Import the libraries

In [1]:
```
#importing the libraries
import tensorflow as tf
from tensorflow import keras
import numpy as np
import pandas as pd
```

```
WARNING:tensorflow:From C:\Users\Teo Boon Kean\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.
```

Load the data and data pre-processing

In [2]:
```
#load the datasets
baseline_df = pd.read_excel('extracted_features_baseline.xlsx')
toolwear_df = pd.read_excel('extracted_features_toolwear.xlsx')
```

In [3]:
```
#labelling the datasets. 0 for baseline, 1 for toolwear. This will be the variable the model tries to predict
baseline_df["state"] = 0
toolwear_df["state"] = 1
```

In [4]:
```
#concantanate the datasets
combined_df = pd.concat([baseline_df, toolwear_df], axis=0)
print(combined_df.shape)
```

```
(840, 67)
```

In [5]:
```
#getting the y label
state = combined_df["state"].values
print(state.shape)
```

```
(840,)
```

In [6]:
```
#getting the features to train the model
features = combined_df.drop('state', axis=1).values
print(features.shape)
```

```
(840, 66)
```

In [7]:
```
#train test split
from sklearn.model_selection import train_test_split
X_train, X_test, Y_train, Y_test = train_test_split(features, state, test_size=0.2, random_state=50)
```

In [8]:
```
#data scalling
from sklearn.preprocessing import StandardScaler

sc = StandardScaler()
X_train = sc.fit_transform(X_train)
X_test = sc.transform(X_test)
```

Specify the architecture of the stacked autoencoder

In [9]:
```
#specify the number of condensed features for the 3 autoencoders. This will be the number of neurons in the hid
condensed_e1 = 50
condensed_e2 = 30
condensed_e3 = 10
```

Construction and training of the first-Level autoencoder

In [10]:
```
#constructing the first Autoencoder

#input layer
in_ae1 = keras.Input(features.shape[1])
#hidden encoded layer
h_ae1 = keras.layers.Dense(condensed_e1)(in_ae1)
#output layer
out_ae1 = keras.layers.Dense(features.shape[1])(h_ae1)

ae1 = keras.Model(in_ae1, out_ae1)
encoder1 = keras.Model(in_ae1, h_ae1)

ae1.summary()
```

```
WARNING:tensorflow:From C:\Users\Teo Boon Kean\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\backend.py:1398: The name tf.executing_eagerly_outside_functions is deprecated. Please use tf.compat.v1.executing_eagerly_outside_functions instead.

Model: "model"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_1 (InputLayer)        [(None, 66)]              0

 dense (Dense)               (None, 50)                3350

 dense_1 (Dense)             (None, 66)                3366

=================================================================
Total params: 6716 (26.23 KB)
Trainable params: 6716 (26.23 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [11]:
```
#compilling and training the first autoencoder
ae1.compile(optimizer='adam', loss='mse')
ae1.fit(X_train, X_train, epochs = 50, batch_size = 8, validation_split = 0.1)
```

```
WARNING:tensorflow:From C:\Users\Teo Boon Kean\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\optimizers\__init__.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.

Epoch 1/50
WARNING:tensorflow:From C:\Users\Teo Boon Kean\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\utils\tf_utils.py:492: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

76/76 [==============================] - 1s 4ms/step - loss: 0.9498 - val_loss: 0.4013
Epoch 2/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3305 - val_loss: 0.2230
Epoch 3/50
76/76 [==============================] - 0s 2ms/step - loss: 0.1962 - val_loss: 0.1463
Epoch 4/50
76/76 [==============================] - 0s 2ms/step - loss: 0.1315 - val_loss: 0.1025
Epoch 5/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0950 - val_loss: 0.0772
Epoch 6/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0734 - val_loss: 0.0611
Epoch 7/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0597 - val_loss: 0.0511
Epoch 8/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0502 - val_loss: 0.0430
Epoch 9/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0431 - val_loss: 0.0376
Epoch 10/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0374 - val_loss: 0.0330
Epoch 11/50
76/76 [==============================] - 0s 3ms/step - loss: 0.0330 - val_loss: 0.0292
Epoch 12/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0291 - val_loss: 0.0265
Epoch 13/50
76/76 [==============================] - 0s 3ms/step - loss: 0.0262 - val_loss: 0.0238
Epoch 14/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0237 - val_loss: 0.0222
Epoch 15/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0215 - val_loss: 0.0202
Epoch 16/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0198 - val_loss: 0.0193
Epoch 17/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0183 - val_loss: 0.0175
Epoch 18/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0170 - val_loss: 0.0160
Epoch 19/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0158 - val_loss: 0.0152
Epoch 20/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0147 - val_loss: 0.0141
Epoch 21/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0138 - val_loss: 0.0131
Epoch 22/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0128 - val_loss: 0.0129
Epoch 23/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0122 - val_loss: 0.0120
Epoch 24/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0114 - val_loss: 0.0116
Epoch 25/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0108 - val_loss: 0.0111
Epoch 26/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0102 - val_loss: 0.0100
```

```
Epoch 27/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0096 - val_loss: 0.0092
Epoch 28/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0090 - val_loss: 0.0092
Epoch 29/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0086 - val_loss: 0.0084
Epoch 30/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0081 - val_loss: 0.0082
Epoch 31/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0077 - val_loss: 0.0077
Epoch 32/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0073 - val_loss: 0.0073
Epoch 33/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0070 - val_loss: 0.0072
Epoch 34/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0066 - val_loss: 0.0067
Epoch 35/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0063 - val_loss: 0.0065
Epoch 36/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0060 - val_loss: 0.0063
Epoch 37/50
76/76 [==============================] - 0s 3ms/step - loss: 0.0058 - val_loss: 0.0057
Epoch 38/50
76/76 [==============================] - 0s 3ms/step - loss: 0.0055 - val_loss: 0.0056
Epoch 39/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0052 - val_loss: 0.0054
Epoch 40/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0050 - val_loss: 0.0051
Epoch 41/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0048 - val_loss: 0.0047
Epoch 42/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0045 - val_loss: 0.0046
Epoch 43/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0044 - val_loss: 0.0043
Epoch 44/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0042 - val_loss: 0.0044
Epoch 45/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0040 - val_loss: 0.0041
Epoch 46/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0038 - val_loss: 0.0039
Epoch 47/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0037 - val_loss: 0.0037
Epoch 48/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0035 - val_loss: 0.0037
Epoch 49/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0035 - val_loss: 0.0033
Epoch 50/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0033 - val_loss: 0.0034
```

Out[11]:
```
<keras.src.callbacks.History at 0x11d3d978250>
```

Encoding the original features with the first-level encoder

In [12]:
```
#obtaining the first level of encoded features using the first encoder
encoded1 = encoder1.predict(X_train)
```

```
21/21 [==============================] - 0s 1ms/step
```

Construction and training of the second-level autoencoder

In [13]:
```
#constructing the second Autoencoder

#input layer
in_ae2 = keras.Input(condensed_e1)
#hidden encoded layer
h_ae2 = keras.layers.Dense(condensed_e2)(in_ae2)
#output layer
out_ae2 = keras.layers.Dense(condensed_e1)(h_ae2)

ae2 = keras.Model(in_ae2, out_ae2)
encoder2 = keras.Model(in_ae2, h_ae2)

ae2.summary()
```

```
Model: "model_2"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_2 (InputLayer)        [(None, 50)]              0

 dense_2 (Dense)             (None, 30)                1530

 dense_3 (Dense)             (None, 50)                1550

=================================================================
Total params: 3080 (12.03 KB)
Trainable params: 3080 (12.03 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [14]:
```
#compilling and training the second autoencoder using the encoded features from the first encoder
ae2.compile(optimizer='adam', loss='mse')
ae2.fit(encoded1, encoded1, epochs = 50, batch_size = 8, validation_split = 0.1)
```

```
Epoch 1/50
76/76 [==============================] - 1s 3ms/step - loss: 1.5454 - val_loss: 0.7970
Epoch 2/50
76/76 [==============================] - 0s 2ms/step - loss: 0.6806 - val_loss: 0.4427
Epoch 3/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3951 - val_loss: 0.2988
Epoch 4/50
76/76 [==============================] - 0s 2ms/step - loss: 0.2691 - val_loss: 0.2177
Epoch 5/50
76/76 [==============================] - 0s 2ms/step - loss: 0.2003 - val_loss: 0.1681
Epoch 6/50
76/76 [==============================] - 0s 2ms/step - loss: 0.1570 - val_loss: 0.1333
Epoch 7/50
76/76 [==============================] - 0s 2ms/step - loss: 0.1272 - val_loss: 0.1089
Epoch 8/50
76/76 [==============================] - 0s 2ms/step - loss: 0.1054 - val_loss: 0.0908
Epoch 9/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0892 - val_loss: 0.0786
Epoch 10/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0775 - val_loss: 0.0683
Epoch 11/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0683 - val_loss: 0.0619
Epoch 12/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0612 - val_loss: 0.0555
Epoch 13/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0554 - val_loss: 0.0514
Epoch 14/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0511 - val_loss: 0.0472
Epoch 15/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0469 - val_loss: 0.0433
Epoch 16/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0435 - val_loss: 0.0412
Epoch 17/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0408 - val_loss: 0.0377
Epoch 18/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0381 - val_loss: 0.0357
Epoch 19/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0355 - val_loss: 0.0338
Epoch 20/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0334 - val_loss: 0.0324
Epoch 21/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0315 - val_loss: 0.0303
Epoch 22/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0297 - val_loss: 0.0285
Epoch 23/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0281 - val_loss: 0.0273
Epoch 24/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0267 - val_loss: 0.0261
Epoch 25/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0253 - val_loss: 0.0243
Epoch 26/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0242 - val_loss: 0.0237
Epoch 27/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0230 - val_loss: 0.0226
Epoch 28/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0221 - val_loss: 0.0215
Epoch 29/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0212 - val_loss: 0.0203
Epoch 30/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0203 - val_loss: 0.0196
Epoch 31/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0195 - val_loss: 0.0191
Epoch 32/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0187 - val_loss: 0.0184
```

```
Epoch 33/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0182 - val_loss: 0.0177
Epoch 34/50
76/76 [==============================] - 0s 3ms/step - loss: 0.0174 - val_loss: 0.0171
Epoch 35/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0167 - val_loss: 0.0161
Epoch 36/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0163 - val_loss: 0.0160
Epoch 37/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0158 - val_loss: 0.0151
Epoch 38/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0153 - val_loss: 0.0150
Epoch 39/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0148 - val_loss: 0.0145
Epoch 40/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0143 - val_loss: 0.0136
Epoch 41/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0139 - val_loss: 0.0135
Epoch 42/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0134 - val_loss: 0.0130
Epoch 43/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0131 - val_loss: 0.0126
Epoch 44/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0130 - val_loss: 0.0125
Epoch 45/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0124 - val_loss: 0.0124
Epoch 46/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0122 - val_loss: 0.0117
Epoch 47/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0118 - val_loss: 0.0118
Epoch 48/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0115 - val_loss: 0.0115
Epoch 49/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0111 - val_loss: 0.0109
Epoch 50/50
76/76 [==============================] - 0s 2ms/step - loss: 0.0110 - val_loss: 0.0109
```

Out[14]:
```
<keras.src.callbacks.History at 0x11d40233e10>
```

Encoding the first-level features with the second-level encoder

In [15]:
```
#obtaining the second level of encoded features using the second encoder
encoded2 = encoder2.predict(encoded1)
```

```
21/21 [==============================] - 0s 1ms/step
```

Construction and training of the third-level autoencoder

In [16]:
```
#constructing the third Autoencoder

#input layer
in_ae3 = keras.Input(condensed_e2)
#hidden encoded layer
h_ae3 = keras.layers.Dense(condensed_e3)(in_ae3)
#output layer
out_ae3 = keras.layers.Dense(condensed_e2)(h_ae3)

ae3 = keras.Model(in_ae3, out_ae3)
encoder3 = keras.Model(in_ae3, h_ae3)

ae3.summary()
```

```
Model: "model_4"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_3 (InputLayer)        [(None, 30)]              0

 dense_4 (Dense)             (None, 10)                310

 dense_5 (Dense)             (None, 30)                330

=================================================================
Total params: 640 (2.50 KB)
Trainable params: 640 (2.50 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

In [17]:
```
#compilling and training the third autoencoder using the encoded features from the second encoder
ae3.compile(optimizer='adam', loss='mse')
ae3.fit(encoded2, encoded2, epochs = 50, batch_size = 8, validation_split = 0.1)
```

```
Epoch 1/50
76/76 [==============================] - 1s 3ms/step - loss: 2.9634 - val_loss: 2.0508
Epoch 2/50
76/76 [==============================] - 0s 2ms/step - loss: 1.9371 - val_loss: 1.4378
Epoch 3/50
76/76 [==============================] - 0s 2ms/step - loss: 1.3860 - val_loss: 1.0655
Epoch 4/50
76/76 [==============================] - 0s 2ms/step - loss: 1.0697 - val_loss: 0.8773
Epoch 5/50
76/76 [==============================] - 0s 2ms/step - loss: 0.8973 - val_loss: 0.7657
Epoch 6/50
76/76 [==============================] - 0s 2ms/step - loss: 0.7824 - val_loss: 0.6811
Epoch 7/50
76/76 [==============================] - 0s 2ms/step - loss: 0.6939 - val_loss: 0.6094
Epoch 8/50
76/76 [==============================] - 0s 2ms/step - loss: 0.6218 - val_loss: 0.5525
Epoch 9/50
76/76 [==============================] - 0s 2ms/step - loss: 0.5620 - val_loss: 0.5022
Epoch 10/50
76/76 [==============================] - 0s 2ms/step - loss: 0.5144 - val_loss: 0.4610
Epoch 11/50
76/76 [==============================] - 0s 2ms/step - loss: 0.4758 - val_loss: 0.4301
Epoch 12/50
76/76 [==============================] - 0s 2ms/step - loss: 0.4462 - val_loss: 0.4060
Epoch 13/50
76/76 [==============================] - 0s 2ms/step - loss: 0.4238 - val_loss: 0.3905
Epoch 14/50
76/76 [==============================] - 0s 2ms/step - loss: 0.4056 - val_loss: 0.3760
Epoch 15/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3924 - val_loss: 0.3668
Epoch 16/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3816 - val_loss: 0.3568
Epoch 17/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3723 - val_loss: 0.3501
Epoch 18/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3649 - val_loss: 0.3424
Epoch 19/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3580 - val_loss: 0.3370
Epoch 20/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3526 - val_loss: 0.3312
Epoch 21/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3476 - val_loss: 0.3291
Epoch 22/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3428 - val_loss: 0.3243
Epoch 23/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3390 - val_loss: 0.3197
Epoch 24/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3349 - val_loss: 0.3178
Epoch 25/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3318 - val_loss: 0.3130
Epoch 26/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3291 - val_loss: 0.3114
Epoch 27/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3265 - val_loss: 0.3085
Epoch 28/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3237 - val_loss: 0.3062
Epoch 29/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3220 - val_loss: 0.3049
Epoch 30/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3199 - val_loss: 0.3039
Epoch 31/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3178 - val_loss: 0.3016
Epoch 32/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3159 - val_loss: 0.2998
Epoch 33/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3139 - val_loss: 0.2995
Epoch 34/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3124 - val_loss: 0.2997
Epoch 35/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3110 - val_loss: 0.2969
Epoch 36/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3097 - val_loss: 0.2955
Epoch 37/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3086 - val_loss: 0.2932
Epoch 38/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3073 - val_loss: 0.2946
Epoch 39/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3058 - val_loss: 0.2925
Epoch 40/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3048 - val_loss: 0.2930
Epoch 41/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3035 - val_loss: 0.2916
Epoch 42/50
```

```
76/76 [==============================] - 0s 2ms/step - loss: 0.3028 - val_loss: 0.2910
Epoch 43/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3019 - val_loss: 0.2898
Epoch 44/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3007 - val_loss: 0.2898
Epoch 45/50
76/76 [==============================] - 0s 2ms/step - loss: 0.3000 - val_loss: 0.2897
Epoch 46/50
76/76 [==============================] - 0s 2ms/step - loss: 0.2994 - val_loss: 0.2882
Epoch 47/50
76/76 [==============================] - 0s 2ms/step - loss: 0.2985 - val_loss: 0.2882
Epoch 48/50
76/76 [==============================] - 0s 2ms/step - loss: 0.2979 - val_loss: 0.2864
Epoch 49/50
76/76 [==============================] - 0s 2ms/step - loss: 0.2970 - val_loss: 0.2870
Epoch 50/50
76/76 [==============================] - 0s 2ms/step - loss: 0.2968 - val_loss: 0.2853
```

Out[17]: `<keras.src.callbacks.History at 0x11d400ca790>`

Constructing the final stacked autoencoder

In [18]:
```
#constructing the stacked autoencoder. Initialising all the layers

#input layer which number of neurons equals the number of original features
l_in = keras.Input(features.shape[1])

#hidden layer of encoder 1
l_e1 = keras.layers.Dense(condensed_e1)(l_in)

#hidden layer of encoder 2
l_e2 = keras.layers.Dense(condensed_e2)(l_e1)

#hidden layer of encoder 3
l_e3 = keras.layers.Dense(condensed_e3)(l_e2)

#hidden layer of decoder 2
l_d2 = keras.layers.Dense(condensed_e2)(l_e3)

#hidden layer of decoder 1
l_d1 = keras.layers.Dense(condensed_e1)(l_d2)

#output layer which is the same as the input
l_out = keras.layers.Dense(features.shape[1])(l_d1)

#defining the autoencode
stacked_ae = keras.Model(l_in, l_out)

stacked_ae.summary()
```

```
Model: "model_6"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 input_4 (InputLayer)        [(None, 66)]              0

 dense_6 (Dense)             (None, 50)                3350

 dense_7 (Dense)             (None, 30)                1530

 dense_8 (Dense)             (None, 10)                310

 dense_9 (Dense)             (None, 30)                330

 dense_10 (Dense)            (None, 50)                1550

 dense_11 (Dense)            (None, 66)                3366

=================================================================
Total params: 10436 (40.77 KB)
Trainable params: 10436 (40.77 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Transferring the trained weights of the 3 autoencoders to the stacked autoencoder

In [19]:
```
#setting the weights of the stacked autoencoder to those in the trained autoencoders

#first encoder layer
stacked_ae.layers[1].set_weights(ae1.layers[1].get_weights())
```

```
#second encoder layer
stacked_ae.layers[2].set_weights(ae2.layers[1].get_weights())
#third encoder layer
stacked_ae.layers[3].set_weights(ae3.layers[1].get_weights())
#first decoder layer
stacked_ae.layers[4].set_weights(ae3.layers[2].get_weights())
#second decoder layer
stacked_ae.layers[5].set_weights(ae2.layers[2].get_weights())
#second decoder layer
stacked_ae.layers[6].set_weights(ae1.layers[2].get_weights())
```

Evaluating the performance of stacked autoencoder

In [20]:
```
from sklearn.metrics import mean_absolute_error

pred = stacked_ae.predict(X_test)
print(mean_absolute_error(X_test,pred))
```

```
6/6 [==============================] - 0s 1ms/step
0.3070091387501804
```