**Benchmarking Machine learning on mobile phone CPU and GPU**

# Project Description:

In this project goal is to develop a mobile app, which can perform different Machine Learning algorithms on a dataset and reports their performance and execution time. A dataset should be chosen from UCI repository ([here](https://archive.ics.uci.edu/ml/datasets.html)) and stored on mobile. Dataset should be partitioned into a training and testing part which error and execution time for each should be reported. For classification, you can either use any available library (for example [weka](http://www.cs.waikato.ac.nz/ml/weka/)) or implement them yourself. The project should have the following functionalities and specifications:

* Four different machine learning algorithms should be implemented chosen from this list (SVM, KNN, NBC, Decision Tree, Logistic Regression, Random Forest). If you have another algorithm in mind before implementation check it with project TA.
* User should be able to decide the partition of dataset to be used for training (it should be between 1% to 99%), the rest will be used for testing.
* User should be able to select which classifiers (at least 1 and at most 4) will execute.
* The app should report the performance (TAR: True Accept Rate, TRR: True Reject Rate, FAR: False Accept Rate, FRR: False Reject Rate and HTER: (FAR+FRR)/2) and execution time for training and testing phase.
* There is a log file that all the reports should also be written into it. For each report there should be a header stating the configuration of the run (e.g. partition rate, timestamp, selected classifiers, etc.). This log should be accessible in app.
* For GPU implementation, you should use OpenCL ([here](https://www.khronos.org/opencl/)) which allows offloading to mobile GPU. The kernel code will be in C/C++ and it will use android NDK ([here](https://developer.android.com/ndk/index.html)) to execute with the app.
* Code should follow the usual software engineering practices; especially it should be modular and organized and avoid hard-codding parameters. In addition, it **MUST** be well documented; meaning each class, method, and variable should have description, and code should have enough comments to be understandable.

# Deliverables:

There will be a demo of the app at the end of class. Besides code and report should be submitted. Report should be between 5-10 pages explaining the code, challenges of development and lessons learnt. Report should also contain a future direction section suggesting improvements and next phases for the app.

# Extra points:

Implementation of multi-threaded version of classifiers on CPU.

**Tasks:**

*Task 1: UI development*

- Create UI​ (dropdown classifier)

- Choose 3 classifiers (SVM,MLP,Random Forest)

- Integrate Weka with Android

- Create interface for uploading training and test data

- Implement cloud interface to upload training and test data and execute the machine learning algorithm in the cloud.

*Task 2:- Random Forest Classifier*

a)Implementing Random Forest Classifier with WEKA on android

b)Train the model

c)Test the model and tune parameters

d)Accuracy estimate and

e) Execution times

*Task 3: SVM Classifier*

a)Implementing SVM Classifier with WEKA on android

b)Train the model

c)Test the model and tune parameters

d)RMSE loss comparisons and

e) Execution times

*Task 4: MLP*

a)Implementing MLP with WEKA on android

b)Train the model

c)Test the model and tune parameters

d)RMSE loss comparisons and

e) Execution times