ADLAS

Documentation

Overview

ADLAS is an Autonomous Dynamic Learning Apprentice System, which is designed to assist the user in his daily activities. It observes the user's behaviour, and extracts patterns that the user follows. ADLAS predicts what the user intends to do in the near future, and helps him by autonomously opening applications that he most likely to execute in that particular situation. The system also features functionality such as tracking user's actions. The system can also be overwritten such that given a group of applications if one of them is executed, ADLAS will autonomously execute the other applications.

Requirements

ADLAS has been written in the following languages :

- C++
- Python
- MATLAB
- Powershell

Hence, the compilers for the same are required to be present in the system.

The extra python dependencies to be installed are:

- Tkinter
- Pygubu
- Glob
- Pandas
- Win32com.client
- Subprocess

Operating System

ADLAS has been trained and tested on WIndows 8.1/10.

User Guide

Training Phase

Enter the `Training` folder.

- Run `Testing.cpp`: This will track the user behaviour for a specified amount of time , and generate `out.csv` containing the user activity log for that time session. This time is set to 60 minutes by default, but can be changed by the user.
- Run `RunMatlab.py` to generate `Clusters.txt` which contains the group of applications (clusters) as learned by ADLAS.

Running ADLAS

Enter the `ADLAS` folder.

- The `Clusters.txt` file will be present here as well.
- Run `ADLAS.py`: This open up a GUI with a “start” button. Click on the button to start ADLAS. Once activated, ADLAS assists the user by autonomously executing applications which the user is likely to open.

Algorithm

The training phase contains the following files:

- `Testing.cpp` is the main file which executes the sub-processes, along with the following functionalities :
    - Generating event files to “Event`Files” folder.
    - Creates a `LogFile` of start and end time for each application and saves the activity log to `out.csv`
- `FindNum.py` : identifies the number of `EventFiles` generated.
- `CleanData.py`: cleans the `EventFiles` created with using a pre-defined array of applications (since not all applications are of use).
- `RunMatlab.py`: Runs MATLAB to create the clusters using NCToolBox (Self-organising maps) to `Clusters.txt` and `cluster_img.jpeg` using `get_cluster.m` and `create_text.m`.

The ADLAS folder contains the following files:

- `ADLAS.py` is the main file which has to be executed. It contains a GUI containing a “start” button. When the button is pressed, `main.cpp` is compiled and executed. `gui.ui` has been used to create the GUI.

- `main.cpp` :
    - Parses Clusters.txt to make it usable.
    - Detects what new applications have been opened at this instant and matches with the previous clusters from `Clusters.txt`.
    - If a match is found, it autonomously runs all the applications from that cluster using `RunAnApp.py`.
- `CleanData.py`: cleans the `EventFiles` generated with using a pre-defined array of applications (since not all applications are of use).
- `RunAnApp.py`: Given an application, it runs this application using command line.

Supported Applications:

The following applications are currently supported by ADLAS :

```
"chrome",              "IDMan",               "notepad",
"sublime_text",        "mspaint",             "WINWORD",
"AcroRd32",            "POWERPNT",            "MicrosoftEdge",
"DCPlusPlus",          "SkyDrive",            "StikyNot",
"cmd",                 "WinRAR",              "MusicBee",
"devcpp",              "TeamViewer_Service",  "EXCEL",
"OneDrive",            "wordpad",             "steam",
"powershell",          "wmplayer",            "Photoshop"
"explorer",            "vmware",              "firefox",
"vlc"
```