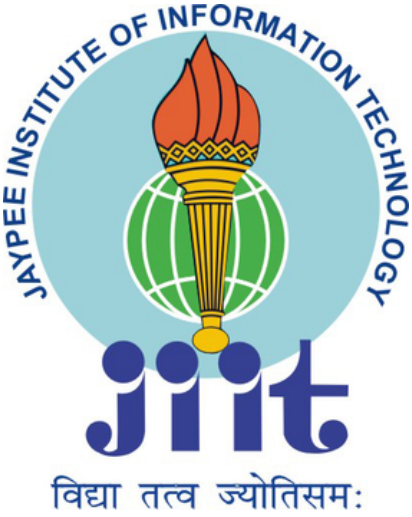
JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY



BACHELOR OF TECHNOLOGY,5th SEMESTER

Minor Evaluation Report

Detection of Geographical Hotspots and its Applications

Submitted By:

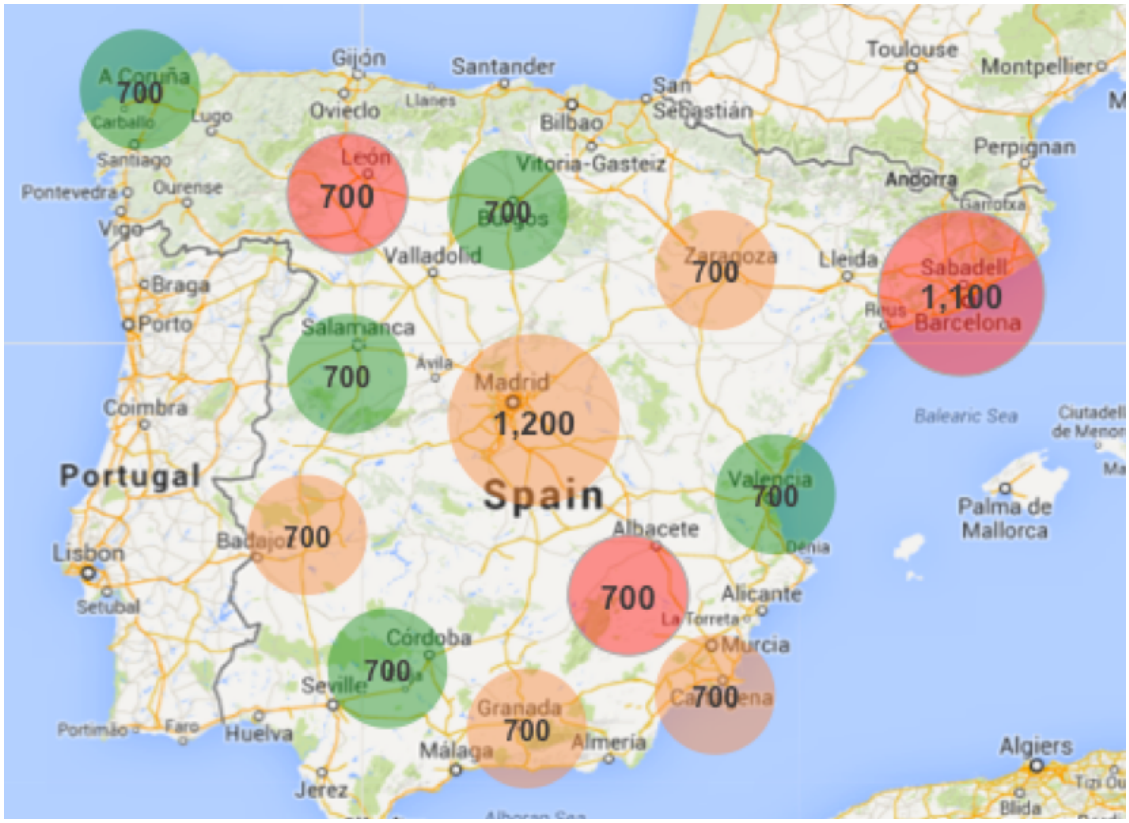
1. Akash Kumar Rai (17103177)
2. Akash Sharma

Supervised By:

Manish Kumar Thakur

WHY TO DETECT GEOGRAPHICAL HOTSPOTS:-

ARE THEY EVEN USEFUL?



Geographical location has played a key role in many scenarios. Hotspot detection is a known spatial clustering process in which it is necessary to detect spatial areas on which specific events thicken ; the patterns are the events georeferenced as points on the map; the features are the geographical coordinates (latitude and longitude) of any event. Hotspot detection is used in many disciplines, as in crime analysis, for analyzing where crimes occur with a certain frequency, in fire analysis for studying the phenomenon of forest fires, and in disease analysis for studying the localization and the focuses of diseases. It could help us to find factors that could link to the occurence of a particular instance. It could be also used to optimize our service and create robust logistical services and efficient planning. Generally speaking, for detecting more accurately the geometrical shapes of hotspot areas algorithms based on density are used and they measure the spatial distribution of patterns on the area of study, but these algorithms have a high computational complexity.

However, the task of detecting the circles of interests is quite challenging from computational point of view. In this report we plan to explain a method that can be used to decrease the computational load and scale the Hotspot detection algorithm. This algorithm has other advantages such as eliminating very small and insignificant hotspots, being robust to the geographical conditions, computationally scalabe etc.

Research Paper Used:

Geographically Robust Hotspot Detection:

By: Emre Eftelioglu, Xun Tang, Shashi Shekhar

https://ieeexplore.ieee.org/document/7395840

Abstract:

Geographically Robust Hotspot Detection (GRHD) discovers hotspot regions where the concentration of points inside is considerably large, given a set of points in two-dimensional space, a minimum radius, a minimum log probability ratio and a meaning threshold. For many applications including environmental criminology, epidemiology, etc., the GRHD issue is societally crucial. Due to the difficulty of listing all possible applicant hotspots and the absence of monotonicity property for the interest measure, namely the log probability ratio test, GRHD is computationally difficult.
Related work may be missing hotspots when hotspots are separated by geographic obstacles (road network, rivers, etc.) or hotspots are near to parks, lakes, hills, etc. A novel method is suggested based on two concepts to solve these constraints: a cubic grid circle enumeration and an upper bound grid log probability ratio.
A case study on actual crime data shows that the proposed approach identifies hotspots that the related work can not discover. Experimental findings indicate that the suggested algorithm produces significant computational savings compared to the job involved.

Geographical Hotspot:

Geographically Robust Hotspot Detection (GRHD) discovers hotspot regions where the concentration of points inside is considerably greater than the amount of points outside, given a set of geolocated points (e.g. disease / crime sites), a minimum radius, a minimum likelihood ratio and a meaning threshold. This document relates to geographically robustness as the capacity to be insensitive to minor gaps in the contiguity of the spatial pattern (e.g. adjacent hotspots separated by street sections, rivers, etc.).

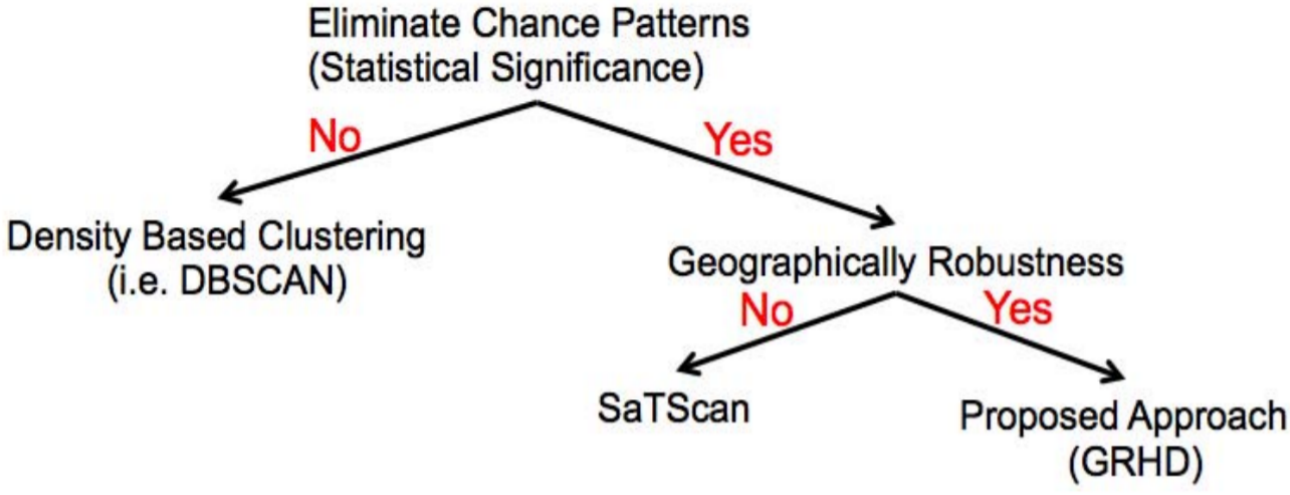
Applications of Hotspots:

Our main aim is to implement an efficient algorithm to evaluate a hotspot in a dataset and use it for various purposes. Some could be listed below:

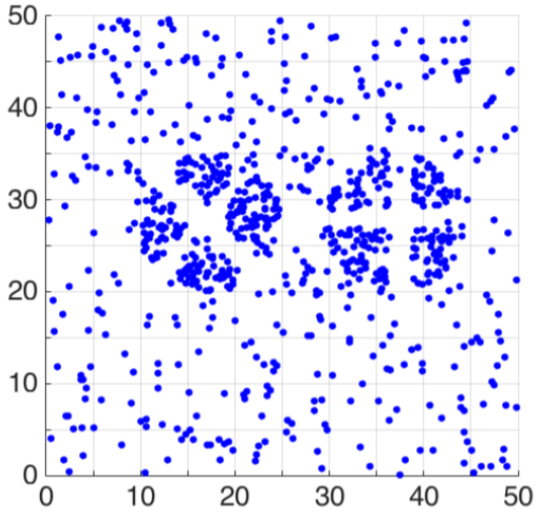
- Hotspot detection can be used to optimize services related to geographical evaluation of distances and services. A taxi service is a good example. A taxi driver can use the information provided to find efficient stoppage points and routes. Same could be done by food delivery services.
- Finding a correlation between various results and factors to establish various predicted results. For eg. a correlation between crime and various other factors such as average income of a household, presence of street lights etc.

There are three important concepts related to the detection of hotspots in these domains:

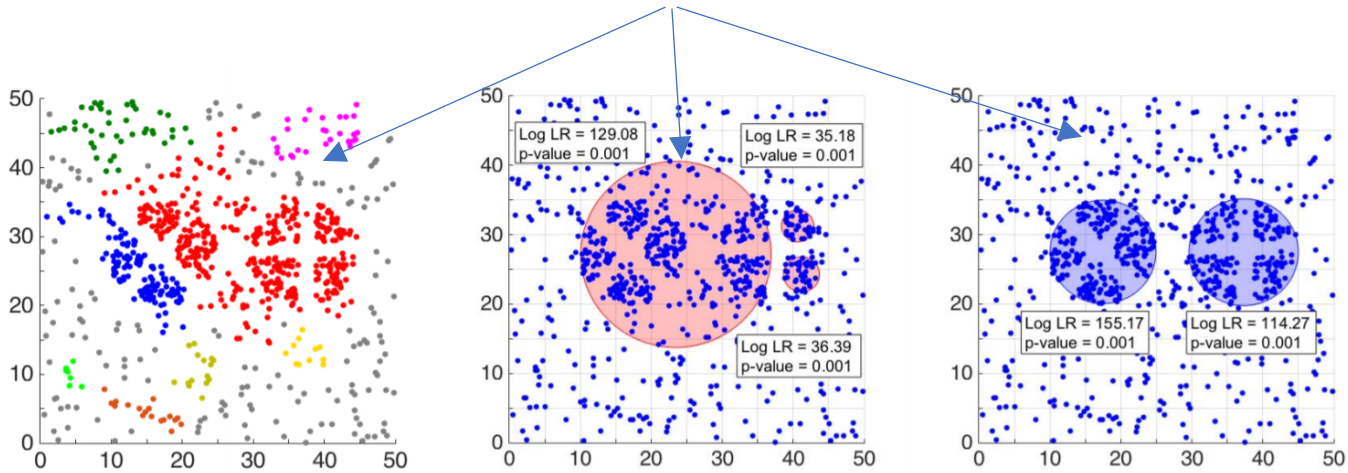
1. Elimination of chance hotspots (prevent false positives).
2. Ability to detect a hotspot if it exists (geographically robustness).
3. Modeling hotspots with respect to diffusion model.



Different types of Hotspot Detection Algorithm



Input



DBSCAN SatScan GHRD(CGC)

We are using GHRD model for the hotspot Detection implemented using CGC (Cubic Grid Circle) Algorithm.

Cubic grid circle Algorithm :

Algorithm 2 Cubic Grid Circle Algorithm

Input:
1) An point set P with points $p(x, y)$,
2) A minimum circle radius r_{min} 3) A log likelihood ratio threshold (θ),
4) A $p\text{-}value$ threshold α_p and 5) A number of trials in Monte Carlo Simulation (m)

Output:
Circles C with $r \geq r_{min}$, $Log\ LR_C \geq \theta$ and $p\text{-}value \leq \alpha_p$

Algorithm:
1: **Filter Phase:** $filteredSet \leftarrow$ all $p \in MECC$ for a $cell_{circle}$ with $Log\ \widehat{LR}_{grid} \geq \theta$
2: **Refine Phase:** $candidateCircles \leftarrow C$ with $Log\ LR_C \geq \theta$.
3: **Monte Carlo Simulation Phase:** $resultSet \leftarrow candidateCircles$ with $p\text{-}value \leq \alpha_p$
4: return $resultSet$

In this paper, a cubic grid circle algorithm (CGC) is proposed in order to address the following issues: (1) eliminate chance patterns, (2) detect non-contiguous (divided by rivers,road segments, etc.) or sparse center hotspots, (3) eliminate very small hotspots, (4) improve the scalability that is affected by the cardinality of *P*. CGC consists of three phases. Filter phase enumerates hotspots in a parametric space, filters those which do not survive an upper bound on likelihood ratio (*Log LRgrid*) and return *filteredSets* ∈ *P*. Refine phase enumerates actual hotspots using *filteredSets* and returns a hotspot with the highest *Log LRC* for each *filteredSet*. Finally, Monte Carlo simulation phase assesses the statistical significance of the enumerated hotspots.

Problem Formulation

The Geographically Robust Hotspot Detection (GRHD) problem is formulated as follows:
Given:
1) A set of points *P* where each *p* ∈ *P* has *x* and *y* coordinate in a two dimensional Euclidean space,
2) A minimum radius *rmin*
3) A log likelihood ratio threshold (*θ*),
4) A *p-value* threshold(*ap*) and a number of Monte Carlo simulation trials (*m*)
Find: Circular Hotspots (*C*(*x, y, r*)) in the study area *S* with *Log LRC* ≥ *θ* and *p-value* ≤ *ap*.
Objective: Computational efficiency and scalability

Constraints:

1) Correctness of the result set,
2) Detected circular hotspots do not overlap

In other words, Given a set of activity points in a geographical space evaluate non overlapping

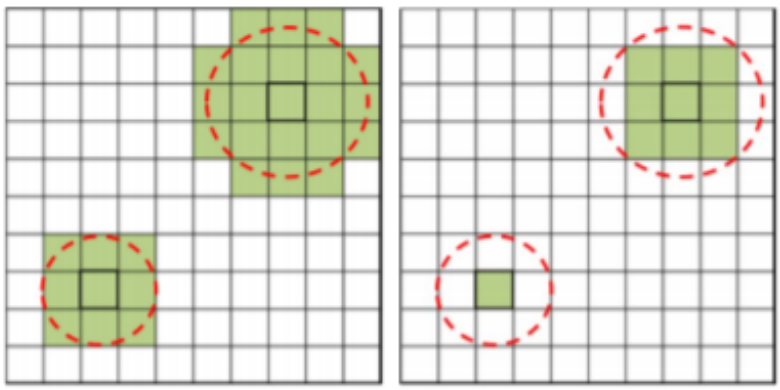
circular hotspot that have a significantly high concentration of activity points and satisfying the constraints stated above. Then we can use the evaluated hotspots for various purposes as listed above.

Definitions

A **count grid** with cell length *lcell* is a partitioning of the study area *S* into a 2-dimensional grid where each cell is a square with an area of *lcell* × *lcell*. The number of count grid cells is denoted by *N* ×*N*, where *N* = *side length*(*S*)/*lcell*. Each count grid cell (*cellcount*) is defined by its coordinate intervals ([*xmin, xmax*], [*ymin, ymax*]) and the count (*ccell*) of the points inside.

A **Minimum enclosing cell collection (MECC)** is a collection of *cellcount* which encloses a set of circles with radius $r \leq rMECC$ where *rMECC* is the radius of the MECC.

A **Maximum fit cell collection (MFCC)** is a collection of *cellcount* which can fit completely inside a circle with radius *r* where $rMFCC = r\sqrt{2}Lcell$ and thus $rMFCC \leq r$.



(a) MECC (b) MFCC

Log Likelihood Ratio ($Log\ LR_C$) is the interest measure that is used as the test statistic for a candidate circle C [15], [18], [19]. The equation can be shown as;

$$Log\ LR_C = Log\left(\left(\frac{c}{B}\right)^c \times \left(\frac{|P|-c}{|P|-B}\right)^{|P|-c} \times I()\right) \quad (1)$$

$$B = \frac{|P| \times area(C)}{area(S)} \quad and\ I() = \begin{cases} 1, & \text{if } c > B \\ 0, & \text{otherwise,} \end{cases}$$

B is the expected and c is the observed number of points in a particular area, $|P|$ is the cardinality of P and $I()$ is an indicator function. $I() = 1$ when the candidate hotspot has more points than expected ($c > B$) and $I() = 0$ otherwise [11].

Algorithm consist of three phases which are explained below:

- ### Filter Phase

First, *S* is discretized into count grid and cubic circle grid using *lcell* = *rmin* 2 (lines 1-2). The cell length *lcell* is selected by the given input *rmin*, because in order to detect all the hotspots with $r \geq rmin$ at least one cell should fit completely inside the hotspot. Next, the number of expected points inside a single cell is determined by *Bcell* ← |*P*|/(*N*×*N*) (line 3). *Bcell* is used when

computing the *U*(*B*) and *L*(*B*), since MECC and MFCC are constituted of cells. For every *cellcircle*, *Log LRgrid* are computed (lines 5-7). To prevent the overlapping *filteredSet*s, once *Log LRgrid*s are computed, the *cellcircle* with the highest *Log LRgrid* is stored in *celltop circle* if *Log LRgrid top* ≥ *θ* (line 8) and points, which are associated with *celltopcircle*, are stored as a *filteredSet* (line 9-10) and are removed from *P*. This process is repeated until none of *Log LRcell* ≥ *θ* or *P* = ∅ (lines 4-11). Finally, *filteredSet*s and *cellcount*s are sent to the refine phase.

Algorithm 3 Filter Phase Algorithm of CGC

Input:
1) A log likelihood ratio threshold (θ)
2) $l_{cell} = \frac{r_{min}}{2}$ for the cell size of count grid and circle grid

Output:
All $filteredSet$s for each non-overlapping $cell_{circle}$ with $Log\ \widehat{LR_{grid}} \geq \theta$

Algorithm:
1: Create count grid with $N \times N$ cells using l_{cell}
2: Create cubic circle grid with $N \times N \times N$ cells
3: Compute the expectation for a single $cell_{count}$ by $B_{cell} \leftarrow |P|/(N \times N)$
4: **while** $P \neq \emptyset$ || $Log\ \widehat{LR_{grid}}^{top} \geq \theta$ **do**
5: **for each** $cell_{count}$ in count grid **do**
6: **for each** $r \leftarrow 1$ *to* $N/2$ **do**
7: $cell_{count} \leftarrow Log\ \widehat{LR_{grid}}$
8: $cell_{count}^{top} \leftarrow cell_{count}$ with the highest $Log\ \widehat{LR_{grid}}$
9: **if** $Log\ \widehat{LR_{grid}}^{top} \geq \theta$ **then**
10: $filteredSet_{1...k} \leftarrow$ *all* $p \in MECC$ of the $cell_{count}^{top}$
11: $P \leftarrow [P - filteredSet]$
12: return $cell_{count}^{1...k}$ and $filteredSet_{1...k}$

- ## Refine Phase

The pseudocode of the refine phase in algorithm 4 starts by ordering the points in the *filteredSet* by their distance to the center of the *cellcount* (line 2). Next, the minimum enclosing circle (*C*) of the *filteredSet* is determined [22](line 4). If the center of *C* is inside the center interval of *cellcount*, the *Log LRC* is computed using the count of the points inside (which is equal to the cardinality of *filteredSet*) and the area of *C* (line 5-7). If the *Log LRC* is greater than the one computed previously as *Log LRpreviousC* , it is saved (line 8-9). Next, the point farthest from the center of the *cellcount* is removed from the *filteredSet* (line 10) and the algorithm repeats the process of finding a new minimum enclosing circle for the new [*filteredSet* − *pfarthest*] until |*filteredSet*| = ∅ or the radius of the generated *C* is *r* ≤ *rmin* (line 3-10). Finally, the *C* with the highest *Log LRC* is saved in *candidateCircles* (line 11). This process is done for all *filteredSet*s returned by the filter phase (line 1-12).

Algorithm 4 Refine Algorithm of CGC

Input:
1) A log likelihood ratio threshold (θ)
2) $filteredSet_{1...k}$ and $cell^{1...k}_{count}$ returned by the filter phase

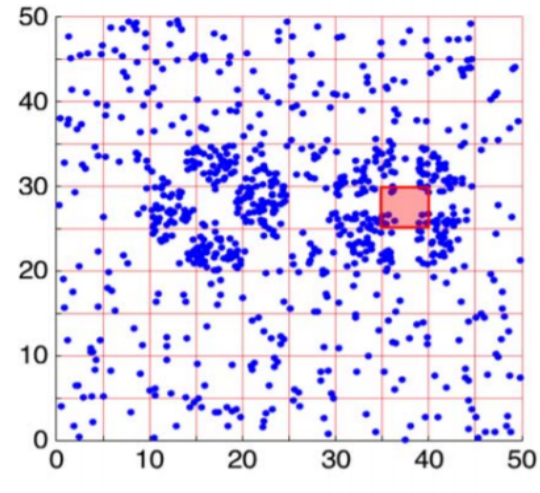
Output:
A circular hotspot C for each $filteredSet$ if $Log\ LR_C \geq \theta$

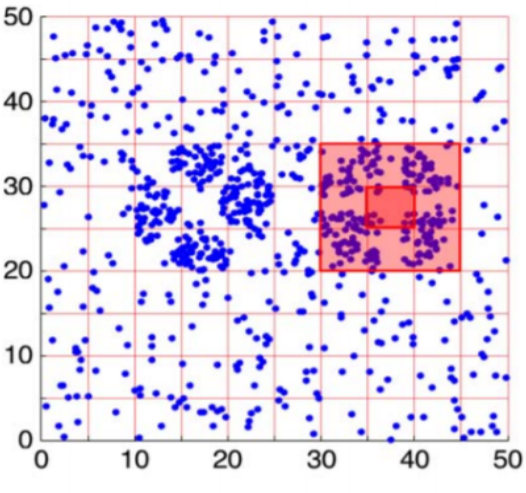
Algorithm:

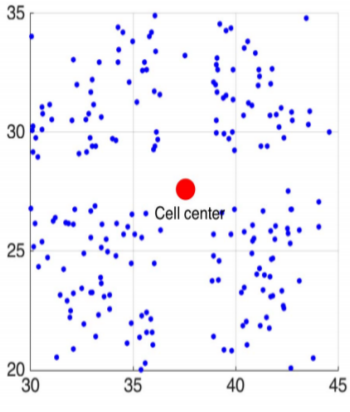
```
1: for each filteredSet_i ∈ filteredSet_{1...k} do
2:   order p ∈ filteredSet_i by distance to the center of cell^i_count
3:   while filteredSet_i ≠ ∅ or r_i ≥ r_min do
4:     C_i ← SEC(filteredSet_i, tmpSet) where tmpSet = ∅
5:     if center^i_C is inside the center interval of cell^i_count
       and r_i ≥ r_min then
6:       compute B and c
7:       Log LR^i_C ← Log Likelihood Ratio
8:       if Log LR^i_C ≥ Log LR^previous_C then
9:         Log LR^previous_C = Log LR^i_C and C_previous = C_i
10:    filteredSet_i = filteredSet_i − p_farthest
11:  candidateCircles ← C_previous and Log LR_previous
12: return candidateCircles

13: procedure SEC(filteredSet, tmpSet) [22]
14:   if filteredSet == ∅ || |tmpSet| == 3 then
15:     C ← computeCircle(tmpSet)
16:   else
17:     select a random p ∈ filteredSet
18:     C ← SEC(filteredSet − p, tmpSet)
19:     if p is not in C then
20:       C ← SEC(filteredSet − p, tmpSet ∪ p)
21: end procedure
```

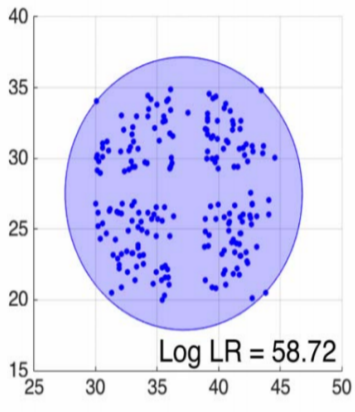
- ## Monte Carlo Simulation Phase

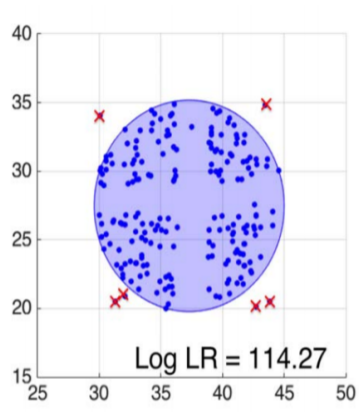
For the circles *C* ∈ *candidateCircles*, a *p-value* is computed by Monte Carlo simulation. First, *m* random datasets with Poisson distribution are generated. For each random dataset, new circles are enumerated and the maximum *Log LRC* of each random dataset is stored in *Log LRMCS C* in decreasing order. To find the significance of a *C*, the position of the *Log LRC* associated with *C* is determined within the *Log LRMCSC* list. This position is divided by (*m* +1) to determine the *p-value* (line 9-12). Finally, all non-overlapping circles with
p-value ≤ *ap* are returned by the algorithm
During Monte Carlo simulation, the filter phase and refine phase of CGC algorithm is run for each individual random datasets created for the Monte Carlo simulation and the highest *Log LRC* are stored and the circles' p-values are determined using the ordered list
of these highest *Log LRC*.



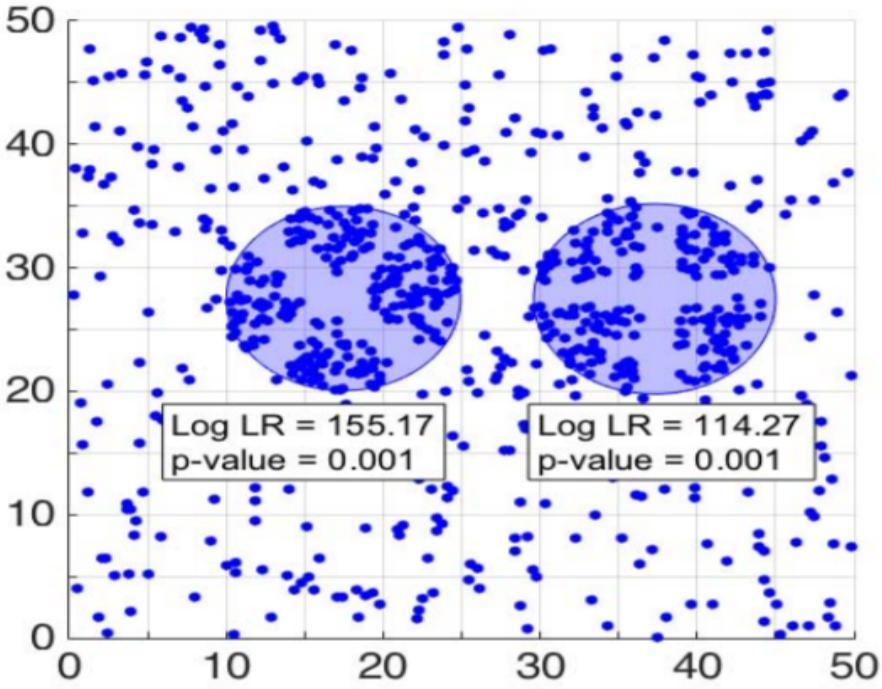



Cell center


Log LR = 58.72


Log LR = 114.27

- Output


Log LR = 155.17
p-value = 0.001
Log LR = 114.27
p-value = 0.001

Dataset used:

https://data.world/data-society/uber-pickups-in-nyc/activity

Context

This data contains on over 4.5 million Uber pickups in New York City from April to September 2014, and 14.3 million more Uber pickups from January to June 2015. Trip-level data on 10 other for-hire vehicle (FHV) companies, as well as aggregated data for 329 FHV companies, is also included. All the files are as they were received on August 3, Sept. 15 and Sept. 22, 2015. We will use as many points as our implementation would allow.

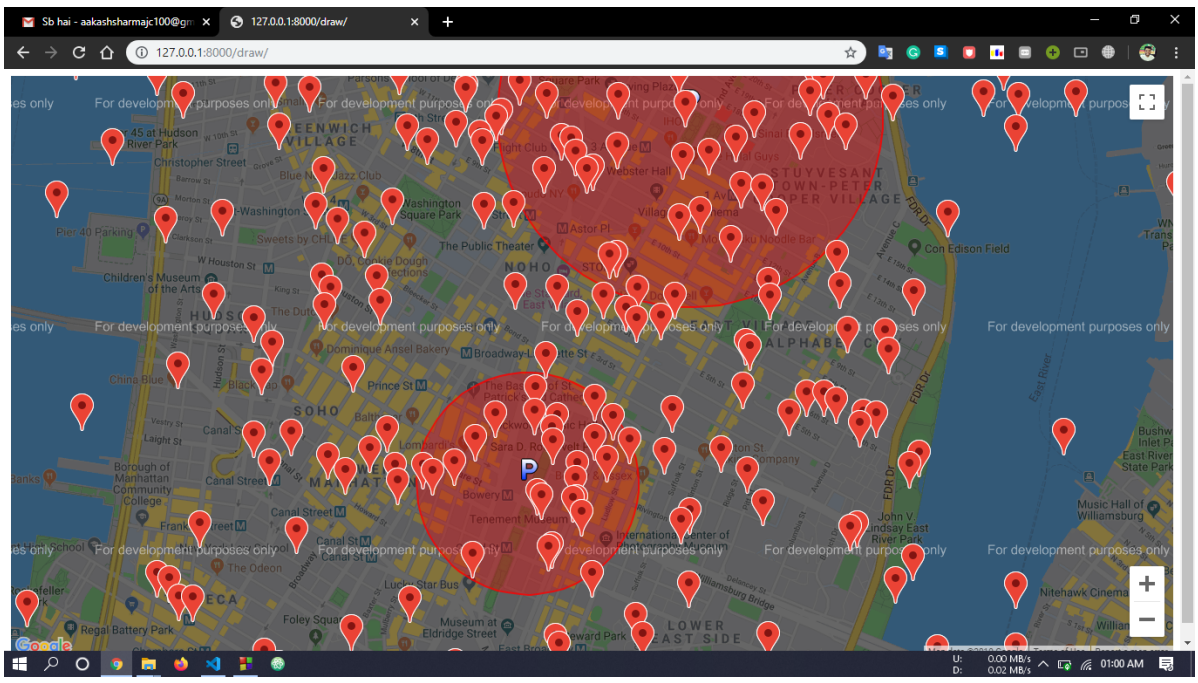
Content:

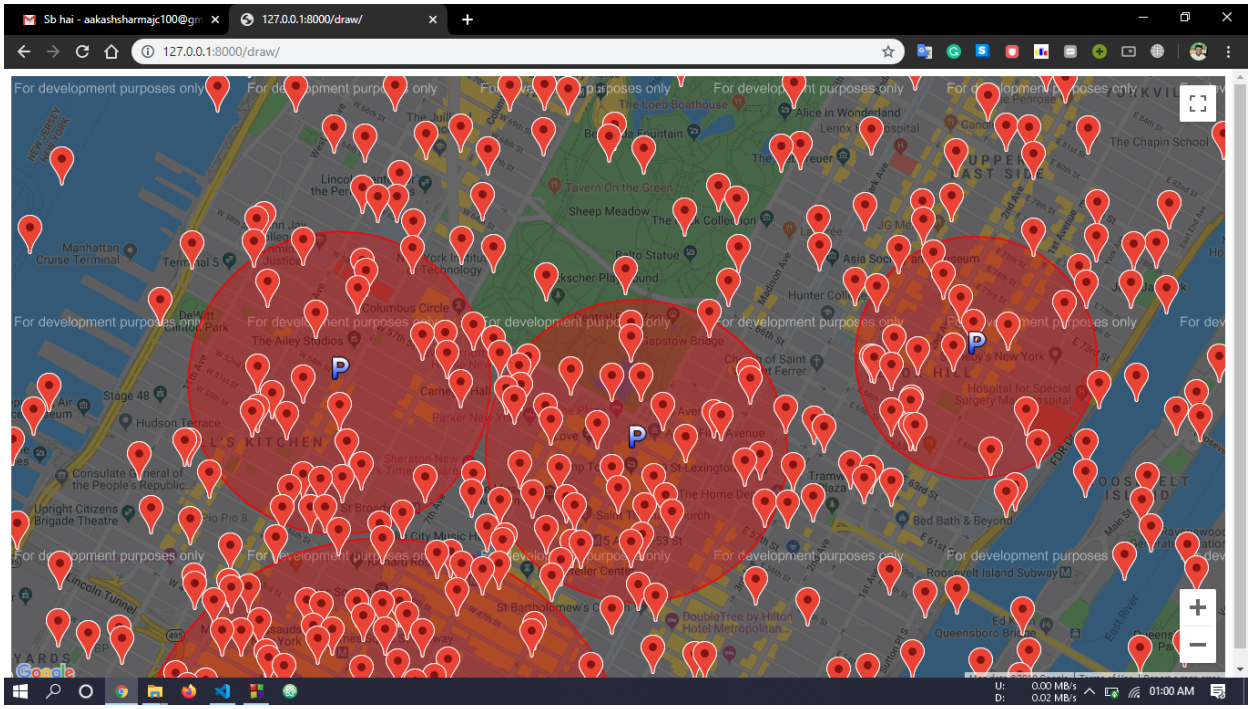
It contains the following 4 fields:

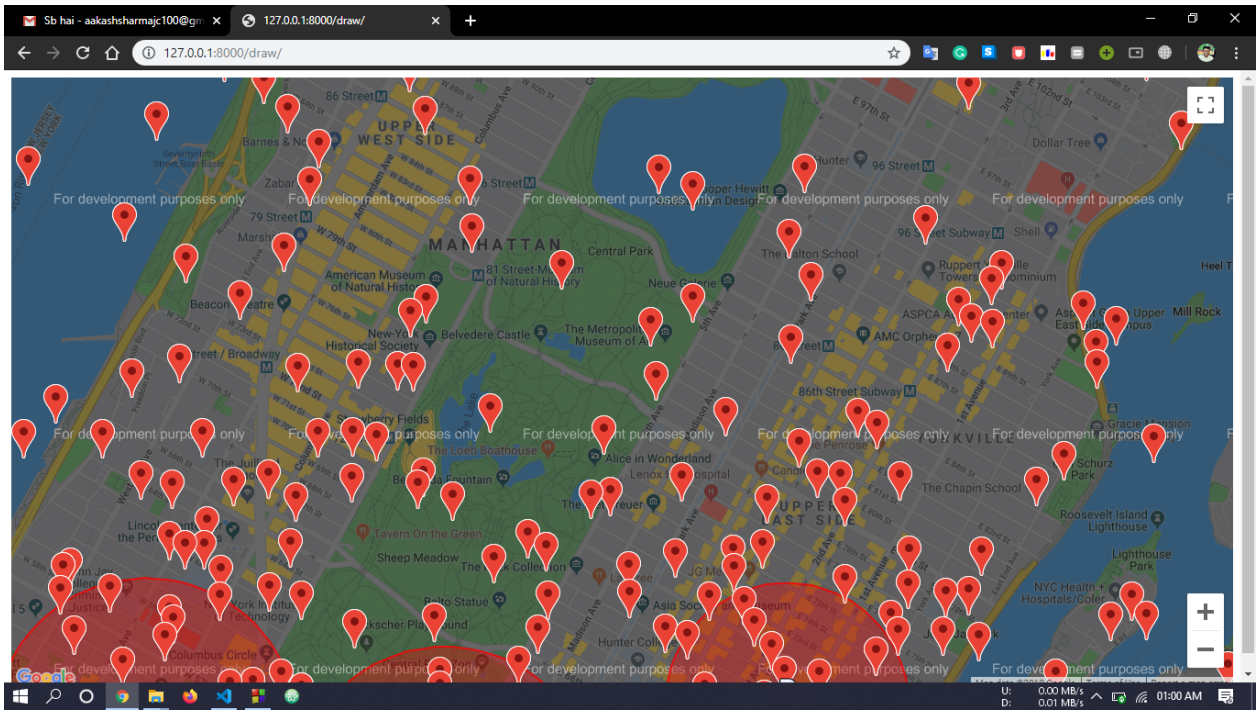
- Date/Time : The date and time of the Uber pickup
- Lat : The latitude of the Uber pickup
- Lon : The longitude of the Uber pickup
- Base : The TLC base company code affiliated with the Uber pickup

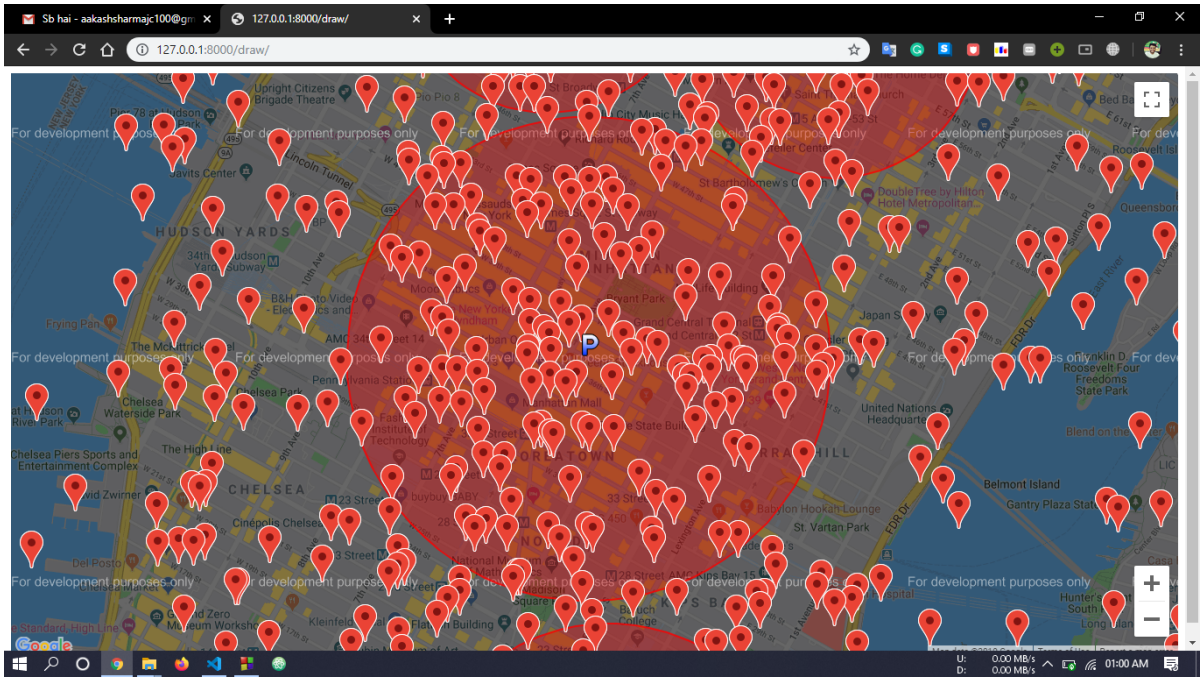
Execution and Outcomes

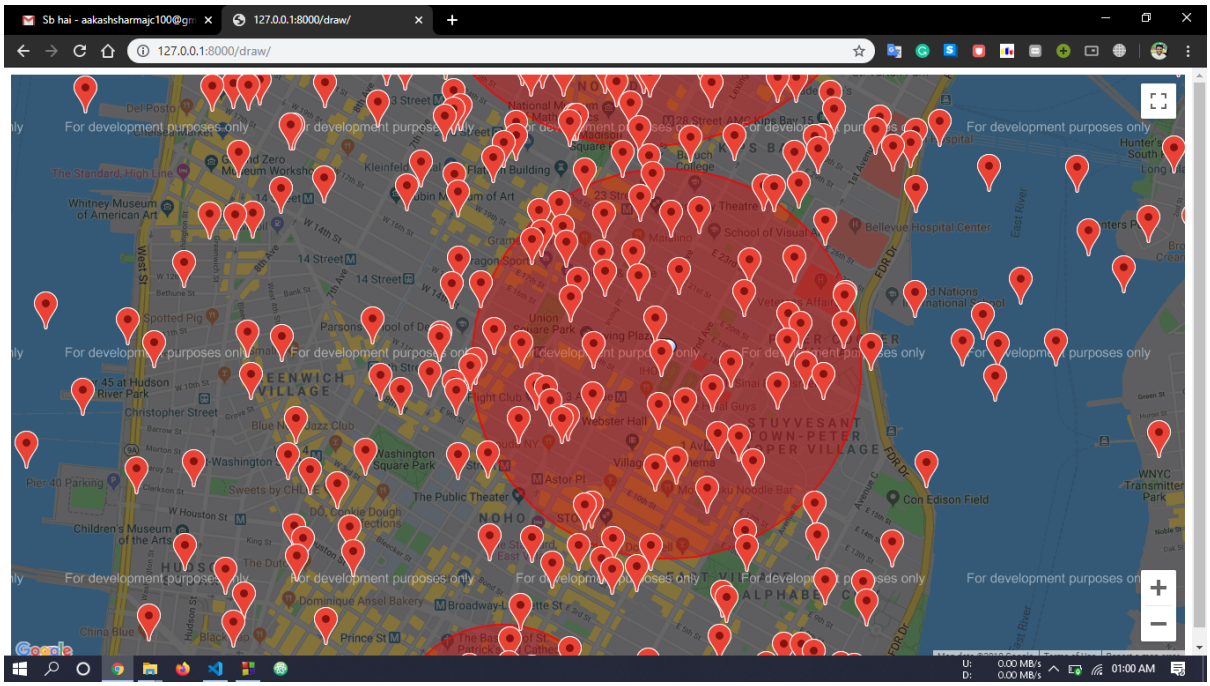
Test data 1:



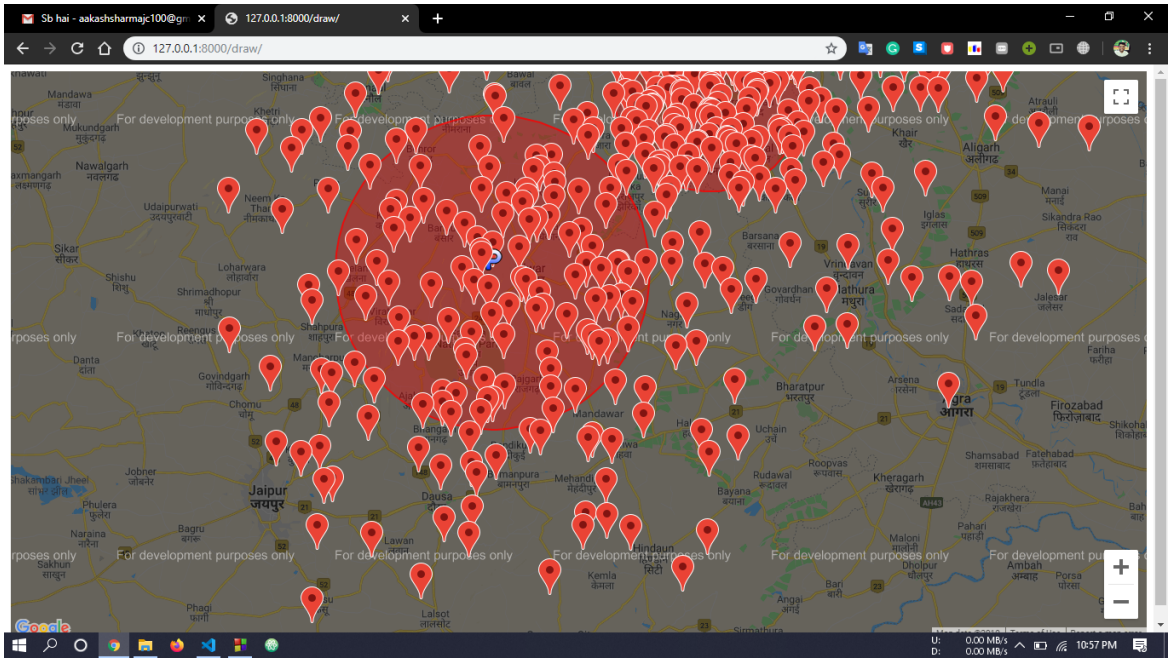

Sb hai - aakashsharmajc100@gm
127.0.0.1:8000/draw/
For development purposes only
Google
01:00 AM

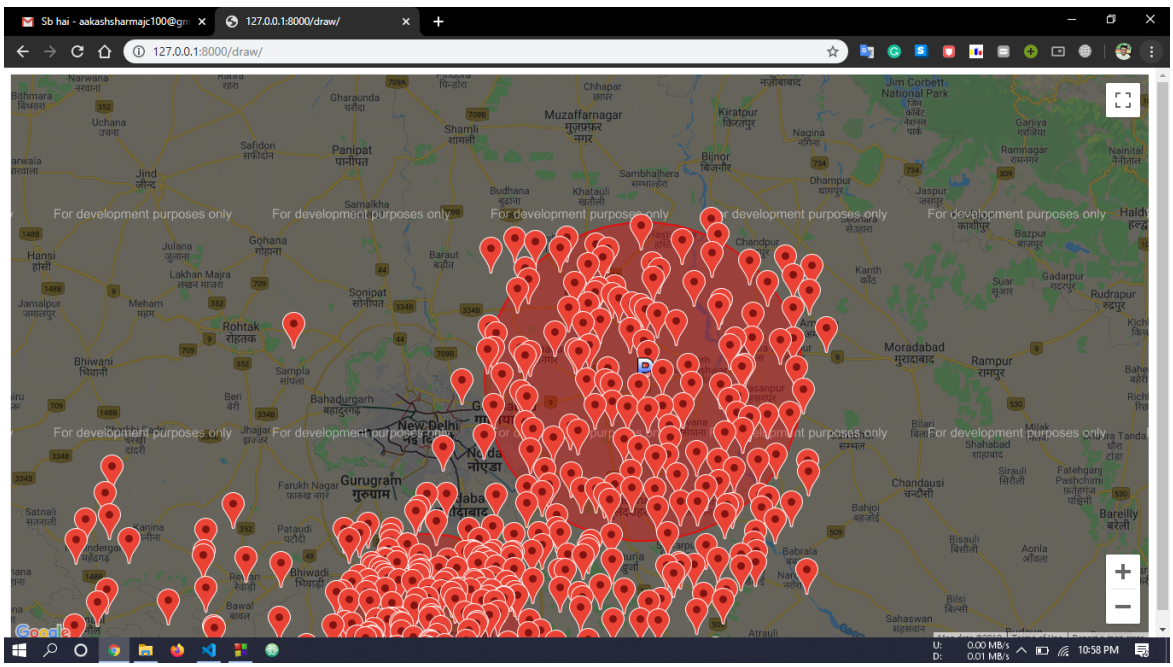

Sb hai - aakashsharmajc100@gm
127.0.0.1:8000/draw/
For development purposes only
MANHATTAN
UPPER WEST SIDE
UPPER EAST SIDE
Google
01:00 AM

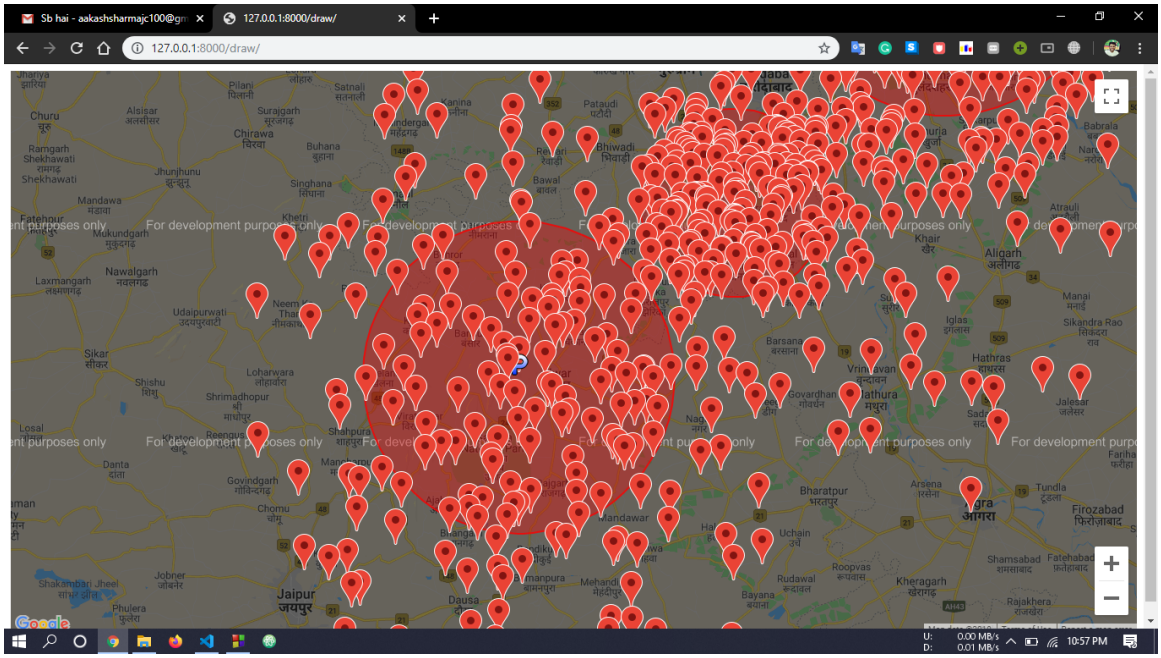

Sb hai - aakashsharmajc100@gm
127.0.0.1:8000/draw/
127.0.0.1:8000/draw/
For development purposes only
HUDSON YARDS
CHELSEA
Chelsea Piers Sports and Entertainment Complex
The High Line
Chelsea Waterside Park
Cinépolis Chelsea
Del Posto
United Nations Headquarters
Belmont Island
Gantry Plaza State
St. Vartan Park
Roosevelt Four Freedoms State Park
Google


Sb hai - aakashsharmajc100@gm
127.0.0.1:8000/draw/
127.0.0.1:8000/draw/
For development purposes only
Whitney Museum of American Art
14 Street
Washington Square Park
The Public Theater
Con Edison Field
Bellevue Hospital Center
Google

Test data 2:


Sb hai - aakashsharmajc100@gm
127.0.0.1:8000/draw/
127.0.0.1:8000/draw/
For development purposes only
Jaipur
जयपुर
Agra
आगरा
Mathura
मथुरा
Aligarh
अलीगढ़
Firozabad
फिरोज़ाबाद
Bharatpur
भरतपुर
Hathras
हाथरस
Google
10:57 PM


Sb hai - aakashsharmajc100@gm
127.0.0.1:8000/draw/
127.0.0.1:8000/draw/
For development purposes only
Gurugram
गुरुग्राम
Noida
नोएडा
New Delhi
Moradabad
मुरादाबाद
Rampur
रामपुर
Bareilly
बरेली
Rohtak
रोहतक
Panipat
पानीपत
Muzaffarnagar
मुज़फ़्फ़रनगर
Google
10:58 PM



Conclusion

This work explored the Geographically Robust Hotspot Detection (GRHD) problem, which is important for societal applications e.g. epidemiology, environmental criminology, etc. GRHD is challenging due to the difficulty of enumerating all possible hotspots and the high computational cost of the statistical significance test. This paper proposed a CGC algorithm which discovers statistically significant hotspots which couldn't be discovered previously by SaTScan. A case study demonstrated CGC's superior performance over SaTScan on a real crime dataset. Experiments also showed that the proposed algorithm (CGC) is highly scalable.