**XposedModuleApp tutorial**

**Intro**

The XposedModuleApp is Xposed Framework Application, works on Rooted Android phones, suitable for Android 4. Versions (there is a problem to install XposedBridge on Android 5.0 and above).

The application designed to sample selected application and the selected data from specific application, and create pcStream that create models from the data.

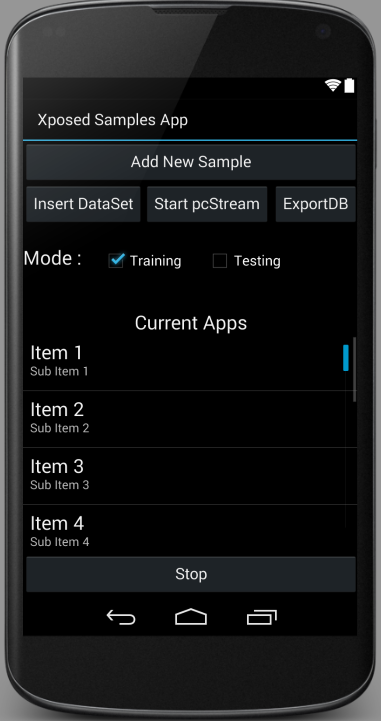
The XposedModuleApp works with Xposed Module that can alert when one of the selected sample App is created.

For working with the app the following package is needed:

* jblas-1.2.3.jar
* jmat\_5.0.jar
* opencsv\_3.5.jar
* simple-xml-2.7.1.jar
* XposedBridgeApi-54.jar

**Class overview**

**MyActivity**



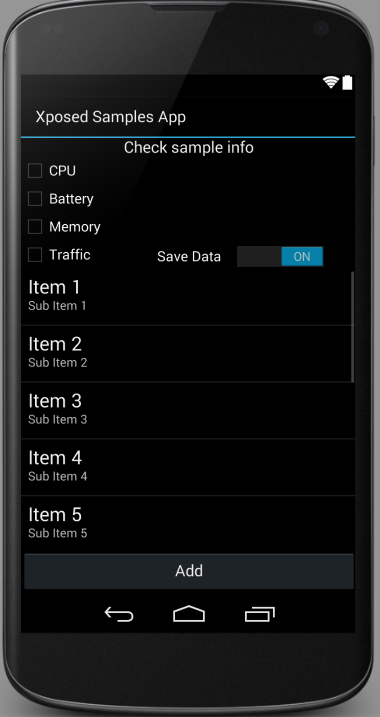
The main UI interface for the user to connect with the XposedModuleApp.

The user can see what apps is currently selected to sample, and can stop them. And can add new App to sample.

There is an option to load data set and run it on the pcStream algorithm.

When getting intent from ActivityTwo process the data and send to AppService the intent with the data.

**ActivityTwo**



UI interface for the user to select what he what to sample and which app to sample.

This Activity return intent with the selected app and the data to sample to onActivityResult function in MyActivity.

**onBoot**

Run on boot startup when the device is started and start the AppSrvice.

**EventsSenderReceiver**

Is a broadcast receiver that listening to the Xposed Module AppListener to notice when selected app to sample is created and send intent to the AppService to start sample.

**AlarmReceiver**

Send intent to AppStream every predetermined time.

**AppService**

Service that manage all samples, in charge of starting new or existing app sample, and send commend to all TakeAppSapmle to take sample every time interval.

Start on boot (need to add when started to check which app running and start sample it).

When getting an intent to onStartCommend function the function check whick kind of commend it is and acting in accordance with the received intent.

All coomend kinds:

1. Sample (take sample from app in the list). The intent have Boolean extra need to be true.
2. StartNewApp / StartExistApp. The intent have String extra with sample information and check if we already sample the app and we StartExistApp and read the pcStream from XML. If we don't have XML for the app we will start new pcStrem.
3. StopSample to remove app from sample list.
4. pcSStop intent that send from the pcStream class when the pcStream algorithm stop and save the object on XML in folder with the name of the app. And remove the app from pcStream list.

* The service have 3 list, 1 to all TakeAppSample, 1 to pcStram, and 1 to the names of all apps.

The service work with alarm manager that initialize the time interval. And sent intent command to sample all apps.

**TakeAppSample**

Class that received application name add what to sample from the application. On create the class create data base with the name of the sampled app. The main function in the class is Run that get cpu and battery stat from AppService. When the function is called it calculate and insert to data base all properties.

The class get pcStream object and send him the data row by row (insert to a queue in psStream).

* Need to add in the catch section in the run function to check if the app is running (because we get error) and if not to send the AppService intent to stop sampling the app.

**pcStream**

Algorithm class that have the pcStream clustering algorithm.

When initialize get the properties that the algorithm will work with (driftThreshold, maxDriftSize, percentVarience, modelMemory, name).

The main function in this class is pcS() that runs all of the time that we sample app.

* There is a semaphore for the queue in the class to prevent errors and to force the thread to sleep when there is no data to calculate.

**AppListener**

This is the Xposed Module is hooking to all onCreate functions on the device.

When onCreate is happened the Module check the called application name and check if it's on the application to sample list, if the application is in the list the module send broadcast to the EventsSenderReceivar and it send intent to the AppService to start sampling the app.

**IStream**

Supposed be the connection between TakeAppSample and pcStream. And contain the queue with the data.

I had concurrency problems with the queue and data was lost, I don't manage to handle it so I remove the queue to the pcStream class and don't have the concurrency problem.

What happened when user start to sample:

