Redes Neuronales Convolucionales para la Detección de Infarto de Miocardio en Electrocardiogramas

Reporte de finalización de la estancia de investigación

Autor:

Miguel Calvo Valente

Supervisor:

Edgar Francisco Román Rangel

Maestría en Ciencia de Datos
Instituto Tecnológico Autónomo de México
Ciudad de México

12 de marzo del 2023

Tabla de Contenidos

1 Contexto de la organización

Hay diversas organizaciones involucradas en el proyecto:

Astra Vision: *Startup* mexicana enfocada en implementar inteligencia artificial en el sector médico. El proyecto con mayor impacto que han desarrollado consiste en un algoritmo de redes neuronales convolucionales (*CNN*, por sus siglas en inglés) para identificar la presencia de COVID-19 en pacientes a través de la lectura de radiografías. Este mismo fue implementado en *Amazon Web Services* (*AWS*), y ha tenido gran impacto en comunidades rurales donde no hay suficiente personal capacitado para atender los frecuentes casos que se presentan. Las personas en *Astra Vision* nos ayudarán a coordinar esfuerzos entre todas las organizaciones.

Indiana University (IU): Universidad estadounidense que cuenta con una gran diversidad de programas, desde tecnología, ciencia y medicina, hasta artes y humanidades. Particularmente cuentan con una maestría en Ciencia de Datos, con cuyos administradores *Astra Vision* ha trabajado de la mano en el proyecto de radiografías para la identificación de COVID-19. Los alumnos de la *IU* nos ayudarán investigando el estado del arte del proyecto en cuestión, así como corriendo diversos modelos de aprendizaje profundo.

Instituto Nacional de Cardiología - Ignacio Chávez (*INC*): Institución mexicana de salud e investigación con especialidad en cardiología que pertenece a la Secretaría de Salud. Es parte de los 13 institutos de investigación en ciencias biomédicas que conforman los Institutos Nacionales de Salud en México. Los doctores del Instituto nos apoyarán con su conocimiento médico en el proyecto, explicándonos conceptos y validando la calidad de los datos.

Instituto Tecnológico Autónomo de México (*ITAM*): Universidad mexicana que cuenta con un enfoque científico, técnico y humanista en la formación de sus alumnos. El Dr. Edgar Francisco Román Rangel participó en el proyecto de radiografías y COVID-19. Estaré trabajando con él y las demás organizaciones en este proyecto; de mi lado, desde la parte de la exploración de los datos hasta el modelado de las redes.

2 Descripción del problema a resolver desde un punto de vista del negocio

Astra Vision actualmente tiene un proyecto en conjunto con el *INC*, la *IU* y el *ITAM* para la detección de diversos diagnósticos con el uso de modelos de aprendizaje profundo a partir de electrocardiogramas (*ECG*). Un *ECG* es un procedimiento médico que registra la actividad eléctrica del corazón a través de la colocación de electrodos en la piel del paciente [1]. Esto proporciona información sobre la salud del corazón, lo que ayuda a detectar problemas cardíacos. En el *INC* buscan generar un modelo que prediga los diagnósticos de un *ECG*, para que pueda servir como un primer dictamen automático para auxiliar al médico en la toma de decisiones.

El contexto actual es que la precisión que tienen los médicos para diagnosticar a partir de un *ECG* varía dependiendo de su grado de especialidad [2]. Mientras que los cardiólogos hacen diagnósticos certeros, los médicos menos especializados, como los generales, de urgencia, enfermeros, residentes, entre otros, usualmente no tienen los conocimientos necesarios para detectar diversos patrones. Esto es grave ya que muchos hospitales o centros de salud no cuentan con la cantidad de cardiólogos necesarios para evaluar el flujo constante de pacientes.

Para solventar esto, el proyecto busca dar un diagnóstico indicando el riesgo de tener cada uno de los padecimientos en cuestión. Con esto, los médicos no especializados pueden apoyarse para tomar decisiones, al combinar su conocimiento con el resultado del modelo predictivo. Dentro de los múltiples diagnósticos que se buscan poder detectar, el de mayor importancia y el cual se contempla en este reporte es el de los infartos de miocardio (*MI*, por sus siglas en inglés). Estos son obstrucciones totales de las arterias que bloquean el flujo de la sangre al resto del cuerpo, y es una de las principales causas de muerte en el mundo[3].

3 Metas y objetivos a alcanzar

Se busca generar predicciones de diagnósticos que en promedio sean al menos superiores a las del médico no especializado, crear una interfaz gráfica fácil de ocupar para auxiliar al médico, y eventualmente poner en producción el *software* para que diversos hospitales y centros de salud lo ocupen. Este proyecto tiene una extensión de 2 a 3 años para cumplir con todos los objetivos. No obstante, como entregable para la estancia de investigación, al momento se pondrá enfasis en la exploración de los datos para realizar tratamientos adecuados, y el crear y mejorar los modelos predictivos.

Las metas de la exploración y el trataminto incluyen el entender qué es un *ECG*, como se ven las señales que lo componen, comprender como se pueden manipular para limpiar el ruido que contengan, y realizar dicha limpieza. Las metas del modelaje son crear *CNN* que aprovechen la naturaleza repetitiva de los *ECG* para que los filtros aprendan patrones, agregar diferentes elementos, como capas de *Batch Normalization*, *Dropout*, modificación de los pesos de

clase, entre otros, para mejorar la capacidad predictiva, y refinar el modelo con los elementos que muestren tener un mayor impacto positivo en las métricas de evaluación.

4 Situación actual con respecto a la problemática.

El análisis de un *ECG* es frecuentemente la primera linea de defensa al detectar diagnósticos cardíacos en pacientes. Si se detecta adecuadamente un diagnóstico que requiere atención urgente, se puede proceder inmediatamente a realizar los estudios consecuentes pertinentes para determinar el tratamiento o llevar a cabo una cirugía de ser necesaria. No obstante, en muchas ocasiones los hospitales y centros médicos no cuentan con el suficiente personal capacitado para la lectura correcta del *ECG*, lo que lleva a una capacidad de reacción lenta frente a diversos padecimientos y en consecuencia a una mortalidad elevada.

Los algoritmos de aprendizaje de máquina, particularmente de aprendizje profundo, han tomado fuerza y popularidad en recientes años. Esto es en gran parte debido al acceso universal a máquinas, personales o en la nube, con alta capacidad de procesamiento y memoria. El triunfo que han logrado estos algoritmos se ha visto en diversos campos, como las redes sociales, las finanzas, el transporte, entre otros [4]. La gran ventaja de los modelos de aprendizaje de máquina yace en la velocidad y automatización de generar predicciones una vez que fue entrenado con un conjunto de datos.

La intersección entre el sector salud y el aprendizaje de máquina igualmente ha mostrado resultados prometedores en los últimos años. No obstante, su aceptación universal ha sido difícil debido a diversos factores [5], como la falta de interpretabilidad, la heterogeneidad en los datos, adaptarse a diagnósticos desconocidos, entre otros. El objetivo de los modelos que se desarrollaron no es sustituir al médico, sino servir como una herramienta adicional para auxiliar en el dictamen final del diagnóstico.

5 Planteamiento del problema desde la perspectiva de un científico de datos

El problema en un alto nivel consiste en generar un modelo de clasificación a partir de un conjunto de *ECG*. Esto a su vez involucra entender los datos tanto para limpiarlos como para elegir un modelo adecuado. Los conjuntos de datos tradicionales son de forma tabular, en los que cada columna representa una variable y cada renglón un registro. No es así el caso para los *ECG*. Para un *ECG* de una señal, se cuenta con una serie de tiempo con patrones de elevación y descenso de acuerdo a las señales eléctricas del corazón, registradas por un dispositivo de eventos cardíacos [6].

La base de datos con la que se están entrenando los modelos es la *PTB-XL* [7][8], una base de *ECG* públca con una gran diversidad de diagnósticos y agrupamientos de los mismos. Esta cuenta con 21,837 electrocardiogramas digitales recopilados por el *Physikalisch-Technische Bundesanstalt (PTB)*, el instituto nacional de metrología de Alemania. Todos los *ECG* son de 12 señales, con 10 segundos de duración, y con información referente a la edad, sexo y más importantemente, el diagnóstico del paciente.

Es por ello que un modelo sensible para aprovechar la estructura temporal y repetitiva de los *ECG* es una red neuronal convolucional (*CNN*). Estos modelos son usualmente ocupados en problemas de imagenes, ya que van aprendiendo filtros que a su vez reconocen patrones en distintos lugares de la imagen. Es decir, explotan las características locales de los datos. Por ejemplo, una *CNN* que busque determinar si la imagen contiene un pájaro muy probablemente aprenderá a reconocer los patrones del pico, las alas, la cola, etc. desde diferentes ángulos, posiciones y tamaños. En el contexto de los *ECG*, la red buscará elevaciones o depresiones en la señal, amplitudes, repetición de patrones, distancias entre los latidos del corazón, etc. para determinar el diagnóstico del paciente.

6 Métricas de evaluación de impacto

Medimos el desempeño del modelo con base en 3 métricas: sensibilidad, especificidad y precisión. La primera indica el porcentaje de *MI* correctamente clasificados, la seguna el porcentaje de no *MI* correctamente clasificados, y la tercera el porcentaje que indica con que frecuencia un *ECG* que se clasifica como *MI* realmente lo es.

En diversas aplicaciones del sector salud, es muy importante incrementar la sensibilidad sobre la especificidad. Esto ya que el no identificar a alguien que sí tiene un diagnóstico puede llevar a la muerte del paciente. Determinar que alguien tiene un diagnóstico positivo cuando no es el caso también es importante debido a que se puede incurrir en pérdida de tiempo para atender a pacientes que sí lo necesiten, así como gastos innecesarios que harían los pacientes para realizar otras pruebas, como estudios de sangre o angiografías[9], para confirmar el diagnóstico. Si bien los objetivos de este trabajo buscan enfocarse en mejorar la sensibilidad, sería ideal tener una buena especificidad y precisión para minimizar la pérdida de recursos tanto del paciente como del establecimiento.

7 Comparativa y mejoras de modelos

En esta sección se hablará de las técnicas ocupadas para el tratamiento de los datos y la construcción de las *CNN*. Si bien ambos fueron ocurriendo en paralelo, en este reporte primero se hablará del tratamiento a manera de que, cuando se hable de los modelos, se tenga el contexto de lo que se le hizo a los datos en busca de mejorar el desempeño predictivo.

7.1 Tratamiento de los datos

En conjuntos de datos tradicionales, lo usual en el tratamiento de los datos incluye el verificar las distribuciones de las variables predictoras, determinar cuáles son numéricas, ordinales o categóricas, realizar análisis de correlaciones o información mutua, imputar valores faltantes, entre muchas más técnicas. Para el tratamiento de los *ECG* no hay una clara forma de cómo proceder. Incluso, se pueden ocupar los datos tal como están al pasarlos al modelo. En esta sección se detallan los procedimientos que se realizaron para tratar los datos en un esfuerzo de ayudar a que el modelo pudiera aprender mejor los patrones de las señales.

La primera complicación al tratar con los *ECG* surgió del desconocimiento de cómo se deberían ver las señales. Los integrantes del proyecto que nos dedicamos a manipular los datos y generar modelos no tenemos educación ni experiencia en el sector médico, y mucho menos en la interpretación de *ECG*. Para entender los datos, se tuvieron sesiones con los doctores del *INC*, en las cuales ellos nos explicaron a grandes rasgos de qué se compone un *ECG* usual, y más importante aún, qué patrones están asociados a ruido y no aportan información con respecto al diagnóstico.

A grandes rasgos, se pueden clasificar los tipos de ruido en 3 categorías:

- *Baseline drift*
- Alta frecuencia
- Otros patrones

Antes de entrar a detalle con los tipos de ruido, vale la pena mostrar cómo se ve una señal sin ninguno de estos [Fig. 1]. Cabe mencionar que al momento no se le está dando enfoque al diagnóstico que se pueda derivar de la señal, sino únicamente al ruido que puedan tener las señales.

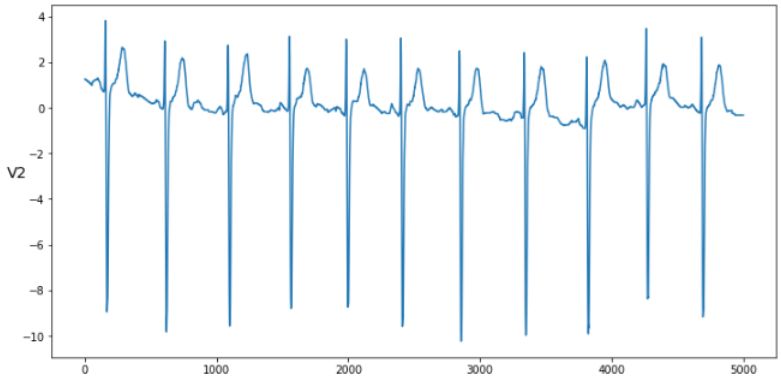


Fig. 1: Ejemplo de una señal sin ningún tipo de ruido.

Los *ECG* por lo usual están centrados alrededor del potencial eléctrico del corazón en reposo, a lo que se le llama el *baseline*. En cada una de las señales, cuando la corriente eléctrica se mueve hacia el electrodo esto se refleja como una desviación positiva del *baseline*, y cuando se aleja ocurre una desviación negativa[10]. El *baseline drift* hace referencia a la separación considerable de los patrones de este valor. Los médicos en general pueden filtrar visualmente esto y fijarse únicamente en los patrones asociados al diagnóstico. No obstante, este es un ruido que puede afectar a la capacidad predictiva de los modelos. El *baseline drift* puede existir de muchas formas y en diversos grados de severidad [Fig. 2], por lo que tratar con él no es tan directo. De lo explorado en los datos, se infiere que el *baseline* en esta base es el 0 (no necesariamente tendría que ser 0 en otras bases).

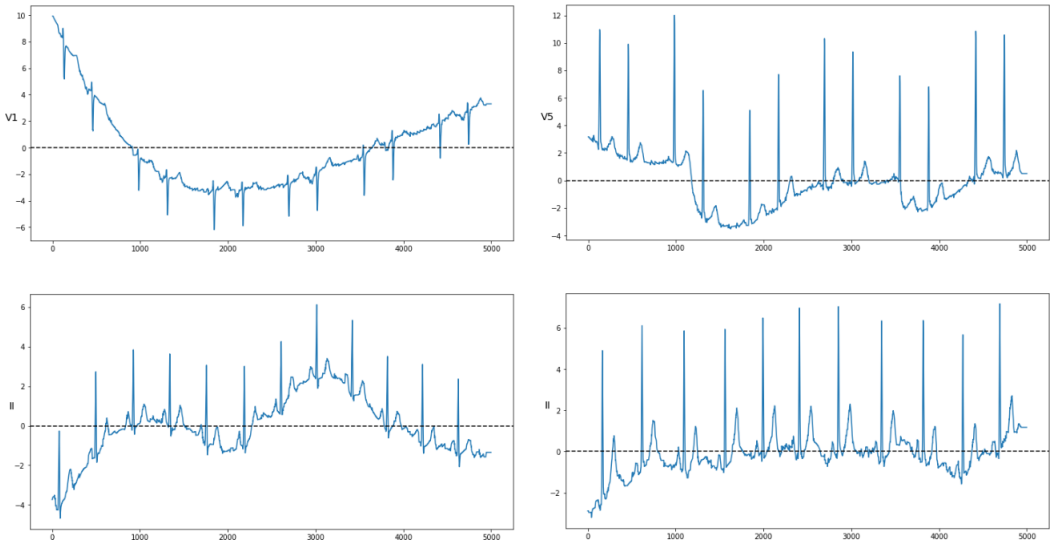


Fig. 2: Ejemplos de señales con *baseline drift*.

El ruido por alta frecuencia se puede apreciar visualmente como si fuera ruido blanco [Fig. 3]. Se le está llamando de esta forma pues, si aplicáramos la transformada de *Fourier* a la señal, habría una gran cantidad de frecuencias altas en comparación con lo usual.

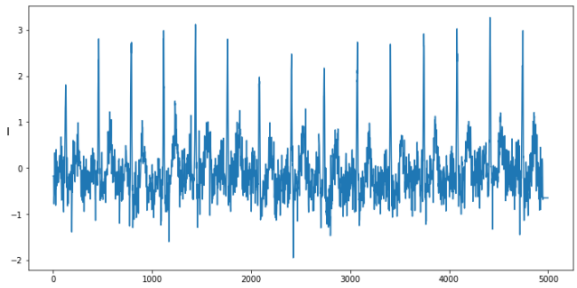


Fig. 3: Ejemplo de una señal con ruido de alta frecuencia.

Los ruidos anteriores son relativamente comunes dentro de las señales. A todos los demás ruidos que se detectaron se les agrupó en la categoría de "otros patrones". Por ejemplo, podrían ser picos inexplicables en algunas de las señales del *ECG*, elevaciones o depresiones inusuales, entre otros [Fig. 4]. Por esto mismo, no existe una manera generalizada de tratar con ellos. En algunos ejemplos, parece que pueda ser en parte ruido de *baseline drift*, por lo que puede ser que corregir esto arregle en parte los de otros patrones

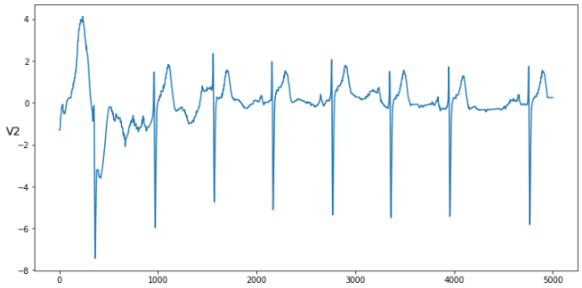


Fig. 4: Ejemplo de una señal con ruido de otros patrones. El ruido en este caso se nota como una elevación anormal en los primeros segundos.

En este trabajo de investigación se le puso particular enfoque a la corrección del *baseline drift* y de los otros patrones, y con los datos corregidos se procedió a entrenar diversos modelos. La corrección del ruido de alta frecuencia no se ha realizado aún, pero la idea en general consiste en aplicar filtros *bandpass* para acotar las señales entre un cierto rango de frecuencias[11], particularmente ajustando el límite superior para retirar las frecuencias altas.

Como se pudo apreciar, el *baseline drift* viene en muchas formas. La idea para corregirlo es que las señales estén centradas en el *baseline*, que en este caso es 0, sin perder información de los patrones asociados al diagnóstico. Esto se podría lograr si contáramos con la componente de tendencia[1][12], tras lo cual bastaría con restar de la señal dicha componente. Como no contamos directamente con tal componente, lo que se busca hacer es aproximarla.

La metodología ocupada para aproximar la componente de tendencia fue aplicar un filtro de mediana en la señal [13]. El filtro de mediana consiste en calcular, para cada punto, la mediana de los puntos adyacentes en una ventana determinada. La razón para ocuparlo es que las medianas, si se elige adecuadamente el valor de la ventana, capturan más la componente de tendencia que los patrones asociados al diagnóstico. Elegir un valor muy bajo hace que la mediana capture demasiado los patrones del *ECG*, por lo que restar el filtro de la señal quitaría tales patrones. Por otra parte, elegir un valor muy alto hace que no se capture adecuadamente la tendencia, por lo que la señal que resulta tras restar el filtro aún contiene tendencia [Fig. 5]. No fue trivial elegir este valor, pues existe un compromiso entre el valor que se obtiene de la señal filtrada y la información que se pierde por lo mismo.

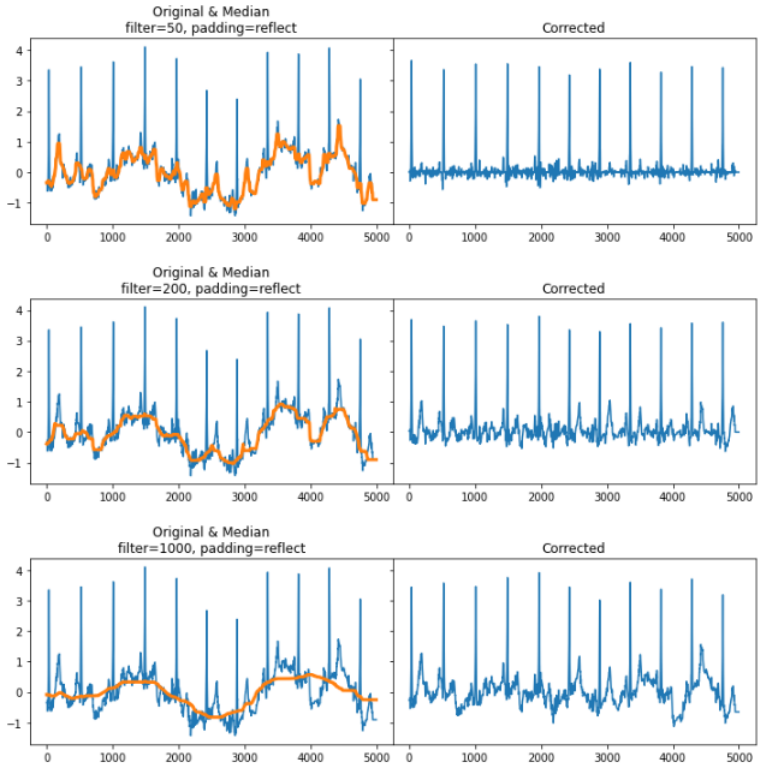


Fig. 5: Aplicación del filtro de mediana con 3 ventanas diferentes. *filter* es el tamaño de la ventana. *padding* hace referencia a cómo se calcularon las medianas para los puntos en los extremos. Por ejemplo, si el tamaño de la ventana es 20 y se quiere calcular la mediana del punto número 1, entonces sólo hay 10 valores a la derecha y ninguno a la izquierda. En este caso, *padding=reflect* significa que se reflejaron los 10 puntos hacia la izquierda para el cálculo de la mediana.

Para elegir el valor de la ventana, se tuvieron sesiones con los doctores del *INC*, en las cuales les mostramos varios ejemplos de señales antes de filtrar y filtradas, con diversos valores de ventana. Ellos nos dieron su opinión de qué tanta información se perdía conforme se reducía el tamaño de ventana. Si bien lo ideal sería aplicar diferentes ventanas a cada señal para corregirlas de manera óptima, esto no es viable en la práctica. Por ello, nos indicaron que una ventana con valor de 200 era la que, en promedio, mejor filtraba las series sin tanta pérdida de información [Fig. 6].

[1]En series de tiempo, la componente de tendencia se define más en términos de si hay cambios lineales o al menos monotónicos a través del tiempo. Se hace un abuso del lenguaje para referirnos en este trabajo a tendencia como todas las desviaciones del *baseline* que no estén asociadas con patrones usuales en *ECG*

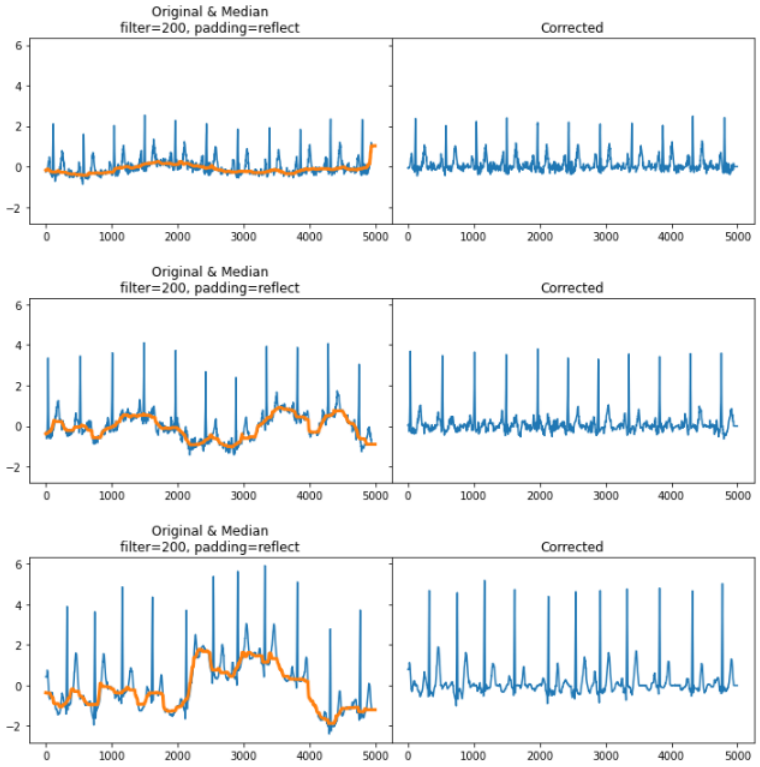


Fig. 6: Aplicación del filtro de mediana con ventana de tamaño 200 a 3 señales. Se aprecia que este tamaño de ventana permite capturar la componente de tendencia en presencia de diversos grados de severidad del *baseline drift*.

Inicialmente, se ocupó una implementación propia, así como la función *median_filter* de *scipy* [14], para el cálculo del filtro de mediana. Con esto se pudieron hacer visualizaciones iniciales para verificar si la metodología funcionaría. Una vez confirmado esto, surgió a la luz un problema: El tiempo para corregir el conjunto entero de datos superaría 4 días continuos de procesamiento. Si bien se puede dejar un servidor que corra durante todo ese tiempo, se contempló investigar si habría una forma más eficaz, particularmente considerando si en futuros trabajos se quisiera cambiar el tamaño de ventana.

Dado que se estará ocupando *tensorflow*[15] para la implementación de las *CNN*, se optó por ocupar la función *median_filter2d* de la paquetería *TensorFlow Addons* [16], en conjunto con una unidad de procesamiento gráfico (*GPU*, por sus siglas en inglés). Se verificó que diera el mismo resultado que los experimentos hechos con las otras metodologías. Ocupar esta función junto con la *GPU* redujó el tiempo de más de 4 días a alrededor de 1-2 horas.

El valor de 200 que se eligió fue determinado por los doctores tras ver algunos ejemplos. Ellos mismos afirmaron que en algunos casos este valor no necesariamente sería el mejor, pero que en promedio parecía ser el que tendría la menor pérdida de información. Con esto en mente, cabe la duda de cómo se podría corroborar de forma analítica y automatizada (sin necesidad de la evaluación de un médico) el que una señal haya sido filtrada adecuadamente. Con un criterio analítico, se podría medir el ajuste, y con ello se podrían considerar a los registros con un mal ajuste como *outliers*, con la intención de quitarlos para que el modelo no aprenda de registros "sucios".

Una primera propuesta que surgió fue comparar la señal filtrada una vez contra la señal filtrada dos veces. La idea es que, si estas dos señales eran muy diferentes, entonces aún habría un componente de tendencia por eliminar. Se propuso una métrica para medir los errores entre estás dos señales:

$$E(\mathbf{x}_1, \mathbf{x}_2) = \frac{\Sigma_{k=1}^{n}(x_{1,k} - x_{2,k})^2}{\sigma_1^2}$$

Donde:

- $\mathbf{x}_1$: Vector con los valores de la señal tras un filtro de mediana.
- $\mathbf{x}_2$: Vector con los valores de la señal tras dos filtros de mediana.
- σ_1^2 : Varianza de $\mathbf{x}_1$.
- **n** : Cantidad de puntos en la señal (5,000 en todos los casos).

Esta métrica tendrá un valor alto si es que las señales filtradas una y dos veces difieren considerablemente. Es básicamente un error medio cuadrático pero ponderado por la varianza de la señal filtrada una vez (nótese que no se divide por **n** ya que no es necesario, todas las señales tienen 5000 puntos). Se divide por σ_1^2 para garantizar que haya invarianza de escala. Es decir, como las señales tienen diferentes rangos de valores en el eje y (milivoltios, mV), digamos, una puede estar entre [-1mV, 2mV] y otra entre [-10mV, 15mV], el ponderar por varianza garantiza que no se generen valores de E más altos en señales con rangos más amplios. Se generó esta métrica para 4,367 *ECG*, obteniendo así 52,404 (12×4367) valores de E [Fig. 7].

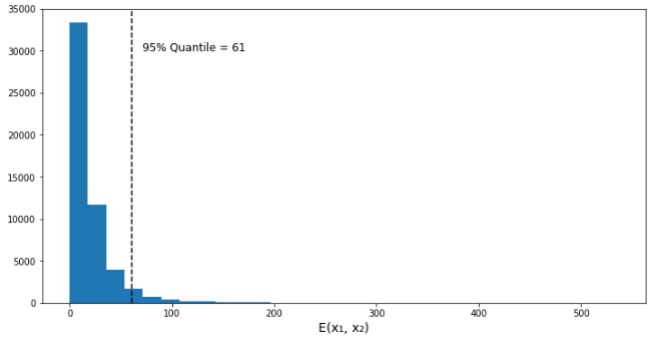


Fig. 7: Distribución de E para un conjunto de *ECG*. Se aprecia que el cuantil 95% tienen un valor de 61, el cual se puede tomar como punto de referencia para determinar si una señal está considerablemente por encima de lo usual.

En Fig. 8 se muestra un ejemplo con un alto valor de E (80.3) en el cual visualmente se aprecia que al realizar una segunda pasada de filtro de mediana aún siguen habiendo componentes de tendencia. O bien, también podríamos considerar que ese es ruido de otros patrones.

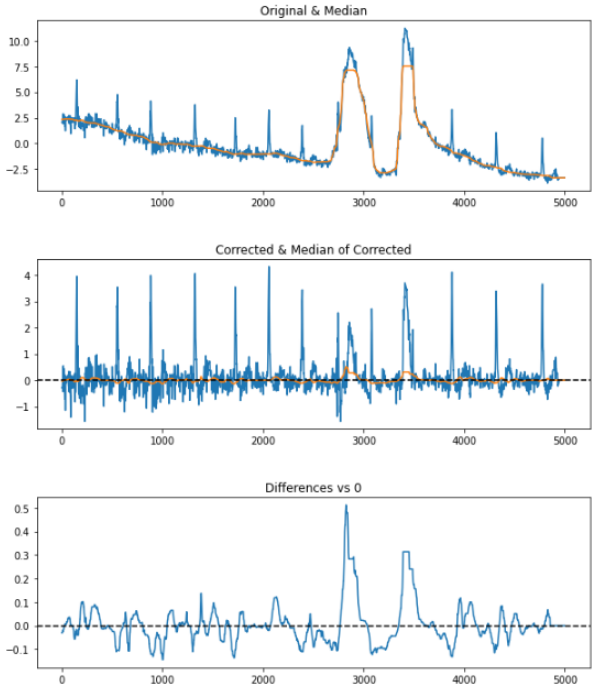


Fig. 8: Ejemplo de una señal cuyo valor de E es alto (80.3). Se muestra la señal original y el filtro de mediana que se restará, la señal tras una iteración del filtro de mediana y el filtro de mediana de esta misma señal, y la diferencia entre una y dos iteraciones del filtro de mediana, en orden. Se aprecia que en el rango del punto 2,800 al 3,800 existen diferencias considerablemente separadas de 0.

Si bien la idea de ocupar E suena bien en principio, nos dimos cuenta que hacerlo generaría falsos positivos y negativos en nuestra elección de *outliers*. Es decir, hay registros con un bajo valor de E que visualmente se aprecia que tienen errores de otros patrones (falso positivo: lo retenemos cuando debería de ser un *outlier*), y también hay otros con un alto valor de E que se ve que ya no tienen componente de tendencia, pero por la elección del filtro con ventana 200 o por los mismos patrones del *ECG* se está disparando la métrica (falso negativo: lo consideramos *outlier* cuando deberíamos retenerlo).

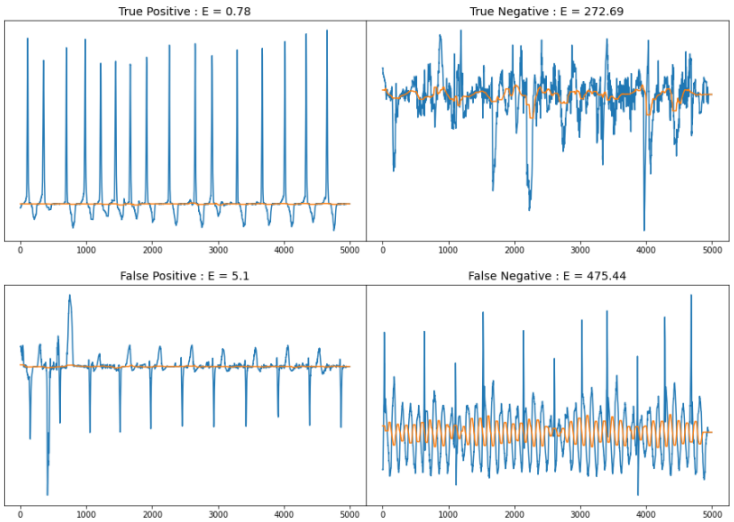


Fig. 9: Ejemplo de un verdadero positivo (*vp*), un verdadero negativo (*vn*), un falso positivo (*fp*) y un falso negativo (*fn*). En azul se muestra la señal tras un filtro de mediana, en naranja se muestra el filtro de mediana de la azul. El *fp* tiene una E baja pero se observa que tiene ruido de otros patrones. El *fn* tiene una E alta, pero no cuenta con tendencia.

Aún si la propuesta de ocupar E como métrica para filtrar *outliers* no fue correcta, nos permitió reafirmar 2 cosas:

- El ruido de otros patrones no es necesariamente eliminado al corregir el de *baseline drift*.
- Idealmente, el valor de la ventana no debería ser único para todas las señales sino específico para cada una.

El primero de estos puntos es lo que genera falsos positivos y el segundo falsos negativos. Si E no está funcionando adecuadamente para retener a los que en efecto se les retiró la tendencia, y al mismo tiempo clasificar como *outliers* a los que tengan ruido de otros patrones, entonces hay que introducir un nuevo criterio. Para el desarrollo de este criterio se llevaron a cabo varios experimentos, probando con distintas ideas y continuamente rebotándolas con los doctores para verificar que las técnicas empleadas tuvieran sentido en el contexto médico.

La idea clave que rodea este nuevo criterio es la siguiente: Establecer intervalos alrededor de las señales, para clasificar como *outlier* a aquellos que rebasen los intervalos. Estos intervalos deben de tomar en cuenta los cuantiles superiores e inferiores de la señal, y tener un margen hacia arriba y hacia abajo de lo que sería rebasar los intervalos significativamente [Fig. 10]. La sensibilidad para determinar los intervalos fue calibrada por medio de compartir ejemplos con los doctores. Nótese que esto sólo detectaría ruido de otros patrones que sean elevaciones o depresiones superiores a las usuales, más no otros ruidos como por ejemplo, que la señal anormalmente se quede en el *baseline* por mucho tiempo (lo cual pudo ser ocasionado por un desconecte breve del monitor de eventos cardíacos).

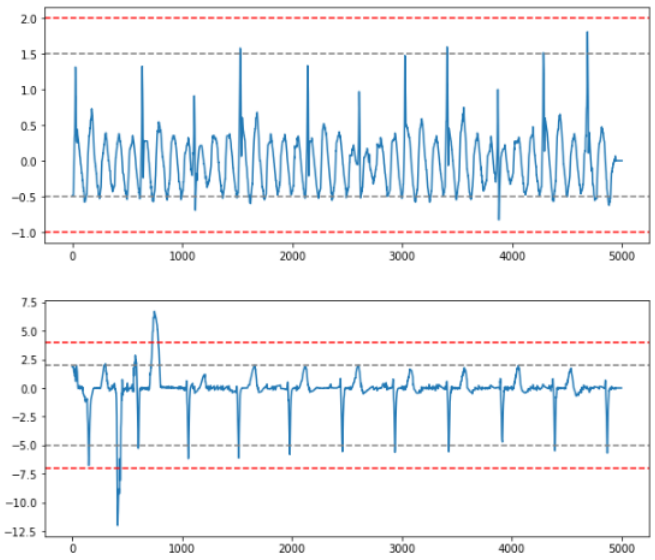


Fig. 10: Ejemplo de intervalos sensibles para filtrar *outliers*. Las lineas punteadas en gris son intervalos sensibles obtenidos a partir de los picos de los segmentos QRS (más adelante se explicará este concepto). Las lineas punteadas en rojo son los intervalos generados en función de los anteriores y de la amplitud de la señal; estos son los márgenes para determinar si la señal en algún momento se sale de lo usual. La primera señal es un ejemplo de un registro que no rebasa en ningún momento los márgenes. La segunda muestra una señal que sí rebasa los márgenes, tanto por arriba como por abajo, y en varios puntos.

Antes de explorar el método que se desarrolló para filtrar *outliers*, vale la pena hacer un *disclaimer*: Los modelos que se entrenaron tras retirar los *outliers* con este método **no** tuvieron un incremento en las métricas de evaluación (las métricas sí mejoraron con el filtro de mediana pero no con la eliminación de *outliers*). Esto puede tener tres explicaciones: el método en si no es adecuado, le falta calibración al método para ser útil, y/o las *CNN* resultan suficientemente poderosas para hacer caso omiso a los segmentos ruidosos.

A pesar de esto, el trabajo realizado para retirar *outliers* ocupó una parte considerable del tiempo empleado en esta investigación, y explicarlo sirve para que en futuros trabajos relacionados, se pueda tomar como punto de partida para mejorar el método, por lo que se incluirá su descripción en este reporte. También se explorarán brevemente algunas ideas previas que se desarrollaron pero que no se ocuparon al final. Se empezará con estas ideas pues sirven de puerta para entender lo que se buscaba realizar en un principio.

Una primera idea fue tratar de ocupar los cuantiles de la distribución de la señal. No obstante, se notó prontamente que no habría un par único de cuantiles que permitiera esto [Fig. 11], y el tratar de forzar a tener un único par implica incrementar ya sea falsos positivos o negativos en la elección de *outliers*.

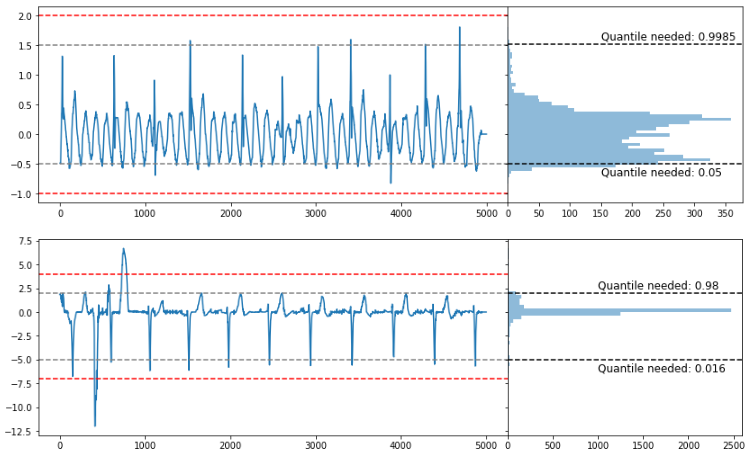


Fig. 11: Ejemplo de como no se podría elegir un único par de cuantiles para generar los intervalos. Las lineas negras punteadas junto con las leyendas indican los cuantiles en los que se podría igualar a las lineas grises punteadas, que son los intervalos sensibles explicados en Fig. 10. Lo importante a notar es como los cuantiles tanto superiores como inferiores (0.9985 *vs* 0.98 y 0.05 *vs* 0.016) no coinciden.

Posteriormente, se exploró si se podía generalizar más fácilmente si en vez de ver la distribución completa, se ocupaban las distribuciones cada cierta cantidad de puntos. Por ejemplo, si ocupamos una ventana de tamaño 100, entonces habrá una distribución para los puntos del 1 al 100, otra de 101 a 200, y asi hasta el 4,901 a 5,000. Esta idea mostró tener potencial para ser el método final, ya que los intervalos que se generan a partir de los cuantiles de las distribuciones se acercan bastante a los ideales que se calibraron con los doctores. No obstante, aún habían algunos casos donde, ya sea el límite positivo o el negativo, se difería bastante del ideal [Fig. 12].

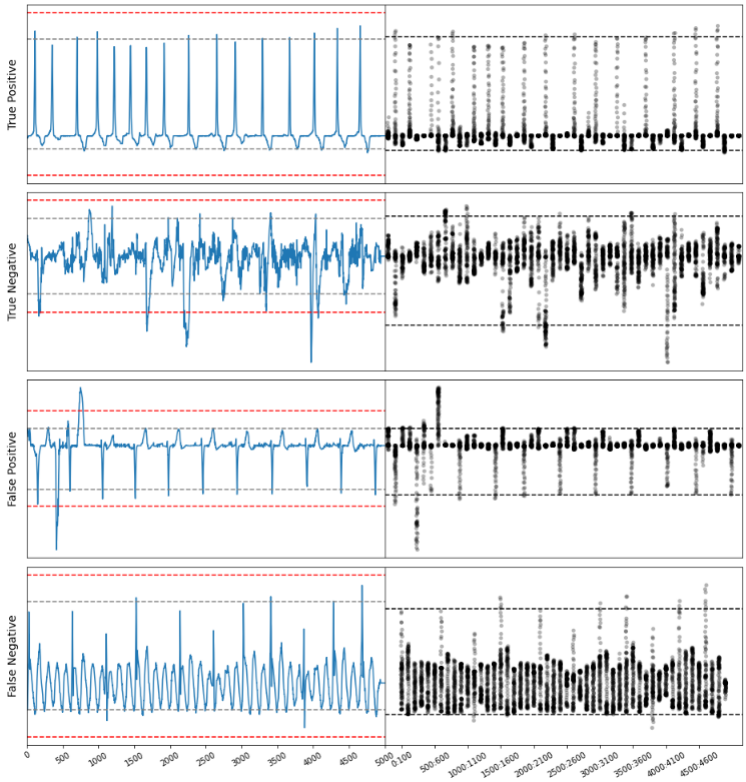
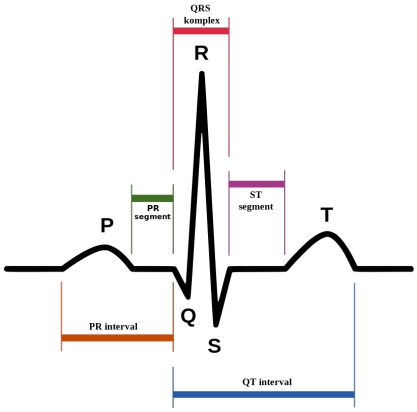
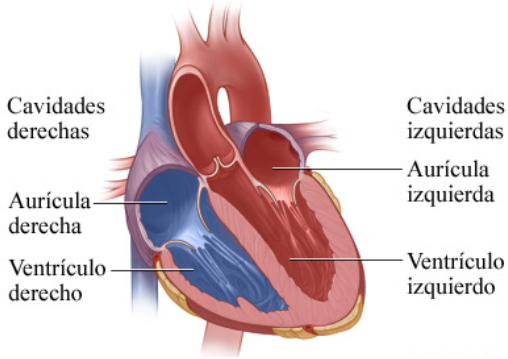


Fig. 12: Ejemplos de ajuste de intervalos por medio de los cuantiles de distribuciones cada 100 puntos. Las lineas punteadas en gris son los intervalos calibrados con doctores, las lineas punteadas en negro son los intervalos de los cuantiles de distribuciones cada 100 puntos. En casi todos los casos, estos intervalos son bastante cercanos. No obstante, en el verdadero negativo de este ejemplo hay una diferencia considerable en la parte negativa del intervalo.



(a) Segmento PQRST en un *ECG*



(b) Imagen de un corazón ilustrando la ubicación de las aurículas y los ventrículos.

Fig. 13: Imágenes explicativas del segmento PQRST y las partes del corazón [18][19]

Esto en parte se debe a que el valor de la ventana con la que se van seleccionando los puntos debería ser elegido de manera óptima para cada señal. El valor que mostró mejores resultados fue una ventana de 100. Si bien podríamos comprometernos de la misma forma que con el tamaño de la ventana en el filtro de mediana, se propuso ver si habría alguna otra manera de clasificar *outliers*.

En todo este tiempo, no se han aprovechado los segmentos que caracterizan a los *ECGs* para apalancarnos de esto en su limpieza. Como se mencionó previamente, un *ECG* registra la actividad eléctrica del corazón. En cierta forma, es ver explicitamente como el corazón se prepara para latir, late, y se relaja, repetitivamente. El estándar para nombrar estas etapas es ocupar letras de la P a la T. Si bien el conocimiento técnico de que significa cada uno de estos segmentos no es requerido para entender este reporte, se agrega una breve descripción pues el complejo QRS es de particular interés[17]:

- P (onda P) : En este momento ocurre una despolarización de las aurículas (las partes superiores del corazón) cuando se contraen para enviar sangre a los ventrículos (las partes inferiores). Es como la preparación del corazón para latir [Fig. 13].
- QRS (complejo QRS) : En este segmento ocurre la despolarización de los ventrículos, que son las partes principales del corazón que bombean sangre al cuerpo. Es el latido en si, y en esta se aprecia la fuerza del bombeo.
- T (onda T) : Finalmente, los ventrículos se repolarizan, se relajan y se preparan para el siguiente latido. Es el final de la actividad eléctrica en un latido, tras el cual el corazón entra en reposo hasta la siguiente onda P.

Dentro de todos los ejemplos que se han incluido, se aprecia que en el complejo QRS es donde la señal tiene sus máximos y mínimos (por lo general). En los métodos que se han desarrollado hasta el momento, en cierta forma se ha buscado aproximarnos a tales valores por medio de cuantiles. De saber donde ocurren los complejos QRS, se podrían recuperar los valores máximos y mínimos en cada uno, con lo que los intervalos podrían ser calculados de mejor manera.

Para recuperar los complejos QRS, se ocupó una función implementada en la paquetería *wfdb*: *xqrs_detect*. Esta es la paquetería con la que también se están extrayendo las señales a partir de los archivos con terminación *.hea* y *.mat*. Fue creada para la lectura y procesamiento de señales del tipo *waveform-database* (*WFDB*), junto con sus anotaciones[20][21]. Para una breve descripción de lo que realiza la función *xqrs_detect* se puede consultar la documentación; para obtener mayor detalle se puede consultar el artículo de *M. A. Z. Fariha et* al titulado *Analysis of Pan-Tompkins Algorithm Performance with Noisy ECG Signals*[22].

Esta función es bastante robusta, ya que puede detectar los segmentos QRS incluso en presencia de ruido [Fig. 14]. Incluso se detectaron casos en los que no se obtenía algún segmento QRS en las señales tras haber pasado por un filtro de mediana, por lo que se optó por ocupar las señales originales para obtener la localización de estos.

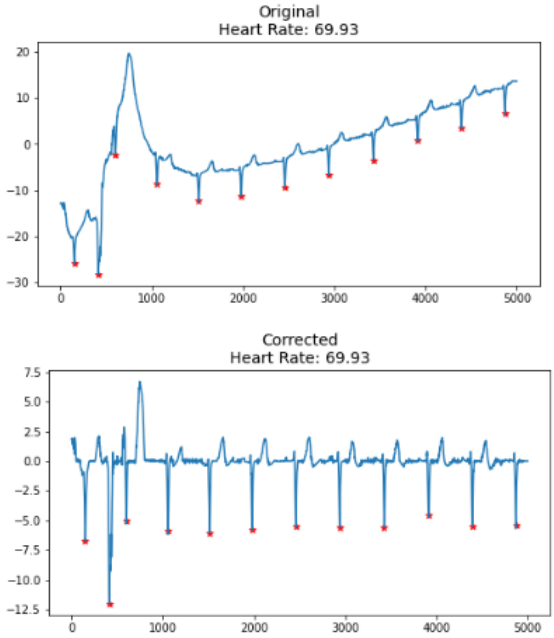


Fig. 14: Ejemplo de la obtención de segmentos QRS. La primera es la señal original y la segunda es tras haber pasado por un filtro de mediana. Se puede observar que a pesar de contar con alto ruido, se están detectando correctamente los segmentos QRS en ambas (estrellas rojas).

Los segmentos QRS usualmente se aprecian como picos "hacia arriba" en la señal. No obstante, algunas de las señales también pueden tener picos "hacia abajo". La función *xqrs_detect* detecta adecuadamente los segmentos sea cual sea el caso. Esto resuelve la mitad del problema, pues ya tendríamos los puntos para generar el intervalo para uno de los lados de la señal. Ahora el asunto es obtener el otro intervalo.

Como se aprecia en el diagrama en Fig. 13 del segmentro PQRST y en los *ECG* vistos hasta el momento, hay otras elevaciones y depresiones que ocurren en la señal, que ocurren en la onda P, en los puntos Q y S, y en la onda T. Al ya saber la ubicación de los segmentos QRS, la idea es obtener vecindades no superpuestas alrededor de estos, y ubicar el punto mínimo (si el segmento QRS es positivo) o el máximo (si es negativo). De esta forma, se pueden obtener los puntos extremos del otro lado en que se detectó el segmento QRS. Con ambos conjuntos de puntos extremos, se calculan las medianas para generar los intervalos [Fig. 15]. Calcular los intervalos de esta forma mostró la mayor cercanía a los intervalos ideales calibrados con los doctores [Fig. 16].

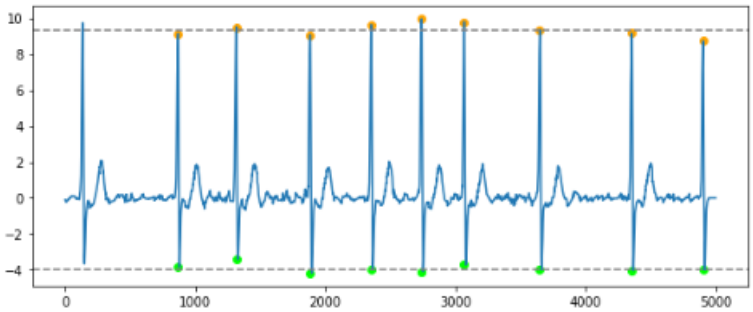


Fig. 15: Ejemplo de la obtención de los puntos extremos en una señal, y los intervalos generados por las medianas de cada conjunto. Cabe notar que en este ejemplo, el primer segmento QRS no fue detectado. Esto fue recurrente en varios casos. Sin embargo, no poder recuperar el primero no representa un problema, pues se pueden calcular los intervalos con los demás.

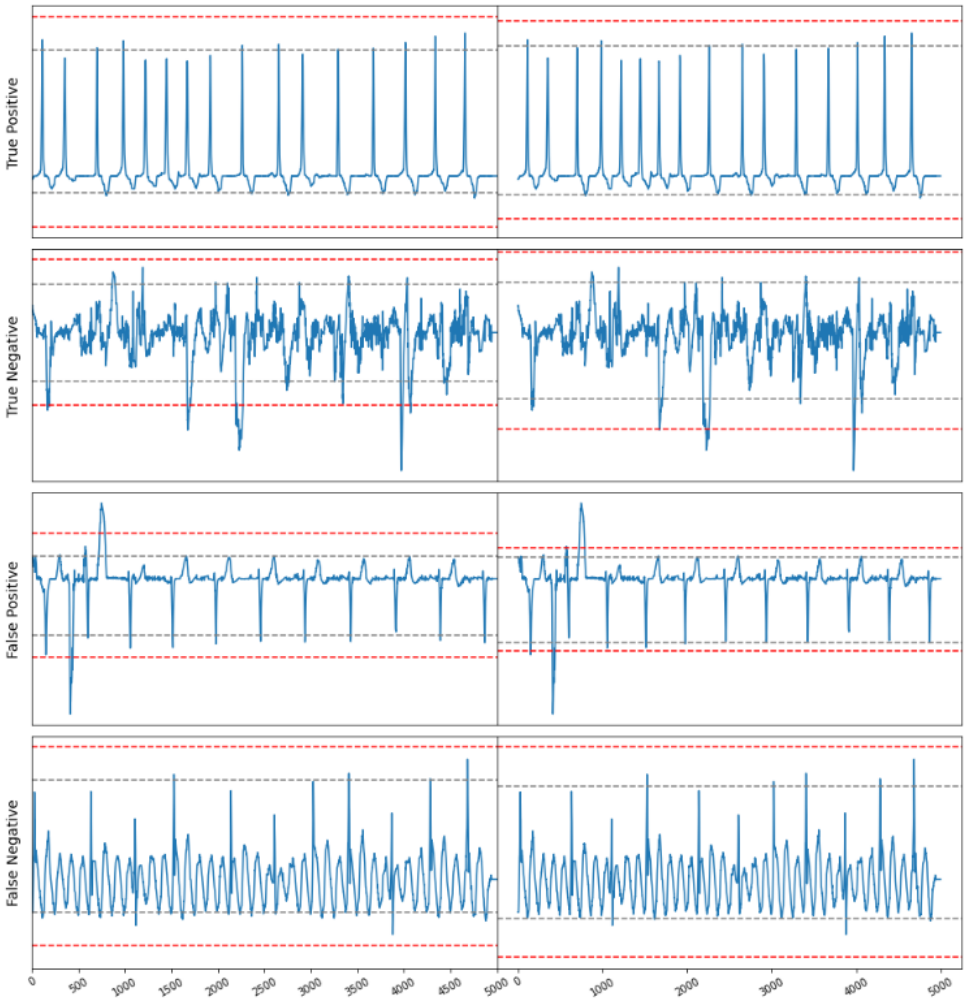


Fig. 16: Ejemplos de ajuste de intervalos por medio de la identificación de segmentos QRS. En comparación con los intervalos generados cada 100 puntos, vemos que en este caso ya también el verdadero negativo está siendo bien ajustado.

El único problema es que el cálculo de los márgenes ha resultado difícil de ajustar a los calibrados con los doctores. Se calculan en función de la amplitud de la señal, donde la amplitud la aproximamos de la siguiente forma:

$$amplitud = intervalo\ superior - intervalo\ inferior$$

Se han probado diversas funciones para hacer que el margen se parezca lo más posible al de los doctores. La idea actual es sumar y restar a cada intervalo una cantidad determinada por la siguiente función:

$$margen = \frac{\sqrt[4]{amplitud}}{2}$$

Esta surgió a partir de ir experimentando y ajustando hacia los márgenes ideales. Finalmente, los márgenes fueron definidos como:

$$margen\ superior = intervalo\ superior + margen$$
$$margen\ inferior = intervalo\ inferior - margen$$

Si bien se puede apreciar que no generó los márgenes más adecuados, por el momento es lo que se ocupó [Fig. 16]. Más ideas para generar intervalos adecuados a futuro incluyen seguir experimentando con diferentes funciones, así como tomar en cuenta no sólo la mediana de los puntos extremos sino algún cuantil para incluirlo a la función.

Ya que contamos con los intervalos y los márgenes, falta determinar con que criterio se estará determinando si algo es un *outlier* o no. Antes de explicar el criterio, cabe hablar sobre un diagnóstico en particular: la ectopía ventricular. Si bien no hemos hablado de otros diagnósticos más que los infartos de miocardio, este diagnóstico es importante por cómo se ve en los *ECG* más allá del detalle médico que conlleva[23].

Cuando un paciente cuenta con ectopía ventricular, el *ECG* presenta elevaciones y depresiones más amplias y/o cortas, y a destiempo, que en los otros segmentos PQRST. Pueden ocurrir tanto una como varias veces en la señal, y además múltiples señales exhiben este comportamiento al mismo tiempo [Fig. 17]. Esto puede parecer ruido de otros patrones, pero la diferencia es justamente el hecho de que ocurre simultaneamente en varias señales. De no tener esto en cuenta, falsamente se clasificaría a prácticamente todos los casos de ectopía ventricular como *outliers* si tomaramos un criterio simple como "*un outlier es todo registro que rebase n veces los márgenes*".

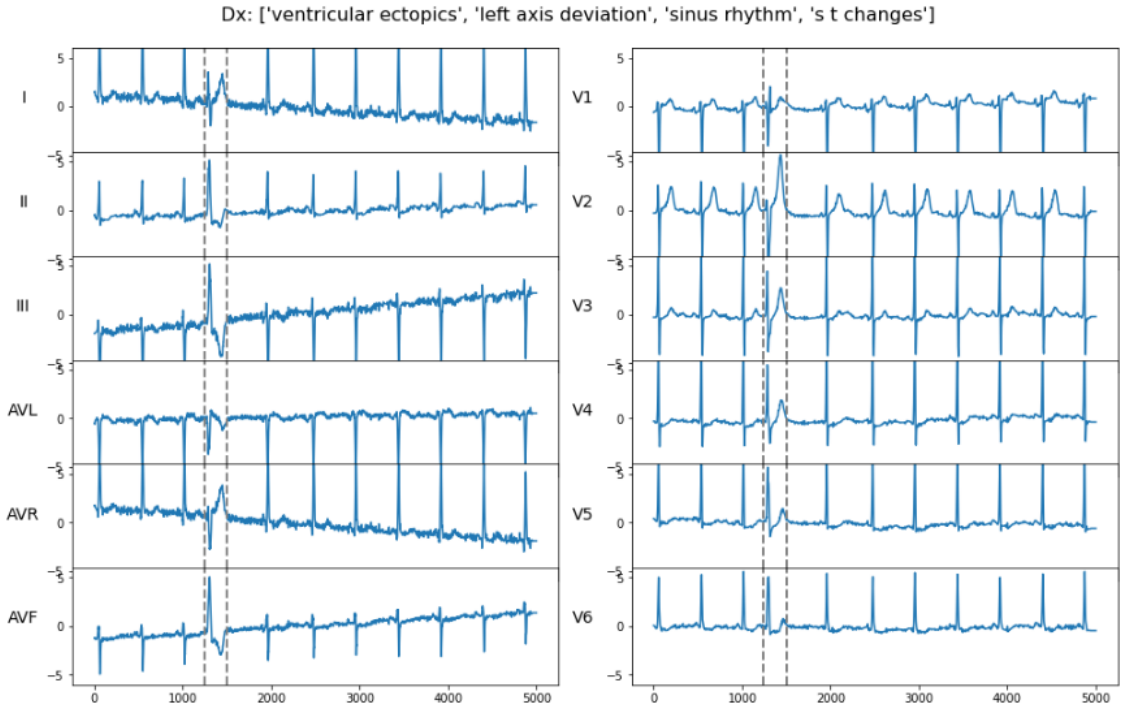


Fig. 17: Ejemplo de las 12 señales de un *ECG* con ectopía ventricular. Se aprecia que el paciente cuenta con este diagnóstico pues entre las lineas punteadas en gris en todas las señales hubo un patrón distinto al que ocurre en el resto de los segmentos PQRST.

Para evitar perder los registros de ectopía ventricular, a la idea de generar un criterio que clasifique *outliers* por el número de veces que se rebasen los márgenes se le debe de considerar el que, si se exceden simultaneamente en varias señales, entonces estos no serían *outliers*. Tomando en cuenta esto, se estuvieron desarrollando ideas con los doctores de cómo calibrar el criterio para minimizar los falsos positivos y negativos. Se llegó a la siguiente lista de condiciones sucesivas que componen el criterio:

1. Si en 5 puntos o más hay 5 o más señales que rebasaron el margen simultaneamente, el *ECG* **no** es *outlier* (pues probablemente es ectopía ventricular).
2. Si no se cumple lo anterior y en 25 puntos o más hay 1 o 2 señales que rebasan el margen, el *ECG* **sí** es *outlier* (pues estos eventos probablemente son de ruido de otros patrones).
3. Si no se cumple ninguno de los anteriores, el *ECG* **no** es *outlier*.

Con el criterio, finalmente podemos clasificar a los registros para detectar los *outliers* [Fig. 18]. Un mínimo deseable es que este criterio pueda captar a algunos de los falsos positivos y negativos que se estudiaron, lo cual sí está cumpliendo [Tab. 1]. Con los registros detectados como *outliers* o no, podemos decidir incluirlos o no dentro de los conjuntos de entrenamiento, validación y prueba. El total de registros clasificados como *outliers* fue de 827 *ECG*, lo que representa un 3.79% de los datos.

	E	E Criteria	was_1_or_2	was_over_4	Median Peak Criteria
true positive	0.779641	regular	10	0	regular
true negative	272.69131	outlier	93	0	outlier
false negative	475.444592	outlier	6	0	regular
false positive	5.095161	regular	140	0	outlier

Tab. 1: Tabla comparativa de detección de *outliers*. Los registros son los mencionados en [Fig. 9]. La columna *E Criteria* muestra la elección que se habría tomado si sólo se ocupara un valor alto de *E* para clasificar *outliers*. La columna *was_1_or_2* indica la cantidad de veces que en 1 o 2 señales se rebasaron los márgenes simultaneamente. La columna *was_over_4* indica la cantidad de veces que en 5 o más señales se rebasaron los márgenes simultaneamente. La columna *Median Peak Criteria* muestra la elección con el criterio de los intervalos de confianza generados a partir de las medianas de los segmentos QRS. Se aprecia como el *E Criteria* hubiera generado un falso negativo y un falso positivo, mientras que el *Median Peak Criteria* clasifica a todos adecuadamente.

El propósito de la clasificación y filtrado de *outliers* fue ayudar a que el modelo no se sesgara en aprender que el ruido estuviese asociado con registros *MI* (o no *MI*). No obstante, como ya se comentaba a manera de *spoiler* de la siguiente sección, los modelos entrenados sin *vs* con los *outliers* no mostraron practicamente mejora alguna, incluso parece afectar el desempeño. Lo valioso de haber realizado esta exploración es que ya se tienen las bases para que, en caso de seguir queriendo encontrar registros con ruido de otros patrones, ya sea para ayudar al modelo o por cualquier otro motivo, no se tenga que empezar desde cero.

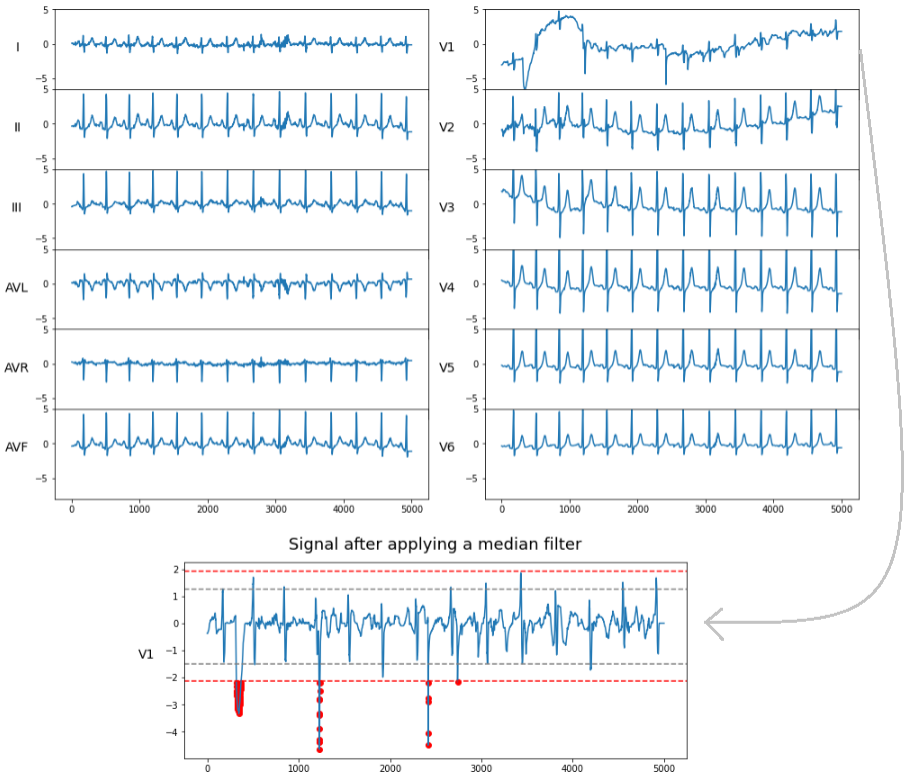


Fig. 18: Ejemplo de un *outlier*. Se muestran las 12 señales originales del *ECG*, dentro de las cuales se aprecia que la señal V1 tienen un ruido de *baseline drift* y posiblemente también de otros patrones considerablemente elevado. Tras realizar una pasada del filtro de mediana y obtener los intervalos y márgenes, se aprecia que esa señal muestra bastantes puntos fuera de los márgenes, 67 en concreto. En ninguna otra señal hubieron puntos que se salieran, por lo que este caso cae en el punto 2 del criterio, siendo así clasificado como *outlier*.

7.2 Comparativa y mejoras de las *CNN*

Las *CNN* son particularmente poderosas para tareas de visión por computadora[24]. Muchos de los problemas que antes se resolvían con otros métodos han sido superados en desempeño por las *CNN*. El más común de estos problemas es la clasificación de imágenes. Los filtros de las redes pueden aprender el "como se ven" los patrones que se asocian a la determinada etiqueta sin importar la posición en los que se encuentren. Por ejemplo, si se quieren clasificar imágenes para determinar si estas contienen un perro o un gato, los filtros aprenden a determinar los sutiles patrones que caracterizan a cada animal: La forma de las orejas, ojos, nariz, etc. También se pueden apoyar en las diferencias de colores, lo cual se logra si se pasan registros que incluyan el canal del color. Es decir, si las imágenes son de 100x100 pixeles y hay 3 canales de color (*RGB* usualmente), entonces hay tres matrices de 100x100, una por cada canal, teniendo así tensores de dimensión (100, 100, 3) por cada imagen.

Si a algún científico de datos se le comenta que se están ocupando *CNN* para la clasificación de *ECG*, probablemente asociaría el uso común de las *CNN* con problemas de imágenes y pensaría que los datos son fotografías de los *ECG*, con una dimensión de 3, quizás de (1080, 1080, 3). No obstante, los datos como tal son los milivoltios (*mV*) de cada una de las 12 señales a través de 10 segundos. La frecuencia de muestreo es de 500 por segundo, por lo que hay 5,000 puntos en cada señal. Es decir, la dimensión de cada *ECG* es de (5,000, 12). Las capas convolucionales que se están ocupando son de 1 dimensión con 12 canales (como si cada señal fuera un color en el contexto de las imágenes). De esta forma, los filtros pueden aprender conjuntamente los patrones de las 12 señales. Para determinar diagnósticos, los doctores buscan patrones que ocurran en varias señales al mismo tiempo, y eso lo pueden captar los filtros pues tienen una visión y aprendizaje transversal entre las 12 señales.

Al ser series de tiempo, cabe la duda de por qué no se ocuparon redes recurrentes, como una *LSTM*, para clasificar los *ECG*. Sin duda están en el radar de esta investigación y en futuros avances se contempla hacer modelos que mezclen las arquitecturas, probablemente uniendo las dos al final en capas densas. No obstante, se optó por las *CNN* en primera instancia ya que, a pesar de ser series de tiempo, la naturaleza repetitiva de los patrones se adecua mucho más a poder ser aprovechada por los filtros. Además, dentro de la investigación del estado del arte que se realizó, los modelos que mejor desempeño muestran tienen una arquitectura principalmente de redes *CNN*[25][26][27][28].

Para entrenar una *CNN*, lo primero que se debe considerar es el tamaño del filtro y de los *strides* (desfase del filtro en cada iteración, usualmente es 1). Un tamaño de filtro muy pequeño aprende únicamente patrones de grano fino, mientras que uno muy grande pierde la sensibilidad de patrones locales. También entra a consideración el ocupar *Max Pooling* para evitar tanto la complejidad computacional como en parte el sobreajuste. Estas dos ideas fueron consideradas en todos los modelos que se iban entrenando. Además, sucesivamente se fueron añadiendo distintas técnicas y modificaciones en busqueda de mejorar el desempeño del modelo, como ocupar *Batch Normalization*, *Dropout*, regularización *L1*, reducción de la tasa de aprendizaje cuando la pérdida deja de disminuir, entre otros.

Como se mencionó, este proyecto en su totalidad no sólo busca predecir si un paciente tiene un infarto de miocardio (*MI*) o no, si no también el hacer diagnósticos generales de los padecimientos que pueda tener un paciente. Durante los primeros meses de desarrollo, los modelos que se entrenaron consideraban alrededor de 30 diagnósticos de interés para los doctores. La salida del modelo tenía una función de activación sigmoide (binaria) para cada uno de los diagnósticos, en conjunto con una pérdida *Binary Cross Entropy*, pues un paciente puede tener múltiples diagnósticos. La forma en la que se entrenaron los modelos fue sucesivamente ir agregando las técnicas e ideas (*Batch Normalization*, *Dropout*, etc.) y modificarlas tal que la pérdida en el conjunto de validación disminuyera.

Las métricas de evaluación de esos modelos son relativamente bajas, hay muchos falsos positivos y negativos dentro de los diagnósticos. Esto es entendible ya que el modelo trata de aprender demasiados patrones y se sesga hacia las clases con mayor cantidad de registros, incluso en precencia de modificar los pesos de las clases (actualizar más bruscamente los pesos de las redes cuando el registro pertenece a una clase con pocos registros). Dado esto, por el momento se decidió acotar el problema a predecir únicamente sí el paciente tiene *MI* o no. El código se pudo reutilizar sin mayor inconveniente pues sólo había que cambiar la salida. Durante el entrenamiento de los nuevos modelos, se hicieron modificaciones pequeñas, como cambios en la tasa de *Dropout*, alteración de los pesos de las clases, entre otros, pues las arquitecturas mostraban en general buenos resultados para el nuevo enfoque.

En un inicio, los modelos se corrieron con los datos en crudo. Es decir, tal cual las señales como vienen desde sus archivos. Más adelante fue cuando, en busqueda de mejorar el desempeño, se consideró realizar el tratamiento de los datos, detallado en la sección anterior. Los 4 conjuntos de datos son: crudos con y sin *outliers*, y tras el filtro de mediana con y sin *outliers*. Todas las arquitecturas fueron calibradas con el conjunto crudo con *outliers*, y posteriormente fueron corridas con los otros 3 conjuntos. Esto se realizó con el motivo de determinar si hay un efecto positivo en el desempeño únicamente asociado al tratamiento de los datos (dejando lo demás fijo).

Ocupar los 10 segundos de la señal fue inviable con el equipo de cómputo ocupado, por lo que se cortaron para tener únicamente los primeros 3 segundos de la señal. Lo bueno de los *ECG* es que los patrones se repiten, por lo que en

general es posible para los médicos (y para el modelo) identificar la mayoría de los diagnósticos con pocos segundos de información. No obstante, hay casos, como la ectopía ventricular mencionada anteriormente, en los que el patrón es más bien esporádico, y podría estar ocurriendo después de los primeros 3 segundos. Sin embargo, de acuerdo a los doctores del *INC*, los infartos de miocardio son identificables con sólo 3 segundos. De cualquier manera, así como se realizó el análisis del efecto del tratamiento de datos, también se corrieron algunos modelos con 5 segundos para ver si había un aumento en las métricas.

Antes de presentar las comparativas de los modelos, es importante explicar 2 cosas:

1. Como se fueron añadiendo las técnicas sucesivamente
2. Cuales son las métricas ocupadas para la comparativa

El primer modelo de todos fue una *CNN* con tamaño de filtro igual a 3, de 3 capas, y con *Max Pooling* después de cada capa. Este modelo tuvo el peor desempeño de todos, como era de esperarse. Posteriormente, nos dedicamos a hallar un tamaño del filtro adecuado, a determinar la cantidad de filtros en cada capa, y a aumentar la cantidad de capas, lo cual ayudó bastante al desempeño. Después experimentamos introduciendo *Batch Normalization*, luego agregando la penalización *L1* para evitar el sobreajuste, y así sucesivamente. Esto es importante porque al final se hizo una especie de análisis de atribución para determinar cuales de estas ideas y técnicas aumentaron con mayor significancia el valor de las métricas: Se tomaron las métricas de los modelos con la nueva idea y las de los modelos con la idea anterior, se restaron, y se obtuvieron promedios para cada una.

Este análisis no es robusto ni indica tal cual la "causalidad" de haber ocupado una técnica en el desempeño del modelo. En todo caso, para obtener un resultado más cercano a lo que sería la verdadera atribución, habría que correr modelos con todas las combinaciones de técnicas y posteriormente obtener la diferencia por cada una de las métricas con vs sin el uso de dicha técnica. Esto es inviable y tampoco hay mucho valor en realizarlo. El motivo por el que se hizo este análisis fue para que, probadas todas las ideas, viéramos cuales parecían ayudar más al desempeño, y que al final optimizáramos un modelo tomando particular atención en estas técnicas que atribuyeron positivamente. El mejor modelo, habiendo jugado con las ideas de mayor atribución, tuvo un desempeño superior que al de todos los anteriores.

El orden de las técnicas e ideas fue el siguiente. También se agrega el diminutivo con el cual se guardaron los archivos, pues posteriormente se mostrará la comparativa de los modelos y con tal diminutivo se puede saber qué técnicas se consideraron:

1. Modificación del tamaño del filtro, del *stride* y número de capas (*ks*)
2. *Batch Normalization* (*bn*)
3. Regularización *L1* (*l1*)
4. *Dropout* (*dr*)
5. Inclusión de las variables de edad y sexo (*as*)
6. Reducción de la tasa de aprendizaje cuando la pérdida deja de disminuir (*lr*)
7. Modificación de los pesos de las clases (*cw*)

Aprovechando el listado de técnicas, se incluyen los diminutivos para identificar otras cosas dentro del nombre de los modelos, como si se ocuparon los datos crudos o modificados, con *outliers* o no, etc.:

1. Datos crudos / originales (*og*)
2. Datos tras filtro de mediana (*mf_200*)
3. Datos sin *outliers* (*ol_5_25*) (el 5 y 25 hacen referencia al criterio de la sección anterior, puede que en algún momento se ocupe otro criterio)
4. Datos con sólo los primeros 3 segundos (*ct_0.3*)
5. Datos con sólo los primeros 5 segundos (*ct_0.5*)
6. Datos con sólo los primeros 8 segundos (*ct_0.8*)
7. Modelos optimizados tras el análisis de atribuciones (*optim*)

Dentro de la investigación del estado del arte, se encontró un *script* en *Github* que contiene la arquitectura de una *CNN* para la detección de diagnósticos en *ECG*[29]. Ocupa varias de las ideas y técnicas que los modelos generados en este proyecto, con la inclusión de una nueva: capas residuales. Se decidió verificar el desempeño de esa aqruitectura y

compararla con los modelos. Los registros que contienen el texto *antonior92* (el nombre del usuario en *Github*) son los generados por dicha arquitectura.

En la sección de las métricas de evaluación, se mencionó que la sensibilidad, especificidad y precisión son en las que nos fijamos. Si bien es cierto esto, hay un detalle importante en cómo se calcularon. En el contexto médico, y más en particular en la detección de si una persona tiene *MI* o no, es de mayor peso el poder identificar correctamente a los que sí padecen el infarto, que falsamente clasificar de positivo a alguien que no lo tiene. Por ello, ocupar un punto de corte en el 50% al momento de clasificar (menor a 50% es predecir que no tiene infarto, mayor a 50% es que sí) puede hacernos perder puntos porcentuales importantes en la sensibilidad del modelo. Para esto decidimos tomar 2 enfoques: 1. Elegir el punto de corte tal que al menos se tuviera una sensibilidad del 90%, y 2. Elegir el punto de corte tal que la sensibilidad fuera igual a la especificidad.

El primero busca optimizar la sensibilidad para poder identificar a un gran porcentaje de pacientes con el diagnóstico positivo. No obstante, esto es a costa de perder precisión en el modelo, es decir, forzar a recuperar pacientes con infarto va a generar que hayan más personas sin infarto a los que se les diagnostique como positivos. El segundo busca un equilibrio entre sensibilidad y especificidad, tal que, en porcentaje, se recuperen al mismo número de pacientes tanto con como sin infarto. Velar por la especificidad implicitamente ayuda también a la precisión. Esto sería a costa de perder algunos puntos porcentuales en la sensibilidad. Dentro de las gráficas que se muestren, el sufijo *target* indica que se eligió el punto de corte para alcanzar la sensibilidad objetivo del 90%, mientras que *equal* indica que el punto de corte fue elegido para igualar la sensibilidad a la especificidad.

El análisis de atribuciones se hizo con respecto a estas métricas. Se tomaron las de los modelos con la nueva técnica y se les restaban las de los que no contaban con dicha técnica, todo lo demás fijo. Así se obtuvieron los promedios de cuántos puntos porcentuales aumentaba o disminuiba el incluir la nueva idea al modelo [Fig. 19]. Algunas ideas aumentaron considerablemente el valor de las métricas, otros resultaron ser perjudiciales, e interesantemente, algunos mostraron ayudar a las métricas del enfoque *target* pero empeorar a las del enfoque *equal*, o viceversa. Es decir, si nos interesara mejorar algún enfoque en particular, habría que poner enfasis en las ideas que incrementaron los puntos porcentuales y considerar retirar las que perjudican al modelo.

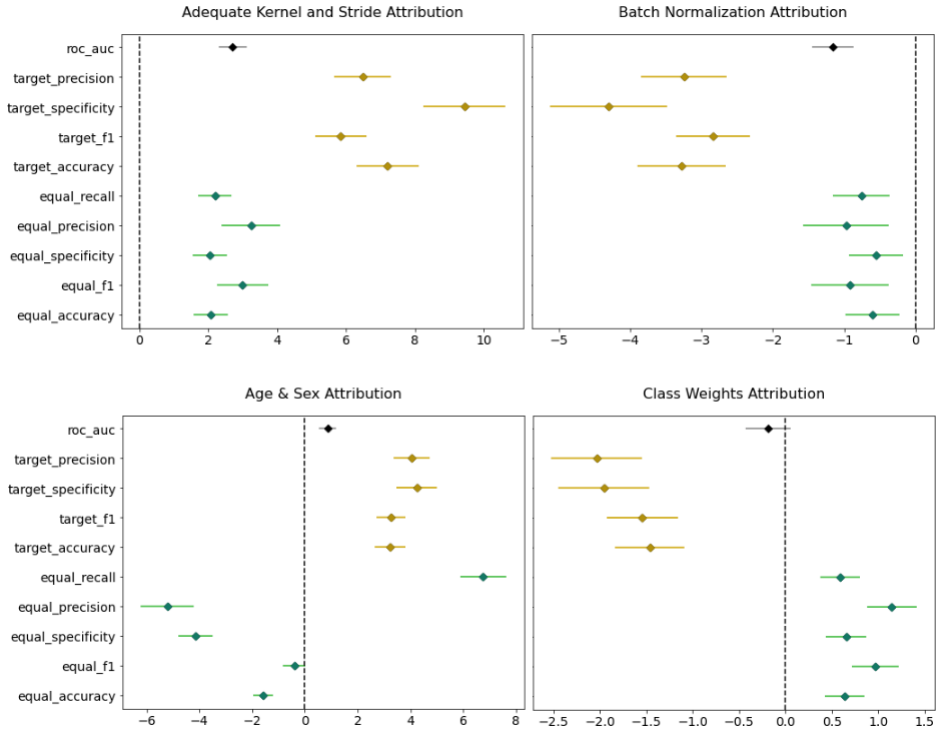


Fig. 19: Ejemplos del análisis de atribuciones. Los puntos son los valores promedio de las diferencias en cada métrica. Los intervalos son de 1 error estándar de las diferencias (desviación sobre raiz de la cantdad de ejemplos). Se puede apreciar que modificar el tamaño del kernel aumentó considerablemente los puntos porcentuales de las métricas. *Batch Normalization* por otro lado empeoró todas. Agregar la edad y el sexo ayuda al enfoque *target* pero perjudica al *equal*, y modificar los pesos de las clases hace lo contraio.

Fue en este análisis que se pudo determinar si las técnicas del tratamiento de datos habían contribuido positivamente o no a las métricas. Resultó ser que ocupar las señales filtradas tuvo un efecto postivo, con un promedio de 1.6 puntos porcentuales de incremento, mientras que quitar los *outliers*, al contrario, tuvo un efecto negativo, en alrededor de -0.5 puntos porcentuales [Fig. 20]. Como se comentaba en la sección del tratamiento, aún hay campo para mejorar la clasificación de *outliers*, y esto se reflejó en parte en este análisis. No obstante, eso también son noticias positivas con respecto al modelo: incluso en presencia de registros que tienen ruido, se está logrando aprender a no ponerle atención a ello sino a los patrones en los segmentos que no tienen ruido, generando así predicciones adecuadas.

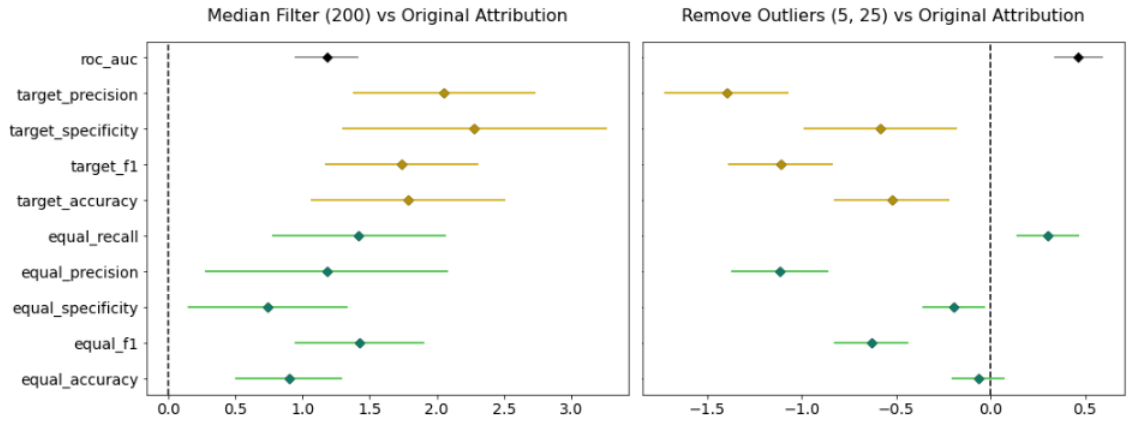


Fig. 20: Análisis de atribuciones de ocupar los datos tras un filtro de mediana *vs* los datos crudos, y de quitar *outliers vs* dejarlos.

Aún no se ha definido cual de los dos enfoques (*target vs equal*) es al que se le va a tomar mayor importancia. Por lo tanto, desarrollamos una métrica que ponderara de igual forma a todas las demás: Sensibilidad, especificidad y precisión, así como otras que se incorporaron para el cálculo de esta métrica general (*F1 score*, exactitud / *accuracy* y área bajo la curva *ROC*). Todas están entre 0 y 1, y entre más cercano a 1, mejor. Tomando esto en cuenta, la métrica general para comparar modelos se definió de la siguiente forma:

$$\pi(M) = \left(\Pi_{k=1}^{n} metrica_k\right)^{\left(\frac{1}{n}\right)}$$

Donde:

- **M** : Modelo
- **métrica**$_k$: k-ésima métrica de evaluación
- **n** : Cantidad de métricas

Es decir, es la media geométrica de las métricas. Para su cálculo se ocupó el conjunto de prueba, que son datos que el modelo nunca vió durante su entrenamiento. Se calculó para todos los modelos y se ordenaron de mayor a menor [Fig. 21]. Los mejores modelos resultaron ser algunos de los que se optimizaron tras el análisis de atribuciones. No obstante, también los peores fueron algunos de los que se generaron en ese proceso de optimización. Esto ocurrió ya que, en el primer modelo a optimizar se optó por pasar de un tamaño de filtro de 14 a 20. Esto mejoró el desempeño, por lo que se probó posteriormente con tamaños aún más elevados. No obstante, filtros tan grandes dejan de tener sensibilidad de los patrones, lo que afecta gravemente a la capacidad de aprendizaje y en consecuencia a las métricas. También se probó elegir un valor más restrictivo de *alpha* en la regularización *L1* y modificar los pesos de clase para que fueran más bruscos. De la misma forma que alterar el tamaño del filtro, ambas ideas mejoraron el desempeño con valores adecuados, pero lo empeoran significativamente si no se calibran tales valores.

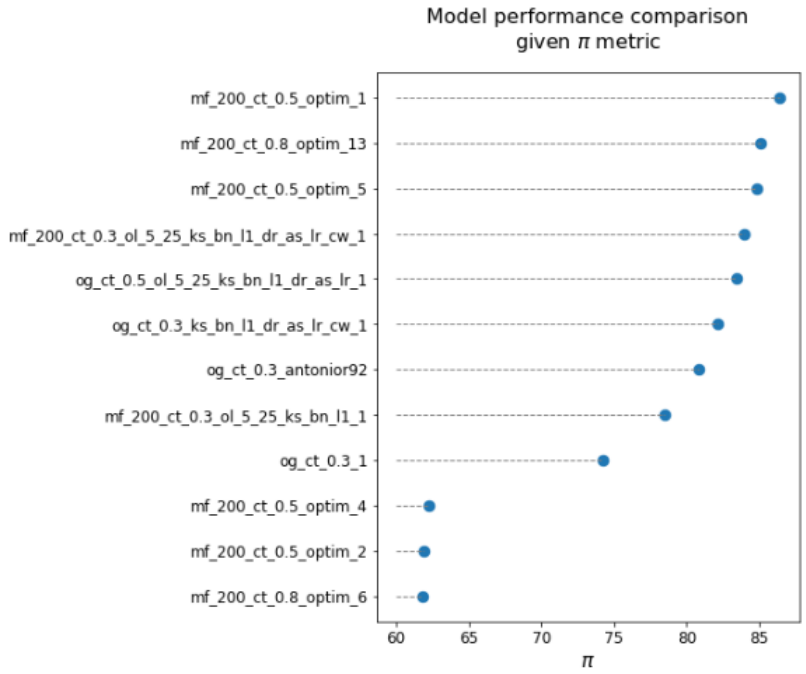


Fig. 21: Comparación de modelos dada la métrica π. En la visualización se conservaron los mejores 3, los peores 3, y una muestra aleatoria de otros en medio, ya que el total de modelos sería demasiado para la visualización.

Si bien el *script* de *antonior92* mostró buenos resultados, se logró superar su arquitectura. Además, los modelos desarrollados internamente tardan en entrenarse entre 10 a 15 minutos y tienen alrededor de 2 a 3 millones de parámetros, mientras que su arquitectura toma más de 50 minutos en el entrenamiento y cuenta con 6.4 millones de parámetros. Es decir, se logró un mayor desempeño de la mano con un menor costo computacional, tanto en tiempo como en memoria. No obstante, la idea de ocupar capas residuales es interesante, y gracias al trabajo en ese *script* será más fácil implementarlo eventualmente en nuestros modelos.

8 Resultados

La arquitectura del mejor modelo con respecto a la métrica π se aprecia en Fig. 22 y Fig. 23:

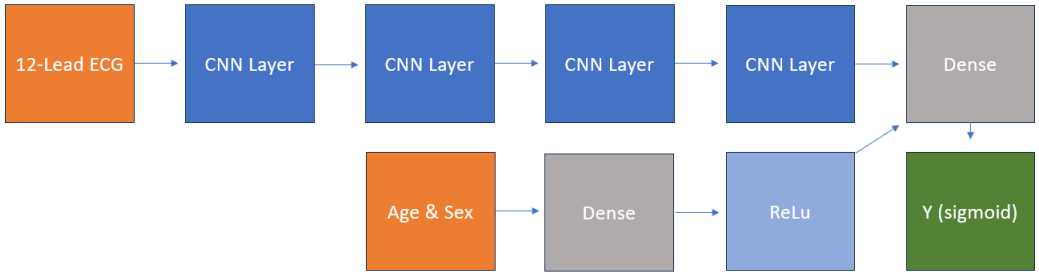


Fig. 22: Arquitectura del mejor modelo. Las 12 señales del ECG son procesadas en cuatro bloques *CNN*, definidos en Fig. 23, cada uno incrementando el número de filtros. En el último, también hay un regularizador *L1* después de la activación *ReLU*. La edad y sexo pasan por una capa densa pequeña, y todo se combina en una capa densa al final, de la cual salen las probabilidades de pertenecer a la clase.

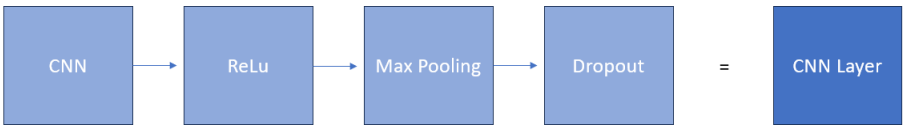


Fig. 23: Los bloques *CNN* del mejor modelo. Consisten en una capa convolucional con el tamaño del filtro igual a 20 y los *strides* con valor de 5, tras lo cual hay una activación *ReLU*, seguido de un *Max Pooling* y finalmente una capa *Dropout*.

Los pesos de las clases se modificaron tal que el aprender de un registro que fuera *MI* actualizara 10 veces más a los pesos de la red. Se ocupó un optimizador *Adam* con tasa de aprendizaje 0.00006. La reducción de la tasa de aprendizaje cuando la pérdida deja de disminuir tiene paciencia de 12 épocas, y la reduce en un factor de 0.2. Se entrena durante 200 épocas, con *early stopping* implementado si la pérdida no ha bajado en 30 iteraciones. La función de pérdida ocupada es *Binary Cross Entropy* pues la salida es binaria. La arquitectura es relativamente pequeña, cuenta con un total de 2,918,303 parámetros.

Durante el entrenamiento de los modelos, en particular los que tenían los pesos de clase modificados, el historial de cambio de la función de pérdida era bastante nervioso. Este fue el caso también para el mejor modelo [Fig. 24], pues se ocupó una relación 10 a 1 para los registros que eran *MI*, lo cual fue un tanto más alto que en promedio. Esto puede generar cierta desconfianza, pues se espera en general que el historial sea más estable. No obstante, todas las métricas de evaluación (y por tanto π) fueron calculadas con el conjunto de prueba, por lo que incluso en presencia de un historial tan nervioso tenemos la certeza de que este modelo generaliza adecuadamente.

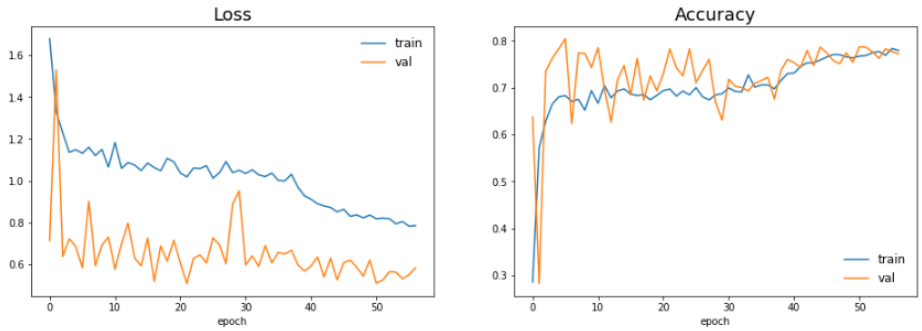


Fig. 24: Cambios en la función de pérdida y la exactitud a través de las épocas de entrenamiento, para los conjuntos de entrenamiento y de validación.

Antes de visualizar los resultados del mejor modelo, vale la pena mencionar que la distribución de registros que son *MI* *vs* los que no, no es 50%-50%. En los datos completos, la distribución es de 5,486 registros *MI* (25.12%) *vs* 16,351 pacientes que no lo padecen (74.88%). Es un desbalance ligero en el contexto de problemas de clasificación[30], y en presencia del cual de todas formas se obtuvieron resultados favorables, por lo que no se consideró balancearlo, por ejemplo, por medio de submuestrear la clase mayoritaria o hacer *data augmentation* desfasando ligeramente los registros *MI*. La única consideración hecha con respecto a esto fue la de cambiar los pesos de las clases.

No obstante, es pertinente mencionarlo, ya que en los resultados se apreciará que, aunque las sensibilidades y especificidades son altas, las precisiones pueden ser un poco más bajas de lo deseado. Esto se debe a que el volumen de registros no *MI* es mayor que el de los *MI*, por lo que los falsos positivos representan un peso elevado en el denominador de la métrica. El análisis de un *ECG* es la primera linea de defensa para detectar infartos, tras el cual se hacen estudios posteriores, por lo que si bien sería ideal tener una alta precisión, es más importante en este contexto el poder recuperar a las personas que sí los padezcan.

El valor debajo de la curva *ROC* es del 0.93, el cual es un valor relativamente alto en problemas de clasificación. En el caso del mejor modelo, los puntos de corte para determinar el enfoque (*target* o *equal*) están bastante cercanos, ya que incluso con un enfoque *equal* se está logrando el 87.19% de sensibilidad, tan solo a 2.81% del mínimo 90% en el *target*. Si se dejara un punto de corte en el 0.5, la sensibilidad aumenta considerablemente hasta el 94.26%, pero eso es a costa de una pérdida significativa en la precisión (corte 0.5 : 52.84%, *equal* : 68.37%, *target* : 63.05%) [Fig. 25].

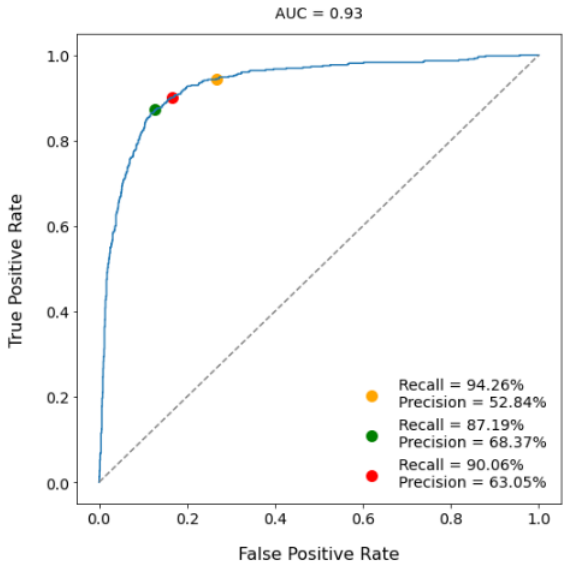


Fig. 25: Curva *ROC* del mejor modelo. En naranja se muestra el punto de corte del 0.5, en verde el del enfoque *equal* (igualar sensibilidad a especificidad) y en rojo el de *target* (mínimo 90% de sensibilidad).

Un vistazo a las matrices de confusión esclarece el efecto que tiene el volumen de los datos en cada clase sobre las métricas [Fig. 26]. Si bien el 94.26% en sensibilidad del punto de corte 0.5 suena atractivo, esto resultaría en una caída importante en la especificidad (73.51%) y precisión (52.84%) en comparación con los enfoques *target* e *equal*.

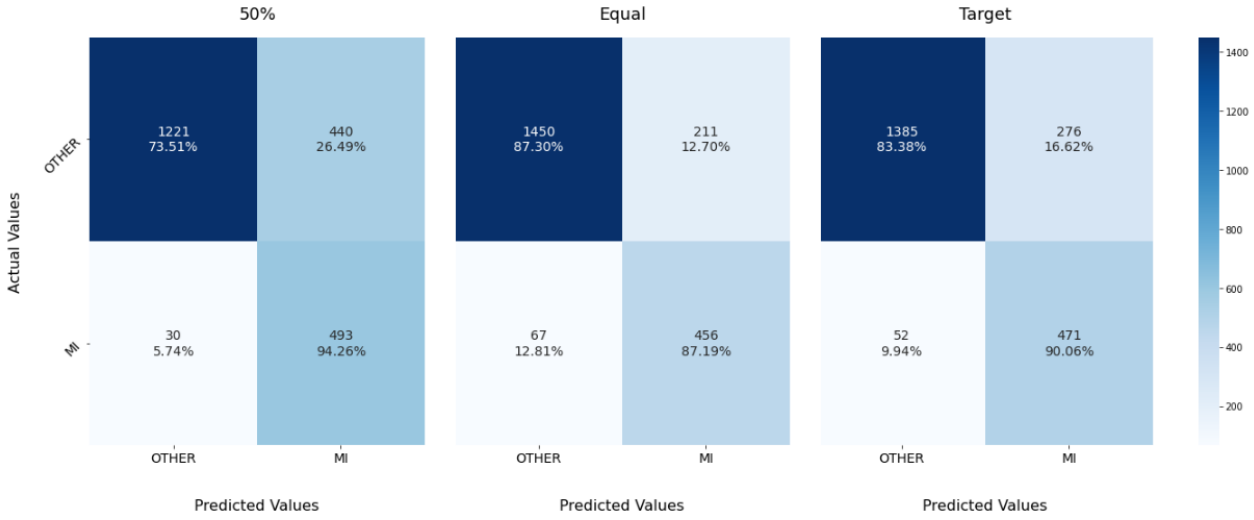


Fig. 26: Matrices de confusión. En cada matriz, la intensidad del color es función de la cantidad de registros en cada cuadrante: izquierda superior son verdaderos negativos, derecha superior son falsos positivos, izquierda inferior son falsos negativos y derecha inferior son verdaderos positivos. Los porcentajes son la cantidad de registros en el cuadrante sobre la suma por fila, por lo que la esquina superior izquierda muestra la especificidad y la inferior derecha la sensibilidad.

Para comprender mejor los cambios en las métricas, nos podemos apoyar de la distribución de la salida en función de si los pacientes tuvieron *MI* o no [Fig. 27]. Se obtuvo una distribución considerablemente buena, ya que visualmente se aprecia que la gran mayoría de registros que no tuvieron *MI* se concentran en valores cercanos al 0, y lo mismo ocurre para los *MI* alrededor del 1. La mayor confusión parece estar ocurriendo a partir del punto de corte del 0.7 en varios registros que no son *MI*. Es ahí donde hay muchos que estarían siendo clasificados como *MI* dependiendo del punto de corte, pero también se perderían más de los que sí padecen el diagnóstico conforme nos movamos a la derecha.

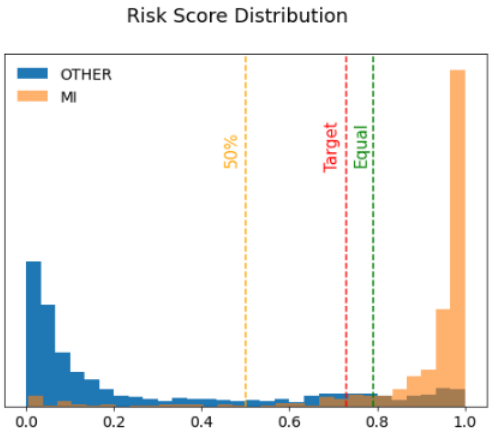


Fig. 27: Distribución de la salida del modelo. En azul se muestra la distribución de los que no son *MI*, en naranja de los que sí lo son. La lineas punteadas muestran el punto de corte para cada enfoque.

Es interesante que los puntos de corte para los enfoques *target* e *equal* hayan estado por encima del 0.5. En muchos de los modelos, antes de quedarnos con el mejor según π, no ocurría esto. La razón puede derivarse de haber determinado la relación 10 a 1 en los pesos de clase, con lo que el modelo está aprendiendo mucho más de los *MI* que de los que no lo padecen.

En resumen, los resultados del modelo fueron favorables de acuerdo con los doctores del *INC* [Tab. 2]. Como nos mencionaron, las métricas parecen ser superiores a las de médicos no especializados en cardiología. Si este modelo se llega a poner en práctica en la vida real, lo ideal sería discutir el punto de corte definitivo con ellos, una vez que se reentrene con registros de la población objetivo.

Enfoque	Sensibilidad	Especificidad	Precisión
50%	94.26%	73.51%	52.84%
Equal	87.19%	87.30%	68.37%
Target	90.06%	83.38%	63.05%

Tab. 2: Métricas del mejor modelo por enfoque

9 Descripción del producto de datos

El producto de datos para fines de esta estancia son los *jupyter notebook* en los que se realizó la investigación. En el *notebook* llamado *ecg-eda-ptbxl-v3.ipynb* se incluye la exploración y el tratamiento de los datos. En este se analizaron las distribuciones de las etiquetas, se visualizaron los *ECG* para comprender sus características, se desarrolló la limpieza por medio de filtros de mediana y se exploraron los criterios para filtrar los *outliers*. En el *notebook* titulado *ecg-classification-ptbxl-v3.ipynb* se incluye la generación de modelos predictivos. En la primera parte hay conjuntos de celdas similares en los que sucesivamente se fueron probando nuevas ideas y técnicas en búsqueda de mejorar el desempeño y la generalización, con lo que posteriormente se realizó el análisis de atribuciones, se optimizó la arquitectura tomando esto en cuenta y se obtuvieron los resultados del mejor modelo.

Este proyecto continuará, pues es de interés de todas las partes involucradas el poder poner eventualmente un modelo en producción. El producto de datos que se visualiza a futuro es que se empaquete en una aplicación para celulares. Esta tendrá la capacidad de, a partir de una imagen de un *ECG*, hacer los diagnósticos por medio del modelo. Para llegar a ello, el obstáculo principal es el cómo se procesará el *ECG*, ya sea que se genere un modelo intermedio que convierta la imagen a las 12 señales en milivoltios, o que se cambie la estructura del modelo para que reciba la imagen tal cual como insumo.

10 Conclusiones y recomendaciones.

Las redes neuronales convolucionales resultaron ser una buena opción de arquitectura para detectar la presencia de infarto de miocardio a partir de electrocardiogramas. Aprovechar su capacidad de reconocer patrones locales en conjunto con diversas técnicas de aprendizaje profundo derivó en un desempeño superior al de médicos no especializados en cardiología. Con el estudio de la distribución de las salidas del modelo y las métricas que se inducen dado el punto de corte, se puede tomar una decisión en función de a que se le prestará más atención, ya sea enfatizar la recuperación de pacientes que sí tengan infarto, o ser más precisos con los diagnósticos predecidos.

Para las siguientes revisiones, se tiene contemplado implementar diversas ideas que podrían mejorar las predicciones. Entre ellas están el implementar mecanismos de atención, como lo son el *Encoder* de un *Transformer* o bloques *Squeeze-Excitation*, ocupar bloques residuales, o buscar pesos preentrenados para hacer *Transfer Learning*. También se tiene en el radar el ocupar redes recurrentes, como una *LSTM*, tal que sea una rama que converja con la parte convolucional y los datos tabulares en la capa densa final.

También se puede explotar la información determinística que se deriva de los segmentos PQRST para calcular métricas similares a las que ocupan los médicos para hacer diagnósticos. Por ejemplo, ver si hay elevación del segmento ST en más de 0.15 mV en más de 3 señales, ritmo cardíaco, varianza de las distancias entre picos R, y más de esa índole. De esta forma, contaríamos con datos tabulares que podrían añadirse en la rama creada para contemplar la edad y el sexo.

Una idea para apoyar al médico a entender el por qué el modelo hizo tal predicción es ocupar *saliency maps*[31]. Estos son principalmente ocupados en problamas de clasificación de imagen, para resaltar las áreas de la misma que están contribuyendo mayormente a hacer la predicción. En el contexto de *ECG*, se resaltarían los patrones en las señales que mayormente contribuyeron a hacer el diagnóstico. Esto ayudaría bastante al médico, pues por una parte puede tener más confianza al ver en lo que se está fijando el modelo, y por otra podría facilitarle a detectar falsos positivos y negativos al combinar su criterio con lo que el modelo considera importante.

Referencias

[1] Electrocardiograma. https://medlineplus.gov/spanish/pruebas-de-laboratorio/electrocardiograma/.

[2] James M. McCabe et al. *Physician Accuracy in Interpreting Potential ST-Segment Elevation Myocardial Infarction Electrocardiograms*.

[3] Infarto de miocardio. https://www.cun.es/enfermedades-tratamientos/enfermedades/infarto-miocardio.

[4] *Top 10 aplicaciones del aprendizaje de maquina*. https://www.edureka.co/blog/machine-learning-applications/.

[5] Yonatan Elul et al. *Meeting the unmet needs of clinicians from AI systems showcased for cardiology with deep-learning–based ECG analysis*.

[6] Monitores de eventos cardíacos. https://medlineplus.gov/spanish/ency/article/007700.htm.

[7] P. Wagner et al. Ptb-xl, a large publicly available electrocardiography dataset.

[8] P. Wagner et al. Ptb-xl database. https://www.physionet.org/content/ptb-xl/1.0.2/.

[9] Angiografía. https://www.cun.es/enfermedades-tratamientos/pruebas-diagnosticas/angiografia.

[10] Desviaciones en las señales. http://www.imperialendo.co.uk/Newskills/ecg/ECG1.html.

[11] *Bandpass Filter*. https://en.wikipedia.org/wiki/Band-pass_filter.

[12] Componente de tendencia. http://www5.uva.es/estadmed/datos/series/series1.htm.

[13] Filtro de mediana. https://www.sciencedirect.com/topics/computer-science/median-filter.

[14] Filtro de mediana (*scipy*). https://docs.scipy.org/doc/scipy/reference/generated/scipy.ndimage.median_filter.html.

[15] Tensorflow. https://www.tensorflow.org/?hl=es-419.

[16] Filtro de mediana (*tensorflow*). https://www.tensorflow.org/addons/api_docs/python/tfa/image/median_filter2d.

[17] Pqrst. https://www.youtube.com/watch?v=RYZ4daFwMa8.

[18] Imagen pqrst. https://es.m.wikipedia.org/wiki/Archivo:ECG-PQRST%2Bpopis.svg.

[19] Imagen corazón. https://www.cigna.com/es-us/knowledge-center/hw/cavidades-del-corazn-tp10241.

[20] wfdb (documentacion). https://wfdb.readthedocs.io/en/latest/.

[21] wfdb (explicacion). https://wfdb.io.

[22] M. A. Z. Fariha et al. *Analysis of Pan-Tompkins Algorithm Performance with Noisy ECG Signals*.

[23] Ectopía ventricular. https://medlineplus.gov/spanish/ency/article/001100.htm.

[24] *CNN* en visión por computadora. https://www.analyticsvidhya.com/blog/2021/01/image-classification-using-convolutional-neural-networks-a-step-by-step-guide/.

[25] Annamalai Natarajan et al. A wide and deep transformer neural network for 12-lead ecg classification.

[26] Zhibin Zhao et al. *Adaptive Lead Weighted ResNet Trained With Different Duration Signals for Classifying 12-lead ECGs*.

[27] E. A. Perez Alday et al. *Classification of 12-lead ECGs: The PhysioNet/Computing in Cardiology Challenge 2020*.

[28] S. Hong et al. *Practical Lessons on 12-Lead ECG Classification: Meta-Analysis of Methods From PhysioNet/Computing in Cardiology Challenge 2020*.

[29] *Github antonior92*. https://github.com/antonior92/automatic-ecg-diagnosis.

[30] Desbalance en problemas de clasificación. https://developers.google.com/machine-learning/data-prep/construct/sampling-splitting/imbalanced-data?hl=es-419.

[31] *Saliency Map*. https://www.analyticsvidhya.com/blog/2022/06/introduction-to-saliency-map-in-an-image-with-tensorflow-2-x-api/.