# Detecting Significant Accident Hotspots in Spatial Data

Enrol.No.s - 16103156, 16103200, 16103230

Name of Students - Rhythm Malhotra, Shreyas Jain, Shruti Pandey

Name of Supervisors - Dr. Sandeep Kumar Singh, Dr. Manish Kumar Thakur, Dr. Indu Chawla


**May - 2020**

**Submitted in fulfillment of the Degree of**

**Bachelor of Technology**

**in**

**Computer Science Engineering**

**DEPARTMENT OF COMPUTER SCIENCE ENGINEERING AND INFORMATION TECHNOLOGY**

**JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA**

**(I)**

# TABLE OF CONTENTS

## (II)
## DECLARATION

We hereby declare that this submission is our own work and that, to the best of our knowledge and belief, it contains no material previously published or written by another person nor material which has been accepted for the award of any other degree or diploma of the university or other institute of higher learning, except where due acknowledgment has been made in the text.

Place: Signature:

| | | | | |
|---|---|---|---|---|
| Date: | Name: | Rhythm Malhotra | Shreyas Jain | Shruti Pandey |
| | Enrollment No: | 16103156 | 16103200 | 16103230 |

**(III)**

## CERTIFICATE

This is to certify that the work titled "**Detecting Significant Accident Hotspots in Spatial Data**" submitted by "**Rhythm Malhotra, Shreyas Jain, and Shruti Pandey**" in partial fulfillment for the award of the degree **B.Tech.** of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor:

| | | | |
|---|---|---|---|
| Name of Supervisor: | Dr Sandeep Kumar Singh | Dr Manish Kumar Thakur | Dr Indu Chawla |
| Designation: | Associate Professor | Associate Professor | Assistant Professor (Senior Grade) |

Date :

**(IV)**

## ACKNOWLEDGMENT

We thank our supervisor Dr. Manish Kumar Thakur, Dr. Sandeep Kumar Singh, and Dr. Indu Chawla, Dept. of Computer Science and Engineering, JIIT for their continuous guidance, encouragement, suggestions and constructive criticism that have contributed immensely to the evolution of our ideas on the project. Turning this idea into a project would not have been possible without their support. We thank them for providing us with all the facilities required.

Signature of the Students

| | | | |
|---|---|---|---|
| Name of Students: | Rhythm Malhotra | Shreyas Jain | Shruti Pandey |
| Enrollment Numbers: | 16103156 | 16103200 | 16103230 |

Date:

## (V)
## SUMMARY

In this project, we implement a graphical representation of the significant hotspots of road accidents based on spatial data, to help identify the accident risk-prone zones on the streets. Our vision is that the project could be improved further and used by traffic police in order to identify certain areas and prioritize the traffic regulation there. This map could also be proved useful to commuters if an alert is added in the GPS to signify when one has walked into a traffic hotspot, so they could lower their speeds and take important precautions.

We first cluster the points using the Density based clustering algorithms clustering algorithm which does not require any previous information of boundaries or shape of clusters. Besides this, since the decision of declaring a cluster as a hotspot is crucial, considering the resources and effort that is going in it , the possibility of chance clusters in the result needs to be eliminated. Hence Significance testing of the clustering result is important.

To ensure only the significant clusters, we have applied the Montecarlo estimation algorithm, implemented in python in order to analyze the data and determine a threshold value for the minimum number of points in a cluster so as to exclude the chance/false positive results. The results obtained are hence plotted on a map.

We use two popular density based algorithms in this project, DBSCAN and OPTICS, in order to compare the results based on the V-Measure obtained by both of the algorithms on the given data. We also study about the advantages and disadvantages in the implementation of one over the other.

**(VI)**

**LIST OF FIGURES**

## (VII)

## LIST OF TABLES

# CHAPTER 1. INTRODUCTION

## 1.1 GENERAL INTRODUCTION

One of the most common causes of loss of human life are road accidents. .The number of automobiles and vehicles on the road have increased with the increasing population and hanging lifestyle. We cannot stop these accidents but we sure can try to be careful in prevention of these mishappenings. One way for this is to detect the geographical sections or areas that see a high number of accidents and provide suitable measures in these areas for their prevention. These places are called “black spots” or “hotspots”.

Various methods are present that can be used to detect hotspots. One method amongst them is the geostatistical technique. This technique is different from other approaches by considering the effects of unmeasurable parameters through the concept of spatial data mining that determines the autocorrelation between the crash event over a geographical space.

Spatial data is being used more and more widely used for various applications in different sectors. It helps us provide a clear picture of thefts, traffic jams, accidents, or a particular disease spread across the map. An analysis of such data can help one identify particular areas that are more prone to a certain phenomenon or activity and keeping that in mind specific steps can be taken to pay more attention towards those areas, deploy more resources if required, and alert the residential or commuters to be more careful in those specific areas. Maps could also be connected to GPS in order to alert drivers to slow down when they drive into accident hotspots.

To determine hotspots in a spatial dataset, clustering of points is required, but clustering results are not always accurate. Some output clusters might be a simple result of coincidence. Considering the scale of seriousness of these issues and the effort and time that needs to go into resolving them, it is crucial to eliminate the chance clusters.

Previous work on significance testing uses clustering techniques like spatial scan statistics to enable this, but these techniques require knowledge of the predefined shape of clusters (circle, disc), etc. In real scenarios, clusters rarely have a defined shape. Density based clustering algorithms, on the other hand cluster even the indefinite shapes without requiring prior knowledge and very few parameters. In this project, we work on this problem with two density

based clustering algorithms, DBSCAN and OPTICS, we perform significant testing on both and analyze the results comparatively.

In order to filter the significant clusters, we use number of points in a cluster as the test rather than density and likelihood ratio that would require the area of clusters which is highly complex to calculate. To calculate the right threshold value according to the data of points given, we use the Monte Carlo estimation algorithm, by taking M no. of iterations, and a random point distribution, and analyzing the maximum cluster size each time, in order to find a threshold value, only the probabilities above which will be considered significant. The significant clusters hence obtained are then represented on the map.

## 1.2 PROBLEM STATEMENT

The problem that is being addressed here is to build a system that helps identify significant accident hotspots in the areas the data of which is provided. Using the past raw data of accident situations that have occurred, the program intends on determining which geographical areas are more prone to accidents than others. Using this information, important steps like deploying more police force in those hotspots and decisions of constructing bridges etc. can be taken in order to reduce the damage caused by those accidents.

Besides this, this map is useful to the regular people driving in the area to identify the more accident-prone paths and take necessary precautions or perhaps avoid them completely. This problem is hence very crucial to be discussed and worked upon.

With the intent of eliminating any chance clusters that might have been an output of clustering, the program also performs significance testing and uses the Monte Carlo estimation approach because deploying resources on a false hotspot can prove to be very costly amidst limited resources. Hence, the problem catered here is to filter the clusters with MonteCarlo estimation value above a decided threshold so that no false clusters appear in the results.

We perform significant clustering on the data using two density based clustering algorithms, DBSCAN and OPTICS. We hence perform a comparative analysis based on the resultant V

Measure[15] to obtain the more conveniently implementable and accurate algorithm that could potentially be used further

This kind of significance testing of DBSCAN and OPTICS resultant clusters without the knowledge of area and shape of clusters is unique to our project and we have tried to do justice to this problem statement.

### 1.3 SIGNIFICANCE/NOVELTY OF THE PROBLEM

Detection of significant spatial clusters has been widely applied in important domains like health, safety, transportation, and environmental science. Significant hotspots are used to monitor and alert the people about disease outbreaks. Hence, significant clustering is an important methodology and tool to identify hotspots and subsequently take measures to eliminate the losses caused by these threats.

In transportation, many local governments have launched "Zero Death" initiatives to save lives from traffic-related accidents. With significant clustering, planners can find roads with significantly high rates of car accidents or pedestrian fatalities, which indicate potentially unsafe driving conditions (e.g., damaged sidewalks, potholes)**.** Besides, drivers can be alerted while driving into an accident hotspot so he/she could decrease the speed and take required precautions to avoid any such situation.

The problem is challenging because many application domains like transportation have low-tolerance to false positives. It means that an area could be wrongly clustered due to noise or some other factors which we may think of as an error. In the case of our domain, identifying an accident-prone point as a safe point could cause danger to the people and in the opposite case, it would lead to loss of funds, people and time. This could cause unnecessary social stress as well as waste of public resources.

Many methods of significant clustering have been devised to solve this purpose but they require a pre-defined geometric shape (ring, circle, ellipse) or require a pre-defined irregular partitioning of the spatial area (e.g., country or state boundaries) . This requirement is a limitation because in real-world data, the shapes of clusters are irregular and cannot be predefined . The clusters are

affected by many factors and may change through time, making it difficult to represent them well with predefined shapes or partitioning.

In this project, hence, we aim to greatly reduce the risk of false positives by using significance testing on DBScan and OPTICS clusters. This clustering or significance testing does not require the shape or area of clusters because it can be very complex to calculate. This makes this project more significant for practical scenarios.

We also compare both the algorithms, DBSCAN and OPTICS based on the outputs obtained on significant testing as well as other advantages and disadvantages of using one over the other, in order to determine the algorithm that might work better for a potential deployed system .

This combination of the clustering technique and significance test is unique to our model.

## 1.4 EMPIRICAL STUDY

During the course of the project, we went through several previously done related works in order to identify the advantages as well as disadvantages of using various algorithms, tools and technologies.

In terms of programming languages, Python is often compared to other interpreted languages such as Java, JavaScript, Perl, Tcl,C++ or Smalltalk. Comparisons to Scheme and Common Lisp are also made. But in our research, it was found that python provides best functionality to deal with problems related to data analysis and statistics as well as scientific functions.It also provides great libraries to deal with data science applications.

One of the main reasons why Python is preferred in research communities and other applications is because of it being very easy to use and the syntax which makes it easy to adapt. According to engineers researching from academics and industry, Artificial Intelligence frameworks available by Python APIs, as well as the scientific packages have made Python so incredibly versatile and productive. There has been a lot of evolution in deep learning Python frameworks and it's rapidly upgrading. Python provides a large collection of libraries that help to solve complex problems easily, and build strong system and data applications. For clustering, post research we decided on using the Density based clustering algorithm for multiple reasons.

Spatial Scan Statistic Clustering is another well known technique that can detect significant clusters, it's strength is that it includes statistical significance by calculating likelihood ratio which eliminates the chance clusters, but it requires the clusters to have a predefined shape like a circle or ellipse. In practical data,though, clusters are irregularly shaped. Not all clusters can be defined in circular frames. The later works that have worked upon modifying Spatial Scan Statistics so as to identify irregular shaped clusters [2][3], still require a pre-defined irregular partitioning of the spatial domain (e.g., county boundaries in a state). This is not available in many practical applications. Besides this, these solutions make the problem more complex and time consuming, and require estimations of area and parameters, which is another complex and highly fallible.

DBSCAN on the other hand, is another algorithm that has a large impact on a wide range of areas in data analysis, including outlier detection, computer vision, and medical imaging. Which

- Can find arbitrarily shaped clusters. Does not require predefined shapes or areas of clusters.
- It does not require for one to specify the number of clusters in the data beforehand
- DBSCAN requires just two parameters, epsilon and min points, (which need to be decided carefully.)
- DBSCAN has a notion of noise, and is robust to outliers making it suitable for our application.

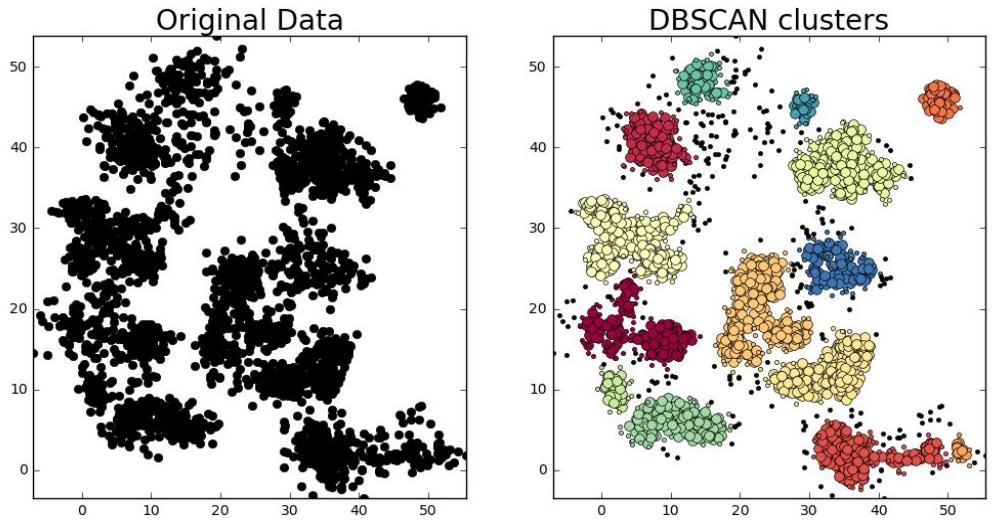



**Fig 1. Depicting result of DBSCAN clustering.** [16]

Despite of this, there are some weaknesses that DBSCAN holds that might affect the results:

- DBSCAN Does not work as well when dealing with clusters that have varying densities. While DBSCAN works great in separating low density clusters from high density clusters, it struggles when dealing with clusters of similar density.
- DBSCAN is highly sensitive to the parameters,i.e. MinPts and ε. Since the user is to choose these parameters, a slight variation can adversely affect the results.

To overcome these weaknesses,we also have implemented another density based clustering algorithm OPTICS, that invokes a different process. It will create a reachability plot that is then used to extract clusters and although there is still an input, maximum epsilon, it is mostly introduced only if you would like to try and speed up computation time. The other parameters don't have as big an effect as their counterparts in other clustering algorithms, and are much easier to use defaults. On the other hand, OPTICS is slower than DBSCAN in term of computational time

None of the two algorithms, based on the research performed can be straight up said to be better than the other. We hence use both the algorithms to cluster the data and compare them on another parameter i.e the V Measure, to conclude what works the best for our application.

Also, For the purpose of determining statistical significance , we explore several test statistics for the significance testing (Table 1). For a given cluster from a point distribution, its test statistic value will be used to determine if it is a chance cluster or not. According to Table 1(obtained by [1]), both density and likelihood ratio require calculation of the cluster's area in Euclidean space. In the framework of spatial scan statistics, cluster regions are pre-defined (e.g., all circular regions of certain areas) so it is trivial to calculate the areas (e.g., $\pi r^2$ ). However, area calculation is not well-defined in the DBSCAN framework. Since DBSCAN clusters can have arbitrary shapes (e.g., concave), a model to depict the region like convex hull modeling will introduce large errors into the estimation. Even on estimating (considering only two dimensional spatial case), just one area calculation will take $O(|c| \ 2 \log |c|)$ time where $|c|$ is the number of core points. This would hence require higher complexity and a huge amount of time in higher-dimensional space.

Cluster size n (i.e., number of points in the cluster) is another measure being used in scan statistics methods that does not require computation of the area[4]. Cluster size is not required to be normalized by area. For DBSCAN, the normalizing conditions come naturally through the required parameters (ϵ,minPts). Hence cluster size(n) is considered our test statistic for this project.

| Test Statistic | Area Of Cluster | Normalisation | Bias Towards Small Clusters | Computation |
|---|---|---|---|---|
| Density *d* | Required | Area | Yes [5],[4] | Area Dependent |
| Likelihood ratio *lr* | Required | Area+ Null Hypothesis | Yes(less)[6],[4] | Area Dependent |
| Cluster Size *n* | N/A | Search context dependent, e.g., fixed radius [11], (ϵ,minPts) in DBSCAN | No | O(1) for a given cluster |

**Table 1. Potential Test Statistics and their properties[1]**

## 1.5 BRIEF DESCRIPTION OF SOLUTION APPROACH

To execute the planned solution approach, we have worked through this project in the following manner.

First Data Gathering, for our application, we needed data of accidents in a particular area over multiple years. Only after analysing a place's data for several years, we become able to obtain sufficient points to perform the clustering as well as significance testing and finally declare a place as a hotspot or not.

We also further analyse the data by plotting the variation of accident severity etc. attributes with respect to time, which could be very informative for taking further actions.

Moving on to the clustering, we coded the standard DBSCAN and OPTICS clustering approach on our particular data, and as we know, selection of ε and *minpoints* is very important in the final DBSCAN output, We analyzed the data and the distance between corresponding points, and chose the parameters such that the ε is not too big or not too little either because both the conditions can affect the output adversely.

Coming to the Significance Testing that is MonteCarlo estimation algorithm, Here , for test statistic we chose 'n' as the number of clusters since it is not dependent on area of cluster. We apply Monte Carlo functions by taking a random point distribution each time in M iterations and then determining the size of the largest cluster each time and sorting the data in ascending order.

In the algorithm, the acceptance or rejection of a cluster as a false onethen is then based on whether its p value which is r by M is greater than 1/α or not. Where r is the rank of that cluster's size on the sorted list. M and α need to be chosen by the programmer.

On the basis of the analysis of the data, we hence decided M and α for this project, and code the Monte Carlo algorithm for execution on our clusters.

The final and very important step would be plotting these clusters on a map in a descriptive sort of way which makes it easy for the user to identify the significant hotspot areas as well as the false ones. For this, we used the openstreet map data and python libraries like GeoPandas, mplleaflet etc. to plot the clusters over the map as well as incorporate text names and zoom functions. Besides that, we used different colors to identify the significant and non significant clusters so as to make it easily differentiable.

## 1.6 COMPARISON OF EXISTING APPROACHES TO PROBLEM FRAMED

Spatial data mining is simply the discovery of informative dependencies and characteristics that may exist in spatial databases. Several applications of Spatial data clustering have been worked upon and several approaches of similar problem statements have been put across. Here we compare the advantages and disadvantages of those approaches with respect to a practical situation.

The applications of spatial clustering in various different papers vary from Disease outbreaks like cancer database, crime rates data, traffic data etc. Our project deals with data of accidents in a particular area over the years.

Coming to the algorithms, in the domain of significant testing, Spatial scan statistics[10] are commonly used for geographic cluster detection and evaluation. Several pieces of work applying Spatial scan statistics [6,19] have been witnessed. The problem with this approach being the requirement of a predefined shape of clusters like circle, ring etc. Whereas in practicality like our project, the clusters are irregular in shape. Improvisations on this technique [6,17] have also been framed for the detection of irregular shaped clusters but they still assume predefined irregular shaped partitions in the data and also make the process more complex and time consuming.

Hence, taking reference from [1], we decided to use the MonteCarlo approach for significance testing along with another clustering technique. We found that not much work had been done on this combination before.

For the clustering algorithm, we studied that there are a few well known fast algorithms for clustering large data sets such as DBSCAN and OPTICS [11,12,16],

CLARANS(Clustering Large Applications based upon RANdomized Search)[9]:

Moreover, the quality of the results can not be guaranteed when N is large since randomized search is used in the algorithm. In addition, CLARANS assumes that all objects are stored in the main memory. This, hence, enables only a limited size of the database to which CLARANS can be applied. CLARANS ‘s computational complexity is still high.

BIRCH(Balanced Iterative Reducing and Clustering using Hierarchies)[8]:

BIRCH may not work well when clusters are not of a fixed spherical shape because the algorithm uses the concepts of spheres such as radius to control the boundary of a cluster.

STING(a hierarchical statistical information grid based approach )[7]:it’s probabilistic nature may imply loss of accuracy in query processing.

We decided to use DBSCAN and OPTICS amongst these because of its ease to use and flexibility. It is appropriately fast and detects irregularly shaped clusters requiring only two parameters, which on being calculated wisely, result in satisfactory clusters.

## CHAPTER 2. LITERATURE SURVEY

### 2.1 SUMMARY OF PAPERS STUDIED

**Table 2.**

| | |
|---|---|
| **Title** | Significant DBSCAN towards Statistically Robust Clustering[1] |
| **Authors** | Yiqun Xie,Shashi Shekhar |
| **Year of Publication** | 2019 |
| **Publication Details** | SSTD '19: Proceedings of the 16th International Symposium on Spatial and Temporal Databases, |
| **Summary** | This paper provides a solution for accident hotspot detection using spatial clustering using DBSCAN.. It also makes use of significance testing to avoid false clusters while clustering using baseline Monte Carlo Estimation. This paper throws light on various test statistics that can be used for this testing (density and likelihood ratio) and why we don't use them but instead use cluster size for it for the problem mentioned in the paper. It also proposes a dual convergence algorithm to speed up the process of significance testing. Based on the significantly tested DBSCAN they introduce a heuristic search to identify clusters of varying densities. |

**Table 3.**

| | |
|---|---|
| **Title** | Nondeterministic Normalization based Scan Statistic (NN-scan) towards Robust Hotspot Detection: A Summary of Results [4] |
| **Authors** | Yiqun Xie and Shashi Shekhar. |
| **Year of Publication** | 2019 |
| **Publication Details** | SIAM International Conference on Data Mining (SDM'19). |

| | |
|---|---|
| **Summary** | Existing methods of hotspot detection rely on test statistics (e.g. likelihood ratio, density) do not consider spatial nondeterminism, leading to false and missing detections. In this paper, theoretical insights into the limitations of related work are provided, and a new framework, Nondeterministic Normalization based scan statistic (NN-scan) is proposed, to address the issues.A DynamIc Linear Approximation (DILA) algorithm to improve NN-scan's efficiency is also proposed. In experiments, it is shown that NN-scan can significantly improve the precision and recall of hotspot detection and DILA can greatly reduce the computational cost. |

**Table 4.**

| | |
|---|---|
| **Title** | Transdisciplinary Foundations of Geospatial Data Science. [6] |
| **Authors** | Yiqun Xie , Emre Eftelioglu , Reem Ali, Xun Tang, Yan Li, Ruhi Doshi, Ruhi & Shashi Shekhar. |
| **Year of Publication** | 2017 |
| **Publication Details** | ISPRS International Journal of Geo-Information. |
| **Summary** | Our key takeaway from this paper is why statistical significance testing is important. Conventional clustering algorithms like K-means, DB-Scan approaches do not consider statistical significance. K- Means may form clusters even in a uniform distribution. Density based methods may result chance patterns if applied to geospatial hotspot detection. Methods that use statistical significance testing (e.g. likelihood ratio based p-value test) can eliminate such chance patterns. Thus encouraging us to use significance testing. |

**Table 5.**

| | |
|---|---|
| **Title** | Clustering Methods and Bound Value in Classify Density Traffic Accident Areas [10] |
| **Authors** | Hsien-Tsung Chang, Hieu Nguyen. |
| **Year of Publication** | 2017 |
| **Publication Details** | 5th IIAE International Conference on Industrial Application Engineering 2017 |
| **Summary** | This paper demonstrates the use of DBSCAN algorithm to form the traffic accidents records into clusters on the basis of locations and density of the accident. After that, they identify the similarity or characteristics of each cluster using the Bound Value. The result gives the clusters and their abnormal characteristics that may be the cause why those locations have so many accidents based on the abnormal characteristics. This paper tells that DBSCAN is better than K-means because it works perfectly for density based problems. Also they after many tries they came to a conclusion that the parameter epsilon e should be between 140 to 220 meters and the minPoints should be between 4 to 7 when applying for urban areas |

**Table 6.**

| | |
|---|---|
| **Title** | Improved Density Based Spatial Clustering of Applications of Noise Clustering Algorithm for Knowledge Discovery in Spatial Data [12] |
| **Authors** | Arvind Sharma, R. K. Gupta and Akhilesh Tiwari,<br><br>Department of CSE & IT, RustamJi Institute of Technology, Tekanpur |
| **Year of Publication** | 2016 |

| | |
|---|---|
| **Publication Details** | Hindawi Publishing Corporation ,Mathematical Problems in Engineering, Volume 2016, Article ID 1564516. |
| **Summary** | This paper discusses the various types of clustering techniques (density based methods , hierarchical methods, partition methods, grid based methods). It tells that DBSCAN is the best algorithm for clustering. However it has some drawbacks which they plan to eliminate by proposing an improved version called IDBSCAN. It improves the working of the original algo in determining clusters of points that are close to each other and clusters of varying densities.Further the performance of the suggested algo is discussed. |

**Table 7.**

| | |
|---|---|
| **Title** | A flexibly shaped spatial scan statistic for detecting clusters [13] |
| **Authors** | Toshiro Tango & Kunihiko Takahashi |
| **Year of Publication** | 2005 |
| **Publication Details** | International Journal of Health Geographics |
| **Summary** | Since the scan statistic algorithm uses a circular window to define the cluster areas and thus the non-circular clusters are not detected properly. This paper presents a proposal for detecting non-circular, irregularly shaped clusters that are also much larger than the true clusters in previous research. As a result, though the original spatial scan statistics shows a higher level of accuracy in detecting the circular shaped clusters . The newly proposed modified spatial scan statistic is found to have decently good usual powers as well as the ability to detect the noncircular clusters more accurately than previously. But for large cluster sizes, this method is not as practically feasible. |

**Table 8.**

| | |
|---|---|
| **Title** | Hotspot detection and clustering: ways and means [13] |
| **Authors** | Andrew B. Lawson |
| **Year of Publication** | 2010 |
| **Publication Details** | Springer Science+Business Media, LLC 2010 |
| **Summary** | This paper talks about the development of methods for hotspots identification and clustering. It talks about the use of density estimation, scan testing and model-based approaches to clustering on various fields such as use of likelihood ratio in the health sector. When problems involve adjustments then modeling will be a better approach whereas when adjustment is not required then testing procedures may be the most apt solution. |

**Table 9.**

| | |
|---|---|
| **Title** | A Comparative Study Of K-means, DBSCANAnd OPTICS [14] |
| **Authors** | Hari Krishna Kanagala,Dr. V.V. Jaya Rama Krishnaiah |
| **Year of Publication** | 2016 |
| **Publication Details** | International Conference on Computer Communication and Informatics (ICCCI -2016), Jan. 07 – 09, 2016, Coimbatore, India. |
| **Summary** | This paper evaluates the performance of the different clustering approaches K-Means, DBSCAN, and OPTICS in terms of accuracy, outlier's formation, and cluster size prediction. The K-Means algorithm is only used when the number of clusters is defined and when using a huge dataset. But K-Means cannot identify outliers. DBSCAN can |

| | |
|---|---|
| | determine clusters of random shapes and also classify points as noise or outliers. when compared to other algorithms, it is very fast. DBSCAN requires two parameters ε and MinPts to be set by the user. different parameter settings may lead to different results. It is not completely accurate in distinguishing separated clusters even if they have different densities if they are too close to each other. OPTICS was introduced to overcome these flaws. In OPTICS high-density clusters are given priority over lower density clusters. OPTICS also uses parameters (ε and MinPts) to be determined by the user that may lead to varied results. |

**Table 10.**

| | |
|---|---|
| **Title** | V-Measure: A conditional entropy-based external cluster evaluation measure. |
| **Authors** | Andrew Rosenberg and Julia Hirschberg, |
| **Year of Publication** | 2007 |
| **Publication Details** | Joint Conference on Empirical Methods in Natural Language Processing. |
| **Summary** | In this paper, V-measure, an external entropy based cluster evaluation measure is presented.It is an accurate evaluation and combination of two desirable aspects of clustering, homogeneity and completeness. It is stated that, V-measure provides a solution to several problems that affect the existing cluster evaluation measures which are 1) dependence on clustering algorithms or chosen data set, 2) the problem of matching, such that the only a fraction of clustered data points are evaluated. In this paper, comparison to the other evaluation algorithms like purity and entropy is made as well and advantages of V Measure are stated. It is also shown that V Measure incorporates all the desirable properties of clusters like high no.of clusters and low error probability. |

**Table 11.**

| | |
|---|---|
| **Title** | OPTICS: Ordering Points To Identify the Clustering Structure |
| **Authors** | Mihael Ankerst, Markus M. Breunig, Hans-Peter Kriegel, J&g Sander, University of Munich, Germany |
| **Year of Publication** | 1999 |
| **Publication Details** | SIGMOD 1999 Philadelphia |
| **Summary** | A new algorithm for the purpose of cluster analysis is introduced. This algorithm differs from the existing ones as it does not directly produce the clustering of a data set.; but rather creates an ordering of the database points representing its density-based clustering structure. The information contained in this ordering is equivalent to the density-based clusterings of the dataset corresponding to a range of parameter settings. It is also shown how to efficiently extract not only the traditional clustering information, but also the internal structure of the clustering. For data sets of medium size, the ordering can be represented graphically and for very large data sets, an appropriate visualization technique is introduced. The detailed algorithm of how OPTICS works and how to use the results for analysis is explained. |

## 2.1 INTEGRATED SUMMARY OF LITERATURE STUDIED

Many papers from varied domains such as medical, traffic, geological etc. use hotspot detection. Their case studies and methodologies have greatly increased our knowledge and provided valuable ideas for the implementation of this project

There are many conventional statistics based methods of detecting hotspots in a region. But they do not take into consideration several factors that cannot be numerically expressed. Thus the use

of spatial clustering was introduced. Scan Statistic clustering is popular in the research work involving significant testing but in practical data, since shape and internal segments are not defined, it isn't feasible to use. There are various other clustering methods available like K-Means, DBSCAN, OPTICS. Drawbacks and benefits of these methods are discussed in the studied papers.

The K-Means algorithm is used when the number of clusters is pre-defined and when using a huge dataset. But K-Means cannot identify outliers. DBSCAN can also work for large data sets and determine clusters of random shapes and also classify points as noise or outliers and it is very fast compared to other algorithms. DBSCAN requires two parameters ε and MinPts to be set by the user. different parameter settings may lead to different results. It is not completely accurate in distinguishing separated clusters even if they have different densities if they are too close to each other. OPTICS was introduced to overcome these flaws. In OPTICS high-density clusters are given priority over lower density clusters. OPTICS also uses parameters (ε and MinPts) to be determined by the user that may lead to varied results. Based on our study we found that DBSCAN serves the best for our purpose of traffic hotspot detection.

Various papers also advocated the use of statistical significant clustering to remove false clusters in DBSCAN . Many test statistics have been discussed like density ratio, likelihood ratio, size of cluster, shape based values etc. Considering the clustering method we use, we found that a threshold size of a cluster would be most suitable. To determine this size Monte Carlo estimation is the method, which has been used in some research work before and produced efficient results.

## CHAPTER3. REQUIREMENT ANALYSIS AND SOLUTION APPROACH

### 3.1 OVERALL DESCRIPTION OF THE PROJECT

As a solution approach to this problem, the approach designed is stated. The flowchart below describes the steps taken to execute the solution approach:

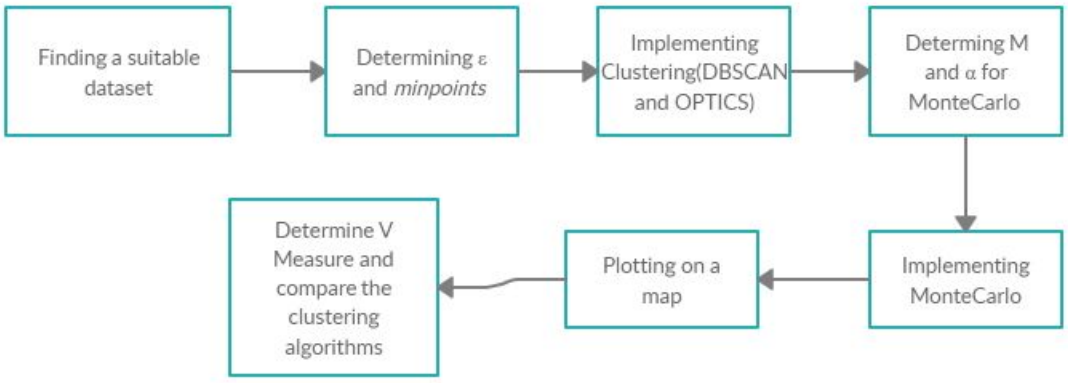



**Fig 2. Flowchart describing Solution approach**

For this application, we were required to have a suitable number of points in our dataset to be able to generate satisfactory results in the form of clusters. Besides, analysing data over the years provides a clearer picture of places that have been hotpots over the time or have better stats in the recent past. The official data of the UK over the past few years was found to be accessible and suitable. Hence, it was taken into consideration.

The next step is clustering the data and eliminating outliers using DBSCAN and OPTICS.

Determining the parameters of the DBSCAN algorithm is a very important step in order to obtain appropriate results. We analysed the data to determine the value of minpoints=10 and epsilon=50 metres by hit and trail . Parameters shouldn't be too large or too small.

For the implementation part of clustering algorithms, we coded the standard DBSCAN as well as OPTICS algorithm on python for our dataset.

```
In [10]: # Parameters
         eps_in_meters = 50.0
         num_samples = 10

         # Cluster the data
         earth_perimeter = 40070000.0  # In meters
         eps_in_radians = eps_in_meters / earth_perimeter * (2 * math.pi)
```

```
In [ ]: data['cluster']=OPTICS( min_samples=num_samples, metric='haversine').fit_predict(data[['rad_lat', 'rad_
        fig = px.scatter(data[['rad_lat', 'rad_lng']], x="rad_lat", y="rad_lng")
        fig.show()
```

**Fig 3. Implementation of DBSCAN**

Moving on to the part of significance testing, for executing the Monte Carlo algorithm, the detection of M i.e the no. of iterations as well as the significance level α is very important. We used hit and trial method for this, for each M*α we check the number of clusters, the density of clusters, whether certain Significant clusters are being included in the result or not, and hence we decide M as 1000 and α as 0.2.

For executing Monte Carlo we coded in python the algorithm that works for M iterations, a random point distribution is generated in which, and the size of the biggest cluster in each case is stored. These values are then sorted in order to now determine r, the rank of the size of the cluster we need to check. If r is greater than the product of M and α, the cluster chosen is significant. Otherwise not.

For the last step of plotting our significant clusters on a map, we coded the map using the data and geopandas library on python. Using another useful library mplleaflet, for zoom in, different colors for significant and insignificant as well as text name tags for different areas etc. functions. The functions of these libraries come handy.

More about the details of algorithms and their advantages and disadvantages have been discussed in the next unit.

## 3.2 REQUIREMENT ANALYSIS

**Functional Requirements**

- System requirements

1. System with Ubuntu or Windows operating system
2. Python 3.7
3. Jupyter Notebook to run the code

- Software requirements

This project aims at providing accidents hotspots as its ultimate output so that proper measures can be taken and resources can be applied to prevent accidents. DBSCAN and OPTICS are applied on spatial data and output is improved using significant testing by Monte Carlo Evaluation method. The outputs given by the two algorithms are also compared using measures completeness, homogeneity and v-measure. Other software requirements are given below.

1. The solution should be encoded in the proper format as accepted by the function in the library.
2. All essential libraries must be installed/imported.
3. Data and arguments passed to our imported functions should be in the correct format and follow the required constraints as given in the documentation.
4. Codes must be syntactically free from errors.

**Non- Functional Requirements**

1. Security and Privacy Requirements - There are no specific security requirements, anyone can access and use the information since it is for public use.
2. Reliability - The solution should provide reliability to the user.`
3. Accuracy - The solution should be able to reach the desired level of accuracy.
4. Usability - A traffic agent should be able to use the system in his job easily and identify the hotspots easily
5. Supportability - The system should be able to work on different datasets without major reengineering.

### 3.3 SOLUTION APPROACH

This project aims at identifying road accident hotspots by clustering of spatial data using Density based clustering algorithms. DBSCAN is a very efficient algorithm that is widely used for clustering. But it is also known that DBSCAN may not be completely effective in clustering the data points, ie., it also forms false positive clusters. It may form spurious clusters with points that are positioned closely. This inefficiency can be very harmful in real life. It can lead to wastage of resources and also loss of life. To get rid of the chance clusters we use significant testing with Monte Carlo Evaluation as the baseline algorithm. Following is the description of the dataset and various algorithms used in the project.

- **Dataset:**

The dataset we have used is the Road Safety Data published by the UK government. This is a very detailed road accident data set that includes the incident's geographical coordinates, as well as other related data such as the local weather conditions, visibility, police attendance and more. The dataset we used were for the year 2015, 2016,2017 and 2018. In total our dataset in CSV format has 561689 rows and 32 columns.

```
data.shape
(561689, 32)
```

**Fig 4. Dataset dimensions**

Several characteristics of the accident other than just position are available as attributes of the our dataset, for example, Longitude, Latitude, Police_Force, Accident_Severity, number_of_Vehicles, Number_of_Casualtis,etc. These can be analysed further to obtain important information like the time of the day when accidents are more likely to happen in certain areas, whether the police force is being involved in the matters as required or not.

An attribute, Accident_Severity has been divided in classes [1,2,3] where 1 is the least severity. The following plot is helpful in determining areas where more serious accidents have happened

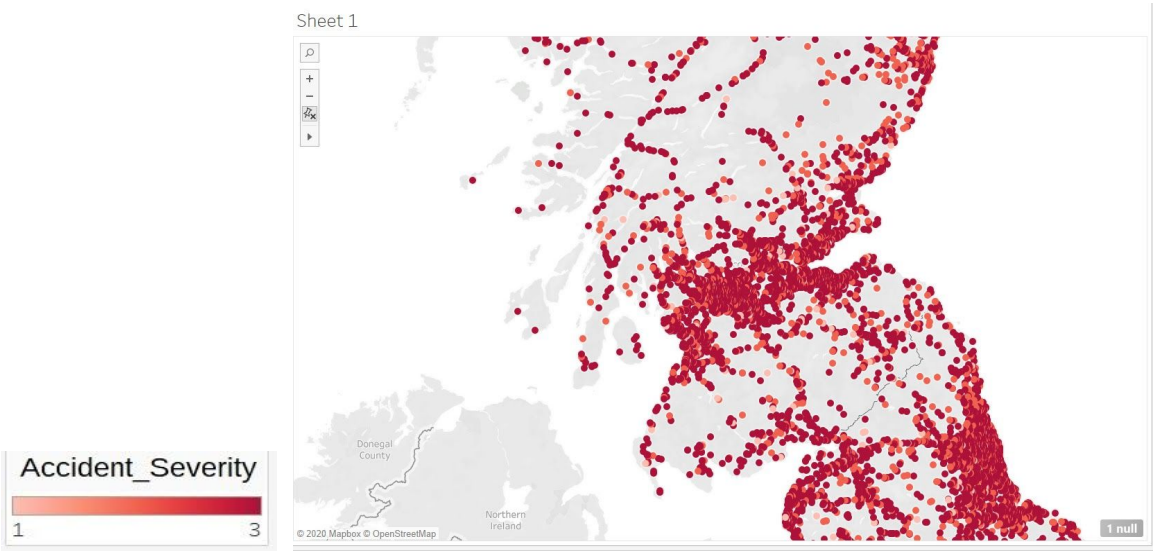



**Fig 5. Map depicting accident severity of dataset points**

- **DBSCAN Algorithm**:

DBSCAN is the first density based clustering algorithm, proposed in 1996 by Ester et al. It was designed to group data of irregular and non geometric shapes in the presence of noise in spatial and non-spatial high dimensional databases.Density-based spatial clustering of applications with noise (DBSCAN) is one of the well-known data clustering algorithms that is commonly used in data mining and machine learning for a given set of points distributed in space DBSCAN groups together points that are close to each. This grouping is done on the basis of two parameters - a distance parameter ε and a minimum number of points ***n***. For the points that are less densely placed, it marks them as outliers.

Parameters DBSCAN (minPoints ***n*** ,epsilon ε )

Epsilon ε : This parameter mentions the distance between two points within which they can be considered in a cluster. It means that if the distance between two points is less than or equal to this value (eps), these points are considered in the same cluster.

minPoints ***n*** : This parameter mentions the minimum number of points required to be in a group to form a cluster.

For the points that are not included in any cluster, they are considered noise or outliers

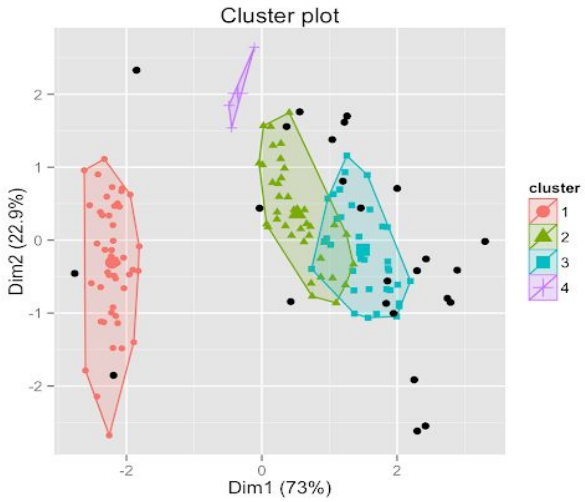



**Fig 6. Sample data showing four clusters and background noise [17]**

**Table 12. Pseudocode of DBSCAN**

| **DBSCAN Pseudocode** |
|---|
| Require:<br>• Dataset D<br>• DBSCAN parameters (ϵ,minPts )<br><br>1. DBSCAN(D, eps, MinPts)<br>2. C = 0<br>3. for each unvisited point P in dataset D<br>4. mark P as visited<br>5. NeighborPts = regionQuery(P, eps)<br>6. if sizeof(NeighborPts) < MinPts<br>7. mark P as NOISE<br>8. else<br>9. C = next cluster<br>10. expandCluster(P, NeighborPts, C, eps, MinPts)<br><br>1. expandCluster(P, NeighborPts, C, eps, MinPts)<br>2. add P to cluster C<br>3. for each point P' in NeighborPts<br>4. if P' is not visited<br>5. mark P' as visited<br>6. NeighborPts' = regionQuery(P', eps)<br>7. if sizeof(NeighborPts') >= MinPts<br>8. NeighborPts = NeighborPts joined with NeighborPts'<br>9. if P' is not a member of any cluster yet<br>10. add P' to cluster C |

1. regionQuery(P, eps)
2. return all points within P's eps-neighborhood (including P)

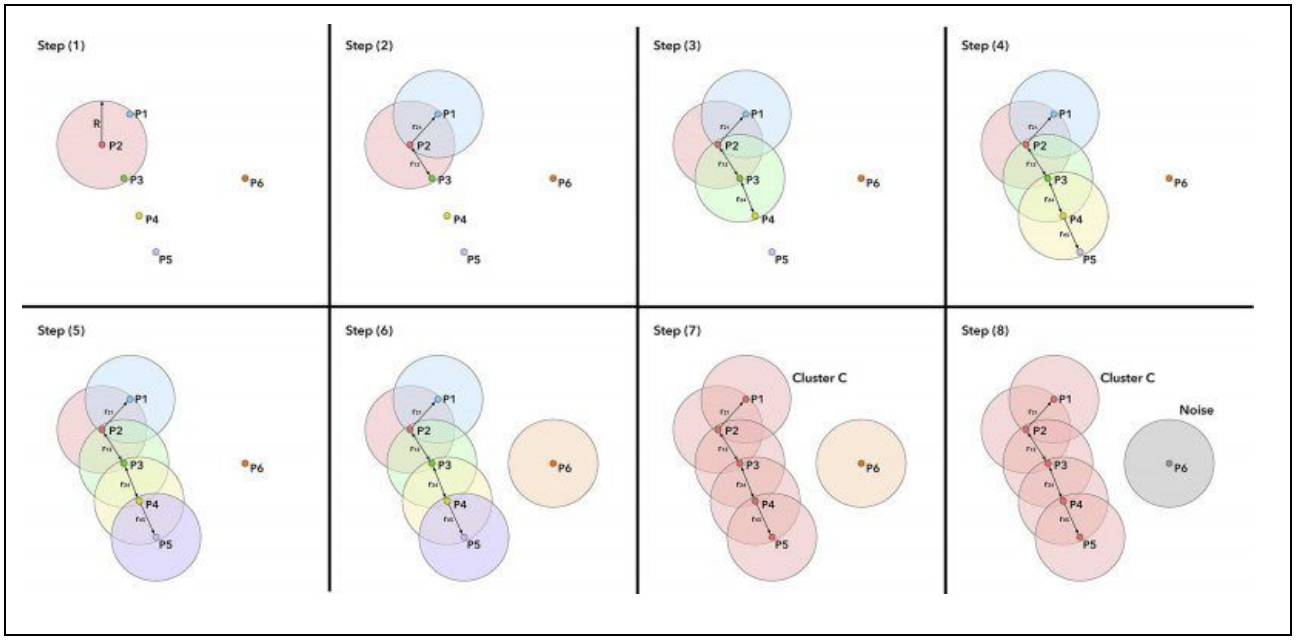



**Fig 7 . Illustration of clustering of points using DBSCAN [18]**

Parameter Values:

Setting parameter values is the most important step in every data mining task.

eps(ε): if the eps value selected is too small, a large part of the data will not be clustered. It will be considered outliers because it would not attain the number of points to create a dense area. Whereas if the chosen value is too high, clusters will group together and the maximum points will be in the same cluster. The eps should be selected based on the distance of the dataset.

*minPoints*: The minimum value considered for the minPoints must be 3, but it greatly depends on the dataset. The larger points in the data set, the larger the minPoints value must be chosen

Value of the epsilon chosen in our project is 50 metres and minPoints is 10. It means that all accident spots within 50 mtr of each other and with 10 neighbourhoods are grouped into a cluster. The value of epsilon is converted to radians and then used for clustering. We use

Haversine distance between the points for clustering which gives accurate distance between spatial data points.

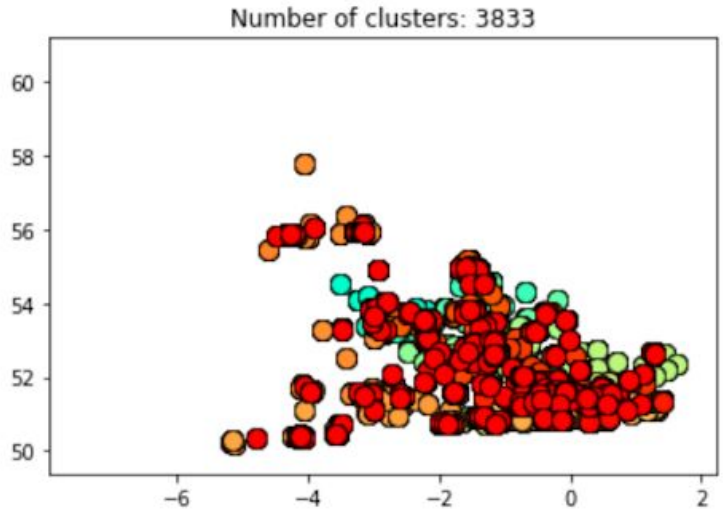



**Fig 8. Clusters formed by performing DBSCAN on dataset**

Advantages

- Resistant to Noise
- Can handle clusters of different shapes and sizes
- Efficient on large data sets.
- Minimal requirements of the domain knowledge to determine the values of its input parameters, which is very important problem especially for large data sets

Disadvantages

- DBCAN is not well with handling data of varying densities
- Sensitive to parameters—hard to evaluate the appropriate set of parameters
- May form chance clusters

**OPTICS Algorithm**

OPTICS stands for Ordering Points To Identify Cluster Structure. It is inspired by the DBSCAN clustering algorithm, yet it differs from other clustering algorithms as it does not produce a clustering of a data set explicitly; but instead creates an ordering of the points in the database representing the structure of its density-based clustering . This ordering can be used to analyse

the data as well as obtain clusters.Two concepts form the basis of OPTICS clustering- core distance and reachability distance. This algorithm solves one of DBSCAN's major issues that is detecting clusters in data of varying density. The points of the database are arranged in such a manner that spatially nearest points become neighbors in the ordering. Hence the output of OPTICS is an order of the points in a particular ordering arranged according to their reachability distance. This is represented as a dendrogram as a reachability plot.

**Table 13. Pseudocode of OPTICS**

**OPTICS Pseudocode**

Require:
- Dataset D of size n
- DBSCAN parameters (ϵ,minPts )

Output: Cluster ordering CO

```
1.  procedure OPTICS (X, E, MinPts)
2.  for each unprocessed point x ∈ X
3.  expandCluster(x, , E MinPts, X, CO)
4.  write noise in CO;

1.  procedure expandCluster(x, E, MinPts, X, CO)
2.  compute neighbourhood and core distance for x
3.  if x is core then processPoint(x, PQ, CO)   // PQ is a priority queue
4.  while PQ is not empty
5.  dequeue a point q from PQ compute neighbourhood and core distance for q
6.  if q is core then processPoint(q, PQ, CO)

1.  procedure processPoint(x, PQ, CO)
2.  write x in CO
3.  if x is core then updateNbhd(PQ, x)
4.  mark x as processed

1.  procedure updateNbhd(PQ, x)
2.  for each q ∈ N (x) E
3.  if q is not processed dist ← max(x.coreDist, distance(q, x))
4.  if q.reachDist is UNDEFINED
5.  q.reachDist ← dist enqueue q in PQ
6.  else if q.reachDist > dist
7.  q.reachDist ← dist update q.reachDist in PQ
```

Parameters for OPTICS:

OPTICS requires the same parameters as DBSCAN epsilon and minpts. However it introduces two new concepts in the algorithm.

Core Distance:

It is the minimum value of distance required to call a given point as a core point. Core Distance is undefined if the given point is not a Core point.

Reachability Distance:

The Reachability distance between a point p and q is the larger of the Core Distance of p and the Distance between p and q. If q is not a Core point then the Reachability Distance is not defined.

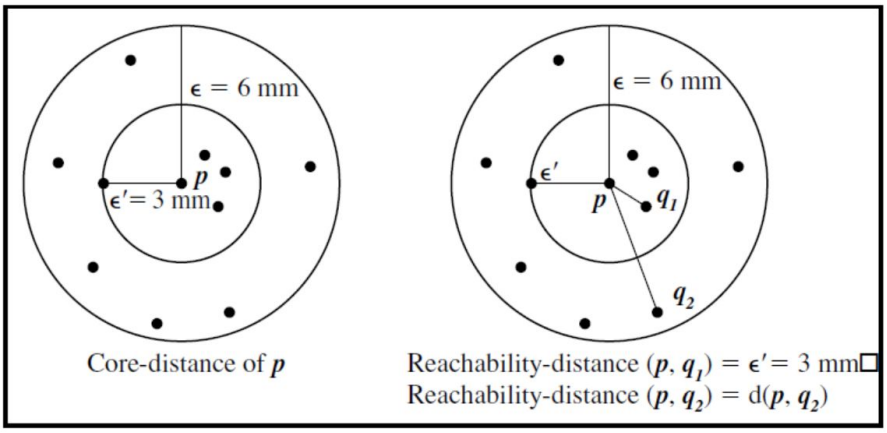



**Fig 9. Figurative representation of Core Distance and Reachability Distance [19]**

For an insufficiently dense cluster, ie., if a point is not a core point, both core-distance and reachability-distance are undefined. In case of a large $\varepsilon$, this never happens, but then every ε-neighborhood query returns all the points of the dataset. Hence, the epsilon ε parameter is needed to discard the density of clusters that are larger for our interest, and to speed up the algorithm.

A reachability-plot which is a special kind of dendrogram can be obtained by plotting the ordering of the points as given by OPTICS on the x-axis and the reachability distance on the y-axis. The clusters can be identified as valleys in the reachability plot. Identifying clusters from

this plot can be done manually or by various algorithms that try to detect the valleys by steepness, knee detection, or local maxima.

This technique is unique from other clustering techniques as it does not explicitly separate the data into clusters but produces a visualization of Reachability distances and uses this visualization to cluster the data.

Advantages

- Does not need to tune the epsilon parameter and is only used to reduce the time taken. This lessens the time of the analytical process of parameter tuning.
- Relatively unaffected by parameter settings.

Disadvantages

- Requires greater memory requirement as it maintains a priority queue to store order of points.
- Requires more computational power because the nearest neighbour queries are more complex than radius queries in DBSCAN.
- Does not classify the given data into clusters. It only produces a Reachability distance plot and rest is left upon the user interpretation to cluster the points accordingly.

**Statistical Significance Testing**

One drawback of DBSCAN is that it does not take into consideration the statistical significance of identified clusters, which may include chance patterns in the output. Although DBSCAN has a default handling of noise in the data, in a random point distribution (i.e., non clustered regions), points that appear next to each other may be considered as valid points. Tiny clusters of points may be greatly similar to each other just by chance. We do not want any spurious clusters of points. Marking a region as a crime-affected by mistake can greatly decrease the number of people visiting the region, lowering property values and damaging local businesses. A solution to this problem is significant testing.

For attaining solution quality we make use of statistical significance.

For a given N point distribution, the spatial statistic finds a list of candidate regions using a score calculated using a test statistic (e.g., density, likelihood ratio)[4]. The scores represent the

trueness of the clusters. Then statistical significance testing is used to confirm if the score of a candidate is high enough to be a hotspot.

The test statistic is important as it is used to score and rank candidates, and directly determines result correctness. Currently algebraic or exponential functions are used (e.g., density, density ratio and likelihood ratio) as the test statistic. In order to make the scores of clusters comparable indifferent areas and count comparable, normalisation is used. When these density based test statistics are compared, these normalizations are found to be biased towards smaller candidate regions [5]. Density cannot detect hotspots with an area greater than N/d.[1,5] Also, these density based test statistics require the area of the cluster formed which is not feasible in case of DBSCAN as it forms irregular shapes whose area cannot be easily determined. So we use the size of clusters as a test statistic [1]. We find a threshold for cluster size $n\alpha$ using the significance level $\alpha$ to identify the groups as significant ($n > n\alpha$ ) and non-significant ($n \leq n\alpha$ ) using the MOnte Carlo Evaluation as explained ahead.

**Monte Carlo Evaluation**

For our problem, we define the probability of having a cluster of threshold size $n\alpha$ or greater in a N-point homogeneous point distribution in spatial domain S as $p_{null}$ which is also the p-value. This $n\alpha$ is the cluster size which determines the significance of the cluster.

[1]Cluster C is statistically significant if its p-value,

$$\mathbf{p_{null}\ (nC, N, S, \epsilon, minPts\ ) < \alpha}$$

Here , $\alpha$=>the significance level (e.g., 0.01, 0.05)

nC=> the sizeof a detected clusterC ,

N=> total number of points in the point distribution ,

S=>The spatial domain of the point distribution

There are many statistical models such as Non Deterministic Normalization based Scan Statistic but they cannot calculate the probability pnull in closed-form to directly compute or evaluate the statistical distribution of the key parameter $n\alpha$ as well as the randomness associated with distributing N points in spatial domain S which can have irregular shape. Thus, we use a Monte Carlo method to estimate $p_{null}$ .

In Monte Carlo estimation we run M simulation trials. to approximate the distribution of cluster size n (i.e., the test statistic) in point distributions generated by a homogeneous point process. In each trial, a random N point distribution is generated using a homogeneous point process, and then DBSCAN with the input (ϵ,minPts ) is run to get the best cluster size ñ in the trial.we get M best ñ values from the after M trials . These M values are then sorted in descending order and the p-value/$p_{null}$ of a cluster C detected from the real data is estimated by checking its rank r in the sorted list:

$$\mathbf{p_{null}(nC, N, S, \epsilon, minPts) = r/M.}$$

M has to be at least 1/α to evaluate the significance. [1]

These best cluster sizes are the scores on the basis of which we evaluate a cluster's significance. The best scores from all trials are recorded. Now for each point if the score falls among the top α*100% (say 1%), corresponding to the significance level of α (=0.01) of the recorded best scores, we classify it as a significant cluster and identify it as a hotspot. The non-significant groups ie., non hotspot points are filtered out by using the minimum of the top 1% scores as a threshold. In other words it can be said that we just need to find the (αM)th largest value in the sorted list and use that as a threshold (denoted as nα ) of cluster size to filter out non-significant clusters.

In this way we identify a cluster as significant if its $p_{null}$ is less than the predefined significance level α where $p_{null}$ is defined as the probability of having a cluster of size nα or greater in a N-point random distribution in the spatial domain of data.

**Table 14. Pseudocode of MonteCarlo**

| **Monte Carlo Evaluation Pseudocode[1]** |
|---|
| Require:<br>• total number of points N and spatial domain S<br>• DBSCAN parameters (ϵ,minPts )<br>• significance level α and number of Monte Carlo trials M<br>1: assert(M ≥ 1/α)<br>2: nList = new List(M)<br>3: for i = 1 to M do<br>4: datar = getRandomPointDistribution(N, S)<br>5: clusters = DBSCAN(datar , ϵ,minPts) |

```
6: nList (i ) = max(clusters.getSizes())
7: end for
8: nlist = nlist.sort('DESC')
9: return nα = nList (ceil (α · M))
```

Parameters in Monte Carlo Estimation

- M- Number of trails to be run
- α- significance level which determines if the cluster is significant or not.

Note that M has to be at least 1/α to evaluate the significance.

We have used α as 0.02 and M as 1000 keeping in mind the above said note. Large value of M helps increase the correctness in determining nα value whereas a small value of α decreases the variation in recorded values of best ñ to find nα. Both these factors help us in finding the best possible results for hotspot filtering.

**Plotting points on maps**

We used Geopandas library to generate the map of the UK whose data we have and Matplotlib library to plot the cluster points in the map. We used the shape of each cluster's cloud to draw a boundary around it, a geofence. There are many ways to draw this for example drawing a convex hull or even a concave hull. But our clusters may be irregular that may not be covered by these shapes. So we use a simple approach of drawing a circle around each point and merging them all together. First all location points are enlarged into circles of a given radius, and then all circles are merged together into a single boundary as shown;

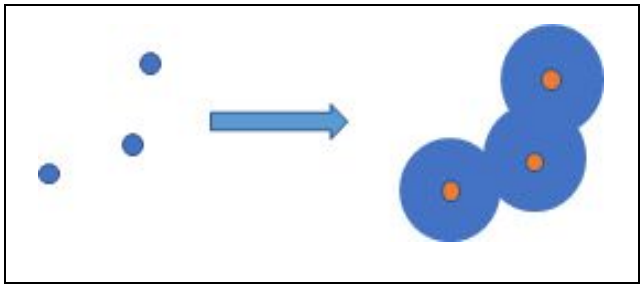


**Fig.10 Enlarging location points into circles and merging them**

To make the map nice and interactive we used a Leaflet map so we can pan and zoom and actually see the street name. We used the mplleaflet library for this.

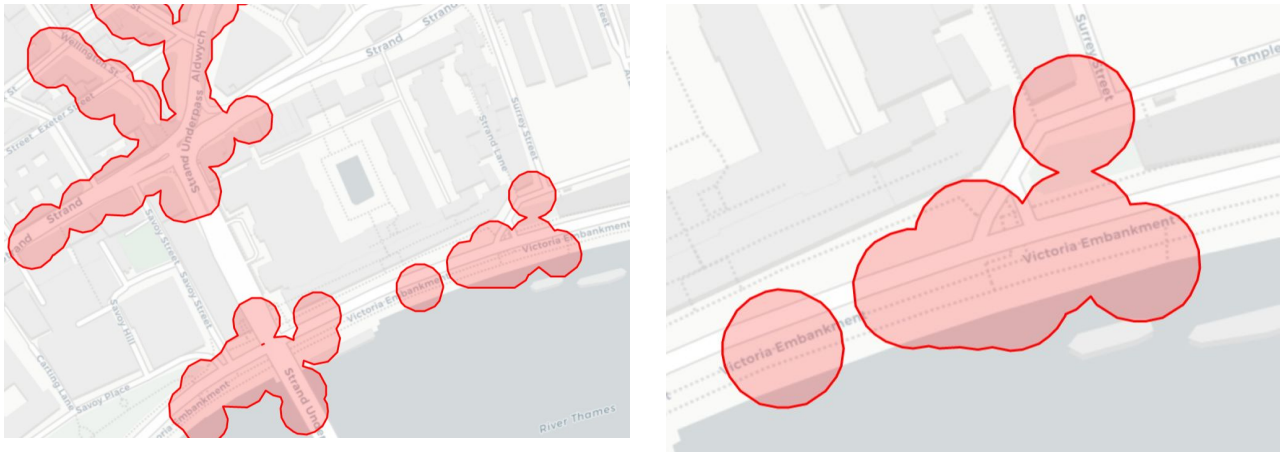



**Fig 11 . Enlarged clustered data points with merged boundaries**

**Comparison of OPTICS and DBSCAN**

It is easy to evaluate a clustering technique and compare it with another when the true labels or classification is available beforehand. But how to compare when there are no correct labels with us. Difficulty to evaluate its performance is one of the primary disadvantages of any clustering technique. To handle this difficulty, the V-measure metric was developed. We compare the results obtained by the two algorithms using three metrics- homogeneity, completeness and V-measure. These are three key measures of the quality of a clustering technique.

The reason that we use these metrics is that they are not dependent on either the clustering algorithm or the data set and also give accurate comparison results. All existing evaluation methods such as entropy, f-measure, purity have these drawbacks. Since these measures classification of data into classes, we have taken accident severity as class labels. This is a data column available in our dataset for each accident with three classes- 1,2,3 where the severity increases from 1 to 3.

• Homogeneity

This measures how many data-points in a cluster belong to the same class label. A perfectly homogenous solution means there are the same number of data points and same class labels and each point is a cluster. It is given by the following formula

$$h = \begin{cases} 1 & \text{if } H(C,K) = 0 \\ 1 - \frac{H(C|K)}{H(C)} & \text{else} \end{cases} \quad (1)$$

where

$$H(C|K) = -\sum_{k=1}^{|K|}\sum_{c=1}^{|C|}\frac{a_{ck}}{N}\log\frac{a_{ck}}{\sum_{c=1}^{|C|}a_{ck}}$$

$$H(C) = -\sum_{c=1}^{|C|}\frac{\sum_{k=1}^{|K|}a_{ck}}{n}\log\frac{\sum_{k=1}^{|K|}a_{ck}}{n}$$

.[15]

Where N - number of data samples,

C -different class labels,

K- no. of clusters and

$a_{ck}$ - number of data-points belonging to the class c and cluster k.

• Completeness

This measures the ratio of the points of a given class that is assigned to the same cluster. Completeness is maximum when all data points are in the same cluster.

$$c = \begin{cases} 1 & \text{if } H(K,C) = 0 \\ 1 - \frac{H(K|C)}{H(K)} & \text{else} \end{cases} \quad (2)$$

where

$$H(K|C) = -\sum_{c=1}^{|C|}\sum_{k=1}^{|K|}\frac{a_{ck}}{N}\log\frac{a_{ck}}{\sum_{k=1}^{|K|}a_{ck}}$$

$$H(K) = -\sum_{k=1}^{|K|}\frac{\sum_{c=1}^{|C|}a_{ck}}{n}\log\frac{\sum_{c=1}^{|C|}a_{ck}}{n}$$

[15]

• V-measure

It is the harmonic mean of the above stated metrics. The 'V' stands for "validity", used to represent the correctness of a clustering solution.

$$V_\beta = \frac{(1+\beta)*h*c}{(\beta*h)+c} \quad [15]$$

All these measures fall in the range 0.0 to 1.0 where 0 is the worst score and 1 is the best score. Hence they are easy to understand and interpret the clustering algorithm's performance.

```
print("Homogeneity: %0.3f" % metrics.homogeneity_score(labels_true, labels))
print("Completeness: %0.3f" % metrics.completeness_score(labels_true, labels))
print("V-measure: %0.3f" % metrics.v_measure_score(labels_true, labels))

Homogeneity: 0.017
Completeness: 0.006
V-measure: 0.009
```

**Fig 12. Evaluated metrics for DBSCAN**

```
print("Homogeneity: %0.3f" % metrics.homogeneity_score(labels_true, labels))
print("Completeness: %0.3f" % metrics.completeness_score(labels_true, labels))
print("V-measure: %0.3f" % metrics.v_measure_score(labels_true, labels))

Homogeneity: 0.075
Completeness: 0.006
V-measure: 0.011
```

**Fig 13. Evaluated metrics for OPTICS**

As observed here, the metric results of OPTICS are preferable than that of DBSCAN.

Advantages

- Bounded scores: between 0 and 1
- Intuitive interpretation: clustering with unsatisfactory V-measure can be analyzed in terms of homogeneity and completeness to determine what type of mistakes is done by the assignment.
- Cluster shape or structure is not considered
- independent of the clustering algorithm, size of the data set, number of classes and number of clusters.

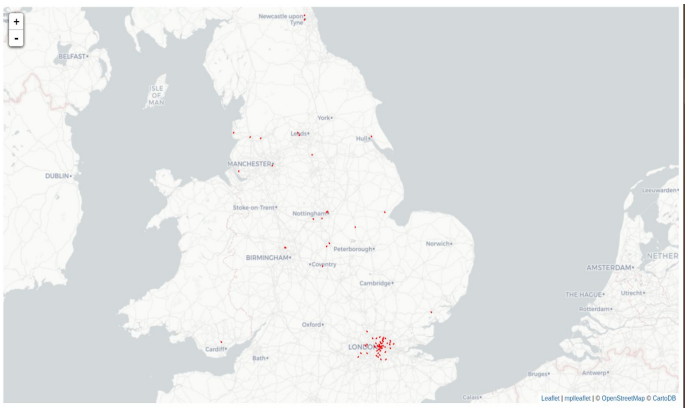
Disadvantages

The knowledge of the ground true labels is required for the working of this metric which might not be available or may require assigning them manually.

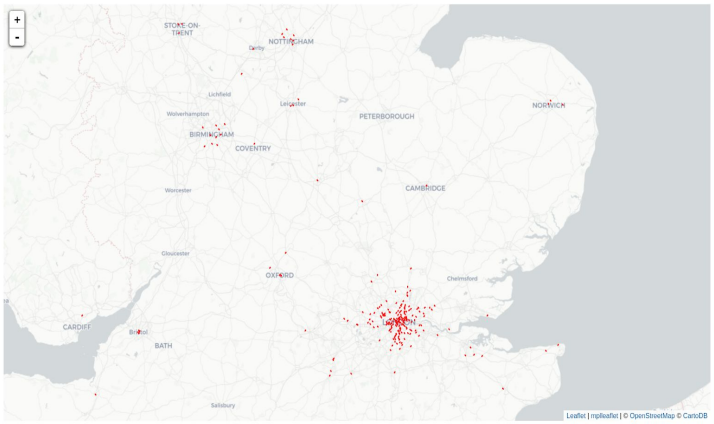
### 3.4 FINAL OUTPUT

**Fig 14. Zooming in into the map**

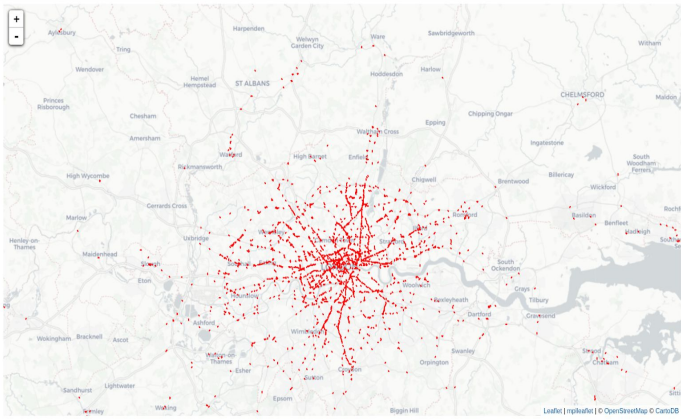

a. Preview of initial output map



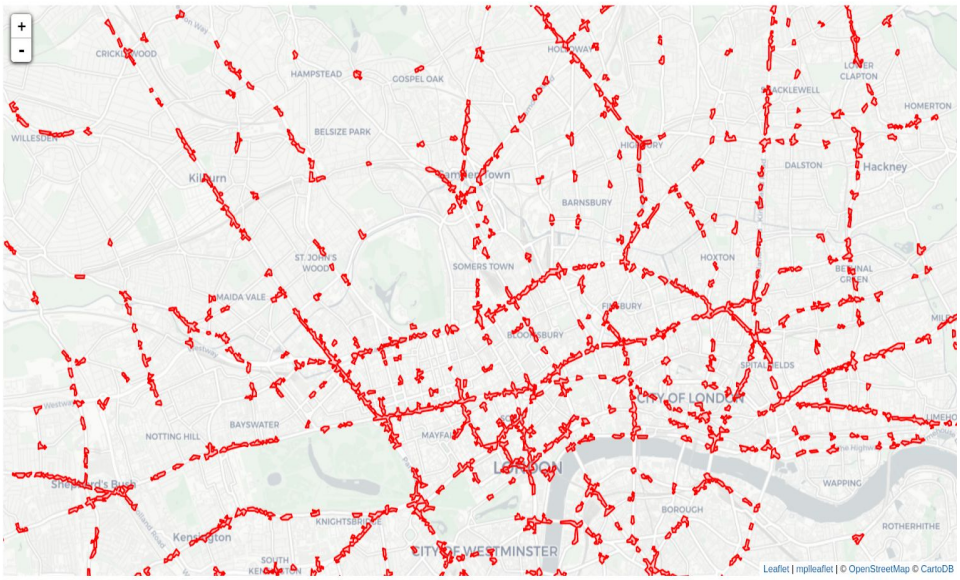

b. Zooming into London




c. Zooming in further to see clusters




d. Zooming further




e. Clusters (hotspots) can be seen formed on the roads of the city

## DBSCAN CLUSTER MAPS


**Fig 15. Clusters formed using DBSCAN**

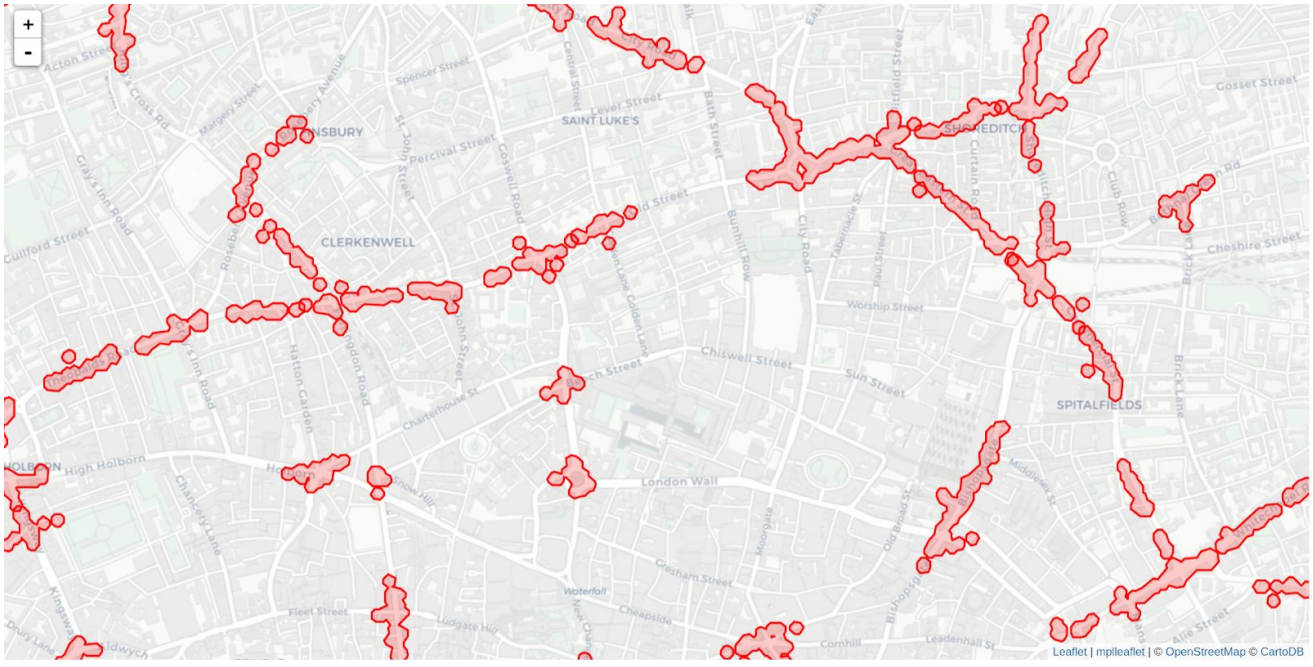



**Fig 16. Clusters formed after significance testing i.e. Significant clusters**

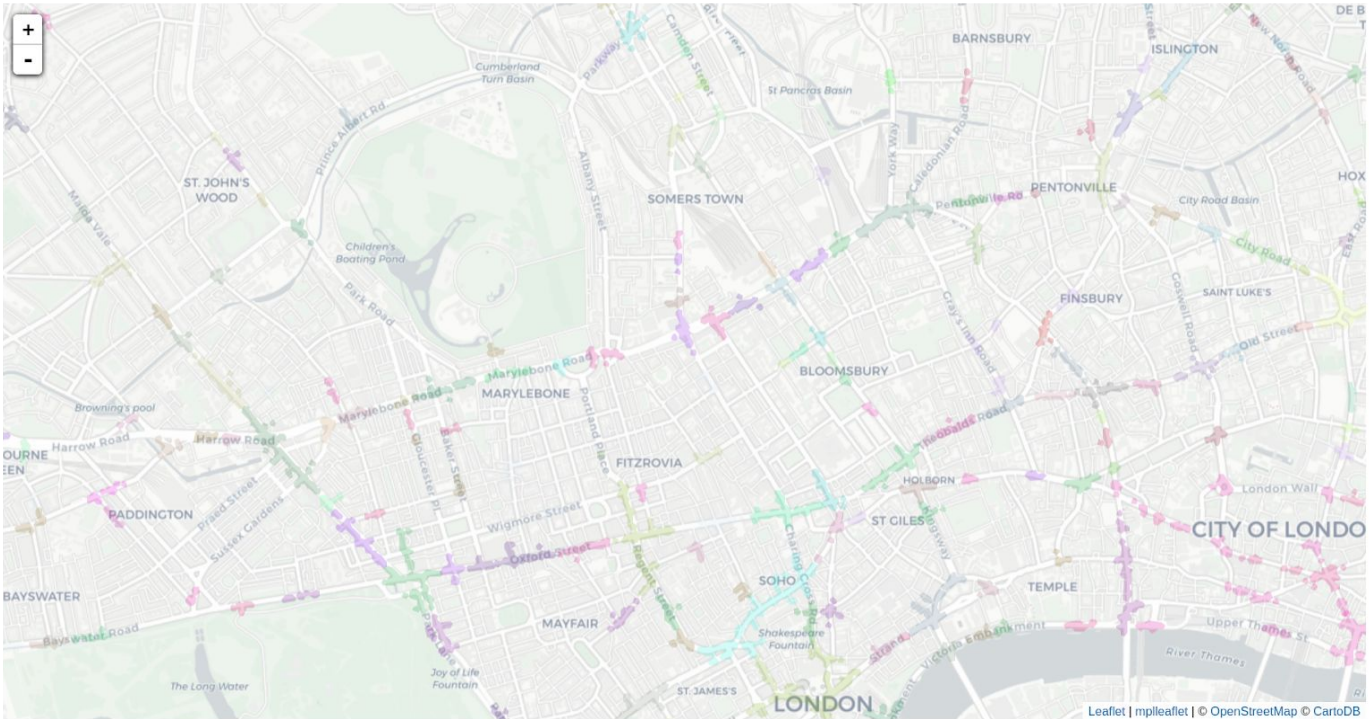



**Fig 17. Clusters(colored) formed using DBSCAN**

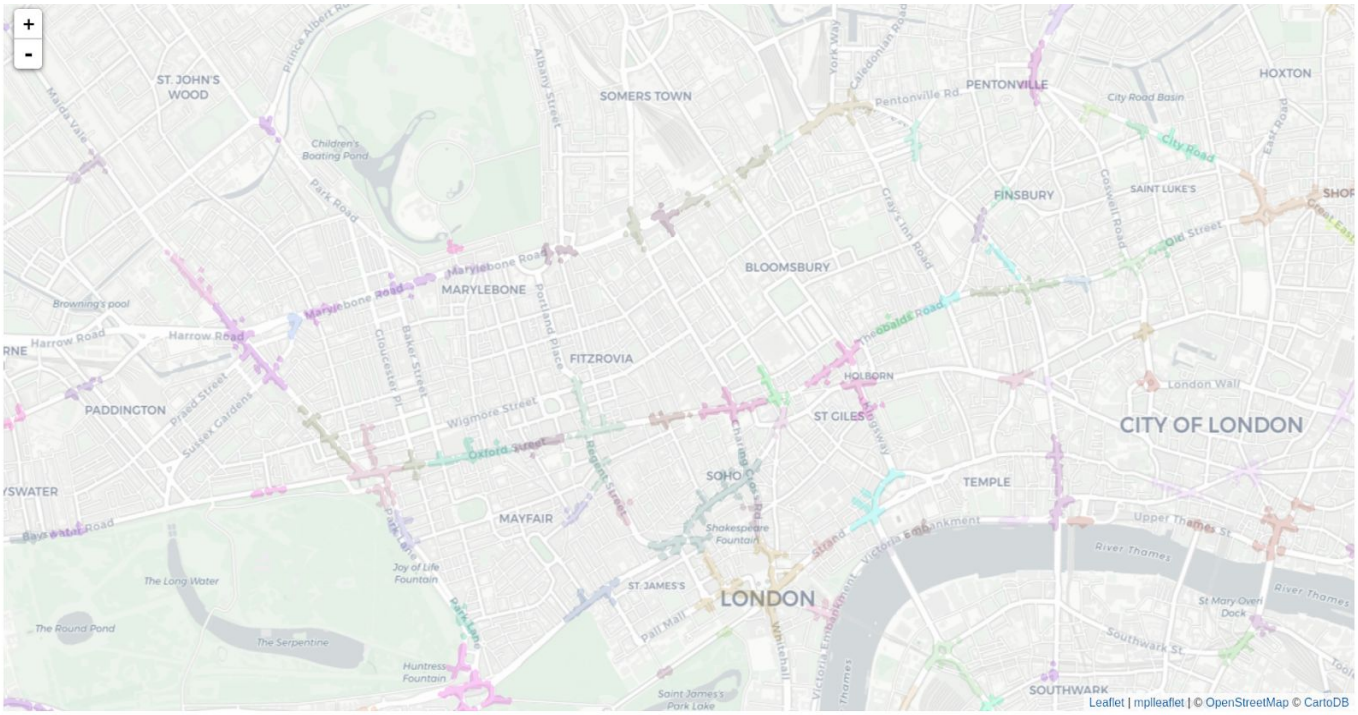



**Fig 18. Clusters formed after significance testing i.e. Significant clusters**

## OPTICS CLUSTER MAPS

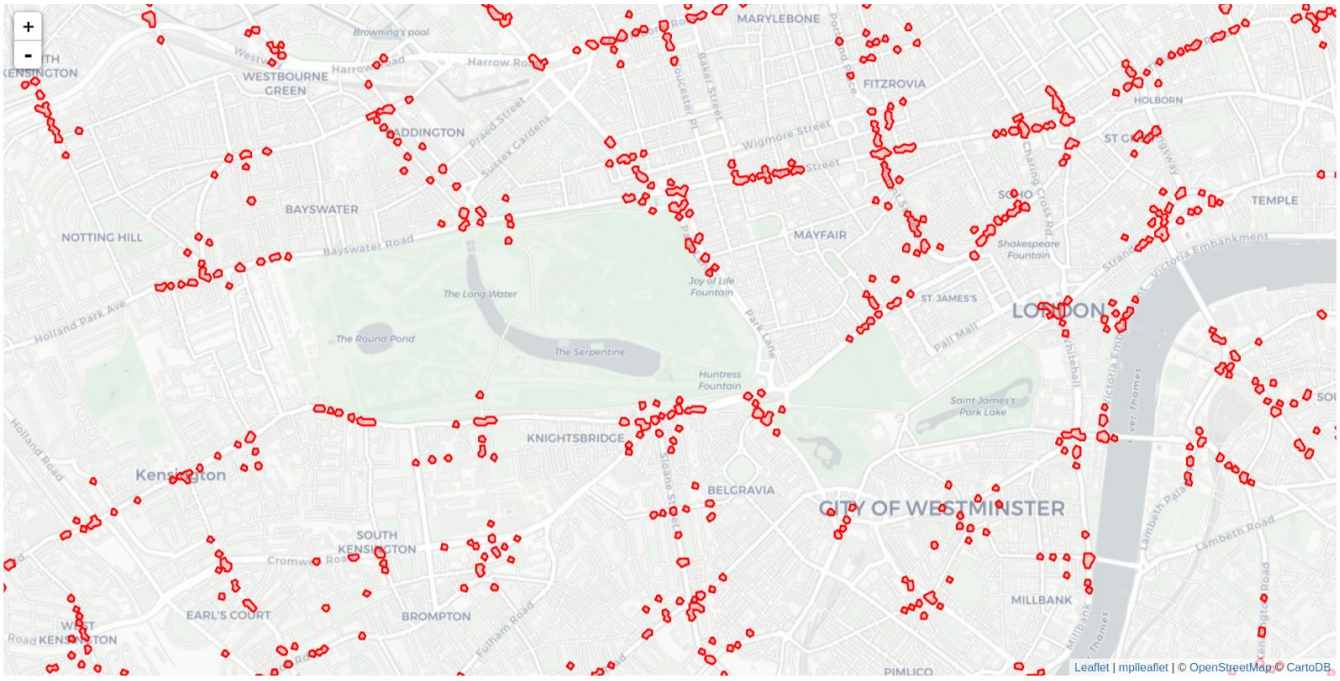



**Fig 19. Clusters formed using OPTICS**

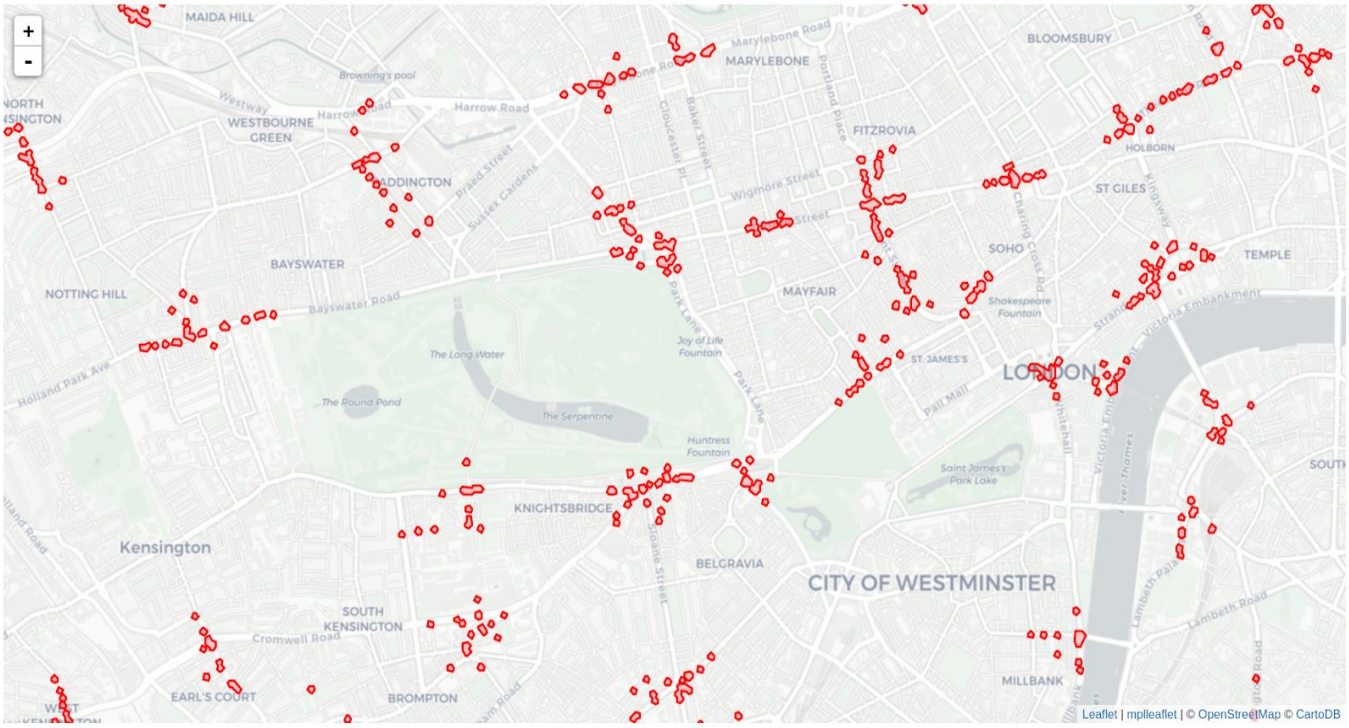



**Fig 20. Clusters formed after significance testing  i.e. Significant clusters**

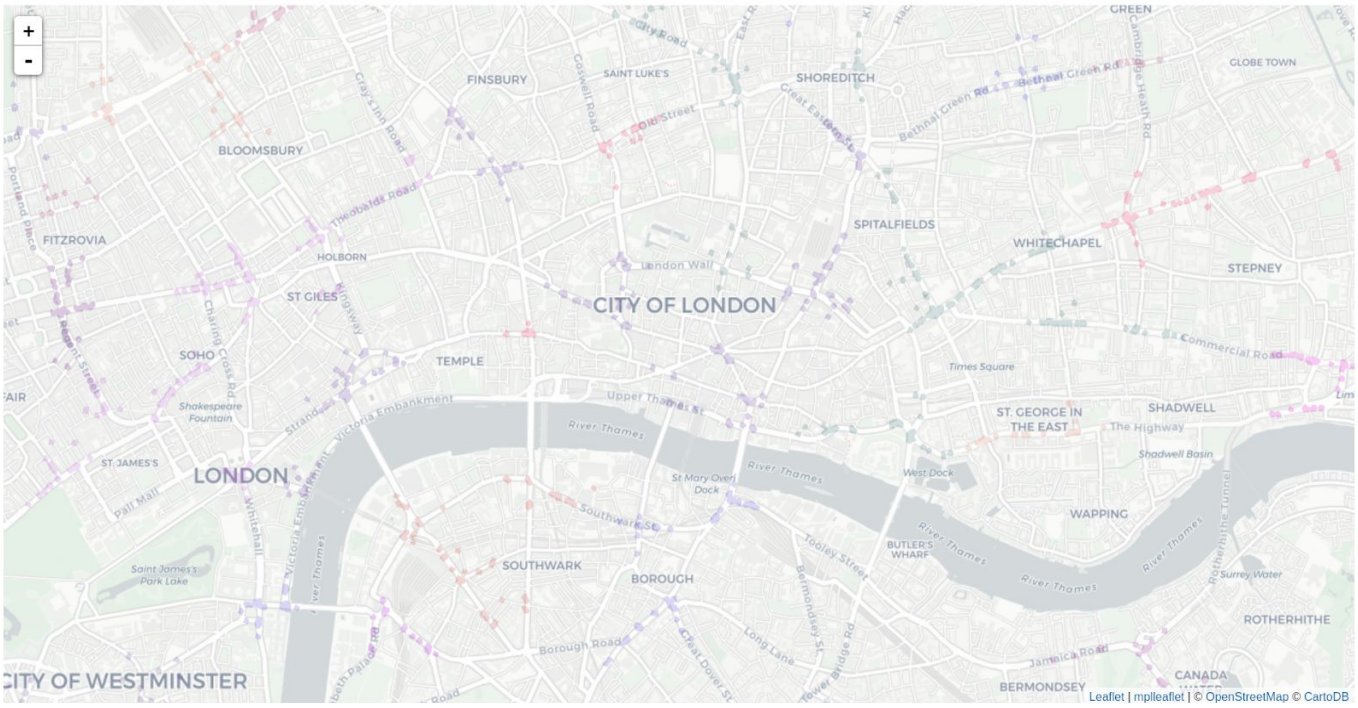



**Fig 21 . Clusters(colored) formed using OPTICS**

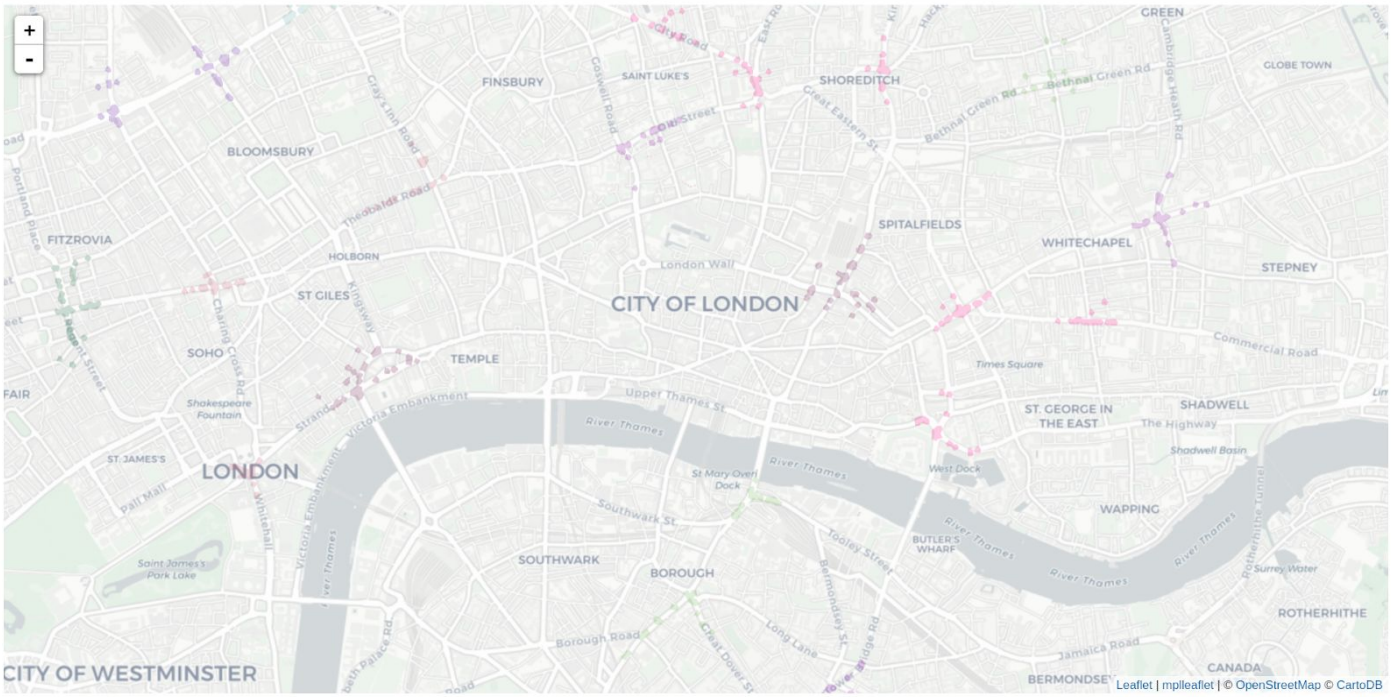



**Fig 22. Clusters formed after significance testing i.e. Significant clusters**


**Fig 23(a) DBSCAN Clustering**

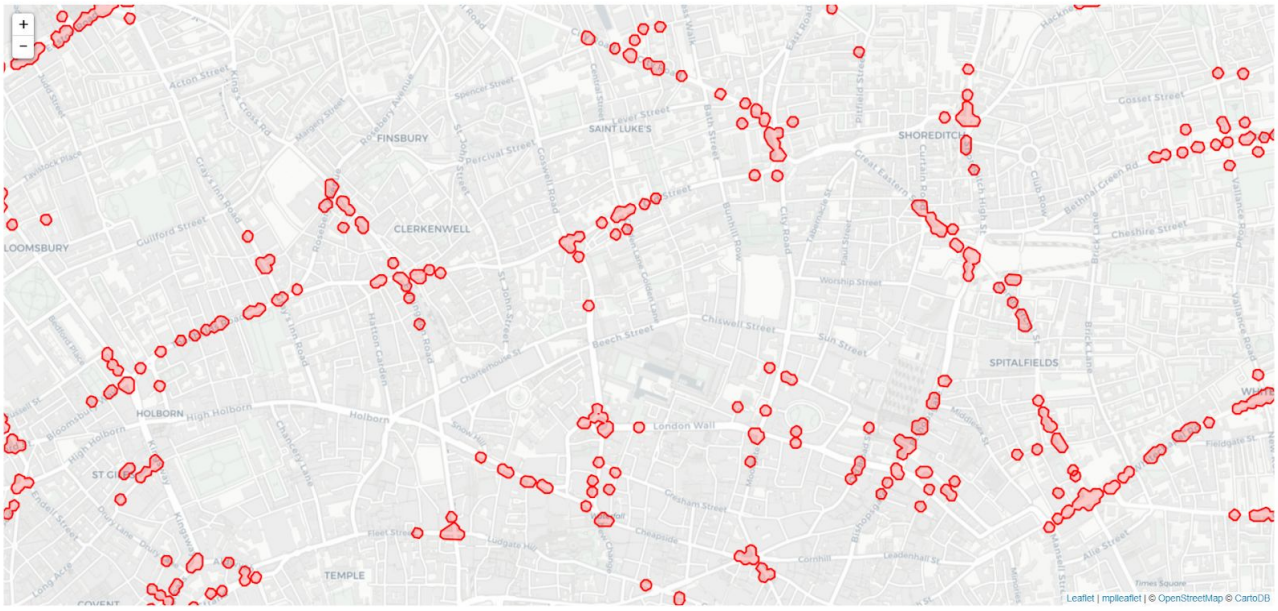



**Fig 23(b) OPTICS Clustering**

**Fig 23. Clusters obtained after DBSCAN vs OPTICS.**

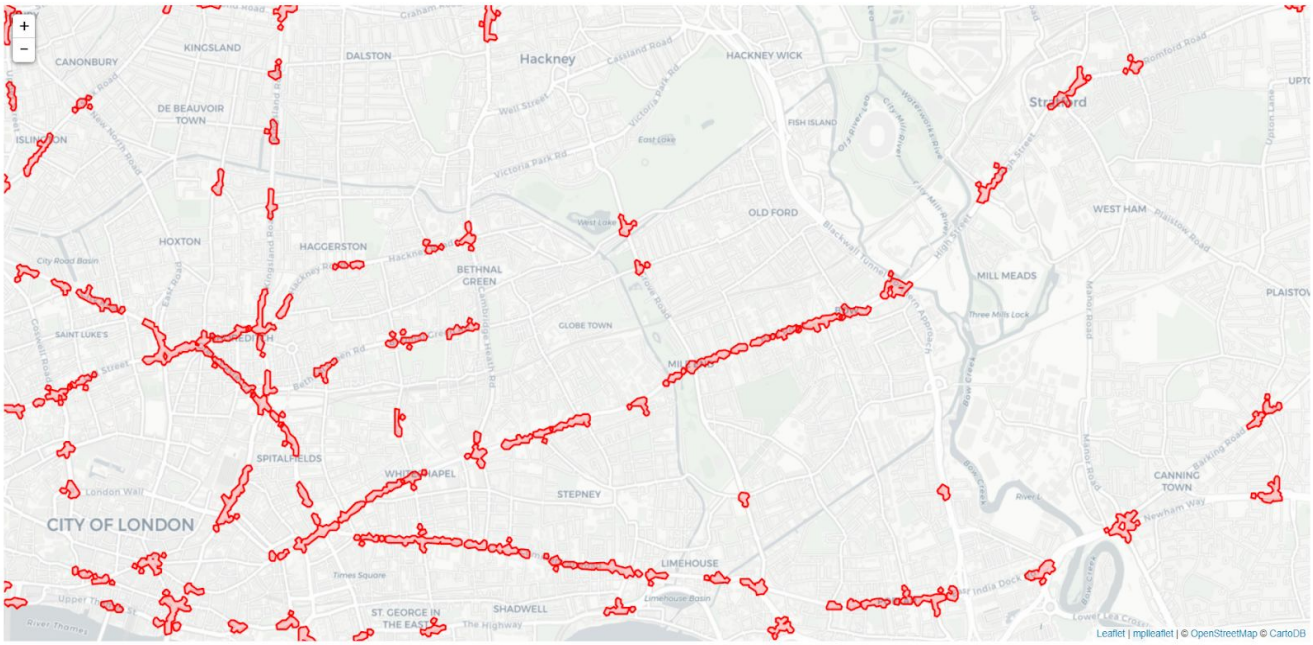



**Fig 24(a) DBSCAN Clusters after Significant Testing**

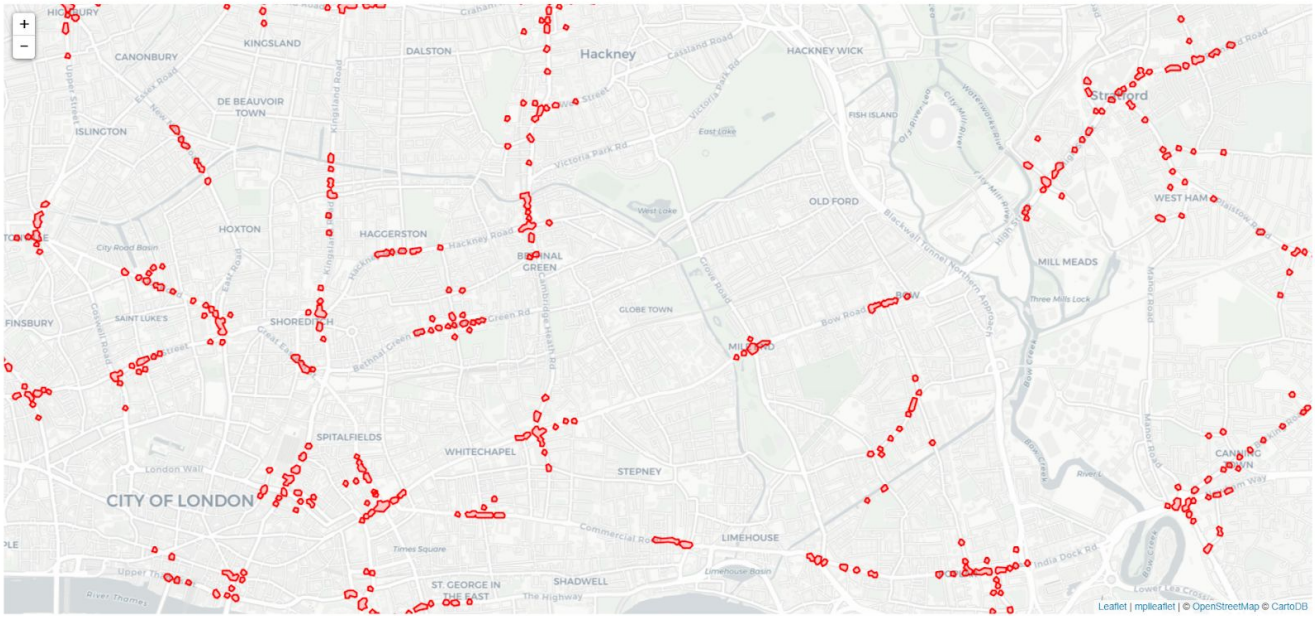



**Fig 24(b) OPTICS Clusters after Significant Testing**

**Fig 24. Significant clusters obtained after DBSCAN vs OPTICS**

# CHAPTER 4. MODELLING AND IMPLEMENTATION DETAILS

## 4.1 IMPLEMENTATION DETAILS AND ISSUES

We implemented this solution approach on python. The reason why we choose Python over other programming languages for our project is because of the following advantages it provides:

1. Presence of Third Party Modules
2. Extensive Support Libraries
3. Open Source and Community Development
4. Learning Ease and Support Available
5. User-friendly inbuilt Data Structures
6. Productivity and Speed

Following are the python libraries/packages that have been used in our code for various tasks:

- NumPy: NumPy is extremely fast, used to process multidimensional data in Python designed specifically for scientific computing. Apart from amazing speed, NumPy easily interacts with other libraries. NumPy can exchange data with GDAL, Shapely and many other scientific computing Python libraries in other fields.
- Pandas: This package is used to handle the csv data by converting into spatial data frames which in turn helps us to use the information. It provides data structures and data analysis tools.

Numpy and Pandas are one of the most integral libraries of python, functionalities of which we used in various parts of our code, be it clustering or map plotting.

- Geopandas : iI makes working with geospatial data easy in python. Also it is used to support geographic data in pandas objects
- Shapely: It is used for various geometric objects and operations such as point, line and polygon creation.
- Pyproj: this allows transformation between any two coordinate systems
- Mplleaflet: Used to create actual real time map on which the clustered points are plotted

These libraries are used for geographical representation of data. Here we have used them to plot the resultant maps of data and clusters.

- Sklearn: It is used to provide various features like classification, regression and clustering algorithms.

Sklearn makes it much easier to implement clustering and data analysis on vector arrays or distance matrix in Python by providing predefined functionalities like sklearn.cluster.DBSCAN and sklearn.cluster OPTICS

- Tableau:It is a data visualization tool that helps in simplifying raw data into an understandable format.We have used it in this project to plot the maps.

### 4.2 RISK ANALYSIS

| Risk ID | Classification | Description of Risk | Risk Area | Probability | Impact | RE(P*I) |
|---|---|---|---|---|---|---|
| 1 | Development Environment | Familiarity | Team members had prior knowledge of Python but had to learn how to use Jupyter Notebook and use of maps in Python | .5 | 5 | 2.5 |
| 2 | Design | Testability | Tested only individual functions of the code | .1 | 3 | .3 |
| 3 | Design | Hardware Constraints | Processor speed and efficiency | .05 | 5 | .25 |

**Table 15. Risk analysis**

## CHAPTER 5. TESTING

### 5.1 TESTING PLAN

**Table 16. Testing**

| Type of Test | Will test be performed | Comments/Explanation | Software Components |
|---|---|---|---|
| Requirements Testing | Yes | Will be done to verify whether all requirements are being fulfilled. | The entire project to be tested. |
| Unit | Yes | To be done to confirm the proper functioning of individual modules. | The entire project to be tested. |
| Integration | Yes | To check upon completion whether all modules are interacting with each other as per requirements. | The entire project to be tested. |
| Compliance testing | No | | |
| Security Testing | No | | |

### 5.2 LIMITATIONS OF THE SOLUTION

Despite our efforts to choose the best and most simplistic algorithms for this project, there are some limitations which the algorithms fail to achieve,which need to be worked upon further. Some of them being :

- DBSCAN lags in dealing with variable densities of clusters. In this project we deal with a single (ε, minpoints). For varied density and different values of ε and minPoints, it might generate unwanted clusters or miss some required clusters. This is minimised by using the right ε and significance testing in our project.

- DBSCAN is fairly complex and time consuming. Even though for the flexibility and convenience it provides, it's a wise choice.
- Monte Carlo's dependence on M and α is very high. The wrong values result in wrong clusters.
- Monte Carlo has high computational requirements.
- OPTICS is computationally complex, it's time complexity being O(nlogn).
- VMeasure depends on the number of samples, clusters and ground truth classes.
- DBSCAN and the overall depends heavily on user inputs(parameters).

# CHAPTER 6. FINDINGS, CONCLUSION AND FUTURE WORK

## 6.1 FINDINGS

During the course of this project our knowledge increased greatly. We read several research papers apart from the ones mentioned in references.

We came to know about the various spatial clustering techniques like BIRCH, CURE, K-means clustering kernel and spatial statistical methods like density estimation, kriging estimation, principal component clustering and many others. We used those algorithms that we found computationally best and accurate that are DBSCAN, Monte Carlo and OPTICS.

DBSCAN can wrongly classify points that are close to each other into the same clusters. This classification of spurious clusters can be improved by adding statistical testing in the process of clustering. The baseline algorithm of Monte Carlo Evaluation can be used for this purpose. We use the cluster size as a test statistic during this process and not density or likelihood ratio because they are dependent on the area of the cluster formed for normalisation. The area of the cluster is hard to determine in DBSCAN clustering owing to the non-geometric shapes of clusters formed.

While our study we also came across the OPTICS algorithm for clustering. It is an advance in the DBSCAN algorithm and covers its flaw of not functioning properly in varied density datasets. OPTICS gives us an order of points which are close to each other and can be clustered together in the form of a reachability graph. This graph is then used to create clusters.

Both these methods have been compared using the entropy based metric - V-measure. It is calculated using homogeneity and completeness which are independent of the clustering method or size of the dataset. These metrics have bounded values and lie between 0 and 1. Hence they can be easily used for comparison. In our comparison we find that OPTICS yield better results than DBSCAN.

## 6.2 CONCLUSION

We realise that life is precious and we want to make our effort in saving it. The leading cause of deaths in the world is road accidents. We can decrease this by recognising the areas which have the most frequency of these mishaps. The classification of road accident hotspots in road safety still remains a major and yet neglected and underdeveloped issue.

Studies and methodologies have been put to use but mostly they are statistical methods that rely on raw statistics alone and do not take into consideration the potential indicators found in complementary datasets such as those referring to the environment, land use, accident victims etc., for statistical clustering.

We present a solution to locate high density/frequency accident hotspots which may consequently determine causal indicators more likely to be present at certain clusters, therefore being able to compare like with across time and space. By identifying such points we can take necessary precautions and safety measures at those places. The local governmental policy can focus on the traffic in that community and neighbourhood. Also, this might allow for a greater neighbourhood participation in understanding road user risk and make them more careful. For this purpose we apply spatial clustering on longitude and latitude data ,ie,. spatial data by use of the DBSCAN and OPTICS algorithm .

DBSCAN is a computationally efficient and easy to use algorithm. It is popularly used in the field of clustering and data mining. The main objective of DBSCAN is that the cardinality of the neighborhood has to exceed some threshold, that is, each point of a cluster the neighborhood of a given radius (Eps) has to contain at least a minimum number of objects (MinPts), only then it becomes a cluster. Else it is just noise. However, DBSCAN has its flaws and we will have to pay a heavy price in the real world for these flaws. DBSCAN can find arbitrary shaped clusters but it is unable to correctly classify those clusters which are very close (but not overlapping). Such a result in the real world can be very dangerous and harmful. So we bring into the picture statistical significance testing as proposed in [1]. We classify the clusters as significant or nonsignificant on the basis of a threshold cluster size $n\alpha$ determined by a significance level using

the MonteCarlo Estimation algorithm. This helps us clear out the false clusters and identify the true accident hotspots in the area that need to be taken care of.

To overcome and compare the problems faced by DBSCAN, we also apply the OPTICS algorithm to identify the hotspots. OPTICS may be considered as an advance in the DBSCAN algorithm. DBSCAN does not work very well for varied densities and is highly dependent on the parameters provided by the user. OPTICS eliminates these issues. Also it requires lesser parameters therefore decreasing the analytical process of tuning parameters. OPTICS is different from other clustering methods as it does not explicitly classify the points into clusters. It gives a plot of reachability distances and leaves it upon the discretion of the user to identify the clusters.

Further we analyze the results given by the two clustering methods using three parameters- homogeneity, completeness and V-measure. We find that according to these comparison metrics OPTICS give better results than DBSCAN. We observe a larger and lesser number of clusters in DBSCAN than its improved version.

Thus we tried to create a system that is efficient and fast in detecting accident hotspots and helps the community. We used what we thought worked best for us. There are a lot of other methods and algorithms that can be used and may give better results.

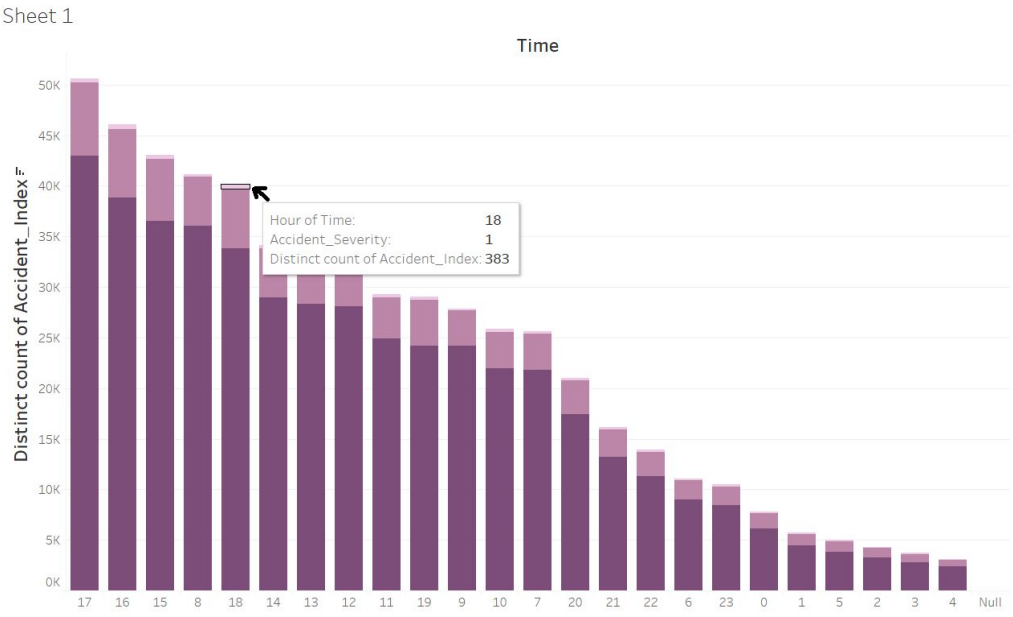
### 6.3 FUTURE WORK

To improve the computational efficiency of this method we can use the method of dual convergence as stated in [1]. The thought is to reduce the number of exact DBSCAN runs in the Monte Carlo trials by limiting the size $n_{max}$ of the largest cluster in a given data . Since DBSCAN clustering output in real data may be a combination of both significant and non-significant clusters, an acceleration has to consider both cases to be truly efficient. The Dual- convergence algorithm given in the paper plans to achieve this with: (1) an upper bound of $n_{max}$ to accelerate the validation of significant clusters (2) a lower bound of $n_{max}$ and an early-termination technique with a probabilistic performance guarantee to speed-up the filtering of non-significant clusters and a dual-convergence framework which makes the above techniques work together to maximize the speed-up. To sum up , the aim of the upper bound is to quicken the validation of

significant clusters and the aim of the lower bound and early-termination is to increase the speed of the filtering of nonsignificant ones.

Other improvements that can be implemented are as follows:

- Further advance the system by Identifying the contribution of crash influencing factors like weather conditions and traffic exposure in hotspot detection
- Incorporating multi-basis ranking methods to prioritize clusters.
- Influence of conditions of land, driver's condition,speed of vehicle ,alignment of road, shape of road ( curved, u turn) and weather conditions could be included in hotspot detection.
- Scrutiny in different areas such as straight roads, junctions, and parking lots could be carried out to find which type of analysis method is suitable for these areas.
- An application can be created that warns the drivers when entering a accident-prone area using these clusters we found above and use of GPS and suggest them take the safer travel route
- The dataset can be analysed further to obtain insightful information about the incidents and hence even more specific precautionary measures can be worked upon.
  Like here, by plotting no.of accidents and accident severity against the time of accident to determine the trend of hours when the most serious accidents have been taking place.



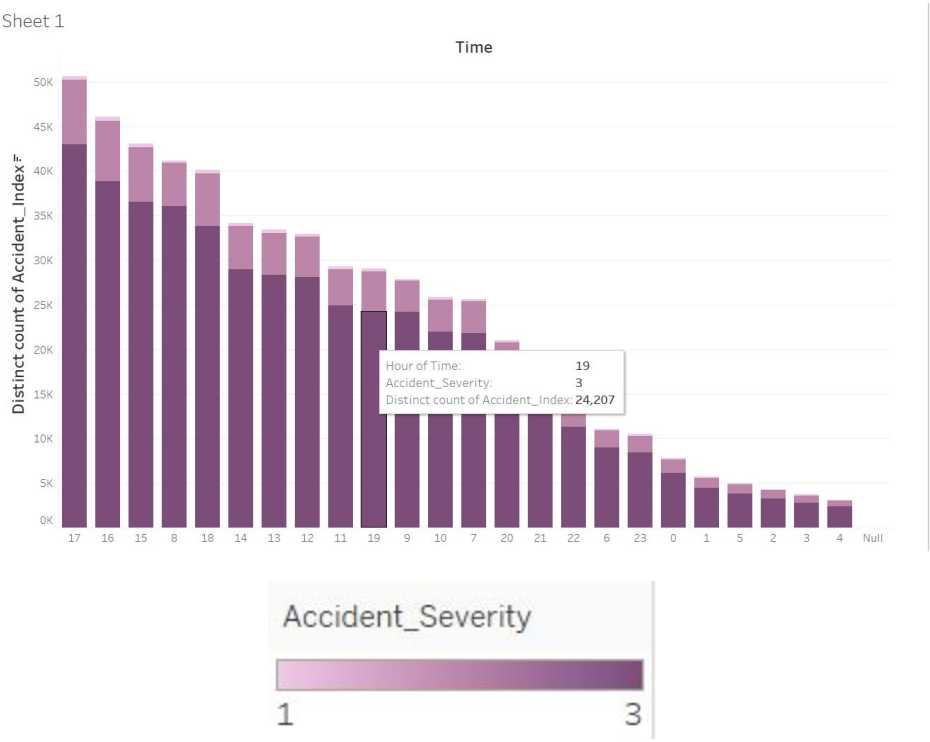



**Fig 25. Plot depicting no.of accidents and corresponding accident severity with varying time**

The system can hence be further improved to provide real time severity of being at a particular place based on all these attributes, and precautionary resources can be deployed accordingly.

**REFERENCES**

[1] Yiqun Xie,Shashi Shekhar, “Significant DBSCAN towards Statistically Robust Clustering”, SSTD '19: 16th International Symposium on Spatial and Temporal Databases (2019)

[2] Duczmal, Luiz et al. “Evaluation of spatial scan statistics for irregularly shaped disease clusters.” (2005).,Journal of Computational and Graphical Statistics.

[3] Tango, T., Takahashi, K. “A flexibly shaped spatial scan statistic for detecting clusters”. Int J Health Geogr 4, 11 (2005)

[4] Xie, Yiqun and Shashi Shekhar. “A Nondeterministic Normalization based Scan Statistic (NN-scan) towards Robust Hotspot Detection: A Summary of Results.” SDM (2019).

[5] Daniel B. Neill and Andrew W. Moore.“Rapid detection of significant spatial clusters”, ACM SIGKDD international conference on Knowledge discovery and data mining (2004).

[6] Xie, Yiqun & Eftelioglu, Emre & Ali, Reem & Tang, Xun & Li, Yan & Doshi, Ruhi & Shekhar, Shashi. “Transdisciplinary Foundations of Geospatial Data Science.” ISPRS International Journal of Geo-Information.(2017)

[7] Wei Wang, Jiong Yang, and Richard Muntz, “ STING : A Statistical Information Grid Approach to Spatial Data Mining” Department of Computer Science,1997.

[8] Tian Zhang, Raghu Ramakrishnan, and Miron Livny “BIRCH: an efficient data clustering method for very large databases”. SIGMOD Rec. 25, 2 (June 1996)

[9] R. T. Ng and Jiawei Han, "CLARANS: a method for clustering objects for spatial data mining," in IEEE Transactions on Knowledge and Data Engineering,Sept.-Oct. 2002.

[10] Hsien-Tsung Chang, Hieu Nguyen, “Clustering Methods and Bound Value in Classify Density Traffic Accident Areas”, 5th IIAE International Conference on Industrial Application Engineering 2017

[11] Kulldorff M, “A spatial scan statistic” Communications in Statistics:Theory and Methods andre1997.

[12] Arvind Sharma, R. K. Gupta and Akhilesh Tiwari, ”Improved Density Based Spatial Clustering of Applications of Noise Clustering Algorithm for Knowledge Discovery in Spatial Data”, Hindawi Publishing Corporation ,Mathematical Problems in Engineering, Volume 2016, Article ID 1564516.

[13] Andrew B. Lawson, “Hotspot detection and clustering: ways and means”, Springer Science+Business Media, LLC 2010

[14] Hari Krishna Kanagala,Dr. V.V. Jaya Rama Krishnaiah, “A Comparative Study Of K-means, DBSCANAnd OPTICS”, International Conference on Computer Communication and Informatics (ICCCI -2016), Coimbatore.

[15] Andrew Rosenberg and Julia Hirschberg “V-Measure: A conditional entropy-based external cluster evaluation Measure”, Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Prague, June 2007.

[16] https://github.com/chriswernst/dbscan-python/blob/master/README.md

[17] http://www.sthda.com/english/wiki/wiki.php?id_contents=7940

[18]http://www.francescogrigoli.it/tutorial/how-dbscan-clustering-algorithm-works-an-easy-guide-thorugh-sketches.html

[19] https://towardsdatascience.com/clustering-using-optics-cac1d10ed7a7