**Models analysis**

**CNN Model**

- Purpose: The CNN model is designed to extract features from input sequences using convolutional filters, followed by down-sampling to reduce dimensionality and computational load.

**Layer Details**

1. Conv1D (1st layer):

- Filters: 128

- Kernel Size: 3

- Activation Function: 'relu'

- Input Shape: (X\_train.shape[1], 1) (sequence length, 1)

2. MaxPooling1D (1st layer):

- Pool Size: 2

3. Conv1D (2nd layer):

- Filters: 64

- Kernel Size: 3

- Activation Function: 'relu'

4. MaxPooling1D (2nd layer):

- Pool Size: 2

5. Flatten Layer:

- Converts 2D feature maps into 1D feature vectors.

6. Dense Layer:

- Units: 256

- Activation Function: 'relu'

7. Dropout Layer:

- Rate: 0.5 (used to help prevent overfitting).

**LSTM Model**

- Purpose: The LSTM model is structured to handle sequential data and is capable of learning long-term dependencies within the input sequences.

**Layer Details**

1. LSTM (1st layer):

- Units: 128

- Input Shape: (X\_train.shape[1], 1) (sequence length, 1)

- Return Sequences: True (so that the output can be fed into the next LSTM layer).

2. LSTM (2nd layer):

- Units: 64

3. Dense Layer:

- Units: 128

- Activation Function: 'relu'

4. Dropout Layer:

- Rate: 0.5 (to mitigate overfitting).

**Ensemble** **Model**

- Purpose: The ensemble model combines the outputs of both the CNN and LSTM models. This leverages the feature extraction capabilities of CNNs alongside the temporal processing of LSTMs.

**Layer Details**

1. Concatenate Layer:

- Combines the outputs from the CNN and LSTM models.

2. Dense (Final Layer):

- Units: 3 (for three potential classes)

- Activation Function: 'softmax' (to produce probability distributions for the classes).