According to the SMART flowchart (Figure 3 of paper), input is an access log file which should be prepared for first step (Session identification). The following explanations cover the pre processing steps implemented to convert the input access log files in a proper format.

First, we execute a MATLAB script named ‘ImportLog.m’ to read the data set (text file) and specify the necessary fields. The data sets used in this project have the Common Log Format (CLF) which looks something like this:

127.0.0.1 - frank [10/Oct/2000:13:55:36 -0700] "GET /apache\_pb.gif HTTP/1.0" 200 2326

The lines below show the related codes of this script to specify the nine necessary field:

Space=find(line==' ');  
 IP{i}=line(1:Space(1)-1);  
 DateTime{i}=line(Space(3)+2:Space(4)-1);  
 HttpMethod{i}=line(Space(5)+2:Space(6)-1);  
 File{i}=line(Space(6)+1:Space(7)-1);  
 ErrorCode{i}=line(Space(8)+1:Space(9)-1);  
 DataVolume{i}=line(Space(9)+1:Space(10)-1);  
 Referrer{i}=line(Space(10)+2:Space(11)-2);  
 UserAgent{i}=line(Space(11)+1:end);

The output is a .mat file in the same path of the input. It is worth mentioning that a file with a .mat extension contains MATLAB formatted data (the fields indicated above) and this data can be loaded from or written to this file by using the functions LOAD and SAVE, respectively. Also, the name of output file is similar to the name of input file plus the postfix ‘Converted.mat‘.

In the next step, we execute a script called ‘Cleaning.m’. This script asks the user for the input file which have been produced in the previous step (.Converted’ file). The aim of this step is to:

1. Ascendingly order all requests based on their time.  
2. Remove requests which have all identical fields (in some data sets, there are some completely similar requests which seem to have been duplicated and carry no useful information.)

3. Change all data to lower case for improving the performance of  
later analysis.

4. Convert ‘NaN’ to the value 0 for some null fields.

Please notice that the following .m files are the functions used to identify the duplicated IPs, User-Agent strings, Error Code, Referrer, and Data transferred fields:

CheckDuplicatesUserAgent.m

CheckDuplicatesReferrer.m

CheckDuplicatesIP.m

CheckDuplicatesErrorCode.m

CheckDuplicatesDataVolume.m

The output of Cleaning.m is a .mat file with similar name of the input plus a postfix 'AndCleaned.mat'.

After this step, the data is ready for the Session identification. A function called SessionIdentifier (SessionIdentifier.m) tries to identify all the sessions according to the definitions explained in the paper (page. 132). Please note that a MATLAB script named ‘FeatureExtraction.m’ calls the SessionIdentifier function to first identify the sessions and then extract features (Table. 1 of the paper) for each session. Similarly, it asks the user to specify the input file (produced in the previous step with postfix ‘Cleaned’) and creates an output as an .mat file with ‘Sessions’ postfix in its name. For more information about the features extracted in this step, please refer to both Table 1 of the paper and the comments provided in FeatureExtraction.m.

According to the SMART flowchart, after the pre-processing step, the features and sessions are ready to be used for feature selection, and clustering. To do so, we implement a script named SMART.m which contains the related codes respectively. Similarly, the following explanations are based on the flowchart of SMART in the paper.

First to generate the packages, SMART calls a function called