I used:  
X\_jma shape: (85330, 5, 10)

y\_jma shape: (85330, 1)  
  
X\_test shape: (17066, 5, 10)

y\_test shape: (17066, 1)  
  
  
Model: "model\_1"

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Layer (type) Output Shape Param # Connected to

==================================================================================================

input\_4 (InputLayer) [(None, 5, 10)] 0 []

conv1d\_12 (Conv1D) (None, 5, 16) 816 ['input\_4[0][0]']

batch\_normalization\_12 (Ba (None, 5, 16) 64 ['conv1d\_12[0][0]']

tchNormalization)

max\_pooling1d\_12 (MaxPooli (None, 3, 16) 0 ['batch\_normalization\_12[0][0]

ng1D) ']

conv1d\_13 (Conv1D) (None, 3, 32) 1568 ['max\_pooling1d\_12[0][0]']

batch\_normalization\_13 (Ba (None, 3, 32) 128 ['conv1d\_13[0][0]']

tchNormalization)

max\_pooling1d\_13 (MaxPooli (None, 2, 32) 0 ['batch\_normalization\_13[0][0]

ng1D) ']

conv1d\_14 (Conv1D) (None, 2, 64) 6208 ['max\_pooling1d\_13[0][0]']

batch\_normalization\_14 (Ba (None, 2, 64) 256 ['conv1d\_14[0][0]']

tchNormalization)

max\_pooling1d\_14 (MaxPooli (None, 1, 64) 0 ['batch\_normalization\_14[0][0]

ng1D) ']

conv1d\_15 (Conv1D) (None, 1, 128) 24704 ['max\_pooling1d\_14[0][0]']

batch\_normalization\_15 (Ba (None, 1, 128) 512 ['conv1d\_15[0][0]']

tchNormalization)

max\_pooling1d\_15 (MaxPooli (None, 1, 128) 0 ['batch\_normalization\_15[0][0]

ng1D) ']

bidirectional\_3 (Bidirecti (None, 1, 256) 263168 ['max\_pooling1d\_15[0][0]']

onal)

dropout\_2 (Dropout) (None, 1, 256) 0 ['bidirectional\_3[0][0]']

bidirectional\_4 (Bidirecti (None, 1, 128) 164352 ['dropout\_2[0][0]']

onal)

dropout\_3 (Dropout) (None, 1, 128) 0 ['bidirectional\_4[0][0]']

attention\_1 (Attention) (None, 1, 128) 0 ['dropout\_3[0][0]',

'dropout\_3[0][0]']

dense\_3 (Dense) (None, 1, 32) 4128 ['attention\_1[0][0]']

dense\_4 (Dense) (None, 1, 10) 330 ['dense\_3[0][0]']

dense\_5 (Dense) (None, 1, 1) 11 ['dense\_4[0][0]']

==================================================================================================

Total params: 466245 (1.78 MB)

Trainable params: 465765 (1.78 MB)

Non-trainable params: 480 (1.88 KB)

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
  
  
  
  
# Callbacks  
from tensorflow.keras.callbacks import EarlyStopping, ReduceLROnPlateau  
early\_stopping = EarlyStopping(monitor='val\_loss',  
 patience=10,  
 restore\_best\_weights=True)  
  
  
reduce\_lr = ReduceLROnPlateau(monitor='val\_loss',  
 factor=0.1,  
 patience=5,  
 min\_lr=1e-6)  
  
# Train the model  
history = model.fit(X\_jma, y\_jma,  
 batch\_size=64,  
 epochs=100,  
 validation\_split=0.2,  
 callbacks=[early\_stopping])

Epoch 1/100

1067/1067 [==============================] - 13s 8ms/step - loss: 0.0114 - mae: 0.0809 - val\_loss: 0.0082 - val\_mae: 0.0720

Epoch 2/100

1067/1067 [==============================] - 7s 7ms/step - loss: 0.0073 - mae: 0.0651 - val\_loss: 0.0085 - val\_mae: 0.0745

Epoch 3/100

1067/1067 [==============================] - 7s 7ms/step - loss: 0.0065 - mae: 0.0613 - val\_loss: 0.0067 - val\_mae: 0.0640

Epoch 4/100

1067/1067 [==============================] - 7s 7ms/step - loss: 0.0062 - mae: 0.0596 - val\_loss: 0.0060 - val\_mae: 0.0591

Epoch 5/100

1067/1067 [==============================] - 7s 7ms/step - loss: 0.0060 - mae: 0.0586 - val\_loss: 0.0057 - val\_mae: 0.0569

Epoch 6/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0059 - mae: 0.0581 - val\_loss: 0.0057 - val\_mae: 0.0582

Epoch 7/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0058 - mae: 0.0577 - val\_loss: 0.0069 - val\_mae: 0.0662

Epoch 8/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0058 - mae: 0.0574 - val\_loss: 0.0063 - val\_mae: 0.0623

Epoch 9/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0057 - mae: 0.0571 - val\_loss: 0.0060 - val\_mae: 0.0599

Epoch 10/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0056 - mae: 0.0567 - val\_loss: 0.0054 - val\_mae: 0.0552

Epoch 11/100

1067/1067 [==============================] - 7s 7ms/step - loss: 0.0056 - mae: 0.0567 - val\_loss: 0.0055 - val\_mae: 0.0557

Epoch 12/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0056 - mae: 0.0564 - val\_loss: 0.0058 - val\_mae: 0.0589

Epoch 13/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0055 - mae: 0.0561 - val\_loss: 0.0056 - val\_mae: 0.0566

Epoch 14/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0055 - mae: 0.0558 - val\_loss: 0.0069 - val\_mae: 0.0660

Epoch 15/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0054 - mae: 0.0557 - val\_loss: 0.0056 - val\_mae: 0.0561

Epoch 16/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0054 - mae: 0.0553 - val\_loss: 0.0057 - val\_mae: 0.0573

Epoch 17/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0053 - mae: 0.0552 - val\_loss: 0.0056 - val\_mae: 0.0561

Epoch 18/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0053 - mae: 0.0550 - val\_loss: 0.0056 - val\_mae: 0.0556

Epoch 19/100

1067/1067 [==============================] - 8s 7ms/step - loss: 0.0053 - mae: 0.0549 - val\_loss: 0.0068 - val\_mae: 0.0652

Epoch 20/100

1067/1067 [==============================] - 7s 7ms/step - loss: 0.0052 - mae: 0.0547 - val\_loss: 0.0055 - val\_mae: 0.0555  
  
  
I got:  
534/534 [==============================] - 1s 2ms/step - loss: 0.0055 - mae: 0.0554

Test Loss (MSE): 0.005500436760485172

Test MAE: 0.055425096303224564  
  
534/534 [==============================] - 1s 1ms/step

Test RMSE: 0.07416494240247815

R^2 Score: 0.5015814295333142  
  
  


