

1. Resultados

1.1. RAW data

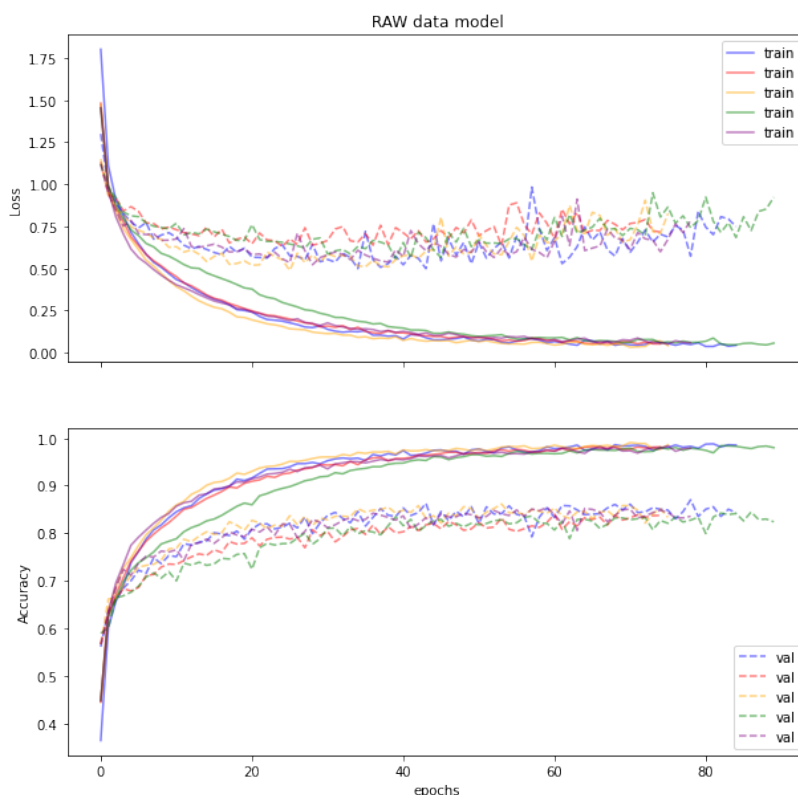


Figura 1: Curva de loss y accuracy sobre conjunto de entrenamiento y validación para modelo de raw data. Se utiliza early stopping con paciencia de 50 épocas para encontrar el mejor modelo.

A este modelo se le entregan los datos de las ventanas sin procesar, y se tiene una arquitectura de dos capas densas con 32 neuronas cada una (la misma que en clases auxiliares). El modelo se entrena por 500 épocas con una tasa de aprendizaje de 0.003, utilizando la estrategia de *early stopping* para encontrar el mejor modelo sobre el conjunto de validación. Luego, se evalúa el modelo sobre el conjunto de test para obtener el accuracy.

El tamaño de la ventanas a utilizar es de 500 con un traslape de 200 samples.

Tabla 1: Iteraciones del modelo sobre los datos, se obtiene el mejor modelo en la época, con un loss y accuracy mostrado.

epoch	loss	acc
85	0.5816	0.8648
75	0.5330	0.8458
76	0.6041	0.8507
90	0.6702	0.8451
79	0.6554	0.8317

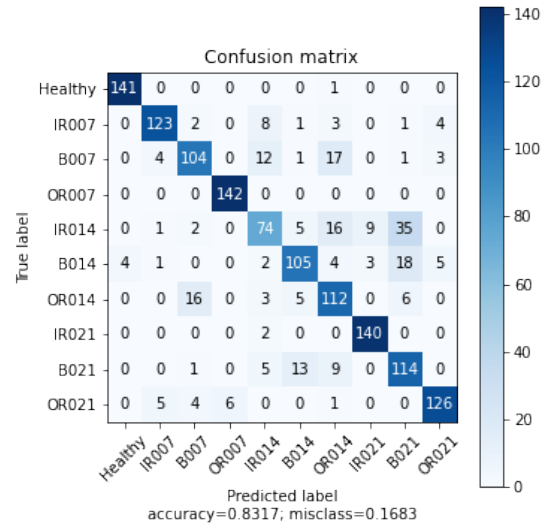


Figura 2: Matriz de confusión entrenamiento modelo con raw data

1.2. Características extraídas de los datos

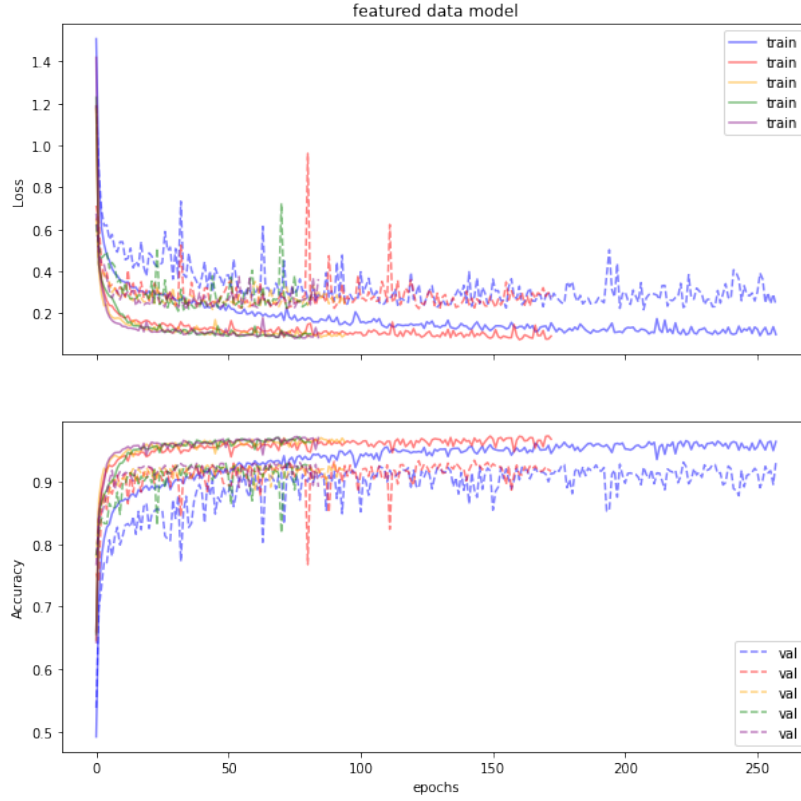


Figura 3: Curva de loss y accuracy para entrenamiento sobre features extraídos de las ventanas de tiempo.

La extracción de características se realiza sobre las ventanas de tiempo y sobre la primera, segunda y tercera derivada de la señal, además de la integral. Se obtienen las siguientes características: promedio, varianza, la raíz del promedio, el máximo, el mínimo, el rango, crestfactor, skewness y kurtosis. Obteniendo así 45 características de la serie de tiempo.

Tabla 2: Iteraciones del modelo sobre los datos de características.

epoch	loss	acc
258	0.2610	0.9063
173	0.3047	0.8944
95	0.2495	0.9085
82	0.3054	0.9035
85	0.3274	0.8894

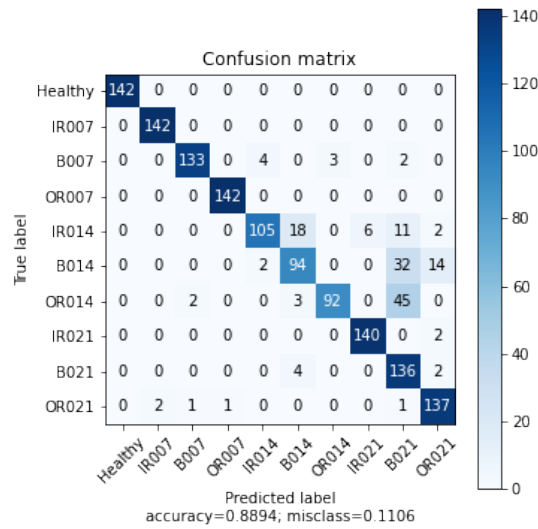


Figura 4: Matriz de confusión entrenamiento sobre conjunto de características de las ventanas.

1.3. Spectrogramas de los datos

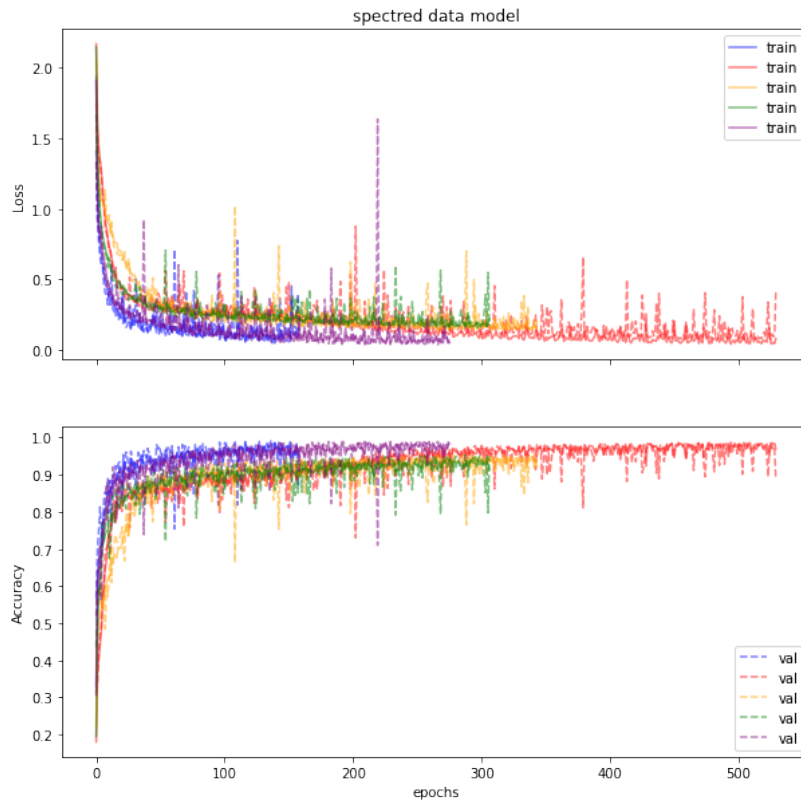


Figura 5: Curva de Loss y accuracy para entrenamiento sobre el conjunto de espectros de las ventanas.

Para la extracción de los espectrogramas, se utiliza la función `scipy.signal.spectrogram` cuyos parámetros se ajustan para obtener la misma cantidad de ventanas en las que se divide el conjunto original, así obteniendo una cantidad de 251 características a entregar a la red.

Tabla 3: Iteraciones del modelo sobre los datos de espectrogramas.

epoch	loss	acc
159	0.1415	0.9331
530	0.3547	0.8944
344	0.1199	0.9599
307	0.1520	0.9563
276	0.0685	0.9796

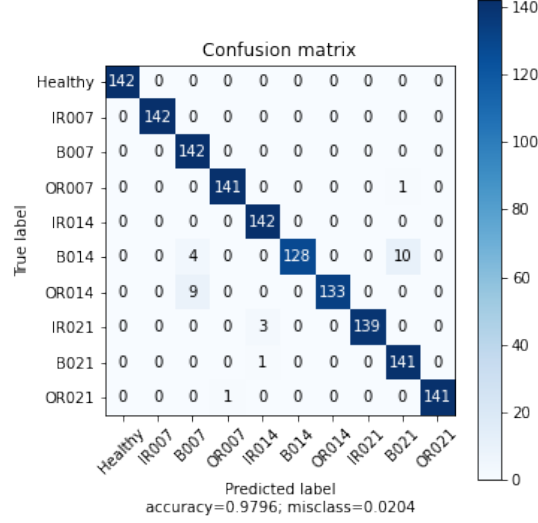


Figura 6: Matriz de confusión de modelo sobre espectrogramas.

2. Análisis

Con una misma arquitectura para todos los conjuntos de características, se tiene que el mejor modelo es el que utiliza los espectrogramas de las ventanas de tiempo. Seguido por el que utiliza las características extraídas de los datos y muy de cerca está el que utiliza los datos crudos.

Con el entrenamiento de cada uno, se puede visualizar que el primero modelo tiene un overfit muy grande y con ello se obtiene que la curva de loss sobre el conjunto de validación, se separa muy pronto de la del conjunto de train. Por ello, tampoco obtiene un buen rendimiento en comparación a los demás.

Mientras que el modelo que más demora, es el que utiliza los espectrogramas, ya que corre muchas más épocas que los modelos anteriores, pero logra un accuracy mucho mayor. También se puede ver que este modelo no tiene overfit tan notorio y rápido como los modelos anteriores.

3. Conclusiones

Se puede ver que el mejor modelo se obtiene al procesar los datos para sacar los espectrogramas y luego pasarlos por una red densa con dos capas ocultas y por un mínimo de 600 épocas. Una observación que se tiene en este entrenamiento, es que al utiliza una tasa de aprendizaje muy baja o muy alta, la red no puede optimizar los pesos, es por ello que demora tanto en entrenar, ya que con un valor muy alto, la red no aprende.