Principles and Design of IoT Systems (PDIoT) [INFR11150/INFR11239]

Notes on submission of material for peer review

Please submit one zip file containing the following:

1. **Installation files (folder):**
    a. The .apk file for your Android app, either the debug or release versions. To obtain the apk, in Android Studio select Build > Build Bundles / APK(s) > Build APK(s). The dialogue box will help locate the apk file in your app's *build* folder. You need to ensure that this apk file is installable on a wide range of Android phones.
    b. If applicable, code needed for installing new firmware on the Thingy.
    c. If applicable, code needed for installing software on a laptop.

2. **Installation guide.pdf (file) containing :**
    a. A description of your app's system requirements (such as the minimum and target sdk version) to ensure the reviewers install your app on an appropriate phone.
    b. If applicable, the instructions for installing new firmware on the Thingy.
    c. If applicable, any other instructions for installing software on a laptop (for example if your app requires a nearby backend server).
    d. A list of any non-standard python packages you are using in your code for your reviewers to install when running your evaluation script (described below).

3. **Model description.pdf (file)**:
    a. A description of the capabilities and performance of your classification model (list of activities that your model can successfully classify).
        - For each of these classes provide the accuracy, precision, recall and f-score.
        - Provide the overall accuracy, precision, recall and f-score as well. You should report these accuracies on unseen subjects.
        - You will get the best estimate of your model's performance in real time if you do LOSOXV - that is, test your model on one left-out subject while training it on the rest of the available data, until all the subjects have been part of the test set at least once.
        - For speedup you may also try testing on batches of 2-5 unseen subjects at a time.
        - Your submitted model should be trained on all the available data.

4. **Models (folder):**
    a. Your model file, exported in .tflite.
    b. If you are using multiple models, please export all of them.
    c. If you are using a sklearn model, then package it accordingly.

5. **evaluate_models.py (file):**
    a. A python script that takes the following arguments:
        i. The path to your exported model (if you are using multiple models, the path to a common folder where these models are stored)
        ii. The path to the test data (which will be provided to your peer reviewers). The structure of the test data is exactly the same as https://github.com/specknet/pdiot-data/blob/master/2022/Respeck_recordings_clean.csv and

https://github.com/specknet/pdiot-data/blob/master/2022/Thingy_recordings_clean.csv . You should specify in your instructions which dataset to use or if your peer reviewers should use both.

iii. An example of how to run your script, for example:

```
| $ ./evaluate_models.py --
model_path=./Models/my_model.tflite --
test_data_path=./Data/Respeck_recordings_unseen.csv
```

Your script should apply any preprocessing to the test data, split it into sliding windows and infer its classes. Call the classification_report function of sklearn and display its results. For each of the classes of your model, also print out the accuracy, precision, recall and f-score.

b. Your script should print out:

i. The classification_report for your model(s), e.g:

	precision	recall	f1-score	support
0	1.00	1.00	1.00	2
1	1.00	1.00	1.00	3
accuracy			1.00	5
macro avg	1.00	1.00	1.00	5
weighted avg	1.00	1.00	1.00	5

ii. A list of class names (activity names) and their resulting accuracies, precisions, f-scores, etc. e.g.,

```
Sitting/Standing ……. Accuracy: w, Precision: x,
Recall: y, F-Score: z
Lying down ……. Accuracy: w, Precision: x, Recall:
y, F-score: z
Walking ……. Accuracy: w, Precision: x, Recall: y,
F-score: z
Running ……. Accuracy: w, Precision: x, Recall: y,
F-score: z
```

iii. A confusion matrix, exported as a png file with dpi=300, saved in the same location where the script was called, with the name **confusion_matrix.png**. If you have multiple models working together to produce one classification, please create a single confusion matrix that shows all the activities classified. If you have attempted training multiple disjoint models (e.g. one for a subset of activities and one for all activities) then create a confusion matrix for each. You can name these confusion_matrix_model1.png, confusion_matrix_model2.png etc. **Your labels should be normalised (i.e. between 0 and 1)!**

Your submission folder structure should look like this:

```
.
├── your_group.zip
│   ├── Installation Files
│   │   ├── application.apk
│   │   ├── thingy_files
│   │   ├── additional_files
│   └── Installation Guide.pdf
│   ├── Model Description.pdf
│   ├── Models
│   │   ├── model_first.tflite
│   │   ├── model_second.tflite
│   │   ├── model_sklearn.joblib
│   └── evaluate_models.py
```

Here is an example of how your confusion matrix should look like: