UNIVERSITY OF OKLAHOMA

GRADUATE COLLEGE

BIG DATA ANALYTICS SOLUTION FOR SMALL CELLS DEPLOYMENT USING MACHINE LEARNING TECHNIQUES

A THESIS

SUBMITTED TO THE GRADUATE FACULTY

in partial fulfillment of the requirements for the

Degree of

MASTER OF SCIENCE IN TELECOMMUNICATIONS ENGINEERING

By

SACHIN DAWARE
Norman, Oklahoma
2016

BIG DATA ANALYTICS SOLUTION FOR SMALL CELLS DEPLOYMENT USING MACHINE LEARNING TECHNIQUES

A THESIS APPROVED FOR THE
SCHOOL OF ELECTRICAL AND COMPUTER ENGINEERING

BY

____
Dr. Ali Imran, Chair

____
Dr. Pramode Verma

____
Dr. Gregory Macdonald

Acknowledgements

I would like to express my deepest gratitude to my advisor Dr. Ali Imran, who has led me to the field of Big Data and advanced technologies in 5G, who has inspired me a lot in my thesis work. His enthusiasm for research has motivated me to work hard all the time and I am very happy that I had an opportunity to carry out my very first research work under his guidance.

His in-depth understanding of the research topic and convincing explanations helped me to deal with advanced research topics and approach the research from several different angles.

Next, I would like to thank the other committee members, Dr. Pramode Verma, for believing in me and giving me the opportunity to pursue my graduate degree here, and Dr. Gregory Macdonald, for valuable comments on my work and sharing important suggestions on different data mining tools which were helpful for my task.

As an international student, editing my thesis for grammatical correction was a handful task, for which I want to thank Murray Patricia from writing services at OU-Tulsa.

Finally, I am grateful to have wonderful supporting friends and delightful people like Renee Wagenblatt in my life.

I would like to dedicate this work to my beloved family who always encouraged me throughout my Masters.

Table of Contents

List of Tables

List of Figures

Abstract

This thesis presents a “Novel Small Cell Planning Solution using Machine learning”. The Telecom service providers are interested in estimating various trends in order to plan future upgrades and deployments driven by real data. Fundamentally, the service provider landscape is changing. The numbers of devices are increasing in the network such as small cells to cater the growing demands. Also, the increasing amount of data has caused a big data revolution that is having an impact on telecom.

With the advance big data analytics solutions and with fine grained analytics in real time, needs in bandwidth change from one place to another throughout the day, week, month, etc, becomes predictable. Hence, big data analytics solutions can help in deciding footprint of small cells and efficiently deployment of small cells.

In this thesis, I have used the open big data that is published at the site: https://dandelion.eu/datamine/open-big-data/ under Open Data Commons Open Database License (ODbL) license. This dataset provides information about the telecommunication activities over the city of Milano. The dataset is the result of a computation over the Call Detail Records (CDRs) generated by the Telecom Italia cellular network over the city of Milano.

Data mining is the technique to find concealed and fascinating pattern from dataset, which can be used in decision making and future prediction. In this thesis, data preprocessing has been performed on hadoop framework using hive with Cloudera's open source platform, CDH cloudera-quickstart-vm-5.3.0-0-vmware.

In this thesis, the (Eps, MinPts) DBSCAN density based spatial clustering algorithm is used clustering the geospatial data. DBSCAN clusters a spatial data set based on two parameters namely physical distance from each point and a minimum cluster size. This method is best fit for spatial latitude-longitude data.

In this thesis, the scikit-leran machine learning platform is used to implement the solution, scikit-learn in python is one of the widely used machine learning platform, it provides a wide range of supervised and unsupervised learning algorithms via a consistent interface in Python.

For the validation of the clustering results, the data mining tool WEKA 3.6.11 is used. For benchmarking of the proposed solution, the DBSCAN algorithms clustering result is compared with the WEKA cluster's results. The final results show that the solution produces very promising results. The three promising results are , it is able to reveal all the objects from the datasets on the basis of user defined algorithm input parameters. The input parameters have a decisive impact on the cluster result. It can extract spatial, temporal and semantically separated clusters.

The detected clusters are visualized using Matplotlib plotting library for the Python, WEKA and geojson.io online tool.

Chapter 1: Introduction

1.1 Introduction and Background

4G LTE networks are now well established both in terms of the handset available and network infrastructure. Now, the eyes of the mobile communication industry are turning towards the next generation of technology, 5G. So why do we need 5G? We need 5G because we are seeing a growing demand in terms of capacity, capability and greater quality-of-service. Clearly more spectrum will add to the capacity, but capabilities are also very important such as high speed and low latency; therefore, we see there are more expectations with respect to quality-of-service.

1.2 Purpose and Motivation

Now, using the small cells is an elegant way to fulfill these growing demands. The small cells are essentially small cellular base stations designed to provide targeted capacity and coverage to improve the user experience. They operate in the license spectrum and use mobile technology. They are really designed to work with internet connected backhaul to keep cost down thereby adding to business case worth. There are several varieties of small cells like femtocells, picocells, and microcells. Using these varieties can help to fulfill the growing demand of dense networks. Despite the high expectations, the small cells market is only just emerging in many countries, and growth is on the way. The small cells are alternatives to costly macro cells and greatly complement to macro cell deployment today.

There are a couple of challenges in small cells deployment, namely lack of suitable tools and processes that allow the service provider to plan quickly and deploy massive numbers of small cells. The service providers can’t keep up with the rapidly changing new technology while continuously maintaining and expanding the current network infrastructure. The high complexity and sheer volume of tasks involved in the massive deployment of small cells frequently limits the service providers from meeting their time, cost, and quality targets.

The service providers are facing significant challenges. One of the biggest problems with small cells is the complexity of the network. The complexity comes with number of different backhaul technologies and possible permutations that must exist to deploy a small cell network. Apart from the complexity, there are the sheer volumes of tasks that must be completed in order to deploy the small cells. Also service providers have to manage hundreds of related projects across hundreds or even thousands of people and to deal with lower skilled human resources to install the equipment cost effectively. So, apart from the volume and complexity, service providers also faced lack of automation in their tools and processes that they need to deploy small cells in a cost-effective manner to meet the return on investment targets. Commonly, the biggest challenges for service providers for deploying small cells are network and project complexity, high volumes of small cells, outdated tools and processes, cost and time required for deploying.

How does one quickly and efficiently overcome these challenges and deal with all the variations in the design and deployment of the network. The traditional approach to deploy network starts simply with a plan. Even with a well-designed plan, there will be changes made over time. There are going to be duplicate processes, number of different access options, different aggregation options, venue-specific options, and so on. Generally the deployment process involves many workarounds, many manual steps, and undocumented processes.

The limitations of today’s planning tools are mainly around the lack of end-to-end orchestration across multiple systems and organizational groups. Also at the operator end, it’s difficult to scale the existing tools with increasing nodes in the network and growing demands.

There are certain levels of business impact due to these limitations. For example, there are higher project costs, things takes longer, the projects get delayed, and the quality of the deployed network may be impaired. If people cut corners and don’t follow processes and procedures, the result might more truck rolls and more site visits. Losing accountability because it’s difficult to trace who actually was responsible for a particular set of actions and changes in the network or indeed in the design. Also, there is a need of high skills resources which increases cost, overhead to train people and so on which leads difficulty in the massive deployment of small cells. The current tools available for small scale deployments are inadequate.

So, there is a need of smart end-to-end small cell deployment solutions that provide complete end-to-end, catalog-driven design and rollout solutions for next-generation networks. There is need of complete automated planning solution that reduces the cost, speeds up the planning which helps efficiently deployment of small cells, it reduces the deployment cost and the network design time by great extent. The solution is highly scalable and can deal with rapid deployment and increases the efficiency hence it helps service providers to improve operational expenses.

The small cells have been a major topic in the history of telecommunication over the last couple of years. Service providers started to plan mass deployments of small cells and realize the challenges and limitations they had with their current planning tools and procedures. Some service providers started performing analytics with available data to improve and sharpen the current network planning tools and process, and speed up decisions by using anonymized data from mobile networks. In a nutshell, this is data-driven decision making

1.3 Workflow of the Thesis

The following are the workflow steps to achieve the thesis objective of detecting the traffic pattern from the spatial temporal telecom Italia real dataset. Figure 1 shows the workflow of this research work. It is divided into 6 levels from a work point of view:

First step is to analyze the spatial temporal Italia telecom operator dataset. The dataset is the result of a computation over the call detail records (CDRs) generated by the Telecom Italia cellular network for the duration of one month consisting of the user activities like voice, text, and data from a specific geographical location represented using Milano grid. The Milano city urban area is spatially aggregated using a 100X100 grid. The dataset represents the traffic distribution across the grid.

In the second step, the data is to process by cleaning and formatting it, the data is processed over the hadoop framework to extract the hidden facts. The processed data is visualized in the form of heat maps to represent the different traffic patterns as per the traffic volume. The maps show dense areas and less dense areas with respect to user activities across Milano Grid.

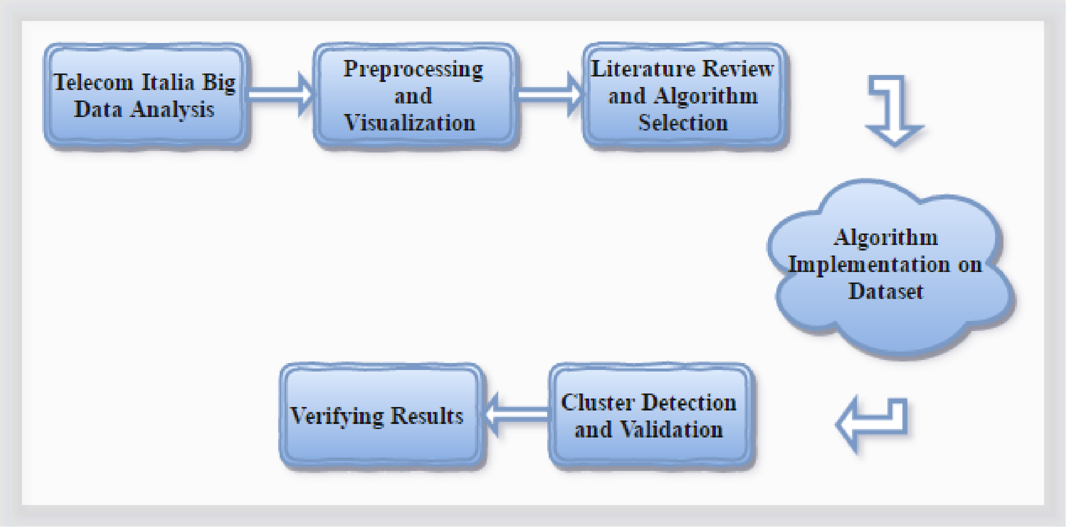

Telecom Italia Big Data Analysis
Preprocessing and Visualization
Literature Review and Algorithm Selection
Algorithm Implementation on Dataset
Cluster Detection and Validation
Verifying Results

Figure 1 Thesis Workflow

In the third step, the clustering algorithm executes on spatial temporal datasets that clusters the data based on traffic volumes from the Italia telecom operator dataset. To achieve the above- described objective, a theoretical literature review was performed of scientific research papers of similar domains. Subsequently, the most suitable research papers were selected. The selection of the algorithm was based on the ability to satisfy the end results and advantages and disadvantages of different investigated algorithms.

In fourth step, the selected algorithm is implemented in Python language, using scikit-learn open source machine learning tool. The algorithm was applied to different datasets, using various input parameters in order to get the desired results.

The fifth step involves plotting of the clusters generated by the proposed clustering algorithm. This is accomplished by using the Matplotlib plotting library for the Python programming language.

The last step verifies the results.

At the beginning of this thesis, the following end objectives were identified:

- What can be done with the spatial temporal Italia telecom operator dataset, the computation over the Call Detail Records (CDRs) generated by the Telecom Italia cellular network.
- How to visualize spatio-temporal data in the form of heat maps in way that different traffic pattern can be visualized clearly.
- Which clustering methods are available and which ones are the most suitable for spatial temporal data analysis?
- Clustering algorithm should give the optimal number of small-cells required to cover 100x100 Milano grid?
- The standard-deviation among traffic across all clusters should be minimum and desired shape of clusters is convex type.
- How to evaluate the clustering result?

1.4 Thesis Outline

The introduction is followed by **Chapter 2 Big Data Analytics in Telecommunication**, which gives a general overview of need for big data analytics in telecommunication and paybacks of data-driven decision making in telecommunications.

Chapter 3 Preprocessing and Visualization Telecom Italia Big Data provides an overview of and insights into the data. This section also includes a detailed explanation of end-to-end steps taken to process the data, including the tools and frameworks used to process the data. This section also gives a description of the representation of the geo spatial data into GeoJSON format and visualizes it.

Chapter 4 Clustering Algorithms for Spatio Temporal Traffic Analysis provides a literature review of clustering algorithms for spatial temporal data analysis, which plays an important role in seeing inside the dataset. The section also provides information about different clustering techniques and advantages and disadvantage of each.

Chapter 5: A Novel Small-Cells Planning Solution Using Clustering Algorithm provides the workflow details and implementation aspects of the proposed clustering algorithm. This section also tells about the different approaches studied to meet the end objective. At the end of this chapter, the validation result of the proposed algorithms is included along with the theoretical results.

Chapter 6 Conclusion and Future Work concludes the thesis with a summary, a description of the encountered problems during experimentation and suggestions for the future research work.

Chapter 2: Big Data Analytics in Telecommunication

2.1 Data Driven Decision Making

Businesses make thousands of decisions every day. Most are routine decisions, but what happens when a decision needs to be taken is beyond day-to-day operations that may shift the course of a business or even an industry and reshape the world. It has been observed that decision making lacks the sophistication and speed needed to ensure competitive advantage. The technology makes possible to bring together the art of instinct and experience with the science of data and analytics. So that decision makers are able to model opportunities based on specific product and market features such as whether to grow or shrink a business or to collaborate with competitors to clearly identify opportunities and associated risks. In a world where opportunities are more complex and more accelerated, technology will continue to deliver with analytical power for those decisions that really count by bending the art of leadership in judgment with the science of analytics excellence we can make better and faster decisions.

2.2 Need of Big Data Analytics in the Telecom Industry

Telecom providers are interested in estimating various trends in order to plan future upgrades and deployments driven by real data. As an example, a typical provider would want to know the equipment, such as a cell tower, that originates the bulk of the calls. Another valuable data point is determining the base stations that serve as the busiest switching hubs during varying times of the day, especially the daily traffic patterns and the sorts of calls frequently made to various locations. This study correlates base station and cell towers these two big data sources which are typically in the order of hundreds of gigabytes on a daily basis for a moderate-sized cellular network, this contain records for every cellular transaction being made on any cellular network.

Fundamentally, the service provider landscape is changing. There are many more devices and subscribers than in the past and much as well as much more data on a network. Those increases have caused a big data revolution that is having an impact on telecom. The data comes in various forms: consumer data, application- and device-generated data, from services offered through service providers, industrial IOT, machine-to-machine data, even the network and service data from equipment like routers and switches, as well as data from operations like OSS and BSS systems.

2.3 Use of Big Data in Network Planning

In 2020, the world will generate 50 times the amount of data it did in 2011. Telecommunications data, e.g., URLs, text, voice, mobile, broadband, video, P2P, sensors, statistics, RF data, location, tweets and geolocation, is one of the major sources of big data. The data generated by telecommunication networks are fast and rich. It includes charging data records, subscriber locations and movements, transportation patterns, user flows, internet traffic, key demographics, potential markets, and network monitoring.

The service providers collect and store exabytes of data. The data originates and is collected in many different locations and can be used to find meaningful facts. But at the same time, managing and processing that data is time consuming, expensive, prone to error, causing service providers to lose customers, business, and operational advantage. Service providers buffer the system with resources and people and services that ultimately leads to huge operational and maintenance costs.

Now, with the advance of big data analytics solutions and with fine grained analytics in real time, the cost of collecting, maintaining, and using big day goes down by way of lower network consumption for data transfer, fewer resources and less infrastructure used, plus lower CapEx and OpeX. Big data analytics provides a great opportunity to interact with customers and to increase revenue from existing products, and to dynamically provision networks for optimum performance. Businesses must know their customers very well and the services that each customer wants so they can offer better services and promotions to various segments of their market. The service providers can jumpstart innovation; they can build new effective, sustaining revenue streams. Data analytics enable business to deliver better services while increasing application-level security and monitoring.

Big data alone is not the answer to being a successful business. It is how that data is used to understand customer behavior and business performance that reaps financial benefits. Service providers strive to think how this data helps to improve end-to-end services and performance of networks in real time that brings financial success to them and their users. Engineering teams think that if big data can be used to predict network utilization and avoid traffic, network performance increases and productivity rises.

2.4 Business and Network Planning

In the telecom business, business organizations are looking to add new services to increase sales. Data analysis helps those businesses determine the new services that have the most positive impact on customers. Business development teams use big data to figure out how their company can reduce costs, improve the profitability, and meet customers' needs.

So there are some specific operators issues that service providers want to address. For example limiting network visibility is a huge challenge. Because they got access to data internet, they are collecting data through the management systems and it's reported at a level where it is aggregated; therefore, the data is not available at a sufficiently granular level so as to detail what going on within network. Specially while offering mission critical applications or high performance demanding applications that one big issue for certain group of customers. The one is improving video quality and the overall quality and experience.

Deploying and maintaining the edge network is one of the costliest operations for operators of cellular networks. Antennas have to be placed in an optimal way to address present and future needs. The network should also be adaptable to the ground reality over time for a better efficiency, user experience, and reduced CAPEX and OPEX.

Heterogeneous Network is one when different kinds of radio access nodes — which are placed according to the needs of the field are in place. Small Cells, operator-controlled, low powered radio access nodes with a low range. It is a cost-effective alternative to traditional macro networks in remote rural areas with little or no terrestrial network infrastructure. Operators can deploy small cells to improve local coverage where traditional antennas are struggling to meet the needs. Massive MIMO technology: base transceiver stations with hundreds of antennas that can focus their beam to a given direction. It allows for a user to receive data from several channels at a time. An operator would benefit from these technologies only if the nodes are well placed. To achieve this, it is needed to know where the customers are located during a day, month, year, etc., and what kind of services they use.

Long-term trends to determine the evolving needs for the network capacity, coverage and the number of nodes to be placed in the future. Seasonal trends to determine where and when consumptions peaks are expected so those nodes are placed accordingly. Well-placed radio access nodes and prediction of trends would make this network efficient and reduce CAPEX.

To place radio access nodes one correctly and efficiently the use of statistical data is required. In order to predict consumption trends, it is required to analyze the above data in correlation to other contextual data such as weather conditions, social networks, news on the web, etc. A classical computing system cannot deal with such a massive volume, velocity, and variety of data. Big Data solutions need to be deployed as well as Advanced Analytics in order to transform complex data into easily actionable insights.

When an operator deploys a 5G network, many new BTS as well as new kind of cell sites, such as small cells, will have to be installed. As far more objects will be then connected, the operator needs to know and monitor the bandwidth needs with a higher degree of precision. By tracking users’ mobile devices, it is possible to estimate in real-time where, when, and how many consumers need bandwidth, and how much of it do they need. This information is to be collected in real-time so that the operator knows where and how many antennas have to be deployed at a given site.

With a Big Data Analytics solution, how needs in bandwidth change from one place to another throughout the day, week, month, etc, becomes predictable. Also, the operator acquires enough information to send self-driven mobile radio access nodes to the relevant locations, and knows where antennas should be directed so that consumption peaks are absorbed smoothly without compromising customers' experience and satisfaction. Big Data analytics solutions can help in deciding footprint of small cells and efficiently deployment of small cells.

2.5 Applications of Big Data Analytics

New trade area analysis.

Cheaper network load balancing.

Deploying new Network Infrastructure like small-cells.

Effective planning of new technology like 5G.

Improved QoS.

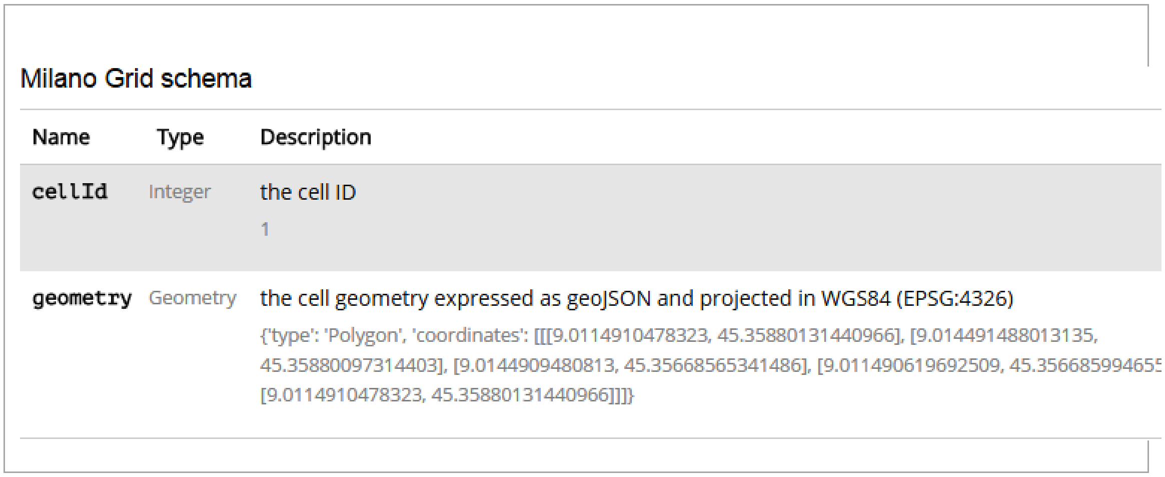
Transportation Management.

Chapter 3 Preprocessing and Visualization Telecom Italia Big Data

3.1 Telecom Italia Big Data

At the beginning of 2014, Telecom Italia launched Big Data Challenge, a contest designed to stimulate the creation and development of innovative technological ideas in the big data field. SpazioDati was the technology partner hosting the data distribution platform, using dandelion.eu. The open big data is published at this site https://dandelion.eu/datamine/open-big-data/ under Open Data Commons Open Database License (ODbL) license.

3.1.1 Milano Grid



Milano Grid schema

Name	Type	Description
cellId	Integer	the cell ID 1
geometry	Geometry	the cell geometry expressed as geoJSON and projected in WGS84 (EPSG:4326) {'type': 'Polygon', 'coordinates': [[[9.0114910478323, 45.35880131440966], [9.014491488013135, 45.35880097314403], [9.0144909480813, 45.35668565341486], [9.011490619692509, 45.35668599465 [9.0114910478323, 45.35880131440966]]]}

3.1.2 Telecommunications - SMS, Call, Internet – MI

This dataset provides information about the telecommunication activity over the city of Milano. The dataset is the result of a computation over the call detail records (CDRs) generated by the Telecom Italia cellular network over the city of Milano. CDRs log the user activity for billing purposes and network management. There are many types of CDRs. For the generation of this dataset, we considered those related to the following activities:

Received SMS: a CDR is generated each time a user receives an SMS.

Sent SMS: a CDR is generated each time a user sends an SMS.

Incoming Calls: a CDR is generated each time a user receives a call.

Outgoing Calls: CDR is generated each time a user issues a call.

Internet: CDR generate each time a user starts an internet connection. When a user ends an internet connection during the same connection, one of the following limits is reached: 15 minutes from the last generated CDR or 5 MB from the last generated CDR.

Aggregating the aforementioned records created the dataset that provides SMSs, calls, and Internet traffic activity. It measures the level of interaction of the users with the mobile phone network. These different activity measurements are provided for each square of the Milano GRID. Activity measurements are obtained by temporally aggregating CDRs in time slots of 10 minutes.

Table 1 Telecom Italia Dataset Telecommunications - SMS, Call, Internet – MI

```
1      1383260400000  0      0.08136262351125882
1      1383260400000  39     0.14186425470242922   0.1567870050390246
       0.16093793691701822   0.052274848528573205  11.028366381681026
1      1383261000000  0      0.13658782275823106
       0.02730046487718618
1      1383261000000  33                                  0.026137424264286602
1      1383261000000  39     0.27845207746066025   0.11992572014174135
       0.1887771729145041    0.13363747203983203   11.100963451409388
1      1383261600000  0      0.05343788914147278
1      1383261600000  39     0.3306414276169333    0.1709520296851148
       0.13417624316013174   0.05460092975437236   10.892770602791096
1      1383262200000  0      0.026137424264286602
1      1383262200000  39     0.6814340796890551    0.22081529861558874
       0.02730046487718618   0.05343788914147278   8.622424590989748
1      1383262800000  0      0.02730046487718618
1      1383262800000  39     0.24337810266887647   0.1928905642458027
       0.05343788914147278   0.08073835401865896   8.009927462445756
1      1383263400000  0      0.02730046487718618
1      1383263400000  39     0.0563882398598718    0.24337810266887647
```

```
        0.02730046487718618   0.02730046487718618   8.1184195540896
1       1383264000000  0      0.029712044475285478
        0.003574620210998875
1       1383264000000  39     0.13533928377303134   0.0849372437222577
        0.05343788914147278   0.0017873101054994376 8.026269748512151
1       1383264600000  0      0.02730046487718618
1       1383264600000  39     0.1887771729145041    0.026137424264286602
        0.0017873101054994376 0.05460092975437236   8.514178577183893
1       1383265200000  39     0.24221506205597687   0.16031366742441833
        0.10803881889584513   0.026137424264286602  6.8334248963840505
1       1383265800000  0      0.02730046487718618
        0.02730046487718618
1       1383265800000  39     0.2944899105845501    0.2457041838946756
        0.02730046487718618   0.08073835401865896   6.55460504454769
1       1383266400000  0      0.026137424264286602
1       1383266400000  39     0.10803881889584513   0.10803881889584513
        7.338716012816092
1       1383267000000  0      0.02730046487718618
1       1383267000000  39     0.05701250935247166   0.14490010379312837
        6.779704731239178
1       1383267600000  39     0.13953817347663006   0.05522519924697222
        7.19216205511537
1       1383268200000  39            0.05343788914147278
        0.02730046487718618   7.503314068316409
1       1383268800000  39     0.02730046487718618   0.05460092975437236
        0.02792473436978604          6.169533683920985
1       1383269400000  0      0.02730046487718618
1       1383269400000  39     0.05996286007087068   0.0563882398598718
        0.029087774982685617  7.605452185775456
1       1383270000000  39     0.13417624316013174   0.05343788914147278
        0.05343788914147278   6.56956533543098
```

3.1.3 Schema of Dataset

Square id: The id of the square that is part of the Milano GRID; TYPE: numeric

Time interval: The beginning of the time interval expressed as the number of millisecond elapsed from the Unix Epoch on January 1st, 1970, at UTC. The end of the time interval can be obtained by adding 600000 milliseconds (10 minutes) to this value. TYPE: numeric

Country code: The phone country code of a nation. Depending on the measured activity, this value assumes different meanings that are explained later. TYPE: numeric

SMS-in activity: The activity in terms of received SMS inside the Square id, during the time interval and sent from the nation identified by the country code. TYPE: numeric

SMS-out activity: The activity in terms of sent SMS inside the Square id, during the time interval and received by the nation identified by the country code. TYPE: numeric

Call-in activity: The activity in terms of received calls inside the Square id, during the time interval and issued from the nation identified by the country code. TYPE: numeric

Call-out activity: The activity in terms of issued calls inside the Square id, during the time interval and received by the nation identified by the Country code. TYPE: numeric

Internet traffic activity: The activity in terms of performed internet traffic inside the Square id, during the time interval and by the nation of the user performing the connection identified by the country code. TYPE: numeric

3.2 Big Data with Hadoop

With rapid innovations, frequent evolutions of technologies, and a rapidly growing internet population, systems and enterprises are generating huge amounts of data to the tune of terabytes and even petabytes. Since data is being generated in very huge volumes with great velocity in all multi-structured formats like images, videos, weblogs, sensor data, etc. from a variety of sources, there is a huge demand to efficiently store, process, and analyze this large amount of data to make it usable.

Hadoop is an open-source software framework for storing data and running applications on clusters of commodity hardware. It provides massive storage for any kind of data, enormous processing power, and the ability to handle virtually limitless concurrent tasks or jobs. The core of Hadoop consists of a storage part, known as Hadoop Distributed File System (HDFS), and a processing part called MapReduce. Hadoop splits files into large blocks and distributes them across nodes in a cluster. It then transfers packaged code into nodes to process the data parallelly. This approach takes advantage of data locality and nodes manipulating the data they have access to in order to allow the dataset to be processed faster and more efficiently than it would be in a more conventional supercomputer architecture that relies on a parallel file system where computation and data are distributed via high-speed networking.

The main elements of Hadoop architecture are as follows.

Hadoop Common – contains libraries and utilities needed by other Hadoop modules.

Hadoop Distributed File System (HDFS) – a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster.

Hadoop YARN – a resource-management platform responsible for managing computing resources in clusters and using them for scheduling of users' applications.

Hadoop MapReduce – an implementation of the MapReduce programming model for large scale data processing.

Because data is growing rapidly, the traditional data processing technologies are not efficient and cost effective. So there is a need for efficient data processing techniques. With data volumes and varieties constantly increasing, especially from social media and the Internet of Things (IoT) and in order to improve the user experience, there is a need to analyze the real-time data and calibrate the processes, infrastructure, and business models. Hadoop is key element in processing large amounts of data. The Hadoop framework has the ability to store and process huge amounts of any kind of data quickly. Hadoop's distributed computing model processes big data fast. The more computing nodes, the more processing power can be added to the system. Data and application processing are protected against hardware failure. If a node goes down, jobs are automatically redirected to other nodes to make sure the distributed computing does not fail. Multiple copies of all data are stored automatically. Unlike traditional relational databases, there is no need to preprocess data before storing it. Limitless amounts of data, including unstructured data like text, images, and videos, can be stored so the decision of how to use it can occur later. The open-source framework is free and uses commodity hardware to store large quantities of data. The system can easily handle more data simply by adding nodes.

In this thesis, I have used CDH Cloudera-Quickstart-vm-5.3.0-0-vmware. Cloudera's open source platform is the most popular distribution of Hadoop and related projects in the world.

3.3 Using Hive for Big Data Analytics

Hive is a data warehouse infrastructure built on top of Hadoop for providing data summarization, query, and analysis. Hive is very simple to use. It projects a structure onto the data and queries this data following an SQL-like query structure to perform Map and Reduce tasks on large datasets.

Table 2 Hive Table Description

```
hive> describe smscallinternetmi20131201;
OK
squareid     int
tmrval double
countrycode  int
smsinactivitydouble
smsoutactivity      double
callinactivity      double
calloutactivity     double
internettrafficactivity   double
```

Table 3 Hive Query Output

```
hive> select squareid , count(callinactivity) from smscallinternetmi20131201
group by squareid;
Total MapReduce jobs = 1
Launching Job 1 out of 1
Number of reduce tasks not specified. Estimated from input data size: 1
In order to change the average load for a reducer (in bytes):
  set hive.exec.reducers.bytes.per.reducer=<number>
In order to limit the maximum number of reducers:
  set hive.exec.reducers.max=<number>
In order to set a constant number of reducers:
  set mapred.reduce.tasks=<number>
Starting Job = job_201510281144_0009, Tracking URL =
http://t1.thadoop.com:50030/jobdetails.jsp?jobid=job_201510281144_0009
Kill Command = /usr/lib/hadoop/bin/hadoop job  -kill job_201510281144_0009
Hadoop job information for Stage-1: number of mappers: 2; number of
reducers: 1
2015-10-30 02:22:30,044 Stage-1 map = 0%,  reduce = 0%
2015-10-30 02:22:48,276 Stage-1 map = 25%,  reduce = 0%
2015-10-30 02:22:54,314 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU
15.14 sec
2015-10-30 02:22:55,319 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU
15.14 sec
2015-10-30 02:22:56,324 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU
15.14 sec
2015-10-30 02:22:57,328 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU
15.14 sec
2015-10-30 02:22:58,333 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU
15.14 sec
2015-10-30 02:22:59,339 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU
15.14 sec
2015-10-30 02:23:00,348 Stage-1 map = 50%,  reduce = 0%, Cumulative CPU
15.14 sec
2015-10-30 02:23:01,358 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
18.07 sec
2015-10-30 02:23:02,365 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
18.07 sec
2015-10-30 02:23:03,370 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
18.07 sec
2015-10-30 02:23:04,377 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
18.07 sec
2015-10-30 02:23:05,382 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
18.07 sec
2015-10-30 02:23:06,390 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
18.07 sec
2015-10-30 02:23:07,395 Stage-1 map = 100%,  reduce = 0%, Cumulative CPU
18.07 sec
2015-10-30 02:23:08,399 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU
19.92 sec
2015-10-30 02:23:09,405 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU
19.92 sec
2015-10-30 02:23:10,412 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU
```

```
19.92 sec
2015-10-30 02:23:11,423 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU
19.92 sec
2015-10-30 02:23:12,427 Stage-1 map = 100%,  reduce = 100%, Cumulative CPU
19.92 sec
MapReduce Total cumulative CPU time: 19 seconds 920 msec
Ended Job = job_201510281144_0009
MapReduce Jobs Launched:
Job 0: Map: 2  Reduce: 1   Cumulative CPU: 19.92 sec   HDFS Read: 298846908
HDFS Write: 88422 SUCCESS
Total MapReduce CPU Time Spent: 19 seconds 920 msec
OK
```

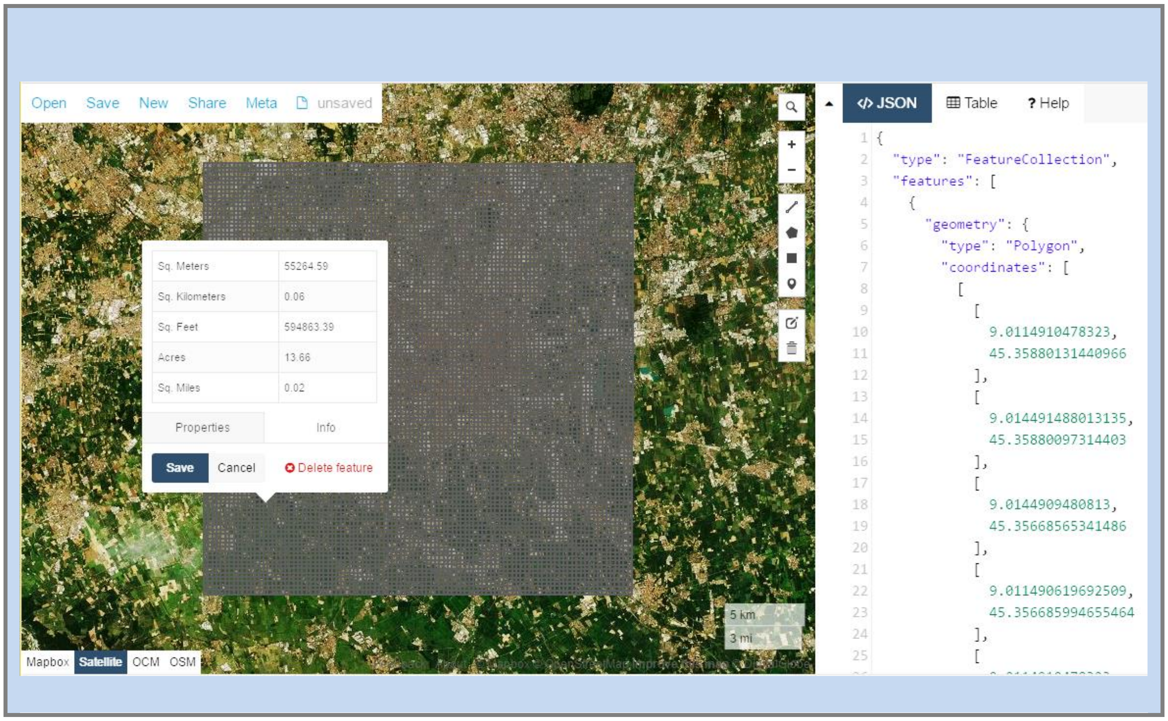
3.4 GEOJSON

GeoJSON is a format for encoding a variety of geographic data structures.

```
{
  "type": "Feature",
  "geometry": {
    "type": "Point",
    "coordinates": [125.6, 10.1]
  },
  "properties": {
    "name": "Milano Urban Area"
  }
}
```

A GeoJSON object may represent geometry, a feature, or a collection of features. GeoJSON supports the following geometry types: Point, Line String, Polygon, Multipoint, Multiline String, MultiPolygon, and Geometry Collection. Features in GeoJSON contain a geometry object, additional properties. A feature collection represents a list of features.

A complete GeoJSON data structure is always an object (in JSON terms). In GeoJSON, an object consists of a collection of name/value pairs -- also called *members*. For each member, the name is always a string. Member values are string, number, object, array, or one of the literals: true, false, and null. An array consists of elements where each element is a value as described above.



Sq. Meters	55264.59
Sq. Kilometers	0.06
Sq. Feet	594863.39
Acres	13.66
Sq. Miles	0.02

```
{
  "type": "FeatureCollection",
  "features": [
    {
      "geometry": {
        "type": "Polygon",
        "coordinates": [
          [
            [
              9.0114910478323,
              45.35880131440966
            ],
            [
              9.014491488013135,
              45.35880097314403
            ],
            [
              9.0144909480813,
              45.35668565341486
            ],
            [
              9.011490619692509,
              45.356685994655464
            ],
            [
```

Figure 2 Milano-Grid and object features with GeoJson

Chapter 4: Clusteing Algorithms for Spatio Temporal Traffic Analysis

4.1 Overview of Cluster Analysis

When flying over a city, one can easily identify fields, forests, commercial areas, and residential areas based on their features, without anyone's explicit training this is the power of cluster analysis.

A cluster is actually a collection of data objects; those objects are similar within the same cluster. That means the objects are similar to one another within the same group and they are rather different or they are dissimilar or unrelated to the objects in other groups or in other clusters. The cluster analysis which is also called clustering or data segmentation. The cluster analysis is to partition the data into a set of clusters or set of groups, they are as similar as possible within the same group and as far apart as possible among different groups. Cluster analysis is unsupervised learning in the sense there are no predefined classes this is very different from classification which needs supervised learning or needs to give in the class labels then we can construct the classification models. There are many ways to use or apply cluster analysis essentially cluster analysis can either provide as a stand-alone tool to get insight into your data distribution like a summary or we can use it to serve as a preprocessing steps or intermediate steps for other algorithms like classification or a prediction or like many other tasks including data mining and other applications.

4.1.1 Application of Clusters

Cluster analysis has lots of applications. For example, it has been properly used as pre-processing step or intermediate step for other data mining tasks. For example, you can generate complex summary of data for classification, pattern discovery, hypothesis generation and testing and many others. And it also has been popularly used for outlier detection, because outliers can be considered those points that are far away from any cluster. Cluster analysis also has been used for data summarization, compression and reduction. For example, in image processing, vector quantization has been using cluster analysis quite a lot. Cluster analysis also can be used for collaborative filtering, recommendation systems or customer segmentation, because clusters can be used to find like-minded users or similar products. Cluster analysis also has been used for trend detection, for dynamic data. For example, we can cluster stream data or detecting trends and patterns in dynamic data strings. And cluster analysis also has been used for multimedia data analysis, biological data analysis and social network analysis. For example, we can use clustering methods to cluster images or videos or audio clips or we can use cluster analysis on genes and protein sequences and many other interesting tasks.

4.1.2 Prerequisites and challenges

There are many things to consider for cluster analysis. For example, the first one people are often to consider is partitioning criteria. That means whether we want to get a single level or multiple-level hierarchical partitioning. In many cases, multi-level hierarchical partitioning or what we call hierarchical clustering could be quite desirable.

The second thing we want to consider whether how the clusters can separate the objects. It can be hard or exclusive, or soft, non-exclusive. The exclusive means one object like one customer can belong to only one region; Non-exclusive for example one object may belong to more than one cluster.

Then for a similarity measure, there could be many different kinds of similarity measures. For example, distance-based using Euclidean space or a road network or vector space or we can use connectivity-based, for example, based on density or contiguity, also clustering space like full space clustering or we subspace clustering. The full space means especially in the low dimensional, for example in 2D, two dimensional space, in the area or in the region, whether we want to find the clusters in those region, in the full 2D space or only we want to project on the 1D subspace.

However, in many high dimensional clustering, it is hard to find meaningful clusters in a very high dimension space but it is possible to find interesting clusters in the subspaces for example, in the corresponding lower dimensional space so the subspace clustering also becomes very important. Then, for cluster analysis, there are many requirements and challenges. The first, most important thing is the quality of clustering means whether we are going to deal with different kinds of attributes, numerical one, categorical data, text, multimedia data, networks, or a mixture of multiple types. Whether we can discover clusters with arbitrary shape, like an oil spill, or whether we will be able to deal with noisy data.

Then the second important issue people consider is scalability that means whether we will be able to deal with very big amount of data environment, whether we can cluster all the data or we must draw some sample and cluster only on the sample rate whether we can handle high dimensional data, or we can only project them in the low dimension space or whether we can do timely incremental clustering, especially for data streams or whether we can do stream clustering especially whether the cluster results may not be sensitive to the input order of the data.

The other important issue we need to handle is constraint-based clustering that means users usually have their own thinking on the preference or constraints of the cluster they want to derive. They may have domain knowledge. They may know certain subspaces could be more interesting. They may even raise queries. So can we handle clustering given user preference or constraints.

The final important issue is interoperability and usability, the clustering whether they can derive meaningful clusters which can be understood by people, can be used by many applications, can interpret data nicely.

4.2 Overview of Clustering Methods

There are many clustering methods they can be roughly categorized into a few clustering methodologies.

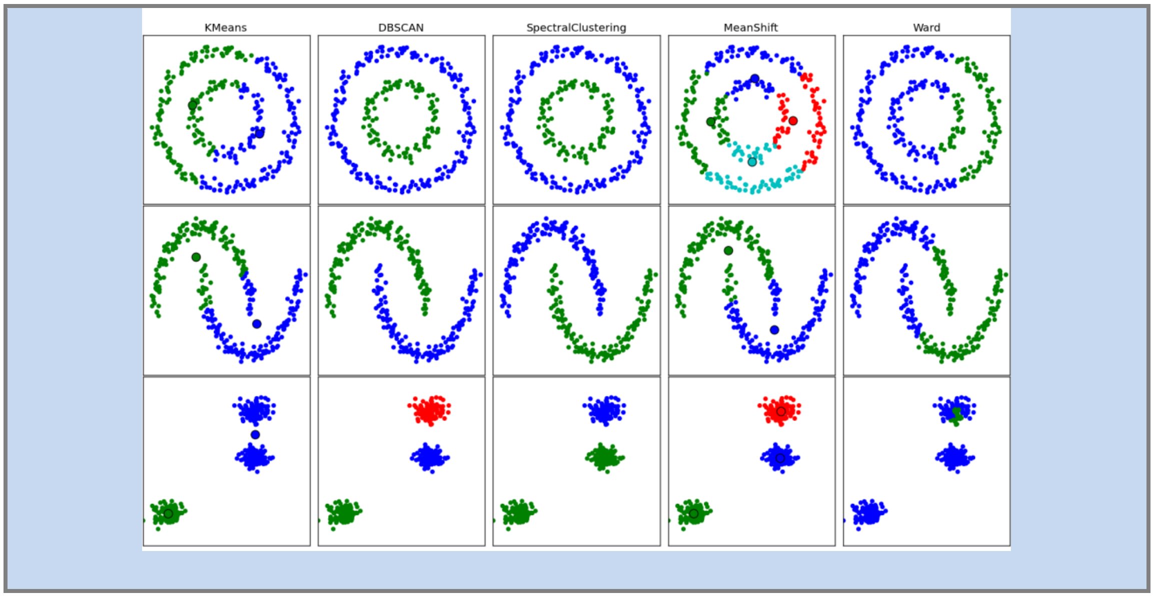


Figure 3 Overview of Clustering Methods

The first method called distance-based methods essentially there are two kinds of methods one called partitioning algorithms and the other called hierarchical algorithms. Partitioning algorithms essentially is, partitioning the data in a high dimension space, into multiple clusters. The typical methods include K-Means, K-Medoids, and K-Medians. For hierarchical algorithms, we call bottom-up merging many points into clusters, merging small clusters into bigger ones then form hierarchical clusters or we can use divisive methods start with a single bin all the data is encased in one cluster and then we try to split them divide them using the top-down splitting, into smaller and smaller clusters.

The second methodology called density-based methods. The third one called grid-based methods. Density-based method essentially is, we can think the database space, made at a high-level granularity, may you think about it with certain grade, or certain structures, or certain k nearest neighbors, we may find certain density. For the dense small regions, we can merge them into bigger regions. In this way, we will be able to find clusters, those events regions, with arbitrary shape.

The grid-based method is partitioning the data space into grid-like structures each grid is a summary of the characteristics of the data, in the lower grid or lower cells. Then there is other method is called as probabilistic and generative models in this method we can model data from a generative process, with the data, we can think the current points are generated from these underlying generative model, or underlying generative mechanisms. Then we can model parameters, using an Expectation-Maximization algorithm we are going to introduce based on the available data sets, we may try to find the maximum likelihood fit. And based on this fit, we will be able to estimate the generative probability of the underlying data points. And based on this, we may find the models. Then, in many cases the data may be sitting in a high-dimensional space.

There is another influential method called subspace clustering where you try to find the clusters on various subspaces. So, that method can be categorized into bottom-up methods, top-down high-dimensional clustering method, or correlation-based methods. For high-dimensional clustering, there are many methods developed for dimensionality reduction, essentially is we can think the height dimension is in a vertical form there, containing lots of columns and for those columns, when we perform clustering, essentially the columns are clustered, the rows or columns can be clustered together, we call co-clustering. Another very interesting method is called spectral clustering in this we use a spectrum of the similarity matrix of the data, to perform dimensionality reduction. The higher dimension reduced into the lower, fewer dimensions and then we can perform clustering in fewer dimensions.

4.3 Clustering Different Types of Data

We encounter various kinds of types of data. For example, the early clustering algorithm most times with the design was on numerical data. And the second type of data is category data, including the binary that most people consider as also can be handled in categorical data or category. For example, gender, race, zip code, or market-basket data, you would consider the data is discrete. There's no natural order, but certain kind of data is text data, this is very popular in social media, on the Web, or social networks.

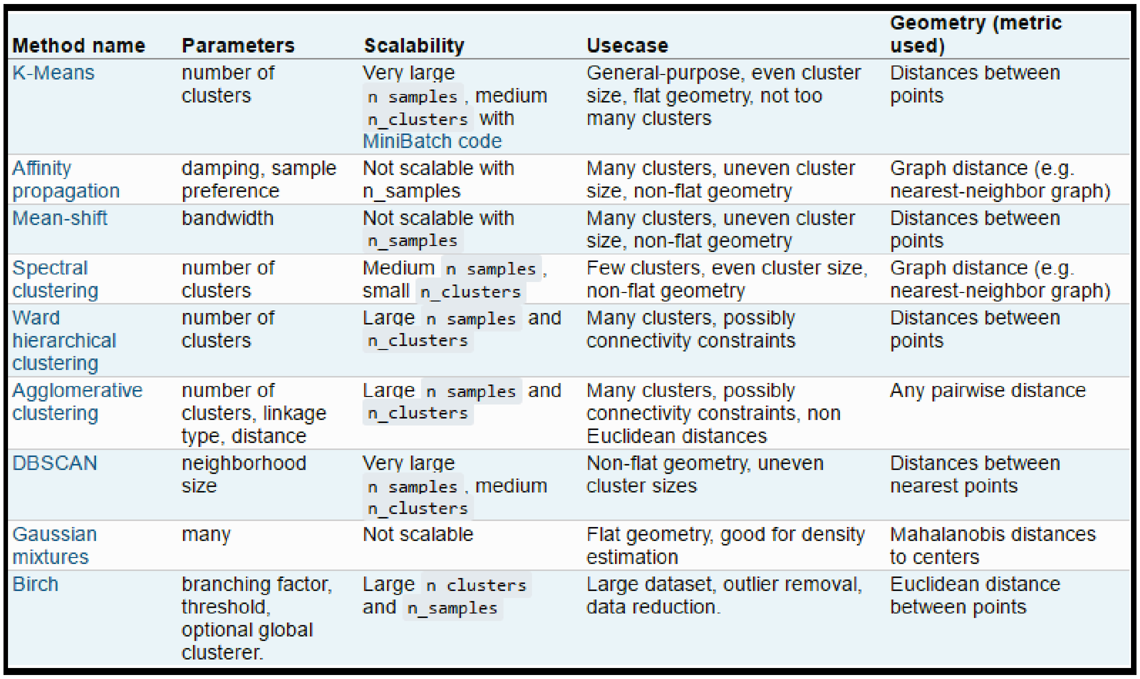
Multimedia data is another kind of data needs clustering. Usually images, audio, video, those are Flickr and YouTube actually are multimedia data. They are multimodal in a sense. They have image, audio, video, many cases they have text, as captions as well.

We can consider contextual data as well, they contain both behavioral and the contextual attributes. For example for images, we can consider the position of a pixel represents its context; the value represents its behavior. For video and music data, we can consider temporal ordering of the records represents its meaning. Time-series data is another popularly encountered data like in sensors, stock markets, temporal tracking, or forecasting, those task are usually handled time-series. Time-series, we usually consider data are temporally dependent.

Sequence data is another kind of data. Usually in weblog analysis or biological sequence analysis, or analyze the system commands. These, we can think the placement rather than time, the contextual attribute. That means where they are.

Stream data is another kind of data. Stream data can be considered as a real-time data, like water flowing in and out. It may evolve a long time. It may have concept shifts because stream coming and go. So, we need a single pass algorithm, rather you can see it again and again.

Then a little bit beyond homogeneous network are heterogeneous networks. This kind of network consists of multiple types of nodes and edges. Like a bibliographical data, hospital, handling disease, patients, doctors, and treatments to cluster different kinds of nodes and links together. There are some interesting algorithms, like the NetClus algorithm.



Method name	Parameters	Scalability	Usecase	Geometry (metric used)
K-Means	number of clusters	Very large `n_samples`, medium `n_clusters` with MiniBatch code	General-purpose, even cluster size, flat geometry, not too many clusters	Distances between points
Affinity propagation	damping, sample preference	Not scalable with n_samples	Many clusters, uneven cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Mean-shift	bandwidth	Not scalable with `n_samples`	Many clusters, uneven cluster size, non-flat geometry	Distances between points
Spectral clustering	number of clusters	Medium `n_samples`, small `n_clusters`	Few clusters, even cluster size, non-flat geometry	Graph distance (e.g. nearest-neighbor graph)
Ward hierarchical clustering	number of clusters	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints	Distances between points
Agglomerative clustering	number of clusters, linkage type, distance	Large `n_samples` and `n_clusters`	Many clusters, possibly connectivity constraints, non Euclidean distances	Any pairwise distance
DBSCAN	neighborhood size	Very large `n_samples`, medium `n_clusters`	Non-flat geometry, uneven cluster sizes	Distances between nearest points
Gaussian mixtures	many	Not scalable	Flat geometry, good for density estimation	Mahalanobis distances to centers
Birch	branching factor, threshold, optional global clusterer.	Large `n_clusters` and `n_samples`	Large dataset, outlier removal, data reduction.	Euclidean distance between points

Figure 4 Comparison of Clustering Algorithms

Uncertain data means the data may contain noise, may have approximate values, or multiple possible values. Usually we need to incorporate some probabilistic information, like distribution, or approximate values that were improve the quality of clustering. Recently, there are lots of discussions on big data, that we can model systems, that may store and process huge big data, like weblog analysis. Usually, like Google's MapReduce framework is you have map function to distribute the computation across many machines ,then we have reduce function to aggregate the results obtained from the Map step. So we can see, the data is very rich that's why the clustering is a rather sophisticated methodologies, trying to handle different kinds of data. The validation of clusters plays a vital role in cluster analysis that means you may want to evaluate the quality of the clusters generated based on certain validation methods.

Chapter 5: A Novel Small-Cells Planning Solution Using Clustering Algorithm

5.1 Literature Review

Data mining is the technique to find concealed and fascinating pattern from dataset, which can be used in decision making and future prediction. Spatial temporal analysis of any geospatial data mainly depends on the clustering method that is applied on the dataset to find a meaningful result. For literature review, the research papers were searched with relevant keywords like spatial clustering, clustering algorithm, real-time algorithm, algorithms for spatial temporal data. All papers were selected from well-known publications and journals. The 30 papers were selected for literature review out of the many papers found. Summary of the top research papers are shown in below table which is mainly focused on the algorithm used, its input parameters, dataset type on which the algorithm was applied.

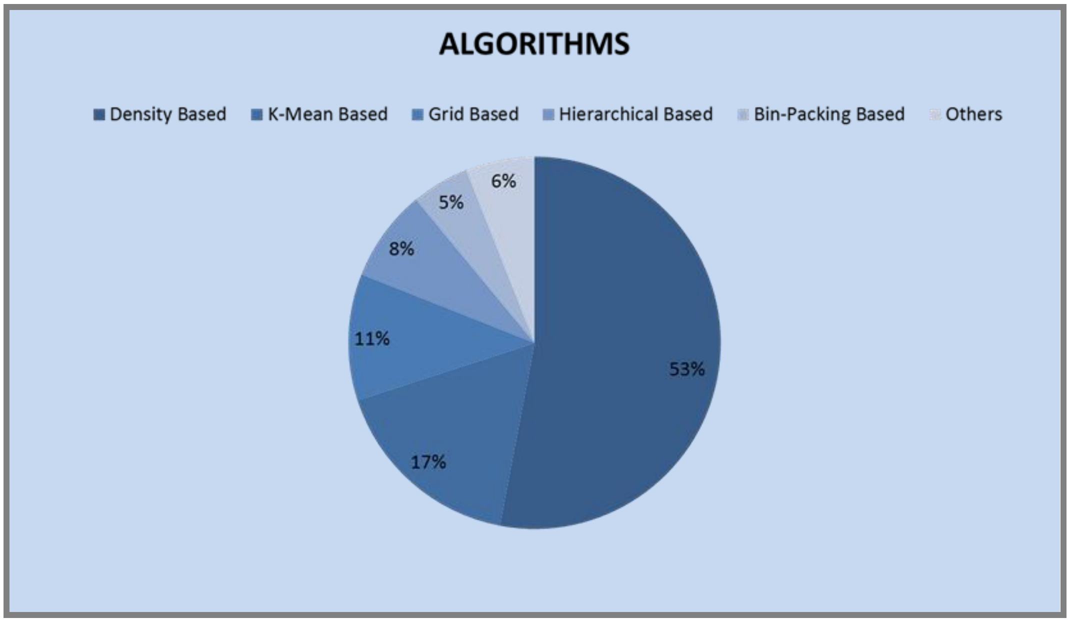
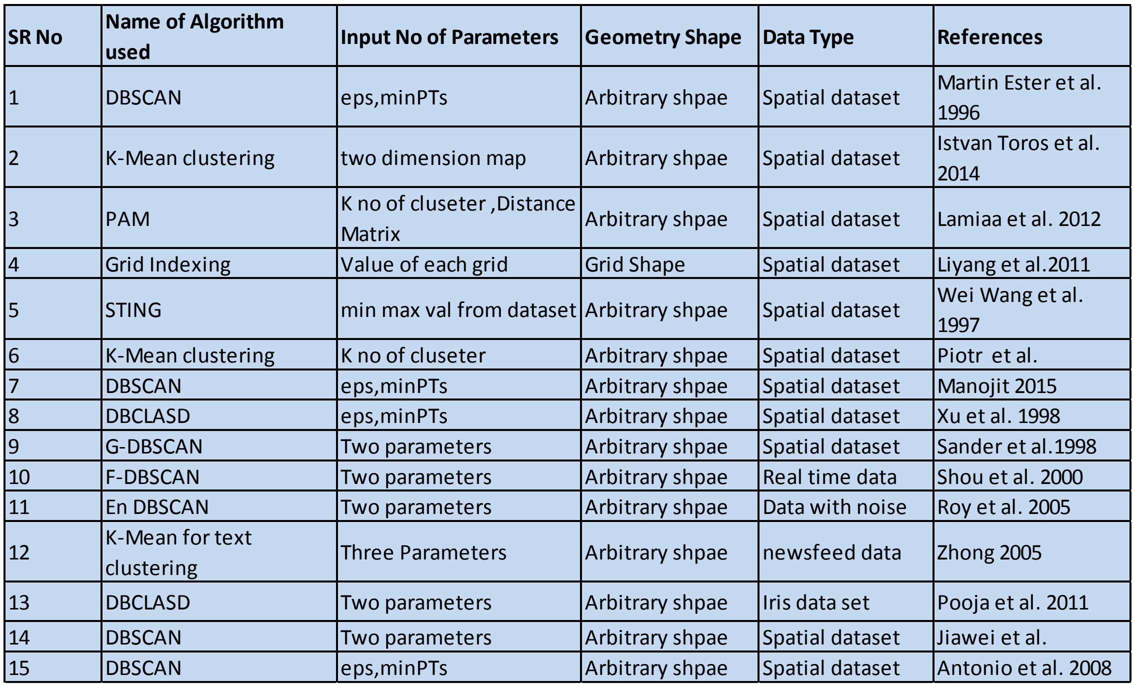


Figure 5 Summary of Literature Review

Table 4 Summary of Research Papers



SR No	Name of Algorithm used	Input No of Parameters	Geometry Shape	Data Type	References
1	DBSCAN	eps,minPTs	Arbitrary shpae	Spatial dataset	Martin Ester et al. 1996
2	K-Mean clustering	two dimension map	Arbitrary shpae	Spatial dataset	Istvan Toros et al. 2014
3	PAM	K no of cluseter ,Distance Matrix	Arbitrary shpae	Spatial dataset	Lamiaa et al. 2012
4	Grid Indexing	Value of each grid	Grid Shape	Spatial dataset	Liyang et al.2011
5	STING	min max val from dataset	Arbitrary shpae	Spatial dataset	Wei Wang et al. 1997
6	K-Mean clustering	K no of cluseter	Arbitrary shpae	Spatial dataset	Piotr et al.
7	DBSCAN	eps,minPTs	Arbitrary shpae	Spatial dataset	Manojit 2015
8	DBCLASD	eps,minPTs	Arbitrary shpae	Spatial dataset	Xu et al. 1998
9	G-DBSCAN	Two parameters	Arbitrary shpae	Spatial dataset	Sander et al.1998
10	F-DBSCAN	Two parameters	Arbitrary shpae	Real time data	Shou et al. 2000
11	En DBSCAN	Two parameters	Arbitrary shpae	Data with noise	Roy et al. 2005
12	K-Mean for text clustering	Three Parameters	Arbitrary shpae	newsfeed data	Zhong 2005
13	DBCLASD	Two parameters	Arbitrary shpae	Iris data set	Pooja et al. 2011
14	DBSCAN	Two parameters	Arbitrary shpae	Spatial dataset	Jiawei et al.
15	DBSCAN	eps,minPTs	Arbitrary shpae	Spatial dataset	Antonio et al. 2008

As a conclusion of review work of scientific research papers mentioned in table 4 which are taken from different time frame and different journals and magazine; it is found that density based algorithms specially DBSCAN has significant impact on the research in the similar domain.

Above figure 5 shows a short summary of the percentage of different type of clustering algorithms used in reviewed literatures. As clearly visible 53% of papers applied density based algorithm, 17% have applied k-mean, 11% have been applied grid based clustering algorithm, 8% have applied hierarchical based clustering, 5% papers applied bin-packing based approach and 6% papers had applied rest of clustering algorithm other than mentioned above. This in itself also shows the significant impact and importance of density-based algorithm in the scientific research community in the context of problem of the similar domain and dataset/data type. The k-means algorithm is likely the most common clustering algorithm but for spatial data, the DBSCAN algorithm is far superior. The k-means algorithm groups N observations into k clusters, however k-means is not an ideal algorithm for latitude-longitude spatial data because it minimizes variance, not geodetic distance. The algorithm would still work but its results are poor.

Instead, DBSCAN algorithm works better with arbitrary distances, DBSCAN clusters a spatial data set based on two parameters a physical distance from each point and a minimum cluster size. This method works much better for spatial latitude-longitude data.

Based on the various algorithm's advantages, disadvantages and of the type of dataset on which the algorithm is applied and final results of above said papers; I have finally selected density-based criteria and "Density-based Spatiotemporal Clustering Algorithm" is selected as a base research paper for this Thesis.

5.3 Motivation to choose DBSCAN algorithm

1. It can be used for real world and real time data. We can use it locally as well as global level. This is main reason that is very popular as compared to other algorithms. It does not require prior knowledge of number of clusters as k-mean, which is quite tough job when the dataset is large.

2. It is an unsupervised clustering algorithm, which has the ability to detect the clusters even in noisy dataset.

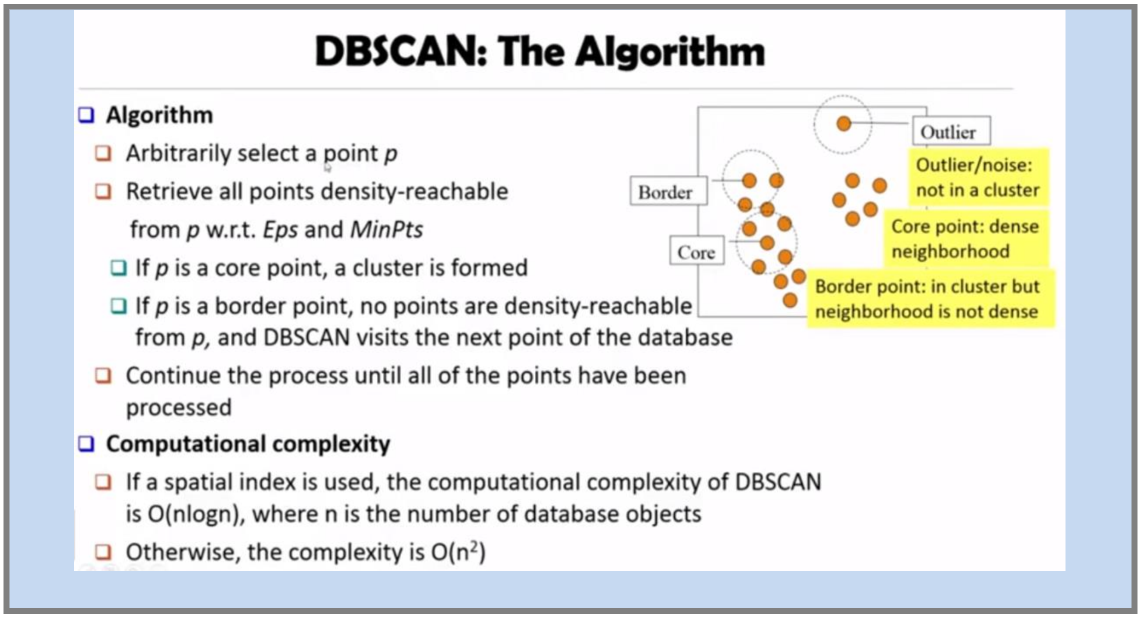
3.As we know that geo spatial data usually vary in different form and normal clustering algorithms are not able to handle the variation in such type of data.

DBSCAN is most appropriate approach to handle it. As mention above the characteristics of DBSCAN popularity is its simplicity, which motivates the scholars to work on this algorithm.

4. Last but not the least is the accuracy of this algorithm and the performance of the algorithm with large dataset including run time and output results.

5.4 DBSCAN - Density-Based Spatial Clustering of Applications with Noise

The DBSCAN algorithm views clusters as areas of high density separated by areas of low density. There are two parameters to the algorithm, min samples and Eps, which define formally what we mean when we say dense. Higher min samples or lower Eps indicate higher density necessary to form a cluster. A cluster is therefore a set of core samples, each close to each other (measured by some distance measure) and a set of non-core samples that are close to a core sample (but are not themselves core samples). More formally, we define a core sample as being a sample in the dataset such that there exist min samples other samples within a distance of Eps, which are defined as neighbors of the core sample.



DBSCAN: The Algorithm

- Algorithm
  - Arbitrarily select a point *p*
  - Retrieve all points density-reachable from *p* w.r.t. *Eps* and *MinPts*
    - If *p* is a core point, a cluster is formed
    - If *p* is a border point, no points are density-reachable from *p,* and DBSCAN visits the next point of the database
  - Continue the process until all of the points have been processed
- Computational complexity
  - If a spatial index is used, the computational complexity of DBSCAN is O(nlogn), where n is the number of database objects
  - Otherwise, the complexity is $O(n^2)$

Outlier/noise: not in a cluster

Core point: dense neighborhood

Border point: in cluster but neighborhood is not dense

Figure 6 DBSCAN (Eps, MinPts)

As figure shows a cluster is a set of core samples that can be built by recursively taking a core sample, finding all of its neighbors that are core samples, finding all of their neighbors that are core samples, and so on. A cluster also has a set of non-core samples, which are samples that are neighbors of a core sample in the cluster but are not themselves core samples. Intuitively, these samples are on the fringes of a cluster. Any core sample is part of a cluster, by definition. Any sample that is not a core sample, and is at least Eps in distance from any core sample, is considered an outlier by the algorithm.

In DBSCAN the key concept is of “data point density”, where the clusters are defined as the maximal group of dense object within a certain radius Eps and low dense areas are called noise. The beauty of DBSCAN algorithm is that it can discover clusters even in noisy dataset; noise is defined as those data points which do not belong to any generated clusters. Due to such inherent wonderful properties of DBSCAN, it is quite well used in spatial temporal datasets. In another words, a cluster is made by set of data points which has certain density which is defined by parameters “min no of points” denoted by MinPts and “radius” criteria and both of these are predefined values configured by the user.

```
DBSCAN(D, eps, MinPts) {
   C = 0
   for each point P in dataset D {
      if P is visited
         continue next point
      mark P as visited
      NeighborPts = regionQuery(P, eps)
      if sizeof(NeighborPts) < MinPts
         mark P as NOISE
      else {
         C = next cluster
         expandCluster(P, NeighborPts, C, eps, MinPts)
      }
   }
}
expandCluster(P, NeighborPts, C, eps, MinPts) {
   add P to cluster C
   for each point P' in NeighborPts {
      if P' is not visited {
         mark P' as visited
         NeighborPts' = regionQuery(P', eps)
         if sizeof(NeighborPts') >= MinPts
            NeighborPts = NeighborPts joined with NeighborPts'
      }
      if P' is not yet member of any cluster
         add P' to cluster C
   }
}
regionQuery(P, eps)
   return all points within P's eps-neighborhood (including P)
```

Figure 7 Pseudocode for DBSCAN

Step 1: In the first step, for each point p in dataset, the function DBSCAN checks whether point p is already assigned to a cluster or not.

Step 2: Then the (p,Eps) density based neighborhood of point p is obtained using the function NeighborPts if size of neighbor points in less than MinPts them marked p as noise.

Step 3: In step 2, if MinPts are presents in neighborhood then assigned it to new cluster and expand clusters with further points.

5.5 Input Parameter estimation for DBSCAN

MinPts: As a rule of thumb, a minimum MinPts can be derived from the number of dimensions D in the data set, as MinPts ≥ D + 1. Therefore, MinPts must be chosen at least 3. The larger the data set, the larger the value of MinPts should be chosen.

Eps: In general, small values of Eps are preferable, and as a rule of thumb only a small fraction of points should be within this distance of each other.

Distance function: The choice of distance function is tightly coupled to the choice of Eps, and has a major impact on the results. In general, it will be necessary to first identify a reasonable measure of similarity for the data set, before the parameter Eps can be chosen.

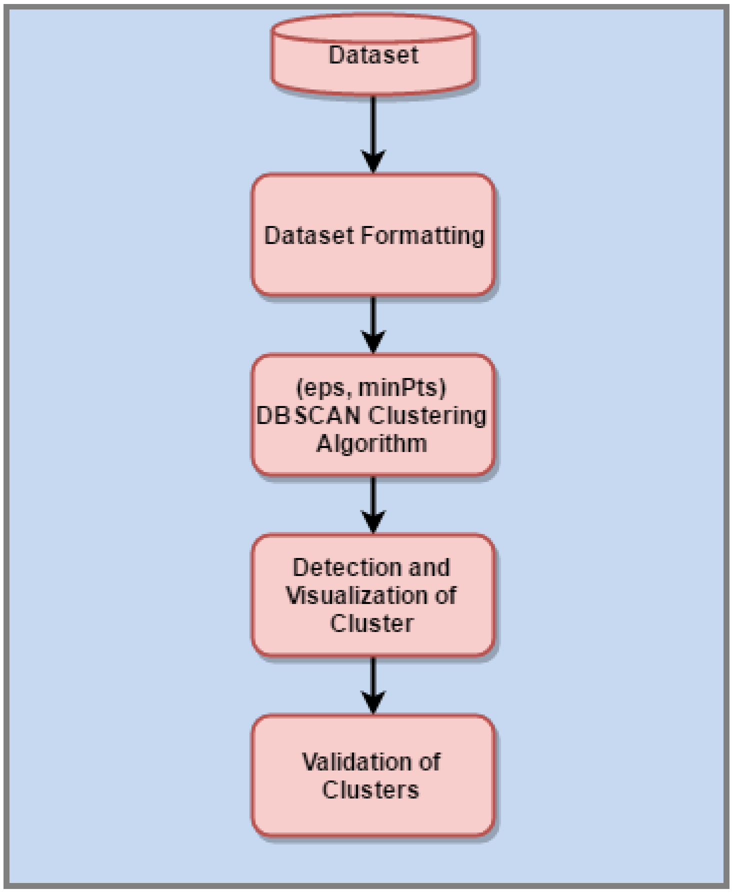


Figure 8 Workflow of DBSCAN (Eps, MinPts)

5.6 Experiment

5.6.1 Dataset

In this study, I have used the open big data that is published at the site: https://dandelion.eu/datamine/open-big-data/ under Open Data Commons Open Database License (ODbL) license. This dataset provides information about the telecommunication activities over the city of Milano. The dataset is the result of a computation over the Call Detail Records (CDRs) generated by the Telecom Italia cellular network over the city of Milano.

5.6.2 Data Preprocessing

Data preprocessing has been performed on hadoop framework using hive, I have used CDH cloudera-quickstart-vm-5.3.0-0-vmware, Cloudera's open source platform. The main steps involved are data cleaning like removal of the null records from dataset or replace null records with mode or median, process the data and find out the hidden facts from dataset in which we find out the traffic volume per squared id using hive, which in-turn involves loading the dataset into hive table and execute the hive query required to get the important facts.

5.6.3 Selection of the clustering algorithm

As mentioned before, based on the literature review it has been found that density based algorithms specially DBSCAN has significant impact on the research in the similar domain. DBSCAN algorithm works better with arbitrary distances. DBSCAN clusters a spatial data set based on two parameters namely physical distance from each point and a minimum cluster size. This method is best fit for spatial latitude-longitude data. Based on various inputs finally I have selected density-based criteria and "Density-based Spatiotemporal Clustering Algorithm" to implement "A Novel Small-Cells Planning Solution".

5.6.4 Parameter selection for proposed (Eps, MinPts) DBSCAN algorithm

In order to get the desired and accurate results, following values of different parameters are considered as show in the table below. These values are considered after multiple runs of the program during development and manual verification of output from the dataset. Also, during literature review I found out that such parameters' values might be different for different types of dataset which in itself indicate that this algorithm is sensitive to the input parameters' values, which is well the fact of DBSCAN algorithm.

Table 5 Different parameters values of (Eps, MinPts) DBSCAN Algorithm

Input Parameter Name	Various Values considered			
Radius (m)	200	500	700	1000
Min no of points	2	3	4	6
Dataset	10,000	10,000	10,000	10,000

Various different parameter combinations were prepared based on above mentioned values. They were cross checked with all the clustering results against different parameters through cross validation in weka tool and best fit combination was selected for algorithm.

Main motivation to consider these values is to minimize the noise and unclustered instances, after seeing the clustering results it is observed that with these values, algorithm majorly covered all input objects from the dataset and the unclustered instances were minimum as expected.

Also, the radius has been taken in the range 500m - 1000m. The main motivation to select this value depends upon the nature of dataset, location of dataset and the algorithm used. For current dataset in this experiment, these values are very good for Milano city to get the desired results. With these values we get very refined clusters of user activities which are visible in the algorithm output and experiment results. Also, these results are as per the expected theoretical calculation done with the domain knowledge and given parameters.

5.6.5 Selection of the open source tool for machine learning

The task of selecting machine learning tools for big data is becoming challenging with increasing amount of options. A machine learning platform provides capabilities to complete a machine learning project from beginning to end such as some kind of data analysis, data preparation, modeling and algorithm evaluation and selection. The scikit-learn in python is one of the widely used machine learning platform, it provides a wide range of supervised and unsupervised learning algorithms via a consistent interface in Python. The scikit-learn is built upon the scientific python and which includes various libraries such as NumPy, SciPy, Matplotlib, IPython, Sympy, Pandas which are very useful for any machine learning task. Based on these inputs I have selected scikit-learn as a machine learning platform for this solution.

5.6.6 Other tools used in Experiment

The details of all the tools used in this experiment including hardware and software are as follows:

Hadoop framework: CDH cloudera-quickstart-vm-5.3.0-0-vmware, Hive

Machine learning platform: Scikit-learn, WEKA.

Visualization tools: WEKA, Matplotlib, geojson.io

Language: Python version 2

Hardware used in this experiment setup was as below:

Processor: Intel(R) Core(TM) i7-4510U CPU @ 2.00GHz

RAM: 6GB

Operating System: Linux Ubuntu

5.6.7 Implementation of DBSCAN with scikit-learn and Python

Scikit-learn are simple and efficient tool for data mining and data analysis which is built on NumPy, SciPy, and matplotlib. This tool can be in various contexts classification, regression, clustering, dimensionality reduction, model selection, preprocessing.

I imported the necessary python modules and loaded the full dataset. I converted the latitude, longitude and volume into a three-dimensional numpy array called data. Next, I computed DBSCAN with the various inputs parameters mentioned in table 5.2. The figure shown below is the implementation of DBSCAN algorithm with scikit-learn module in python.

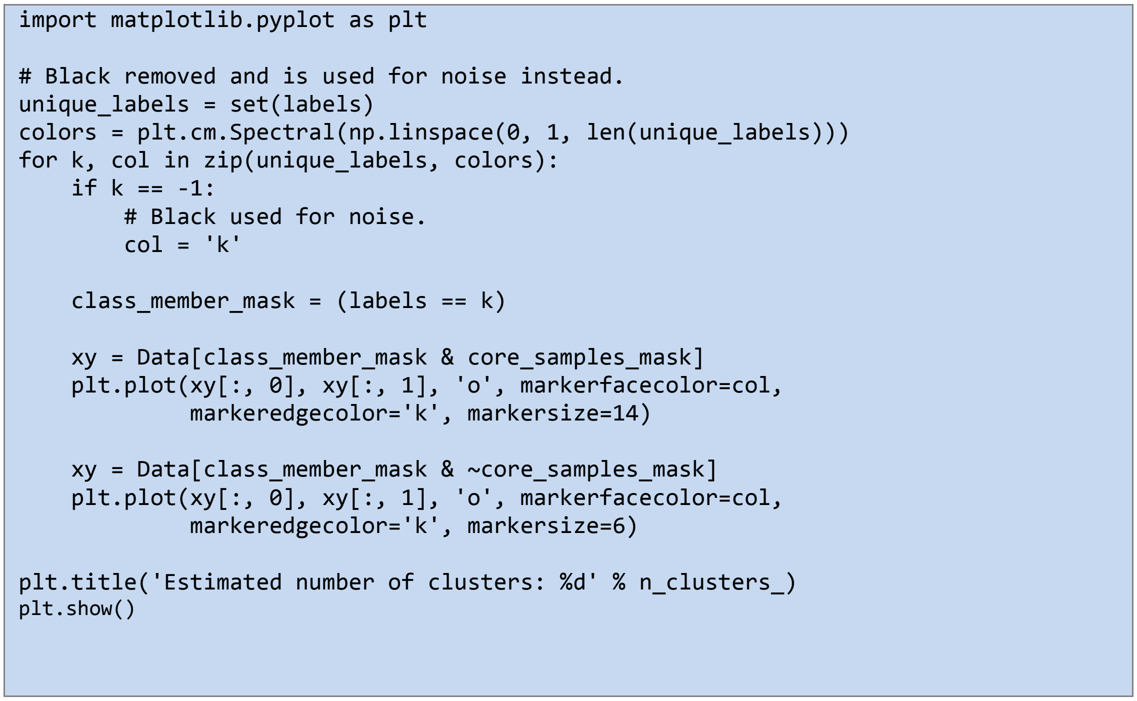
```
Import csv
import numpy
from sklearn.preprocessing import StandardScaler

dataPath      =      'C:\Python27\Lib\site-packages\DBSCAN-master      (1)\DBSCAN-
master/call1.csv'
Data = []
with open(dataPath,'rb') as f:
    reader = csv.reader(f)
    for row in reader:
        #row = re.split(r'\t+',row[0])
        Data.append([float(row[1]), float(row[2]), float(row[3])])

print Data,"original"
Data = np.array(Data)
#Data = StandardScaler().fit_transform(Data)
#print Data,"modified"

from sklearn.cluster import DBSCAN
import numpy as np
db = DBSCAN(eps=0.2, min_samples=2,algorithm='auto').fit(Data)
core_samples_mask = np.zeros_like(db.labels_, dtype=bool)
core_samples_mask[db.core_sample_indices_] = True
labels = db.labels_
print labels
#print labels
n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
print n_clusters_,"no of clusters"
from collections import Counter
Counter(labels)

#Plot result
```



```
import matplotlib.pyplot as plt

# Black removed and is used for noise instead.
unique_labels = set(labels)
colors = plt.cm.Spectral(np.linspace(0, 1, len(unique_labels)))
for k, col in zip(unique_labels, colors):
    if k == -1:
        # Black used for noise.
        col = 'k'

    class_member_mask = (labels == k)

    xy = Data[class_member_mask & core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=col,
             markeredgecolor='k', markersize=14)

    xy = Data[class_member_mask & ~core_samples_mask]
    plt.plot(xy[:, 0], xy[:, 1], 'o', markerfacecolor=col,
             markeredgecolor='k', markersize=6)

plt.title('Estimated number of clusters: %d' % n_clusters_)
plt.show()
```

Figure 9 Implementation of DBSCAN (Eps, MinPts) with scikit-learn and Python

From the below figure, which is cluster plot plotted using matplotlib, we can visualize that the algorithm generated around 800 clusters. In figure 5.7, we can clearly see the desired convex shaped clusters which are plotted with reduced dataset for better visualization.

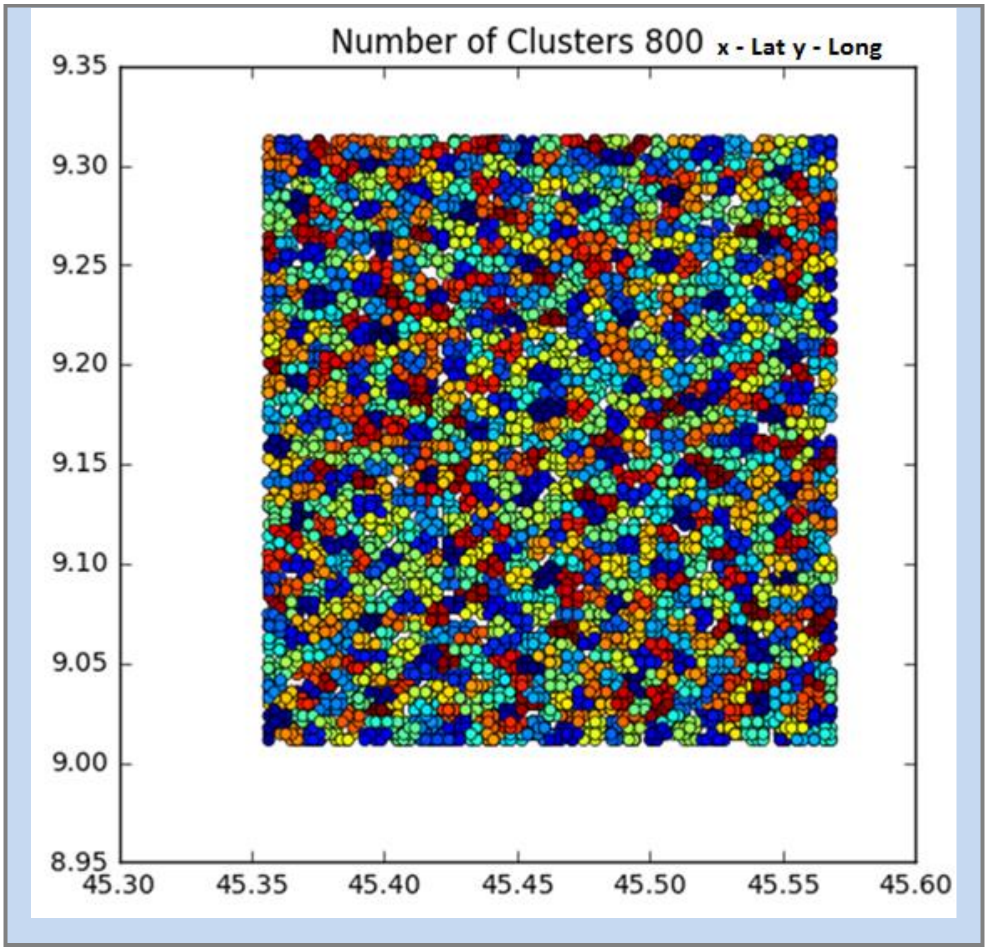


Figure 10 Clusters generated by algorithm with full dataset

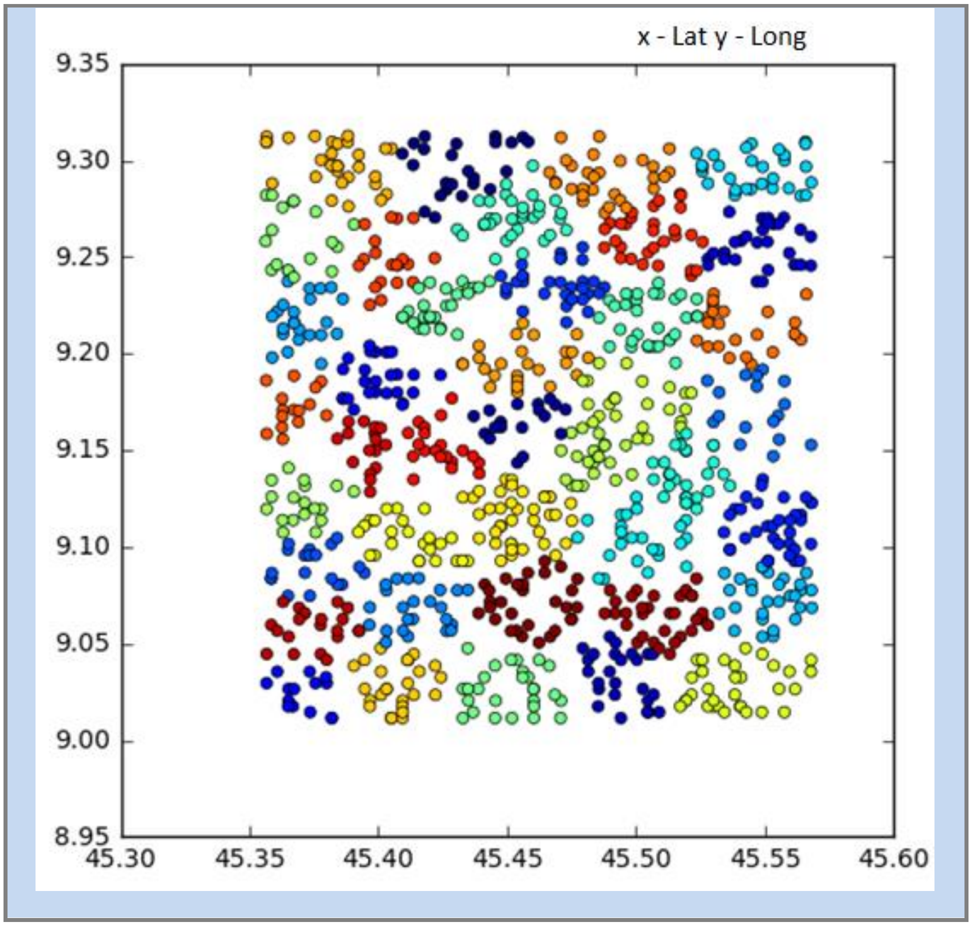


Figure 11 Clusters generated by algorithm with reduced dataset

5.7 Cluster Validation

Cluster validation is one of the important parts to evaluate the cluster results produced by (Eps, MinPts) DBSCAN algorithm. There are many methods to explore the clustering results and researchers are using

- either 2D or 3D visualization of results to verify the validity of results or
- Statistical/quantitative approach to check i.e. how well the clustering algorithm discovered the clusters from the dataset.
- manual result verification

In order to evaluate the clustering results, Weka 3.6.11, data mining tool is used. As mentioned earlier in Chapter 1, the end objectives were identified and verified below as follows:

5.7.1 Optimal no of Small cells

From the given dataset we know 100X100 Milano grid is of 552250000 meter square and if we assume the coverage of small cells between 500-1000 meter we would require around 700-800 small cells to cover whole Milano-Grid. From algorithm results we can see that 800 different clusters are generated which is approximately equal to the theoretical calculation of number of small cells.

5.7.2 Minimum Standard Deviation

The next objective was that standard deviation of traffic carried across all the clusters had to be minimum which means that the whole traffic gets distributed uniformly across all the clusters. From below table we can see that the traffic volume carried by each cluster is similar and hence the standard deviation of 15.34336862. This indicates the accuracy of proposed algorithm and the solution.

Table 6 WEKA Run Information Clusters Final Results

Cluster	Objects assigned to Cluster	Lat-Long Range	Volume per Cluster
0	6	45.4623-9.1618	210.1345
1	18	45.5667-9.2044	244
2	16	45.5238-9.2991	158.7222
3	9	45.4532-9.2788	132.625
4	19	45.3833-9.1254	129.3333
5	13	45.3829-9.1083	167.5789
6	18	45.5627-9.0786	219.9231
7	2	45.5219-9.0182	160
8	11	45.3947-9.0731	54
9	12	45.4694-9.2794	176.3636

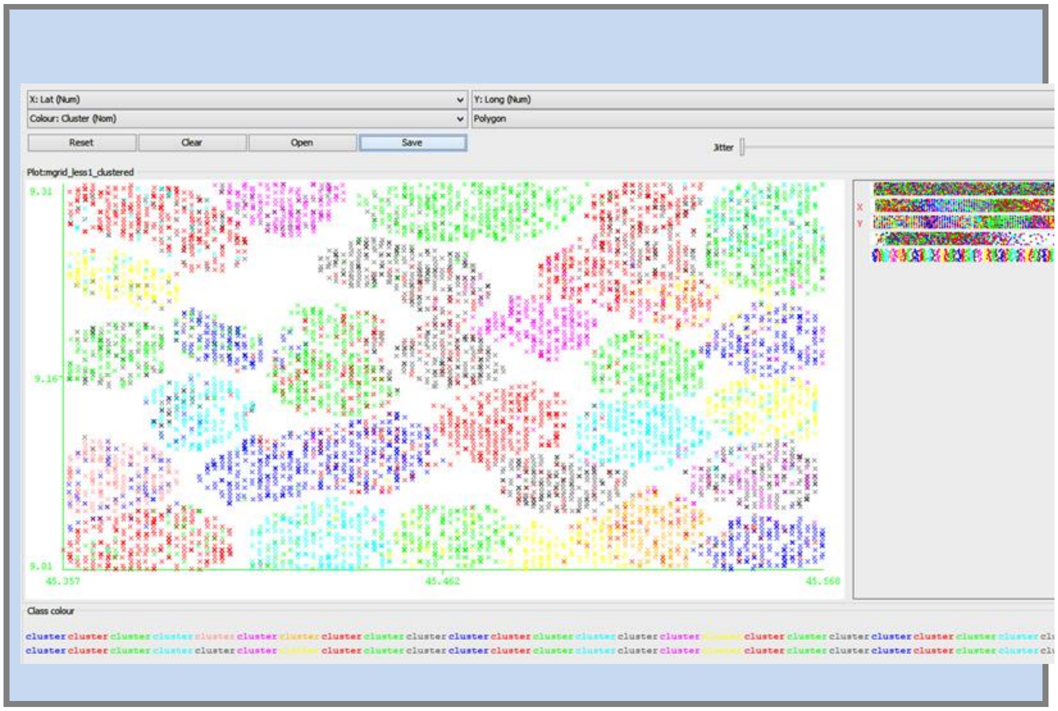
10	11	45.4511-9.2733	326.8333
11	9	45.3842-9.1924	156.1818
12	17	45.4954-9.2281	137.7778
13	9	45.5625-9.3027	158.3529
14	20	45.378-9.296	453
15	13	45.3696-9.1882	282.35
16	11	45.4931-9.1998	168.5385
17	17	45.379-9.2241	168.4545
18	18	45.4952-9.2298	201.4706
19	13	45.4196-9.1456	135.2222
20	13	45.4531-9.1936	241.3846
…			
770	17	45.5272-9.2922	118.0909
771	15	45.3861-9.035	167.7059
772	14	45.4518-9.0426	261.4
773	10	45.5391-9.1365	252.2857
774	25	45.358-9.2788	144.9
775	19	45.4295-9.1752	139.44
776	10	45.4239-9.0242	226.7895
777	22	45.5266-9.1476	330.4
778	6	45.527-9.1289	220.4545
779	4	45.5638-9.2594	231.8333
StdDev of volume across all clusters			15.34336862

5.7.3 Convex shape clusters

One of the important objectives was to get the convex shaped clusters, for telecom network planning ideally convex shaped clusters are desired. From figure 5.8, we can see that we are getting the cluster close to convex shaped.

5.7.4 Average coverage range of 1000 meters

From the experimental results we can conclude that the desired results are achieved with the input radius values between 500m-1000m.



X: Lat (Num)

Y: Long (Num)

Colour: Cluster (Nom)

Polygon

Plot:mgrid_less1_clustered

Figure 12 WEKA Cluster Visualization

5.8 Enhancements

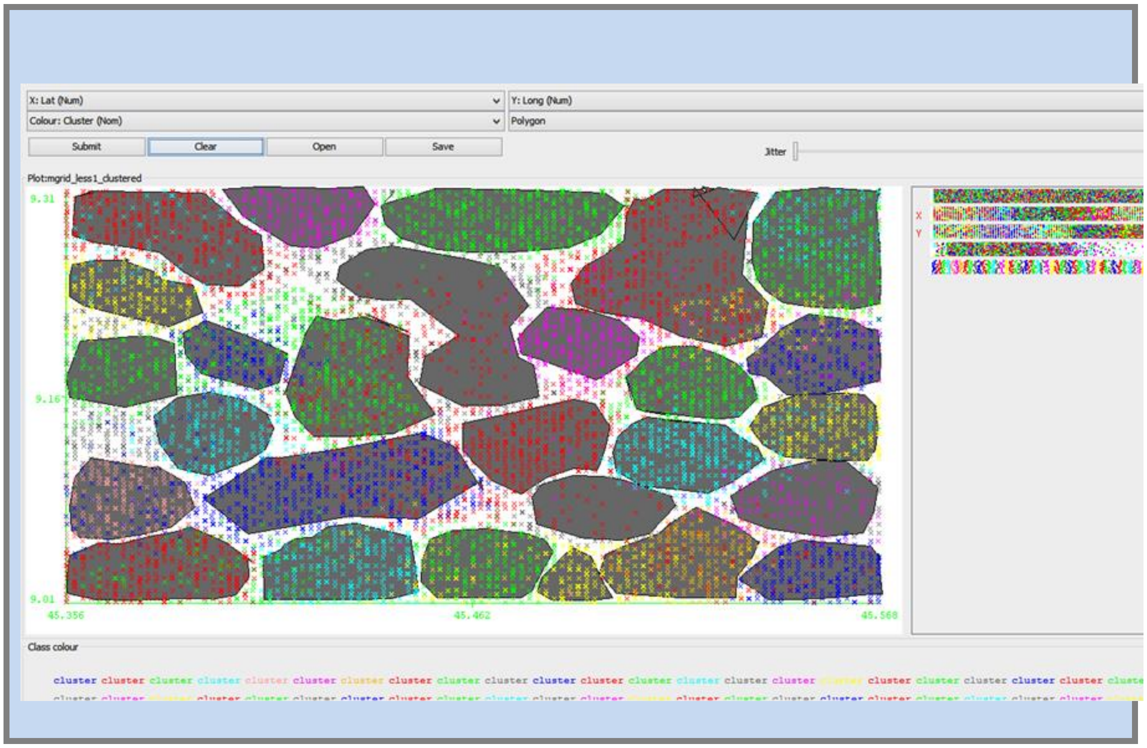
5.8.1Directional Small Cells in Millimeter Wave Cellular Networks

Based on the solution I have discussed we can efficiently plan the small cells deployment. , now we have a freedom to place a small cells but at the same time with increasing dense networks it becoming completed to select the exact location of small cells.

The Increasing bandwidth demands and available spectrum has brought millimeter wave (mmWave) bands as a candidate solution for the next-generation cellular networks. In millimeter wave highly directional transmission are essential for cellular communication in these frequencies , directional beamforming complicates the process of selecting the small cell location.

We can address this problem by extending this solution, using an extension of DBSCAN algorithm we can produced sharp contour clusters with the precise selection of input parameters, and clustering the highly dense objects.

The fig shows the arbitrary shapes of clusters with contour which facilitates to place the directional small cells for millimeter wave cellular networks.



X: Lat (Num)

Y: Long (Num)

Colour: Cluster (Nom)

Polygon

Plot:mgrid_less1_clustered

Submit
Clear
Open
Save
Jitter
9.31
9.16
9.01
45.356
45.462
45.568
X
Y
Class colour
cluster cluste

Figure 13 WEKA Generated Clusters with Contour

Chapter 6: Conclusions and Future work

The primary objective of this thesis was to get insights of spatio temporal telecom networks' data to detect meaningful traffic patterns. The traffic pattern is a result of various user activities like voice, text, and internet, which happened at a specific time or within a particular duration at a specified location.

To accomplish the desired goal and end objectives, The DBSCAN (Eps, MinPts) Martin Ester et.al (1996) was identified as the base algorithm after conducting a literature review of many scientific research papers based on machine learning and clustering analysis. The next major challenge was to select the input parameters and optimization of data sets in order to achieve the thesis's objective and desired results. In density-based, spatial-clustering algorithms, the three dimensional data has been provided and another parameters Eps and MinPts given to the algorithm. The algorithm has been run and results verified for various values of input Eps and MinPts in order to receive better and refined results and fulfill the thesis's objective. Here better and refined results means clusters results, which reveal the facts of different clusters, the optimal number of clusters required, minimum standard deviation of traffic volume across all clusters, convex shaped cluster and these results stands true for average 1000 m coverage range of small cells.

The DBSCAN algorithm is sensitive to its input parameters, hence to use this algorithm the dataset and domain knowledge is required so that inputs parameters can be selected precisely to get the desired results. During experimentation, 10,000 input parameter combinations were executed with the algorithm, and the results were further verified with the Weka tool for all input data. Also, manually and based on the domain knowledge, I have verified all the results.

In this analysis, it is evident that this input parameter with the algorithm covers all the geographical area in the form of 800 clusters from the Milano-City dataset. Whereas other approaches like bin-packing algorithms and grid-based algorithms are giving correct results theoretically but practically are not suitable for the analysis of spatial data.

The main advantages of this algorithm are the ability to extract all meaningful and hidden facts from the datasets based on the input parameters. The input parameters have a decisive impact on the cluster result.

The proposed solution can extract spatial- and semantic-based clusters that allow service providers to plan the network planning effectively. This approach can be useful for similar types of dataset, but certain extra preprocessing might be required based on the nature of the data.

During the execution of the clustering algorithm, many problems occurred. For example the precise selection of input parameters to overcome the noise or to minimize the noise, data cleaning to filter out the null and unused values from datasets, visualizing the data in order to represent the results which can be understand were among the challenging tasks.

The objective of this thesis has been achieved, but there are still a few aspects that can be considered further. As mentioned earlier, the proposed solutions can be considered for Directional Small Cells in Millimeter Wave Cellular Networks.

References

1. Elbatta MNT. 2012. An improvement for DBSCAN algorithm for best results in varied densities [disertasi]. Gaza (PS): Islamic University of Gaza.

2. Ester M, Kriegel HP, Sander J, Xu X. 1996. A density-based algorithm for discovering Clusters in large spatial databases with noise. Di dalam: Simoudis E, editor. Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining (KDD-96); 1996 Agu 4-6. hlm 226-231.

3. Gaonkar MN, Sawant K. 2013. AutoEps DBSCAN: DBSCAN with Eps automatic for large dataset. International Journal on Advanced Computer Theory and Engineering. 2(2): 11-16.

4. Han J, Kamber M, Pei J. 2012. Data Mining: Concepts and Techniques. 3rd ed. San Francisco (US): Morgan-Kauffman.

5. Usman M. 2014. Spatial clustering berbasis densitas untuk penyebaran titik panas sebagaiindikator kebakaran hutan dan lahan gambut di Sumatera [tesis]. Bogor (ID): Institut Pertanian Bogor.

6. Ankerst, M., Breunig, M. M., Kriegel, H. P., & Sander, J. (1999). Optics: Ordering points to identify the clustering structure. In ACM Sigmoid Record (Vol. 28, No. 2, pp. 49-60). ACM.

7. Bao, B. K., Min, W., Lu, K., & Xu, C. (2013). Social event detection with robust high-order coclustering. In Proceedings of the 3rd ACM conference on International conference on multimedia retrieval (pp. 135-142). ACM.

8. Berkhin, P. (2006). A survey of clustering data mining techniques. In grouping multidimensional data (pp. 25-71). Springer Berlin Heidelberg.

9. Birant, D., & Kut, A. (2007). ST-DBSCAN: An algorithm for clustering spatial–temporal data. Data & Knowledge Engineering, 60(1), 208-221.

10. Spatial Clustering Methods in Data Mining: A Survey – J.Han M. Kamber and A.K.H. Tung.

11. Spatial Ordering and Encoding for Geographic Data Mining and Visualization - Diansheng Guo1 and Mark Gahegan2 Department of Geography, University of South Carolina.

12. Enhancing Clustering Algorithm to Plan Efficient Mobile Network - Lamiaa Fattouh Ibrahim* Manal El Harby.

13. Using Hyper Clustering Algorithms in Mobile Network Planning - Lamia Fattouh King Abdulaziz University.

14. Comparative Study of Density based Clustering Algorithms Pooja Batra Nagpal Department of Computer Science NIT Kurukshetra Priyanka Ahlawat Mann Department Of Computer Science NIT Kurukshetra.

15. Clustering to Reduce Spatial Data Set Size -2016 Geoff Boeing PhD candidate, urban planning at UC Berkeley.

16. Applying data mining methods for cellular radio network planning - Piotr Gawrysiak, Michaª Okoniewski Institute of Computer Science, Warsaw University of Technology.

17. An Algorithm for Automatic Base Station Placement in Cellular Network Deployment Istvan Toros, Peter Fazekas.

18. Spatial Cluster Analysis by the Bin-Packing Problem and DNA Computing Technique - Xiyu Liu and Jie Xue.

19. Hybrid Genetic Algorithms are better for Spatial Clustering - Vladimir Estivill-Castro.

20. An Efficient Packing Algorithm for Spatial Keyword Queries Jinkun Pan1, Dongsheng Li1 , and Liming Li2