# **PROJECT REPORT**

This project is basically a practical exhibition in which, by using 54 user’s internet usage data of two weeks and finding whether each user is distinguishable or indistinguishable with that of the all other’s and how the size of the window used affects the result of being distinguishable or indistinguishable.

In this project, I used two weeks of data with the exceptions of only considering data between 8 am and 5 pm every day and leaving out Saturday and Sunday of every week completely.

For this project I used python for the reasons of being easy working with huge datasets and wide modules and libraries availability. The input of this project is the 54 user’s internet usage data stored in excel sheets. We compared three windows of different size: 10 seconds, 227 seconds and 300 seconds (5 minutes).

The results from the program are saved into three respective sheets named after the window size.

RESULTS and OBSERVATIONS:

The results are the P-values that are calculated from the internet usage data of two users. If the P-value happens to be less than or equal to 0.05 (P≤0.05) then the users are said to have distinct internet usage. Else if the P-value is greater than 0.05 (P>0.05), then the users are said to have indistinct internet usage.

I took all the possible combinations between every user i.e.; in total of 2916 combinations for every window size considered. This is because, we compare the first user’s both weeks of data but only second week data of the second user’s data. This may show the difference in the P-value when user A and user B are compared, with the P-value of user B and user A compared.

|  |  |  |  |
| --- | --- | --- | --- |
| **NUMBER OF COMBINATIONS/**  **WINDOW SIZE** | **10 SECOND**  **WINDOW** | **227 SECOND**  **WINDOW** | **300 SECOND**  **WINDOW** |
| Indistinguishable Combinations | 1772 | 1847 | 1890 |
|  |  |  |  |
| Distinguishable Combinations | 1144 | 1069 | 1026 |

Ideally speaking, a user’s data must be distinct with any other user’s data, but it also depends on the time window on which their usage comparison is based on. Because when the window size is as small as 10 Seconds is considered and there is a much higher chance of the users being distinct. But when the window is 300 seconds, we see an increase in number of combinations that are indistinguishable. This may be a concern when the data is used for much important purposes like in some fields of Cyber Security and fields where the authenticity of person is what matters most and given preference to.

But this comparison based on higher window size is also useful in some fields of Security where Anonymity is transparent or not much appreciated or in cases like nabbing someone who has/had access to something to which they are not supposed to have, through the patterns in internet usage.

EXPLAINATION OF CODE:

I used python3 to code because of fact that python wide availability of modules and libraries, also for its simplicity of usage. I used modules like SciPy, NumPy for calculations and data frames, pandas and openpyxl for extracting and inserting the data from and into the excel files.

I have divided the code into two parts. The first part is for pre-processing the data in the data files. The second part is the part where the processed data is used to find the P-values.

Pre-Processing:

A screenshot of a cell phone

Description generated with very high confidence

In this part of the code; all the excel files are taken one after another, and the data from the excel is used to extract the data just what we need. The week data is created based on the timeslot window. This extracted data is then saved into the same file as a different sheet.

In the createWeek function, all the entries in the excel files are considered. If it meets all the conditions, then the doctets/Duration value is calculated, and the doctets/Duration value is dropped into the excel as Sheet2 for week1 data and Sheet3 for week2 data.

A screenshot of a cell phone

Description generated with very high confidence

All the week data is saved as a data frame. This data frame is passed into a function called findD.

In this findD function, the week data is used, and the data is split for each day as five days. This day data is used to find the average doctets/Duration value in a window from 8am to 5pm.

This is done by using another function avg which takes the day data and timeslot. I created a list that ranges from 8 am (considered as 28800 seconds) t0 5 pm (considered as 61200 seconds). The difference between every element in this list will be the timeslot.

The total number of entries that fall in a single gap between two elements are averaged and the values is saved. This is done for every day for a week1.

A screenshot of a cell phone

Description generated with very high confidence

Finally, all this week data is stored in the excel file as a Sheet2 for week 1 data and as Sheet3 for week 2 data.

This is done for all the excel files in the folder and all their week data is stored in the same respective excel files.

Calculations:

In this part of the code; I took every possible combination between the 54 users and calculated the P-values. The total number of combinations are 2916.

This loop calculates the P-value for every combination of a given timeslot and stores all the P-values in an excel with rows and columns named after the file names of the excel sheets.

The first step is to pick a combination and from which the first and second users are selected. The next step is to extract the week data from the processed data that is stored in excel sheets of both the selected users.

This data is then passed to the function spearmans\_rank\_correlation.

A screenshot of a cell phone

Description generated with very high confidence

In the spearmans\_rank\_correlation, two lists of doctets/Duration values for any week of users is sent. The lists are then sorted, ranked and then the standard deviation is calculated. Eventually, the spearman\_rank\_correlation value is returned.

A close up of a screen

Description generated with very high confidence

The ranks are then used to find the the Z value.

I created a function findZ, which takes the ranks and returns the Z value.

A close up of a screen

Description generated with very high confidence

A screenshot of a computer

Description generated with very high confidence

Then the Z value is used to find the P value.

Then the final P value is stored in a data frame.

When all the P-values are found for the given timeslot, and the data frame is filled with these values, the data frame is saved as a xlsx file.

SHRAVAN KUMAR KARNATI