WN

ABHISHEK

2021121

WIRELESS NETWORK
ASSIGNMENT 4

a

```python
def preprocess_file(file_path):
    result = pd.read_csv(file_path)
    len_= result.shape[0]
    len_=int(abs(0.03*len_))
    # print(len)
    result=result.iloc[len_:-len_, 128:] # remove first 128 rows.

    label= pd.DataFrame(result['headlabel'])
    # label.to_csv(f'{save_path}/{i}_amp/label_{j}', index=False)
    result=result.drop(columns=['timestamp', 'headlabel'], axis=1)

    result.columns=range(len(result.columns))
    # remove null and pilot subcarriers

    delete_idxs =  np.asarray([0,1,2,3,118,119,120,121,122,123,124,125,126,127,128,129,130,131,132,133,134,135,136,137,138,139,254,255])
    result = result.drop(result.columns[delete_idxs], axis=1)
    result.columns=range(len(result.columns))
    data.append(result)
    labels.append(label)
```

b

```python
def amplitude(df):

    df = np.array(df)  # Convert DataFrame to numpy array for efficient computation
    amp = []

    for i in range(len(df)):  # Iterate over each row in the dataset
        imaginary = []
        real = []
        amplitudes = []

        for j in range(len(df[i])):  # Iterate over each element in the row
            if j % 2 == 0:  # Even index corresponds to imaginary part
                imaginary.append(df[i][j])
            else:  # Odd index corresponds to real part
                real.append(df[i][j])

        # Calculate amplitude for each subcarrier
        for k in range(len(imaginary)):
            amplitudes.append(np.sqrt(imaginary[k] ** 2 + real[k] ** 2))

        amp.append(amplitudes)

    # Convert the amplitude list back into a DataFrame
    amp_df = pd.DataFrame(amp)
    amp_df = amp_df.reset_index(drop=True)
    print("Amplitude calculation completed.")
    return amp_df
```

```
Experiment with 1000 retraining samples:
Epoch 1/5
13/13 ━━━━━━━━━━━━━━━━━━━━ 2s 175ms/step - accuracy: 0.3874 - loss: 4.7443 - val_accuracy: 0.4350 - val_loss: 5.0873
Epoch 2/5
13/13 ━━━━━━━━━━━━━━━━━━━━ 1s 92ms/step - accuracy: 0.3966 - loss: 4.2873 - val_accuracy: 0.4100 - val_loss: 5.1030
Epoch 3/5
13/13 ━━━━━━━━━━━━━━━━━━━━ 1s 94ms/step - accuracy: 0.4132 - loss: 4.3112 - val_accuracy: 0.4200 - val_loss: 5.0971
Retraining Time: 4.76 seconds
Testing Time: 11.88 seconds
Testing Accuracy: 0.3757
--------------------------------------------------
```

```
Experiment with 5000 retraining samples:
Epoch 1/5
63/63 ━━━━━━━━━━━━━━━━━━━━ 6s 89ms/step - accuracy: 0.3807 - loss: 4.9879 - val_accuracy: 0.3990 - val_loss: 4.6730
Epoch 2/5
63/63 ━━━━━━━━━━━━━━━━━━━━ 10s 89ms/step - accuracy: 0.3894 - loss: 4.9936 - val_accuracy: 0.4100 - val_loss: 4.6934
Epoch 3/5
63/63 ━━━━━━━━━━━━━━━━━━━━ 7s 113ms/step - accuracy: 0.3861 - loss: 5.0633 - val_accuracy: 0.4130 - val_loss: 4.6729
Epoch 4/5
63/63 ━━━━━━━━━━━━━━━━━━━━ 6s 96ms/step - accuracy: 0.3809 - loss: 5.0840 - val_accuracy: 0.3900 - val_loss: 4.6717
Epoch 5/5
63/63 ━━━━━━━━━━━━━━━━━━━━ 6s 91ms/step - accuracy: 0.3846 - loss: 4.9702 - val_accuracy: 0.4010 - val_loss: 4.6782
Retraining Time: 34.94 seconds
Testing Time: 11.74 seconds
Testing Accuracy: 0.3757
```

```
--------------------------------------------------
Experiment with 10000 retraining samples:
Epoch 1/5
125/125 ━━━━━━━━━━━━━━━━━━━━ 13s 105ms/step - accuracy: 0.3891 - loss: 5.0052 - val_accuracy: 0.3990 - val_loss: 4.9769
Epoch 2/5
125/125 ━━━━━━━━━━━━━━━━━━━━ 19s 91ms/step - accuracy: 0.3957 - loss: 4.9742 - val_accuracy: 0.3645 - val_loss: 4.9852
Epoch 3/5
125/125 ━━━━━━━━━━━━━━━━━━━━ 12s 97ms/step - accuracy: 0.3702 - loss: 5.0895 - val_accuracy: 0.3745 - val_loss: 4.9731
Epoch 4/5
125/125 ━━━━━━━━━━━━━━━━━━━━ 21s 101ms/step - accuracy: 0.3564 - loss: 5.0787 - val_accuracy: 0.3725 - val_loss: 4.9886
Epoch 5/5
125/125 ━━━━━━━━━━━━━━━━━━━━ 20s 100ms/step - accuracy: 0.3821 - loss: 4.9809 - val_accuracy: 0.3825 - val_loss: 4.9773
Retraining Time: 85.35 seconds
Testing Time: 10.83 seconds
Testing Accuracy: 0.3671
```

```
Experiment with 15000 retraining samples:
Epoch 1/5
188/188 ━━━━━━━━━━━━━━━━━━━━ 18s 96ms/step - accuracy: 0.3816 - loss: 4.9526 - val_accuracy: 0.3557 - val_loss: 4.9595
Epoch 2/5
188/188 ━━━━━━━━━━━━━━━━━━━━ 18s 95ms/step - accuracy: 0.3835 - loss: 4.9472 - val_accuracy: 0.3617 - val_loss: 4.9403
Epoch 3/5
188/188 ━━━━━━━━━━━━━━━━━━━━ 22s 103ms/step - accuracy: 0.3857 - loss: 5.0866 - val_accuracy: 0.3493 - val_loss: 4.9785
Epoch 4/5
188/188 ━━━━━━━━━━━━━━━━━━━━ 19s 95ms/step - accuracy: 0.3775 - loss: 5.0326 - val_accuracy: 0.3583 - val_loss: 4.9716
Retraining Time: 76.95 seconds
Testing Time: 10.08 seconds
Testing Accuracy: 0.3688
```

```
Experiment with 20000 retraining samples:
Epoch 1/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 24s 95ms/step - accuracy: 0.3817 - loss: 5.0074 - val_accuracy: 0.3845 - val_loss: 4.9517
Epoch 2/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 41s 97ms/step - accuracy: 0.3797 - loss: 4.9988 - val_accuracy: 0.3845 - val_loss: 4.9312
Epoch 3/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 25s 101ms/step - accuracy: 0.3732 - loss: 5.0151 - val_accuracy: 0.3725 - val_loss: 4.9498
Epoch 4/5
250/250 ━━━━━━━━━━━━━━━━━━━━ 23s 93ms/step - accuracy: 0.3611 - loss: 5.0885 - val_accuracy: 0.3370 - val_loss: 4.9797
Retraining Time: 131.51 seconds
Testing Time: 8.74 seconds
Testing Accuracy: 0.3827
```

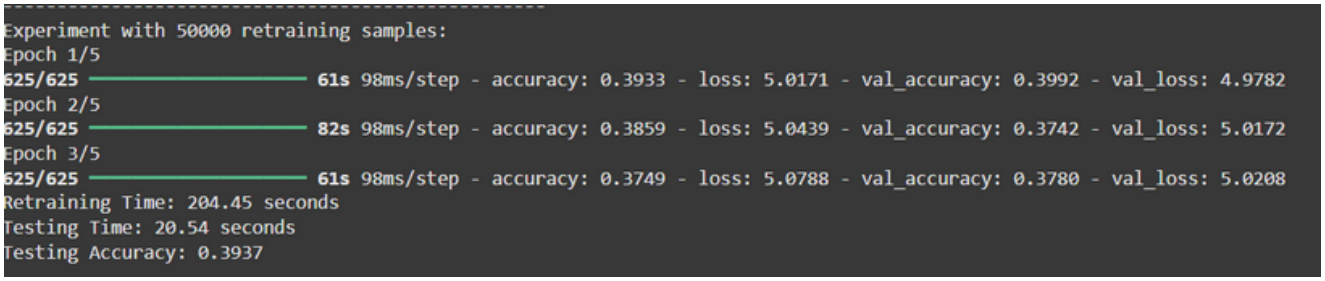
```
Experiment with 25000 retraining samples:
Epoch 1/5
313/313 ━━━━━━━━━━━━━━━━━━━━ 33s 105ms/step - accuracy: 0.3836 - loss: 5.0403 - val_accuracy: 0.3806 - val_loss: 4.9203
Epoch 2/5
313/313 ━━━━━━━━━━━━━━━━━━━━ 33s 106ms/step - accuracy: 0.3821 - loss: 4.9388 - val_accuracy: 0.3684 - val_loss: 4.9769
Epoch 3/5
313/313 ━━━━━━━━━━━━━━━━━━━━ 39s 99ms/step - accuracy: 0.3704 - loss: 5.1276 - val_accuracy: 0.2970 - val_loss: 5.0462
Retraining Time: 114.83 seconds
Testing Time: 9.56 seconds
Testing Accuracy: 0.3765
```

```
Experiment with 30000 retraining samples:
Epoch 1/5
375/375 ━━━━━━━━━━━━━━━━━━━━ 37s 99ms/step - accuracy: 0.3826 - loss: 5.0709 - val_accuracy: 0.3960 - val_loss: 4.9498
Epoch 2/5
375/375 ━━━━━━━━━━━━━━━━━━━━ 42s 101ms/step - accuracy: 0.3754 - loss: 5.0835 - val_accuracy: 0.3607 - val_loss: 4.9909
Epoch 3/5
375/375 ━━━━━━━━━━━━━━━━━━━━ 39s 96ms/step - accuracy: 0.3591 - loss: 5.0262 - val_accuracy: 0.3777 - val_loss: 4.9980
Retraining Time: 118.14 seconds
Testing Time: 10.86 seconds
Testing Accuracy: 0.3894
```

```
Experiment with 35000 retraining samples:
Epoch 1/5
438/438 ━━━━━━━━━━━━━━━━━━━━ 44s 100ms/step - accuracy: 0.3927 - loss: 4.9948 - val_accuracy: 0.3884 - val_loss: 4.9957
Epoch 2/5
438/438 ━━━━━━━━━━━━━━━━━━━━ 44s 100ms/step - accuracy: 0.3767 - loss: 4.9990 - val_accuracy: 0.3849 - val_loss: 5.0163
Epoch 3/5
438/438 ━━━━━━━━━━━━━━━━━━━━ 47s 107ms/step - accuracy: 0.3798 - loss: 5.0505 - val_accuracy: 0.3826 - val_loss: 4.9980
Retraining Time: 169.55 seconds
Testing Time: 10.86 seconds
Testing Accuracy: 0.3844
```

```
Experiment with 40000 retraining samples:
Epoch 1/5
500/500 ━━━━━━━━━━━━━━━━━━━━ 49s 97ms/step - accuracy: 0.3872 - loss: 5.0325 - val_accuracy: 0.3149 - val_loss: 5.0624
Epoch 2/5
500/500 ━━━━━━━━━━━━━━━━━━━━ 82s 98ms/step - accuracy: 0.3527 - loss: 5.1328 - val_accuracy: 0.3618 - val_loss: 5.0039
Epoch 3/5
500/500 ━━━━━━━━━━━━━━━━━━━━ 83s 99ms/step - accuracy: 0.3683 - loss: 5.0319 - val_accuracy: 0.3714 - val_loss: 4.9945
Epoch 4/5
500/500 ━━━━━━━━━━━━━━━━━━━━ 82s 99ms/step - accuracy: 0.3586 - loss: 5.1228 - val_accuracy: 0.3677 - val_loss: 5.0076
Epoch 5/5
500/500 ━━━━━━━━━━━━━━━━━━━━ 86s 107ms/step - accuracy: 0.3584 - loss: 5.0350 - val_accuracy: 0.3579 - val_loss: 5.0104
Retraining Time: 409.62 seconds
Testing Time: 11.19 seconds
Testing Accuracy: 0.3676
```

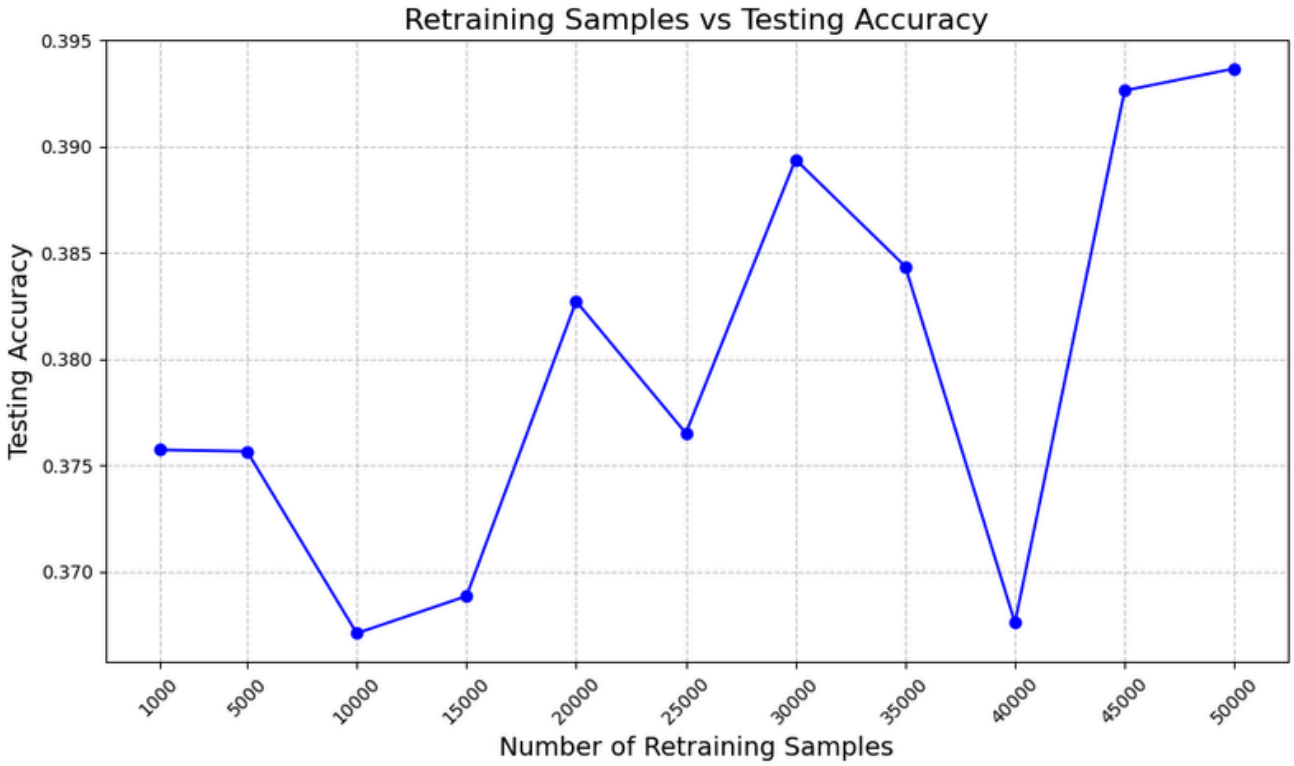
```
Experiment with 45000 retraining samples:
Epoch 1/5
563/563 ━━━━━━━━━━━━━━━━━━━━ 55s 98ms/step - accuracy: 0.3762 - loss: 5.0357 - val_accuracy: 0.3908 - val_loss: 5.0375
Epoch 2/5
563/563 ━━━━━━━━━━━━━━━━━━━━ 90s 111ms/step - accuracy: 0.3867 - loss: 5.0595 - val_accuracy: 0.3788 - val_loss: 5.0438
Epoch 3/5
563/563 ━━━━━━━━━━━━━━━━━━━━ 76s 102ms/step - accuracy: 0.3814 - loss: 5.0541 - val_accuracy: 0.3972 - val_loss: 5.0209
Epoch 4/5
563/563 ━━━━━━━━━━━━━━━━━━━━ 80s 98ms/step - accuracy: 0.3882 - loss: 5.0636 - val_accuracy: 0.3943 - val_loss: 5.0053
Epoch 5/5
563/563 ━━━━━━━━━━━━━━━━━━━━ 85s 103ms/step - accuracy: 0.3921 - loss: 4.9980 - val_accuracy: 0.3804 - val_loss: 5.0261
Retraining Time: 409.63 seconds
Testing Time: 10.92 seconds
Testing Accuracy: 0.3926
```



Retraining the model with samples didnt helped much with the accuracy .There is only a slight change in the accuracy

Retraining Experiment Results:
{'Retraining Samples': 1000, 'Testing Samples': 28031, 'Retraining Time (s)': 4.757209062576294, 'Testing Time (s)': 11.87537431716919, 'Testing Accuracy': 0.3757268786430359}
{'Retraining Samples': 5000, 'Testing Samples': 28031, 'Retraining Time (s)': 34.94026851654053, 'Testing Time (s)': 11.743176698684692, 'Testing Accuracy': 0.37565553188323975}
{'Retraining Samples': 10000, 'Testing Samples': 28031, 'Retraining Time (s)': 85.34972310066223, 'Testing Time (s)': 10.83246898651123, 'Testing Accuracy': 0.367093563079834}
{'Retraining Samples': 15000, 'Testing Samples': 28031, 'Retraining Time (s)': 76.94955086708069, 'Testing Time (s)': 10.082382202148438, 'Testing Accuracy': 0.36884164810180664}
{'Retraining Samples': 20000, 'Testing Samples': 28031, 'Retraining Time (s)': 131.5059471130371, 'Testing Time (s)': 8.740862369537354, 'Testing Accuracy': 0.38271912932395935}
{'Retraining Samples': 25000, 'Testing Samples': 28031, 'Retraining Time (s)': 114.82512307167053, 'Testing Time (s)': 9.55783748626709, 'Testing Accuracy': 0.37651172280311584}
{'Retraining Samples': 30000, 'Testing Samples': 28031, 'Retraining Time (s)': 118.1352002620697, 'Testing Time (s)': 10.857497215270996, 'Testing Accuracy': 0.38939031958580017}
{'Retraining Samples': 35000, 'Testing Samples': 28031, 'Retraining Time (s)': 169.54871273040771, 'Testing Time (s)': 10.858779668807983, 'Testing Accuracy': 0.3843601644039154}
{'Retraining Samples': 40000, 'Testing Samples': 28031, 'Retraining Time (s)': 409.62419271469116, 'Testing Time (s)': 11.186261892318726, 'Testing Accuracy': 0.36759302020072937}
{'Retraining Samples': 45000, 'Testing Samples': 28031, 'Retraining Time (s)': 409.6297392845154, 'Testing Time (s)': 10.91507863998413, 'Testing Accuracy': 0.3926367163658142}
{'Retraining Samples': 50000, 'Testing Samples': 28031, 'Retraining Time (s)': 204.45383596420288, 'Testing Time (s)': 20.53784489631653, 'Testing Accuracy': 0.39367130398750305}

[28]
1s



- **Small Retraining Sample Sizes:**

1. **Advantages: Faster retraining time and lower computational cost.**
2. **Disadvantages: Limited accuracy improvement as the model doesn't have enough data to generalize effectively.**

- **Large Retraining Sample Sizes:**

3. **Advantages: Greater accuracy, as the model can learn from a more diverse and extensive set of examples.**
4. **Disadvantages: Higher computational cost and retraining time. There's also a risk of overfitting if the data is not properly regularized or validated.**