# Machine Learning based Social Network Data Analysis and Prediction for Wireless Communication Network Optimization

**Bozhong Chen**

Department of Electronic and Electrical Engineering

This thesis is submitted for the degree of the

*Doctor of Philosophy*

March 2021

I would like to dedicate this thesis and everything I do to my parents. I would not be who I am today without their love and support.

# Acknowledgements

First and foremost, I am most grateful to my supervisor, Professor Jie Zhang, whose useful suggestions, incisive comments and constructive criticism have contributed greatly to the completion of this thesis. He devotes a considerable portion of his time to reading my manuscripts and making suggestions for further revisions. His tremendous assistance in developing the framework for analysis and in having gone through the draft versions of this thesis several times as well as his great care in life deserve more thanks than I can find words to express. Besides, I would like to thank my second supervisor Dr Xiaoli Chu for her great support in my entire study.

I am also greatly indebted to all my teachers who have helped me directly and indirectly in my studies. Any progress that I have made is the result of their profound concern and selfless devotion. Among them, the following require mentioning: Dr Zitian Zhang and Dr Bowei Yang.

I thank my colleagues, Dr Hao Li, Dr Qi Hong, Dr Baoling Zhang, Dr Weijie Qi, Ms Hui Zheng, Dr Kan Lin, Dr Haonan Hu, Dr Yuan Gao, Dr Bolin Chen and Mr Bo Ma, whose brilliant ideas and perceptive observations have proved immensely constructive.

I would also like to acknowledge my indebtedness to Mr Yafeng Zhang and many others who have contributed their time, thoughts, skills and encouragement to this thesis.

I am also grateful to all my classmates and friends who have given me generous support and helpful advice in the past few years.

Last but not least, I would like to thank my family: my parents, my grandparents, for supporting me spiritually throughout my life.

# Abstract

Due to the rapid development of the wireless communication network, the total amount of data in the future is expected to triple. In the next decade, its total will grow by a factor of 1000, especially in the field of wireless communication networks. With the popularity of mobile devices and the rapid development of multimedia applications such as social networking, video sharing, and telepresence, mobile communication networks have become an integral part of people's daily lives. With in-depth research, researchers wish to find that through mining analysis, effective information and patterns can be found from mobile traffic and social network data to optimize wireless networks. This also echoes the main elaboration and research of this thesis.

Firstly, based on the geographical information of the collected Twitter traffic, the density-based noisy application spatial clustering (DBSCAN) is a traffic distribution that is relatively suitable for reality by comparing clustering algorithms. Then, a framework based on social network data that identifies and clusters mobile traffic hotspots using the DBSCAN algorithm is proposed. By comparing the hotspots of the cluster with the existing macro base station (MBSs) locations, it can be determined whether other small base stations (SBSs) need to be deployed and that such deployments can effectively improve the quality of service (QoS).

On the other hand, research focuses on the temporal and spatial dimensions of data. After partitioning the data into the grid, the region can be classified by the functions it contains. A Twitter traffic prediction framework is proposed, which aggregates geographic regions with similar traffic patterns into a group and predicts the Twitter traffic long-term and short-term memory (LSTM) network for each regional group based on each group of geographic regions. The proposed framework not only allows multiple regions to share the same LSTM prediction model but also extends the training set of each model, thereby reducing the risk of over-fitting during training.

With the study of social network data, some special crowding events will cause the population to aggregate in a small area. This phenomenon will lead to a significant increase in the total amount of traffic over a period. Thus, in another research direction. The areas affected by special crowding events by detecting and classifying Twitter traffic. Then train and test through the traffic pattern in these areas. The result is positive. The framework based on the dynamic time warping (DTW) algorithm can well determine whether an area is affected by a certain crowding event.

# Table of contents

# List of figures

# List of tables

# List of Publications

## Published

[1] **B. Chen**, Z. Zhang, X. Chu, and J. Zhang, ‘Data driven optimisation of small cell deployment,’ *IET Electronic Letter*, *vol. 56, no. 1*, pp. 48-50, 2020.

[2] B. Yang, W. Guo, **B. Chen**, G. Yang, and J. Zhang, ‘Estimating Mobile Traffic Demand Using Twitter,’ *IEEE Wireless Communication Letters*, *vol. 5, no. 4*, pp. 380-383, 2016.

[3] W. Qi, B. Zhang, **B. Chen**, Jie Zhang, ‘A user-based K-means clustering offloading algorithm for heterogeneous network,’ in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018.

# Chapter 1

# Introduction

## Overview

Due to the rapid development of the wireless communication network, the total amount of data is expected to increase by two times per year in the future. In the next decade, it will reach a 1000X growth in total, especially in the area of wireless communication networks. With the proliferation of mobile devices and the drastic growth of multimedia applications such as social networking, video sharing, and telepresence, mobile communication network-s have become an integral part of people's daily lives. This phenomenon leads to the rapid growth of wireless communication capacity as well [1]. Therefore, it becomes essential to optimise the wireless communication network to prevent this predictable problem of redundant data, and in turn, to provide a better online experience for people in the future. Small base stations' (SBSs) deployment based on the existing macro base stations (MBSs) has been proposed and researched for the mobile network [2] [3]. In addition to that, forecasting for mobile traffic can also help optimise the network [5].

Table 1.1 List of Abbreviations

| | |
|---|---|
| 5G | the 5-th Generation |
| BS | Base Station |
| MBSs | Macro Base Stations |
| SBSs | Small Base Stations |
| OPTICS | Ordering points to identify the clustering structure |
| LSTM | Long Short-Term Memory |
| PPP | Poisson Point Process |
| DBSCAN | Density-based spatial clustering of applications with noise |
| SNS | Social Network Service |
| RNN | Recurrent Neural Network |
| QoS | Quality of Service |
| DTW | Dynamic time warping |
| UE | Mobile User |
| OSN | online social network |
| PCC | Pearson correlation coefficient |
| ARIMA | Autoregressive Integrated Moving Average model |

## 1.1 Background and motivation

This chapter introduces the existing research based on mobile traffic and proposes the importance of social network data for the mobile network. The challenges of social network analysis will also be raised. It will explain the two machine learning algorithms used to analyse social network data: Clustering and RNN. Motivation will be presented at the end.

### 1.1.1 SBSs deployment based on MBSs

The most traditional MBSs deployment model is the hexagonal grid model [2]. This is a simple and effective model. It meets the coverage satisfaction of cellular network without massive modulation. However, the capacity demand is floating over time, and the assumption of this model is too ideal in reality. Thus, it is difficult and unrealistic to copy and deploy a hexagonal grid model in the real world. In recent decades, the PPP model replaced the traditional hexagonal grid model for base-stations deployment [7]. The characteristic of the randomness of the PPP model makes it a much suitable model for realistic MBSs.

On the other hand, it is because of this randomness, and the PPP model sometimes cannot correspond to the actual environment. The Poisson distribution has been questioned and challenged in recent years [8]. In [9], author Zhou proved the PPP model only suitable for the urban area with dispersive mobile users and suburban area. Except for its randomness, the human behaviours and the realistic constraint (like geographical environment) also influenced the real world MBSs deployment [8]. As a result, compared to the real-life application, this model is better for theoretical research. Improving the utilisation of the PPP model in the real world for MBSs, data from social networks need to be considered. Given the fact that more and more people now choose to use the social network for communication, and that the popularity of smart-phones is constantly increasing, it is believed that social network data analysis could make a certain contribution to MBSs deployment and optimisation.

### 1.1.2 Social network data

A social network structure is made up of many nodes. A node usually refers to an individual or an organization, and a social network represents a variety of social relationships. Before the birth of the Internet, social network analysis was an important research branch of sociology and anthropology. Early social networks mainly refer to professional networks established through cooperative relationships, such as research cooperation networks and actors cooperation networks [10].

The social network analysis referred to in this article refers to Social Network Analysis, which was born with the emergence of the SNS [11]. There are four types of online social services: instant messaging applications (QQ, WeChat, WhatsApp, Skype, etc.), online social applications (QQ space, Renren, Facebook, Google+, etc.), Weibo applications ( Sina Weibo, Tencent Weibo, Twitter, etc.), shared space applications (forums, blogs, video sharing, evaluation sharing, etc.) [11].

Social networks have four characteristics: speed, contagion, equality and self-organization. Because of these characteristics, social networks have had billions of users on the Internet for a few decades, and it has an impact on all aspects of the real world.

### 1.1.3 Unsupervised learning: Clustering algorithm

There are numerous methods to do data analysis, including regression algorithm, instance-based algorithm, decision tree algorithm, clustering algorithm, etc [12]. Considering the data analysis in the early stages, the most appropriate way to analyse the traffic with the location parameter is a clustering algorithm. Clustering is a method which separates data source into several groups [12]. Data objects will only be gathered into one group when they share certain conditions, or they are similar to each other. Different groups are formed because they have different characteristics, or the points within them are dissimilar to each other. Clustering algorithms include many specific methods such as K-means, hierarchical clustering, DBSCAN, OPTICS, etc [12]. The basic concept of clustering is to put similar things into a group, which is different from classification. Classification needs to set instructions in advance to classify various objects. Ideally, a classifier system will "learn" from the training, thereby providing the ability to classify unknown objects, which are often referred to as supervised learning.

On the contrary, clustering does not need to understand or distinguish the features of objects. The only task it has to accomplish is to cluster similar data objects together. Therefore, it is usually enough for a clustering algorithm to only have the similarity calculation process. Besides, the clustering algorithm does not require the training for similarity calculation, which is called unsupervised learning in machine learning [16].

Generally, statistical clustering is a primary type of cluster analysis. A database contains typically different features at the same time. Statistical clustering needs to cluster the data objects based on some of the features, these features are the mapping from the original database. In other words, the database will be clustered based on specific rules. The data objects which are gathered in the same group have the most similarity with each other. The most useful feature of the collected Twitter data is the geographic location. Hence, the database needs numerical clustering for geographic location claudication. The two clustering algorithms used in the research are DBSCAN [13] and Euclid distance calculation which is the one part of the process of K-means [14].

### 1.1.4 Deep leaning: RNN

The Recurrent Neural Network (RNN) is a neural network for processing sequence data. It can process data from sequence changes compared to a typical neural network. For example, the meaning of a word will have different meanings because of the different content mentioned above, and RNN can solve such problems well [16].

RNN is necessarily based on the original fully connected neural network, adding a concept of timeline. The impact is: the same data, different input order, will get different results. This effect is inherently suitable for processing data related to time sequence (timing), such as speech, text, translation, etc. In fact, with the development of RNN, some variants can even be skillfully used in image processing problems [15]. RNN is so effective because it has memory capabilities, especially LSTM (a widely used variant of RNN) that has an excellent performance in long-term memory.

Besides, compared to fixed networks, the sequence operation mechanism is destined to go through a fixed number of computational steps, so it is self-styled, so it is more attractive to those who are eager to build smarter systems. Moreover, the RNN combines the input vector with its state vector and a fixed (but learned) function to produce a new state vector [17].

### 1.1.5 Motivation

Recently, traffic data has increased rapidly over the past 20 years. People, devices and their interactions produced massive quantity information in various fields. After sort out, excavation and developed an enormous quantity of information, it gives us a new opportunity to discover the hidden value. It may help to solve research issues (find out hot-spots and black-spots) and optimise methods (like base station deployment and traffic prediction). Compared to traditional carrier data, the collection and acquisition of social network data are relatively easy. Besides, most of the social network data contain time and space parameters. These advantages are very helpful for using the data analysis of the integrated network. With the development of 5G, network slicing began to enter people's sights. Network slicing is an on-demand networking method that allows operators to separate multiple virtual end-to-end networks on a unified infrastructure. Each network slice is logically isolated from the wireless access network bearer network to the core network to adapt to various types of applications. At the moment, collecting and analyzing single application-level data is also important.

In the face of social network data with time and space parameters, a major challenge is to analyze the choice of algorithms. The clustering algorithm is choose to analyse the data with geographic parameters, which is an unsupervised method of unlabeled learning. Therefore, it is important to analyze and test the effectiveness of different clustering methods on social network data. The previous work mentioned the deployment of MBSs, but the traditional hexagonal cellular network does not match the real-life population density distribution. In reality, the deployment of MBSs does not have a very unified planning solution, which leads to an uneven distribution of resources of MBSs. Therefore, the following research will focus on the deployment of the small base station when a suitable clustering method is found out for analyzing social network data.

The pattern in time sequence is as important as the positional parameters contained in the social network data. Prediction of data in time and space can be a good help for network optimization. The predictive analysis here will use the LSTM algorithm, which is an evolution of the RNN network. We partition the geographic parameters and then find areas of similar functionality. Then, forecasting after data integration for different categories of functional areas.

The next challenge is that predictions cannot focus only on everyday events when the forecasting model can be used to predict social network traffic. Unusual crowding events like large-scale activity (Ball game, vocal concert, academic conference etc.) or a sudden attack caused a large gathering of people. Their events occur multiplier increase in social network data. We will use the detection model to detect social network traffic for different unusual crowding events. This model can filter out the regional data affected by the crowding event by comparing the regional data pattern.

## 1.2 Objectives of the thesis

Based on application-specific mobile traffic, the main objectives of this thesis listed as follows:

- To collect and organize social network data, and find a suitable clustering algorithm for social network data analysis
- To use the clustering method for social network data to find hotspots, and optimize the deployment of SBSs by finding hotspot locations (based on existing MBSs environment)
- To propose a prediction model of social network data, which combines regional classification and integration of similar regional data
- To test model's prediction effect on different unusual crowding events

## 1.3 Structure of the thesis

The structure of this thesis is organised following layout:

### Chapter 2: Literature review

This chapter presents the existing MBSs and SBSs deployment model research, the overview of social network data, the most typical clustering algorithms background, and the RNN and LSTM achievements in an wireless communication network.

### Chapter 3: Twitter traffic analysis and clustering algorithms evaluation

The clustering algorithm is the best choice for processing data with geo-location parameters. There are two proposed clustering methods for Twitter traffic: partitioning methods K-means based Euclidean distance clustering algorithm and DBSCA. This chapter mainly analyzes the effects of the two clustering methods, and the central point of the clustering is reversed by the Google Maps API. It is the geographical location to view the real location attribute of the cluster centre point. Experiments showed that density-based methods are more suitable for Twitter traffic. The clustering algorithm is the best choice for processing data with geo-location parameters. The overview of Twitter traffic is mentioned and proposed two clustering methods: partitioning methods K-means based Euclidean distance clustering algorithm and the DBSCAN. This chapter mainly analyzes the effects of the two clustering methods, and the Google Map Geo-coding API reverses the central point of the clustering. It is the geographical location to view the real location attribute of the cluster centre point. Experiments show that Twitter traffic is more suitable for density-based methods.

**Contributions:** 1) Partitioning methods K-means based Euclidean distance clustering algorithm and the DBSCAN are evaluated. 2) The clustering effect and the density map of the cluster are visualized. 3) The Google Map Geocoding API confirms that the central point of the cluster is relatively dense and busy in the real world. 4) By observing the distance between the overall Twitter traffic and the cluster centre point, it proves the effectiveness of clustering in finding hot spots.

Twitter traffic is a representative social network data. In wireless communication, social network data is more representative than operator data, and its pattern has a strong positive correlation with operator traffic. It is meaningful for base station deployment to study the geographic location information contained in social network data. Moreover, clustering can better find the high traffic area and get the area attributes through Google map API. Thus, the contribution of this chapter is helpful to later research as SBS deployment and traffic region segment.

### Chapter 4: Data driven optimization of small cell deployment

Increasingly popular online social networks contribute to the massive growth in mobile traffic. The social network data is easier to obtain than carrier data, it can be used optimise the deployment of SBSs. Based on the research in this chapter, the clustering algorithm is suitable for social network data to cluster the hot-spot. Thus, the research results of the previous chapter are used for the deployment of SBS. In this chapter, a framework based on analyzing social network data is proposed to identify and cluster mobile traffic hot-spots by using the DBSCAN algorithm. By comparing the clustered hot-spots with the existing MBSs locations, additional SBSs need to be deployed and show that such deployment can effectively improve the QoS. Our simulation results show that the way deploys SBSs around hot-spot has relatively better user received signal power on mobile communication network than randomly SBS deployment following the normal distribution or the Poisson distribution.

**Contributions:** 1) Twitter traffic, existing MBSs' locations and clustered hotspots are visualized. 2) SBSs deployment framework is proposed, which is based on analyzing social network data to identify and cluster mobile traffic hot-spots by using the DBSCAN algorithm. 3) The number and location of mobile users based on Twitter traffic are simulated by the relationship between Twitter traffic and real 3G users. 4) Calculate the user received signal power as a reference for the SBSs deployment effort.

Traffic distribution is uneven in real wold. Hot spots can be found by clustering traffic, which is meaningful for SBS deployment. Compare Poisson distribution and normal distribution SBS deployment, locate hot spots through clustering and deploy SBS is more fit to the real data distribution. Evaluate three different SBS deployment models, hotspot cluster deployment SBS has the greatest improvement to user's received signal power.

### Chapter 5: Deep learning enabled prediction of Twitter traffic with a regional granularity

The research in the previous chapter focuses on social network data with geo-location. However, the social network data also has a temporal pattern. Because the existing research has used LSTM's research on operator data. So the research in this chapter uses different data sources which are collected from the social network. Besides, the spatial analysis of data also used geographic location information. Application-aware network slicing requires an accurate and high-resolution forecast of application-specific mobile traffic. In this chapter, the temporal-spatial characteristics of Twitter traffic with a regional granularity is investigated and a Twitter traffic prediction framework is proposed, which clusters the geographical regions with similar traffic patterns into a group and predicts Twitter traffic for each group of regions based on an LSTM network. The proposed framework not only allows multiple regions to share the same LSTM prediction model but also expands the training set for each model, thus reducing the risk of over-fitting in the training process. Experiment results using Twitter records collected in London and the surrounding suburbs show the validity and efficiency of the proposed traffic prediction framework.

**Contributions:** 1) The framework based on LSTM algorithm is proposed. 2) The framework achieves the function of grouping data of similar functional regions by correlation coefficients. 3) The validity and efficiency of the proposed traffic prediction framework were verified by predicting the Twitter traffic collected by London. 4) The proposed framework achieves multiple regions sharing the same LSTM prediction model, and also extends the training set for each model.

Existing research has used LSTM to predict various data with time series. However, in wireless communication, it is very time-consuming to use LSTM to perform traffic prediction for every single area. Traffic has very similar patterns in areas with similar attributes. The traffic in these areas with similar attributes can be clustered in a group and use the same LSTM model to predict the traffic in these areas. This can not only predict traffic more effectively but also increase the training set data in the LSTM model. Because the traffic in the similar attributes area has similar patterns and these data can be used in an LSTM model at the same time.

### Chapter 6: Prediction of Twitter traffic with unusual crowding event by deep learning

With the study of social network data, we believe that some special crowding events will cause the population to aggregate in a small area. This phenomenon will lead to a significant increase in the total amount of traffic over a period. The traffic prediction research in the previous chapter did not distinguish special events. But there are many special events in real life, and these special periods will affect traffic pattern. Thus, this chapter finds areas affected by special crowding events by detecting and classifying Twitter traffic. Find a standard traffic pattern by expanding the traffic of the crowding event in the time series. The Dynamic Time Warping (DTW) algorithm was proposed in 1970 to be an algorithm that was applied to compare the similarity of two-time series [42]. The DTW algorithm can well determine whether an area is affected by a certain crowding event and distinguish which of the two crowding events the traffic pattern in this area belongs to.

**Contributions:** 1) The classification framework based on DTW algorithm is proposed. 2) Experimental results show that special crowding events affect regional traffic patterns. 3) The framework trained to find out whether the traffic pattern in a region is affected by a crowding event.

In daily life, special events will affect the traffic pattern. Existing research and traffic prediction are based on daily traffic. Therefore, the classification of traffic patterns for specific different crowding events is meaningful. This research is helpful for future special events classification and the specific crowding event traffic prediction.

### Chapter 7: Conclusions and future work

The concluding remarks are presented with directions for the future work.

## 1.4 Published paper

[1] **B. Chen**, Z. Zhang, X. Chu, and J. Zhang, ‘Data driven optimisation of small cell deployment,’ *IET Electronic Letter*, *vol. 56, no. 1*, pp. 48-50, 2020. Main contribution: Collect data, implement algorithm, complete paper.

[2] B. Yang, W. Guo, **B. Chen**, G. Yang, and J. Zhang, ‘Estimating Mobile Traffic Demand Using Twitter,’ *IEEE Wireless Communication Letters*, *vol. 5, no. 4*, pp. 380-383, 2016. Main contribution: Collect data, implement algorithm.

[3] W. Qi, B. Zhang, **B. Chen**, Jie Zhang, ‘A user-based K-means clustering offloading algorithm for heterogeneous network,’ in *2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC)*, 2018. Main contribution: implement algorithm

# Chapter 2

# Literature Review

## Overview

This chapter first describes the existing BS deployment techniques and research. Then describe the social network data source for this thesis, which is Twitter traffic. In order to dig deeper into the information of social network data, some algorithms for machine learning are used for this purpose. The latter part of this chapter describes Unsupervised learning and Deep Leaning for machine learning. The technologies include K-means and DBSCAN in clustering algorithm; and RNN and LSTM in neural networks.

## 2.1 Reviews of base station deployment

### 2.1.1 Overview of micro base station deployment

More and more people have started using smart mobile devices to connect through online social networks. It contributes to the rapid growth of mobile traffic, which is pushing the existing cellular networks to their limits. With the rapid development of smartphones and

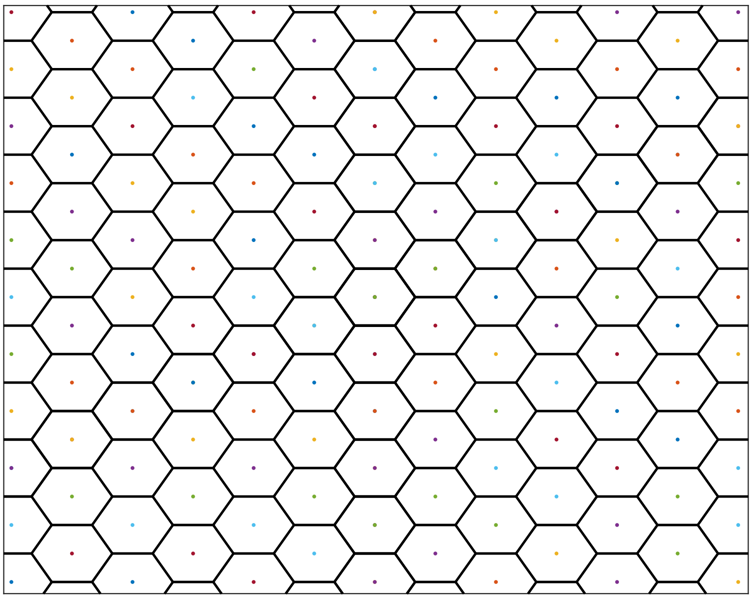


Fig. 2.1 Theoretical Macro Base Station: This is a example of hexagonal cellular MBSs deployment. This structure achieves complete coverage and save the number of theoretically. The centre point represents the MBSs and the distance between each MBS is one unit.

social networks, the use of traffic data has dramatically exceeded voice calls and SMSs services.

As social networks start to draw public attention and people produce a large number of data daily. Literature in the past had suggested discovering useful information from social networks. Data analysis has now been used in various field, especially social networks. The traditional data analysis follows the sequence of collecting, managing, and mining data. According to the statistics accounting, there are more than 1.71 billion monthly active users on Facebook, and more than 310 million monthly active users on Twitter. Massive data have been produced in daily life [36] [11]. People are used to publishing interesting photos and their feelings on social media. Besides uploading their personal lives on the internet, an increasing amount of people now like to browse, share, and "like" that information on social

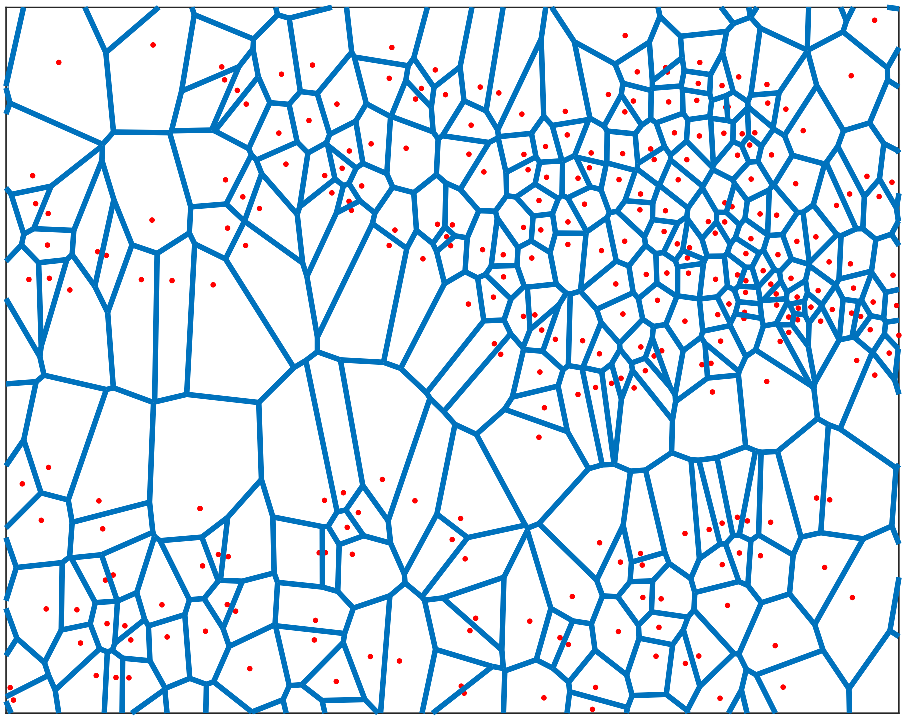


Fig. 2.2 Real World Macro Base Station: This is a example of real world MBSs deployment. The red point represents the MBSs and the distance between each MBS is not even. The regions represent the coverage of each MBS. The MBSs in real world is not like the theoretical MBSs deployment.

media. As a result, analysing the data from social networks have a practical application and theoretical significance.

### 2.1.2 Existing small base station deployment research

With the rapid development of wireless communication networks, it is difficult to migrate or change the distribution of already deployed MBSs. So deploying SBSs in an MBSes-based environment has become a trend [34]. However, the spatial distribution of mobile traffic in realistic scenarios is extremely uneven, especially for urban areas where various regions (e.g., shopping mall, central business district, rail station, university campus, etc.) have quite different population densities and communication demands [32]. How to plan and deploy SBSs in advance to provide satisfactory services for a colossal amount of mobile users becomes an essential but challenging problem.

Some initial efforts have been made efficient accommodation of exploding wireless communication network. The joint deployment of macro base stations (MBSs) and SBSs over the urban area is presented in [33]. Author Byungchan studied the traffic distribution of mobile network and proposed a novel characterization which is the joint deployment of MBSs and SBSs to meet the fast-growing demands of mobile users of the urban spatial traffic distribution. The deployment of SBSs in scenarios where MBSs are given in advance is investigated in [34] and [35], respectively. In [34], author Xu proposed cooperative distributed of hyper-dense SBSs which can alleviate the energy consumption of the mobile devices and MBSs. In [35], author Pak proposed the cell-edge deployment of SMSs based on the performance of signal strength enhancement and the interference of other SMSs. However, the user in [35] is uniformly distributed and this is doesn't match the uneven distribution of users in reality. Nevertheless, the works in [33] [34] need to obtain realistic mobile network traffic records. But these records are usually held by diverse operators and are not easy to acquire in practice. Therefore, mining traffic data collected to optimise the wireless communication network has become especially important.

## 2.2 Reviews of Twitter traffic

Many researchers have done related social networks analysis [36]. These researches are mostly related to the hot degree of online topics, like mining, information predicting, and the user's relationship of social networks communication (like friends, parents and tutors) [37]. There is a research that studied a social network named "MicroBlog" in China, which is a Chinese version of Twitter. [37]. Author Zhao indicated that social networks are becoming the most popular platform for people to communicate with others during spare time. Sharing interesting topics, daily life pictures, entertainment news and sports news are the primary purposes of social media. Social network data is very close to real life and analysing these data from is practical application significant. Author Zhao discussed three related themes, which are the process of social networks data collection, the real-time data management, and the hot degree of topics on "MicroBlog" [37]. According to the paper, when dealing with a massive amount of social network data, the collection is just the first step of data analysis. Following the collection part, managing and filtering the database is the next essential step. Then, the last and most critical step will be mining useful information and final data analysis.

As mentioned before, data management for social network services is an essential part. A structured database is better for storage and mining. The paper [36] focused on the management and the storage of social network data. It lists out the main features of data from two social networks, Twitter and Facebook. Besides, the paper [36] demonstrated the similarities and differences between them. Table 2.1 presents some useful information includes the features, the number of users of Twitter and Facebook, and the amount of data produced from Twitter and Facebook. It can be seen from the table that there are more daily and monthly active users on Facebook than on Twitter, which is due to the different characteristics that the two social networks have. It also shows that Facebook has more diversification than Twitter. The posts on Twitter are called 'tweets'. Although tweets already contain many forms of information such as text, picture, and video, Facebook provides more ser-

| No. | Parameters | Twitter | Facebook |
|---|---|---|---|
| 1 | Daily Active Users | 100 Million | 1.13 Billion |
| 2 | Monthly Active Users | 310 Million | 1.71 Billion |
| 3 | Features | -Tweet<br>-Pictures<br>-Video and link<br>-Follow and Retweet | -Content<br>- Pictures<br>-Video and link<br>-Friends<br>-Groups<br>-Chat<br>-Apps<br>-Games |
| 4 | Data Produced | -6000 Tweets per second<br>-0.35 Million Tweets per minute<br>-500 Million Tweets per day<br>-200 Billion Tweets per year | -250 Million posts per day<br>-300 Million pictures uploaded per day<br>-2.5 Billion items shared per day<br>-Over 2 Billion Likes per day |

Table 2.1 Comparison Between Facebook and Twitter [11]

vices like messages, games, and groups etc. Different functions will produce different types of data, which resulted in the differences in the number of active users between two social networks. The comparison indicates that Facebook is more popular and entertaining than Twitter, and it means that the data produced by Facebook are more complex and diverse than the data produced by Twitter. On the other hand, Twitter data is relatively simpler than Facebook data. In other words, Data collected from Twitter are simpler structured, and the data produced have fewer content types. Hence, it is more convenient to collect and analyse the data from Twitter during the early stage of the research.

In [56], the work proves the strong positive correlation between Twitter traffic and mobile network data. Author Yang described the log-linear relationship between the two types of data as equation (2.1), where the $a_{UL}$ is 0.86 kb/Tweet and the $b_{UL}$ is 1.97 kb/s. r (kb/s) is the estimated traffic load, and n(Tweets/s) is the Twitter traffic load. This log-linear relationship is used to simulate mobile network data which is got from Twitter in different London wards(wards are defined by the London government).

$$log_{10}(r) = a_{UL} * log_{10}(n) + b_{UL} \tag{2.1}$$

## 2.3 Reviews of clustering algorithms

### 2.3.1 Supervised learning and unsupervised learning

Supervised learning is to know the relationship between input and output results based on the existing data set. According to this known relationship, an optimal model is obtained through training. In supervised learning [18], a function (model parameter) is learned from a given training data set. The result will be predicted based on this function when a new input is loaded. The training set requirements for supervised learning include input and output, which can also be said to be features and labels. The labels in the training set are labelled by people. Supervised learning is the most common classification problem. A function will learn through the existing training samples (samples include input data and its corresponding output) to train an optimal model. This optimal model has the best performance under a certain evaluation standard. Then use this model to map all inputs to corresponding outputs, and make simple judgments on the output to achieve the purpose of classification.

Supervised learning based researches are already used in information research system and information filtering. Researches in [19], multimedia personalization is dependent on user's performance learning. Author Yu proposes a preferred learning method based on Master-Slave architecture for acquiring and updating multimedia personalized users in a popular computing environment. Author Chen studied the Youku video's content-requesting by using a neural network to predict a user's request in [20]. A novel algorithm based on the machine learning framework of concept-based echo state networks (ESNs) is proposed in this paper. Vector of user context is the labelled input of this network and the user's context request is the output of this network. Author Chen used the same network jointly incorporates backhaul and fronthaul loads and content caching as an optimization problem in [21].

Regression and classification are the most common of supervised learning. Author Szabo found out Youtube video's popularity logarithmical transformed has a strong linear correlation between early and later time [22]. Furthermore, author Ahmed used classified the types of content based on the evolution of its popularity over time by counting the number of hits it receives and the value of future popularity of content can predict base on the classification [23]. Author Tanaka in [24] proposed an example that uses a naive Bayes classifier to identify stable and highly popular YouTube videos based on popularity patterns and content request time.

In unsupervised learning [18], the input data is not marked, and there is no definite result. Because the type of sample data is unknown. It's necessary to classify the sample set according to the similarity between the samples to minimize the gap between data in the same class and maximize the gap between classes. In practical applications, the label of the sample cannot be known in advance in many cases. So the model can only be learned from the original sample set which is based on the training sample without the corresponding category. The goal of unsupervised learning is to let the computer learn how to do it on its own.

Unsupervised learning is a method of training machines to use data that is neither classified nor labelled. This means that training data cannot be provided, and the machine can only learn by itself. The machine must be able to classify the data without providing any information about the data in advance. So unsupervised learning has no training set, only one set of data. However, a supervised learning method must have a training set and test samples. Find a pattern in the training set, and use this pattern on the test sample. The idea is to put the computer in contact with a large amount of changing data and allow it to learn from this data to provide previously unknown insights and identify hidden patterns. Therefore, unsupervised learning algorithms do not necessarily have clear results. Rather, unsupervised learning can find that the data set exhibits a certain aggregation, which can be classified according to the natural aggregation. However, it is not to assign certain pre-classified labels to numbers.

### 2.3.2 Unsupervised learning: Clustering

Clustering is one of the most common unsupervised learning methods [18]. The method of clustering involves organizing unlabeled data into similar groups, called clustering. Therefore, a cluster is a collection of similar data items. The main goal here is to find similarities in data points and group similar data points into a cluster. Anomaly detection is a method for identifying data that are significantly different from most data. Usually, the reason for looking for anomalies or outliers in data is that they are suspicious.

A significant amount of various and heterogeneous sensors [25] with different functions and capacities consist of a wireless sensor network [26]. A wireless sensor network is a powerful tool for many realistic applications in the field of environment monitor [27] and target detection [28]. Author Deng summarized four typical coverage model of wireless sensor network's coverage ability [29]. Furthermore, author Deng concludes coverage optimization problem in three main directions which are the lowest cost of deployment, the longest network lifetime and the detection and healing of coverage black hole. The clustering algorithm also used in the study of a wireless sensor network. Author Chen proposed a novel framework to cluster the sensors according to their information content [30]. It's successfully for sensor clustering and communication efficient field reconstruction. Moreover, the framework in [30] reduced the communication cost. The study of base station also included deployment and coverage problem. The social network data mentioned in the front is a non-labelled data. Thus, unsupervised learning clustering is suitable for social network data analysis.

### K-means

K-means clustering algorithm is a typical distance-based clustering algorithm, the distance between two objects stands for the similarity evaluation index [38]. The closer the distance, the greater the similarity. The mathematical expression formula of K-means is represented as following [38]:

$$min\sum_{i=1}^{K}\sum_{x\in(C_i)}||x-u_i||^2 \tag{2.2}$$

$$u_i=\frac{1}{|C_i|}\sum_{x\in(C_i)}x \tag{2.3}$$

$u_i$ is the mean vector of cluster $C_i$. i is the order number of clusters $C_i$. x represents the points in cluster $C_i$. K is the number of the randomly selected cluster's centre points.

K-means algorithm is to divide similar points into different groups or more subsets by static classification method. Thus, the member points in the same subset have some similar attributes(closer to the centre point of a cluster). The goal is to group and allocate data into different clusters through distance. The first step of this algorithm is to set a number of K cluster and to give a maximum radius for those centre of clusters. Thus, the selection of the number of K clusters and the value of the radius will have great effects on the final results. This algorithm will recalculate the distance between each object and the centre of the cluster, and then reassign the objects to the closest cluster. The centre object of each cluster will be recalculated after each iteration. When the results of centre objects stay unchanged, and all objects are allocated in the same cluster as the last iteration, the algorithm is completed, and the results are converged. It means the position of the cluster centre will not change after iterations.

The Euclidian algorithm is part of K-means. It inherits the distance-based clustering process, but abandons the process of repeated iteration and setting the number of clusters in advance. The reason for eliminating some processes of K-means is to reduce the limitation, which is the number of clusters needs to be set in advance. Moreover, there will be a great influence on clustering results of setting the number of clusters in advance. By abandoning these processes, the number of clusters would not need to be defined at the beginning when analysing the real dataset. The Euclid distance algorithm only does iteration for one time, and the centre of clusters will be selected randomly. Objects will be allocated in the cluster when the distance from the centre of the cluster is less than the given radius at the beginning. The position of all objects in the same cluster will be recalculated to find out the most proper centre of the cluster. Even if lacking repeated iteration may cause inaccuracy in the final results, this new method is more convenient and will be better for analyzing datasets with geographic location. Compared to the K-means, this method no need to set the number of clusters in advance, which will lead to inaccurate results.

The principle of K-means is simple and easy to implement. However, this algorithm has some shortages. First of all, the number K of cluster centres needs to be given in advance, but it is difficult to estimate the K value. In many cases, it is hard to know how many categories of a dataset should be divided into. Secondly, K-means needs to manually determine the initial cluster centres. Different initial cluster centres may lead to completely different clustering results. When the data is relatively large, the algorithm consumes a lot of time to cluster.

### DBSCAN: Density-based spatial clustering of applications with noise

DBSCAN is the short term of density-based spatial clustering of applications with noise. This is a spatial clustering algorithm based on the density of database [39]. DBSCAN algorithm divides the regions into different clusters according to the level of database density. The density of the database means the number of points per unit area. Data objects with insufficient densities will be filtered out as the noise. In the basic theory, DBSCAN algorithm marks all points as core point, boundary point and noise point [39]. Moreover, a distance E (a value of distance) will be set in advance and it needs to be set according to the area of the data group and the experimental background (one week) Data objects within that distance E will be merged into the same cluster. Any boundary points that are close enough to the core point will also be allocated into the same cluster. Therefore, the main result of the DBSCAN algorithm is the largest cluster produced which consists of a set of points sharing similar densities.

DBSCAN is a clustering algorithm, and it also has two adjustable parameters. There are some definitions of DBSCAN:

- Distance E: the radius of core points or the maximum distance of the surrounding points from core points
- MinPts: the minimum number of surrounding points of a given centre point;
- Centre points: if the number of sample points in the neighborhood of a given object is greater than or equal to MinPts, the object is called the core object;

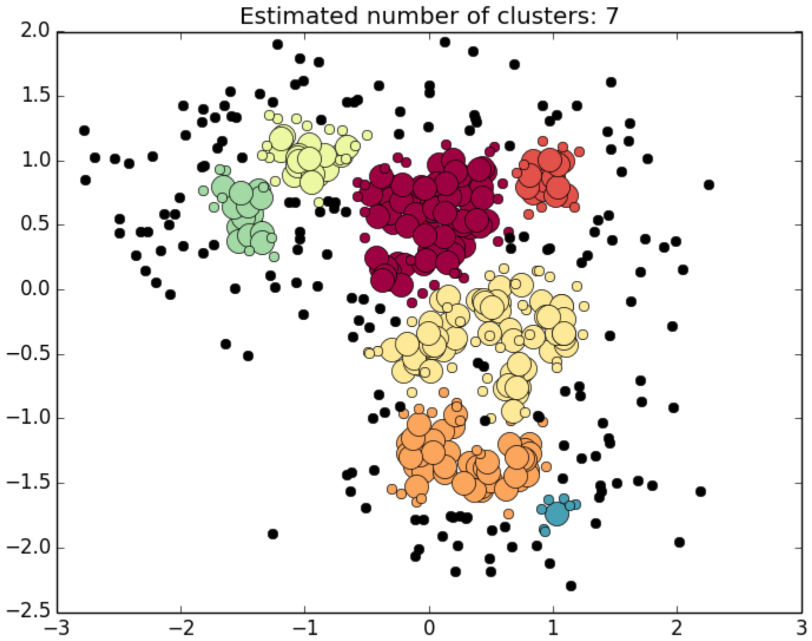


Fig. 2.3 DBSCAN sample: This is an example of DBSCAN clustering results. The value of points are selected randomly from -3 to 3 in this example. Different color of points represent different clusters. The big color points are the core points and small color points are density-reachable to the core points. Density-reachable means for sample set $D$, given a sample of points $p_1, p_2...p_n$, $p = p_1$, $q = p_n$, the object $q$ is reachable from object $p$ if the object $p_i$ has a direct-density from $p_i - 1$. The black points are noise in this sample.

- Directly density reachable: for sample set $D$, if sample point $q$ is within the domain of $p$, and $p$ is the core point, then the object $q$ is directly reachable from object $p$;

- Density-reachable: for sample set $D$, given a sample of points $p_1, p_2 ... p_n$, $p$ = $p_1$, $q$ = $p_n$, the object $q$ is reachable from object $p$ if the object $p_i$ has a direct-density from $p_i - 1$;

- Density-connected: there is a point $o$ in the set $D$, and if the object $o$ to the object $p$ and the object $q$ are density-reachable, then $p$ and $q$ density are associated.

It can be found that density-reachable is not an asymmetric relation. Density-connectedness is an asymmetric relationship. The purpose of DBSCAN is to find the largest set of density-connected objects [39].

Compared with the traditional K-Means algorithm, the biggest difference of DBSCAN is that there is no need to enter the category number K in advance. At the same time, it can also find abnormal points while clustering, and it is not sensitive to abnormal points in the data set. However, this algorithm still has some shortage in some cases. The clustering quality is poor when the density of spatial clustering is not uniform. Besides, more memory is required to support I/O consumption when the dataset is relatively large. Moreover, the clustering effect of the algorithm depends on the selection of the distance formula. Euclidean distance is commonly used in practical applications.

## 2.4 Reviews of recurrent neural network

### 2.4.1 RNN: Recurrent neural network

Human thinking about each problem generally does not start from the beginning. Traditional neural networks can't do this either, and it seems like a huge drawback. For example, suppose you want to classify the type of data at each point in time for an event. One of the critical aspects of RNN is that they can be used to connect previous information to the current task [40]. RNN is a type of recursive neural network that takes sequence data as input, recurses in the evolution direction of the sequence, and all nodes (recurrent units network) are connected in a chain. The research on recurrent neural networks began in the 1980s and 1990s, and developed into one of the deep learning algorithms in the early 21st century [15]. However, RNN also has a fatal flaw that is difficult to imitate long-term dependencies. The Long-Short Term network commonly referred to as LSTM, and it's a particular type of RNN that can learn long-term dependency information.

### 2.4.2 LSTM: Long short-term memory

LSTM was proposed in [41] and Fig. 2.4 illustrates the details of the algorithm. It avoids long-term dependencies by deliberate design. It remembers that long-term information is the default behavior of LSTM in practice, not the ability to get by lots of training. Compared to a simple layer of neural networks, LSTM has four layers that interact in a particular way. The LSTM does have the ability to delete or add information to the state of the cell, a capability that is conferred by a structure called a Gate. Gate is a way to pass information selectively. It consists of a Sigmoid neural network layer and a point multiplication operation. Three gates

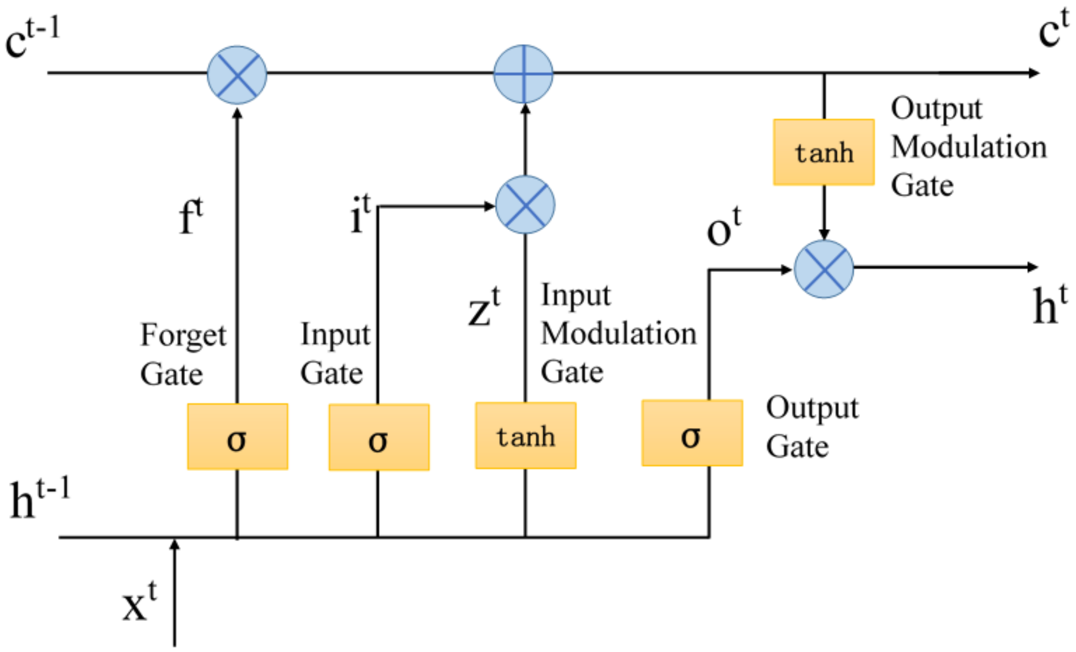


Fig. 2.4 LSTM Process

are placed in a cell, called input gate, forgetting gate and output gate. A message enters the LSTM network and can be judged according to the algorithm. Through the calculation of the input data(information), the weight of the input data to the predicted data is obtained. Only data that complies with the algorithm's certification will be left, and the data that does not match will be forgotten through the Forgotten Gate.

In Figure 2.4, this is the transformation of the cell state. The transformation of the C value has undergone a total of two operations. These two operations determine the forgetting and updating of $C^{t-1}$. First, $C^{t-1}$ undergoes the point operation of multiplication. This step determines whether the value of $C^{t-1}$ is forgotten or not. The number passes through a sigmoid layer multiplied by $C^{t-1}$ comes and then the value (0,1) of the data that penetrates downward. When the input value is greater than 3 or less than -3, the value of sigmoid is close to 1 and 0. It means remembering C or forgetting $C^{t-1}$. If a number between (0,1) is multiplied by $C^{t-1}$ , it means how many $C^{t-1}$ values need to be remembered. Next, the value of $C^{t-1}$ encounters an addition, which is a change to the value of $C^{t-1}$>0. Thus, the operation of C value is finished in the forgetting mechanism. This neural network can remember and forget and the addition of new information means new information needs to be remembered.

In the operation process of the forgotten gate, there are two inputs:

$$f^t = \sigma * (W_f * [h^{t-1}, x^t] + b_f) \tag{2.4}$$

The $h^{t-1}$ is the information passed from the last cell and $x^t$ is the input of the current cell. The $\sigma$ is the sigmoid function, $W_f$ is the weight and $b_f$ is the bias of the gate. The formula makes a linear transformation into nonlinear transformation. At this time, the value of $f^t$ is (1,0). This value determines whether the value of $C^{t-1}$ is remembered or forgotten, or how much is remembered. This is the function of this forgetting gate.

In the input gate, the function $i^t$ is similar to the $f^t$ :

$$i^t = \sigma * (W_i * [h^{t-1}, x^t] + b_i) \tag{2.5}$$

and it means the input gate also has the forgotten function. The forgetting ability of $i^t$ corresponds to $Z^t$. $Z^t$ is the new value that LSTM needs to remember in operation process:

$$Z^t = tanh * (W_Z * [h^{t-1}, x^t] + b_Z) \tag{2.6}$$

The value range of $Z^t$ is (-1,1) due to the tanh function. The new memory can be either positive or negative, which can be understood as a positive memory and negative memory. $W_Z$ is the weight and $b_Z$ is the bias.

$$o^t = \sigma * (W_o * [h^{t-1}, x^t] + b_o) \tag{2.7}$$

$$h^t = o^t * tanh(C^t) \tag{2.8}$$

$h^t$ is the output of the current cell and the operation process:

$$o^t = \sigma * (W_o * [h^{t-1}, x^t] + b_o) \tag{2.9}$$

$$h^t = o^t * tanh(C^t) \tag{2.10}$$

$h^t$ is equal to $o^t$ multiplied by tanh($C^t$) and it means the output gate $o^t$ controls the value of output $h^t$. $W_o$ is the weight and $b_o$ is the bias of output gate.

At last, the $C^{t-1}$ is the cell state of the last cell and the $Z^t$ is the new information that needs to be remembered in the current cell. The C value of the current cell state is determined as:

$$C^t = f^t * C^{t-1} + i^t * Z^t \tag{2.11}$$

.

## 2.5 Reviews of DTW algorithm

The Dynamic Time Warping (DTW) algorithm was proposed in 1970 to be an algorithm that was applied to compare the similarity of two-time series [42]. The DTW algorithm has been used in signature classification comparisons [43], and in the text-dependent speaker verification system also used this algorithm [44]. From the two words Time Warping in the algorithm name, the DTW algorithm is not sensitive to the extension and compression of the sequence. Therefore, DTW algorithms and various algorithms based on DTW improvements have many applications in the field of speech recognition.

For two sequences of different lengths, the DTW distance can calculate the Euclidean distance between them. The two time series for calculating the similarity are X and Y, and the lengths are |X| and |Y| respectively. The form of the round path is $W = w_1, w_2, ..., w_K$, where $Max(|X|,|Y|) <= K <= |X|+|Y|$. The form of $w_k$ is (i, j), where i represents the i's coordinate in X and j represents the j's coordinate in Y in Fig.2.5.

The final rounding path is the one with the shortest distance:

$$Dist(W) = \sum_{k=1}^{k=K} \qquad Dist(w_{ki}, w_{kj}) \tag{2.12}$$

Among them, $Dist(w_{ki}, w_{kj})$ is any classical distance calculation method, such as Euclidean distance. $w_{ki}$ refers to the i-th data point, and $w_{kj}$ refers to the j-th data point.

Suppose there are two-time series X and Y, their lengths are i and j: (in actual speech matching, one sequence is a reference template, the other is a test template, and the value of each point in the sequence is the eigenvalue of each frame in the speech sequence. For example, the speech sequence X has i frames, and the eigenvalue (a number or a vector) of the i frame is $X_i$. As for the features, the discussion of DTW is not affected here. The function needs to match the similarity of the two speech sequences in order to recognize which word our test speech is.)

$$X = x_1, x_2 ..., x_i Y = y_1, y_2 ..., y_j \tag{2.13}$$

If i = y, then there is no need to toss about, just calculate the distance between two sequences. It needs to be aligned when n is not equal to m. The simplest alignment is linear scaling. The short sequence is linearly enlarged to the same length as the long sequence, or the long linear sequence is shortened to the same length as the short sequence for comparison. However, such calculation does not take into account the changes in the duration of each segment in different situations, so the recognition effect is not optimal. Therefore, the method of dynamic programming is more used.

To align the two sequences, it needs to construct an n x m matrix grid. The matrix element (i,j) represents the distance D$(X_i, Y_j)$ between $X_i$ and $Y_j$ (that is, the similarity between each point of sequence Q and each point of C. the smaller the distance, the higher the similarity. In this paper, regardless of the order, the Euclidean distance is generally used, D(i, j) = $(X_i - Y_j)^2$ (also can be understood as distortion). Each matrix element (i, j) represents the alignment of points $X_i$ and $Y_j$. The dynamic programming algorithm can be used to find out the path through several grid points in this grid. The grid points through which the path passes are the aligned points calculated by two sequences.

The idea of dynamic programming is used in DTW, where D(i,j) represents the rounding path distance between two time series of length i and j:

$$D(i,j) = Dist(i,j) + min[D(i-1,j), D(i,j-1), D(i-1,j-1)] \tag{2.14}$$

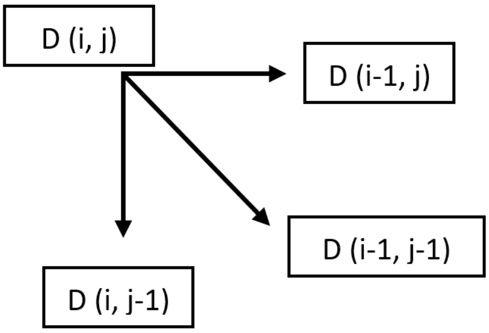


Fig. 2.5 Dynamic Programming

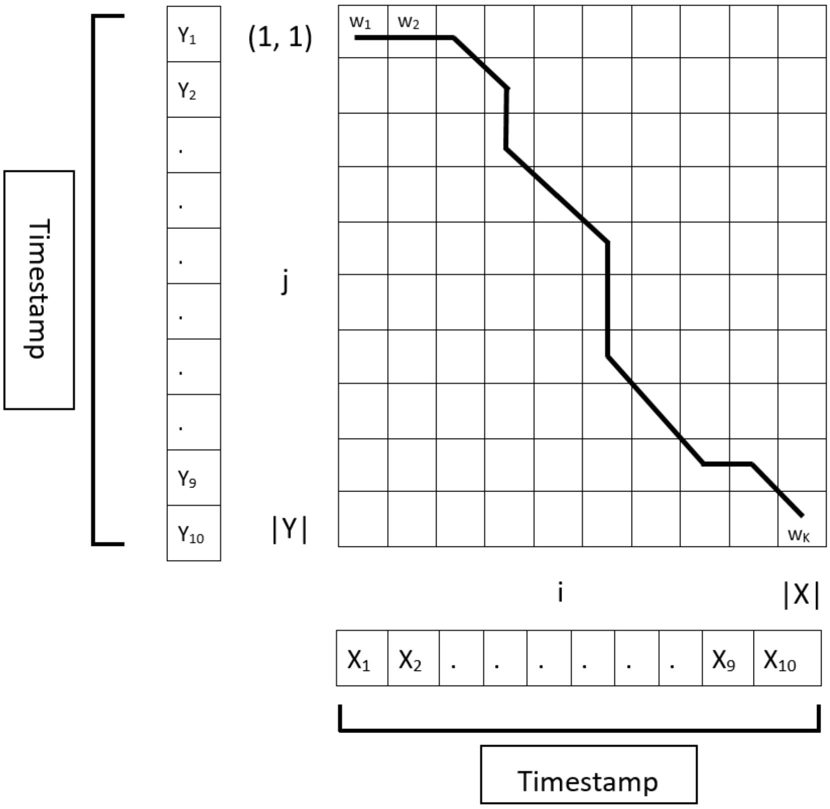


Fig. 2.6 DTW Process

The dynamic programming algorithm is used to find the path. Suppose the minimum cumulative distance D(i, j) to the position (i, j) is required. The minimum cumulative distance can be found out in D(i-1,j), D(i,j-1) and D(i-1,j-1). Figure 2.5 shows the optimal path is found out from three adjacent locations.

The goal of DTW is to extend and shorten the two-time series to get the shortest warping between the two-time series, that is, the most similar warping. This shortest distance D$(x_i, y_j)$ is the final distance measurement of the two-time series. Starting from the (0, 0) point, the two sequences X and Y are matched. For each point, the distances calculated by all previous points are accumulated. After reaching the endpoint (i, j), the cumulative distance is the total distance as mentioned above. In other words, it's the similarity between sequence X and Y as shown in Fig.2.6.

# Chapter 3

# Twitter traffic analysis and Clustering Algorithms Evaluation

## 3.1 Introduction

Because of the rapid development of networks, the total amount of network data is expected to increase by two times per year in the future. In the next decade, it will reach a 1000 times growth in total, especially in the area of wireless communication networks [45]. The BSs deployment is essential for ensuring the quality of service of wireless communication. Meanwhile, the SBSs is proposed to be a good optimization method for network data bursts. SBSs is deployed in a family unit to provide better QoS and coverage. However, it's facing some commercial and technical challenges [46].

The most traditional BSs deployment model is the hexagonal grid model [47]. This model set the base station coverage as a regular hexagon, a base station has the same distance between each other, and the coverage of a base station will not affect others. Thus, this is a simple and effective model. It meets the coverage satisfaction of cellular network without massive modulation. However, the capacity demand, in reality, is floating over time, and the assumption of this model is too ideal. Thus, it is difficult and unrealistic to copy and deploy a hexagonal grid model in the real world. In recent decades, the PPP (Poisson Point Process) model replaced the traditional hexagonal grid model for BSs deployment [7, 48]. Traditional modelling research methods for a single cellular network are mainly concentrated in the hexagonal grid model. But this model is very ideal, the actual base station distribution will not be a standard hexagonal grid model. This model can only get the upper bound of the coverage probability. The characteristic of the randomness of the PPP model makes it a much suitable model for realistic BSs deployment. On the other hand, it is because of this randomness; the PPP model sometimes cannot correspond to the actual environment. The base station deployment is affected by the actual geographical location (like rivers, mountains and buildings) and cannot fully fit the PPP model. The Poisson distribution has been questioned and challenged in recent years [8, 9]. Except for its randomness, the human behaviours and the realistic constraint (like geographical environment) also influenced the real world BS deployment [50]. Author Du also proposed the semi-Markov model is more efficient in characterizing user mobility patterns than standard Markov[50]. This result is based on the correlation of community and geography information in Mobile Social Networks (MSNets), and user's sojourn time distribution over communities.

Recently, data has increased rapidly in the past 20 years. People, devices and their interactions produced a massive quantity of information in various fields. People start to sort out, excavate and develop a massive quantity of information. It gives us a new opportunity to discover the hidden value. Author Ahn used clustering distribution, clustering coefficient, degree correlation and average path length to analyze social network data [51]. In general, the concept of 'data analysis' includes storing and analysing a large scale of data. Exploratory Data Analysis (EDA) is a method to explore the structure and regularity of data under assumptions [52]. The main tasks of exploratory data analysis (EDA) are: cleaning the data, describing the data (describing statistics, charts), viewing the distribution of data, comparing the relationship between data, cultivating the intuition of data, summarizing the data, etc. After the emergence of EDA, the process of data analysis is divided into two steps, the exploration phase and the verification phase. The exploration phase focuses on discovering patterns or models contained in the data. The validation phase focuses on evaluating the patterns or models found. Many machine learning algorithms (two steps of training and testing) follow this idea [53].

In this chapter, data analysis refers to the process of detailed study and summarization of the data by using appropriate statistical analysis methods to analyze the collected data. Data analysis has the main purpose to discover useful information hidden in data and the process contains inspecting, cleaning, transforming, and modelling. Useful information is normally used for supporting decision-making and help to build new theory. For different disciplines, industries, and research areas, there are lots of data analysis methods. In this chapter, traffic data with geographical location are the main research dataset. Thus, the clustering algorithm is the main methods used for data analysis.

## 3.2 Data source and analysis methods

### 3.2.1 Twitter traffic

At first, the traffic transmission speed is the first characteristic of data sources. Low-speed traffic is including sending text and pictures to social networks. Thus, after searching for lots of social networks. The low-speed traffic data sources have been selected because of the following reasons. Firstly, in comparison with high-speed traffic, low-speed data sources are easier to find. Secondly, according to the easy collection, only upload traffic will be analysed. The question 'Is it meaningful to study the low-speed upload social network data?' is derived. This problem is already solved and will be described in detail later. The last point is the capacity to upload Twitter traffic is relatively small. Thus, the upload time will be short and the position of traffic occurs can be considered as fixed. To analyse the relationship between low-speed traffic with BSs will be easier under this assumption. Fig. 3.1 shows the Twitter traffic in the Great London areas within two weeks period. The traffic map displays an uneven distribution, a large amount of data is concentrated in the central city and the number of traffic decreases when the map turns to the suburbs.

### 3.2.2 Overview of two clustering algorithms

Twitter becomes the primary data source which will provide the data for early research. There is an API that can be used to collect tweets on Twitter. Data collected from Twitter have many characteristics that are very useful for this research. From the data sources, each tweet contains upload date, location, content, user id, the link of this tweet, and so on. Location is one of the research parameter focus in this chapter. By collecting and analysing

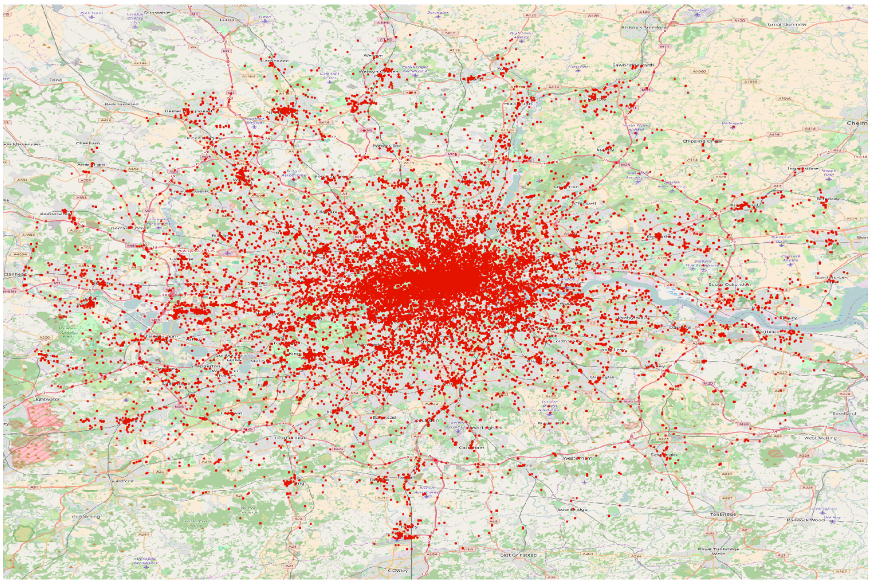


Fig. 3.1 Twitter data in London: The figure shows the Twitter data collected in London. Each point represents one tweet.

the location of data from Twitter, the high traffic load area could be discovered. There are two methods to find a high traffic load area. The first method is calculating the Euclid distance between surrounding points and randomly selected central points. A radius of a randomly chosen central location and the minimum number of surrounding points of this central point are settled before the calculation. If the number of surrounding points reached a settled count within a specific range of area, this central point would be defined as a high traffic load spot. This method gives an outlook of the sample data straightforwardly, and high traffic load points will be clearly shown on the map. However, the distance from each point to the nearest centre point needs to be recalculated after each iteration, which could reduce the efficiency of data analysis.

The second method is using the DBSCAN algorithm to discover a high traffic spot. As stated in the literature review, the DBSCAN algorithm also has two settled parameters, which are MinPts and $\varepsilon$(direct reach distance of points to the cluster centre). The two settled parameters are a compensated distance between surrounding points and the central point, and the minimum number of surrounding points. Different from the Euclid distance method, which only undertakes fundamental analysis, the second method digs into a more in-depth level. However, even if the DBSCAN algorithm does a more in-depth analysis, it needs to be noticed that the final area of the cluster might be relatively broader than the coverage area of BSs in multiple levels of analysis.

To summarise, the Euclid distance method is more convenient when finding out high traffic areas. The parameters (radius of centre points and the minimum number) are controllable. One shortcoming of this method will be the lack of analysis of the relationship between groups. The DBSCAN algorithm will carry out a more in-depth analysis. Similar groups will be gathered together to show different high traffic Areas from the results of the first method. Nevertheless, since the detailed analysis is not controllable, the shape and size of clusters also cannot be controlled.

## 3.3 High traffic areas allocation results

### 3.3.1 K-means based Euclidian algorithm

For the first method, Euclid distance method is an easy way to achieve for two-dimension data sets. The number of high traffic areas will be controlled by two parameters, the radius of the centre point and the minimum number of surrounding points. For SBSs, the coverage radius is normally between 100-200 meters [66]. SBSs can provide effective QoS within this distance. Moreover, the density of data with geographic location is relatively low. Thus, at the this research stage, the radius of centre points has been set as the maximum coverage radius of SBSs (200 meters).

### 3.3.2 K-means based Euclidian algorithm analysis results

The high traffic area depends on the minimum surrounding points(minimum points in clusters) and the radius of the centre points in the Euclid distance method. As the radius of the centre point has been decided to be 200 meters based on literature review. Fig. 3.2, 3.3, and 3.4 show the results of the number of high traffic areas under different parameters. From those diagrams, the number of high traffic areas increased along with the minimum surrounding points decreasing. Moreover, more than half high traffic areas are gathered at the centre of London, and the rest of them are separately scattered in the outside layer of London. The locations of all high traffic areas are gathering at the centre of London when the minimum surrounding points is relatively large. This phenomenon may cause a different function in this area. The different types of functions of a region such as tourist attractions, business centre, and entertainment areas might affect the density of the real data set. St Albans is the second-highest density place in the research target area and it is a tourist attraction. This problem will be noted in the subsection related to geocoding and geographic location reverse analysis. Through a preliminary analysis, this method clearly shows a high traffic area of the sample data in a straightforward way. High traffic area allocation results are relatively controllable by two parameters: the radius of the centre point and the minimum surrounding points. However, it has a significant drawback which is the density of the data sample affects the clustering results. It needs lots of time to test the approximate range

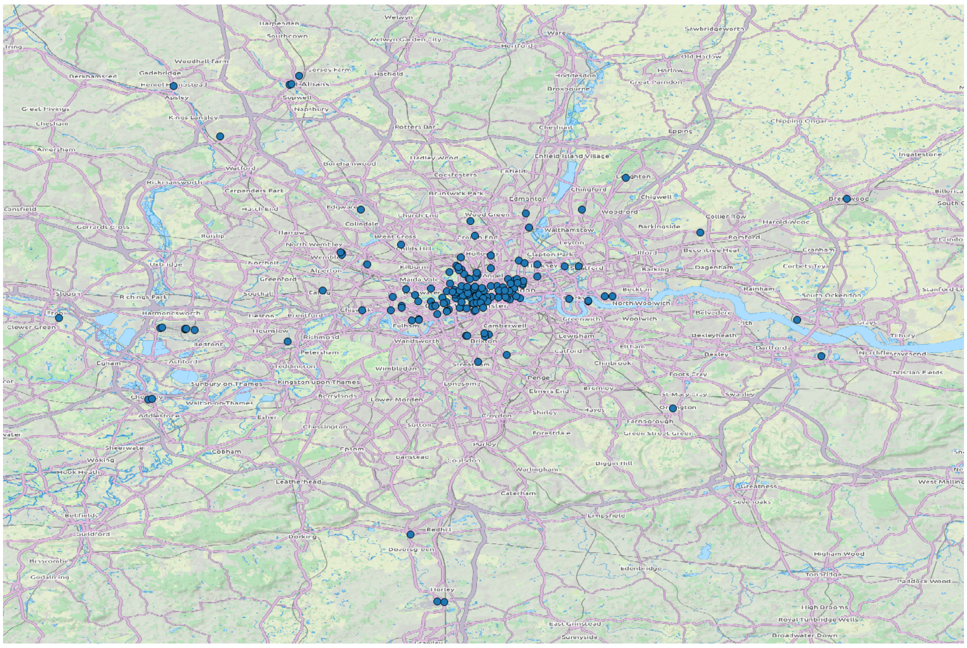


Fig. 3.2 The result of Euclid distance algorithm (Radius:0.2km MinPts:100 High Traffic Spots:184)

for two adjustable parameters. Even though, some high traffic areas will be ignored because of the low a data sample density on that certain area.

### 3.3.3 DBSCAN algorithm analysis results

As the Euclid distance method, the two adjustable parameters may affect the clustering results of the DBSCAN algorithm as well. Those parameters were set as same as the first method used for comparison. Fig. 3.5, 3.6, and 3.7 show the relationship between the number of high traffic areas and MinPts. The number of high traffic areas has a relatively small change against MinPts. The number of high traffic areas also increased along with the MinPts decreasing as the first method. However, the increase in the number of high traffic areas is not significant in DBSCAN method. The diagrams show clearly that the Euclid distance method has a significant change against minimum surrounding points and

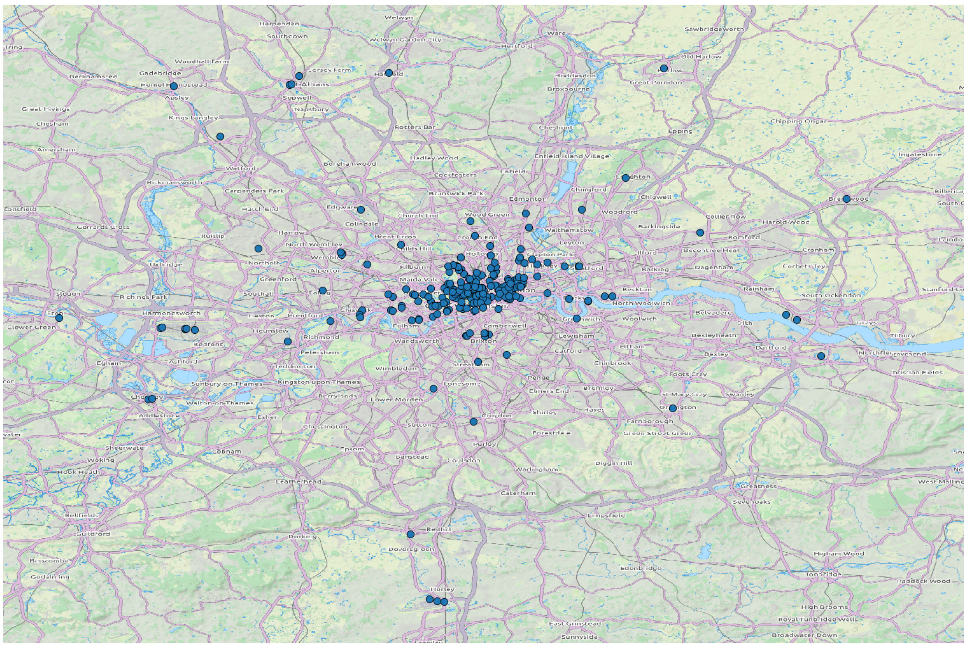


Fig. 3.3 The result of Euclid distance algorithm (Radius:0.2km MinPts:75 High Traffic Spots:253)

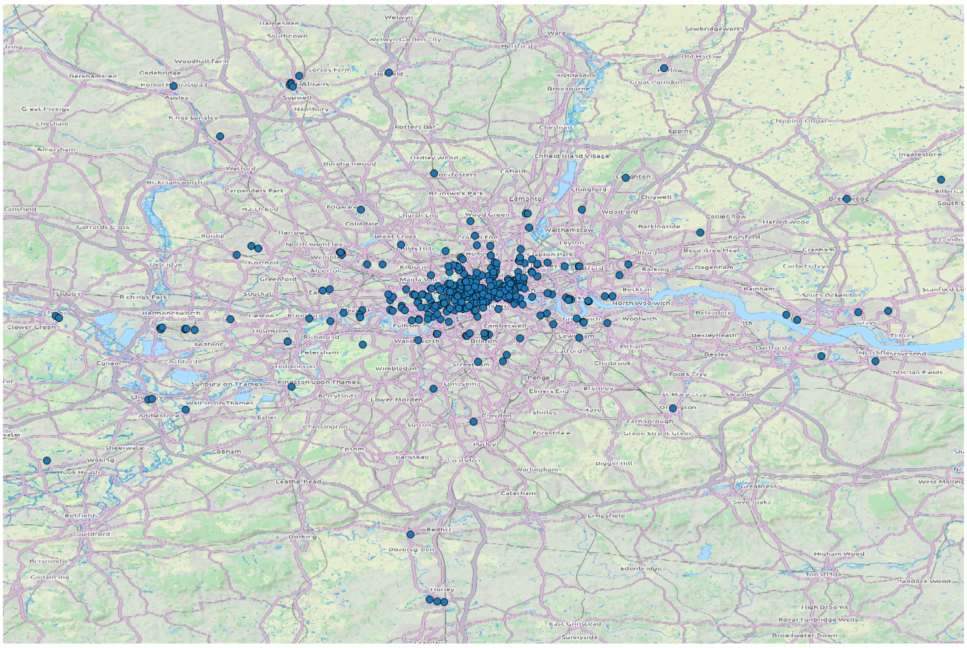


Fig. 3.4 The result of Euclid distance algorithm (Radius:0.2km MinPts:50 High Traffic Spots:376)

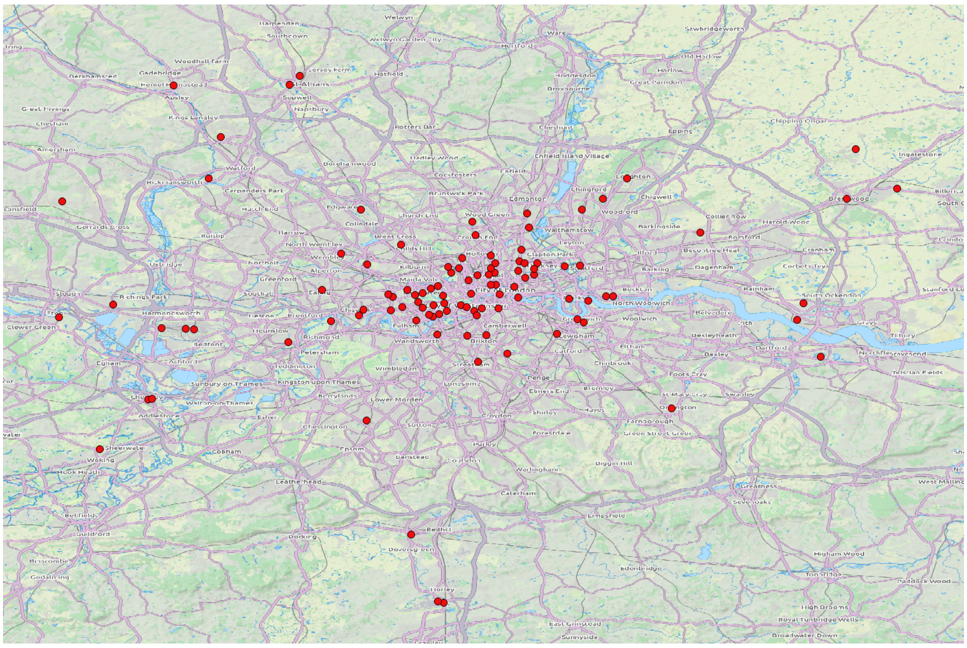


Fig. 3.5 The result of DBSCAN algorithm (Radius:0.2km MinPts:100 High Traffic Spots:106)

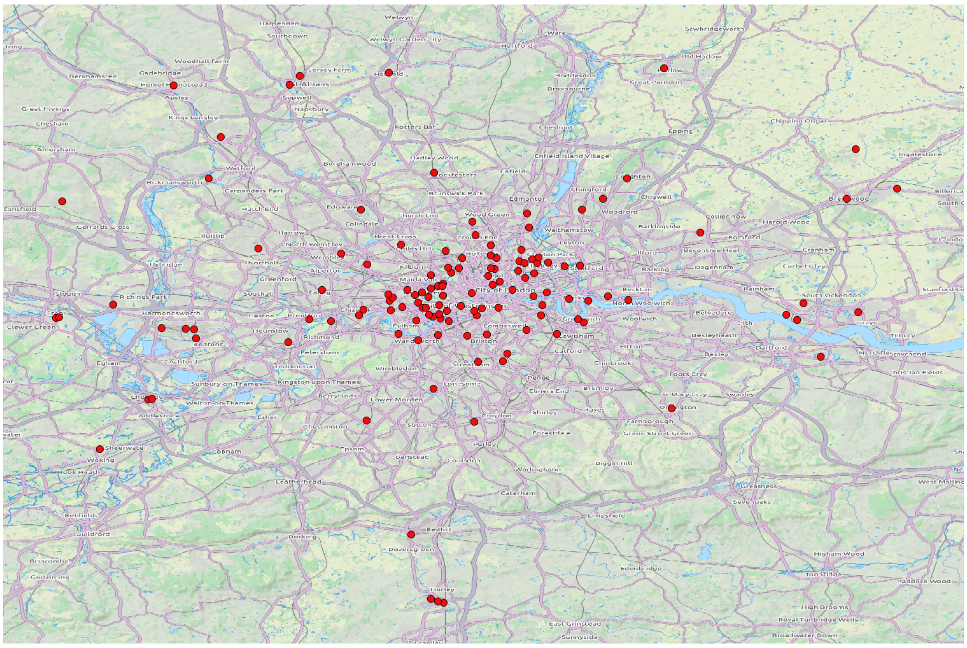


Fig. 3.6 The result of DBSCAN algorithm (Radius:0.2km MinPts:75 High Traffic Spots:132)

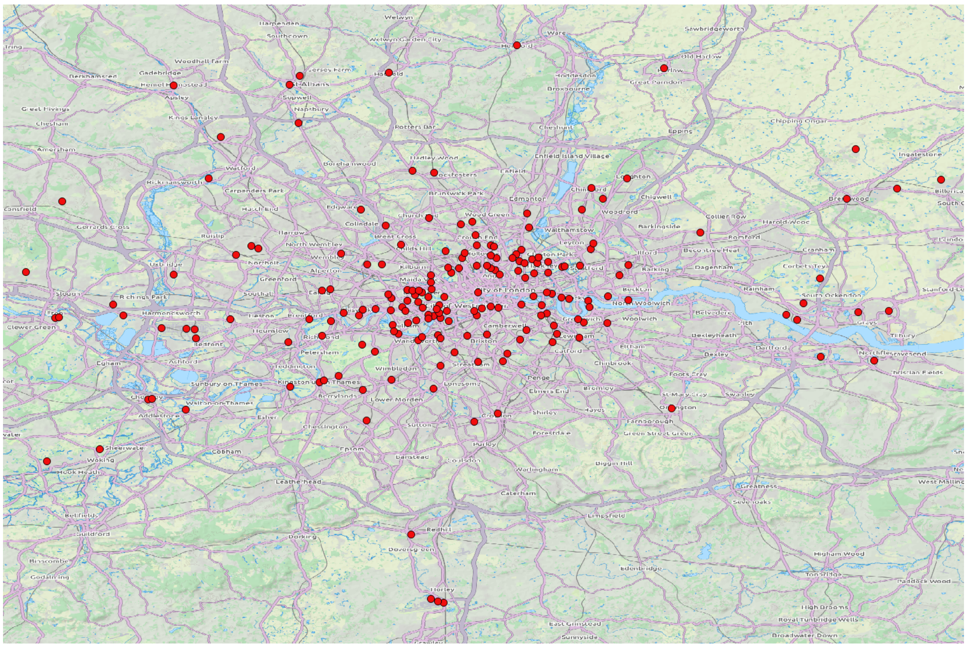


Fig. 3.7 The result of DBSCAN algorithm (Radius:0.2km MinPts:50 High Traffic Spots:194)

DBSCAN algorithm less so. The different algorithm process causes these results. Seeking the largest set of density-connected objects is the purpose of the DBSCAN algorithm. Thus, all points directed to the centre point of the clusters are allocated in the same cluster. This is the reason that the results of the DBSCAN algorithm show less number of high traffic areas under the same conditions. Furthermore, the distribution of high traffic areas also shows a different pattern. Results of the Euclid distance method show high traffic areas are more centralized at the centre of London. The results of the DBSCAN algorithm show another distribution in which the high traffic areas were more marked in the rural area and less at the centre of London. Because the traffic density in the centre of London is relatively higher than in the rural area. Thus, most of the traffic is clustering into a very big cluster. In the end, only a large high traffic cluster clustered in the centre of London on the traffic map.

## 3.4 Hot-Index of high traffic areas

### 3.4.1 Overview of hot-index of clusters

The location of high traffic areas is the most intuitive and the easiest information to analyse. But observing the distribution of high traffic areas is not enough. Out of the location of high traffic areas, more informative data can be mined. This hotness can act a better index to show the high traffic areas. The hot-index is a quantity value that represents the sum of all points in the clusters in both two methods mentioned. The value of the hot-index shows the degree of hotness of the high traffic area. It also can be explained as the density of traffic on a specified area.

Fig. 3.8, 3.9, 3.10 and 3.11 are four examples of high traffic areas with hot-index and the size of the circle represents the degree of hotness. In fig. 3.12, the results of both Euclidian distance method and DBSCAN algorithm are plotted on the same traffic map.

### 3.4.2 Overall renderings in London

After plotting the high traffic areas with hot-index by two methods individually, two results perform different distribution as expected. The number of high traffic areas has to be controlled to be same for a better observation. Fig. 3.8 and 3.9 show the whole high traffic areas with hot-index results of two methods. For the Euclid distance method, most of the high traffic points are gathering together at the centre of the urban area rather than the remote area. The result of the DBSCAN algorithm found out more high traffic areas in the rural area. The DBSCAN algorithm may perform better in allocating high traffic areas.

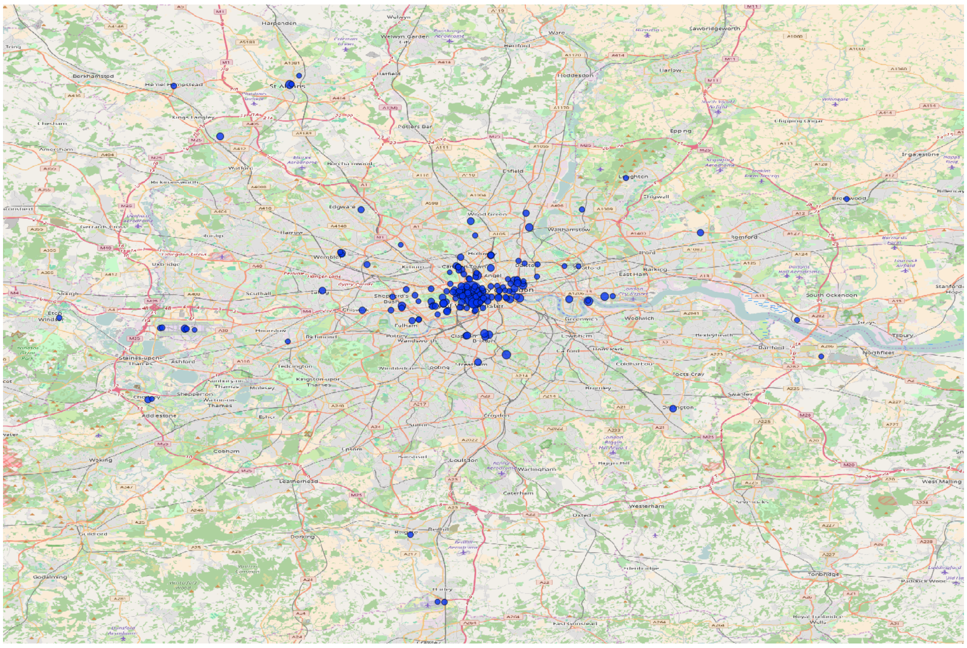


Fig. 3.8 The overall view of clustering results with hot-index of the result of Euclid distance algorithm: The size of the circle represents the hotness of the cluster (bigger circle means more tweets in a cluster).

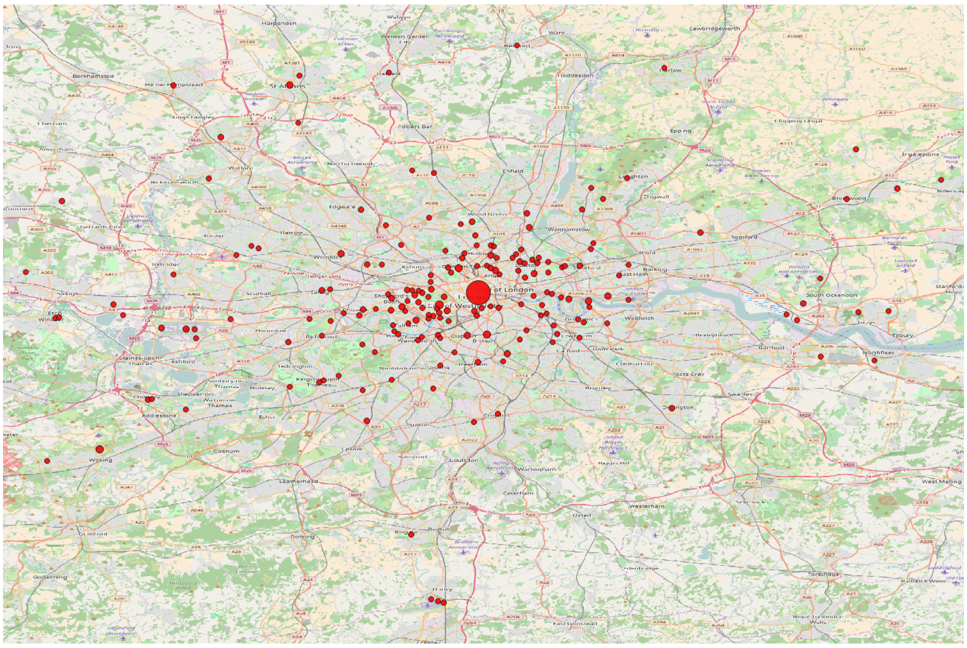


Fig. 3.9 The overall view of clustering results with hot-index of the result of DBSCAN algorithm: The size of the circle represents the hotness of the cluster (bigger circle means more tweets in a cluster). The results of DBSCAN can better show the uneven distribution of data in real wold.

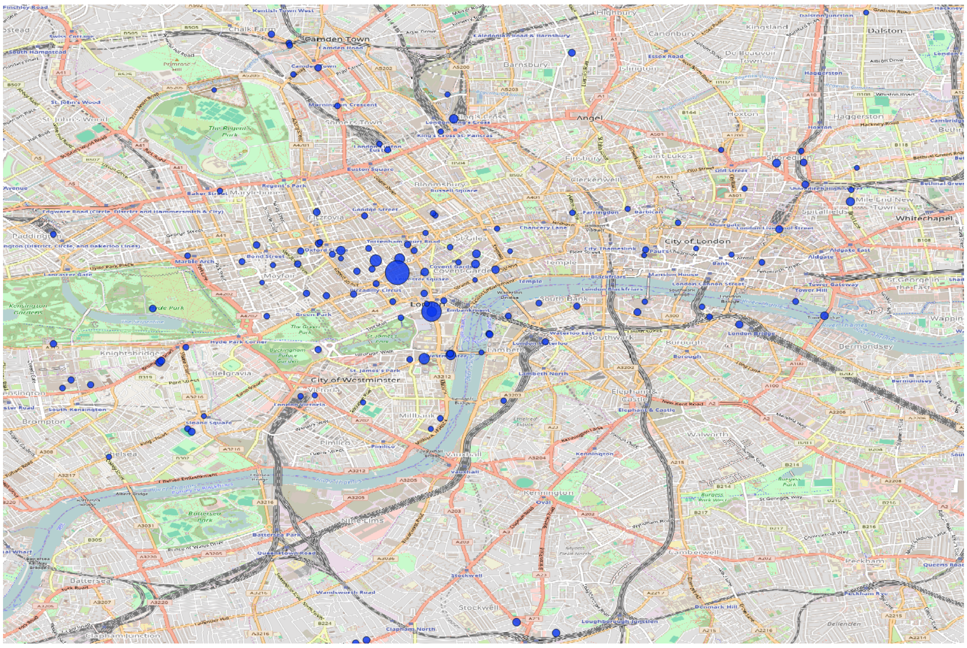


Fig. 3.10 The partial view of clustering results with hot-index of the result of Euclid distance algorithm: Figure 3.10 shows the partial view of traffic map in the centre of London.

### 3.4.3 Urban district renderings in London

Fig. 3.10 and 3.11 show a partial view of the traffic map in the centre of London. Lots of high traffic areas are gathered together at the centre of London by the Euclid algorithm. This represents the density of traffic in the centre of London is relatively high than in remote areas. The results of the DBSCAN algorithm shows the same phenomenon. According to the algorithm process of DBSCAN algorithm, all points directed to the centre point of the clusters are allocated in the same cluster. Thus, only one big circle displays in the centre on the map in fig 3.11. From observation, both of the two methods proved the highest traffic density is at the centre of London. However, the result of the DBSCAN algorithm more clearly reflect the traffic density in the city centre is much higher than in the suburbs.

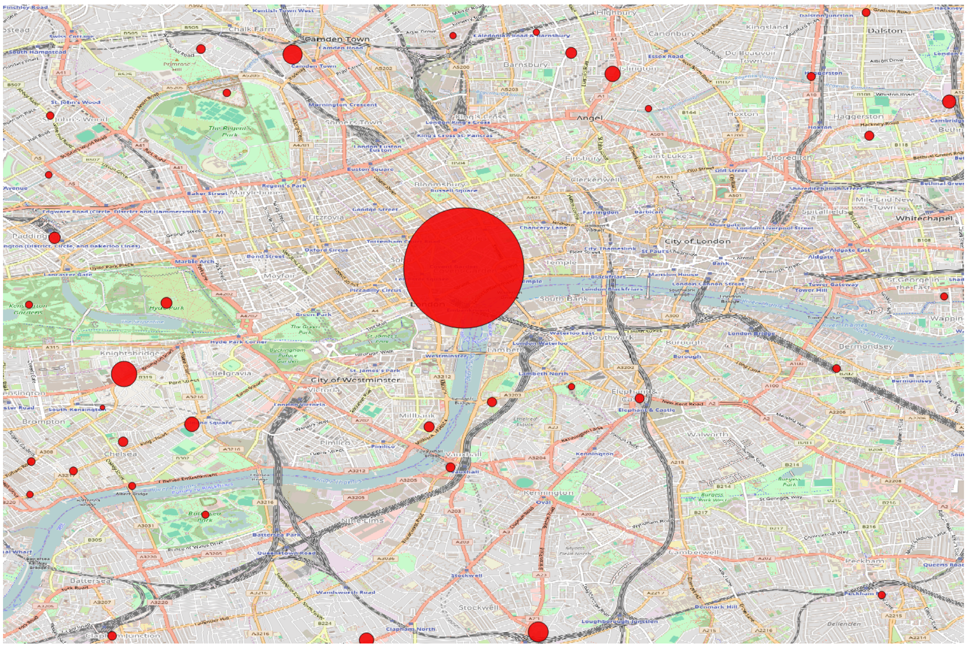


Fig. 3.11 The partial view of clustering results with hot-index of the result of DBSCAN algorithm: Figure 3.11 shows the partial view of traffic map in the centre of London. The result of DBSCAN reflect the phenomenon of data gathering in the city centre.

### 3.4.4 Renderings of integration of two methods

Fig. 3.12 shows the overlapping of the diagrams of the two algorithms together. The figure shows the difference between the two algorithms more clearly. The DBSCAN algorithm better aggregates the busiest areas into one large cluster, which more clearly reflect the traffic density in the city centre is much higher than in the suburbs.

### 3.4.5 Discussion

These two methods can find out the high traffic areas based on the geological location of Twitter traffic. Both two methods analysed Twitter traffic and found out the centre points of high traffic areas. Results show the distribution of high traffic areas and both two methods can distinguish the density of sample data set by hot-index. The two methods delivered

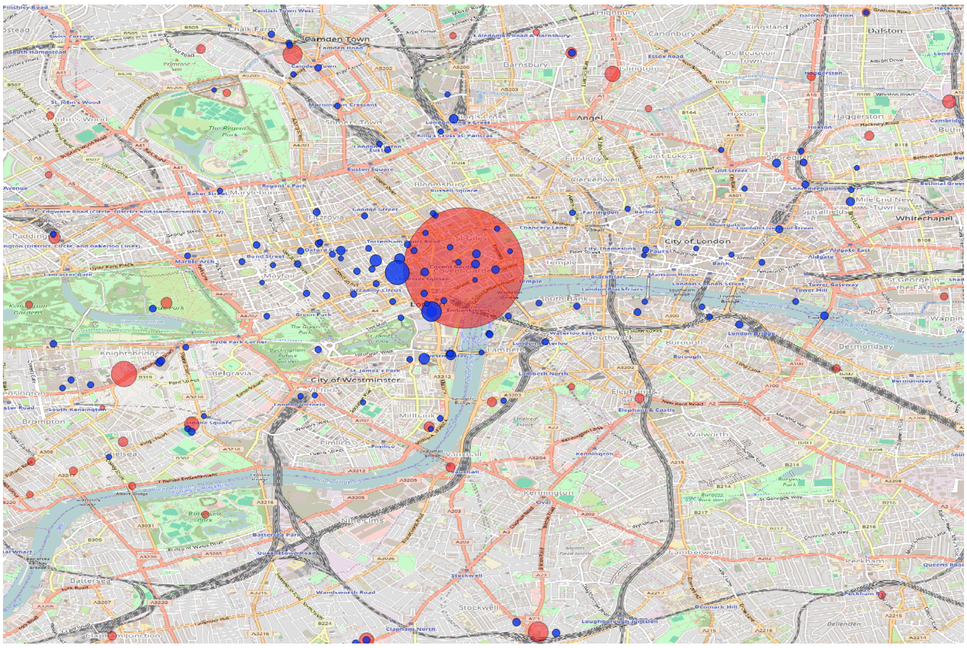


Fig. 3.12 The partial view of clustering results with hot-index of Euclid and DBSCAN algorithms: Figure 3.12 is the overlap of figure 3.10 and figure 3.11

two different results on the traffic map. The Euclidian distance method discovers the high traffic areas on the low traffic density place(suburbs) and DBSCAN algorithm reflects the traffic density in the city centre is much higher than in the suburbs. However, both methods have shortcomings. The Euclidian distance method is a straightforward algorithm and it is relatively easy to understand. But it can't discover the difference in traffic density in a certain area. The shortage of the DBSCAN algorithm is the size and shape of the cluster are uncontrollable.

## 3.5 Geo-coding and geographic location reverse analysis results

### 3.5.1 Geo-coding and geographic location reverse analysis

Based on the last section, two mentioned methods can discover high traffic spots. However, these high traffic spots might shift from real-life high traffic load spots. There are two ways to minimise the gap between theoretical results and real-world high traffic regions. Firstly, calculating the distance between two points on the surface of Earth needs to use "Haversine" formula [54]. The second way is to analyse the surrounding landmarks and region properties of the theoretical high traffic load spots.

Geographic location reverse analysis is used to find out the surrounding landmarks and region properties of their ideal high traffic load spots. Google maps geocoding API provide the service to get the building of the high traffic spots. The way to convert geographic coordinates into a complete, readable location address is called reverse geocoding. It is not only used for converting the longitude and latitude to a human-readable address but also the Google Maps API is enabled to search the place around specific geographic coordinates. The programme needs three main necessary parameters: access API key, geographic location of a certain point, and the radius of the range.

### 3.5.2 Results of geographic location reverse analysis

Furthermore, not only the range is controllable. It also can set the building type and search keywords. Table 3.1 shows the surrounding information on six high traffic regions. The

| No. | Type | High traffic regions |
|---|---|---|
| 1 | -lodging<br>-restaurant<br>-food<br>-point of interest<br>-establishment | Premier Inn London Leicester Square |
| 2 | -lodging<br>-restaurant<br>-food<br>-point of interest<br>-establishment | W London, Leicester Square |
| 3 | -bar<br>-lodging<br>-restaurant<br>-food<br>-point of interest<br>-establishment | Every Hotel Piccadilly |
| 4 | -lodging<br>-point of interest<br>-establishment | Radisson Blu Edwardian, Hampshire |
| 5 | -lodging<br>-point of interest<br>-establishment | The Z Hotel Soho |
| 6 | -lodging<br>-restaurant<br>-food<br>-point of interest<br>-establishment | Radisson Blu Edwardian, Mercer Street |

Table 3.1 Geographic location reverse of six high traffic regions: This table lists six high traffic regions. Those high traffic regions have the same regional attributes like lodging, point of interest and establishment.

information includes geographic location, types, and vicinity. Further information like an icon of architecture and street; the view of the picture; and the link to the address on the Google map also included in the table.

## 3.6 The shortest distance test results

The shortest distance test between traffic location with the BS's location is another way to check the coverage of the BSs. The data of the geographic location of BSs in London have been collected from Opensignal. The database shows there are more than 2000 BSs in London of one operation in the same area range as the data collected from Twitter [49]. Twitter traffic has divided into two parts. The first-week traffic is used to clustering the high traffic spots (the centre of high traffic areas). Then, set another week's traffic as the object (set as the location of users). DBSCAN clustering more than 1600 high traffic spots and those points are set as the fake geographic location of BSs. In the meantime, there was 688898 user's location tested for the shortest distance.

Figure 3.11 shows the results of the shortest distance between the centre of high traffic spots and tweets. It represents 8 distance intervals and the number of tweets in each interval. It shows 43662 tweets are very close (less than 100 meters) to a high traffic spot. Followed by the next interval, 14934 tweets are a little bit away from the nearest high traffic spots. Except for the first two intervals, all the rest of the interval contains less than 5000 tweets. If there is BS deployment at those high traffic spots, it will cover 85% of tweets even if it is SBSs (the coverage of an SBSs is 100-200meter in radius). Thus, the results show the social network's data are valuable for SBSs deployment.

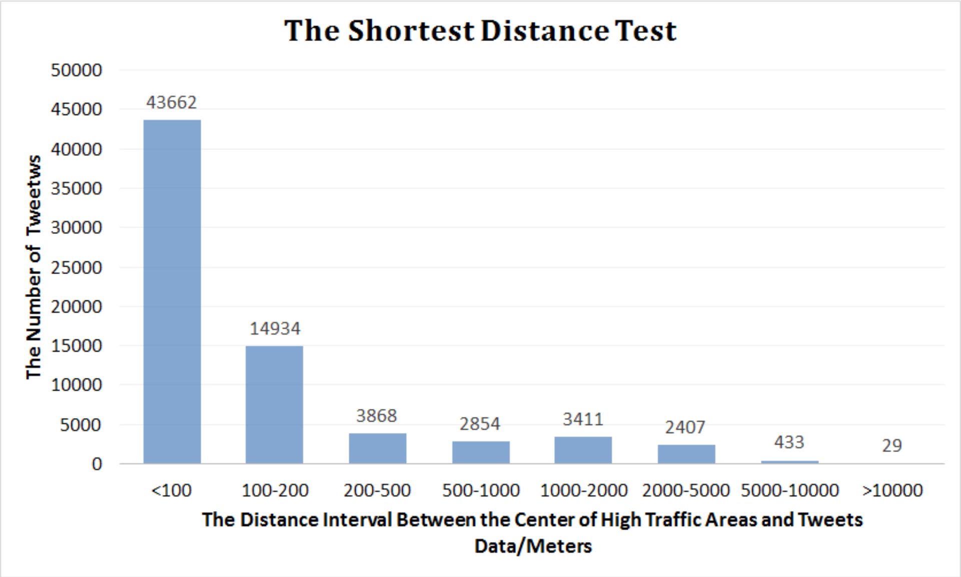


Fig. 3.13 The shortest distance test: This figure shows the results of the shortest distance between the centre of high traffic spots and tweets. The high traffic spots found by DBSCAN are very effective in reality, more than 85% of tweets that are within 200 meters from the high traffic centre.

## 3.7 Conclusion

In this chapter, the research aims to find out a clustering algorithm of base station deployment of London. The hexagonal grid model can achieve the largest coverage area when using the smallest base station. The hexagonal grid model will be affected by similar rivers, mountains, and tall buildings in the real environment. The randomness of the PPP model is more suitable for the deployment of base stations in reality. But these models have no function to find high traffic areas in reality. As the popularity of smartphones continues to increase, more and more people choose to use social networks for communication. The social network data analysis can make a significant contribution to the deployment and optimization of base stations.

Twitter traffic is used as the dataset in this chapter. In the wireless communication network, the traffic demand can be predicted by social network data. Moreover, Twitter data has a strong positive relationship with operator traffic demand. At the same time, the location information provided by Twitter is needed to find high-traffic areas. Euclid and DBSCAN clustering algorithms are used to locate the high traffic areas. These two methods can find out the high traffic areas based on the geological location of Twitter traffic.

It is cable to plot the location of the high-traffic area on the map through the Google API. Moreover, it also provides the information likes business district, area attributes, and vicinity. This information can be used to find the similarities and differences between high traffic areas. The shortest distance between the user's location with the clustered high traffic spots is calculated. At least 85% of users are only 200 meters away from the base station. The results of the shortest distance test indicate the social network's data are valuable for allocating the high traffic spots.

# Chapter 4

# Data Driven Optimization of Small Cell Deployment

## 4.1 Introduction

With the proliferation of mobile devices and the drastic growth of multimedia applications such as social networking, video sharing, and telepresence, mobile communication networks have become an integral part of people's daily lives. It contributes to the rapid growth of mobile traffic, which is pushing the existing cellular networks to their limits. In the fifth-generation (5G) and beyond mobile networks, the deployment of SBSs is seen as one of the promising solutions to enhance the capacity of the radio access networks, and thus meet the explosive growth of mobile traffic demand.

However, the spatial distribution of mobile traffic in realistic scenarios is extremely uneven, especially for urban areas where various regions (e.g., shopping mall, central business district, rail station, university campus, etc.) have quite different population densities and communication demands [55]. How to plan and deploy SBSs in advance to provide satisfactory services for a colossal amount of mobile users becomes an essential but challenging problem.

Some initial efforts have been made in developing efficient SBS deployment strategies to improve mobile network performance. The joint deployment of MBSs and SBSs over the urban area is presented in [33]. Author Byungchan studied the traffic distribution of mobile network and proposed a novel characterization of urban spatial traffic distribution based algorithm for the joint deployment of MBSs and SBSs to meet the fast-growing demands of mobile users. The deployment of SBSs in scenarios where MBSs are given in advance is investigated in [34] and [35], respectively. In [34], Author Xu proposed cooperative distributed of hyper-dense SBSs, which can alleviate the energy consumption of mobile devices and MBSs. In [35], author Pak proposed the cell-edge deployment of SMSes based on the performance of signal strength enhancement and the interference of other SMSes. However, the user in [35] is uniformly distributed, and this does not match the uneven distribution of users in reality. Nevertheless, the works in [33] [34] need to obtain realistic mobile network traffic records. However, these records are usually held by diverse operators and are not easy to acquire in practice.

To fill the gaps above, a data-driven deployment framework is proposed for SBSs with the presence of MBSs based on Twitter Records, which are operator-neutral. In [56], the author demonstrated that in a certain region, Twitter records density and mobile traffic have a log-linear relationship during the same period. Inspired by these observations, we propose a framework to use Twitter records to find the hot-spots of mobile traffic and deploy SBSs in these hot-spots. Without loss of generality, we use geo-tagged Twitter records generated in London and the surrounding suburbs. We first use the density-based spatial clustering of applications with noise (DBSCAN) algorithm [57] to identify hot-spots. Then we set a total amount of SBSs and SBSs assigned to hotspots are proportional to Twitter traffic and inversely proportional to the surrounding MBSs. We evaluate the performance of the proposed SBS deployment framework by imitated mobile user distributions. Experimental results show that, by deploying the same number of SBSs, the proposed framework can achieve a 16.6% increase in user received signal power compared with random SBS deployment strategies.

## 4.2 Twitter traffic form and model design

### 4.2.1 Parameters included in Twitter traffic

The dataset is based on more than four hundred thousands realistic Twitter records generated in Greater London and its surrounding suburbs over two weeks from 10 Jun 2012 to 24 Jun 2012. Specifically, these Twitter records contain information like timestamp and geographic location, as shown in Table 4.1. The collected Twitter data has two weeks' time span, and it is divided into two data sets evenly. To avoid the same data set causing over-fitting in the simulation, the data of the first week is used to find the location of the hot-spots and use data of the second week to represent the real location of users.

| User | ID | Time | Latitude & Longitude | Text |
|---|---|---|---|---|
| RiXXXX | XXXX92 | 10 Jun 2012 01:40:01 | 51.625918, 0.325665 | And my artwork... |
| SiaXXXX | XXXX15 | 20 Jun 2012 13:11:13 | 51.521388, -0.104591 | @rob...Street... But off the street... |
| SukXXXX | XXXX88 | 24 Jun 2012 02:50:12 | 51.798109, -0.092110 | I'm at Hertford... |

Table 4.1 Twitter record format: This table shows details of each tweet record, the main information as time and geo-location are included.

### 4.2.2 Model design

The two-week Twitter traffic is separated into two data sets weekly. Clustering hot-spots by DBSCAN algorithm is using the first-week traffic, and the second-week traffic is used to simulate the location of mobile users. In Fig. 4.1, the blue triangles in the map are the central points of relatively high traffic regions. The green star represent existing MBSs and red points represents the position of Twitter traffic.

## 4.3 SBS deployment framework

### 4.3.1 Overview

An SBSs deployment framework based on the DBSCAN algorithm is proposed. Firstly, the data passing the time is divided into two sets, namely, the hot-spot clustering data set and the data set simulating the location of the mobile user. Then, the DBSCAN algorithm is used to cluster the first week of Twitter traffic to get hot-spots. Finally, SBSs are deployed by getting the approximate coverage of the hot-spots and the number of Twitter traffic within

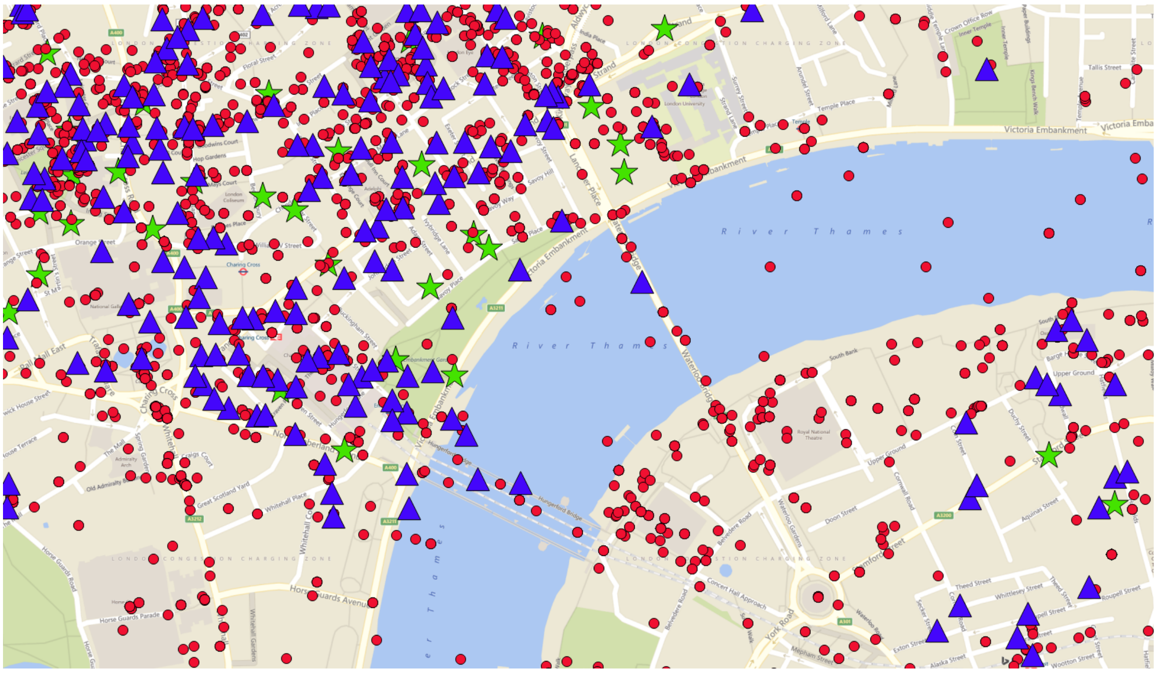


Fig. 4.1 Hot-spots, MBSs and Twitter traffic on the Great London: The blue triangles are the central points of relatively high traffic regions, the green star represents existing MBSs and red points represents the position of Twitter traffic.

the range. SBSs that are proportional to the Twitter traffic is deployed within the coverage of the hot-spot.

## 4.3.2 SBSs allocation formula

To allocate around two thousands of SBSs in London urban area (more than $100km^2$), the proposed framework bases on the number of MBSs and the number of Tweets in the coverage of hot-spots. Equation (4.1) shows the allocation coefficient $P$ is in proportion to the number of Tweets ($T_i$) in the coverage of a hot-spot and negative proportion to the number of MBSs ($M_i$). The $\gamma$ is the weight of the number of MBSs ($M_i$), and the $\sigma$ is the weight of the number of Tweets ($T_i$). The number of SBSs is allocated by the ratio of $P$ of the certain hot-spot to the whole. At last, SBSs assigned to each hot-spot within the coverage of the hot-spot is randomly deployed.

$$P = \sigma * T_i - \gamma * M_i \tag{4.1}$$

### 4.3.3 Path-Loss formula and received signal power of UE

Deploying SBSs in the current MBSs' environment, different path-loss algorithms are used to calculate the users' received signal power corresponding to the services of different base stations. For the MBSs in the considered London area, equation (4.2) & (4.3) calculate the path loss $L_0$ of the wireless users at a distance $d$ its associated BS. $f$ is the transmission frequency of MBS and $(h_B)$ is the MBS's antenna sufficient height. $(h_R)$ is the mobile antenna effective height, and $C$ is 3dB for metropolitan area [58].

$$\begin{aligned} L_0 = 46.3 + 33.9 * log_{10}(f) - 13.82 * log_{10}(h_B) - \alpha(h_R, f) \\ +(44.9 - 6.55 * log_{10}(h_B) * log_{10}(d) + C) \end{aligned} \tag{4.2}$$

$$\alpha(h_R, f) = (1.1 log_{10}(f) - 0.7) * h_R - (1.56 log_{10}(f) - 0.8)) \tag{4.3}$$

Equation (4.4) is the path-loth formula for simulated SBSs. $L_1$ is the calculated path loss with frequency $f$ from its associated SBS. The $c$ is the speed of light (in metres/s) [59].

$$L_1 = 20 * log_{10}(4 * pi * f / c) \tag{4.4}$$

# 4.4 Evaluation

## 4.4.1 Overview

In the actual mobile user distribution framework, the relationship existing in the previous section is used to simulate the number of mobile users with the second week's twitter strength. Using the second week's twitter strength to reverse the number of mobile users, then simulate the distribution of mobile phone users in different areas, and finally to the location of the simulated mobile user. For each mobile user, the received signal power of the nearest MBSs and SBSs is calculated and then takes stronger signal strength for users.

## 4.4.2 Two contrast framework

To demonstrate the validity of the proposed SBS deployment framework, two comparison frameworks are designed for comparison. The distribution of SBSs obeys Poisson and normal distribution in two comparison frameworks. Equation (4.5) is the probability density function of the Poisson distribution. The Poisson distribution is suitable for describing the number of random events occurring in a unit of area (or time). $\lambda$ of the Poisson distribution is the average number of occurrences of random events per unit area (or unit time). $e$ is the Euler's number, which is the base of the natural logarithms and $x$ is the natural number 0,1,2,.... In the comparison framework 1, $x$ represents the number of SBSs in Great London, and simulated SBSs deployment conforms to normal distribution.

$$f_p(x \mid \lambda) = \frac{\lambda^x}{x!}e^{-\lambda} \tag{4.5}$$

Equation (4.6) is the probability density function of a normal distribution. If the random variable x obeys a normal distribution with a mathematical expectation of $\mu$ and a variance of $\sigma^2$, it is denoted as $N(\mu, \sigma^2)$. In the comparison framework 2, $x$ represents the number of SBSs in London and simulated SBSs deployment conforms to a normal distribution. The mathematical expectation or expected value of the normal distribution is equal to the positional parameter, which determines the position of the distribution; the square root of the variance $\sigma^2$ or the standard deviation $\sigma$ is equal to the scale parameter, which determines the magnitude of the distribution.

$$f_n(x \mid \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{\frac{-(x-\mu)^2}{2\sigma^2}} \tag{4.6}$$

### 4.4.3 Framework evaluation and discussion

| Name | Parameter |
|---|---|
| MBSs transmission power | 46dB |
| SBSs transmission power | 30dB |
| Mobile Antenna effective height $(h_R)$ | 1.5m |
| MBSs Antenna sufficient height $(h_B)$ | 30m |
| Transmission frequency $f$ | 2100Mhz |

Table 4.2 Simulation parameters: This table lists the main parameters of MBS and SBS. They are used in equation 4.3, 4.4 and 4.5 of this chapter.

The received signal strength of the mobile user in the three frameworks is calculated and compared to the primary environment of the only MBSs. In the simulation, the essential parameters are listed in Table 4.2. It includes the effective height of the MBSs and mobile devices and the transmission power of MBSs and SBSs. The transmission frequency of the MBSs in the UK is set to 2100Mhz.

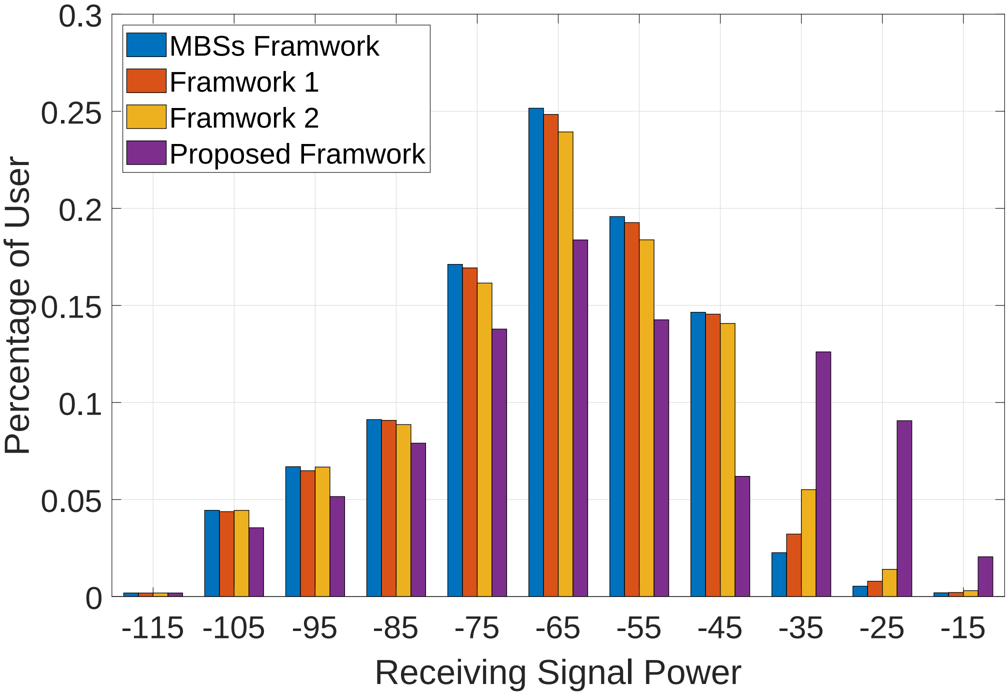


Fig. 4.2 UE Received signal power on overall London urban area: The bar chat shows the evaluation results. The received signal power is used to evaluate the performance of 3 framework. SBS deployment obeys Poisson distribution in the framework 1, and SBS deployment obeys normal distribution in the framework 2.

The simulation results of mobile users' received signal power in four frameworks are shown in Fig. 4.2. It shows the proportion of users in each received signal power interval. It can be observed that the framework of SBSs and MBSs joint deployment compares with a single MBSs framework; the users' received signal power is enhanced overall. In Fig. 4.2, the proposed SBSs deployment framework has the most apparent optimisation for the overall network.

In Table 4.3, the proposed framework can achieve a 16.4% increase in received signal power compared with conventional SBS deployment strategies, and it is the most effective framework. Moreover, Table 4.3 shows the Matlab operation time of 3 different SBSs deployments. The Poisson distribution is used in Framework 1 and the simulation time is

0.047s. The normal distribution is used in Framework 2 and the simulation time is 0.047s. The operation of the proposed framework needs 16.46s to simulate the SBSs deployment. Although the proposed framework requires more and more time to deploy SBSs, it improved better-received signal power. Also, the 16.46s simulation time is acceptable for the deployment of SBSs cell in a city centre.

| Simulation Area | Proposed Framework | Framework 1 | Framework 2 |
|---|---|---|---|
| Overall London Urban Area | 16.6% | 0.7% | 1.96% |
| Matlab Simulation Time (seconds) | 16.46 | 0.047 | 0.047 |

Table 4.3 Received signal strength optimization results: This table includes two significant results. The first row shows the improvement of user's received signal power of 3 frameworks when compared with the MBSs framework (MBSs framework only used the real world MBSs location). The second row shows the simulation time of SBSs deployment in 3 different frameworks.

## 4.5 Conclusion

In this chapter, an SBSs deployment framework is proposed for urban areas. The purpose is to find out the better SBSs deployment for urban areas of London. Twitter traffic is used as the main data set. Each tweet has precise geo-location and it is suitable to analyse by the clustering algorithm. DBSCAN is the clustering algorithm used in this chapter and this algorithm is better to find out the hot-spot of traffic in a certain area. After finding out the hot-spot by DBSCAN, the SBSs can be thoroughly targeted deployed.

To reflect the proposed framework has better performance in SBS deployment, the signal received power is used for evaluation. Moreover, two compared frameworks are also applied in this chapter. Those two comparison frameworks have different SBS deployment methods. SBS deployment obeys Poisson distribution in one framework, and SBS deployment obeys normal distribution in the other.

After evaluating the proposed framework and two comparison frameworks, the proposed framework has better-received signal power with conventional SBS deployment strategies. Although the proposed framework requires more simulation time to deploy SBS, it is almost negligible for deploying SBSs in a city centre area. Moreover, the proposed framework is very effective in user's received signal power.

Compared to the existing deployment of SBSs using the operator's data, social network traffic is easier to obtain than operator data and can be used to optimize the network. DBSCAN algorithm is used to analyse Twitter traffic to discover mobile traffic hot-spots. The Twitter data-driven SBSs deployment can effectively improve the user's received signal power.

# Chapter 5

# Deep Learning Enabled Prediction of Twitter Traffic with a Regional Granularity

## 5.1 Introduction

In the past few years, the number of OSN users has increased rapidly and OSN applications have become an important part of people's daily lives. According to a previous report, users in the U.S. are averagely spending one hour in OSN applications every day, while this amount reaches four hours in Philippines [65].

Developing trend of OSN applications has attracted many research efforts on the study of OSN application-specific traffic recently. In [61] and [62], author Qiu and Takeshita presented to use geo-tagged Twitter records to detect the QoS complaints from mobile users and the core network failures, respectively. Guo et al. further apply natural language processing techniques in Twitter contexts to uncover blackspots in mobile networks in London [63]. However, most existing works about OSNs mainly focus on exploiting the content in OSN records to solve sentiment-related problems. The characterization and prediction of OSN application-specific traffic have not been well studied yet.

Zhang proposes the Autoregressive Integrated Moving Average Model(ARIMA) to predict future values in a general time series based on the historical data [64]. In [65], author Tran decompose the time series of traffic load at every cell into three components (i.e., error, season, and trend) and then use the exponential smoothing model to predict each of them. In [66], author Wang, use stacked auto-encoders to extract the spatial correlation of mobile traffic and exploit the LSTM algorithm to forecast future traffic loads for different cells. Mobile traffic prediction models based on support vector machine and multi-layer perception are proposed and examined in [67]. Moreover, the study in [68] unveils that a strong heterogeneity exists in demand for different mobile applications, even if the applications belong to the same category, e.g., video streaming behaves quite differently in YouTube, Netflix, and iTunes platforms. Nevertheless, all these works do not discuss the traffic characteristics of OSN applications. Considering the high dynamics in OSN traffic, those existing prediction methods may not work well.

This chapter presents the first attempt at the characterization and prediction for Twitter traffic. The motivations are twofold: 1) In the fifth-generation (5G) and beyond to offer tailored services for specific applications, accurate understanding and prediction of the traffic load associated with a mobile application can enable the optimized planning and proactive optimization of the virtual network slice for the mobile application [69]. 2) In [56], author Yang have shown that the instantaneous Twitter traffic load and the instantaneous aggregated mobile traffic load in the same geographic location or region obey a strong power-law relationship. Thus, predicting the Twitter traffic loads with a regional granularity can provide an operator-neutral approach to understanding the distribution of aggregated mobile network traffic, which otherwise would be hard in practice since the mobile network data are operator-specific.

Based on the observation that the temporal variations of Twitter traffic loads generated in regions with the same social attribute (e.g., business district, residential district, and university) are likely to have a similar pattern, a Twitter traffic prediction framework is proposed, which clusters the geographical regions with similar traffic patterns into a group and predicts the Twitter traffic for each group of regions based on an LSTM algorithm. The proposed framework not only allows multiple regions to share the same prediction model but also expands the training set for each LSTM network, thus reducing the risk of over-fitting in the training process with respect to Twitter traffic's high dynamics. More specifically, the work is based on realistic geo-tagged Twitter records generated in London and the surrounding suburbs. The corresponding area is segmented into hundreds of small regions and present a PCC based clustering algorithm that can automatically cluster the regions with a similar Twitter traffic pattern into a region group. For each region group, the prediction model is trained by using the Twitter data collected from all the regions in this group. Experiment results show that the proposed framework can achieve a high prediction accuracy and enormously decrease the dependency on the relatively large training dataset, if it is compared to existing methods.

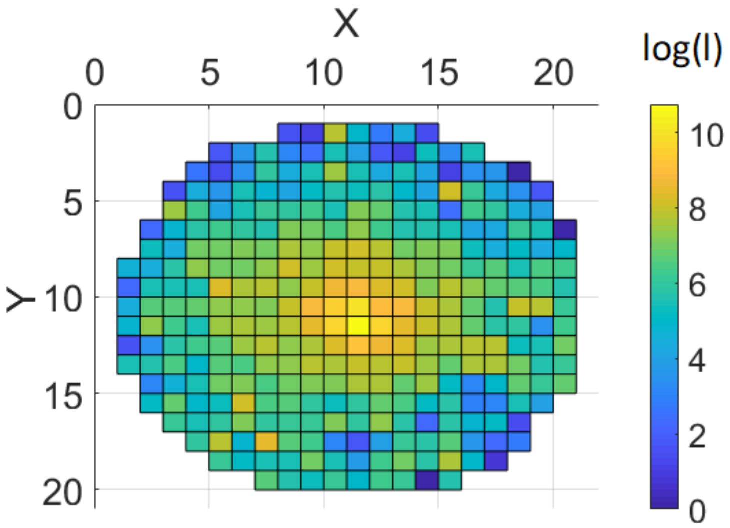


Fig. 5.1 Region segmentation of London with Twitter traffic: This figure shows the Twitter traffic in London with an area grid. The colour represents the number of tweets in each area in logarithmic form(l is the number the number of tweets in each area). The x-axis and y-axis are the order number of area grid. Moreover, the x-axis is the result of equal division of longitude, and the y-axis is the result of equal division of latitude.

## 5.2 Twitter traffic dataset and spatial-temporal characteristics

### 5.2.1 Twitter traffic display in area grid

In this chapter, the temporal resolution is set(e.g., the Twitter records collection interval), $\Delta t$, to 1 hour. The map of Greater London and its surrounding suburbs are segmented into 336 square regions, each of which has the same size of $4km \times 4km$. For simplicity of notation, $(x,y)$ is used to denote the region located at the intersection of the $x$th column and the $y$th row in the grid shown in Fig. 5.1. and use $l_{(x,y)}(t)$ to denote the number of Twitter records generated in this region from $t$ to $t+\Delta t$.

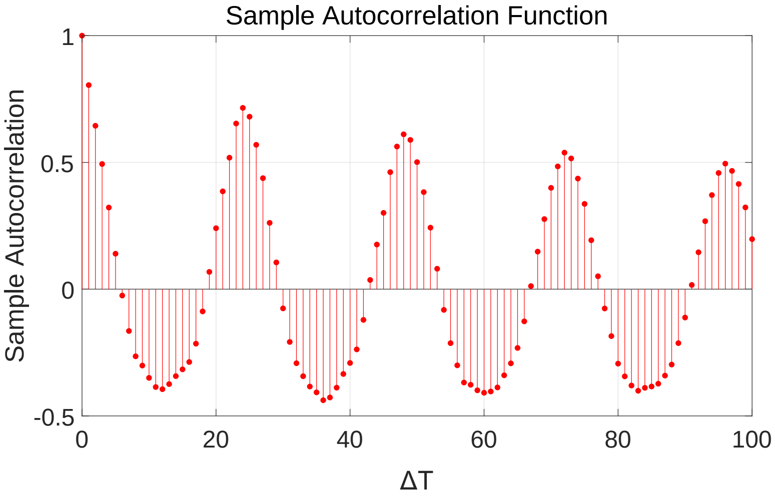


Fig. 5.2 Autocorrelation of Twitter records in a randomly selected region: This figure shows the autocorrelation of Twitter traffic. $\Delta T$ is 1 hour time interview. The strongest autocorrelation of Twitter traffic is when $\Delta T$ equal to 1 hour. The strongest period autocorrelation of Twitter traffic is 24 hours.

Fig. 5.1 gives an overview of the spatial distribution of Twitter records during the two weeks, where the volumes are shown in logarithmic form. The spatial distribution of Twitter traffic is extremely uneven over space. The central region of Great London, i.e., (11,11) in Fig. 5.1, contributes to the largest amount of Twitter records and its Twitter traffic volume is thousands of times those in the suburban regions at the edge in Fig. 5.1. Apart from this, more Twitter records tend to occur in regions like airports, tourist attractions, business districts, etc.

### 5.2.2 Spatial-temporal characteristics of Twitter traffic

Fig. 5.2 shows the temporal autocorrelation coefficient of a Twitter traffic vector generated in one randomly selected region. Operating autocorrelation is to prove that the data is cyclical and predictable on the time axis. The autocorrelation coefficient, $a_{\Delta T}$, is computed as follows:

$$a_{\Delta T} = \frac{\sum_{t=1}^{336-\Delta T} (l_{(x,y)}(t) - \bar{L}_{(x,y)})(l_{(x,y)}(t+\Delta T) - \bar{L}_{(x,y)})}{\sum_{t=1}^{336} (l_{(x,y)}(t) - \bar{L}_{(x,y)})^2} \tag{5.1}$$

,where $\bar{L}_{(x,y)}$ represents the mean value of $L_{(x,y)}$ over time domain. The traffic vector$L_{(x,y)}$, it archives the volume of Twitter records collected in region $(x,y)$ at different intervals. Each $L_{(x,y)}$ has the size of 336 (14day · 24hour). From Fig. 5.2, the Twitter traffic vector exhibits non-zero autocorrelations in the time domain, and the autocorrelation coefficient reaches the peak value when $\Delta T$ is an integer multiple of 24h. The similar phenomena in the Twitter traffic vectors collected in different regions, which implies that predict future Twitter traffic load can be predicted in an individual region by learning from historical Twitter records in this region.

In order to further explore the spatial-temporal characteristics of Twitter traffic, the PCCs between Twitter traffic vectors corresponding to different regions is calculated. PCC is defined as follows:

$$PCC_{X,Y} = \frac{E[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y} \tag{5.2}$$

where vectors $X$ and $Y$ have the same dimension, $\mu$ and $\sigma$ are the mean value and the standard deviation of a vector, respectively. When the PCC of two traffic vectors approaches 1, the Twitter traffic loads in the two regions will have similar temporal patterns.

| Strong Correlation | | |
|---|---|---|
| B(11,10) | B(11,11) | 0.8467 |
| R(10,9) | R(11,12) | 0.7843 |
| S(10,8) | S(13,11) | 0.6126 |
| Weak Correlation | | |
| A(5,9) | A(14,10) | 0.2426 |
| B(11,11) | R(11,12) | 0.3197 |
| B(11,11) | S(13,11) | 0.3035 |

Table 5.1 PCCs of Twitter traffic in Different Regions: B is business district, R is residential district and A is airport. Areas with the same regional attributes have a strong correlation with traffic.

Table 5.1 exhibits the PCCs between Twitter traffic vectors generated in some representative regions: regions (11,10) and (11,11) are business(B) districts, regions (10,9) and (11,12) are residential(R) areas, regions (5,9) and (14,10) contain airports(A), and regions (10,8) and (13,11) are in suburb (S). From Table 5.1, 1) the temporal variations of Twitter traffic in the considered regions may be quite different (e.g., the PCC between regions (11,11) and (13,11) is only about 0.3); 2) for regions with the same social attribute, the temporal patterns of Twitter traffic loads are more likely to be analogous (i.e., their PCCs are larger than 0.6); and 3) not all regions with the same social attribute have similar temporal traffic patterns (e.g., the PCC between regions (5,9) and (14,10) is less than 0.3).

The temporal variations of Twitter traffic loads in regions (11,10), (11,11), (10,9), and (11,12) during the same two days are shown in Fig. 5.3. From Fig. 5.3, the temporal traffic patterns in business districts are different from those in residential areas, while the regions with the same social attribute may have their peak traffic at similar times.

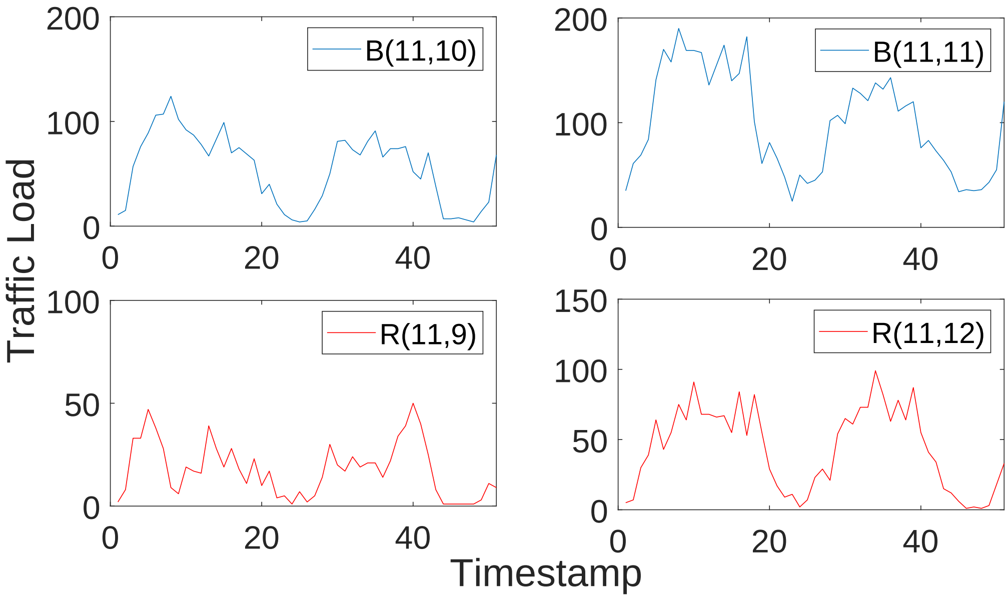


Fig. 5.3 Traffic Pattern: Business Districts vs. Residential Districts: B is business district and R is residential district. Areas with the same regional attributes have similar traffic pattern.

## 5.3 Prediction framework for Twitter traffic

Based on the above-observed characteristics of the Twitter traffic vectors, the LSTM network is proposed to forecast a region's future Twitter traffic load according to the region's historical Twitter traffic records. Note that training a distinct LSTM network for each region requires not only a huge workload (especially when the number of considered regions is large) but also adequate training data for each region. As regions of the same societal attribute may have analogous temporal Twitter traffic patterns, a dynamic region clustering algorithm is proposed and a prediction model is presented to predict traffic for regions with similar Twitter traffic patterns in the following two subsections, respectively.

## 5.3.1 Region clustering algorithm

Since two regions with a PCC approaching 1 are likely to have similar temporal Twitter traffic patterns, a dynamic region clustering algorithm based on PCC is presents in Algorithm 5.1. For a given region set *D*, the algorithm first finds the two regions with the largest PCC and determines whether the PCC is bigger than a certain threshold, $pcc_{thd}$, to put them into an initialized group, *G*. If *G* is set, the algorithm will continuously add regions from set *D* into group *G*, which make the mutual PCCs between regions in *G* larger than the threshold, $pcc_{thd}$. When no region can be added into *G*, the algorithm will output *G* as a new region group, subtract regions in *G* from *D*, and recurrently execute the above procedures until *D* is empty.

**Algorithm 5.1** Region clustering algorithm based on PCC

```
 1: input the region set $D$, Twitter traffic vector for each region in $D$, and the grouping
    threshold $pcc_{thd}$;
 2: if $|D| < 2$ then
 3:     output the region in $D$ as an individual group and stop the algorithm;
 4: else
 5:     calculate the PCC between every region pair in set $D$;
 6:     find regions $r_1$ and $r_2$ with the largest PCC, $pcc_{max}$;
 7:     if $pcc_{max} < pcc_{thd}$ then
 8:         output each region in $D$ as an individual group, set $D = []$;
 9:     else
10:         put $r_1$ and $r_2$ into a new group $G$, set $D = D - r_1, r_1$;
11:         for region $r_i$ in $D$ do
12:             put $r_i$ into $G$ if the PCC between $r_i$ and an arbitrary region in $G$ is not smaller
                than $pcc_{thd}$;
13:         end for
14:         output group $G$, set $D = D - G$;
15:     end if
16: end if
17: go to step 1;
```

Compared with traditional clustering algorithms such as K-means and DBSCAN, the proposed Algorithm 5.1 does not need to calculate the Euler distances between different Twitter traffic vectors, nor does it need to specify the expected number of groups or beginning points when implementing the algorithm. So Algorithm 5.1 has a relatively low computational complexity and can automatically capture the spatial characteristics of Twitter traffic. Moreover, Algorithm 1 can be used for dynamic region clustering in different cities and/or mobile traffic generated by other applications.

### 5.3.2 LSTM network based traffic prediction model for each region group

Recurrent neural networks (RNNs) allow the information of historical inputs to be stored in the internal state, thus being capable of learning the knowledge of temporal sequences. However, for the stand RNN architecture, the influence of a past input on the network output would either decay or blow up exponentially when it cycles around the network's recurrent connections. LSTM algorithm is used to model the long-term influence of temporal sequences [70], which has been successfully applied in various temporal sequence processing such as language modelling, semantic analysis, and stock trend forecasting. It contains a memory cell to remember the temporal states of the network and three gates, which are the input gate, the forget gate, and the output gate, respectively, to control the flow of information

In order to accelerate the training speed of the prediction models, the Twitter traffic vector related to each region is separately normalized into the range of (0.1) using max-min scaling method. For a group of regions formed in Algorithm 1, a shared LSTM network-

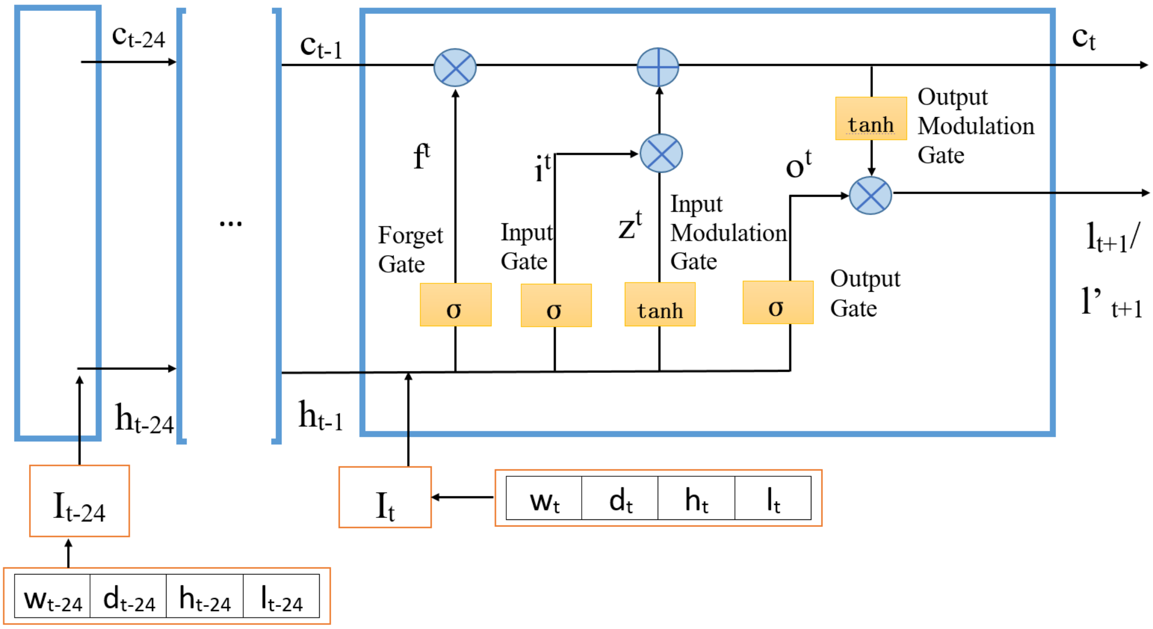
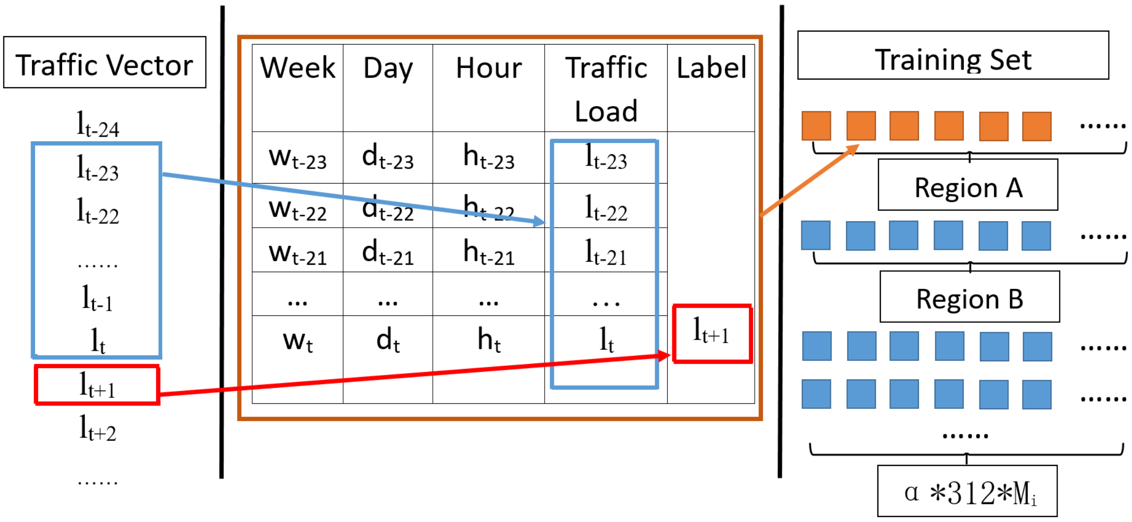


Fig. 5.4 An illustration of LSTM Process

based traffic prediction model is proposed as illustrated in Fig. 5.4. If group $i$ consists of $M_i$ regions, its prediction model will be trained and tested by samples gathered from all the $M_i$ regions. Based on the observation in Fig. 5.2., that the autocorrelation coefficient of a Twitter traffic vector varies roughly in a period of 24 hours, the length of each sample's input sequences in the set as 24 to get the best prediction accuracy. The input sequences of a sample are denoted by $[I_{t-23},...I_{t-1},I_t]$. Each input vector, $I_t$, for the LSTM network contains two attributes: one is the normalized Twitter traffic load generated a specific region in the $t_{th}$ time interval, $(l_t)$ and the other is the particular time interval, which is further segmented into the week $(w_t)$, the day in a week $(d_t)$, and the hour in a day $(h_t)$. A sample's label is given as the actual number of Twitter records generated in the next time interval, $(l_{t+1})$. Specifically, $w_t$, $d_t$, and $h_t$ are also normalized into the range of $(0,1)$.

If apply a fixed sliding window with a length of 25 to split a region's Twitter traffic vector, then 312 samples will be built for this region. Thus, for group $i$ consisting of $M_i$



| Week | Day | Hour | Traffic Load | Label |
|---|---|---|---|---|
| $w_{t-23}$ | $d_{t-23}$ | $h_{t-23}$ | $l_{t-23}$ | |
| $w_{t-22}$ | $d_{t-22}$ | $h_{t-22}$ | $l_{t-22}$ | |
| $w_{t-21}$ | $d_{t-21}$ | $h_{t-21}$ | $l_{t-21}$ | |
| … | … | … | … | |
| $w_t$ | $d_t$ | $h_t$ | $l_t$ | $l_{t+1}$ |

Fig. 5.5 An illustration of LSTM Training Set

regions, a total number of $312 \cdot M_i$ samples are gathered. $\alpha \cdot 312 \cdot M_i$ samples is randomly selected to construct the training set for its LSTM network-based traffic prediction model while the remaining samples are used to construct the testing set. The construction of an LSTM network's training set and testing set is illustrated in Fig. 5.5.

### 5.3.3 Traffic prediction performance evaluation

In this chapter, the performance of the proposed Twitter traffic load prediction framework is tested. The prediction model is constructed for each region group as a three-layer LSTM network which is stacked by three LSTM memory blocks. Specifically, the output vectors of the first and the second LSTM memory blocks have the dimensions of 2 and 1, respectively. As the dataset only contains the Twitter records generated in two weeks, the number of training samples related to one region is relatively small. In order to evaluate the prediction accuracy of the proposed framework, performance with several of the most up-to-date methods is compared:

- ARIMA (Autoregressive Integrated Moving Average model) [64]: As the most widely used time series forecasting method, ARIMA is a representative statistic based model and has low computational complexity.

- Conventional LSTM network [66]: Conventional LSTM network is a deep learning based method that achieves the state-of-the-art mobile traffic prediction performance.

For conventional LSTM network-based perdition algorithm, an individual LSTM network is trained for each region by samples collected from it. In order to guarantee fairness, the conventional LSTM network has the same structure as the ones in the proposed framework. The same ratio of samples is used in a region to construct the training set ($\alpha \cdot 312$ samples).

In the experiments, $\alpha = 0.5$ is set. MSE is choosing as the loss function, while the Adaptive Moment Estimation (Adam) algorithm with default learning rate is utilized to optimize the deep learning-based models. MAE and MSE are used to estimate the prediction accuracy in this work. MAE is the average absolute difference between realistic and predicted traffic demand. MSE is the average squared difference between realistic and predicted traffic demand. Lower values of both two-parameter are better and zero means no error. The expressions of MAE and MSE are displayed as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|x^i - y^i| \tag{5.3}$$

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(x^i - y^i)^2 \tag{5.4}$$

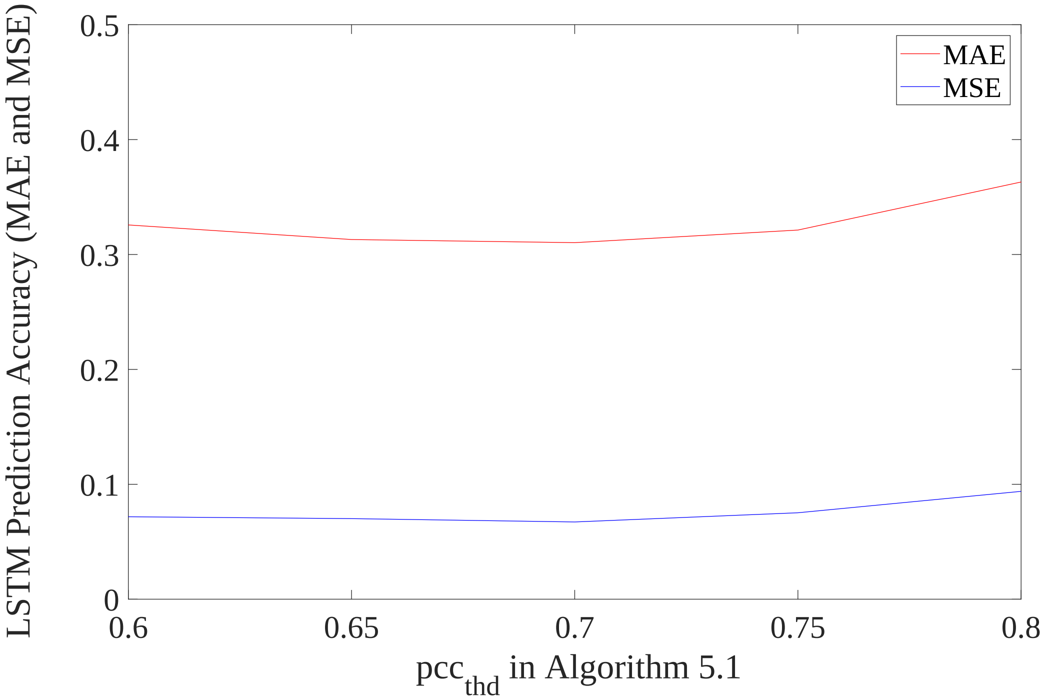


Fig. 5.6 The relationship between LSTM prediction accuracy with the $pcc_{thd}$ in Algorithm 5.1: x-axis is the value of $pcc_{thd}$ and y-axis is the value of MSE and MAE.

where $n$ is the total number of tests, $x^i$ and $y^i$ are the real value and the predicted value, respectively.

Fig. 5.6 shows the relationship between the proposed framework's prediction accuracy with the $pcc_{thd}$ in Algorithm 5.1. From Fig. 5.6, the proposed framework's prediction accuracy will first increase as $pcc_{thd}$ gets larger and then decrease after $pcc_{thd}$ reaches a certain level. This phenomenon can be explained as when $pcc_{thd}$ is small, regions with quite different Twitter traffic patterns are likely to be grouped together and using these regions' samples to train an LSTM network may jeopardise the prediction performance on every individual region. On the other hand, when $pcc_{thd}$ approaches 1, very few regions will be clustered in each region group and the prediction model related to this group will be over-fitted due to the lack of training data. The proposed framework can obtain the optimal performance if $pcc_{thd}$ is between 0.65 and 0.75 from Fig. 5.6. This finding might provide guidance to the proposed framework in practical scenarios when the training samples for every single region is inadequate.

Fig. 5.7 illustrates the average MAE and MSE achieved by the proposed framework with $pcc_{thd}$ = 0.7 and the baseline methods over regions in the Greater London and surrounding suburbs. As can be seen from Fig. 5.7 clearly, ARIMA performs the worst among all the considered methods. This can be explained as mobile Twitter traffic loads have highly nonlinear patterns in the temporal dimension, which makes the future traffic load forecasting beyond the ability of linear models. Due to the high complexity of structure, conventional LSTM network has a strong ability to represent the nonlinearities in mobile Twitter traffic patterns and might learn the deep dependency between traffic loads generated in various time intervals. So in Fig. 5.7, the conventional LSTM network performs better than ARIMA. The proposed framework achieves the highest prediction accuracy among the considered methods. This can be explained as conventional LSTM network-based prediction models need a large number of samples to train their parameters. Assembling samples from multiple regions with similar Twitter traffic patterns in the proposed framework will enlarge an LSTM network's training set and then relieve the over-fitting problem.

Table 5.2 demonstrates the performance of the prediction frameworks for seven representative regions in Greater London, which have the highest volumes of Twitter records. The B(11,10) and B(12,11) are the business district in the centre of Greater London and other regions in Table II are the residential district. As the number of considered regions in Greater London surrounding suburbs is very large, the considered methods' prediction performance can not be listed for them neatly due to the limited space.

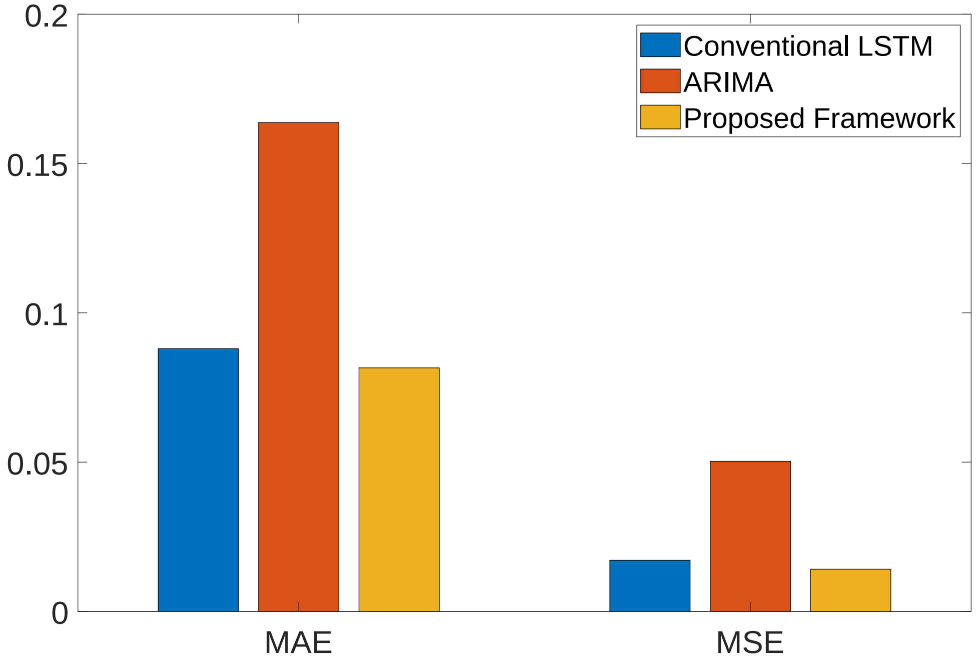


Fig. 5.7 The average MAE and MSE of three frameworks: MSE is the average squared difference between realistic and predicted traffic demand. MAE is the average absolute difference between realistic and predicted traffic demand. y-axis is the value of MSE and MAE. Lower values of both two parameter are better and zero means no error.

Table 5.2, it shows that the proposed framework achieves the best prediction accuracy for most of the listed regions. This can also be explained by the fact that the PCC based region clustering algorithm will group the regions with similar Twitter traffic patterns. Trained by samples from regions in the same group, an LSTM network-based prediction model can not only capture the inherent characteristics of these similar temporal patterns but also avoid the risk of over-fitting in the training process.

## 5.4 Conclusion

According to realistic geo-tagged Twitter record dataset generated in Greater London and surrounding suburbs, the temporal-spatial characteristics of Twitter traffic in a region gran-

| No. | Conventional LSTM | | ARIMA | | The proposed Framework | |
|---|---|---|---|---|---|---|
| | MAE | MSE | MAE | MSE | MAE | MSE |
| R(10,9) | 0.107 | 0.024 | 0.167 | 0.047 | 0.082 | 0.014 |
| R(11,9) | 0.072 | 0.016 | 0.098 | 0.021 | 0.077 | 0.015 |
| R(9,10) | 0.073 | 0.013 | 0.112 | 0.027 | 0.066 | 0.010 |
| R(11,12) | 0.059 | 0.011 | 0.104 | 0.022 | 0.066 | 0.012 |
| R(13,10) | 0.102 | 0.019 | 0.161 | 0.047 | 0.088 | 0.014 |
| B(11,10) | 0.120 | 0.025 | 0.267 | 0.107 | 0.111 | 0.021 |
| B(12,11) | 0.083 | 0.012 | 0.237 | 0.081 | 0.081 | 0.013 |

Table 5.2 Prediction Results for Ten Regions in Greater London

ularity is studied. Thus, proposed a Twitter traffic prediction framework, which combines a region clustering algorithm and the deep learning technology. Experiment results prove that the presented LSTM network-based prediction model will efficiently capture the trend of regional Twitter traffic and the sharing strategy of multiple regions with a similar traffic pattern can further improve the prediction model's accuracy distinctly.

Because the characterization and prediction of OSN application-specific traffic have not been well studied yet. In this chapter, the research aims to propose a framework for the application-specific traffic prediction. Based on the study in the previous chapter, it shows the instantaneous Twitter traffic load and the instantaneous aggregated mobile traffic load in the same geographic location or region obey a strong power-law relationship. Thus, predicting the Twitter traffic loads with a regional granularity can provide an operator-neutral approach to understanding the distribution of aggregated mobile network traffic, which otherwise would be hard in practice since the mobile network data are operator-specific.

Firstly, the traffic data is segmented into the grid and study the relationship between the region attributes with Twitter traffic. Based on the observation, the time variation of Twitter traffic load generated in areas with the same social attributes (for example, business districts, residential areas, and universities) may have similar patterns. Then, the PCC is used to clustering Twitter data with area grid based on the Twitter traffic load generated in areas with the same social attributes have similar patterns. The conventional LSTM algorithm is already used to predict traffic in the wireless communication network. However, training different LSTM networks for each region not only requires a huge workload (especially when the number of regions considered is large) but also requires sufficient training data for each region. Thus, a Twitter traffic prediction framework that divides geographical areas with similar traffic patterns into a group is proposed and it can be used to predict the Twitter traffic of each area group based on a long short-term memory (LSTM) network.

In the evaluation section, the ARIMA and conventional LSTM are used as comparison algorithms as well. The MAE and MSE are used to estimate the prediction accuracy in this chapter. The proposed framework combined the regional clustering of traffic data and LSTM algorithm. The corresponding areas are divided into hundreds of small areas and propose a clustering algorithm based on the PCC, which can automatically cluster areas with similar Twitter traffic patterns into area groups. Thus, this section expresses the PCC impact on the results of regional clustering based on traffic. According to the evaluation results, the proposed framework which is LSTM network-based prediction model will efficiently capture the trend of regional Twitter traffic. The improvement of user's received signal power proved the sharing strategy of multiple regions with a similar traffic pattern can further improve the prediction model's accuracy distinctly.

# Chapter 6

# Crowding Events Twitter Traffic Pattern Detection and Classification

## 6.1 Introduction

As researchers learn more about mobile traffic, they find that traffic has its pattern. However, some special events affect the traffic pattern in a short time. Author find that in some crowded events, the amount of mobile traffic increased by a few dozen for the usual situation. Besides, different events have different effects on the traffic pattern. This impact on cellular network operators is a big challenge [71].

Author Xu proposed a model to extract and simulate the mobile traffic of large cities [72]. The time and the location of the tower and the traffic spectrum are used to extract and simulate the traffic pattern of the base station and classify them with different functional areas. Different functional local users face various daily events, which leads to different mobile traffic patterns.

Daily events do not affect mobile traffic pattern much. The impact of some special events on mobile traffic increases in a short period. The prediction of these special events is particularly important for the wireless communication network's optimisation and prevention. Author Wang analysed the flow of data from non-sensitive data of traffic cards to find out the fluctuations of data before and after special events [73]. Moreover, time series and machine learning are used to find early unconventional special events. Author Yang analysed the occurrence of special events, such as graduations, data collected on vehicle GPS, and user's traffic smart card data. [74].

Author Au found out the significant special events attracted mobile phone users to certain areas, which leads to a sudden surge in traffic. Author Au proposed a method to estimate the probability of occurrence of the event. This method is used to detect the location and time of the event, and to evaluate the spatial and temporal impact of the event [75]. This approach combines the Poisson process with the Markov Modulated Nonhomogeneous Poisson Process.

In this chapter, analysing traffic pattern in different regions by DTW [76] is used to verify different congestion events. Firstly, finding out the significant increase in the social network traffic of small regions when a crowding event happened. The occurrence of crowding event leads to population aggregation in some regions. Then, the grid is used to classify the traffic map and collect data for each region. It includes areas that are affected by crowding events and are not affected. The detection function is used to detect traffic sets with the effect of the crowding events. The small regions of the different patterns are integrated into the standard crowding events patterns. The DTW algorithm is used to test the accuracy of the remaining traffic sets. The trained method can distinguish the traffic pattern when different crowding events occur.

## 6.2 Crowding events traffic pattern and detection function

### 6.2.1 Overview

The dataset comprises more than four hundred thousands realistic Twitter records generated in the New-York City and Mexico City area from 20/02/2017 to 05/03/2017 and the Paris area from 02/11/2015 to 22/11/2015. Specifically, each tweet record contains a time-stamp and the geographic location information.

The timeline of the dataset contains two unusual crowding events, which are Man-made social destruction and 2017 Mobile World Congress. The 2017 Mobile World Congress is set as Event A and Man-made social destruction in Paris is set as Event B. The characteristics and traffic pattern of unusual crowding events is different than traffic pattern in usual life. This chapter proposes the detection function and the DTW algorithm analysis of the traffic with crowding events.

### 6.2.2 Crowding events traffic pattern

Fig. 6.1 shows the traffic pattern of Event A in New York City (Mexico City is similar to New York City) and Event B in Paris. Two different patterns have a significant increase in traffic at a particular time point of events. The first half part of the figure represents

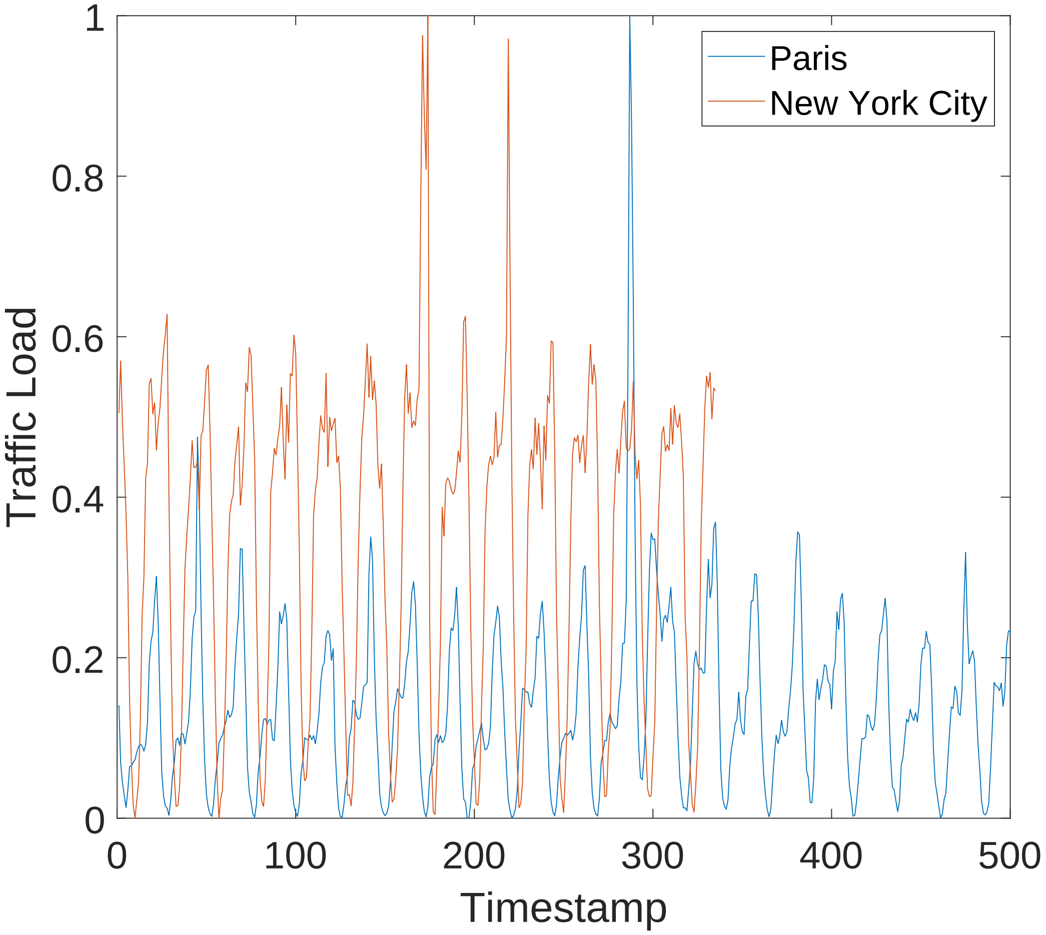


Fig. 6.1 Events Overall Pattern

the traffic pattern of normal life. However, the increase in traffic was caused by people's attention to the event when crowding events occurred.

### 6.2.3 Detection function

The detection function is used for filtering traffic dataset. Each city's traffic map is divided into 400 small areas using a grid. Twitter traffic is divided into different regions according to their geographic location. Each data set represents the traffic pattern of a grid region. The detection algorithm filters out small areas with insufficient total traffic and then find datasets similar to the crowding events pattern. Both New York City and Mexico City have data in 75 areas with enough traffic to form a 14-day traffic pattern. The detection algorithm finds out 21 and 18 regions in these two cities that match Event A's traffic pattern. Paris has 112 areas of data with enough traffic to form a 22-day traffic pattern and 42 regions matching Event B's traffic pattern. Table 6.1 shows the filtering results. The proposed detection algorithm is in Algorithm 6.1.

**Algorithm 6.1** : Traffic Pattern Detection Algorithm

```
input the region set $D$, Twitter traffic vector for each region in $D_{xy}$, the Traffic amount
  threshold $A$ (the average daily traffic demand), and the standard traffic set $S$ of Event A
  or B;
for region $x$ $y$ in $D$ do
    if $Sum(D_{xy}) < A$ then
        $D_{xy}$ is removed from $D$;
    end if
    find the time interval $T$ in $S$ which the crowding event occurred;
    find regions max traffic time interval $t_{xy}$ contrast with $T$;
    if $t_{xy} \in T$ then
        transfer the traffic set $D_{xy}$ in output set $O$;
    else
        $D_{xy}$ is removed from $D$;
    end if
    until $D$ becomes an empty set;
end for
output $O$;
```

| *City* | Total Number of Traffic sets | Number of Traffic sets effected by Crowding Events |
|---|---|---|
| Paris | 112 | 42 |
| New York City | 75 | 21 |
| Mexico City | 75 | 18 |

Table 6.1 Number of Traffic Sets

## 6.3 Classification framework and DTW algorithm

### 6.3.1 DTW algorithm results

Euclidean distance (also known as Euclidean metric) is a commonly used distance definition. It refers to the true distance between two points in the m-dimensional space, or the natural length of the vector (that is, the distance from the point to the origin). Euclidean distance in two-dimensional and three-dimensional space is the actual distance between two points. However, the traffic pattern in a regional grid is a continuous time-series data and Euclidean distance is not suitable to describe the distance between two continuous time-series data. The DTW algorithm is used to measure the similarity of two-time series, and the lengths of the two-time series may not have to be equal. Thus, the length of the traffic data does not need to be adjusted to be the same in advance. The algorithm calculates the distance between tracks by an iterative method, which is a classic dynamic programming problem. It is based on the idea of dynamic programming (DP), which solves the problem of the template (or traffic data) matching with different lengths. In conclusion, this algorithm is used to find the shortest path by constructing an adjacency matrix.

Fig. 6.2 shows the result of the DTW algorithm. The left side of the figure is the traffic pattern of Event A, and the line below is a testing sample. The distance represents the value represented by each small square in the middle of this large matrix. This value represents the Euclidean distance between the two training patterns. The continuous time-series data is normalized from range 0 to 1. The white line in the middle of the figure is the shortest

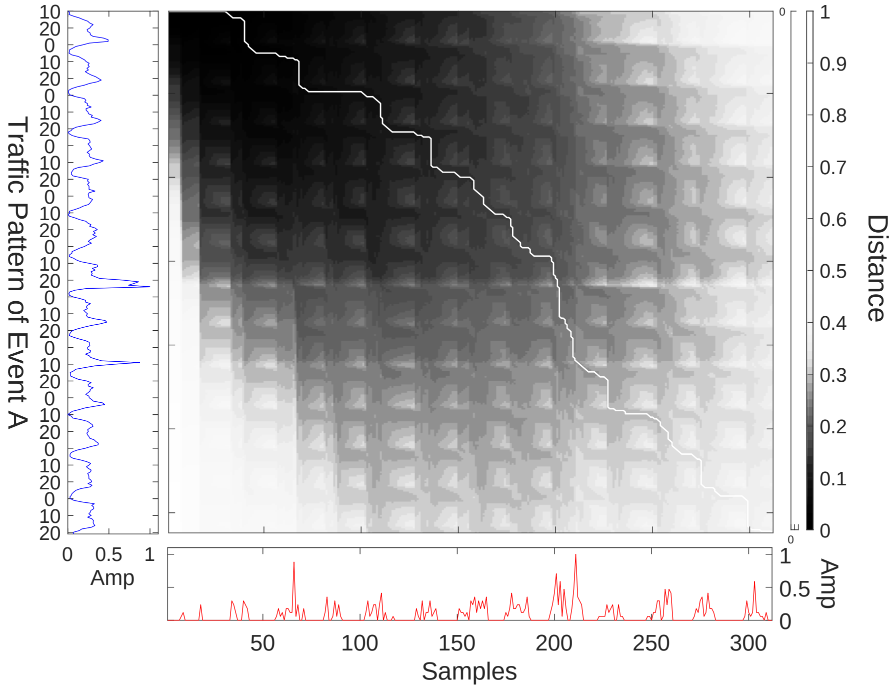


Fig. 6.2 Example of DTW Algorithm

path by constructing an adjacency matrix. Fig. 6.3 shows the warping of traffic set in the timestamp. The left-hand side image shows the original traffic curve, and the lengths of the two curves on the time axis are different. The warping result shows in the right-hand side image. In the DTW algorithm, the time series is extended and shortened at first. Then, it calculates the Euclidean distance between the two continuous time-series data.

### 6.3.2 Classification framework

The proposed classification is in Algorithm 6.2. The classification based on the DTW algorithm is proposed as mentioned. The detection algorithm is used to get the filtered traffic of Event A and Event B and label the different event sets. Then, randomly select 70% of the data set as the training set. Meanwhile, the framework trains the standard pattern for Event

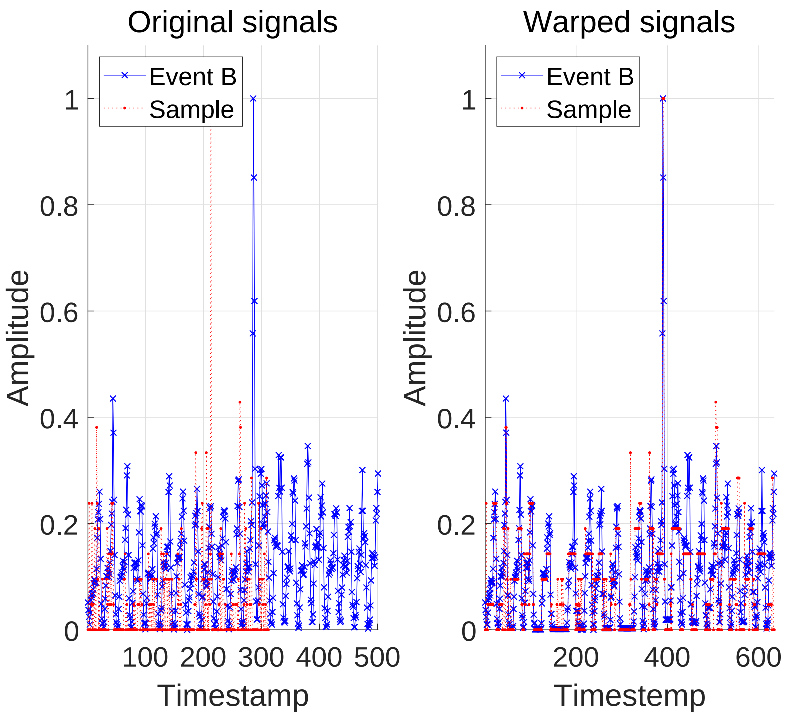


Fig. 6.3 Warping renderings

A and Event B. The remaining 30% of the traffic set is used as the test set and classify it by the DTW algorithm. Finally, the classification accuracy is obtained by comparing the label and classification results of the test set.

### 6.3.3 Unusual crowding events traffic classification evaluation

Table 6.2 shows the accuracy of the classification evaluation. After 50 cycles of the entire framework, it achieves the highest accuracy of 10 classification results. Because the training set and testing set are randomly selected from the dataset. The iteration is needed to prevent the classification is an effective method to classify the different traffic pattern of events. The high accuracy indicates that the proposed framework can classify traffic patterns in different

**Algorithm 6.2** : Classification Framework

```
input the Events traffic set $E$ and input the label $L$;
for Number of repeats do
    randomly select 70% of $E$ set as training set $T$;
    put rest 30% of $E$ set as test set $t$;
    train our standard pattern of Event A $P_A$ and Event B $P_B$ by superimposing the mean of training set $T$;
    for Number of test set $t$ do
        find the similarity $s_A$ of standard pattern $P_A$;
        find the similarity $s_B$ of standard pattern $P_B$;
        compare $s_A$ and $s_B$ to get classification results $C$;
    end for
    compare classification results $C$ and label $L$ to get classification accuracy $Acc$;
end for
output $Acc$;
```

regions with different events. In conclusion, this sires method not only find out the traffic pattern in the region is affected by the crowding events, but also Identified which events belong to the region where the traffic pattern is located.

| $No.ofEvaluation$ | Accuracy | $No.ofEvaluation$ | Accuracy |
|---|---|---|---|
| 1 | 94.4% | 6 | 77.8% |
| 2 | 88.9% | 7 | 77.8% |
| 3 | 83.3% | 8 | 77.2% |
| 4 | 77.8% | 9 | 77.2% |
| 5 | 77.8% | 10 | 77.2% |

Table 6.2 Accuracy of Classification Evaluation

## 6.4 Conclusion

Based on the real geotagged Twitter record dataset generated by Paris, New York City, and Mexico City, the temporal and spatial characteristics of Twitter traffic are analysed when it was affected by two different special events. A region traffic pattern framework based on detection algorithm and time-series mining DTW is proposed. Two different traffic patterns can be aligned by linear uniform expansion. However, this calculation does not take into account the long or short changes in the duration of each segment of data in different situations. Therefore, the recognition effect may not be the best. Dynamic programming (DP) in the DTW algorithm can solve this problem well.

The experimental results show that special crowding events influence the regional traffic pattern. The framework can be used to classify the traffic patterns of unknown regions by learning the standard patterns of special crowding events. The result can indicate whether the area is affected by the particular crowding events and which event is affected. However, this result needs more data to support (Multiple types of special crowding events' traffic data, traffic data in a different type of areas, and traffic data from different countries).

# Chapter 7

# Conclusion

## 7.1 Conclusion

In this thesis, social network data is fully explored and analyzed. There are three main researches aspect of the social network data. First, we use DBSCAN to find hot spots for location Twitter to guide the deployment of SBSs. Then, we propose a Twitter traffic prediction framework that combines regional clustering algorithm and deep learning technology. Finally, we propose a classification framework for the regional traffic pattern by analyzing the special Crowding event.

In the first research aspect, we mainly explored the geographical location information of Twitter traffic. We use traditional mining analysis methods and unsupervised learning clustering algorithm to process Twitter traffic. We compared two clustering algorithms, K-means and DBSCAN. From the clustering effect diagram, we conclude that DBSCAN is more suitable for analyzing Twitter traffic. Density-based clustering better presents the distribution of traffic in real life. According to DBSCAN, hotspots in Traffic maps can be found and combined with the promotion of SBSs deployment in recent years. We propose a framework based on the DBSCAN algorithm to optimize the deployment of small base stations. This framework is suitable for deploying SBS in an existing MBS environment. We assign SBS based on the density of hotspots after clustering and the number of MBSs near the hotspot. The deployment optimization effect is judged by calculating the signal of the UE to receive power. We compared the proposed framework with two traditional distributed model frameworks. Discover that our framework works best for SBS deployments.

In the second research aspect, the research focuses on geographic location information and time-series relationships in Twitter traffic. Based on the real geo-tagged Twitter record dataset generated by Greater London and surrounding suburbs, we studied the Spatio-temporal characteristics of Twitter traffic with regional granularity and proposed a Twitter traffic prediction framework combining regional clustering algorithm and deep learning technology. First, we use a PCC-based region clustering algorithm. This clustering algorithm has the advantage over the traditional unsupervised learning clustering algorithm in that it can find similar regions from the correlation of the data itself. A bit of the Twitter traffic prediction framework combined with the region clustering algorithm and the deep learning technique is that when the time axis span of the data set is relatively short, the training set of the LSTM can be enhanced by integrating data that belongs to the unified class region. Finally, LSTM can effectively predict Twitter traffic. The experimental results show that the proposed LSTM network-based prediction model will effectively capture the trend of regional Twitter traffic, and the sharing strategy of multiple regions with similar traffic patterns can further significantly improve the accuracy of the prediction model.

In the last research aspect, we are studying more deeply based on the second part of the research. From the research results in the second part, we can conclude that our framework can effectively predict Twitter patter in daily life. However, there are always some special crowding events in real life. We have found that these special crowding events can cause people to gather in small areas. This situation leads to a significant increase in the total amount of traffic during the period in which the event occurred. To this end, we are proposing a regional traffic pattern classification framework. We first classify the traffic map and then detect the changes in the traffic pattern within each mesh region. Find the area affected by the special congestion event from the entire traffic set. Then, through training, the traffic pattern of the special crowded event is obtained. Finally, the DTW algorithm is used to classify the regional traffic pattern. The accuracy of our proposed classification framework is relatively high enough to classify regional traffic patterns. It also proves that this framework can detect whether a region's traffic pattern is affected by a particular congestion event.

## 7.2 Future Work

In this thesis, we have conducted a lot of analysis and research on Twitter traffic. The SBS deployment can be optimized based on the geographic location information of the Twitter data. Traffic is effectively predicted by time series and region classification. And proposed a classification framework to classify the regional traffic pattern. Some of the potential future topics based on these efforts are summarized below.

As can be seen in Chapter 3 & 4, we used mining analysis and clustering algorithms for the research of Twitter traffic. In addition to these two methods, more machine learning algorithms can be tried, such as supervised learning and deep learning. Our proposed framework for SBS allocation only considers the density of clustering hotspots and the format of nearby macro base stations. More SBS deployment conditions can be added, such as more realistic geo-environment parameters.

As can be seen in Chapter 4, a Twitter traffic prediction framework combining regional clustering algorithm and deep learning technology is proposed. The prediction algorithm for this framework is LSTM. More Deep learning algorithms can be used in predictive research. Besides, the proposed framework is relatively poorly predictive of suburban traffic. This problem is caused by an insufficient number of traffic in the suburbs. Using the generic adversarial network (GAN) to simulate real data may improve the problem of insufficient traffic in the suburbs.

As can be seen in Chapter 6, a detection-based classification framework was proposed. The main focus of the research is on the classification of different special crowding events. A comparison of the particular crowded traffic pattern and the daily routine traffic pattern can be added. Besides, more special congestion events can be added to the data set. Let the proposed classification framework classify more than two special congestion events.

In addition to the aforementioned future work based on this thesis, the proposed framework can be optimized by adding data capabilities and more Machine learning algorithms. The framework may be used on different data sources to find similarities and differences between different data sources. Use more Machine learning algorithms to mine more meaningful information and patterns hidden in the data. We believe that machine learning-based social network data analysis and prediction makes sense for RAN network optimization.

# References

[1] J. Zander, ‘Beyond the Ultra-Dense Barrier: Paradigm Shifts on the Road beyond 1000x Wireless Capacity’, *IEEE Wireless Communications*, **vol. 24, no. 3**, pp. 96-102, 2017.

[2] Y. Lv, H. Zhang,S. Xueming, ‘Analysis of base stations deployment on power saving for heterogeneous network’, *International Conference on Communication Technology Proceedings, ICCT*, **vol. 2017-October**, pp. 1439-1444, 2018.

[3] P. Zhang, B. Li, ‘Research on optimizing deployment two GSM base stations-based passive radar’, *Proceedings - 2011 4th International Symposium on Knowledge Acquisition and Modeling, KAM 2011*, pp. 2-4, 2011.

[4] G. Aceto, D. Ciuonzo, A. Montieri et al., ‘Mobile Encrypted Traffic Classification Using Deep Learning’, *TMA 2018 - Proceedings of the 2nd Network Traffic Measurement and Analysis Conference*, pp. 1-8, 2018.

[5] G. Aceto, D. Ciuonzo, A. Montieri et al., ‘Network Traffic Classifier with Convolutional and Recurrent Neural Networks for Internet of Things’, *IEEE Access*, **vol. 5** pp. 18042-18050, 2017.

[6] X. He, W. Jia, Q. Wu et al., ‘Basic transformations on virtual hexagonal structure’, *Proceedings - Computer Graphics, Imaging and Visualisation: Techniques and Applications, CGIV’06*, **vol. 2006**, pp.243-248, 2006.

[7] H. Elsawy, E. Hossain, M. Haenggi, ‘Stochastic geometry for modeling, analysis, and design of multi-tier and cognitive cellular wireless networks: A survey’, *IEEE Communications Surveys and Tutorials*, **vol. 15, no. 3**, pp. 996-1019, 2013.

[8] J. Zhang, W. Wang, X. Zhang et al., ‘Base stations from current mobile cellular networks: Measurement, spatial modeling and analysis’, *2013 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*, pp. 1-5, 2013.

[9] Y. Zhou, R. Li, Z. Zhao et al., ‘On the a-Stable Distribution of Base Stations in Cellular Networks’, *IEEE Communications Letters*, **vol. 7798**, pp. 1-1, 2015.

[10] Jure Leskovec, ‘Social media analytics: tracking, modeling and predicting the flow of information through networks,’ pp. 277, March -April 2011.

[11] H. Kwak, C.Lee, H. Park, and S. Moon, ‘What is Twitter , a Social Network or a News Media?,’ *International World Wide Web Conference Com- mittee (IW3C2)*, pp. 1-10, 2010.

[12] T. Warren Liao, ‘Clustering of time series data - A survey’, *Pattern Recognition*, **vol. 38, no. 11**, pp. 1857-1874, 2005.

[13] M. Ester, H. Kriegel, J. Sander et al., ‘A density- based algorithm for discovering clusters in large spatial databases’, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226-231, 1996.

[14] M. Vlachos, J. Lin, E. Keogh, ‘A wavelet-based anytime algorithm for k-means clustering of time series’, *Proc. Workshop on Clustering*, pp. 23-30, 2003.

[15] Schmidhuber, J., ‘Deep learning in neural networks: An overview’, *Neural networks*, 61, pp.85-117, 2015.

[16] A. Ingolfsson, E. Sachs, ‘Recurrent neural network based language model’, *Journal of Quality Technology*, **vol. 25, no. 4**, pp. 271-287, 1993.

[17] W. Zaremba, I. Sutskever, O. Vinyals, ‘Recurrent Neural Network Regularization’, *Proc. Workshop on Clustering*, pp. 1-8, 2004.

[18] M. Mohri, A. Rostamizadeh, A. Talwalkar, ‘Foundations of Machine Learning,’ *The MIT Press*, pp. 7, 2012

[19] Z. Yu, D. Zhang, X. Zhou, and C. Li, ‘User preference learning for multimedia personalization in pervasive computing environment,’ *in Proc. Int. Conf. Knowl.-Based Intell. Inf. Eng. Syst.*, pp. 236-242, 2005.

[20] M. Chen, W. Saad, C. Yin, and M. Debbah, ‘Echo state networks for proactive caching in cloud-based radio access networks with mobile users,’ *IEEE Trans. Wireless Commun.*, vol. 16, no. 6, pp. 3520-3535, Jun. 2017.

[21] M. Chen, M. Mozaffari, W. Saad, C. Yin, M. Debbah, and C. S. Hong, ‘Caching in the sky: Proactive deployment of cache-enabled unmanned aerial vehicles for optimized quality-of-experience,’ *IEEE J. Sel. Areas Commun.*, vol. 35, no. 5, pp. 1046-1061, May 2017.

[22] G. Szabo and B. A. Huberman, 'Predicting the popularity of online content,' *Commun. ACM*, vol. 53, no. 8, p. 80, Aug. 2010.

[23] M. Ahmed, S. Spagna, F. Huici, and S. Niccolini, 'A peek into the future: Predicting the evolution of popularity in user generated content,' *in Proc. 6th ACM Int. Conf. Web Search Data Mining (WSDM)*, New York, NY, USA, pp. 607, 2013.

[24] T. Tanaka, S. Ata, and M. Murata, 'Analysis of popularity pattern of user generated contents and its application to content-aware networking,' *in Proc. IEEE Globecom Workshops (GC Wkshps)*, pp. 1-6, Dec. 2016.

[25] P. Jiang, J. Winkley, C. Zhao, el at.,'An Intelligent Information Forwarder for Healthcare Big Data Systems with Distributed Wearable Sensors,' *IEEE Syst.J*, pp. 1147-1159, 2016.

[26] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, 'A survey on sensor networks,' *IEEE Commun. Mag.*, pp. 102-114, 2002.

[27] F. M. Al-Turjman, H. S. Hassanein, M. A. Ibnkahla, 'Efficient deployment of wireless sensor networks targeting environment monitoring applications,' *Comput. Commun.*, vol. 36, pp. 135-148, 2013.

[28] J. Vilela, Z. Kashino, R. Ly, 'A Dynamic Approach to Sensor Network Deployment for Mobile-Target Detection in Unstructured, Expanding Search Areas.' *IEEE Sens. J.*, vol. 65, pp. 4405-4417, 2016.

[29] X. Denga, Y. Jianga, L. T. Yang, 'Data fusion based coverage optimization in heterogeneous sensor networks: A survey,' *Inf. Fusion*, vol. 52, pp. 90-105, Dec. 2019.

[30] J. Chen, A. Malhotra, I. D. Schizas, ‘Data-driven sensors clustering and filtering for communication efficient field reconstruction,’ *Signal Process.*, vol. 133, pp. 156-168, 2017.

[31] E. Sdshu, L. Irfxvhg, R. Wkh, el at.,‘Comparative analysis of big data management for social networking sites,’ *Computing for Sustainable Global Development (INDIACom), 2016 3rd International Conference on*, pp. 1196-1200, 2016.

[32] S. Almeida, J. Queijo, L. Correia, ‘Spatial and temporal traffic distribution models for GSM,’ *IEEE Communications Letters*, pp. 131-135, 1999.

[33] A. Byungchan, Y. Hyunsoo, C. Jung-Wan, ‘Joint deployment of macrocells and microcells over urban areas with spatially non-uniform traffic distributions,’ *IEEE VTS Fall VTC2000*, **vol. 6**, pp. 2634 - 2641, 2000.

[34] J. Xu, J. Wang, Y. Zhu, et al. ‘Cooperative distributed optimization for the hyperdense small cell deployment,’ *IEEE Communications Magazine*, **vol. 52, no. 5**, pp. 61-67, 2014.

[35] Y. Pak, K. Min, S. Choi, ‘Performance evaluation of various small-cell deployment scenarios in small-cell networks,’ *The 18th IEEE International Symposium on Consumer Electronics*, **vol. 52, no. 5**, pp. 1-2, 2014.

[36] P. Fiadino, M. Schiavone, P. Cases, ‘Vivisecting WhatsApp through large-scale measurements in mobile networks’, *Computer Communication Review*, **vol. 44, no. 4**, pp. 133-144, 2015.

[37] Z. Zhao, Z. Feng, Y. Zhang et al, ‘Collecting, managing and analyzing social networking data effectively’, *2015 12th International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2015*, pp. 1642-1646, 2016.

[38] A. Coates, A. Ng, ‘Learning feature representations with K-means’, *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, pp. 561-580, 2012.

[39] J. Sander, M. Ester, H. Kriegel et al, ‘Density-Based Clustering in Spatial Databases: The Algorithm GDBSCAN and its Applications’, *Springer*, pp. 1-30, 1998.

[40] R. Jozefowics, W. Zaremba, I. Sutskever et al, ‘An empirical exploration of Recurrent Network architectures’, *32nd International Conference on Machine Learning, ICML 2015*, **vol. 3**, pp. 2332-2340, 2015.

[41] F.A. Gers, J. Schmidhuber, F. Cummins, ‘Learning t o Forget : Continual Prediction with LSTM’. *1999 Ninth International Conference on Artificial Neural Networks I-CANN 99.*, pp. 850-855, 1999.

[42] F. Petitjean, A. Ketterlin, and P. Ganarski, ‘A global averaging method for dynamic time warping, with applications to clustering’, *Pattern Recognition*, **vol. 44, no. 3**, pp. 678-693, Mar. 2011.

[43] M. Okawa, ‘Template Matching Using Time-Series Averaging and DTW with Dependent Warping for Online Signature Verification’, *IEEE Access*, **vol. 7**, pp. 81010-81019, 2019.

[44] T. Das, S. Misra, S. CHoudhury et al., 'Comparison of DTW score and warping path for text dependent speaker verification system', *IEEE International Conference on Circuit, Power and Computing Technologies, ICCPCT 2015*, pp. 1-4, 2015.

[45] J. G. Andrews, 'Cellular 1000x?'. Notre Dame University Wireless Leadership Seminar: http://users.ece.utexas.edu/ jan- drews/publications.php, July 2011.

[46] V. Chandrasekhar, J. G. Andrews, and A. Gatherer, 'Femtocell networks: a survey', *IEEE Communications Magazine*, Sept. 2008.

[47] H. Cao, G. Wang, and Y. Xie, 'A two-dimensional logical coordinate system for hexagonal grids in manets', *2009 WRI International Conference on Communications and Mobile Computing*, **vol. 3**, pp.358-363, 2009.

[48] H. Elsawy, E. Hossain, and M. Haenggi, 'Stochastic geometry for modeling, analysis, and design of multi-tier and cognitive cellular wireless networks: A survey', *IEEE Communications Surveys and Tutorials*, **vol. 15, no. 3**, pp.996-1019, 2013.

[49] 'https://www.cellmapper.net/map'

[50] Y. Du, J. Fan, and J. Chen, 'Experimental analysis of user mobility pat- tern in mobile social networks', *2011 IEEEWireless Communications and Networking Conference*, pp. 1086-1090, 2011.

[51] Y. Ahn, S. Han, H. Kwaket al., 'Analysis of topological characteristics of huge online social networking services', *16th International World Wide Web Conference, WWW2007*, pp. 835-844, 2007.

[52] P.F. Velleman, D.C. Hoaglin, ‘Applications, Basics, and Computing of Exploratory Data Analysis’, *The Internet-First University Press*, *chapter 4*, pp. 93-120 , 2004.

[53] F. Pedregosa, G. Varoquaux, A. Gramfort et al., ‘Scikit-learn: Machine Learning in Python’, *Journal of Machine Learning Research*,**vol. 12**, pp. 2825-2830, 2012.

[54] N. Chopde, M. Nichat, ‘Landmark Based Shortest Path Detection by Using A* and Haversine Formula’, *International Journal of Innovative Research in Computer and Communication Engineering Vol.*,**vol. 1, no. 2**, pp. 298-302, 2013.

[55] X. Zhou, Z. Zhao, R. Li, Y. Zhou, J. Palicot, and H. Zhang, ‘Human mobility patterns in cellular networks’, *IEEE Communications Letters*, **vol. 17, no. 10**, pp. 1877-1880, 2013.

[56] B. Yang, W. Guo, B. Chen, G. Yang, and J. Zhang, ‘Estimating Mobile Traffic Demand Using Twitter’, *IEEE Wireless Communication Letters*, *vol. 5, no. 4*, pp. 380-383, 2016.

[57] M. Ester, H. Kriegel, J. Sander, X. Xu, ‘A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise’, *Proceedings of the 2nd International Conference on Knowledge Discovery and Data Mining*, pp. 226-231, 1996.

[58] Sarkar, T.K., Zhong Ji, Kyungjung Kim, Medouri, A., Salazar-Palma, M, ‘A survey of various propagation models for mobile communication’, *IEEE Antennas and Propagation Magazine*, **vol. 45, no. 3**, pp. 51-82, 2003.

[59] I. Poole, ‘Free Space Path Loss: Details, Formula, Calculator’, radio-electronics.com, Adrio Communications Ltd. Retrieved 17 July 2017.

[60] ‘Number of social media users worldwide 2010-2021 - Statista’. [Online]. Available: https://www.statista.com/statistics/278414/numberof-worldwide-social-network-users/. [Accessed: 13-Dec-2019].

[61] T. Qiu, J. Feng, Z. Ge, J. Wang, J. Xu, and J. Yates, ‘Listen to me if you can: tracking user experience of mobile network on social media,’ *10th ACM SIGCOMM conference*, 18, pp. 288-293, 2010.

[62] K. Takeshita, M. Yokota, and K. Nishimatsu, ‘Early network failure detection system by analyzing Twitter data’, *2015 IFIP/IEEE International Symposium*, 18, pp. 279-286, 2015.

[63] W. Guo and J. Zhang, ‘Uncovering wireless blackspots using Twitter data’, *Electron. Lett.*, 18, vol. 53, no. 12, pp. 814-816, June, 2017.

[64] G.P. Zhang, ‘Time series forecasting using a hybrid ARIMA and neural network mode’, *Neurocomputing*, 18,vol. 55, pp. 159-175, 2003.

[65] Q. T. Tran, H. Li, and Q. K., ‘Cellular Network Traffic Prediction Using Exponential Smoothing Methods’, *Journal of ICT*, 18, no. 1, pp.1-18, January, 2019.

[66] J. Wang, J. Tang, Z. Wu et al., ‘Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach’, *IEEE INFOCOM*, May, 2017.

[67] A. Nikravesh, S. Ajila, C. Lung el al., ‘Mobile network traffic prediction using MLP, MLPWD, and SVM’, *IEEE International Congress on Big Data*, pp. 402-409, June, 2016.

[68] C. Marquez, M. Gramaglia, M.fiore et al., ‘Not All Apps Are Created Equal: Analysis of Spatiotemporal Heterogeneity in Nationwide Mobile Service Usage’, *CoNEXT 17*, pp. 180-186, December, 2017.

[69] H. Zhang, N. Liu, X. Chu et al., ‘Network Slicing Based 5G and Future Mobile Networks: Mobility, Resource Management, and Challenges’, *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138-145, August, 2017.

[70] S. Hochreiter and J. Schmidhuber, ‘Long Short-Term Memory’, Neural Computation, Vol. 9, No. 8, 1997, pp. 1735-1780.

[71] M. SHafiq, L. Ji, A. Liu et al., ‘Characterizing and Optimizing Cellular Network Performance during Crowded Events’, *IEEE/ACM Transactions on Networking*, **vol. 24, no. 3**, pp. 1308-1321, 2016.

[72] F. Xu, Y. Li, H. Wang et al., ‘Understanding Mobile Traffic Patterns of Large Scale Cellular Towers in Urban Environment’, *IEEE/ACM Transactions on Networking*, **vol. 25, no. 2**, pp. 1147-1161, 2017.

[73] H. Wang, X. Chen, S. Qiang et al., ‘Early Warning of City-Scale Unusual Social Event on Public Transportation Smartcard Data’, *13th IEEE International Conference on Ubiquitous Intelligence and Computing*, pp. 188-195, 2017.

[74] J. Yang, ‘Travel time prediction using the GPS test vehicle and Kalman filtering techniques’, *Proceedings of the American Control Conference*, **vol. 3**, pp. 2128-2133, 2005.

[75] T. Au, R. Duan, H. Kim et al., ‘Spatiotemporal event detection in mobility network’, *Proceedings - IEEE International Conference on Data Mining, ICDM*, pp. 28-37, 2010.

[76] C. S., ‘Dynamic programming algorithm optimization for spoken word recognition, IEEE Transactions on Acoustics’, *Speech and Signal Processing*, **vol. 26, no. 1**, pp. 43, 1978.