**Malware Classification System**

**Requirements:**

MCS is developed on windows 7 platform using following python packages:

Numpy, Scipy, SKlearn (latest containing MLP in Neural Network package)

**Description:**

**Malware classification system works depicted in fig. below**

JSON Reports

**Cuckoo Sandbox**

(Execution of program in a controlled environment)

**Feature Extraction Module**

(Duration, Network, API Freq, API seq., OS)

**Feature Selection Module**

(Relevance, redundancy, interaction information using information theory concepts)

**Classification Module**

(MLP-ANN)

**Detection result analysis module**

(FP, TP, Confusion Matrix etc.)

**Training dataset**

(Malware + Cleanware)

**Test dataset**

Analysis Report

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Module | Purpose | Input | Output | Working | Usage |
| trainMLP.py | It trains the MLP classifier based upon  JSON reports of malware | JSON reports | 1. Training and test results based upon splitting of dataset in ratio of 70% for training and 30% testing 2. A trained model of MLP for future predictions of malware | 1. extract\_features.py   Extracts features from JSON files to raw\_malware\_dataset\_repository.csv   1. select\_features.py   Selects relevant features using raw\_malware\_dataset\_repository to a file Selected\_features\_indexes.csv   1. Generates a new dataset file to Selected\_Features\_dataset.csv based from raw dataset based upon selected features 2. MLP\_classify.py   Trains a MLP classifier based upon dataset having selected features (Selected\_Features\_dataset.csv)   1. Provide classification results for training and testing 2. Save trained model of classifier to disk for future predictions with file name my\_dumped\_MLPclassifier.pkl | 1. Place all JSON reports for training of the classifier in JSON reports folder (number should be large for better results) 2. Run command   python trainmlp.py (for screen output)  Or  python trainmlp.py>>output.txt   1. Keep JSON reports, JSON\_processed, JSON\_not\_processed folder for every run of MLP classifier. |
| testMLP.py | Here, trained MLP classifier predicts family of malware | JSON reports | 1. Malware family label of test sample | 1. extract\_features.py   Extracts features from JSON files to raw\_malware\_dataset\_repository.csv   1. read selected features indexes from Selected\_features\_indexes.csv 2. Generates a new dataset file to Selected\_Features\_dataset.csv based from raw dataset based upon selected features 3. Load trained model of MLP from my\_dumped\_MLPclassifier.pkl 4. Predict malware family label using a trained MLP classifier based on selected features | 1. Place all JSON reports of test samples in JSON reports folder 2. Copy file Selected\_features\_indexes.csv to current folder 3. Copy file my\_dumped\_MLPclassifier.pkl   To the current folder   1. Run command   python testmlp.py (for screen output)  Or  python testmlp.py>>output.txt   1. Keep JSON reports, JSON\_processed, JSON\_not\_processed folder for every run of MLP classifier. |