A Comprehensive Benchmarking for Deep Learning Models in WiFi Sensing

Wireless Networks Group - 4

(Shreyanshu(2020130), Dheeraj(2020194), Surya(2020138), Sagar Keim(2019196))

Problem Statement

This project aims to benchmark different deep-learning models for WiFi sensing using significant data sets. The project aims to create and train different deep-learning models. The aim is to get the benchmarking details of the deep learning models run on four significant datasets.

Tools/Technology Used: Python, Deep learning models, Pytorch,CSI Modeling

Deep Learning Models:

The Models to be benchmarked include:

- BiLSTM(done)
- CNN+GRU(done)
- ResNET101(done) + extra(Resnet50)
- MLP(done)
- LeNET(done)

Datasets

1. UT-HAR

UT-HAR is the first public CSI dataset for human activity recognition. It was collected using an Intel 5300 NIC with three pairs of antennae that each record 30 subcarriers and comprises seven categories. The setting in which all the data is gathered is the same. However, it does not have any gold labels for activity categorization, and its data is continuously collected. The data is segmented using a sliding window in accordance with previous works, which necessarily results in a lot of repeated data among samples. Thus, there are roughly 5,000 samples in total.

- Platform: Intel 5300 NIC
- CSI size : 1 x 250 x 90
- number of classes: 7

- classes: lie down, fall, walk, pickup, run, sit down, stand up
- train number: 3977
- test number: 996

2. NTU-HAR

This dataset was collected using a new platform based on the Atheros CSI tool, which enabled CSI extraction of 114 subcarriers of 40Hz directly to IOT devices. This is the significant difference between the UT-HAR and NTU-HAR. It includes the Human Activity Recognition (HAR) tasks. The classes used in the dataset are mentioned below.

- Platform: Atheros CSI tool
- CSI size : 3 x 114 x 500 (antenna, subcarrier, packet)
- number of classes: 6
- classes: box, circle, clean, fall, run, walk
- train number: 936
- test number: 264

3. NTU-HumanID

According to research, each human's gait exhibits distinctive feature patterns in the CSI picked up by wireless equipment. Due to individual variability in physical traits and stride, the WiFi signal reflected by a walking human exhibits unique variation. As a result, the CSI can record the biometric variances in gaits between various users. This dataset was also collected using the same Atheros tool as in the NTU-HAR dataset. However, It includes the Human Identification(HumanID) tasks. The gaits of 14 subjects are present in this dataset

- Platform: Atheros CSI tool
- CSI size : 3 x 114 x 500 (antenna, subcarrier, packet)
- number of classes: 14
- classes: gaits of 14 subjects
- train number: 546
- test number: 294

4. Widar

Widar is the largest WiFi sensing dataset for gesture recognition, comprising 22 categories and 43K samples. It is gathered using a 3x3 antenna pair Intel 5300 NIC in a variety of different environments. The data is processed and transformed to the body-coordinate velocity profile (BVP) to remove environmental dependencies.

- Platform: Intel 5300 NIC
- BVP size : 22 x 20 x 20 (time, x velocity, y velocity)
- number of classes: 22
- classes :
- Push&Pull, Sweep, Clap, Slide, Draw-N(H), Draw-O(H),Draw-Rectangle(H), Draw-Triangle(H), Draw-Zigzag(H), Draw-Zigzag(V), Draw-N(V), Draw-O(V), Draw-1, Draw-2, Draw-3, Draw-4, Draw-5, Draw-6, Draw-7, Draw-8, Draw-9, Draw-10

- train number: 34926
- test number: 8726

Parameters calculated

The following parameters will be calculated for each model on each dataset:

- Training accuracy
- Testing accuracy
- Validation accuracy
- Training time
- Inference time
- Memory requirement for training a particular model
- Number of rows
- Number of features of training and testing data

Expected Deliverables:

We will perform extensive benchmarking on the listed models for the given datasets. We will be evaluating the parameters listed above for each scenario.

Results

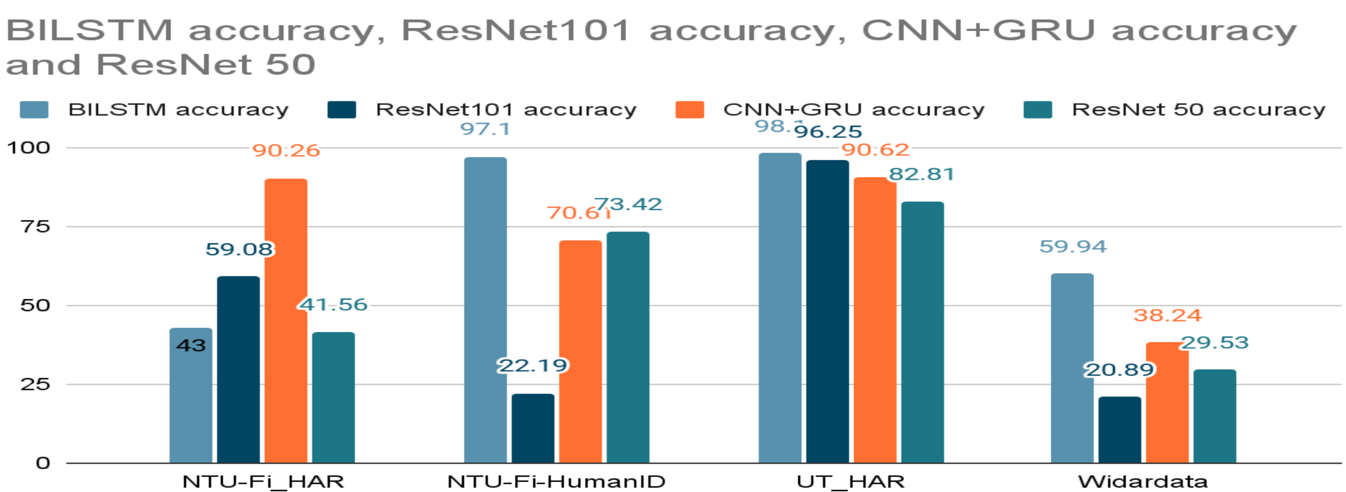

Figure 1.1

Note: as we can see from the plot, the models we have used have been able to successfully attain good performance with respect to the UT-HAR, NTU-FI-HumanID and NTU-FI-HAR. However we need extensive models like the transformer model to assess the performance over the Widar 3.0 dataset, over which the currently used models were not effective enough. The accurate statistical benchmarking figures can be obtained from the output in result.txt file obtained upon running the code in the submission.

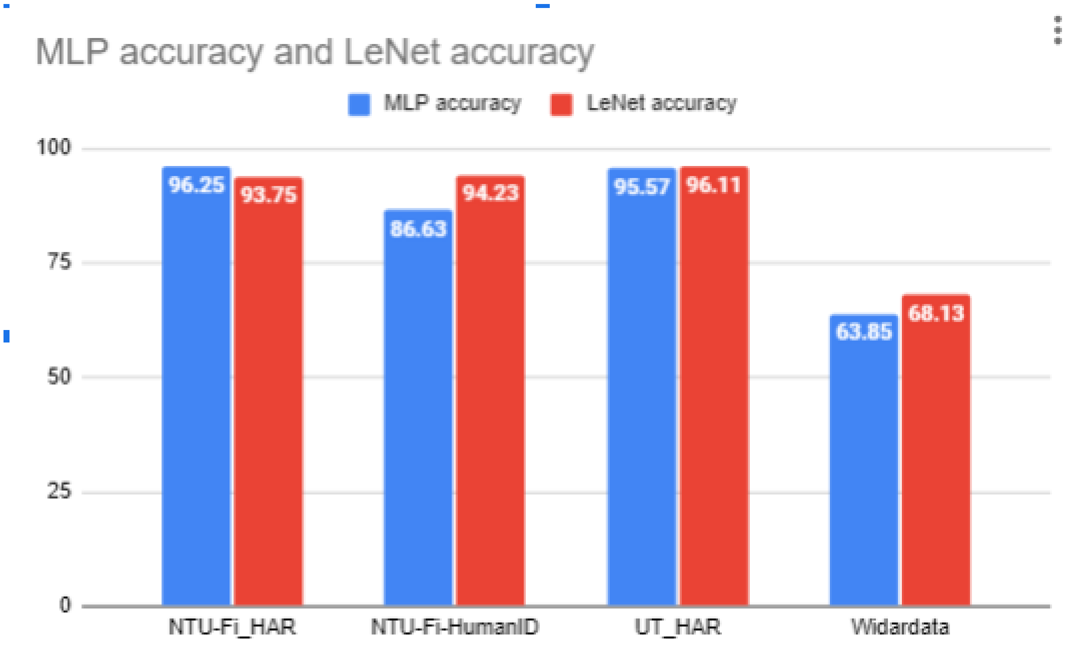

Figure 1.2

Insight

In our comparative analysis, we observed that simple models like LeNet and MLP perform nearly as well as more complex models such as BiLSTM, ResNet101, and CNN+GRU across datasets NTU-Fi_HAR, NTU-Fi-HumanID, and UT_HAR. When evaluating efficiency in terms of inference time and resource utilization (RAM, GPU memory), it becomes evident that the straightforward architectures of LeNet and MLP outperform the others in the mentioned three datasets.

However, for the Widar dataset, our findings indicate that a more powerful model is necessary to achieve higher accuracy. This suggests that dataset-specific characteristics may require a different approach, emphasizing the importance of tailoring model selection to the unique demands of each dataset.

Conclusion

In wrapping up our study, we've learned a lot about how different types of models perform on various datasets. Surprisingly, simple models like LeNet and MLP did really well compared to fancier ones like BiLSTM, ResNet101, and CNN+GRU in most cases. These simpler models also turned out to be more efficient in terms of how quickly they make predictions and how much computer resources they need.

However, there was a twist when we looked at the Widar dataset. Here, we discovered that to get higher accuracy, we need to use more powerful and complex models. So, it seems like the best choice depends on the specific dataset we're working with.

In a nutshell, our benchmark model, with its mix of different kinds of models, can handle a variety of tasks quite well. The key takeaway is that picking the right model depends on the job at hand – sometimes simple is best, but other times, we might need to go for a more powerful approach.

Contribution

- Dheeraj- Benchmark Implementation + ppt
    - Performed the changes to the model code to get benchmarking figures. Comprehended the working of models in order to achieve perfect benchmarking figures
- Sagar Keim- 0%
- Shreyanshu- ppt + report
    - Helped in creation of report, reporting of the results in graph plots, and supported in dataset fitting and researching
- Surya- 0%