**Malware analysis using GPGPU**

**Dataset**

For this experiment we downloaded 11355 malware samples from malacia-project and collected 2967 benign programs (also verified from virustotal .com) from different systems. In the collected dataset it was found that majority of malwares are below 500 KB hence for our study we focused on malware and benign files below 500 KB. After applying the size limit of 500 KB on samples we are left with 2363 benign samples and 11305 malware samples for our work.

**Preprocessing**

The malware and benign samples were processed using objdump utility to get opcodes. A unique opcode list was prepared from the processed samples which was then used to make the feature matrix for each malware or benign sample. For each sample the feature matrix consists of the frequency of a particular opcode in a malware and benign sample. The data was normalized by dividing each malware and benign opcode frequency by corresponding maximum opcode frequency .

**Feature Selection**

To find distinctive features in a group we first divide the normalized frequencies of malware and benign per group by the total number of malwares and benign in that group and then perform column wise sum for that group where each column denotes an opcode. By performing the above procedure, we get two different vectors corresponding to malware and benign. Now we find absolute difference between the malware and benign vector where each column entry corresponds to an opcode. The absolute difference is then sorted in descending order. Finally, top K features (opcodes) are chosen based on the top K absolute difference. The whole process is done to find those opcodes which are able to separate malware and benign clearly in group. This process is then repeated for each group.

It was observed that some of the groups were not having sufficient malware or benign samples for training and testing so we neglected them from our course of study (group 5,8,61,65,97 contain less than 6 samples in either malware or benign classes while group 98 and 100 has 0 malware samples). However, with more malware and benign samples in hand it they can be included. The preprocessed data is the divided into testing and training sets. We used 67% data for training and reaming 33% data for testing.

**Experimental Setup**

The study is performed using two different hardware configurations. The preliminary tests were performed on . The second experiment was performed on a system with Intel i7-7700HQ Quad core processor with base frequency of 2.6Ghz, 8 GB RAM, Pascal architecture (GP107) based Nvidia 1050Ti GPU with 768 CUDA cores distributed across 6 SMP and 4GB GPU DRAM operating on a base speed of 1291 Mhz.

Note:

I rechecked regarding the way we are testing malware or benign samples in a group and found that we followed the same technique as discussed by you (checking the malware samples inside the same group). Although we tried for random samples also in between but the results given to you are according to the original approach. To implement this we are maintaining training and testing list in each group that contain group ids.

Another observation during multiple runs performed by me, I found that the best point for trade-off between accuracy and execution time can be found by setting the no of features to 250. In previous work done by Ashu sir, he has mentioned about the best case being 87.02% which is being gained here also. There is no mention of lowest or average accuracy.

I have something to check regarding the no of files being used for testing so I can not make a clean statement regarding exact accuracy comparison between this work and previous work but will update you after I am done with it.