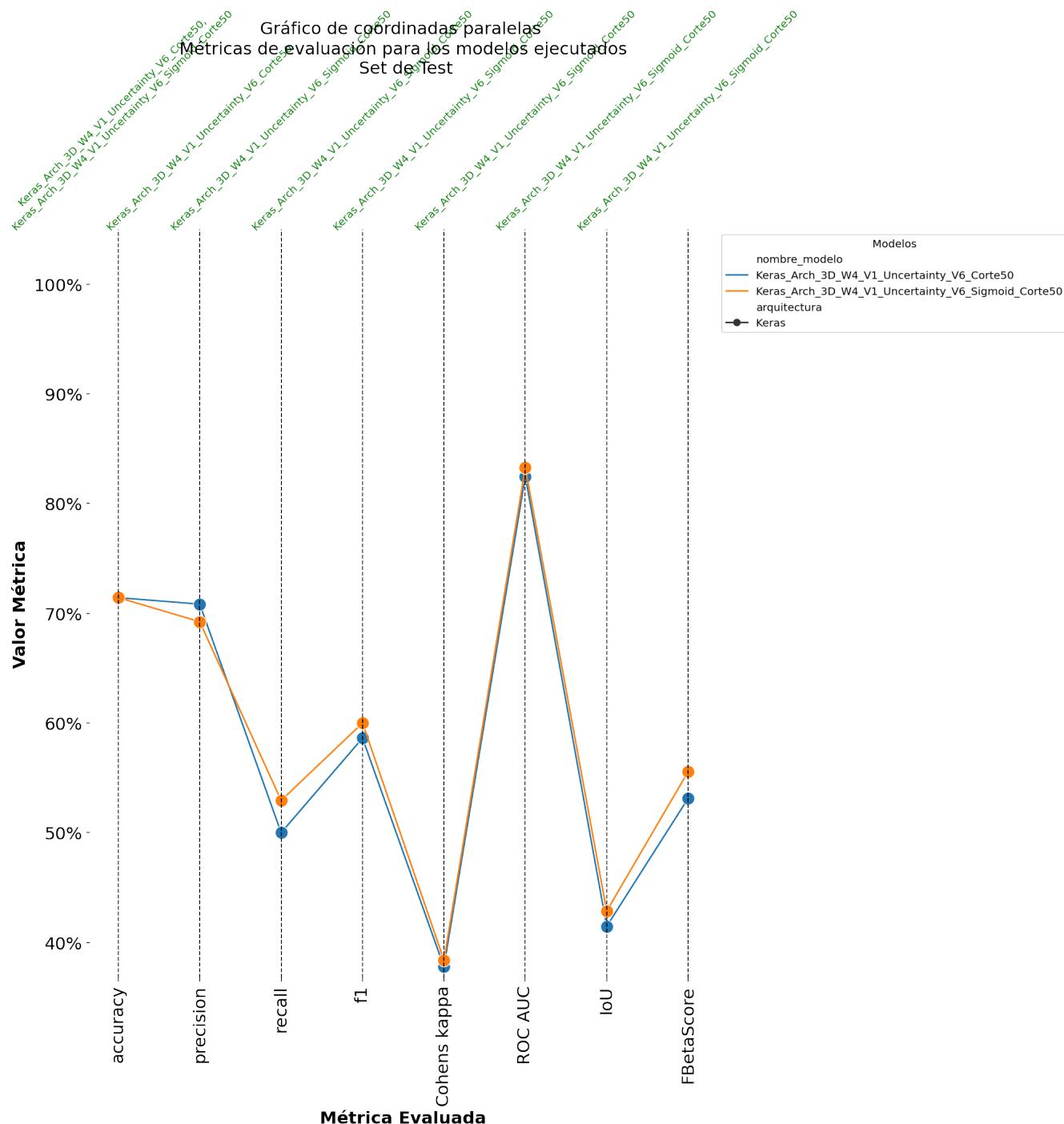


Figura 24: Contaste de métricas evaluadas en los modelos Bayesianos - Set de Test

A.6. Coordenadas Paralelas - Ajuste Umbral

La gráfica relacionada en este anexo comprende el impacto en las métricas para los diferentes modelos bayesianos luego de cambiar el umbral de decisión para la clase 1.

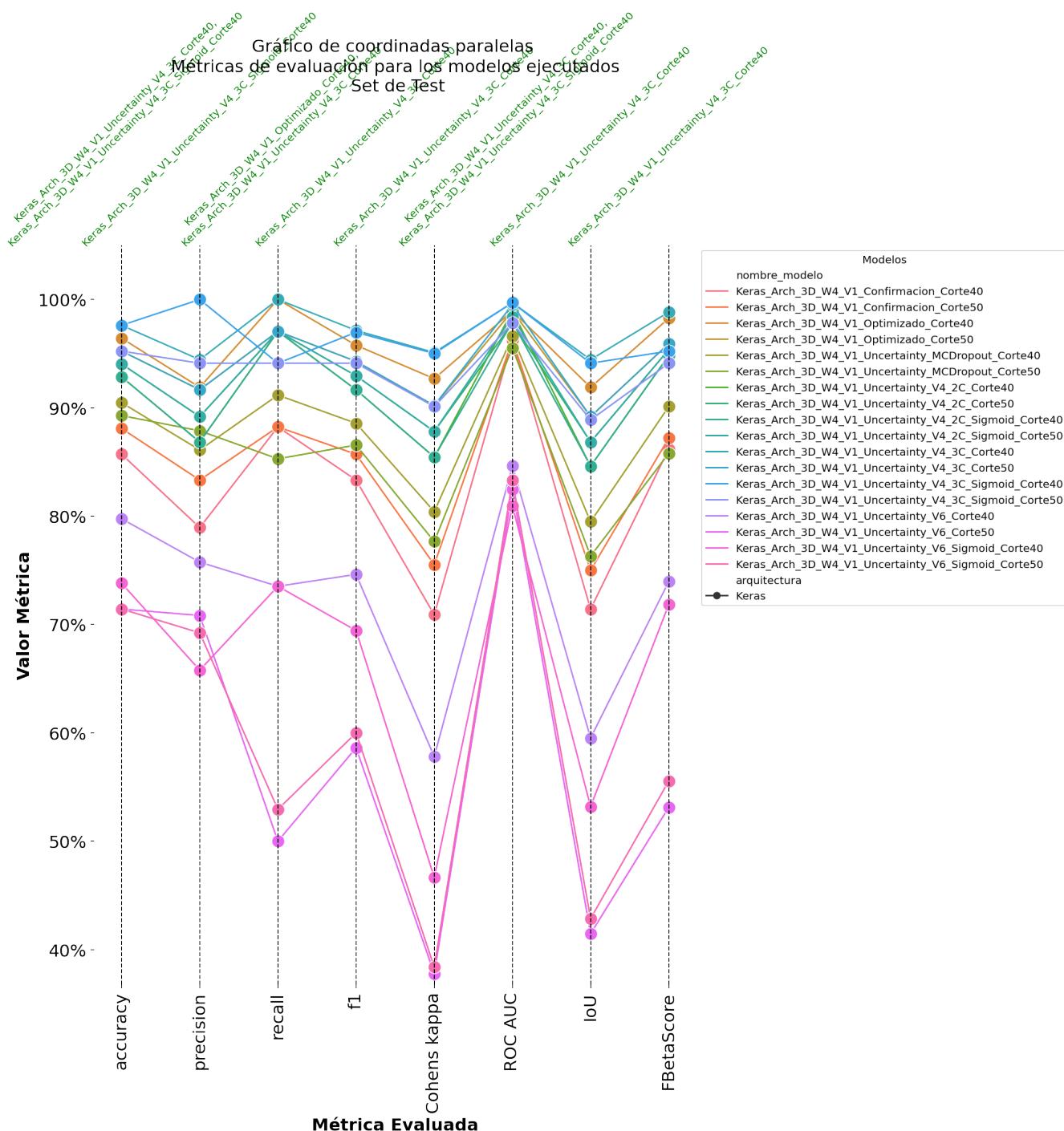


Figura 25: Contaste de métricas evaluadas en los modelos Bayesianos - Set de Test

A.7. Diagramas Confiabilidad

En este anexo presentamos los diagramas de confiabilidad en el set de datos de test para todos los modelos relacionados en la sección 4.3.2.

A.7 Diagramas Confiabilidad

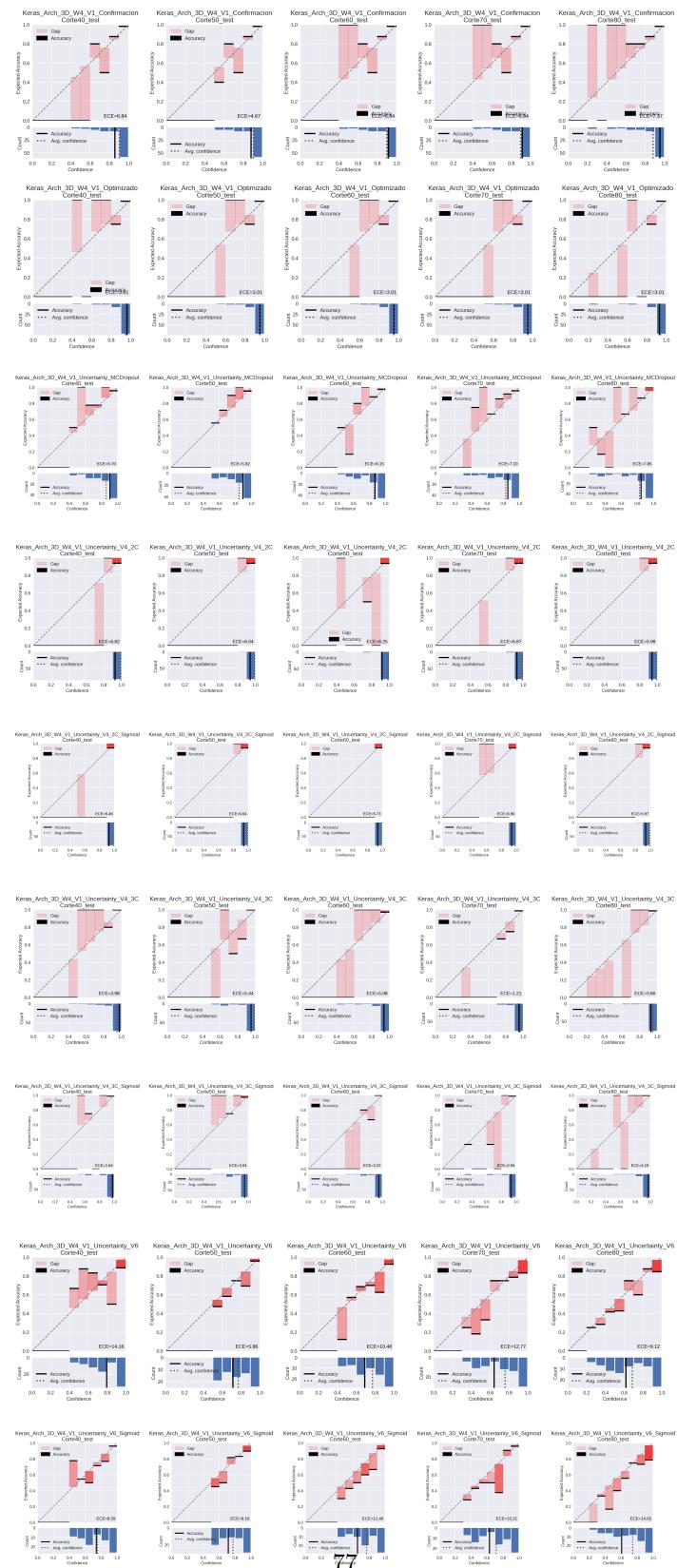


Figura 26: Diagramas Confiabilidad - Set de Test