

```
In [1]: import tensorflow as tf
import pandas as pd
import numpy as np
from sklearn.preprocessing import StandardScaler
import tensorflow.keras as k
from sklearn.model_selection import train_test_split
from matplotlib import pyplot as plt
import time
```

WARNING:tensorflow:From C:\Users\amith\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.

Reading the dataset

```
In [2]: df = pd.read_csv('train.csv')
```

```
In [3]: df.head()
```

```
Out[3]:
```

	MEAN_RR	MEDIAN_RR	SDRR	RMSSD	SDSD	SDRR_RMSSD	HR	pNN25	pNN50	SD1	...	HF	H
0	885.157845	853.763730	140.972741	15.554505	15.553371	9.063146	69.499952	11.133333	0.533333	11.001565	...	15.522602	0.4
1	939.425371	948.357865	81.317742	12.964439	12.964195	6.272369	64.363150	5.600000	0.000000	9.170129	...	2.108525	0.0
2	898.186047	907.006860	84.497236	16.305279	16.305274	5.182201	67.450066	13.066667	0.200000	11.533417	...	13.769729	0.5
3	881.757864	893.460030	90.370537	15.720468	15.720068	5.748591	68.809562	11.800000	0.133333	11.119476	...	18.181913	0.5
4	809.625331	811.184865	62.766242	19.213819	19.213657	3.266724	74.565728	20.200000	0.200000	13.590641	...	48.215822	1.8

5 rows × 36 columns

```
In [4]: label = ['condition']
features = ['MEAN_RR', 'RMSSD', 'pNN25', 'pNN50', 'LF', 'HF', 'LF_HF']
```

In []:

```
In [5]: scaler = StandardScaler()
scaler.fit(df[features])
df[features] = scaler.transform(df[features])
```

```
In [6]: X = df[features].values
y = df[label].values
```

```
In [7]: X_train,X_test,y_train,y_test = train_test_split(X,y,test_size = 0.3,random_state = 40)
```

```
In [8]: print(X_train.shape,y_train.shape,X_test.shape,y_test.shape)
(258502, 7) (258502, 1) (110787, 7) (110787, 1)
```

```
In [9]: y_train = k.utils.to_categorical(y_train)
y_test = k.utils.to_categorical(y_test)
```

Creating Model

```
In [10]: from tensorflow.keras import Sequential
from tensorflow.keras.layers import Dense
```

```
In [11]: model = Sequential()
model.add(Dense(50, activation='relu',kernel_initializer = 'he_normal',input_shape=(X_train.shape[1],)))
model.add(Dense(20, activation='relu',kernel_initializer = 'he_normal'))
model.add(Dense(10, activation='relu',kernel_initializer = 'he_normal'))
model.add(Dense(3, activation='softmax'))
```

WARNING:tensorflow:From C:\Users\amith\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\backend.py:873: The name tf.get_default_graph is deprecated. Please use tf.compat.v1.get_default_graph instead.

```
In [12]: model.compile(optimizer='adam', loss='categorical_crossentropy',metrics=['accuracy'])
```

WARNING:tensorflow:From C:\Users\amith\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.

```
In [13]: history = model.fit(X_train,y_train, validation_data = (X_test,y_test), epochs = 50, batch_size = 1024, verbose =1)
```

Epoch 1/50

WARNING:tensorflow:From C:\Users\amith\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\utils\tf_utils.py:49
2: The name tf.ragged.RaggedTensorValue is deprecated. Please use tf.compat.v1.ragged.RaggedTensorValue instead.

WARNING:tensorflow:From C:\Users\amith\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\engine\base_layer_uti
ls.py:384: The name tf.executing_eagerly_outside_functions is deprecated. Please use tf.compat.v1.executing_eagerly_outside_fu
nctions instead.

253/253 [=====] - 4s 7ms/step - loss: 0.8365 - accuracy: 0.6212 - val_loss: 0.6953 - val_accuracy: 0.7018

Epoch 2/50

253/253 [=====] - 1s 5ms/step - loss: 0.6178 - accuracy: 0.7289 - val_loss: 0.5546 - val_accuracy: 0.7686

Epoch 3/50

253/253 [=====] - 1s 5ms/step - loss: 0.5017 - accuracy: 0.7947 - val_loss: 0.4630 - val_accuracy: 0.8097

Epoch 4/50

253/253 [=====] - 1s 5ms/step - loss: 0.4263 - accuracy: 0.8285 - val_loss: 0.4010 - val_accuracy: 0.8396

Epoch 5/50

253/253 [=====] - 1s 4ms/step - loss: 0.3744 - accuracy: 0.8508 - val_loss: 0.3562 - val_accuracy: 0.8581

Epoch 6/50

253/253 [=====] - 1s 5ms/step - loss: 0.3358 - accuracy: 0.8675 - val_loss: 0.3217 - val_accuracy: 0.8745

Epoch 7/50

253/253 [=====] - 1s 5ms/step - loss: 0.3074 - accuracy: 0.8810 - val_loss: 0.2978 - val_accuracy: 0.8844

Epoch 8/50

253/253 [=====] - 1s 5ms/step - loss: 0.2855 - accuracy: 0.8928 - val_loss: 0.2816 - val_accuracy: 0.8940

Epoch 9/50

253/253 [=====] - 1s 5ms/step - loss: 0.2686 - accuracy: 0.9016 - val_loss: 0.2618 - val_accuracy: 0.9042

Epoch 10/50

253/253 [=====] - 1s 5ms/step - loss: 0.2542 - accuracy: 0.9084 - val_loss: 0.2605 - val_accuracy: 0.9038

Epoch 11/50

253/253 [=====] - 1s 4ms/step - loss: 0.2426 - accuracy: 0.9127 - val_loss: 0.2381 - val_accuracy: 0.9138

Epoch 12/50

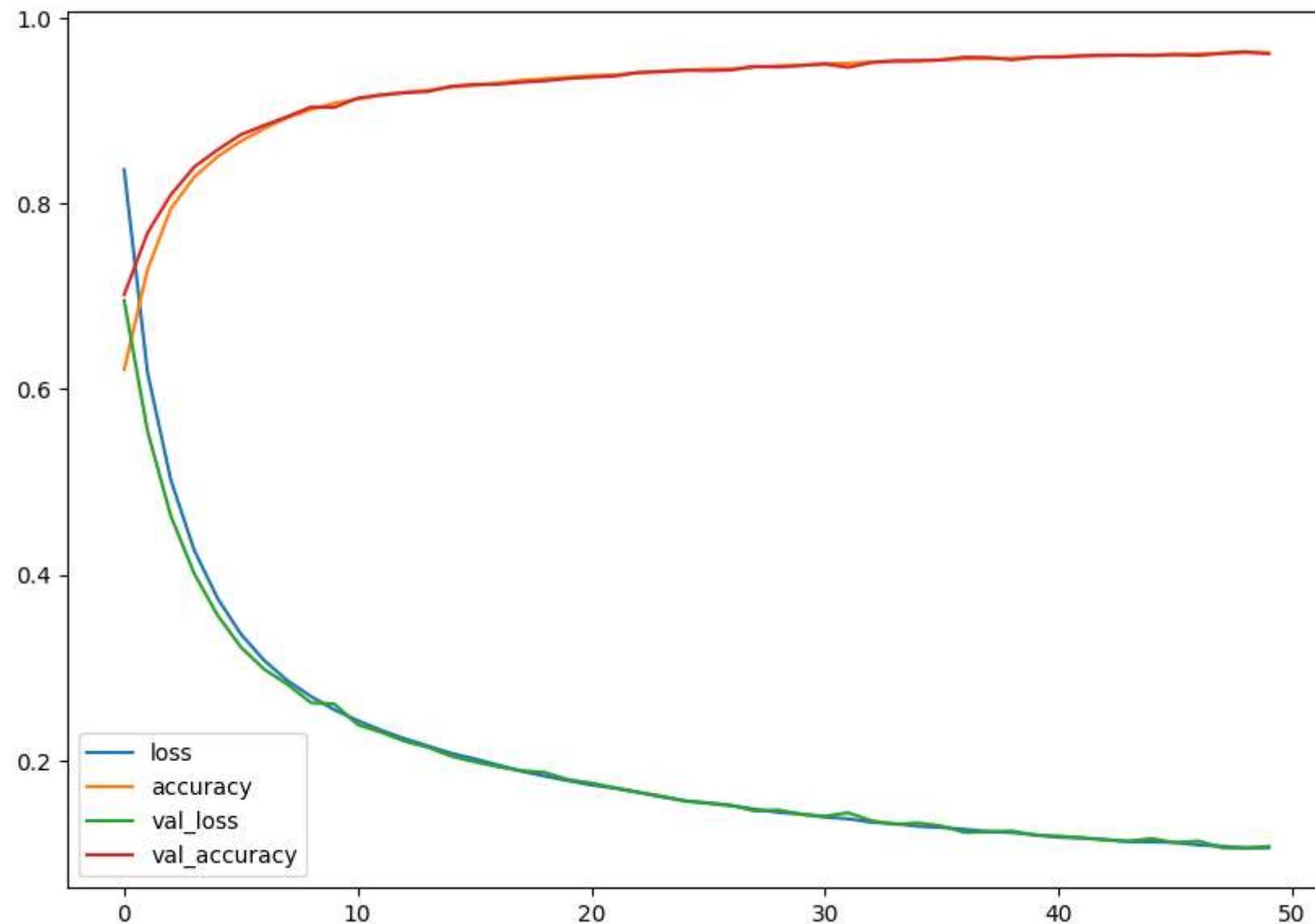
```
253/253 [=====] - 1s 5ms/step - loss: 0.2322 - accuracy: 0.9169 - val_loss: 0.2302 - val_accuracy: 0.9  
173  
Epoch 13/50  
253/253 [=====] - 1s 5ms/step - loss: 0.2231 - accuracy: 0.9200 - val_loss: 0.2204 - val_accuracy: 0.9  
196  
Epoch 14/50  
253/253 [=====] - 1s 4ms/step - loss: 0.2152 - accuracy: 0.9223 - val_loss: 0.2138 - val_accuracy: 0.9  
209  
Epoch 15/50  
253/253 [=====] - 1s 5ms/step - loss: 0.2074 - accuracy: 0.9256 - val_loss: 0.2042 - val_accuracy: 0.9  
265  
Epoch 16/50  
253/253 [=====] - 1s 5ms/step - loss: 0.2014 - accuracy: 0.9273 - val_loss: 0.1982 - val_accuracy: 0.9  
286  
Epoch 17/50  
253/253 [=====] - 1s 5ms/step - loss: 0.1947 - accuracy: 0.9302 - val_loss: 0.1929 - val_accuracy: 0.9  
286  
Epoch 18/50  
253/253 [=====] - 1s 5ms/step - loss: 0.1881 - accuracy: 0.9329 - val_loss: 0.1886 - val_accuracy: 0.9  
309  
Epoch 19/50  
253/253 [=====] - 1s 4ms/step - loss: 0.1828 - accuracy: 0.9347 - val_loss: 0.1867 - val_accuracy: 0.9  
323  
Epoch 20/50  
253/253 [=====] - 1s 5ms/step - loss: 0.1780 - accuracy: 0.9363 - val_loss: 0.1790 - val_accuracy: 0.9  
349  
Epoch 21/50  
253/253 [=====] - 1s 5ms/step - loss: 0.1731 - accuracy: 0.9381 - val_loss: 0.1751 - val_accuracy: 0.9  
364  
Epoch 22/50  
253/253 [=====] - 1s 5ms/step - loss: 0.1698 - accuracy: 0.9388 - val_loss: 0.1702 - val_accuracy: 0.9  
375  
Epoch 23/50  
253/253 [=====] - 1s 5ms/step - loss: 0.1652 - accuracy: 0.9410 - val_loss: 0.1655 - val_accuracy: 0.9  
415  
Epoch 24/50  
253/253 [=====] - 1s 5ms/step - loss: 0.1604 - accuracy: 0.9427 - val_loss: 0.1612 - val_accuracy: 0.9  
423  
Epoch 25/50  
253/253 [=====] - 1s 5ms/step - loss: 0.1563 - accuracy: 0.9438 - val_loss: 0.1557 - val_accuracy: 0.9  
438
```

Epoch 26/50
253/253 [=====] - 1s 5ms/step - loss: 0.1534 - accuracy: 0.9451 - val_loss: 0.1539 - val_accuracy: 0.9434
Epoch 27/50
253/253 [=====] - 1s 5ms/step - loss: 0.1507 - accuracy: 0.9454 - val_loss: 0.1513 - val_accuracy: 0.9441
Epoch 28/50
253/253 [=====] - 1s 5ms/step - loss: 0.1471 - accuracy: 0.9472 - val_loss: 0.1452 - val_accuracy: 0.9481
Epoch 29/50
253/253 [=====] - 1s 5ms/step - loss: 0.1436 - accuracy: 0.9487 - val_loss: 0.1461 - val_accuracy: 0.9472
Epoch 30/50
253/253 [=====] - 1s 5ms/step - loss: 0.1417 - accuracy: 0.9491 - val_loss: 0.1415 - val_accuracy: 0.9490
Epoch 31/50
253/253 [=====] - 1s 5ms/step - loss: 0.1385 - accuracy: 0.9509 - val_loss: 0.1394 - val_accuracy: 0.9506
Epoch 32/50
253/253 [=====] - 1s 5ms/step - loss: 0.1366 - accuracy: 0.9513 - val_loss: 0.1432 - val_accuracy: 0.9469
Epoch 33/50
253/253 [=====] - 1s 4ms/step - loss: 0.1330 - accuracy: 0.9528 - val_loss: 0.1345 - val_accuracy: 0.9522
Epoch 34/50
253/253 [=====] - 1s 5ms/step - loss: 0.1314 - accuracy: 0.9537 - val_loss: 0.1307 - val_accuracy: 0.9539
Epoch 35/50
253/253 [=====] - 1s 5ms/step - loss: 0.1286 - accuracy: 0.9547 - val_loss: 0.1323 - val_accuracy: 0.9537
Epoch 36/50
253/253 [=====] - 1s 5ms/step - loss: 0.1274 - accuracy: 0.9550 - val_loss: 0.1288 - val_accuracy: 0.9550
Epoch 37/50
253/253 [=====] - 1s 5ms/step - loss: 0.1252 - accuracy: 0.9560 - val_loss: 0.1220 - val_accuracy: 0.9580
Epoch 38/50
253/253 [=====] - 1s 4ms/step - loss: 0.1231 - accuracy: 0.9567 - val_loss: 0.1226 - val_accuracy: 0.9575
Epoch 39/50
253/253 [=====] - 1s 5ms/step - loss: 0.1219 - accuracy: 0.9569 - val_loss: 0.1235 - val_accuracy: 0.95

```
550
Epoch 40/50
253/253 [=====] - 1s 5ms/step - loss: 0.1194 - accuracy: 0.9581 - val_loss: 0.1189 - val_accuracy: 0.9
580
Epoch 41/50
253/253 [=====] - 1s 5ms/step - loss: 0.1168 - accuracy: 0.9587 - val_loss: 0.1181 - val_accuracy: 0.9
579
Epoch 42/50
253/253 [=====] - 1s 4ms/step - loss: 0.1159 - accuracy: 0.9591 - val_loss: 0.1162 - val_accuracy: 0.9
594
Epoch 43/50
253/253 [=====] - 1s 4ms/step - loss: 0.1145 - accuracy: 0.9593 - val_loss: 0.1137 - val_accuracy: 0.9
603
Epoch 44/50
253/253 [=====] - 1s 5ms/step - loss: 0.1122 - accuracy: 0.9603 - val_loss: 0.1126 - val_accuracy: 0.9
600
Epoch 45/50
253/253 [=====] - 1s 5ms/step - loss: 0.1118 - accuracy: 0.9603 - val_loss: 0.1154 - val_accuracy: 0.9
596
Epoch 46/50
253/253 [=====] - 1s 5ms/step - loss: 0.1112 - accuracy: 0.9608 - val_loss: 0.1108 - val_accuracy: 0.9
608
Epoch 47/50
253/253 [=====] - 1s 5ms/step - loss: 0.1084 - accuracy: 0.9614 - val_loss: 0.1126 - val_accuracy: 0.9
598
Epoch 48/50
253/253 [=====] - 1s 5ms/step - loss: 0.1071 - accuracy: 0.9622 - val_loss: 0.1055 - val_accuracy: 0.9
621
Epoch 49/50
253/253 [=====] - 1s 5ms/step - loss: 0.1054 - accuracy: 0.9628 - val_loss: 0.1052 - val_accuracy: 0.9
639
Epoch 50/50
253/253 [=====] - 1s 5ms/step - loss: 0.1052 - accuracy: 0.9627 - val_loss: 0.1069 - val_accuracy: 0.9
616
```

```
In [15]: pd.DataFrame(history.history).plot(figsize=(10,7))
```

```
Out[15]: <Axes: >
```



Testing on testset

```
In [16]: df_test = pd.read_csv('test.csv')
```

```
In [17]: df_test[features] = scaler.transform(df_test[features])
df_test[features].head()
```

Out[17]:

	MEAN_RR	RMSSD	pNN25	pNN50	LF	HF	LF_HF
0	-1.001159	-0.634891	-0.598837	-0.874583	-0.575814	0.602911	-0.295775
1	-0.024971	1.048686	1.361573	-0.672601	1.080403	-0.280746	-0.157544
2	0.897836	1.544671	1.743894	0.943254	1.965161	-0.511482	0.037412
3	-0.175046	-0.777935	-0.623241	-0.335965	-0.767443	-0.477196	-0.241658
4	-0.721825	-0.393071	-0.476820	-0.201310	-0.735541	-0.089133	-0.280126

```
In [18]: X = df_test[features].values
y = df_test[label].values
```

```
In [19]: y = k.utils.to_categorical(y)
```

```
In [20]: loss,acc = model.evaluate(X,y,verbose=1)
```

1283/1283 [=====] - 3s 3ms/step - loss: 0.1076 - accuracy: 0.9618

Final Pipeline for prediction

```
In [21]: data = pd.read_csv('test.csv')
```

```
In [22]: t = scaler.transform(data[features].iloc[5545].values.reshape(1,-1))
print(t)
```

[[-0.7699341 0.25391942 0.11699709 0.60661783 0.54497884 0.50787441
-0.26537627]]

```
C:\Users\amith\AppData\Local\Programs\Python\Python311\Lib\site-packages\sklearn\base.py:493: UserWarning: X does not have valid feature names, but StandardScaler was fitted with feature names
  warnings.warn(
```

```
In [23]: y_pred = model.predict(t)
```

```
1/1 [=====] - 0s 188ms/step
```

```
In [24]: print(np.argmax(y_pred[0]))
print(data[label].iloc[5545])
```

```
0
condition    0
Name: 5545, dtype: int64
```

Saving models

```
In [25]: import joblib
```

```
joblib.dump(scaler, 'scaler.joblib', compress=9)
model.save('tfmodel.h5')
```

```
C:\Users\amith\AppData\Local\Programs\Python\Python311\Lib\site-packages\keras\src\engine\training.py:3103: UserWarning: You are saving your model as an HDF5 file via `model.save()`. This file format is considered legacy. We recommend using instead the native Keras format, e.g. `model.save('my_model.keras')`.
  saving_api.save_model(
```

```
In [26]: df = pd.read_csv('train.csv')
```

```
In [ ]:
```