Result and Discussion

On performing data visualization on the dataset, a lot of insights were obtained such as:

5.1 Dataset Analysis

This dataset contains 6 activities having nearly 11 lakhs record. From the figure 5.1, we can see that this dataset contains highly unbalanced data.

```
In [19]:  1 data.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1098203 entries, 0 to 1098203
Data columns (total 6 columns):
 #   Column    Non-Null Count    Dtype
---  ------    --------------    -----
 0   user      1098203 non-null  int64
 1   activity  1098203 non-null  object
 2   time      1098203 non-null  int64
 3   x         1098203 non-null  float64
 4   y         1098203 non-null  float64
 5   z         1098203 non-null  float64
dtypes: float64(3), int64(2), object(1)
memory usage: 58.7+ MB
```

Figure 5.1: Dataset Information

5.2 Records in Activities

In figure 5.2, walking and jogging have more no of records i.e. 424397 and 342176 records respectively while standing has 48395 records only. If we use this dataset as then it is going to highly overfitted and skewed towards walking and jogging. So We need to balance the dataset, for that what we took only 3555 records from each activity.

```
In [21]:  1 data['activity'].value_counts()

Out[21]: Walking       424397
         Jogging       342176
         Upstairs      122869
         Downstairs    100427
         Sitting        59939
         Standing       48395
         Name: activity, dtype: int64
```

Figure 5.2: No. of records for each Activity

5.3 Visualising Accelerometer Data

After exploring dataset, we tried to plot these accelerometer values for timestamp 10sec so that we can see how the accelerometer data looks visually for each activity. Because each activity follows a specific pattern and by looking at these patterns we can classify which accelerometer values belongs which class. From the figure 5.3.1 we can see that, for walking and jogging there is a lots of variation in pattern but for sitting pattern is almost flat. So what our model should learn if there is more variation in pattern then that will be classified into jogging and if there is less of variation that should be classified as sitting.

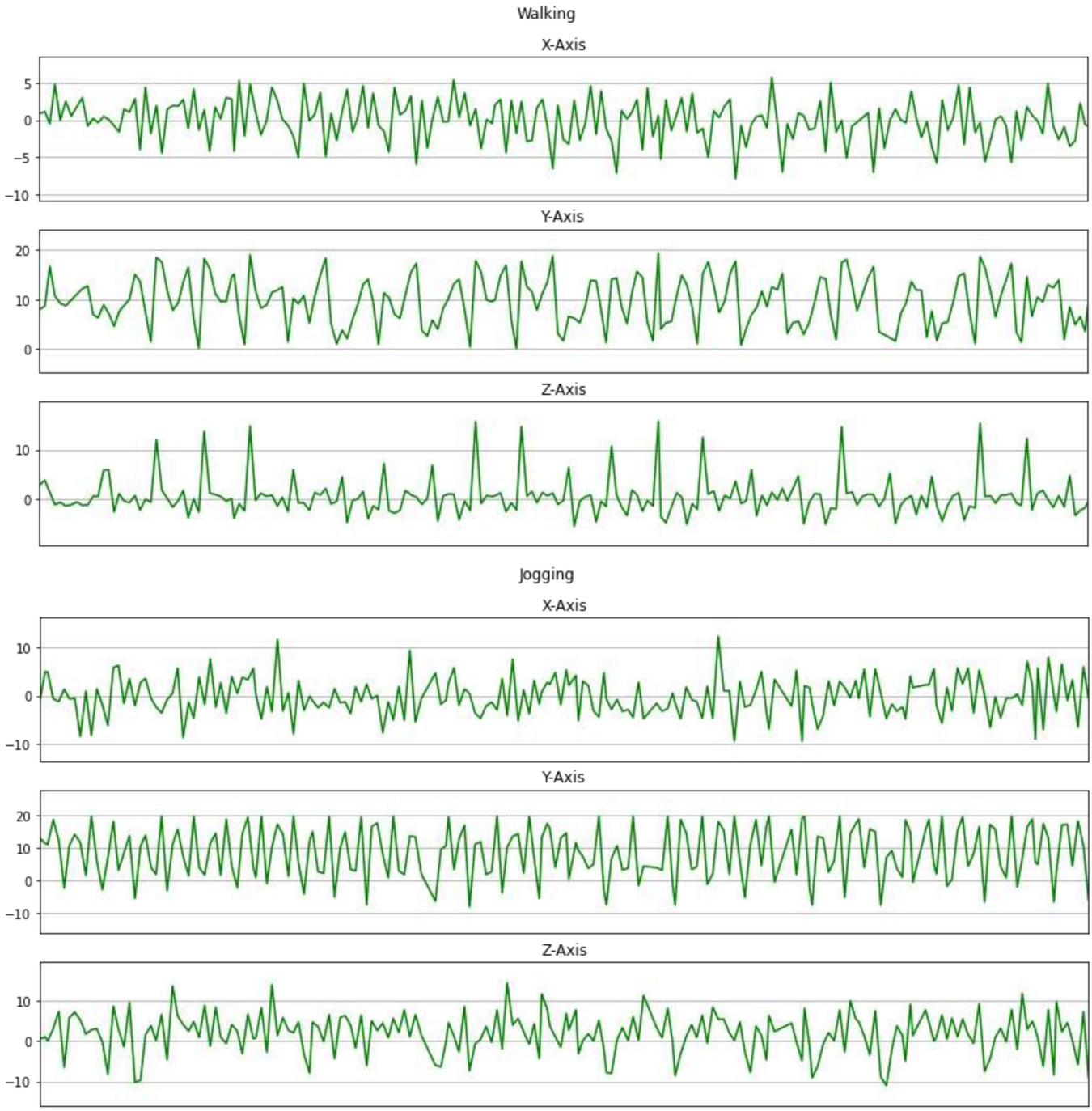

Figure 5.3.1: Walking and Jogging Activitiy

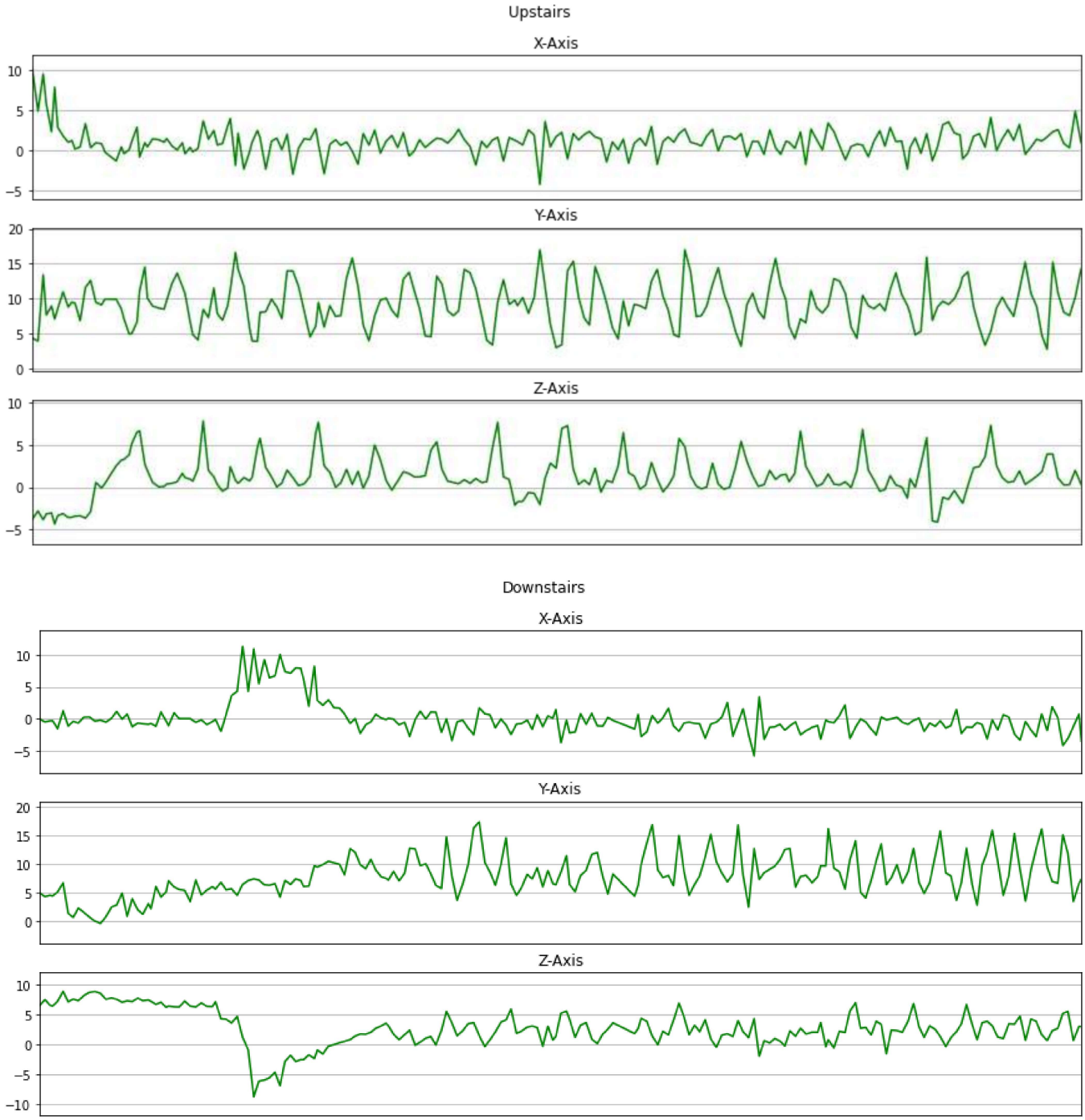

Figure 5.3.1: Upstaires and Downstairs Activitiy

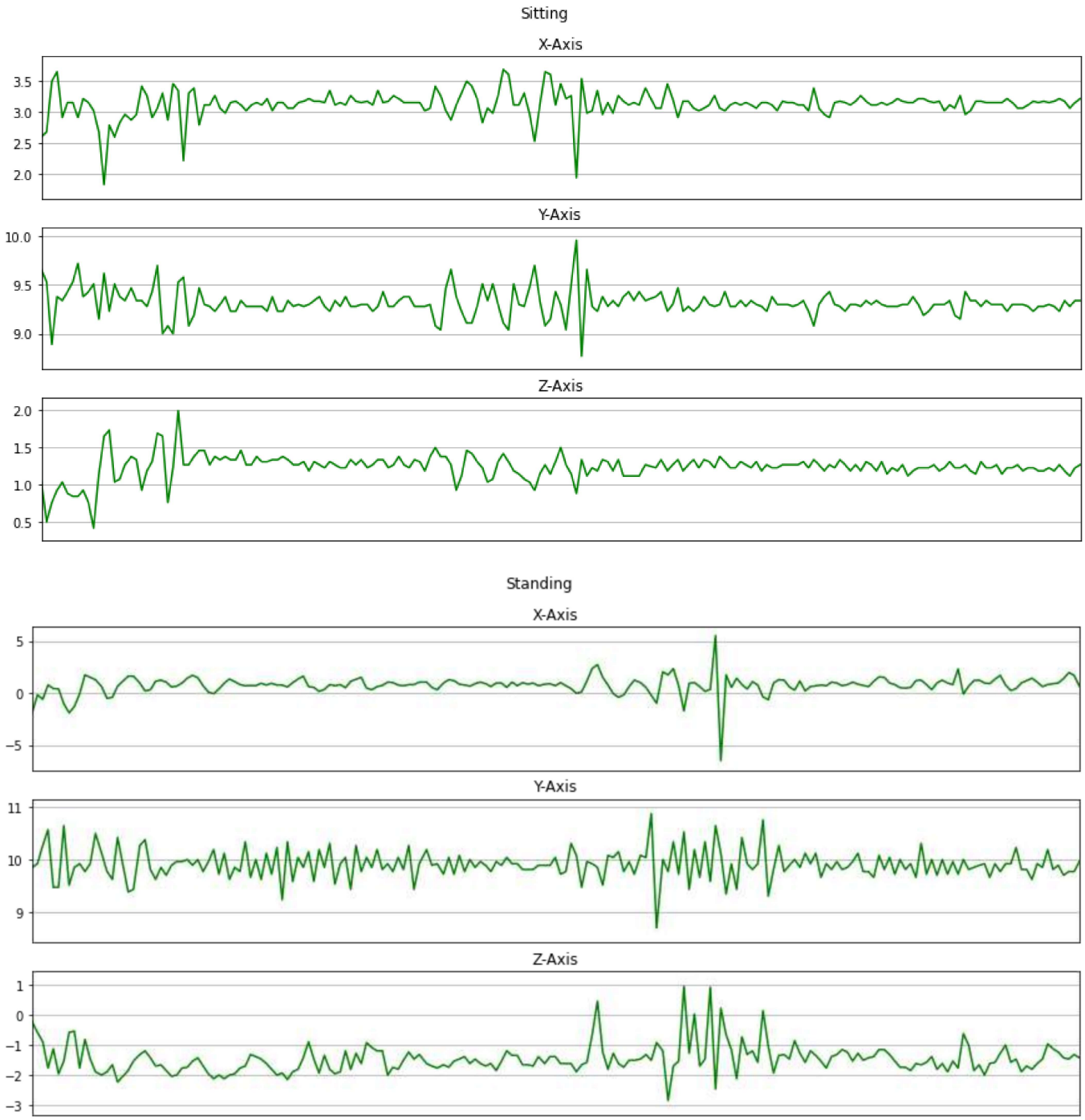

Figure 5.3.1: Sitting and Standing Activitiy

5.4 Implementation Analysis

After implementing this model earlier, we were getting 79.47% training accuracy and 84.57% testing accuracy. Then we tried adjusting frame size and Hopsize. Then by changing it repeatedly, for frame size 300sec and hopsize 40 we got highest percentage of training accuracy and testing accuracy but model was overfitted and loss was very high. But for frame size 80sec and Hopsize 40 we got training accuracy of 92.32% and testing accuracy of 93.11% and loss was too low.

```
1 model.compile(optimizer=Adam(learning_rate=0.001),loss= 'sparse_categorical_
2 history=model.fit(x_train,y_train, epochs = 10, validation_data=(x_test, y_t
```

```
Epoch 1/10
182/182 [==============================] - 2s 7ms/step - loss: 0.9000 - accurac
y: 0.6385 - val_loss: 0.5477 - val_accuracy: 0.7713
Epoch 2/10
182/182 [==============================] - 1s 7ms/step - loss: 0.5721 - accurac
y: 0.7601 - val_loss: 0.4640 - val_accuracy: 0.8113
Epoch 3/10
182/182 [==============================] - 1s 6ms/step - loss: 0.4966 - accurac
y: 0.7947 - val_loss: 0.4112 - val_accuracy: 0.8457
Epoch 4/10
182/182 [==============================] - 1s 6ms/step - loss: 0.4295 - accurac
y: 0.8250 - val_loss: 0.3707 - val_accuracy: 0.8574
Epoch 5/10
182/182 [==============================] - 1s 6ms/step - loss: 0.3730 - accurac
y: 0.8465 - val_loss: 0.3213 - val_accuracy: 0.8919
Epoch 6/10
182/182 [==============================] - 1s 6ms/step - loss: 0.3252 - accurac
y: 0.8688 - val_loss: 0.2895 - val_accuracy: 0.8994
Epoch 7/10
182/182 [==============================] - 1s 6ms/step - loss: 0.2955 - accurac
y: 0.8862 - val_loss: 0.2578 - val_accuracy: 0.9118
Epoch 8/10
182/182 [==============================] - 1s 6ms/step - loss: 0.2563 - accurac
y: 0.8986 - val_loss: 0.2455 - val_accuracy: 0.9174
Epoch 9/10
182/182 [==============================] - 1s 6ms/step - loss: 0.2344 - accurac
y: 0.9079 - val_loss: 0.2291 - val_accuracy: 0.9277
Epoch 10/10
182/182 [==============================] - 1s 6ms/step - loss: 0.2117 - accurac
y: 0.9232 - val_loss: 0.2091 - val_accuracy: 0.9311
```

Figure 5.4: Implementation Analysis

5.5 Learning Analysis

1st graph is plotted between no of epochs and accuracy of model. Then the 2nd graph is plotted between no of epochs and loss of model. Here we got quite good accuracy. As validation loss is less than training loss we can say that our model is neither overfiting and nor underfitting.

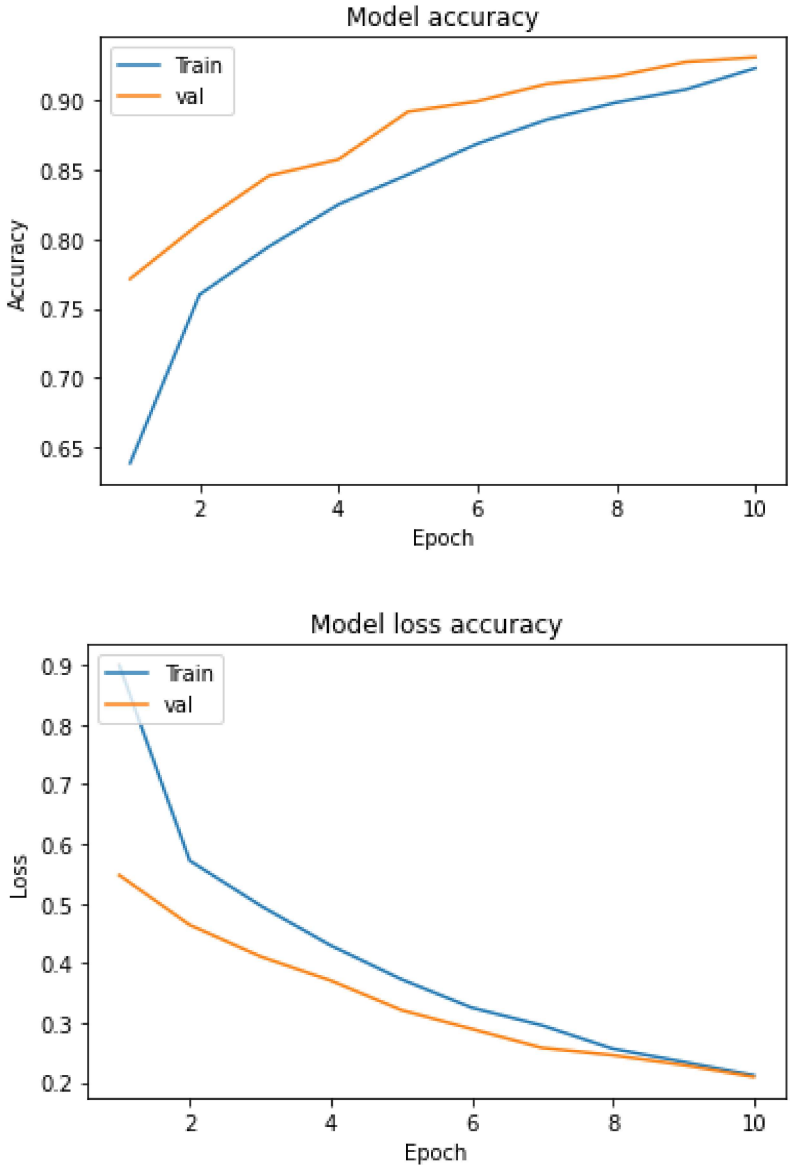

Figure 5.5: Learning Analysis