from keras.layers import Dense, Dropout, Activation

from keras.layers.recurrent import LSTM, GRU

from keras.models import Sequential

def get\_lstm(units):

model = Sequential()

model.add(LSTM(units[1], input\_shape=(units[0], 1), return\_sequences=True))

model.add(LSTM(units[2]))

model.add(Dropout(0.2))

model.add(Dense(units[3], activation='sigmoid'))

return model

def get\_gru(units):

model = Sequential()

model.add(GRU(units[1], input\_shape=(units[0], 1), return\_sequences=True))

model.add(GRU(units[2]))

model.add(Dropout(0.2))

model.add(Dense(units[3], activation='sigmoid'))

return model

def \_get\_sae(inputs, hidden, output):

model = Sequential()

model.add(Dense(hidden, input\_dim=inputs, name='hidden'))

model.add(Activation('sigmoid'))

model.add(Dropout(0.2))

model.add(Dense(output, activation='sigmoid'))

return model

def get\_saes(layers):

sae1 = \_get\_sae(layers[0], layers[1], layers[-1])

sae2 = \_get\_sae(layers[1], layers[2], layers[-1])

sae3 = \_get\_sae(layers[2], layers[3], layers[-1])

saes = Sequential()

saes.add(Dense(layers[1], input\_dim=layers[0], name='hidden1'))

saes.add(Activation('sigmoid'))

saes.add(Dense(layers[2], name='hidden2'))

saes.add(Activation('sigmoid'))

saes.add(Dense(layers[3], name='hidden3'))

saes.add(Activation('sigmoid'))

saes.add(Dropout(0.2))

saes.add(Dense(layers[4], activation='sigmoid'))

models = [sae1, sae2, sae3, saes]

return models