VISUAL ANALYTICS OF MOVEMENT BEHAVIOR PATTERN BASED ON TIME-SPATIAL DATA: A NEURAL NET APPROACH

ZHENGHAO CHEN
SID: 430057654

Supervisor: Xiuying Wang
Associate Supervisor: Jianlong Zhou

This thesis is submitted in partial fulfilment of
the requirements for the degree of
Bachelor of Information Technology (Honours)

School of Information Technologies
The University of Sydney
Australia

June 2017

Student Plagiarism: Compliance Statement

I certify that:

I have read and understood the University of Sydney Student Plagiarism: Coursework Policy and Procedure;

I understand that failure to comply with the Student Plagiarism: Academic Board Policy: Academic Dishonesty and Plagiarism can lead to the University commencing proceedings against me for potential student misconduct under the 2012 Academic Dishonesty and Plagiarism in Coursework Policy.

This Work is substantially my own, and to the extent that any part of this Work is not my own I have indicated that it is not my own by Acknowledging the Source of that part or those parts of the Work.

Name: Zhenghao Chen

Signature: **Date: 13/06/2017**

Abstract

Time-Spatial data plays a crucial role for different fields such as traffic management and urban activities. These data can be collected via devices such as surveillance sensors or tracking systems. However, how to efficiently analyze and visualize these data to capture essential embedded pattern information is becoming a big challenge today. Classic visualization approaches focus on revealing 2D and 3D spatial information and modeling statistical test. Those methods would easily fail when data become massive. Recent attempts concern on how to simply cluster data and perform prediction with time-oriented information. However, those approaches could still be further enhanced as they also have limitations for handling massive clusters and labels. In this thesis, we propose a visualization methodology for mobility data using artificial neural net techniques. This method aggregates three main parts that are Back-end Data Model, Neural Net Algorithm including clustering method Self-Organizing Map (SOM) and prediction approach Recurrent Neural Net (RNN) for extracting the features and lastly a solid front-end that displays the results to users with an interactive system. The neural net SOM is able to cluster the visiting patterns and detect the abnormal pattern. RNN can perform the prediction for time series analysis using its dynamic architecture. Furthermore, an interactive system will enable user to interpret the result with graphics, animation and 3D model for a close-loop feedback. The contributions of this methods are: 1) it processes and prepares the data set in order to perform efficient further analysis in the next step. 2) The neural net algorithms are used to cluster and predict human behaviors for different purposes. 3) An interactive front-end user interface allows the user to analyse data efficiently in real-time. This method can be particularly applied in three tasks that Safety-oriented park management, Commercial-based Promotion and abnormal traffic patterns detection. During our work, we took participation in two public challenges that are VAST 2015 and VAST 2017, as well as provided suggestions to a mobile company for their promotion activity using our approach.

Acknowledgements

I would like to thank my supervisors Dr Xiuying Wang in the school of Information Technology in University of Sydney (USYD SIT), Australia and Dr Jianlong Zhou in Data61 Commonwealth Scientific and Industrial Research Organisation (CSIRO Data61), Australia for their kindness support and continues encourage. This project and thesis cannot be done without their helping.

This work is supported by School of Information Technology in University of Sydney (USYD SIT) and Data61 Commonwealth Scientific and Industrial Research Organisation (CSIRO Data61).

Data sets for this project are two public data sets and one confidential data set. Public data sets are from Visual Analytics Science and Technology Challenge 2017 (VAST Challenge 2017) as well as Visual Analytics Science and Technology Challenge 2015 (VAST Challenge 2015) and the confidential data set is provided by a Chinese mobile company.

List of Publications

Conference

[1] **Zhenghao Chen**, J. Zhou, X. Wang, J. Swanson, F. Chen, D. Feng **Neural Net-Based and Safety-Oriented Visual Analytics for Time-Spatial Data.** *IEEE International Joint Conference on Neural Networks 2017 (IJCNN 2017)*

[2] **Zhenghao Chen**, J. Zhou, X. Wang, F. Chen **Abnormal Trajectory Pattern Detection and Visualization using Neural Net.** *IEEE Visual Analytics Science and Technology 2017 (VAST 2017)* Submitted

Journal

[1] **Zhenghao Chen**, J. Zhou, X. Wang, F. Chen, D. Feng, H. Cui **Visual Analytics of Movement Behavior Pattern based on Time-Spatial Data: A Neural Net Approach** *IEEE Transactions on Human-Machine Systems* to be Submitted

Table of Contents

Abstract **iii**

Acknowledgements **iv**

List of Publications **v**

Table of Contents **vi**

List of Figures **ix**

List of Tables **xiv**

1 Introduction **1**

1.1 Problem and Challenges 1

1.2 Contribution 3

1.3 Thesis outline 5

2 Background **6**

2.1 Time Spatial Data Collection and Processing 6

2.2 Traditional System and Framework Overview 8

2.2.1 Database Methods 9

2.2.2 2D Method 9

2.2.3 3D Method 11

2.2.4 Statistical Method 13

2.3 Aggregation and Grouping 14

2.3.1 Grouping 14

2.3.2 Aggregation 16

2.4 Clustering 17

2.4.1 K-Means and Regression Model 17
2.4.2 Anomaly Detection 18
2.4.3 A Shallow Attempt of Self-Organizing Map Clustering 18
2.5 Prediction 19
2.5.1 Inference Model 19
2.5.2 Deep Learning 20

3 Data Description 22
3.1 Visual Analytics Science and Technology Challenge 2015 23
3.2 Mobility Data of Chinese Mobile Company 24
3.3 Visual Analytics Science and Technology Challenge 2017 25

4 A Static to Dynamic Data Model 27
4.1 Static Database 28
4.1.1 Key-Points and Tree Structure 28
4.1.2 Relationship 30
4.1.3 Index 32
4.1.3.1 Spatial-oriented Index 33
4.1.3.2 Object-oriented Index 34
4.2 Dynamic Database 35
4.2.1 Visiting-Frequency Matrix 35
4.2.2 Time-Spent Matrix 36
4.2.3 Sequence Vector 38
4.2.4 Time-Oriented Matrix 40
4.3 Method Summary 42

5 Self-Organizing Map Based Clustering 43
5.1 Clustering and Pattern Recognition 43
5.1.1 Initialization 45
5.1.2 Competition 46
5.1.3 Cooperation 46
5.1.4 Adaption 47
5.1.5 Group Detection 49
5.1.6 Visiting Pattern Through time 52
5.1.7 Dynamic Time Warping 55

5.2 Experiment Result Visualization 56
5.2.1 Crime Detection for VAST 2015 56
5.2.2 Promotion for Mobile User 62
5.2.3 Abnormal Visiting Pattern for VAST 2017 65
5.3 Method Summary 71

6 Deep Learning Based Prediction 72
6.1 Prediction and Visualization 72
6.2 Classification Methods 74
6.2.1 K-Nearest Neighborhoods 74
6.2.2 Adaptive Boosting 74
6.2.3 Naive Bayes 75
6.2.4 Decision Tree 76
6.2.5 Random Forest 77
6.2.6 Support Vector Machine 77
6.3 Recurrent Neural Net 79
6.3.1 Architecture 79
6.3.2 Long-Short Term Memory 80
6.4 Experiment and Evaluation 81
6.4.1 Travel Flow Prediction for Visual Analytics Science and Technology 2015 82
6.4.2 Station Usage Prediction for Moblie Data 86
6.4.3 Traffic Forecast for Visual Analytics Science and Technology 2017 89
6.5 Method Summary 93

7 Conclusion 94
7.1 Approach 94
7.2 Experiment 95
7.3 Future Work 96

Bibliography 97

List of Figures

1.1 Framework of Our Method 4

2.1 Traditional trajectory system has two essential parts back-end and front-end, users query the data from database and result will be displayed. . 8
2.2 A fail example of 2D display map, data is over massive so there is too much information overlapped, user cannot tell the essential information. 10
2.3 Single trajectory can be clearly revealed by 3D cube display. 12
2.4 Multiple trajectories revealed via 3D cube display is very massive and information cannot be clearly presented. 12
2.5 Hot spot statistic method, too much information showing the chart with massive data make is hard for analysis. 13
2.6 How those individual points formed in Flock, Convoy and Swarm pattern among time t_1, t_2 and t_3. 15
2.7 How those individual points formed in Gathering Pattern among time. 16
2.8 RNN Back-propagation through time [1]. 20

3.1 Map of DinoFun Park from VAST 2015. 23
3.2 Map of Boonsong Lekagul Nature Preserve from VAST 2017. 25

4.1 Architecture of data model, raw data is store in Relational Database statically and will be precessed to dynamic data set once user query. . 27
4.2 Tree structure of data, super domains composite sub-domains. 29
4.3 Relational Database represents moving object visits the stay point. . . 30
4.4 Relationship Data Visualization Example 1, different vehicles (large trunk (yellow), small trunk (yellow) and car (blue) visits the camping 1 (red). 31

4.5 Relationship Visualization Data Example 2, Users (red) use Stations (blue). 32
4.6 This query is relevant to spatial information which scans 104 rows. . . 33
4.7 Same Query to Figure 4.6 with Spatial Index only retrieves 38 rows. . 33
4.8 This query is relevant with object information which scans 2387 rows. 34
4.9 The same query to Figure 4.8 with spatial index only retrieves 50 rows. 34
4.10 Heat map of visiting pattern, the color and size indicates the density of time spent in each location. 37
4.11 Bar chart of visiting pattern indicates frequency of check-in for each category of attractions, y-axis shows the frequency that visitors check in for each category. 37
4.12 Sequence Vector Generation 38
4.13 Animation for showing traveling sequence of a user, red path indicates the sequence of traveling in 2D map. 39
4.14 Example 3D cube for visualizing trajectory using time-oriented vector. Z-axis is time information, X-axis and Y-axis are coordinator on the 2D map. 40

5.1 Initialization shows how to project 3 dimensional vector to a 2D grids 45
5.2 Neighborhood Circle (green), the center of circle is BMU 47
5.3 Example of SOM U-Matrix, each point indicates a individual visiting pattern. The distinct clusters can be seen. Color embeds the difference information to show some clusters are distinct from others. The value is normalized from 0 to 1 shown in the right side. 48
5.4 We select points from cluster 1 and a random bunch respectively for detecting groups. 49
5.5 3D view of random points, this is not a group as most the points surpass the threshold (red line) and their trajectories vary. 50
5.6 3D view of group 1, this is a group as all the points under the threshold (red line) and their trajectories are the same. 50
5.7 Similar patterns, path 1 (orange) and path 2 (blue) have the same trajectory but in different time period. So they are not in a group but similar. 51

5.8 Heat map example of group visiting pattern in VAST 2015, size and color indicate the density of check-in of visitors. 52
5.9 Bar chart example of group visiting pattern in VAST 2015, indicates the check-in frequency for each category. 53
5.10 Relationship visualization example of group visiting pattern in VAST 2015 represents visitor (blue) checks in attraction (red). 53
5.11 Animation for revealing group trajectory of an example in VAST 2015, black path on the map is real-time trajectory, the chart in the downside indicates speed change through time. 54
5.12 Example of dynamic time Warping Result for VAST 2015, y-axis and x-axis represent two different trajectories after dimension reduction. . 55
5.13 Clustering result of SOM in Sunday, there are six outstanding groups, and group 1 and 2 (red points) are considered as suspect groups. . . . 57
5.14 Visiting preferences of group 1 and group 2, group 1 spend quite long time visiting attraction around the crime location while group 2 spend abnormally long time in crime location. 58
5.15 Trajectory of group 1 and group 2, group 1 keeps traveling around crime location in period crime happen while in the same period, group 2 spent their entirely time in the crime place. 58
5.16 Communication of group 1 and group 2, the communication of group 2 turns to a high frequency during the crime time. 59
5.17 Communication between group 1 and group 2, these two groups keeps in touch of whole day, indicates that they know each other. 59
5.18 Visitor attendance at Attraction 32 Creighton Pavilion for 3 days Fri, Sat and Sun, y-axis is the number of visitors checks in, x axis is the time. 60
5.19 Visitor Time-spent at Attraction 32 Creighton Pavilion for 3 days Fri, Sat and Sun, y-axis is the number of visitors checks in, x axis is the time. There are two outliers (red circle) on Sun. 60
5.20 Animation for revealing the crime, there are 2 trajectories (blue and black) the chart shows the speed of them. 61
5.21 Clustering U-Matrix of Mobile Data in Station 1001 from mobile data in China, there are three distinct points in the up right, down right and down left. 62
5.22 Outstanding users (blue) use stations (red) to visit IP address (green). . 63

5.23 Regular users (blue) use stations (red) to visit IP address (green). . . . 63
5.24 Station usage of outstanding user. 64
5.25 Station usage Pattern of Regular User. 64
5.26 U-Matrix result of VAST 2017, there are four outstanding points. . . . 65
5.27 Heat map of outstanding visiting patterns. 66
5.28 Relationship of outstanding results represents different vehicles that are car (blue), big truck (big yellow), small truck (small yellow) checking in locations (red). 67
5.29 3D Trajectories of outstanding patterns. 68
5.30 3D trajectory of a regular pattern example. 68
5.31 3D cube of popular trajectories, they are highly similar but vary in different time periods. They all start from general-gate 1 and end with entrance 4 69
5.32 Heat map of popular patterns, they are annotated with the heat pots. . 69
5.33 Relationship of popular attraction visiting pattern represents trucks (yellow) check in the locations (red). 70

6.1 Process time-oriented matrix to training data. 73
6.2 Structure of decision tree. 76
6.3 Architecture of Long-short Term Memory [1] with input gate, output gate and forget gate. 80
6.4 The prediction evaluation for VAST 2015, y-axis is accuracy, x-axis is method. The accuracy values from left to right are: 39.4%, 19.2%, 38.1%, 28.2%, 38.5%, 39.2%, 37.9%, 46.0% (Recurrent Neural Net), 42.3% (Deep Neural Net) 82
6.5 The prediction accuracy evaluation among time for VAST 2015, x-axis is predicted time set divided by 2 hours, from 8:00 am, 10:00 am, 12:00 am, 14:00 pm, 16:00 pm, 18:00 pm, 20:00 pm and 22:00 pm. Y-axis is the accuracy. 83
6.6 The actual movements for VAST 2015 from 13:00 pm to 14:00 pm, Sat, 07 June 2014, arrows indicate movements in one hour. 84
6.7 The RNN prediction movements for VAST 2015 from 13:00 pm to 14:00 pm, Sat, 07 June 2014, arrows indicate predicted movements in one hour. 84

6.8 Other prediction for VAST 2015, at 14:00 pm Sat, 07 June 2014. . . . 85
6.9 Evaluation for mobile data, y-axis is the accuracy. The accuracy values from left to right are: 97.2%, 6%, 97.5%, 93.2%, 92.4%, 90.5%, 96.2%, 96.4%, 98% 86
6.10 Prediction Evaluation for mobile data through time, x-axis is divided by 50 minutes, starting from 10:00 am ending with 15:00 pm (10:00 am, 10:50 am, 11:40 am, 12:30 pm, 13:20 pm, 14:10 pm, 15:00 pm), y-axis is accuracy. 86
6.11 Actual usage of mobile station at 17:00:00 pm, 14-May-2015. This pattern represents user (blue) uses station (red) 87
6.12 Predicted usage of mobile station at 17:00:00 pm, 14-May-2015. This pattern represents user (blue) uses station (red). 88
6.13 Prediction evaluation for VAST 2017, y-axis is accuracy. The accuracy values from left to right are: 52.3%, 84.2%, 21.2%, 87.6&, 79.0% and 92.3% 89
6.14 Prediction accuracy through time for VAST 2017, x-axis is divided by 120 mins from 0:00 am to 10:00 am. They are 0:00 am, 2:00 am, 4:00 am, 6:00 am, 8:00 am and 10:00 am. Y-axis is accuracy. 90
6.15 The actual movement for VAST 2017 from 19:00 pm to 20:00 pm from 01 Jan 2016 to 29 May 2016, arrows indicate movement in 1 hour. . . 90
6.16 The RNN predicted movement for VAST 2017 from 19:00 pm to 20:00 pm from 01 Jan 2016 to 29 May 2016, arrows indicate movement in 1 hour. 91
6.17 Other predictions for VAST 2017 at 20:00 am. 92

List of Tables

2.1 Drawback of clustering methods K-means and EM. 18

4.1 Visit Frequency Matrix Example 35
4.2 Time Spent Matrix Example 36
4.3 Time Oriented Matrix Example 40

6.1 Evaluation Classifier Parameter 82

Chapter 1

Introduction

Time-spatial data consists of spatial and temporal contextual information, which is a complicated type of data set. Nowadays, with the development of technique, it is becoming more and more convenient to collect such data with detailed day and time information than it was ever before [2]. Taking mobility data as an example which represents the objects movement in real time with recorded the geographical information [3], this type of data can be collected using Global Positioning System (GPS), sensors or tracking systems. The information embed from those data such as behavior pattern can be also implied in urban development, traffic transportation, and big event management. Therefore the understanding of this particular part of data is essential.

1.1 Problem and Challenges

The time-spatial data structure is simple composites with many singular vectors. Specifically, each single vector is associated with dimensions presenting time and geographical and user identity information respectively [4]. However those single vectors would not able to represent useful information without appropriate pre-processings. Therefore, researchers have conducted huge amount of work in terms of time-spatial data processing and analysis. One of the most straightforward way is to construct a visualization framework or system [5]. A system that stored such data as back-end enabling users to inspect every single trajectory and movement which is straightforward yet powerful. It is especially useful when the data set is not massive. Such a system is also

able to display the information using advance graph and multimedia techniques such as 2D heat map, 3D time-cube trajectory, real-time animation and so on [6]. The main challenges include how to efficiently construct a model and a database in order to process time-spatial data together with other useful information to represent that visitors check-in attraction in a certain period of time. Another challenge is how to enhance the front-end visual display via using the graphics (e.g. heat map) multimedia (e.g. animation) and statistical tools (e.g. charts).

In addition, increasing amount of data collection, analysis of trajectories is becoming even more massive. Therefore, single mobility data analysis and visualization become un-meaningful. The state of the art research focus on aggregation, clustering and grouping in order to simplify time-spatial context information via reducing the size of data representation [3]. With those advance techniques, the visualization of time-series data becomes much more elegant than before. The biggest challenge for those research is to generate an efficient algorithm for aggregation, grouping and clustering.

The big breakthrough was conducted with machine learning. Classic clustering methods such as Gaussian Mixture Model (GMM) with Expectation Maximization (EM) and K-means serves as effective tool. However, clustering raw trajectory with multiple-dimension is not sufficient for larger time-spatial data set [3]. Recently, the attempts of using Neural Network Self-Organizing Map (SOM) [7] to cluster the trajectory data is done in the 2012 [8]. That research combines temporal-context information and other associated data such as social-economics background of users. Using such data, researchers tried to visualize the pattern of different groups of visitors and their potential to commit crime. In 2015, the T.Landersberger used spatial-simplification algorithm [3] to dynamically reduce the flow information after aggregation and clustering which is the latest approach to achieve the aggregation task so far. State-of-art method deep learning enables the prediction of time-spatial for next timestamp. Researchers start looking at how to predict the upcoming location of movement [9] and generate an inference or generative model. Recurrent Neural Net (RNN) [10] is one of the popular deep learning models which makes it easier to deal with vectors where each dimension has strong dependency from previous one using the recurrent dynamic method to. The new concern, in this stage, is how to enhance the prediction accuracy and efficiency [9]. Overall, the challenges can be summarized as:

- Data Model: How to construct an efficient data model to store and process the time-spatial information.

- Feature Extraction: What features of this category of data can be extracted for prediction, clustering? What methodologies can be used to enhance the result (e.g. probability of prediction).

- Visual Analytics: How to enhance the front-end data visualization to boost the data interpretation and visualization.

1.2 Contribution

To solve the problems and obtain meaningful results from those data, this thesis proposes a methodology that utilizes neural network together with other visualization approaches. The framework of our method is shown as Figure1.1. There are three majority parts. Firstly, this approach utilizes the relational database combined with dynamic non-relational database as back-end. Temporal database will be generated once user requests from the relational database. The second part is features extraction algorithm. We propose two approaches prediction and clustering. For clustering, SOM will be applied to cluster visiting pattern in order to group the visitors and find the abnormal groups that can be further suspected as crime or others. Moreover, the inspections will be put on such abnormal clusters. Regarding prediction, LSTM network is utilized to predict upcoming locations that travelers would like to go in next time-stamp. Eventually, the extracted information such as clustering and prediction result will be displayed using graphics and multi-media techniques including 3D time-spatial cube, 2D heat map, relational map and animation to users. Such displays will help users understand behavior pattern of moving objects better. With help of this method, we can better resolve time-spatial data analytics problem.

Using our approach, we participated in two public challenges that Visual Analytics Science and Technology Challenge 2015 and 2017 (VAST 2015 & VAST 2017) as well

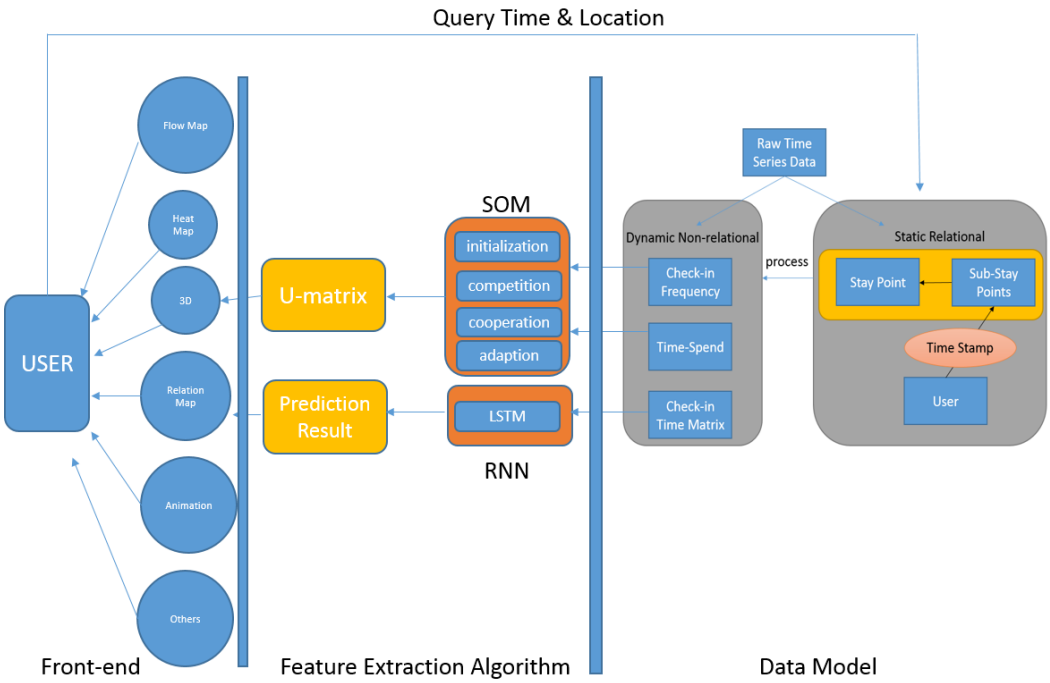

Figure 1.1: Framework of Our Method

as assist a mobile company in China for their promotion activity. First of all, for VAST 2015 , we conducted safety-oriented visualization for crime detection and movement flow prediction [2]. In addition, in terms of commercial based visualization such as mobile users promotion [2], we used this method to obtain user behavior pattern. Based on those information, we can provide our suggestions to the company manger. Lastly, we take traffic visualization and prediction in VAST 2017 to research how movement of vehicles affect wildlife via understanding their pattern. Consequently, we solve the crime detection problem, provide suggestions to a Chinese mobile company for user promotion and successfully predict the affect of vehicles to wildlife in a nature preserve. Overall contributions can be summarized as:

- We set up a time-spatial data visual analytics pipeline to understand movement behavior. This pipeline consists of a static-to-dynamic data model as back-end, feature extraction algorithms for generating behavior pattern information and visual analytical tools for displaying such information to users.

- We modify and utilize artificial neural net approach to propose a clustering

methodology based on SOM for visualizing group patterns and a prediction approach using LSTM to visualize upcoming locations that visitors will attend.

- We use this approach to resolve three data analytics problems. Firstly, we successfully detect unusual behavior patterns including crime patterns to help park manager target the crime groups. Moreover, we successfully find the outstanding patterns among million mobile users and predict the mobile usage patterns which will help the mobile company for commercial activity such as promotion. Lastly, we analyze the traffic movement in a wildlife Nature Preserve to study how those vehicles affect wildlife via predicting and clustering their patterns.

1.3 Thesis outline

This thesis proposes a visual analytics method mainly based on the Neural Network to deal with time-spatial data. It gives a brief overview of the problem statement and the current exiting issues in the introduction part (Chapter 1) to have clear indication of the both challenges and motivations of solving this problem. Literature Review (Chapter 2) reviews some traditional, breakthrough and state-of-art methodologies to this problem. From that we know the most of previous researchers concerned on the visual system as a resolution as well as some good attempts with machine learning approaches, for instance, state-of-art modelling technique deep learning. In addition, from data description part (Chapter 3), we get clear idea what data we will explore in the research. Moreover, Chapter 4-6 will talk about parts of framework we construct with three different methodologies that are data model (Chapter 4), SOM Clustering (Chapter 5) and RNN Prediction (Chapter 6). For data model part, our data model is from static relational model to dynamic model. The way we conduct this model is going to be explained. In clustering part, we will go through the entire SOM clustering algorithm with experiment results of three data set. In prediction part, the RNN model is given and such model will be used for prediction and visualization of flow map. Also, evaluation result with other classification methods will be shown. For last part conclusion, we will recap the framework and discuss the advantages of this method as well as evaluate this research by recalling the experiments.

Chapter 2

Background

This chapter is going to review the related work to this thesis. Time-spatial data problems was summarized as back-end, algorithm and front-end problem as we mentioned in Introduction. The researchers have developed the methods that concerned on:

- Construct visualization system to pipe data.
- Group, aggregation and cluster
- Predict the time-series data.

Hence, the literature review will be based on those points.

2.1 Time Spatial Data Collection and Processing

We begin by looking the data type of this project that is time-spatial data. Time-spatial data is subset of time series data that particularly records the geographical information oriented by time [11].

In my thesis, I will focus on mobility data (or movement data), This kind of data is normally adopted by location-ware devices such as sensor or Global Positioning System (GPS) and can be formatted in the vector $< visitorid, timestamp, x - position, y - position, other - information >$. Other information could contain such as: activity and so forth. For instance, the activity of visitors could be classified to check-in and moving in the trajectory data [5]. And it could be the other userid if it is a communication data, that stands for this user call to another user [12]. Some of those data can be

obtained via the mobile company and the information of those data usually include the time, geographical information plus the connection information between users and station. However as confidential issue, some information such as the talking content and geographical information will be filtered, the difficulty of visualization will be harder than regular time-spatial data [13] as less temporal information observed.

Regular processing can be summarized in the following ways [11]:

- **Noise Filtering**: Erase or re-generate the information like some missing geo-points and users id, the typical way to perform the job includes mean filter, Karlman filter and particle filter.
- **Stay Point**: If stay point is given by the original data set, research only need to summarize those points out from original time series data. Otherwise, the detection will be achieved using either Map API or detection algorithms [14].
- **User**: UserId is normally given by the original data, therefore users detection is a relatively simple step.
- **Compression**: Compression is to help the researcher to deal with inaccurate recording problem that it might be not setting data in precision way but would be accurate and helpful for researcher to perform other tasks.
- **Segmentation**: To perform tasks including classification, clustering or prediction, the segmentation would be an essential step. The segmentation is normally based on time interval via sampling, shape of trajectory and thirdly and semantic meanings.

Those pre-processing methods can help researchers to gain the new information for further database construction and other tasks. The classic new information matrix from pre-processing covers: user information, stay point information and trajectory information for single user. Traditional system also put efforts on detecting groups of visitors during the pre-processing stage [15]. But grouping users straightaway from raw data set is a time and space consuming behavior, so that is not being used in recently.

2.2 Traditional System and Framework Overview

Constructing system to index and retrieve the trajectory and time-spatial information is one of most popular method to manage the temporal context information [15] [11]. Using local or cloud system, user can manage data and display information. By that, users are able to interactive and extract the useful information from data by simply querying.

Classical systems of trajectory would store mobility data in a stored database indexed with visitors. In that way, users can easily query the information by userid and view the information through the visual analytics techniques including 2D, 3D plot and other methods.

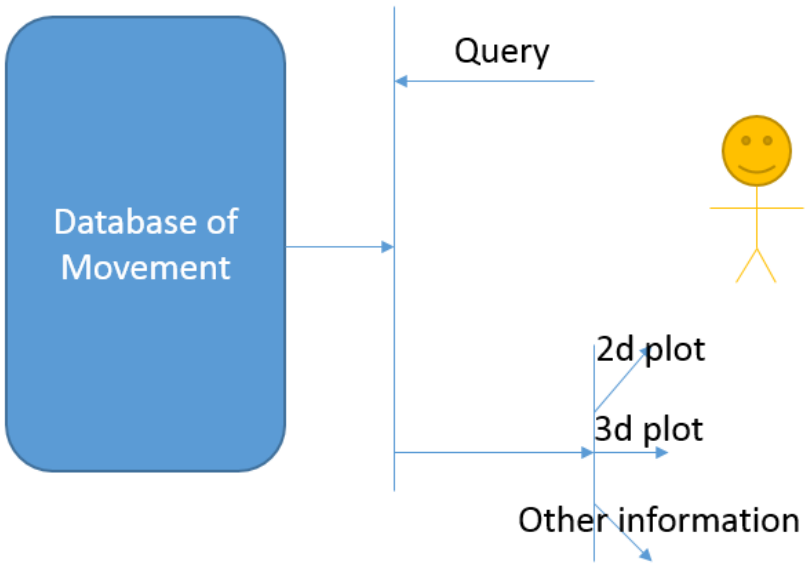

Figure 2.1: Traditional trajectory system has two essential parts back-end and front-end, users query the data from database and result will be displayed.

The typical query is individual query that inspect single trajectory via certain condition using userid, time or place. Another typical query is called group query that aggregates the group of people. Such query is normally generated based on trajectory similarity. Using a system is somehow an efficient way to manage. Therefore, this method is utilized all time.

2.2.1 Database Methods

Effective data management techniques are required in time-oriented visualization that for quickly retrieve data of trajectory. The most of database no matter it is relational or no-relational it would consists of 3 structures [16] [17]:

- **3D-RTree Based**: store the spatial range query in a 3D query box. It is often utilized for indexing trajectories generated in the near recent.
- **HR-Tree Based**: divides time period into several sub-interval using composite design pattern to embed the time slot to larger duration and index by the time which is efficient for retrieving the explicitly timestamp for movement.
- **Grid Index**: This approach segment the time into different 2D grid point with end time to start time as one dimension and the check-in spots as another dimensions.

Overall, constructing the efficient data structure and the tree structure is normally be considered in the first place in the traditional data model .

2.2.2 2D Method

2D method is based on map display [6], the information including the positions, directions and trajectories can be shown either statically or dynamically.

The most useful map display is spatial flow. Spatial indicates where the stay point that visitor start and the attraction that visitors stop in certain time duration which is also defined as flow map [3]. A typical spatial flow map is able to indicate the movement direction and how far the object move for individual or group. This method is useful for prediction or revealing the visiting sequence pattern. The main way to render such flow map can be in two ways which are spatial or temporal.

For one, temporal method uses time series of flow magnitudes to indicate movement pattern. Using animation for revealing the flow in real time is a typical technique to show the such visiting sequence pattern. Animation allows researchers to follow the

sequence of visitors by revealing the trajectory dynamically so that they can track the visit pattern in real time. Animation also can encode the information like change of movement speed, time that individual stay or travel around certain hot spot and so on. It is widely used in the industries [14]. The biggest drawback for animation is that animation takes lone time to produce. It results in researchers might be hard to memorize the information mentally when spatial situation varies. Therefore animation become inefficient when data set is for long series.

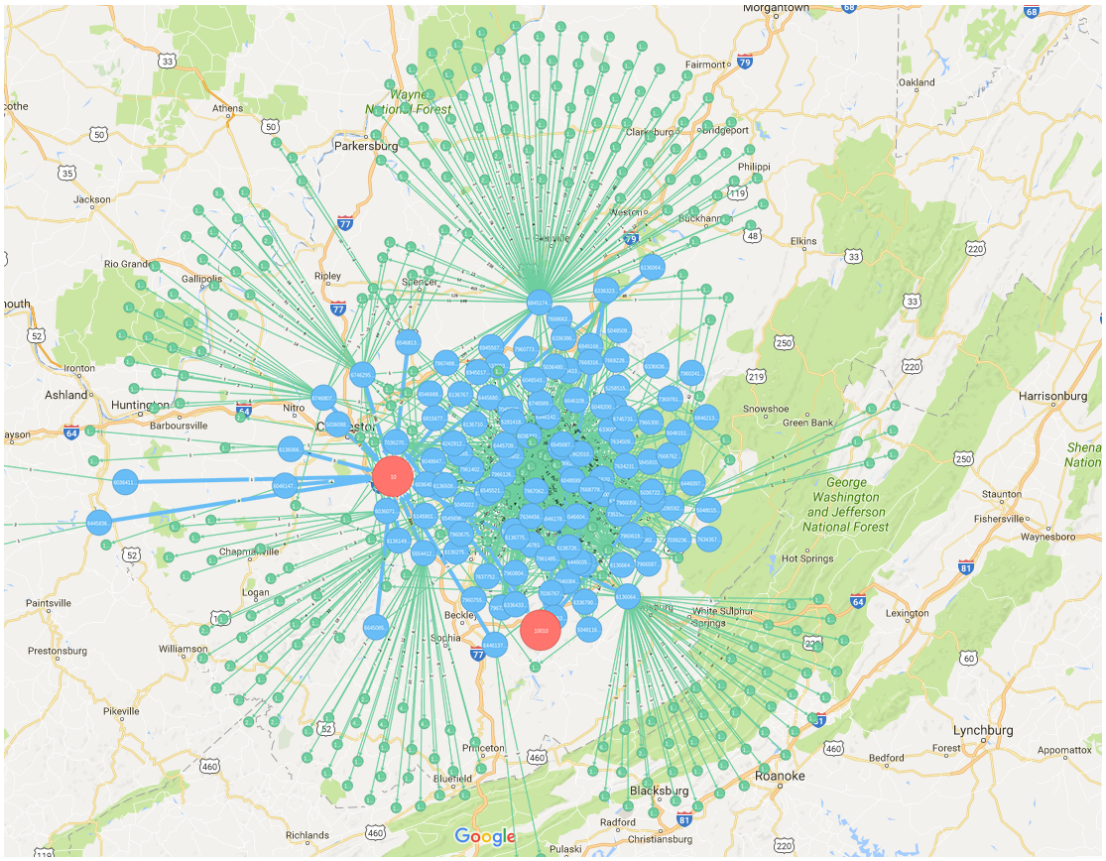

Figure 2.2: A fail example of 2D display map, data is over massive so there is too much information overlapped, user cannot tell the essential information.

For another, spatial presentation using multiple maps is to record the spatial relevant information like density that visitors visit or time that visitors spend in certain hot spot. Other essential presentation like visitors behaviors pattern could also be reflected by analysis from the spatial map for instance, the choice of visiting can be analyzed based on heat map [15]. It is widely used for trajectory pattern mining [11]. However, once

the number of moving objects growing up and the the map will shows over massive information in a map which cannot be analyzed.

Figure 2.2 shows a very typical example that directly display the information of spatial data without any filtering and extraction. This is one frame of animation for spatial map that display the visiting pattern for time series. Obvious with massive clusters of information, researcher will be not able to investigate the complex variation of the visiting pattern in some frames as the over comprehensive data display.

2.2.3 3D Method

As the limit of 2D play can only show the entire movement dynamically using animation, 3D time-space cube is therefore, used to show the whole trajectory statically via time-space bar display. In the cube, the horizontal dimension represents time while vertical one is used to arrange the bar to show the map. Whole trajectory can be segmented with a 3D view. Such temporal display shows the whole movement statically that researchers do not have to memorize the sequence like animation and it is more time-efficient than using animation [6]. To show the individual or a group movement, 3D cube is essentially useful as it can explicitly show the temporal spatial information by horizontal bar and vertical 2D map. However, similar to 2D method, 3D cube is still unable to handle the massive trajectory with many different individuals and groups.

Figure 2.3 shows that 3D cube can explicitly show the information that represents single trajectory. However when it comes to multiple entities of trajectories shows on 2.4, the information displayed becomes over comprehensive that cannot be understood by users.

Hence, those display methods including both 2D and 3D, they can be efficient for users to investigate and explore the visiting sequence based on time for individual user or group, but obviously fail the task for analyze the multiple entities trajectories.

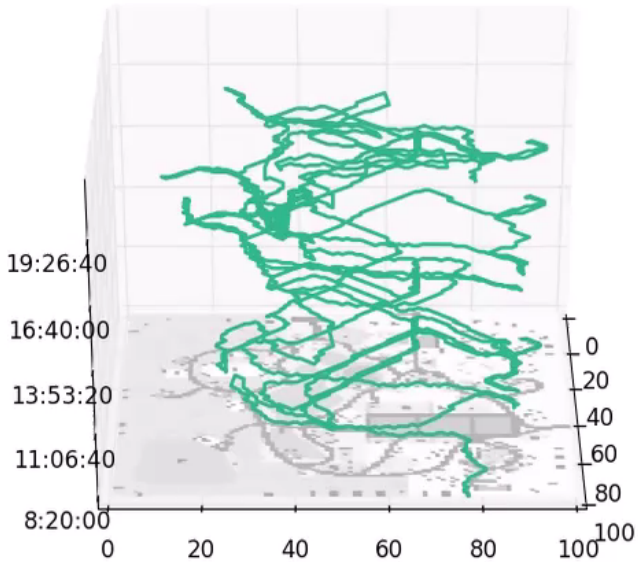

Figure 2.3: Single trajectory can be clearly revealed by 3D cube display.

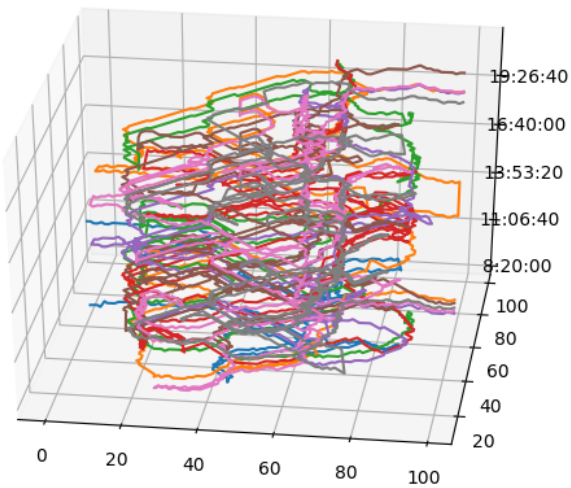

Figure 2.4: Multiple trajectories revealed via 3D cube display is very massive and information cannot be clearly presented.

2.2.4 Statistical Method

The scan statistic is another method to extract the information from the movement data. The typical statistic is for the hot spot [18] [19] [20]. Based on the previous work (Kulldorff et al. 1998), the statistic techniques can be summaries as Poisson And Bernoullet Models and Likelihood model test. Those method are able to detect the movement pattern by building an inference model.

A significant method is called hot spot statistic. It utilizes a slicing windows to analyze the hot spot one by one getting the information like usage of that spot, check-in or pass by frequency and so forth. By analyze that, users would have idea of situation of those hot spots.

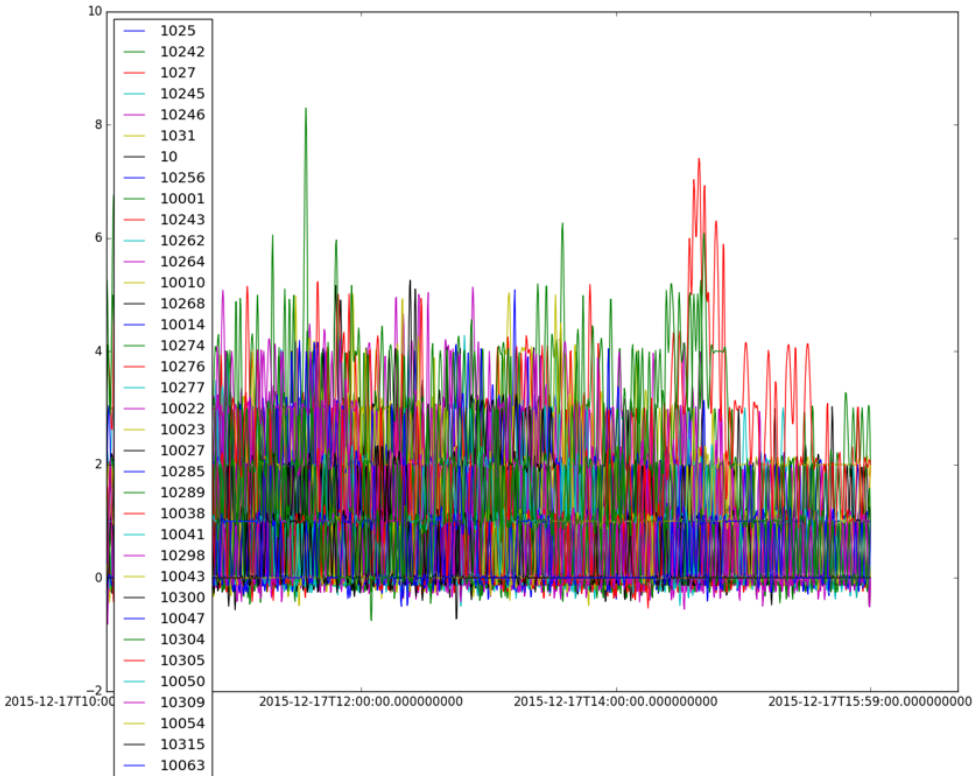

Figure 2.5: Hot spot statistic method, too much information showing the chart with massive data make is hard for analysis.

Figure 2.5 shows that the usage of the station in our work, the line chart illustrates

the temporal popularity of each station for certain time stamp. but this method cannot efficiently handle large data like previous approaches

2.3 Aggregation and Grouping

From last section, we clearly see that the traditional system has the identical problem that cannot handle the larger data or analyze them in the efficient way. Therefore, the attempts of trajectory data reduction and simplification is necessary and being the one of hottest topic in the recent trajectory data analysis. Aggregation and grouping are the most efficient ways to address the large amount of movement data by reducing the number of trajectories.

2.3.1 Grouping

Group in trajectory data study is often defines that the moving together pattern [11]. Specifically, when all objects are moving together as a single unit, we define those objects as group therefore the number of trajectory will be decreased from N the number of objects to 1.

The grouping definition that researchers come up with is that group of object that move together for a certain time period [5]. The traditional categories of group can be summarized as follow:

- **Flock**: A group of objects travel together with a disk of some fixed size for N continuing time stamp. The shape of the flock is normally main problem that must be defined in this problem.
- **Convoy**: A method to capture generic trajectory pattern for different size of moving objects group. An essential parameter of convoy is density that connected with N different moving objects.
- **Swarm**: Swarm is defined as a cluster of objects that move together without any time constraints.

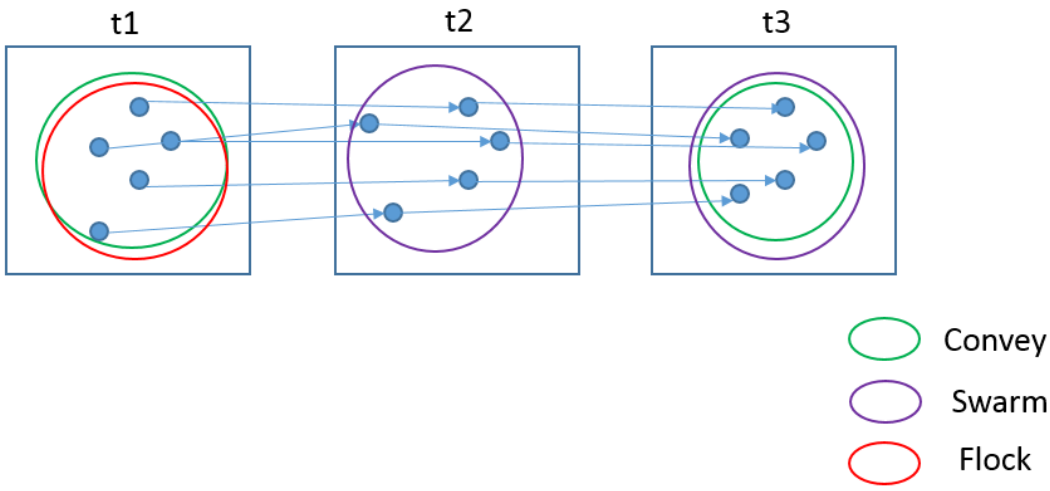

Figure 2.6: How those individual points formed in Flock, Convoy and Swarm pattern among time t_1, t_2 and t_3.

As shown as Figure 2.6, those five objects are defined as flock when timestamp t_1, as the density decrease in t_2, these five objects cannot be flock therefore even though in $t3$, the density increase again they cannot be defined as flock anymore as t_1 and t_3 are not the consecutive time period. However, we can still define that they are convey in t_1 and t_3 as convey does not require fixed shape and they are swarm in t_1, t_2 and t_3 because swarm has not time constraint.

A recently method called **Gathering** pattern is used for objects that join as in-regular group that, those bunch of objects are not necessarily need to be in fixed group in all time. Instead, they can be clustered into different group and reunion as new group with time period changing.

As shown in Figure 2.7 the group is clustered to two groups and reunion with the change of density with timestamp.

To calculate the density function, the Euclidean Distance is one of the most population way that utilized to compute the spatial distance of trajectory among different moving objects.

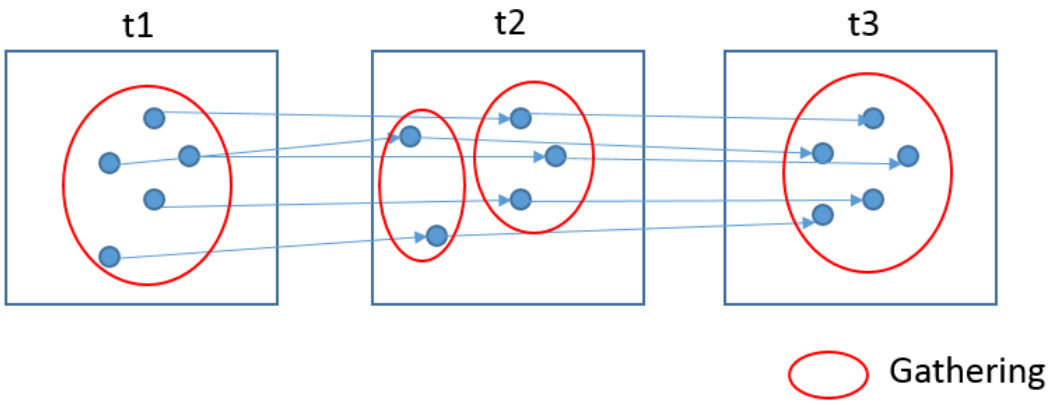

Figure 2.7: How those individual points formed in Gathering Pattern among time.

2.3.2 Aggregation

The recent aggregation work is concentrating on how to simplify and reduce the overlap of flow map on the 2D method [21]. One way of doing is to introduce another dimension of display or using multiple movement maps to represent certain time interval movement.

The work done by N, Andrienko and G, Andrienko points out that there are two specific solutions that can achieve data aggregation. For one, the certain users are grouped by same qualifications, such like moving speeds, path length, time period and so forth. Therefore, moving objects are gathered with same dimensional value. For another, for the movement from place $< p1, p2, p3, p4 >$ for object o_1, $< p1, p2, p6, p4 >$ for object O_2 the travel path is aggregated to be $< p1, p4 >$ for group $< o_1, o_2 >$. Both of method can significantly reduce the statistical computation and the overlapping of flow map that are extremely helpful for simplification of 2D map based method.

Except that, the attributes aggregation is also considered as an important method that the attributes of movements can be reduced from longer time to shorter time. For instance, the movement of whole day can which is divided by hour can be re-distributed by aggregated 2 or 3 hours to reduce the workload of statistics.

Overall, different from group, aggregation is more concentrated on the simplifying movement rather than simply gather moving objects like group.

2.4 Clustering

With the help of machine learning, the clustering techniques are become fairly essential for mining trajectory data [5]. Important tasks including unsupervised classification of patterns, pattern analysis, grouping the abnormal users are all based on an excellent clustering result. Therefore, to get partition of the unlabeled sequence data, different clustering algorithms are being attempted.

2.4.1 K-Means and Regression Model

To perform clustering, two types of traditional methods are often used. K-means, for one, is widely being used for most of clustering case [14, 5, 8], which is approved as an efficient way to perform such task. Using K-means, we able to cluster the moving objects into K groups. However, there are two majors trade-off of K-means. For one, K-means is extremely sensitive to noise [2], for another, defining K is another huge workload for researchers, that can cause another expensive research. Generally, to perform the relative small data set, K-means can be a convention way, but it often fail for using as massive and irregular data set as massive noise and uncertainty K-value.

Another method is Expectation Maximization (EM) which is used for Gaussian Mixture Model (GMM), Gaffney [22] proposed a method that utilize the regression mixture model to cluster the short sequence with entire trajectory. This method computes the partition of data based on probabilistic modeling. The grouping process of trajectory sets is optimized using EM algorithm that based on Maximum Likelihood principle. In this method, we can find out the cluster with mean and variance which help to mine the pattern similarity. However, this method still fail to address the massive amount of data for high computational complex as well as uncertainty number of clusters for initialization. The Table 2.1 reflects the major drawbacks for both methods.

K-means	EM
K-value Uncertainty	High Computational Complex
Sensitive to Noise	Uncertainty number of Cluster

Table 2.1: Drawback of clustering methods K-means and EM.

2.4.2 Anomaly Detection

Using Cluster to perform anomaly detection is also being highlight for sequence data mining. The definition of outliers in time series data is the moving object has the significant different pattern from others [11]. Such patterns include sequence pattern, visiting pattern and other pattern.

In the most work done by researchers, clustering the trajectories and find the individual sequence that does not belong to any clusters is the conventional way to detect the outlier. The reasons that outliers have distinct pattern is also interesting research topic [5, 3], crime detection, for instance, is an application of those researches.

2.4.3 A Shallow Attempt of Self-Organizing Map Clustering

The attempt of neural net method Self-Organizing Map (SOM) has been used in recently for clustering the trajectory data [8, 18]. SOM is an upsupervised learning method that can cluster similar vector into same group without pre-defining the number of clusters. The significance of SOM is that the data will be in better partition and less sensitive to the noise than K-Means and GMM.

In the work done by S.K Shukla [8], SOM is utilized for producing topology map result. In this research, researchers mainly focus on study entire clustering result rather than extracting other information like pattern recognition. Also, such topology result is not applied in any more specific way.

Another work done by J Hagenauer [22] particularly use SOM for finding the crime pattern using the spatial and temporal data combined with socio-economic and environmental context. In that research, the Hot Spot Statistic and SOM are utilized together for getting the topology map from SOM and find out distinct cluster as unusual pattern which is suspected to be crime. However in that experiment, for one thing, the data also includes the social-economical data. Though in our research, only pure trajectory will be provided and utilized. For another, there is not any deep study with such pattern. While deeper understand those pattern, more analysis and visualization should be required.

Generally, Self-organizing Map could be used as an efficient way to cluster data, besides only rendering the heat map, the other attempts of visualization should also be applied in those outstanding clusters on the result map.

2.5 Prediction

Movement prediction is also a huge research topic. The application of that includes forecasting terrorist attack, predicting the traffic jam and so on [9].

2.5.1 Inference Model

Building an inference model method is one of significant methods. There are particularly three types of model that being used:

- **Matrix Factorization (MF) Model**: This model factorize an user-item rating matrix into two low rank matrices. [9] One of significant method called Tensor Factorization (TF) that make time bins as another dimension and generate latent vectors of users. Generally, TF is an efficient model for inference the spatial prediction. The only issue is that if time bin factor is rarely used, then that sort of latent model would be hard to generate.

- **Markov Chain**: Predict the next sequence using exsiting sequence is also commonly used. Markov Chain, for instance, is used for sequence generation often applied in time series data like natural language processing (NPL). Factorizing Personalized Markov Chain (FPMC) using Transition Matrix to model temporal informatio is the most significant approach among those.

- **Other Machine Learning Classification**: Except those, the classic machine learning classification algorithms are also being used in prediction job. K-Nearest Neighbourhood (KNN), Support Vector Machine (SVM), Naive Bayes (NB), Decision Tree (DT) and Random Forest (RF) are the typical algorithms that have relative simple computational complex and high efficiency with those data.

Generally, comential time-spatial prediction is fully based on the machine learning technique that building an inference model for factorized sequence matrix and predict the value for next timestamp.

2.5.2 Deep Learning

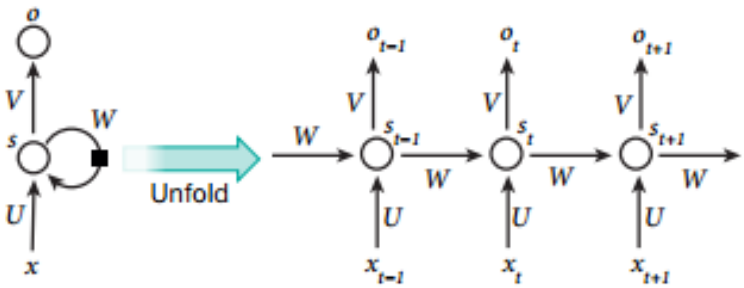

Figure 2.8: RNN Back-propagation through time [1].

Neural Net achieves much success and serves as an effective methodology to address with large and massive data set recently. For the data that formed in sequential inputs, Recurrent Neural Net (RNN) has been approved to be a good way to perform prediction task [1]. With Powerful dynamic system that back-propagation through time,

RNN is able to train the past sequence to predict the final state of the sequence vector.

Figure 2.8 represent the the Back-Propagation Through Time that RNN training, this dynamic training strategy is extremely helpful for time series data set.

In the work of J. Zhang [23], the Deep Spatial Residual network is utilized and modified to be ST-ResNet. Residual learning [24] allows the deep structure that more than 100 layers using the formula:

$$X_{l+1} = X_l + FX_(l) \tag{2.1}$$

Where x_l and x_{l+1} are input and output of the lth residual unit. F stands of residual processing function. In that research, residual network is taken as three different components that are closeness, period and trend. By extracting the features from data set following by forward activate into a two-layer neural net, then dynamically back propagated, the output of prediction will be generated. This method is particularly useful for the large city traffic network prediction. Moreover, abnormal movement features will be extracted using Convolution Net and train using residual net that generate the feature map to have a good prediction result.

Another outstanding work done by Q liu [9] was to build a recurrent architecture with spatial and temporal model (ST-RNN). It points out the limit of convolutional RNN for time interval and geographical information. In this method, researchers incorporate the distance-specific transition matrices that might have influence on human choice behavior to the model. Specifically weight of stay points is highlighted, for instance, the person went to the Great Wall last day and visiting other attractions nearby. Then the Great Wall might have higher importance weight than other temporal context. And such information will be recorded in the transition matrix. In addition, that method used linear interpolation for learning the continues value and provided a solution for modeling the prediction result.

Overall, either ST-ResNet or ST-RNN can be regarded as state-of-art methods in movement prediction with adjusting the ResNet and RNN into the architecture that suits for temporal-spatial data.

Chapter 3

Data Description

This section will introduce the data set I use for this research. There are three different data sets. Firstly, it is movement data from Visual Analytics Science and Technology Challenge 2015 (VAST 2015[1]) that is trajectory data representing visitors move around in a park. Secondly, we have mobility data from a Chinese mobile company that is user visit IP address using mobile device recorded by each mobile station. Finally, the last one is the Visual Analytics Science and Technology Challenge 2017 (VAST 2017[2]) which is the different vehicles traveling around in a nature preserve.

All these three data is formed in very typical temporal-spatial data type that is $< ID, starttime, endtime, x, y, otherinformation >$ vector for each set. Moreover, they also provide associated data for analysis. For instance, VAST 2015 data provides the stay point information, map and communication data of user. The VAST 2017 provides the nature preseve map which can help to detect the stay point.

In addition, each data set have certain tasks for visual analytics. In terms of VAST 2015, the problem is to find out factors that affect the park management and suggest the ideas that enhance safety of park. Mobility data analysis will concern on the commercial promotion based on users and stations, that is which user or station we should put the advertisements. As for the last one VAST 2017, the task is to see which vehicle will probably affect the wildlife of that nature preserve.

[1] `http://vacommunity.org/VAST+Challenge+2015`
[2] `http://vacommunity.org/VAST+Challenge+2017`

3.1 Visual Analytics Science and Technology Challenge 2015

In this challenge, data set is on movement of visitors in a modest-size amusement park, DinoFun World, sitting around 215 hectares with 81 different attractions that can be classified by 7 main categories with Rides, Show & Entertainment, Information & Assistant, shopping, Beer Gardens, Restrooms and Food, where Rides can be further distinguished by Thrill Rides Kiddie Rides and Rides for everyone. All attractions are numbered, named and connected by a visitor pathway throughout the park [25]. Each visitor was tracked by a mobile device that records his/her positions in real time and his/her behaviors and communications. For protecting the privacy of users, devices only recorded two behaviors of customers that movement and check-in [26]. This park hosts thousands of visitor every day.

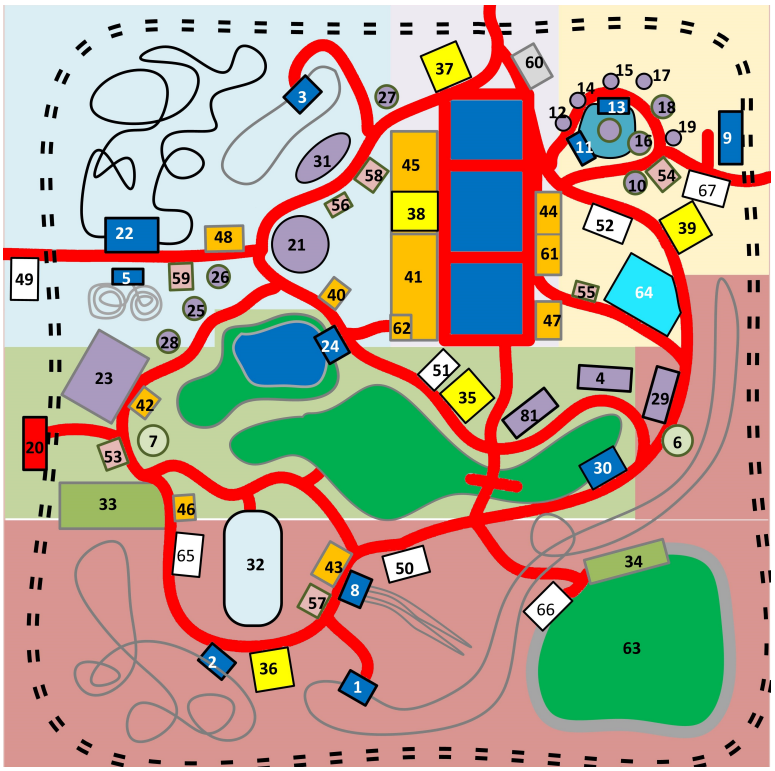

Figure 3.1: Map of DinoFun Park from VAST 2015.

As Shown in Figure 3.1, red lines illustrate the pathway. Attractions 1-8 and 81 are

Thrill Rides, attractions 9 -11 are Kiddie Rides, attractions 20-31 are Rides for everyone. Also, attraction 32, 61, 63 and 64 are Show & Entertainment, 60 and 62 are Information & Assistance, 40- 48 are all shopping place as well as 49-52 and 65-67 are restrooms. Food courts are located in attractions 35-39 and 53-59.

The movement data consists of two main parts: 25 million individual movement data (Mini-Challenge 1 (MC1)) and 4 million communication data (Mini-Challenge 2 (MC2)). Specifically, movement data is in the format of $< timestamp, id, behavior, position >$. Timestamp is the time that recorded by device, id is the person id that customer assigned, behavior is the whether the traveler check-in, check-out or pass by. position consists of geographically x and y data for visitors. Communication data is in the format of $< Timestamp, callfrom, callto, location >$. The only difference from communication data to movement data is that there two ids in this data set that one for representing the user make the call, another one is for user that receive the call.

The data is given for three days. There were 3357, 6411, and 7569 visitors on 6, 7, and 8 June 2014 respectively as an event Scott Jones Weekend was held for celebrating the coming of local star Scott Jones. On Sunday, 8 June 2014, a crime happened in this park and rapidly solved by officials [26]. The challenge is for trying to reveal how the crime happen and suggesting how to enhance the security of this park.

3.2 Mobility Data of Chinese Mobile Company

The second data set is form a Chinese mobile company. It records the users use the mobile phone stations to visit the IP address in the real time. There are more than 100,000,000 users in a day that are activated and call the mobile station to visit the IP address. For confidential agreement, we cannot show the geographical information of this data set.

The data set is formed in $< userId, TimeStamp, Station, positon, IP >$. Specifically, userId is the masked identification of users, timestamp indicates the time that user visit the IP, station and IP are mobile station and IP address.

3.3 Visual Analytics Science and Technology Challenge 2017

In this challenge, the trajectory data is provided from a Nature Preserve Boonsong Lekagul. Roadways are indicated by white colored pixels on the map Figure 3.2. Locations of this preserve can be divided into five categories. Each location type has its own unique color: Entrances (green), General-gates (blue), Gates (red), Ranger-stops (yellow), and Camping (orange). All location is associated with its own sensor to record the real-time data.

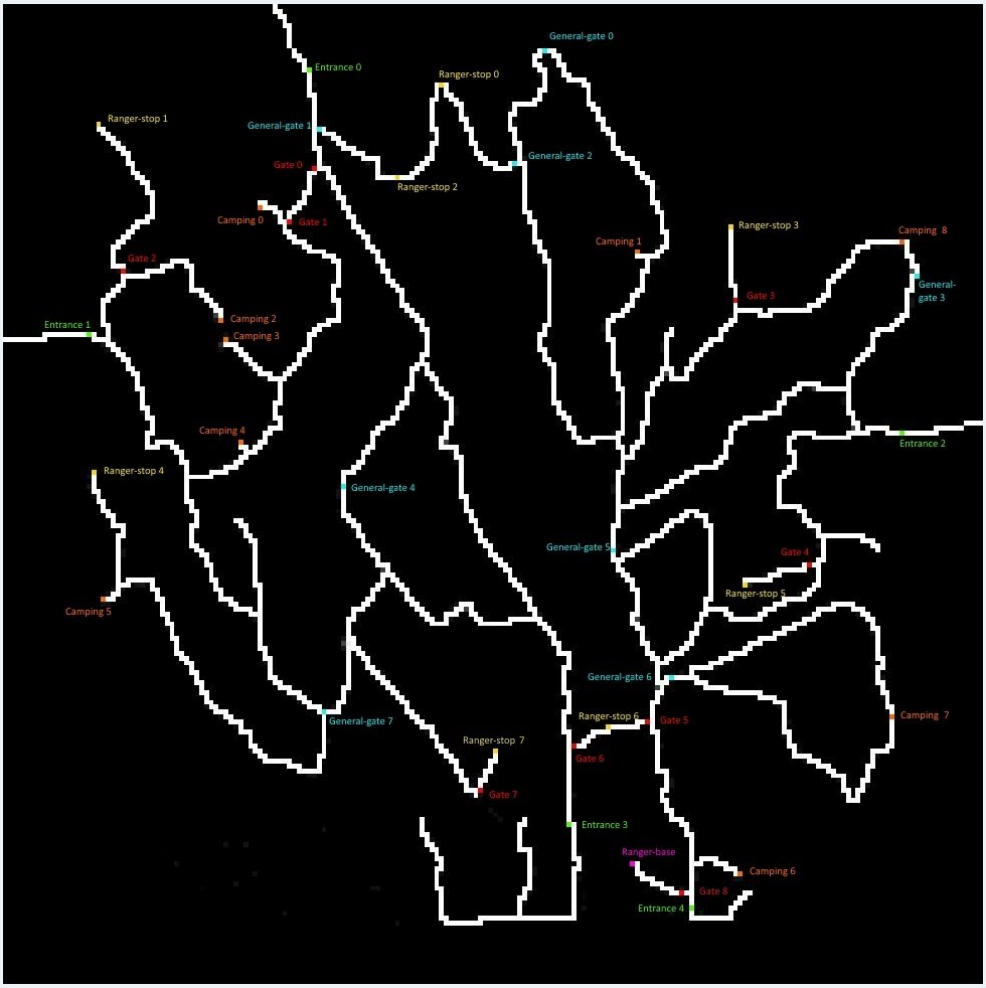

Figure 3.2: Map of Boonsong Lekagul Nature Preserve from VAST 2017.

- **Entrances:** All vehicles pass through an Entrance when entering or leaving the

Preserve.

- **General-gates:** All vehicles may pass through these gates. These sensors provide valuable information for the Preserve Rangers trying to understand the flow of traffic through the Preserve.
- **Gate:** These are gates that prevent general traffic from passing. Preserve Ranger vehicles have tags that allow them to pass through these gates to inspect or perform work on the roadway beyond.
- **Ranger-stops:** These sensors represent working areas for the Rangers, so you will often see a Ranger-stop sensor at the end of a road managed by a Gate. Some Ranger-stops are in other locations however, so these sensors record all traffic passing by.
- **Camping:**. These sensors record visitors to the Preserve camping areas. Visitors pass by these entering and exiting a campground.

The visitors in this data set are vehicles. There are six types of vehicles that are 2-axle car (or motorcycle), 2-axle truck, 3-axle truck, 4-axle (and above) truck, 2-axle bus and 3-axle bus.

Data can be formatted as $< Timestamp, car-id, car-type, gate-name >$. Specifically, Timestamp is the date and time the sensor reading was taken. Car-id is the assigned car ID from the entry gate. Car-type is Vehicle type as enumerated above which "P" is appended when it is a park vehicle. And Gate-name is name of the sensors taking the reading.

The motivation of this data visual analytics is to find out the abnormal trajectory patterns that might have affected the wildlife in this nature preserve.

Chapter 4

A Static to Dynamic Data Model

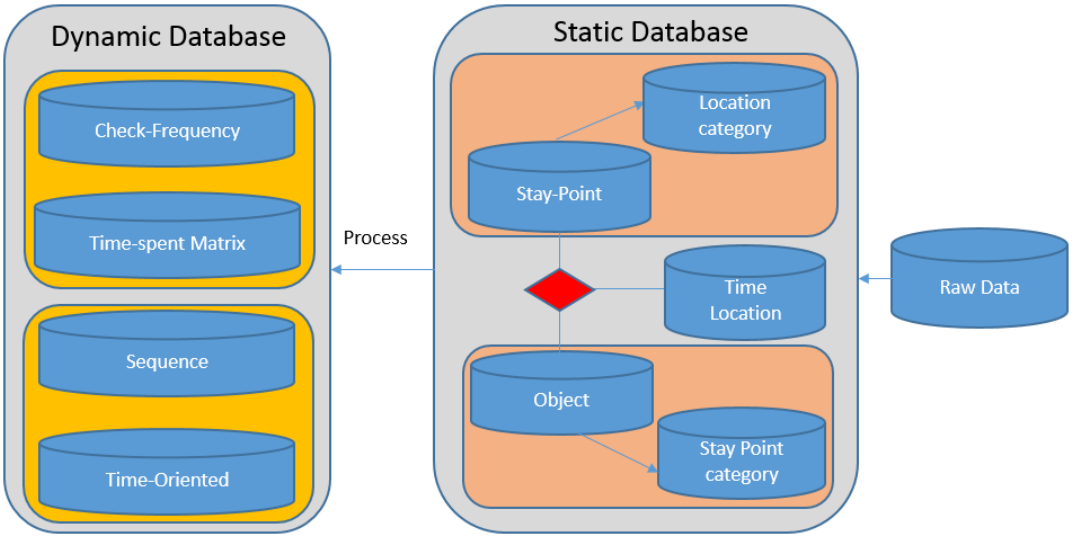

Figure 4.1: Architecture of data model, raw data is store in Relational Database statically and will be precessed to dynamic data set once user query.

This chapter will introduce the techniques that I use to build up the data model for time-spatial information. The architecture of the model can be summarized in Figure 4.1. There are two main types of data. For one, we have that relational database as static database. That is all the data will be stored in the relational database model using relationship $Object - [Activity] - > StayPoint$. Also, both stay points and objects are stored using tree data structure. Specifically, stay point will have sub-stay points as generalization. Moreover, non-relational dynamic database will be generated from static database once user requests. Those data will consist of check-in frequency,

time-spend, squence and Time-Oriented matrix.

4.1 Static Database

In this section, I will introduce the static database method I use for the whole approach. The static database is the model that permanently stores the time-spatial data as back-end. To construct this model, Relational Database Model (RDBM) will be utilized. I will firstly give a detail of data structure representing hierarchy of data, then explain the relationship between each domain that I have set up. Also, index techniques will be utilized to enhance the performance of database.

4.1.1 Key-Points and Tree Structure

The original temporal spatial data can be formatted as $< Object - ID, X, Y, Other - Information >$. Before storing data, following key-points are necessarily to process:

- **Moving Object**: To construct the domain of moving object, we need to extract the object information from raw data. Those moving object is going to be sorted with their id.

- **Stay-in Location**: Stay-in locations will be mapped from continues geographical Coordinates X and Y in the raw data to discrete label of each attraction. Euclidean Distance will be utilized for calculating the distance as well as we will set a threshold. Specifically, for every temporal coordinator $K(X_K, Y_K)$, we find a corresponding location $L(X_L, Y_L)$ with minimal distance to it. if the distance of coordinator K and attraction L is less than threshold, we can define that the object is currently checking in the location L. Following Algorithm will give a detail that how we map each key location.

Algorithm 1 Location Mapping Algorithm

$X_t, Y_t \leftarrow$ Temporal Location of Object
$L(X_L, Y_L) \leftarrow$ Location of visiting Key Points
$T \leftarrow$ Threshold
for $L(X_L, Y_L)$ **in** all Locations L **do**
  $Distance = Argmin \, \|(X_t - X_L)^2 - (Y_t - Y_L)^2\|_2$
  if Distance $< T$ **then**
    Moving Object is visiting location L in time t
  end if
end for

By above processing, we will have two essential domains Stay Point and Moving Objects. In most cases, either moving objects or stay points will have certain hierarchy that one super category contains couple of sub-categories. If the moving object, for example, is a car, then the super category of this database can be the vehicle and the sub-categories can be bus, train, car and so forth. Similar as the location, the large stations, for instance, can contain small stations.

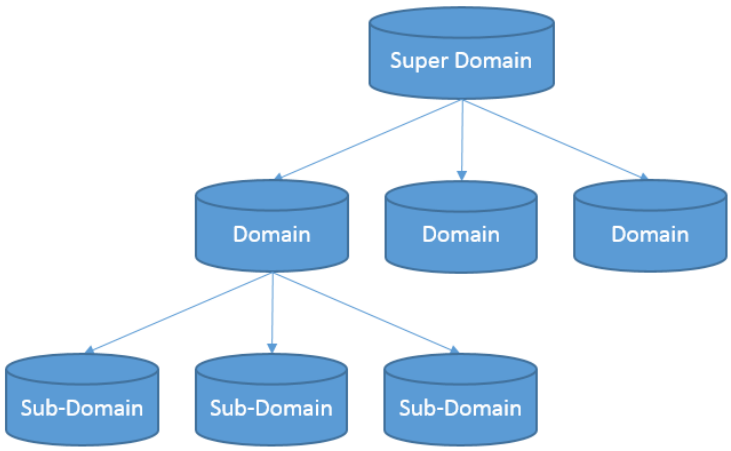

Figure 4.2: Tree structure of data, super domains composite sub-domains.

As shown in the Figure 4.2, the Tree Structure is that the super category composite with the sub-categories. Using this structure, data can be stored in hierarchy. Specifically, the position information and moving object information will be rooted by some

general information such as category.

4.1.2 Relationship

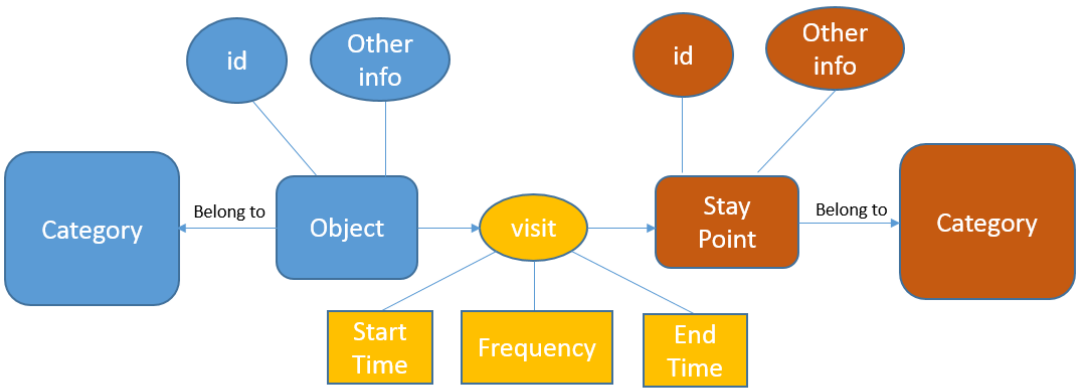

Figure 4.3: Relational Database represents moving object visits the stay point.

In this approach, the Relational Database Model (RDBM) is used for main static model. The main relationship can be summarized as Figure 4.3. Relationship between two main domains is visiting that records the start time, end time and frequency. Fro this relationship, we can know how many check-in activities for a moving objects to a location in certian time period. As mention previously, I store the locations and objects domain using tree structure. So that we are able to visualize the how many different categories of moving objects checking in the certain spatial domain.

Visualization of such data model can be explained from the following example. In the first one, we use data from Visual Analytics Science and Technology (VAST) Challenge 2017, this data represents different vehicles visit the attraction in a nature preserve and the second instance is obtained from mobility data in China which represents users connect to mobile stations to visit the IP in real time.

As shown in the Figure 4.4, there are three different sorts of vehicles rendering by circles with different sizes and colors. Those vehicles are visiting the location camping 1 shown by the red circle. In this example, visiting frequency in the certain time stamp

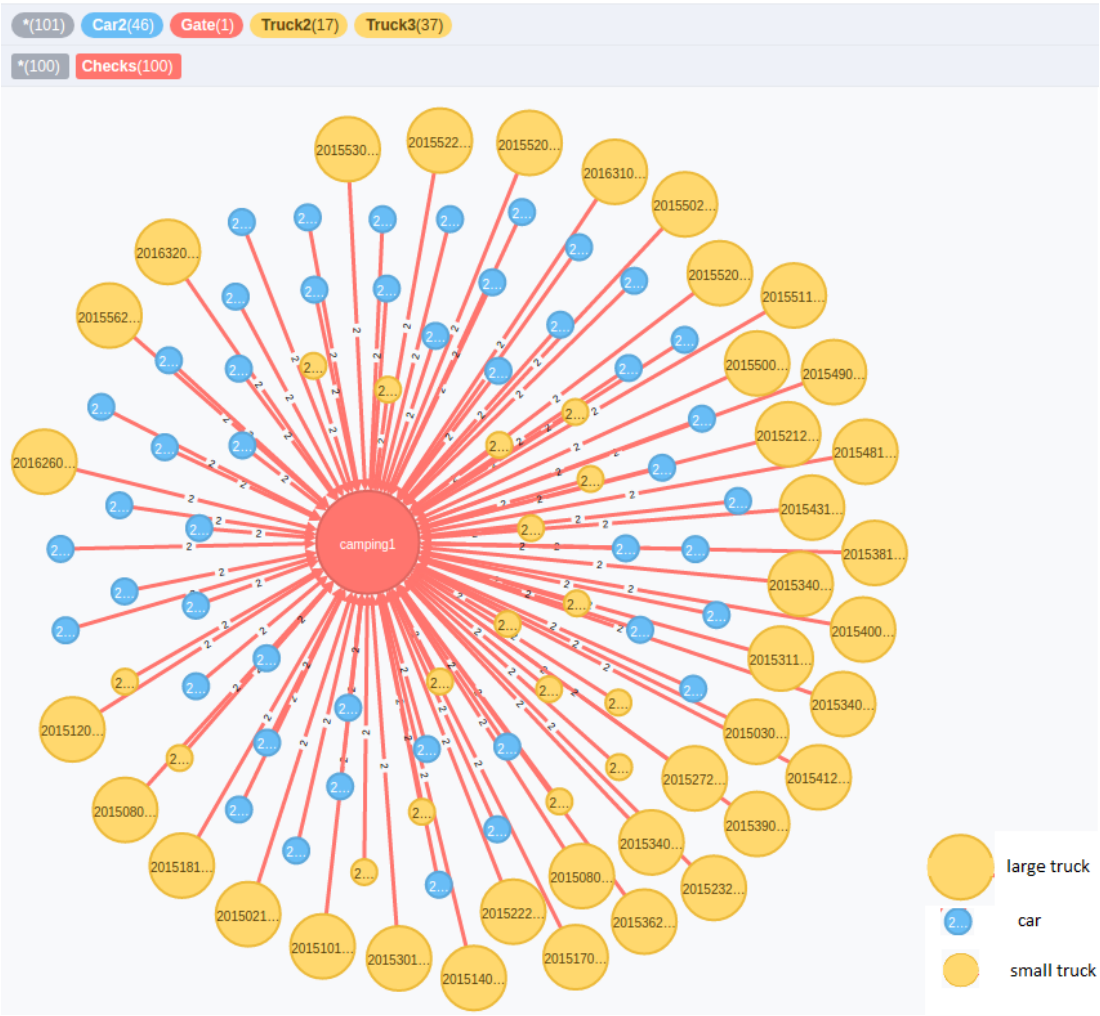

Figure 4.4: Relationship Data Visualization Example 1, different vehicles (large trunk (yellow), small trunk (yellow) and car (blue) visits the camping 1 (red).

is illustrated, so that user can analyze how frequent the vehicle check in this place with this visual result.

Figure 4.5 tells a different story that users (rendering via blue circle) are using stations (red circle). This relationship shows how many users would connect to mobile station and the number of times they call the IP address in certain duration. From that relationship pattern, we can study the visiting frequency and activate pattern of either stations or users.

Overall, using relational database is the best choice to visualize the visiting relationship

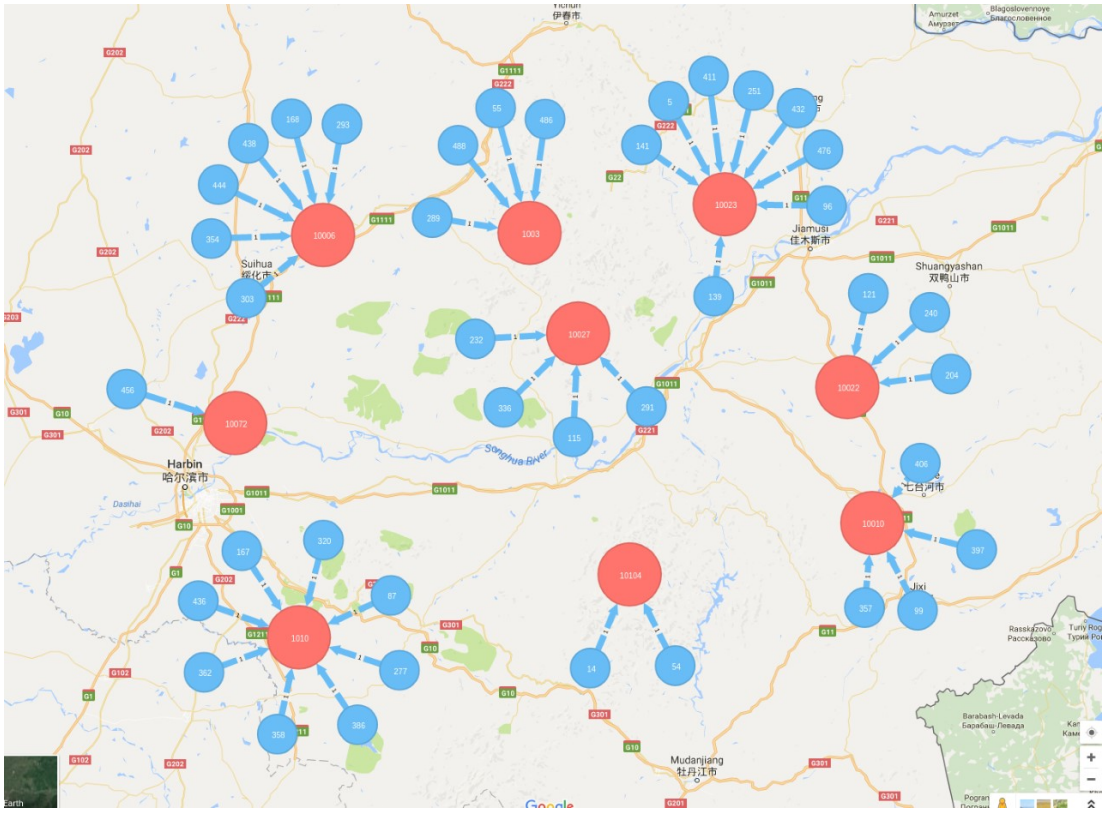

Figure 4.5: Relationship Visualization Data Example 2, Users (red) use Stations (blue).

between location and traveling objects. However, the only problem is that database normally stores massive sets of data, therefore if we only use visual analytics directly from the database, it will cause expensive experiments and unclear results which we have discussed this in the Chapter 2. So in the chapter 4, 5 and 6 we will talk about how to extract the features of those data using this data model. Generally, constructing database is always the first step to consider beyond all other visual analytics and features extraction step. All other methods strongly relies on this data model.

4.1.3 Index

To enhance the performance of database so that we can slice and query data faster. Index is necessary [27] to be emphasized. In our approach, Index will be set up for both Moving object and the location domains.

4.1.3.1 Spatial-oriented Index

Spatial-oriented index is used for querying the visiting information. Common method is to set index based on the coordinators X and Y for 2-dimensional range search [5]. Differently, as we have extracted the key points and mapped the continues coordinators information to the discrete location label for every user in the processing part (section 3.1.1). So the index will be constructed based on the identification (id, label etc) of attractions rather than their spatial information.

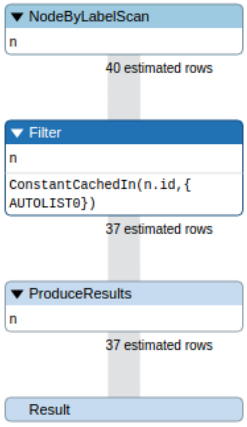

Figure 4.6: This query is relevant to spatial information which scans 104 rows.

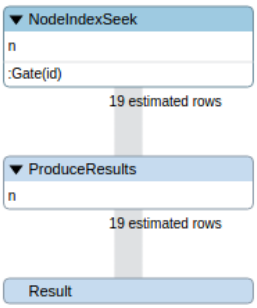

Figure 4.7: Same Query to Figure 4.6 with Spatial Index only retrieves 38 rows.

Figure 4.6 shows that the query relevant with spatial information without the spatial index and the query will scan the whole database. However in Figure 4.7, it shows that index can avoid the whole table scan which significantly enhance the performance.

4.1.3.2 Object-oriented Index

Object index is similar to spatial index but is set up on the moving object domain. The index is based on the object information such as object id. Using this index we are able to enhance all the queries that relevant with objects.

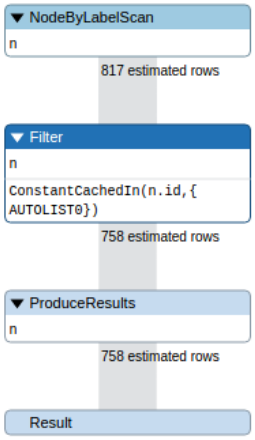

Figure 4.8: This query is relevant with object information which scans 2387 rows.

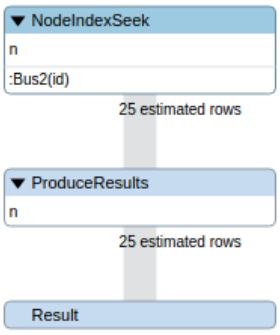

Figure 4.9: The same query to Figure 4.8 with spatial index only retrieves 50 rows.

Figure 4.8 shows the query with moving objects without using index which scan sthe whole object table while Figure 4.9 presents the the same query with index which retrieve the information much efficiently thanks to index.

4.2 Dynamic Database

Dynamic database is only going to be generated temporally once users query from the static database using time and spatial information. Afterwards, those temporal matrices will be piped into the clustering or prediction algorithms respectively for extracting features and then generating the visual analytics result. In order to save memory, the dynamic data would be eliminated once the result is output. Specifically, there are four types of data will be defined as dynamic data model that are Visiting Frequency Matrix, Time Spent Matrix, Sequence Vector and Time-Oriented Matrix.

4.2.1 Visiting-Frequency Matrix

To indicate how many times a visitor visit in every location, we use visiting frequency matrix. In this Matrix, the frequency of check-in will be stored. Table 4.1 is an example of such matrix. This example represents a visitor-location-check-in tensor. Each vector oriented by traveler indicates a visiting frequency pattern and the value of each dimension is the number of check-in. For instance, traveler 2 visits location 2 12 times. To generate such matrix, we need to use algorithm 2.

User/Location	location 1	location 2
Traveler1	12	43
Traveler2	0	12
Traveler3	14	0

Table 4.1: Visit Frequency Matrix Example

Algorithm 2 Visit Frequency Matrix Algorithm

$D(id, location, start, end) \leftarrow$ Raw Data

$count(id_1, id_2, ...) \leftarrow Group[(id_1, id_2, ...), (location_1, location_2, ...)]$

$D_{new}(id, location, count) \leftarrow$ New Data

$D_{out} \leftarrow D_{new}$ pivot (raw = 'id', columns = 'location', values = 'count')

This matrix is used for visiting frequency pattern clustering. From clustering result, we will see the similarity of each moving objects pattern. By this, we can further group

those objects. That is to say if those travelers have exactly the same visit frequency, they could potentially be in the same group.

4.2.2 Time-Spent Matrix

Only using frequency matrix will be not enough for showing the visiting pattern of visitors. Specifically, assuming there is a traveler visiting Attraction A once and Attraction B once. The value in frequency matrix will be 1 for both dimension in the frequency vector. That shows these two places have equivalent weight of visiting for this visitor. In other words, this visitor has the same interests in both Location A and B. However, in this case, this user might spend whole morning checking in Attraction A but only couple of minutes passing by Attraction B which presents entirely different interests and patterns. Therefore, simply using frequency count might lost a lot of information of visitor behavior information. Hence, we also need to consider the period that visitor spend in certain attraction. In this section, another essential data type we are going to mention is called Time Spend Matrix.

User/Location	location 1	location 2
Traveler1	28 min	13 min
Traveler2	23 min	12 min
Traveler3	31 min	42 min

Table 4.2: Time Spent Matrix Example

Above table represents an example of time-spent matrix. In this instance, the matrix illustrates user-location-time-spent information which is time that three visitors spend in two different locations. Each vector records the time-spent pattern for one visitor. Traveler 2, for example, spent 23 minutes in location 1 and 12 minutes in location 2.

Algorithm 3 Time Spend Matrix Algorithm

$D(id, location, start, end) \leftarrow$ Raw Data

$(\sum end_1 - start_1, \sum end_2 - start_2, ...) \leftarrow Group[(id_1, id_2, ...), (location_1, ...)]$

$D_{new}(id, location, \sum end - start) \leftarrow$ New Data

$D_{out} \leftarrow D_{new}$ pivot (raw = 'id', columns = 'location', values = $\sum end - start$)

Similar to frequency matrix, time spent matrix is to record the duration that travelers spend in locations. By calculating the total amount of time that visitor spent on on each location we can see more information of the visiting pattern.

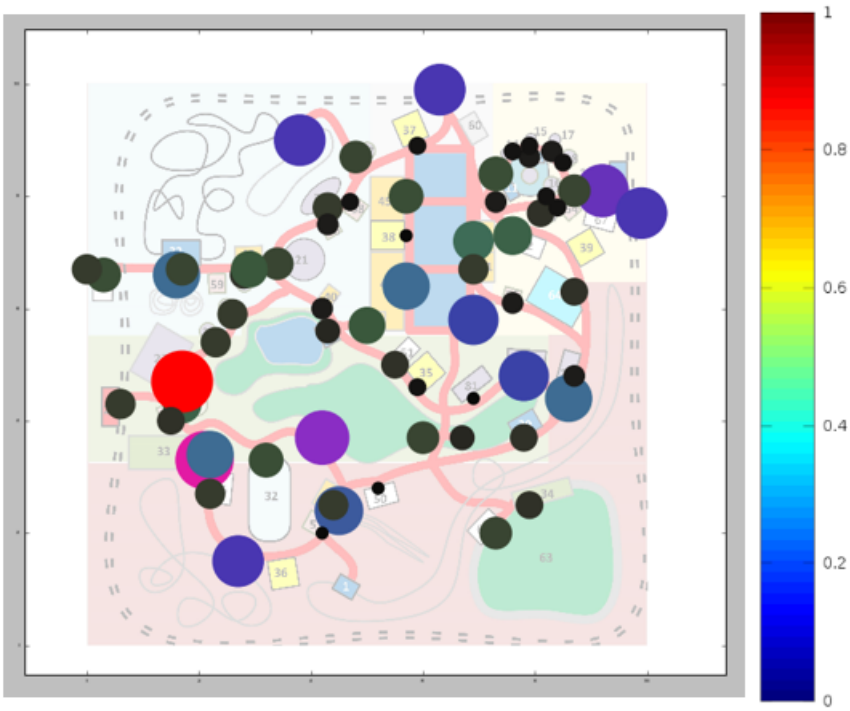

Figure 4.10: Heat map of visiting pattern, the color and size indicates the density of time spent in each location.

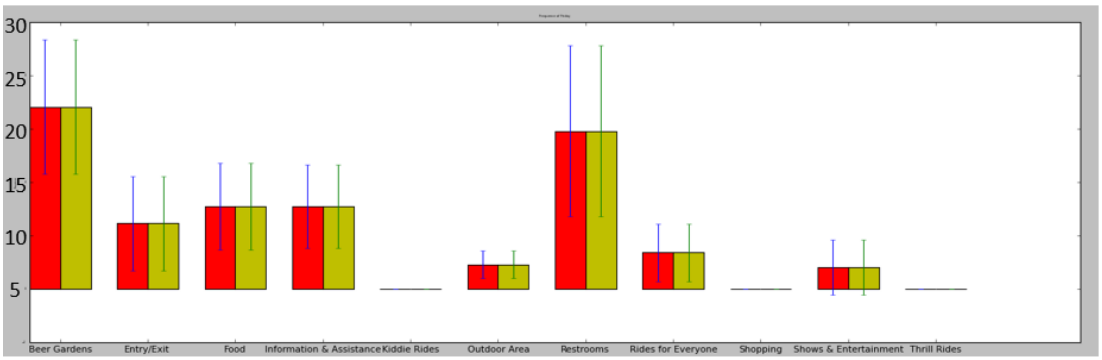

Figure 4.11: Bar chart of visiting pattern indicates frequency of check-in for each category of attractions, y-axis shows the frequency that visitors check in for each category.

The purpose of this dynamic matrix is to complete the information of visiting pattern

that frequency matrix might lost. Using these two matrices frequency and time-spent, we can further analysis the visiting preferences pattern of visitors. By clustering of such pattern, it is possible to detect abnormal travelers if they have in-regular pattern.

Both frequency matrix and time-spent matrix can be visualized using heat map and bar chart. Figure 4.10 indicates heat map of time spent for every single attraction while in Figure 4.11, the frequency of check-in for each category is shown with bar chart.

4.2.3 Sequence Vector

Sequence vector records the location that a traveler visits in certain period without considering time information. Such vector will simply track places a traveler check in using a one-dimensional vector.

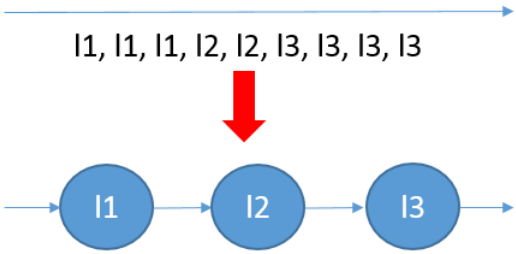

Figure 4.12: Sequence Vector Generation

Algorithm 4 Sequence Vector

$D(id, location, start, end) \leftarrow$ Raw Data
$id_u \leftarrow$ User ID
$locations[] \leftarrow$ User ID
for $D_i(id_i, location_i, start_i, end_i)$ **in** D **do**
  if $id_i = id_u \wedge location_i \neq locations.last$ **then**
    Add location to locations
  end if
end for
return Locations

Figure 4.12 and algorithm 4 show the process to generate sequence vector. This type of vectors is used to reveal the visiting sequence. Those sequences can be rendered as animation to conduct the traveling story of moving objects.

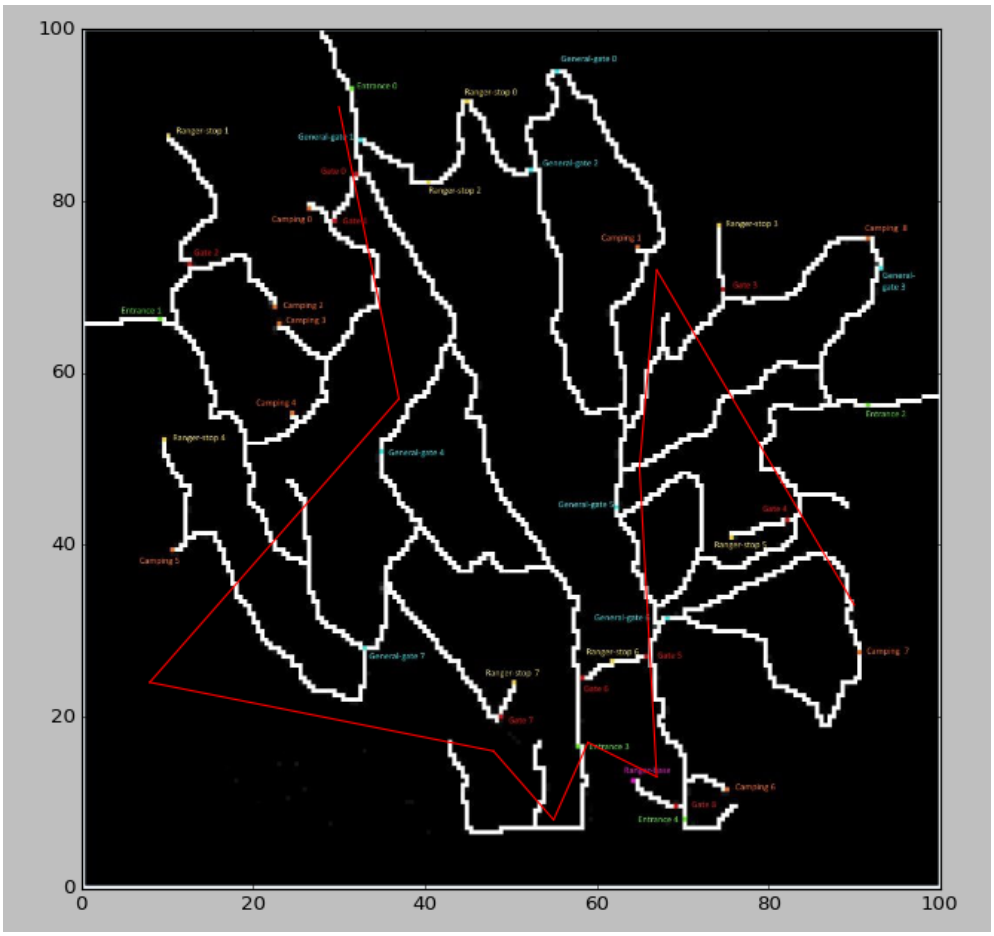

Figure 4.13: Animation for showing traveling sequence of a user, red path indicates the sequence of traveling in 2D map.

The intent of this type of vectors is to show the sequence that a visitor travel around stay points. For instance, Figure 4.13 illustrates the animation of sequence that a visitor travel along in a park. Using such dynamic graphical technique, we can clearly see the traveling order of a visitor therefore summarize the traveling story of this traveler.

4.2.4 Time-Oriented Matrix

The biggest drawback of the sequence matrix is that it dropouts the time dimension, which is a very essential information. So that we will only get the series of stay points that a moving object checks in or passes by without any time spent information. If we only use such matrix for prediction, the prediction result will not be accurate as much information loss. Therefore, to avoid such situation, a new dynamic matrix called time-oriented matrix will be introduced in this section.

User/Location	10:00:01	10:00:02
Traveler1	location 1	location 2
Traveler2	location 3	location 3
Traveler3	location 1	location 1

Table 4.3: Time Oriented Matrix Example

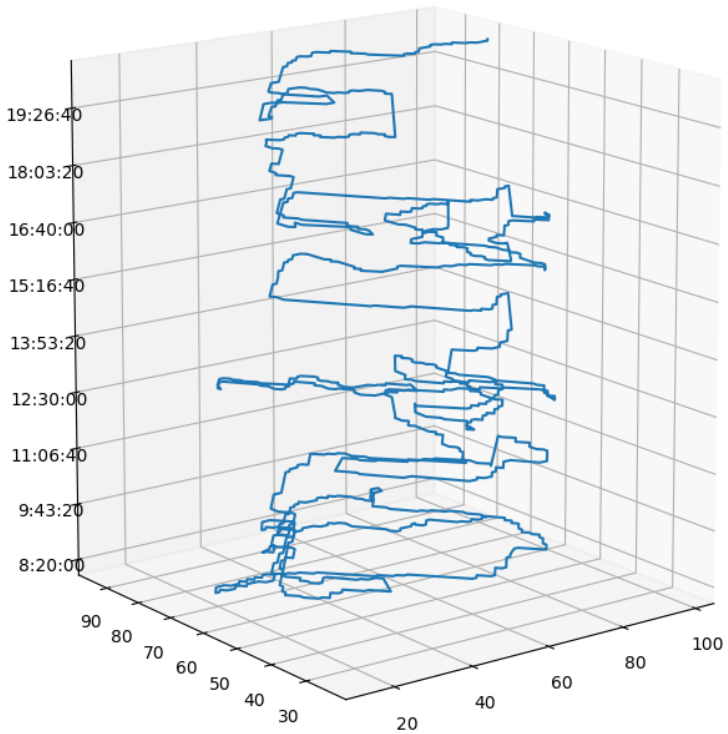

Figure 4.14: Example 3D cube for visualizing trajectory using time-oriented vector. Z-axis is time information, X-axis and Y-axis are coordinator on the 2D map.

Algorithm 5 Time Spend Matrix Algorithm

```
D(id, location, start, end) ← Raw Data
start ← Start Time
time[] ← TimeStamp List
end ← End Time
D_new ← new Data
Dic ← dictionary
Matrix ← Matrix
for D_i(id_i, location_i, start_i, end_i) in D do
    if start_i >= start ∧ start_i <= end OR end_i >= start ∧ end_i <= end then
        Add D to D_new
    end if
end for
for D_i(id_i, location_i, start_i, end_i) in D_new do
    if dic[id_i] then
        Add [start_i, end_i] to dic[id_i][location_i]
    else
        dic[id_i][location_i] = [] Add [start_i, end_i] to dic[id_i][location_i]
    end if
end for
for time_i in times do
    if There is a [start_i, end_i] in dic[id_i][location_i] that start_i < time_i and end_i >
time_i then
        Add location_i to Matrix[id_i][i]
    end if
end for
Return Matrix
```

Time-oriented matrix have two dimensions for representing user-location-traveling-stay information. Specifically, X-dimension represents the users list and Y-Dimension represents the time that is divided into certain time duration. The value of matrix will be the discrete identification value of each location that a traveler stays in at certain time stamp. Table 4.3 shows an example of such matrix. From that example, we can see that every single vector from time-oriented matrix records the a set of stay points

that a visitor checks in in certain time stamp. For instance, Traveler 1 stays in location 1 at 10:00:01 and location 2 at 10:00:02. To compute such matrix, we will use algorithm 5.

This matrix can be utilized to predict the location visitor will attend in next time stamp. We will explain that in detail in Chapter 6 Prediction. Moreover, this information can be used for visualizing the trajectory. That is we can plot the single time-oriented vector using 3D time-cube Plot showing as Figure 4.14, where z-axis is the time information, x-axis and y-axis are the coordinator for 2D map.

4.3 Method Summary

Overall, this chapter summarizes the data model. The architecture is from relational static model to dynamic database. For relational data, the most essential relationship is visitor checks in stay points. Attributes in the relationship model includes the information such as frequency of check-in as well as the time visit start visiting in and end with checking out. To enhance the performance, index is constructed on both moving objects and locations.

Regarding the dynamic data, there are four data that check-in frequency, time-spent matrix, sequence matrix and time-oriented matrix. Check-in frequency and time spend matrix is generated for clustering and can be visualized using methods such as heat map and bar chart. Sequence matrix can be revealed by animation while time-oriented matrix is used for prediction and can be visualized using 3D time cube.

There are two main advantages for using this data model. For one, relational data model set is easy for constructing from original raw data set but powerful enough to store all information we need. For another, dynamic data set will be eliminated after usage, therefore there will be not extra memory required. Overall, the rest two methods of this thesis will highly rely on this data model. That is to say data model is not an independent part but very essential throughout whole framework. It will support features extraction algorithms and front-end visualization tools.

Chapter 5

Self-Organizing Map Based Clustering

In this chapter, I would explain how we use Self-organizing Map (SOM) to cluster and recognize visiting behavior patterns. To begin, moving objects with similar or the same behavior pattern would be clustered and grouped. Afterwards, some distinct groups will be interpreted for analyzing their behavior patterns. Finally, visualization results of those patterns would be rendered using visual analytics tools such as heat map. The beauty of this clustering approach is that we can use it to detect the groups as well as interest preferences of those groups will be analyzed. With those results, some outstanding and abnormal groups will be found out. This method can be applied into different areas such as crime detection.

5.1 Clustering and Pattern Recognition

Classical clustering algorithms of K-means and Gaussian Mixture Model are widely used for clustering trajectory data [11] [21] . However, such clustering methods cannot always efficiently produce meaningful results especially for irregular movement patterns such as people movement in a city. This is mainly because K needs to be defined in advance for K-means and finding an optimal K value is a non-trivial work. In addition, these clustering methods are sensitive to noise especially for massive cluttered movement data [5]. Therefore, in this section, another unsupervised approach Self-organizing Map would be introduced. Furthermore, most of previous work tried to cluster the movement trajectory directly. However, such clustering become very

difficult for clustering large and irregular trajectories. Different from those, in this approach, I cluster moving people based on their preferences of locations (time spent and frequency matrix). This clustering method helps to find people who have the high potential to be in the same groups once they have similar or even the same interests. Based on those, we can further infer what category of visitors they are. For instance, a group of family with children moving in a park would probably have higher check-in frequency on kiddie rides than a group of teenagers

Algorithm 6 SOM Cluster and Visual Analytic

```
data ← Original Data
t ← training time
map ← Initialization
repeat
  for t_i in (0, t) : do
    BMUs ← Competition
    for b in BMUs do
      b_NeighborCircle ← Cooperation
      for n in map do
        if n in C_NeighborCircle then Adaption
        end if
      end for
    end for
  end for
until t_i reach t
U_matrix ← map
for O_outstandingcluser in U_matrix do
  T_trajectory ← data + O_outstandingcluser
  V_pattern ← data + O_outstandingcluser
  for O_otherCluster in U_matrix do
    D_distance ← DTW(O_otherCluster, O_outstandingcluser)
  end for
  Output T_trajectory, V_pattern and D_distance
end for
```

Essentially, there are two parts for this work. For one, we use SOM to train and render a feature map after clustering. For another, after clustering result is rendered, we use associated visual analytics approach to recognize different patterns from the rendered Unified Distance Matrix from SOM. The main algorithm is shown in Algorithm 6. There are couple of essential steps including **Initialization**, **Competition**, **Cooperation Adaption Group Aggregation** and **Visiting Pattern Analysis**. The detail of those parts will be explained in the following sections.

In addition, associated visual analytics is crucial in this approach. Using those approaches to visualize the patterns such as visitor preferences on different locations will be useful to understand movement behaviors. Besides, other features including communication, time spent on a site, preference categories are also considered as essential features for clustering.

5.1.1 Initialization

Initialization creates a 2D-array map with nodes in the lattice and each node has an arbitrary value in its different dimensions. Figure 5.1 shows how SOM projects a three-dimensional vector onto a 4×4 2D lattice map.

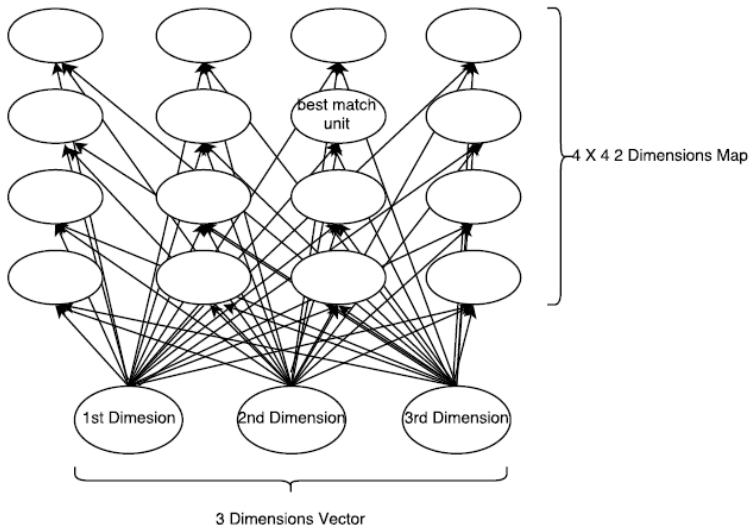

Figure 5.1: Initialization shows how to project 3 dimensional vector to a 2D grids

Using initialization we can construct the two-dimension map with each node representing different vector that are for projecting the input vector.

5.1.2 Competition

Competition uses Euclidean distance as equation 5.1 to calculate the weight distance D between every node W and input vector V in a 2D map. It then defines a node with minimum weight difference as Best-Matching Unit (BMU).

$$D = \left\| \sum_{i=0}^{i=n} (V_i - W_i) \right\|_2^2 \tag{5.1}$$

Hence, for every single input vector, there will be a node that with the smallest distance throughout the map. We then define such node as BMU. This step is to compute all BMUs in order to locate the neighborhood circle in next stage.

5.1.3 Cooperation

Cooperation uses radius decay function to locate the neighborhoods of BMU in each iteration. There are two sub-steps in this stage. The first step is to compute the distance between the node and the BMU with Equation 5.2.

$$D_{bmu} = \left\| (X_{bmu} - X_{node_n})^2 - (Y_{bmu} - Y_{node_n})^2 \right\|_2 \tag{5.2}$$

The second step computes a decay radius of neighborhood circle which takes the BMU as the center. In Equation 5.3, σ_t is the decay radius of neighborhood, σ_0 is the initial radius of circle which is equal to the map width, t is the iteration time and λ is a constant number for learning rate. The last step is to compare every node to see whether they are inside their neighborhood circles, that is $D_{bmu} < \sigma_t$. If yes, such nodes are defined as neighbors of related BMU.

$$\sigma_t = \sigma_0 e^{(-\frac{t}{\lambda})} \quad t = 1, 2, 3, ... \tag{5.3}$$

After we completed these two steps, the neighborhood circle will target the nodes inside the circle and update their values in the next step. The architecture of neighboring is as follow, Figure 5.2 illustrates how radius decay works.

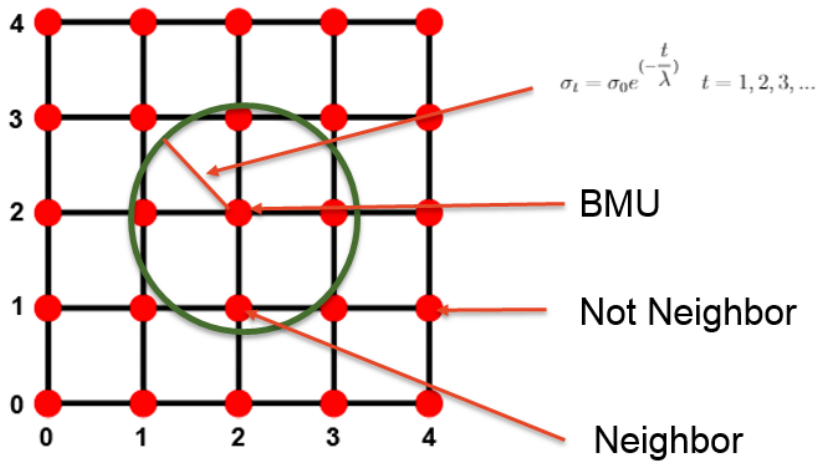

Figure 5.2: Neighborhood Circle (green), the center of circle is BMU

5.1.4 Adaption

After all neighborhood circle is located with BMU as center, SOM will update all the nodes inside the neighborhood circle with their weight W as in Equation 5.4. This step is called Adaption.

$$W_{t+1} = W_t + \Theta_t L_t (V_t - W_t) \tag{5.4}$$

In this equation, Θ_t is used to consider neighborhood in the weight adaptation, L_t is the learning decay, for computing the learning rate in each training iteration. L_t are serving as factors that affect the current iteration time and the initial learning rate L_0. Learning rate through time L_t is calculated with Equation 5.5.

$$L(t) = L_0 e^{(-\frac{t}{\lambda})} \quad t = 1, 2, 3, ... \tag{5.5}$$

The effect of neighborhood on the weight of nodes is the function of the distance between the current node and BMU, the radius of neighborhood circle, and the current training iteration time as shown in Equation 5.6.

$$\Theta(t) = e^{(-\frac{D^2}{2\sigma_t^2})} \quad t = 1, 2, 3, ... \tag{5.6}$$

After iterative update of all nodes, the map is self-organized and all nodes save their trained weights so that the information of the nodes can be encoded and rendered using Unified Distance Matrix (U-matrix). Figure 5.3 shows the an example of U-Matrix that trained using data VAST 2015, the visiting pattern on Saturday 8:00 am to 19:00 pm.

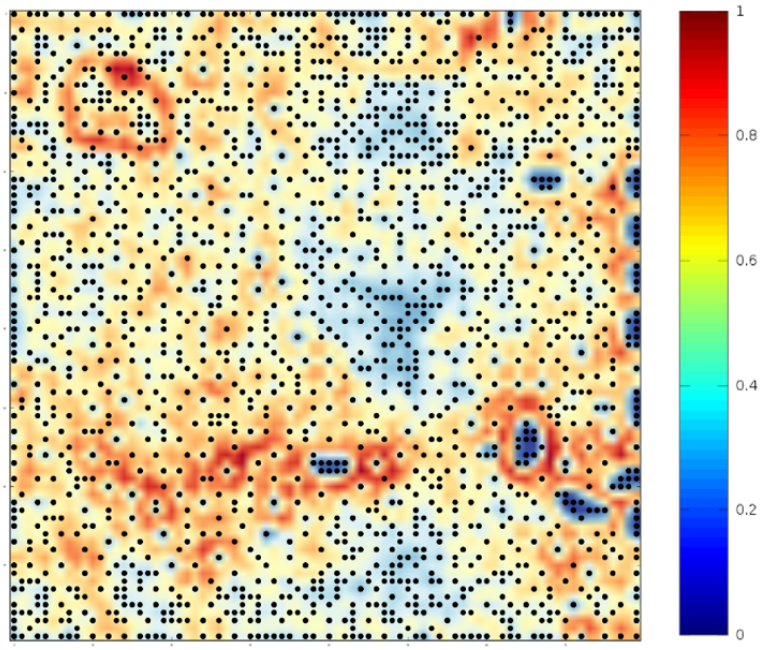

Figure 5.3: Example of SOM U-Matrix, each point indicates a individual visiting pattern. The distinct clusters can be seen. Color embeds the difference information to show some clusters are distinct from others. The value is normalized from 0 to 1 shown in the right side.

5.1.5 Group Detection

After the visiting pattern matrix is rendered, we can use those clustering results to detect the group among visitors. Different from the traditional work [5], our group detection methodology is entirely based on the clustering result. In this method, the definition of group is that only travelers has the same trajectory or the difference of their trajectories is less than a threshold, we will say those objects is in the group. Therefore, only the moving visitors have similar movement pattern (i.e. in the same cluster) can have potential to be in the same group.

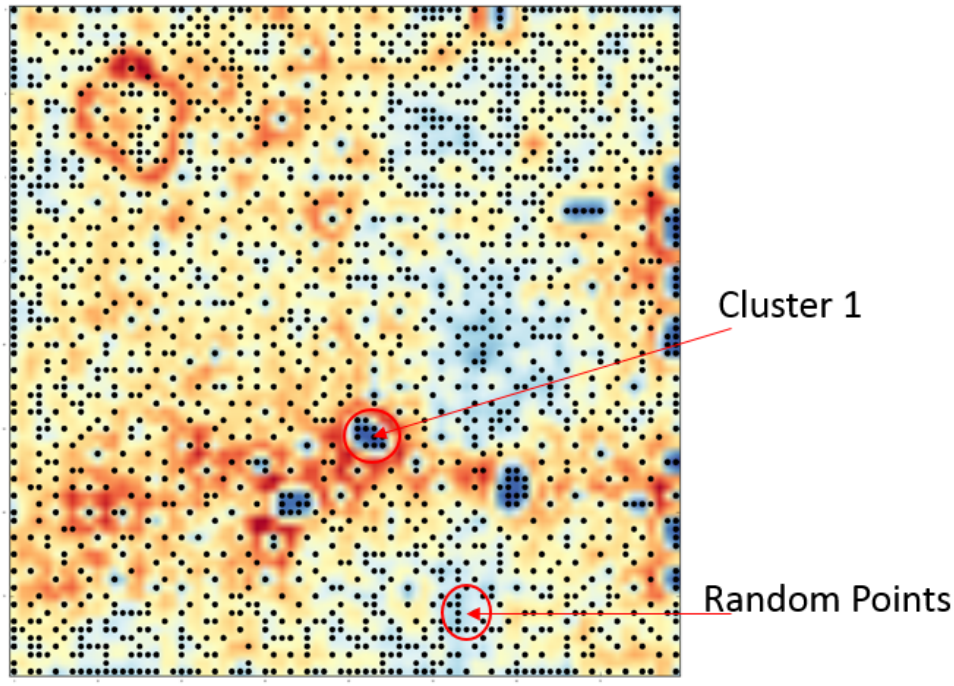

Figure 5.4: We select points from cluster 1 and a random bunch respectively for detecting groups.

In this approach, we have two main steps to analyze the group patterns. Specifically, for one, we compare trajectory of one visitor to trajectories of all other visitors in one cluster. Trajectories firstly need to be re-sampled to have the same resolution. For example, visitor A may have the movement record slightly different from visitor B. We resample both of them to make sure that in every time step they have the position data (x_i, y_i). The Euclidean distance between every time point of two visitors is computed and summed to get the trajectory distance of two visitors as shown in Equation 5.7.

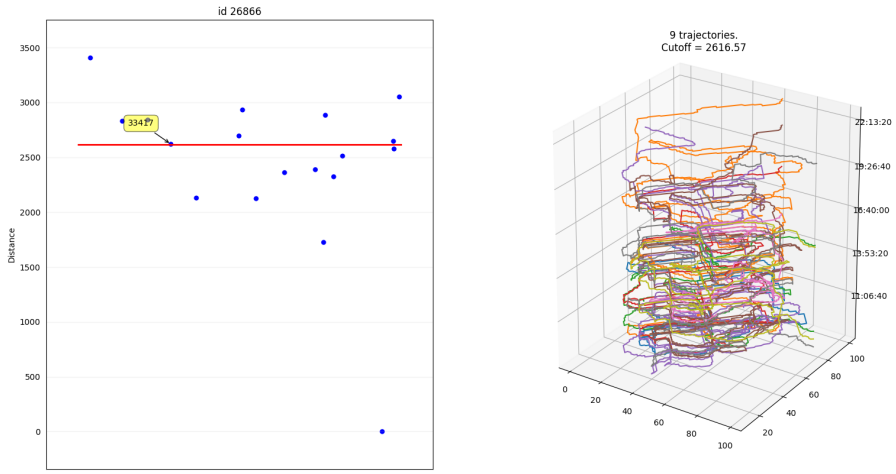

Figure 5.5: 3D view of random points, this is not a group as most the points surpass the threshold (red line) and their trajectories vary.

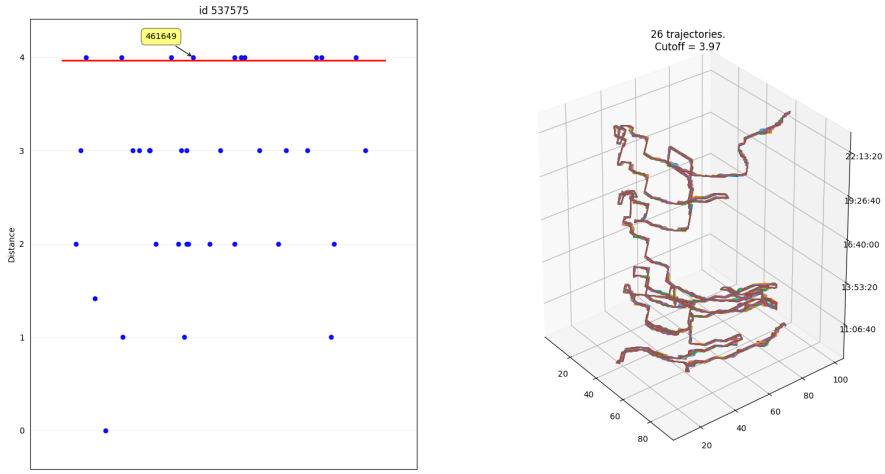

Figure 5.6: 3D view of group 1, this is a group as all the points under the threshold (red line) and their trajectories are the same.

Where D is the trajectory distance between two visitors of A and B, n is the number of time points in a trajectory.

$$D = \left\| \sum_{i=0}^{i=n} ((X_{Ai} - X_{Bi})^2 - (Y_{Ai} - Y_{Bi}))^2 \right\|_2 \tag{5.7}$$

A threshold will be set up, once D is less than that, we can assume A and B is in the same group.

Another way is to use 3D spatio-temporal visualization to show the entire trajectories of visitors. From the 3D view, we can easily tell that the difference and similarity of trajectory. In the following example, I would shows how we detect the group combine the two ways.

Figure 5.4 shows that we select two groups from the U-Matrix. The first group is selected from cluster 1 and the second group is selected from some random points which are not in any cluster.

The result shows that an example for a group and another example for a non-group. Figure 5.6 shows that we set the threshold to be 4 and clearly every point is under that threshold while in 3D view, we can see the trajectories of all the visitors are exactly the same. therefore, we know, this 26 points is in a group. Meanwhile, in Figure 5.5, even though the threshold is set to be very high (2200), most of points are still surpass this value. Also, from the 3D view, we can see that the trajectories vary a lot. Hence, those points are not the group.

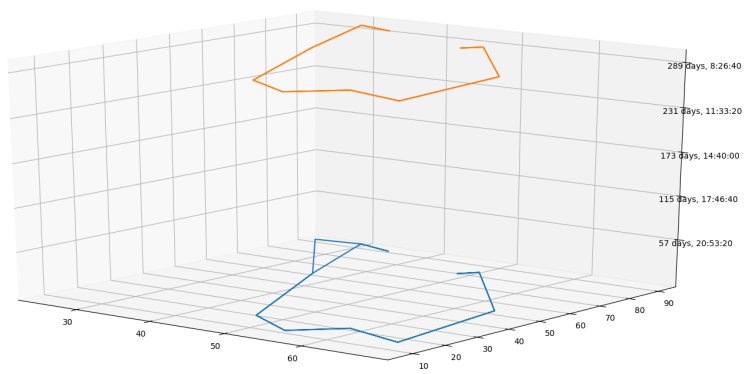

Figure 5.7: Similar patterns, path 1 (orange) and path 2 (blue) have the same trajectory but in different time period. So they are not in a group but similar.

However, there is not guarantee that points in the same cluster are always in a group, the following 3D graph illustrates a counter example. Figure 5.7 shows an example that two travels have similar trajectory that but in different time periods. Therefore, we say these patterns have similar that those visitors have the same preference of locations and travel pattern but not in the same group.

5.1.6 Visiting Pattern Through time

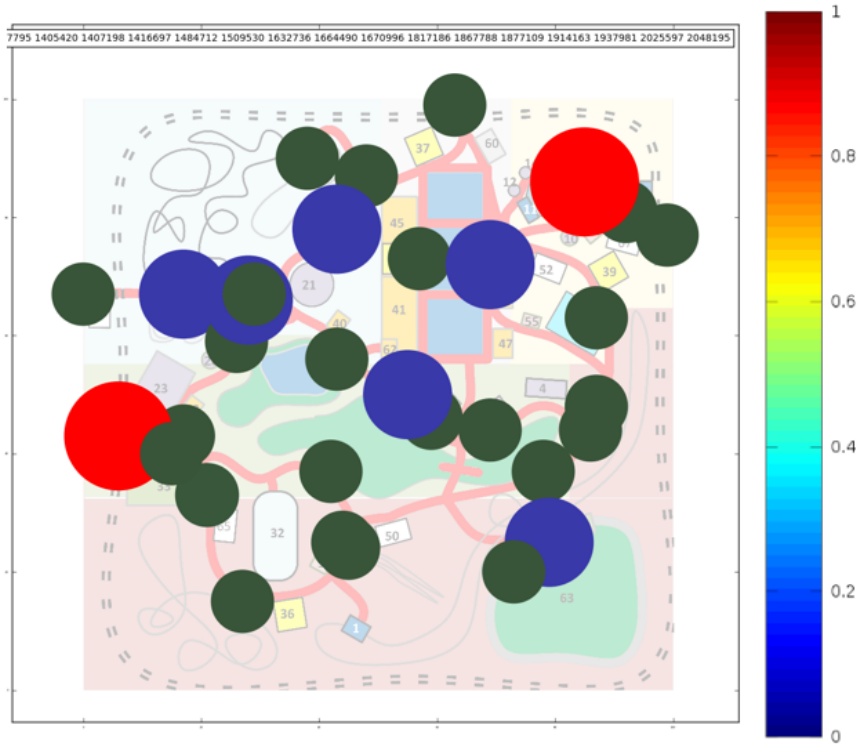

Figure 5.8: Heat map example of group visiting pattern in VAST 2015, size and color indicate the density of check-in of visitors.

Once the groups are detected, we will visualize the features of those group especially for distinct group which they have potential to be abnormal pattern and outliers. Understanding the visiting behavior patterns is essential for trajectory data mining. However traditional SOM clustering method has not any deep study after the U-Matrix is rendering [8, 18]. In my method, I will interpret those features via using the visualization tool including heat map, relationship map, animation and bar chart. With those tools,

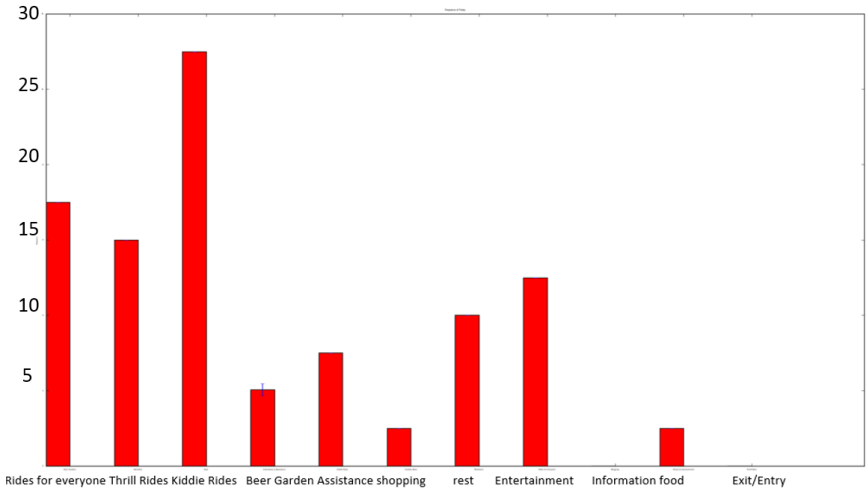

Figure 5.9: Bar chart example of group visiting pattern in VAST 2015, indicates the check-in frequency for each category.

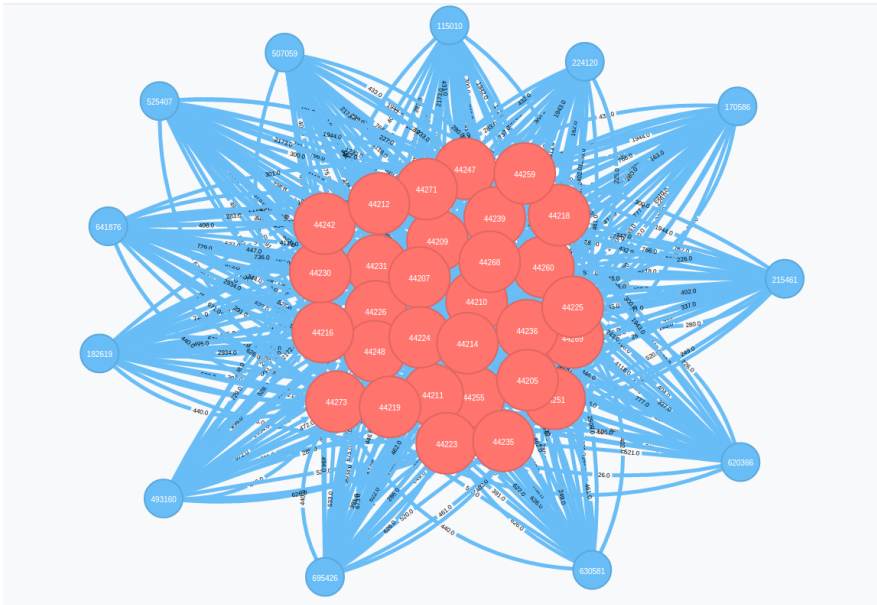

Figure 5.10: Relationship visualization example of group visiting pattern in VAST 2015 represents visitor (blue) checks in attraction (red).

we are able to have deep analyze of those groups and study their patterns.

I will take the group 1 we got from last section as an instance. This group is detected from VAST Challenge 2015, the trajectory data in the park. There are four display visualizations to show the features of that group.

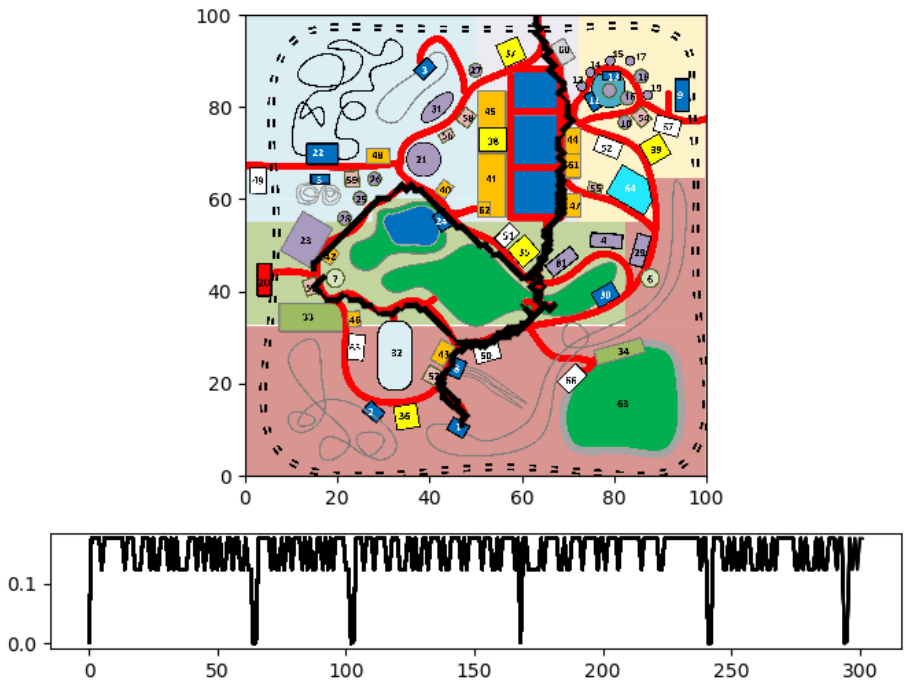

Figure 5.11: Animation for revealing group trajectory of an example in VAST 2015, black path on the map is real-time trajectory, the chart in the downside indicates speed change through time.

First of all, heat map shown in the Figure 5.8 indicates the visiting preference in different locations of this group. We can clearly see that density of check-in frequency from heat map. As the attraction is stored using tree-structure hierarchy, therefore we can further use bar chart (Figure 5.9) to represent the check-in preference of the different categories. This step is essential for understanding the group trajectory data. In that example, we see that group spent quit a lot of time in three kinds of entertainment locations. Specifically, the most time of this group was spent on the kiddie ride. Therefore, we might assume there are kids in that group and this group might be a family. Moreover, the relationship map in Figure 5.10 can show the relationship for different individual in this group checks in each attraction. Finally the whole trajectory of group can be revealed dynamically using animation in Figure 5.11.

Overall, the visual analytics for understanding group visiting pattern is important. Thanks to SOM, we can detect the group easily based on the clustering result. With those visualization, we can study pattern of the group to figure out features of them. For instance, we can target the outliers and abnormal pattern using this approach. Such methodology can be applied to crime detection, promotion and so on.

5.1.7 Dynamic Time Warping

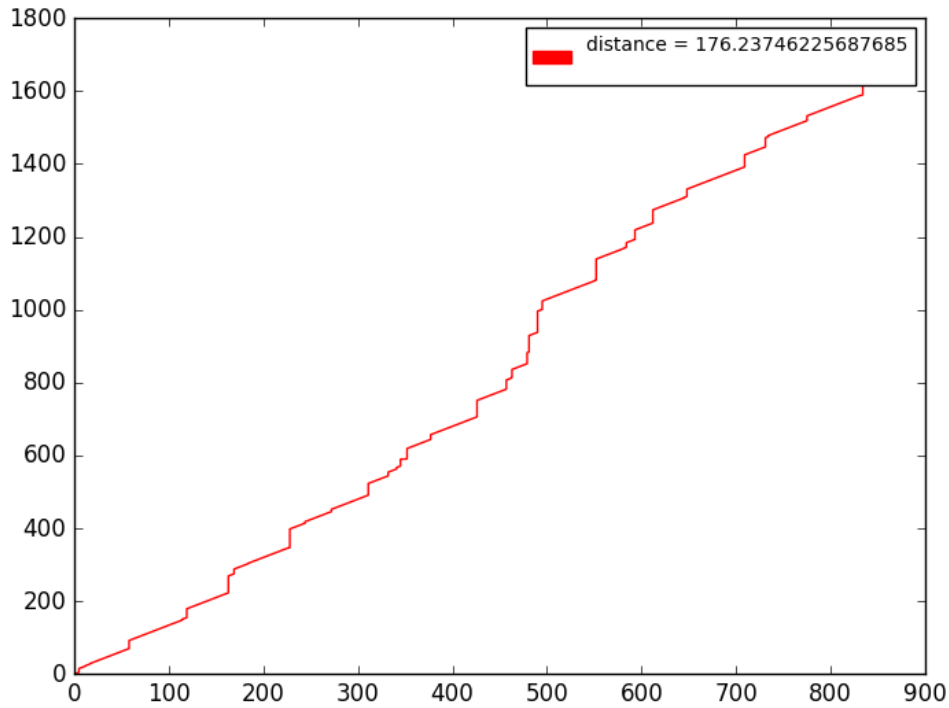

Figure 5.12: Example of dynamic time Warping Result for VAST 2015, y-axis and x-axis represent two different trajectories after dimension reduction.

The other method is compare the distance of time series using Dynamic Time Warping [28] (DTW). In this stage, distance among different groups is going to be computed. DTW is designed to calculate the similarity of two different sequence signal. Particularly, we use this algorithm to compare two trajectories of group and show the difference of them. Figure 5.12 shows the an example of DTW result that from data in VAST 2015. In that example, the distance between 2 groups is around 176. This

step can help us to find outliers. That is if there is one group having extremely large difference from all other groups, we might consider this group as abnormal group.

5.2 Experiment Result Visualization

In this section, the experiments will show how I apply SOM-based clustering to the three data set for three different visual analytics problems. The first Problem is the crime detection in the park, second task is to find out the outstanding user for commercial promotion and the last one is to detect the abnormal vehicle trajectory patterns.

5.2.1 Crime Detection for VAST 2015

As mentioned in Chapter 3, in the visiting data of DinoFun World, the local soccer star Scott Jones had a celebration called Scott Jones Show at DinoFun World from 6 to 8 June 2014 over the weekend. All his personal honors including an Olympic medal were occurred at Creighton Pavilion (Attraction 32). There was a crime group vandalized the exhibiting, breaking into the place and did crime things on Sunday, 8 June 2014 [29]. Using Self-Organizing Map based clustering method we would be able to detect this crime.

The visiting pattern of a crime group was usually different from the regular visitors [30]. We assume regular visitors normally have random distribution on the time-spent and check-in in attractions in the park, while crime person may spend particularly longer time and higher frequency of check-in in certain places which is mostly the attraction that crime occurred. The trajectories of crime groups may also be different from normal visitors, for instance, when the show in Creighton Pavilion is temporarily closed and ready for the next show, normal visitors probably leave the place travelling to another place while crime people more likely still stay there during that time as it is good time to commit crime as no many visitors around there. From these assumptions, our visual analytics firstly conducts the clustering of data in the morning on Sunday. We then visualize the visiting pattern of outstanding clusters, and figure out their trajectory and communication behavior to detect crime groups.

In figure 5.13, visualization of SOM Map shows six distinct clusters. By further analyzing these six clusters, we found that among six clusters, two of them have the highest probability to be relevant with each other. We name these clusters as group 1 and group 2.

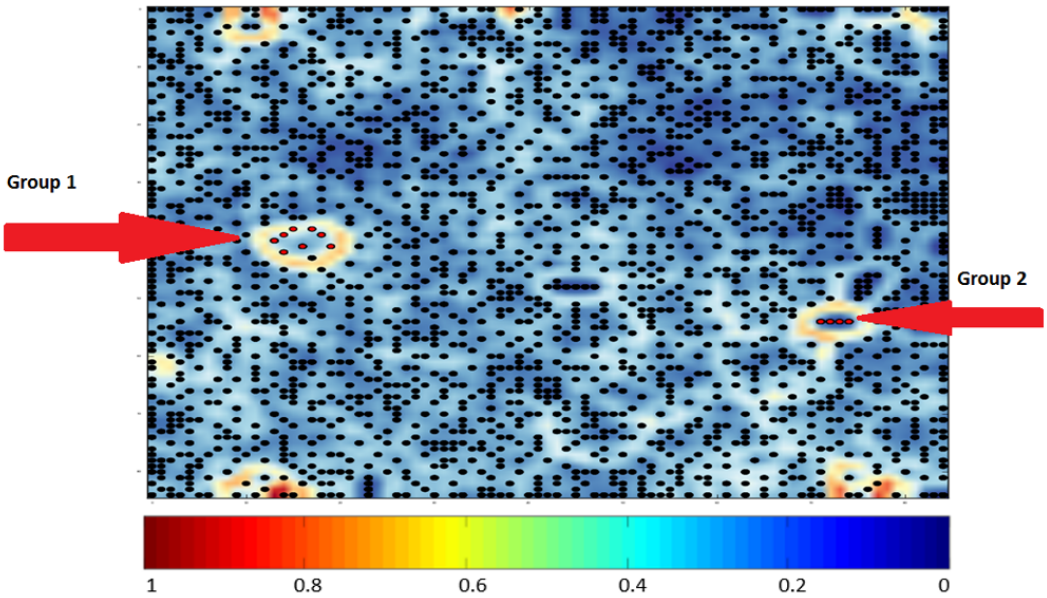

Figure 5.13: Clustering result of SOM in Sunday, there are six outstanding groups, and group 1 and 2 (red points) are considered as suspect groups.

By visualizing visiting patterns of these two groups as shown in In figure 5.14 we found that group 1 spent the most of their morning at two attractions of Attraction 53 Smoky Wood BBQ and Attraction 32 Creighton Pavilion, while group 2 spent almost the whole morning at Attraction 32 Creighton Pavilion. As both of groups spent quite a lot of time at the Attraction 32, we suspect these two groups committed a crime in Attraction 32 because of longer time spent at that attraction.

By analyzing their movement in spatial-time cube visualization as shown in Figure 5.15, we found that from 9:30 am to 11:30 am, group 1 did check-in at Attraction 32 and stayed there until 11:00am, and then moved to Attraction 53. While group 2 spent all the time from about 9:30 am to 11:30 am at Attraction 32 without any further movement.

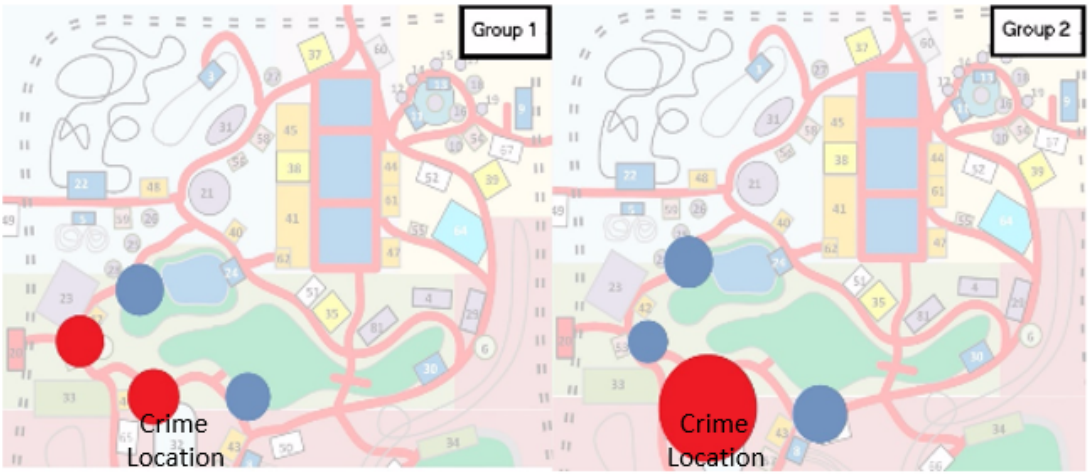

Figure 5.14: Visiting preferences of group 1 and group 2, group 1 spend quite long time visiting attraction around the crime location while group 2 spend abnormally long time in crime location.

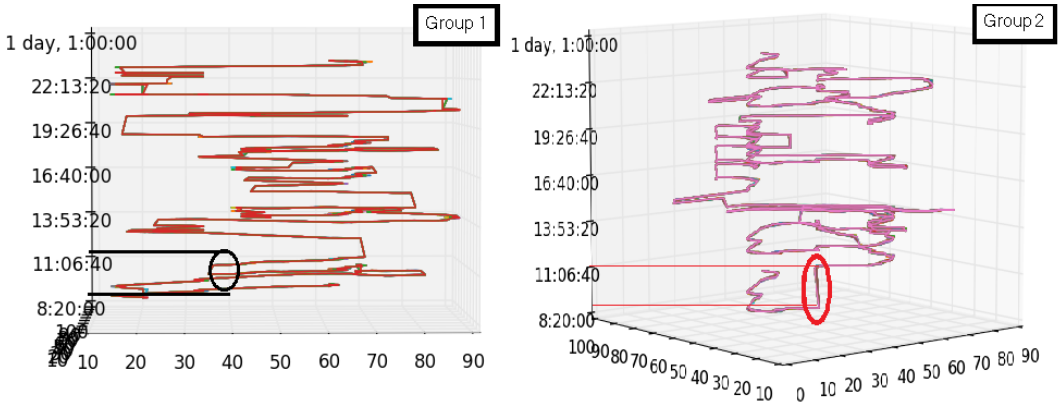

Figure 5.15: Trajectory of group 1 and group 2, group 1 keeps traveling around crime location in period crime happen while in the same period, group 2 spent their entirely time in the crime place.

In their communication visualization as shown in Figure 5.16, we found that the communication frequency of group 2 was much higher during the period from 10:00 am to 11:00 am than other time period. Therefore, we can assume that the time period from 10:00am to 11:00am was highly possibly the crime time. Based on these observations, we can conclude that group 2 committed crime while we still cannot exclude the suspects of group 1.

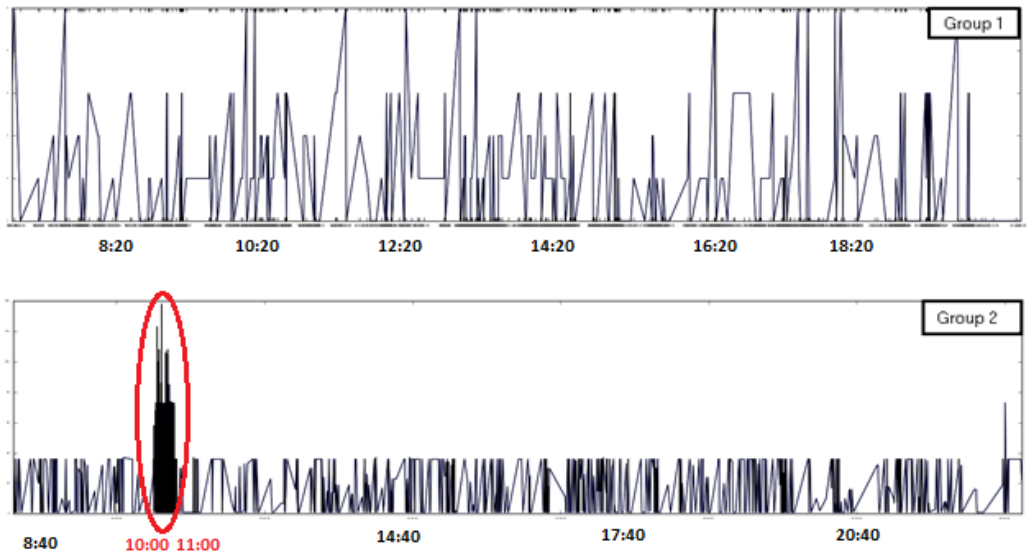

Figure 5.16: Communication of group 1 and group 2, the communication of group 2 turns to a high frequency during the crime time.

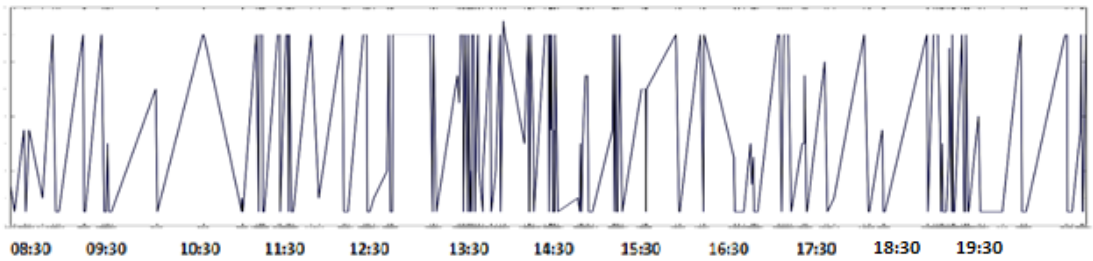

Figure 5.17: Communication between group 1 and group 2, these two groups keeps in touch of whole day, indicates that they know each other.

We further visualize the communication pattern between group 1 and group 2 as shown in Figure 5.17, the result shows that group 1 and group 2 kept in touch during the whole day of 8 June 2014, which means that group 1 and group 2 were not independent groups and they knew each other. Therefore, our conclusion is that group 2 committed crime at Attraction 32 while group 1 guarded Attraction 32 and Attraction 53 to help group 2 to commit the crime from about 10:00am to 11:00am.

Visiting data at Attraction Creighton Pavilion is further analysed to confirm our conclusions. Figure 5.18 and 5.19 shows the visiting pattern at Attraction 32 Creighton Pavilion. As shown in Figure 5.18, we found that there are three large attendance periods on each of three days, which means that Scott Jones Shows took place during these

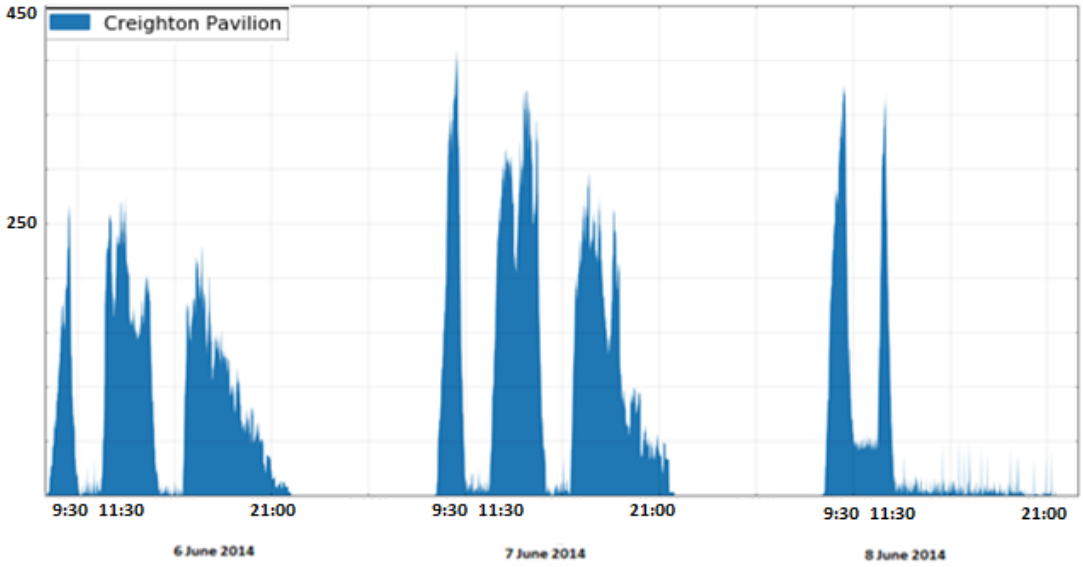

Figure 5.18: Visitor attendance at Attraction 32 Creighton Pavilion for 3 days Fri, Sat and Sun, y-axis is the number of visitors checks in, x axis is the time.

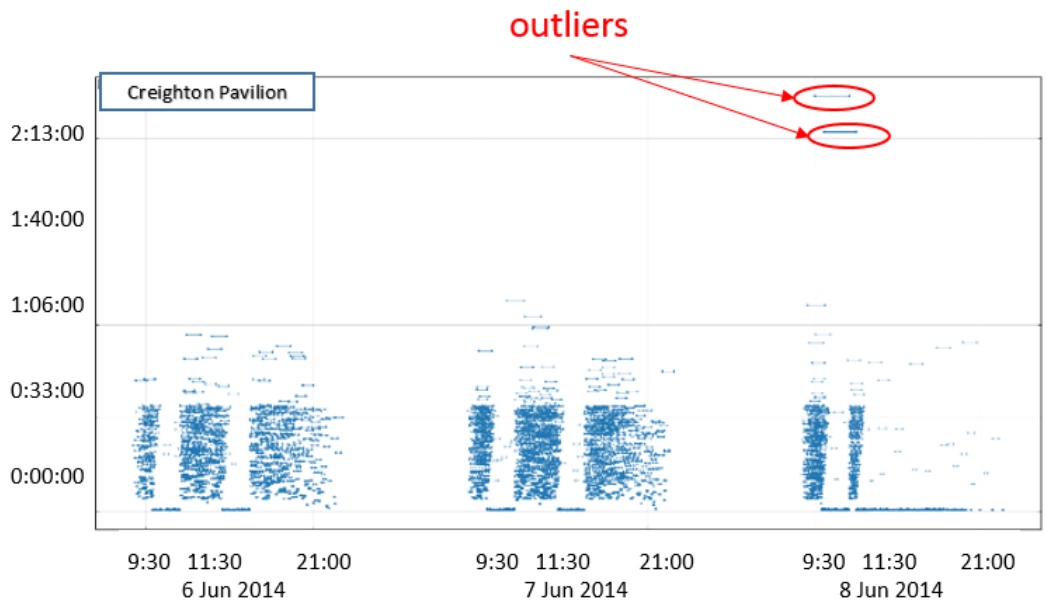

Figure 5.19: Visitor Time-spent at Attraction 32 Creighton Pavilion for 3 days Fri, Sat and Sun, y-axis is the number of visitors checks in, x axis is the time. There are two outliers (red circle) on Sun.

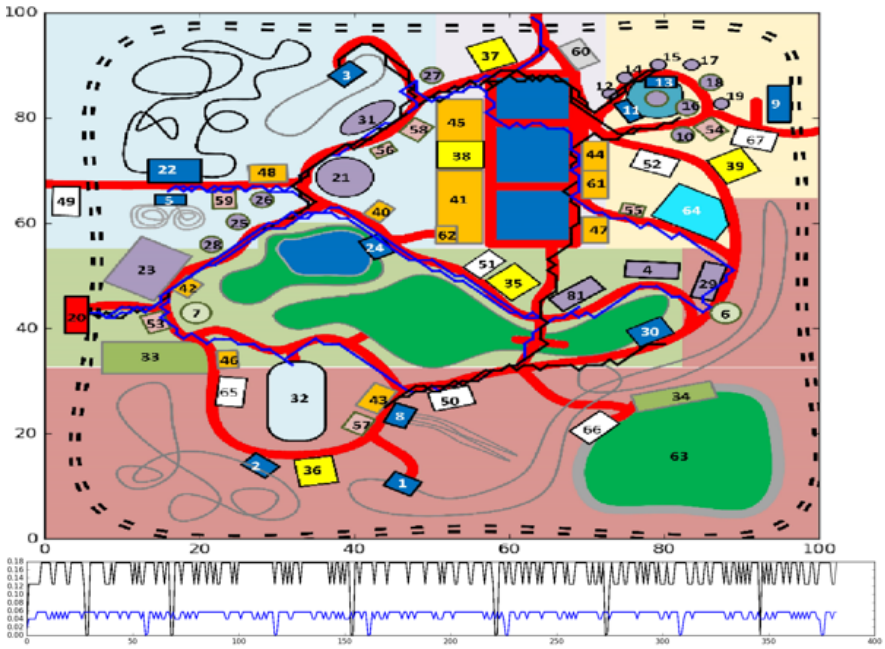

Figure 5.20: Animation for revealing the crime, there are 2 trajectories (blue and black) the chart shows the speed of them.

three time periods on each day. We also found that from around 9:30am to 11:30am every day, there was a sharp decrease followed by a very small number of check-in of visitors and then increased again. From this pattern, we can infer that from 9:30am to 11:30am this attraction was shortly closed until the next shows ready after 11:30am. Normally during this time period visitors most likely left this attraction. However from the time spend on Attraction 32 as shown in Figure 5.19, we found that there were two groups of visitors staying in this place from 9:30am to 11:30am. The time they spent at Creighton Pavilion are clearly outliers as normal visitors spent less than one hour there but they spent more than two hours in this place. Therefore, we can assume that these visitors are related to the crime.

In summary, based on these visual analytics as shown in this section, the whole crime story can be described as follows. Scott Jones Shows took place three times every day from Friday to Sunday. After the first show ending at about 9:30am, the attraction was closed until 11:30am. During this time period on Sunday, the crime people were divided into two groups: the first group mainly acted as assistance group and the second group committed the actual crime in Creighton Pavilion. The first group travelled

around Attraction 32 to guard the crime. At around 10:00am, the second group started working on breaking into exhibition and stolen. Therefore the communication in the second group was increased a lot. Two groups also kept in touch frequently during the whole crime committing period. The whole crime story can be revealed using the animation shown in Figure 5.20.

5.2.2 Promotion for Mobile User

In this task, I would show how to use SOM to cluster the mobile phone users, study their patterns of using mobile stations, and find out what kind of users we should promote to. Using the frequency check-in and time-spent data, we can see the usage pattern of users that how frequent users use stations to visit the IP address in a day as well as how long they spend their time on those stations. From the clustering result, we will be able to detect outstanding users that they use a certain station with a high frequency value.

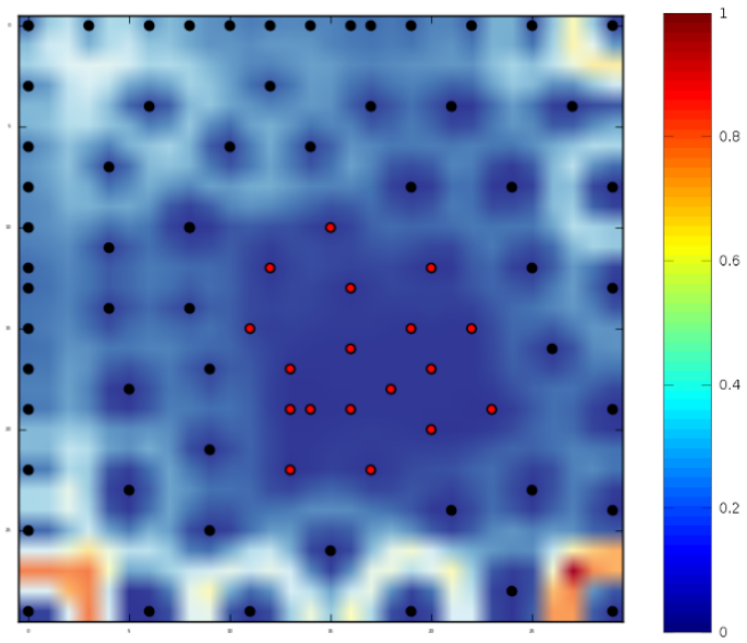

Figure 5.21: Clustering U-Matrix of Mobile Data in Station 1001 from mobile data in China, there are three distinct points in the up right, down right and down left.

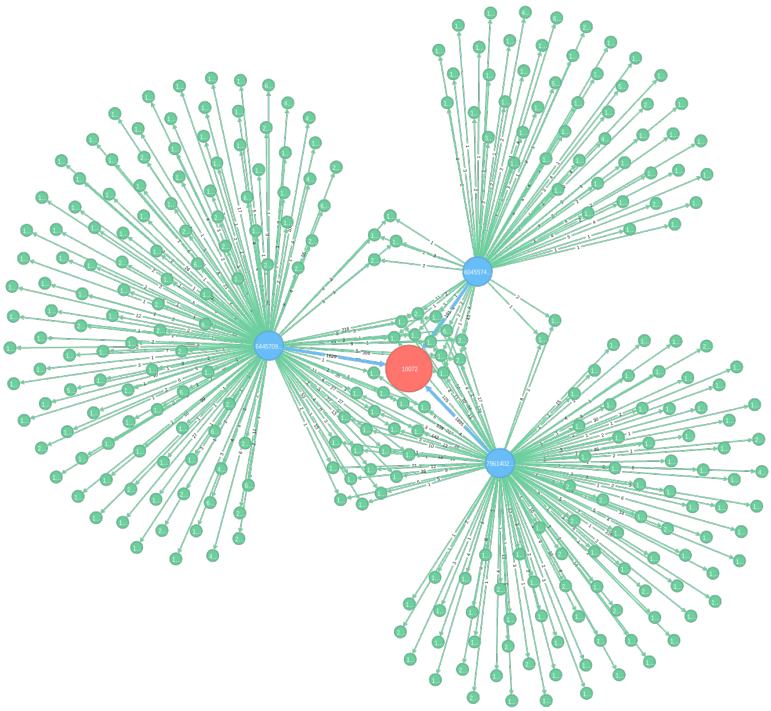

Figure 5.22: Outstanding users (blue) use stations (red) to visit IP address (green).

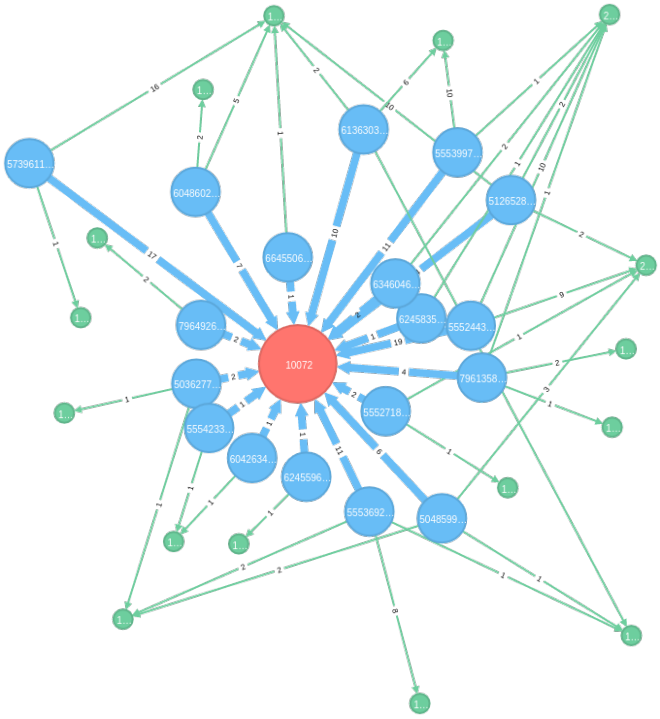

Figure 5.23: Regular users (blue) use stations (red) to visit IP address (green).

Using SOM to cluster the visiting pattern, we can get the result. There are around 150 stations activated per day. We query one station and cluster the visiting pattern for a week (5 working days) to view the result. As shown in Figure 5.21, the down left, down right and up right corners are shown as the outstanding users while the others can be interpreted as regular users.

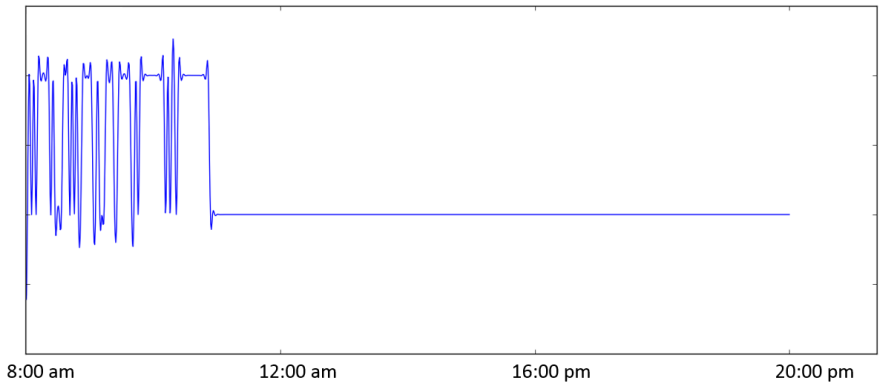

Figure 5.24: Station usage of outstanding user.

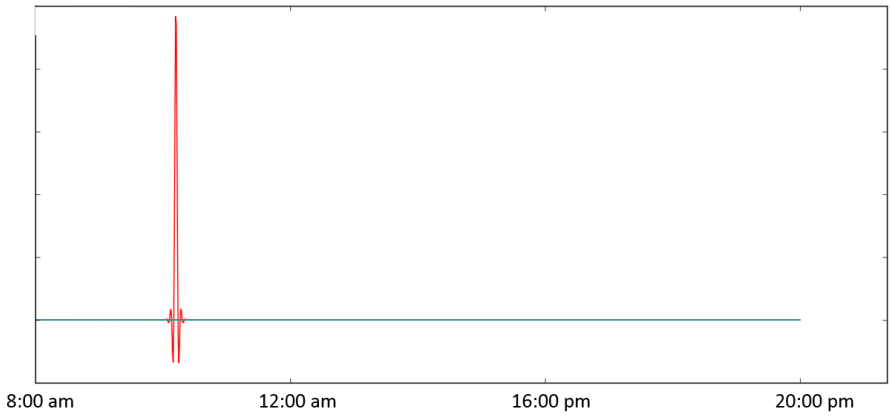

Figure 5.25: Station usage Pattern of Regular User.

By further visualizing the visiting pattern of both outstanding and regular users, we see that, from Figure 5.24 and and Figure 5.22, outstanding users use the station with a large number of frequency (Figure 5.24) and keep activity on the station in quite long

time (Figure 5.22). While the patterns of regular users (Figure 5.23 and Figure 5.25) illustrate that they only use the station with a very small number (Figure 5.23) and activity time is very short (Figure 5.25) too. Therefore, we know that if we want to have a commercial promotion, we should put more efforts on those outstanding users as they are more likely see the advertisements with more time spent on using the mobile station.

5.2.3 Abnormal Visiting Pattern for VAST 2017

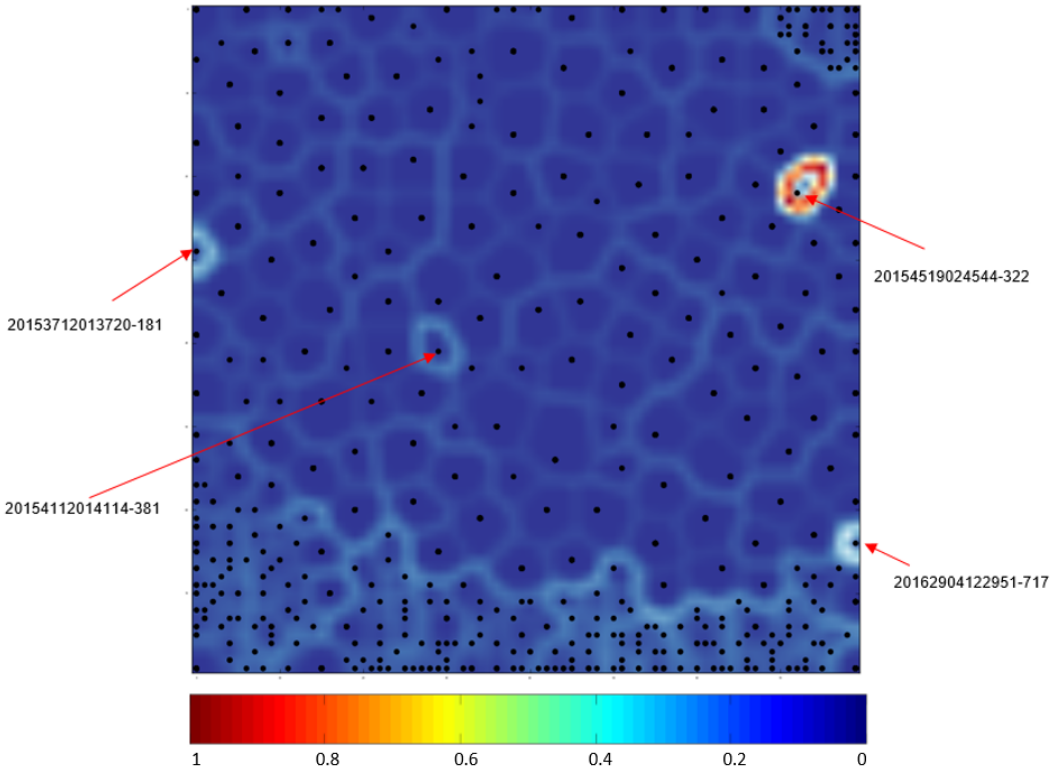

Figure 5.26: U-Matrix result of VAST 2017, there are four outstanding points.

In this experiment, I used the VAST 2017 data set. This task requires us to detect abnormal movements of vehicles that traveling around in a nature preserve. Such patterns would possibly have affected the wildlife. In this task, via using the SOM clustering to cluster the traveling patterns. It is quite obvious to see there are four outliers on the U-matrix map shown in Figure 5.26. The id of these four outliers are 20153712013720-181, 20154519024544-322, 20154112014114-381 and 20162904122951-717. These

four points are all in the distinct clusters especially 20154519024544-322 which have large difference to others.

By further visiting the heat map shown in the Figure 5.27, we can visualize the frequency that those vehicles check-in. 20154519024544-322 and 20154112014114-381 have quite similar patterns that they both spent a lot of time in three ranger gates that are ranger gate 1, ranger gates 2 and ranger gate 5 and two ranger stops that are rangerstop 0 and ranger stop 2. That can also be reflected in the Figure 5.28, the relationship model of their visiting pattern.

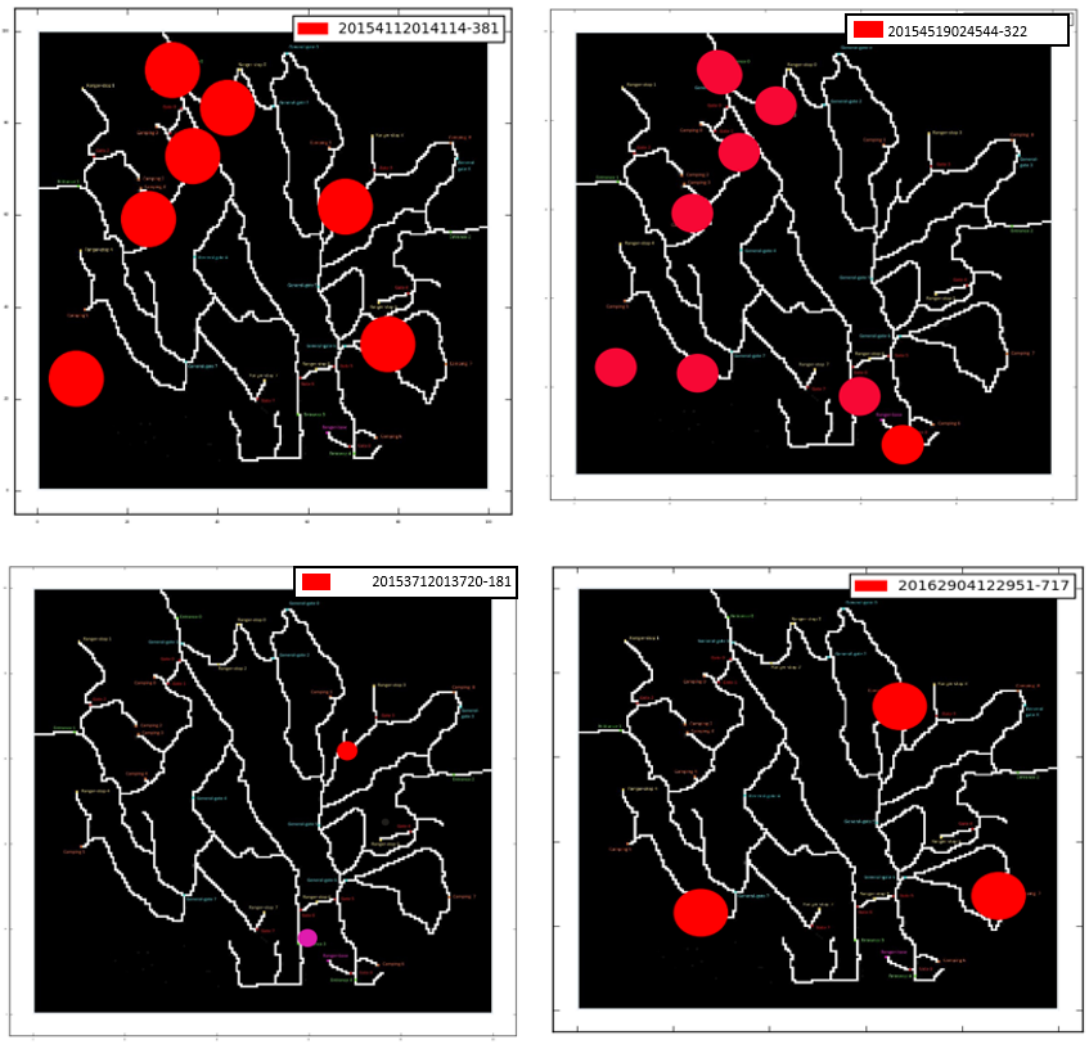

Figure 5.27: Heat map of outstanding visiting patterns.

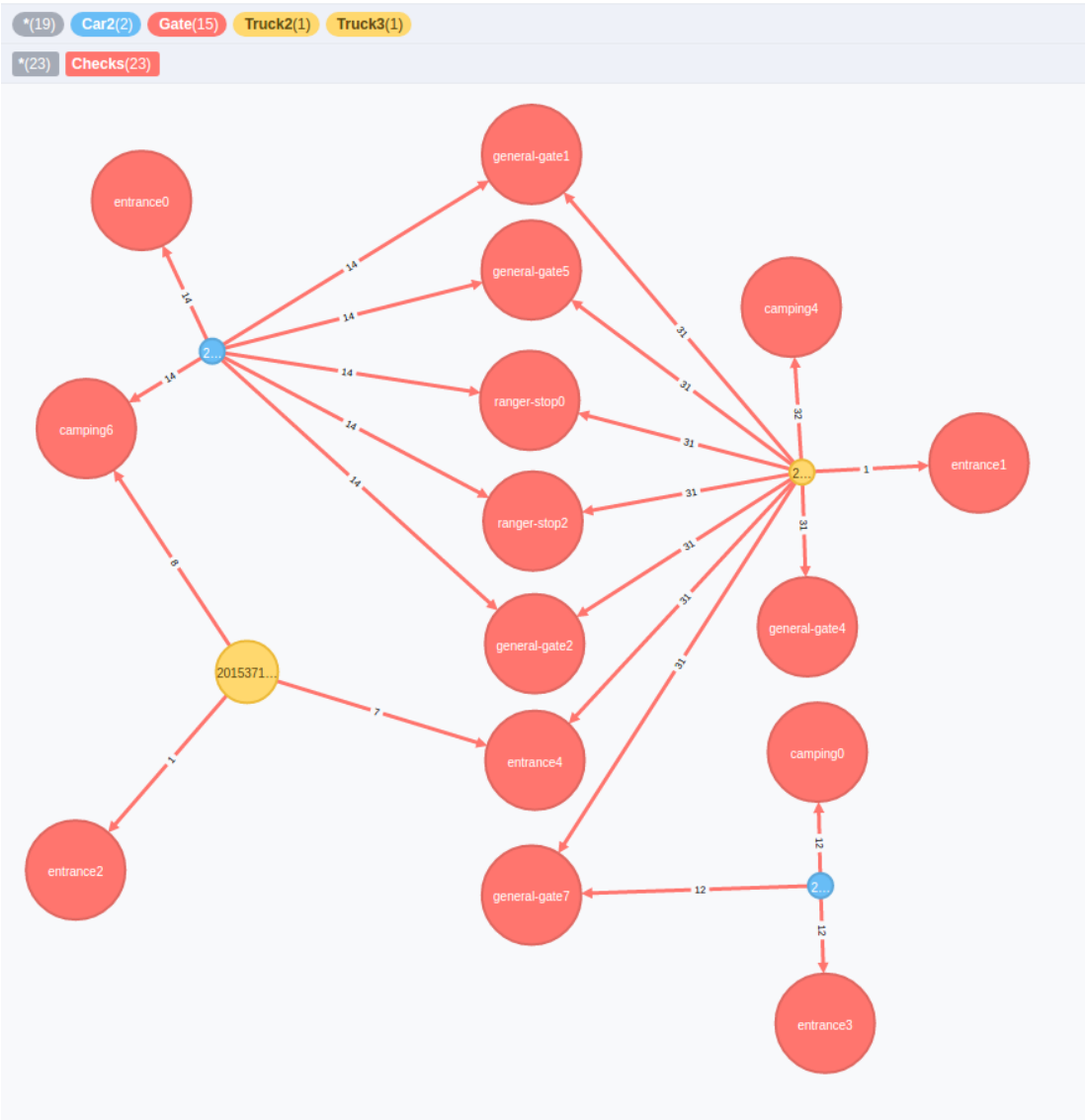

Figure 5.28: Relationship of outstanding results represents different vehicles that are car (blue), big truck (big yellow), small truck (small yellow) checking in locations (red).

The 3D time-cube can clearly show trajectories of these four outstanding patterns. We can see that all of these four vehicles travel around in this nature preserve in a long duration. Especially 20154519024544-322, this vehicle keeps visiting certain locations for 92 days. We can assume this vehicle will have strongest effect on the wildlife as the longest time it spent in this nature preserve. In other words, as this vehicle occurred over multiple days in the nature preserve, the animals around the path it traveled through would most likely to have been affected for a long time. Similar

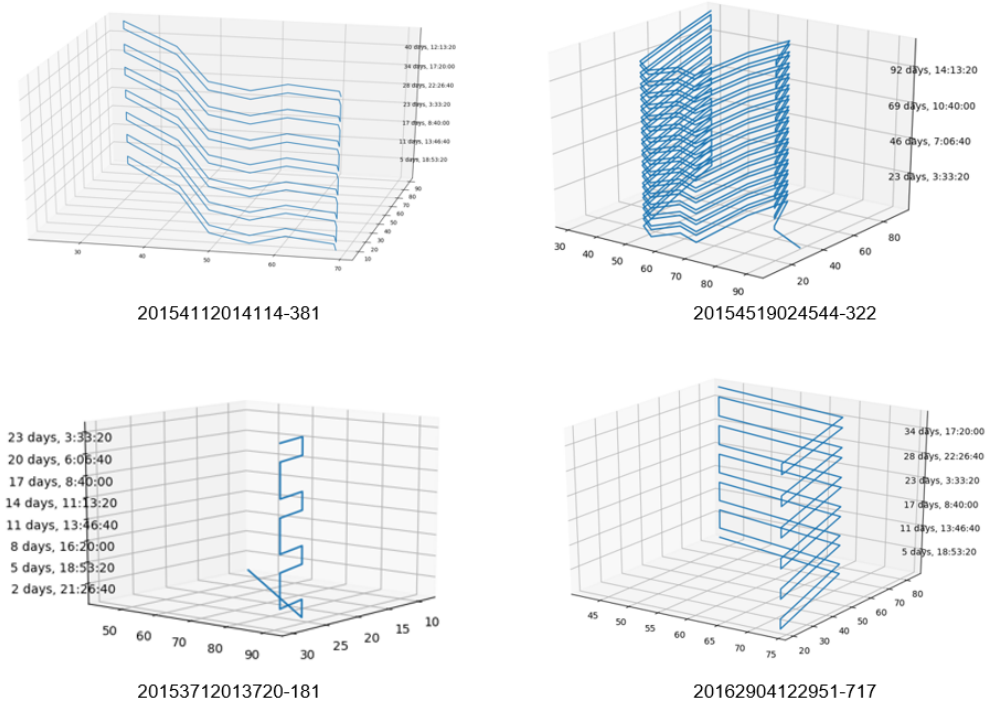

Figure 5.29: 3D Trajectories of outstanding patterns.

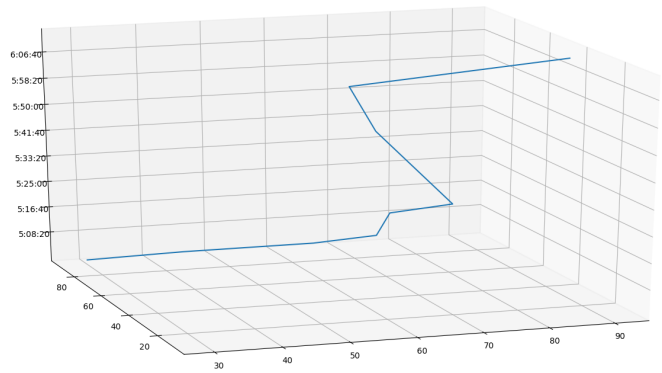

Figure 5.30: 3D trajectory of a regular pattern example.

to 20154519024544-322, other three vehicles spend quite a lot of time on the preserve

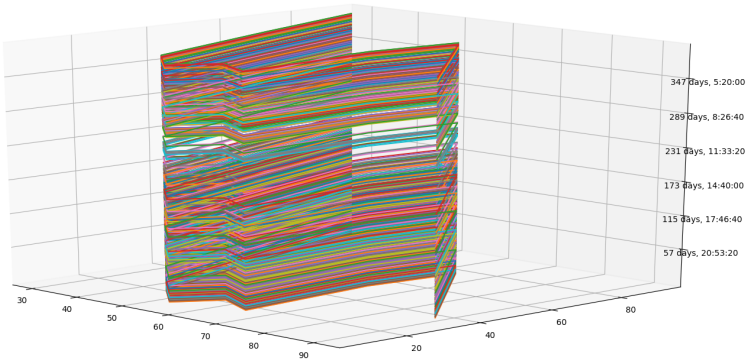

Figure 5.31: 3D cube of popular trajectories, they are highly similar but vary in different time periods. They all start from general-gate 1 and end with entrance 4

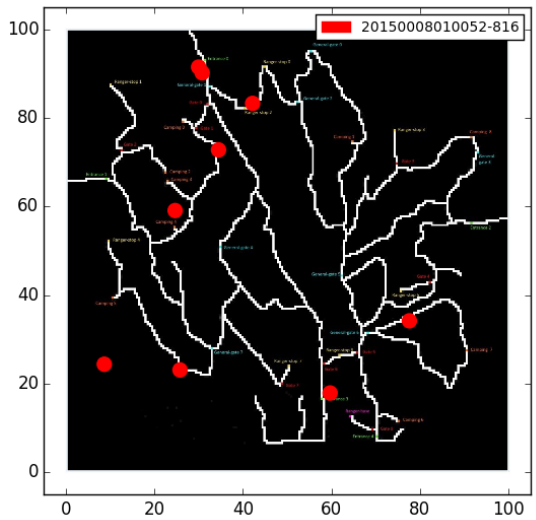

Figure 5.32: Heat map of popular patterns, they are annotated with the heat pots.

for 23, 30, 34 days respectively which are way more longer than other patterns. Correspondingly, they would have much more impact on wildlife than other regular traveling vehicles. To make comparison, the Figure 5.30 shows an example of a regular result which only spent around half day in this preserve. This movement will not have much impact as it stays for not long. Therefore, we can define it as regular movement.

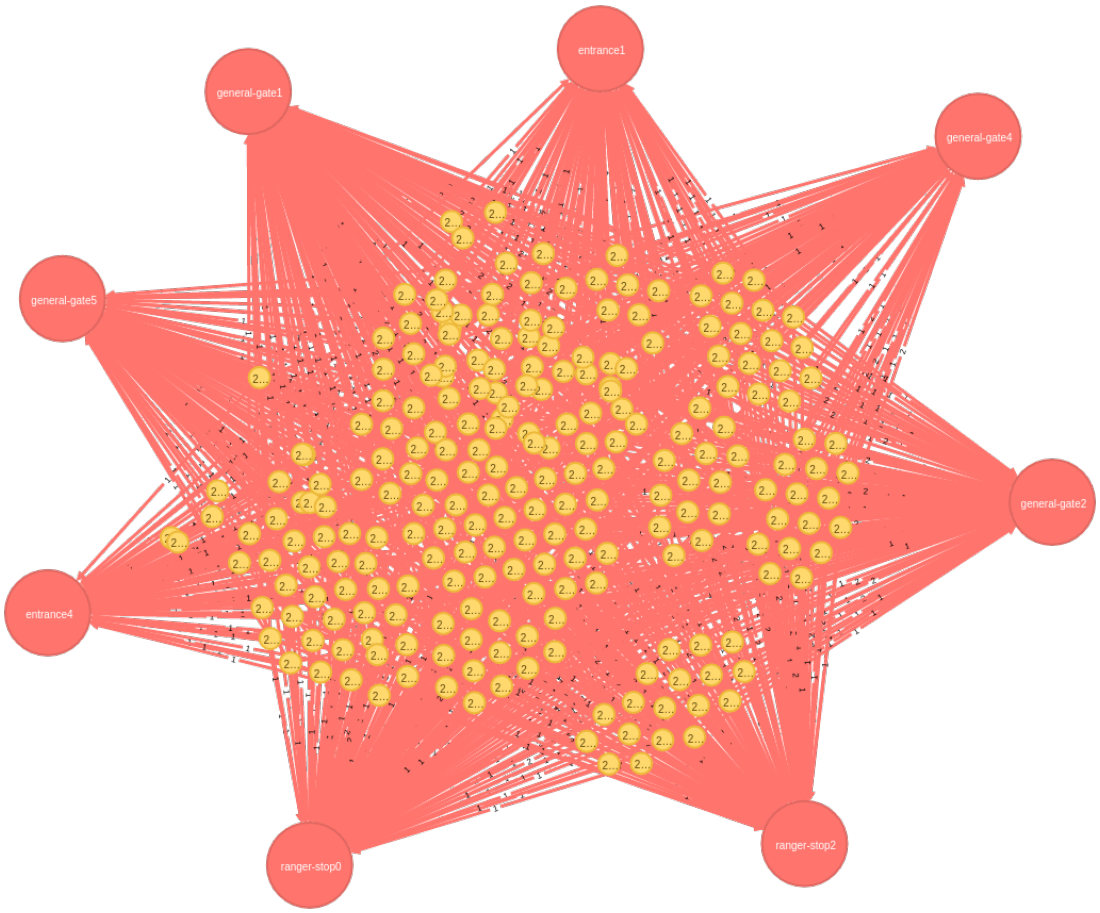

Figure 5.33: Relationship of popular attraction visiting pattern represents trucks (yellow) check in the locations (red).

Another interesting discover is that by inspecting the clustering of SOM, we can also find out some very popular attractions and trajectories. Specifically, there is a large number of individuals have similar patterns which means they all spent quite a lot of time in the same set of attractions as well as their trajectories are highly similar. Here, we will show one example for a popular visiting pattern for trucks.

Figure 5.31 shows an example of a popular movement pattern starting from general-gate 1 ending with entrance 4. From that, we can see even though those vehicles visited those places in different time periods but their trajectories are highly similar and most of them are even exactly the same.

The heat map in Figure 5.32 indicates those location that are popular for trucks while

from the relationship map, showing in Figure 5.33, we are able to visualize that as well.

In summary, experiment results approve that SOM can successfully cluster the patterns using essential features. After clustering, associated visual analytics are also crucial for behavior pattern understanding. Particularly for those three problems, we use those results to resolve crime detection problem, promotion and abnormal behavior.

5.3 Method Summary

Self-organizing Map based clustering is proposed as clustering part for whole method. It helps us find out abnormal visiting behavior patterns .

The main advantage of this method is that there is not parameter we need to set in advance for clustering like K-means and GMM. In that case, we do not have to spend experiment on researching the number of clusters we should set up in advance.

Different from previous work, in our approach, we only use information processed from pure time-spatial data without any other associated data. The other thing is that rather than just rendering the U-Matrix, we concern on interpret the results of U-matrix that we will have deep study of each pattern.

The experiment results show the usability of SOM-based clustering method. For the VAST 2015 data, we successfully find out the crime groups to solve the challenge problems. We have also successfully targeted the outstanding users for commercial promotion in Chinese mobile phone data set. For VAST 2017, we do not only find out the four outstanding patterns which will have strong impact on wildlife but also study the popular patterns using this method.

Overall, SOM-based clustering is proved to be an excellent clustering approach by incorporating different visual analytics methods together with SOM algorithm.

Chapter 6

Deep Learning Based Prediction

It is crucial to know where people is likely to move in upcoming time stamp. By knowing which location that majority of people is about to move, a lot of activities can be taken such as safety management. For instance, urban manager can assign more security guards to that location in case people get injury because of over-crowed. This chapter will introduce a prediction method called Deep Learning based prediction and visualization to resolve the prediction problem.

6.1 Prediction and Visualization

As mentioned in the Chapter 4, we process the time-oriented matrix as dynamic data which indicates where visitor stays at each timestamp. In this chapter, We will use this matrix as training data to forecast the travel flow of visitors. We begin by converting trajectory from continues geographical coordinators (x, y) 2D coordinators into discrete label data by given hot spot labels using Algorithm 1 Location Mapping Algorithm. Consequently, a movement vector will be formed in $(2, 3, 41, 5)$ which means this person visit attraction 2, 3, 41 and 5 in the certain time period. We need to predict the upcoming location this person would be most likely to attend in next timestamp. As we used time-oriented matrix, we are not going to lose any time information which results in better performance than using pure sequence matrix [9]. That is because duration that individual stay in a location is an essential factor of behavior choice [23].

Given an example, if we want to predict which location traveler would like to go at 18:00 pm and we have training time-oriented vector $(2, 2, 2, 3, 2, 41, 41, 2, 3, 5)$ for 10 hours and from 8:00 am to 18:00 pm. We can take vector $(2, 2, 2, 3, 2, 41, 41, 2, 3)$ as training set while 5 as targeted label as visitor is at location 5 at 18:00 pm. Therefore we can de-composite the time-oriented Matrix to be training matrix and labour vector which is shown in Figure 6.1

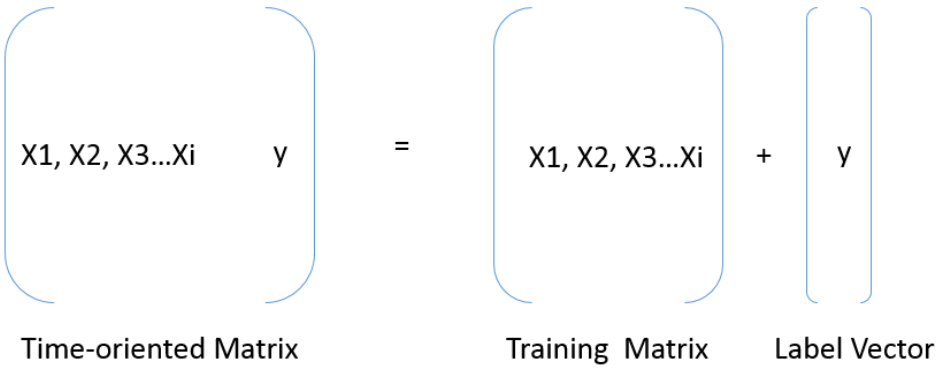

Figure 6.1: Process time-oriented matrix to training data.

Algorithm 7 Prediction and Visualization Algorithm

$T_x \leftarrow$ Time Stamp for prediction
$SM \leftarrow$ Visit Sequence Matrix processed by Original Data

$SM_x \leftarrow$ sequence matrix before T_x
$SM_{x+1} \leftarrow$ sequence vector before T_{x+1}
$T_x \leftarrow$ Training Label Vector obtain by SM_{x+1} - SM_x
$M \leftarrow Predictor(SM_x, T_x)$

$ST_x \leftarrow$ New Visit Sequence Matrix before T_x
for $S_{squencevector}$ **in** ST_x **do**
  $T_{NextLocation} \leftarrow M(S_{squencevector})$
  $T_{PreviousLocation} \leftarrow$ last Location of $S_{squencevector}$
  Output $Visualization(T_{PreviousLocation}, T_{NextLocation})$
end for

The second step is to use a prediction method to generate a model and then input the test data into model to generate the test label which is the predicted result. Afterwards we can visualize the result using several visualization approaches. The algorithm is shown as above in Algorithm 7.

The prediction method is the most crucial part. In this thesis we use Recurrent Neural Net [1] to serve as our predictor as its advantage to deal with time series data. Also the classical classification approaches K-Nearest Neighborhood (KNN), Naive Bayes (NB), Decision Tree (DT), Random Forest (RF), Support Vector Machine (SVM) and Adaptive Boosting (AdaBoost) will be used for evaluation.

6.2 Classification Methods

The classification methods can be utilized as good predictors for our method with good performance. In this section, I am going to introduce some of classical methods. However, as I do not have much contribution on modifying those methods but just simply apply them, so there are not much detail regarding those methods will be given but just briefly introduce them.

6.2.1 K-Nearest Neighborhoods

K Nearest Neighbors [31] is a simple algorithm that can classify data based on calculation of a similarity measured by a distance functions. The distance function can be Euclidean, Manhattan or Minkowsiki. KNN has been used in 1970s as a non-parametric method for pattern recognition and statistical classification. In our case, I set K to be 10 for 10 Nearest Neighbors (10NN).

6.2.2 Adaptive Boosting

Adaptive Boosting (Adaboost) was proposed in 1995 as a method using weak classifiers to generate a stronger classifier. The main idea of this algorithm was keeping a

distribution of weights over the set of instances. Initially, the weight for each instance were equal, however, on each turn, the weight of mis-classified instance would be increased such that the weak classifier is focused on these hard instances [32].

6.2.3 Naive Bayes

The Nave Bayes classifier depends mainly on the Bayes rule:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{6.1}$$

Which can be written as (H for hypothesis, D for data):

$$P(H|D) = \frac{P(D|H)P(H)}{P(H)} \tag{6.2}$$

When an unlabeled instance needs to be classified, the probability for each hypothesis H will be calculated to find the maximum probability.

For $P(D|H)$ which is the probability of data based on the hypothesis, if the values of the data set are continuous, Gaussian distribution can be used to represent the probability:

$$P(x|\mu, \sigma) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \tag{6.3}$$

It should also be noted that, since for each unlabeled instance, the value of $P(D)$ will the same, so during the calculation $P(D)$ can be ignored and it will not affect the results.

Naive Bayes is named Naive because of a strong constraint: all the attributes must be independent from each other, otherwise the result will be inaccurate. Hence, if the attributes of the data set are independent from each other, Naive Bayes is able to produce high prediction performance [33].

6.2.4 Decision Tree

Decision tree is a tree classifier that has nodes with input feature decision. Each internal node will decide an instance into a child node. Each leaf node would be one of the given categories. For each turn, the algorithm would find the most information gained feature as the node to split the training set. If the subset of a node belongs to only one category or most in one category, it will stop the splitting at this node. In order to improve the algorithm, running the tree after the splitting, which removing the brunch that didn't increase the performance. [34].

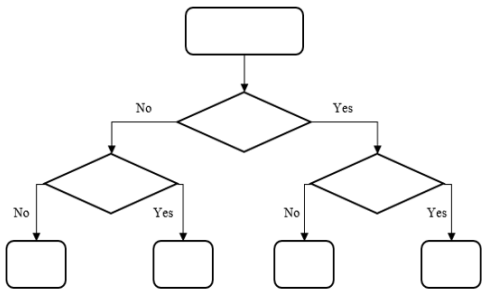

Figure 6.2: Structure of decision tree.

The information gain will calculated by the Entropy function in Equation 6.5

$$E(S) = -\sum_{i}^{n} p_i log p_i \tag{6.4}$$

p_i stands for the probability of each classes. Suppose that dataset S are split into S_1, S_2, ... S_n by some certain criteria, then the information gain is given by

$$Gain(A) = E(S) - \sum_{i=1}^{n} (\frac{|S_i|}{|S|} E(S_i)) \tag{6.5}$$

Therefore, by computing the information gain for each attribute split, we can get a maximum information gain, which is used to split the original dataset into two or more sets. By doing this recursively, the tree will be grow until entropy reaches 0 or the depth reaches a pre-defined number.

6.2.5 Random Forest

Random Forest (RF) is an ensemble learning method consists of multiple decision trees. RF is the variety of bootstrap aggregating (bagging) which sampling from this training set and training on multiple classifiers. After training, the prediction for unseen instances can be averaging the predictions for all individual classifiers. In random forest, the sampling is on features, or called ”feature bagging” [35].

Algorithm 8 Random Forest

```
n_min ← minimum node size
p ← total variables

for b in (1, B) : do
    Z* ← bootstrap samples of size N from training data
    T_b ← random forest tree on Z*
    repeat
        m ← variable at random from p variables
        m_best ← the best variable/split-point among m
        Split into two children node on m_best
        Add tree to T_b
    until reach n_min
end for
Output the ensemble of trees {T_b}_1^B
```

6.2.6 Support Vector Machine

Support vector machine (SVM) is a supervised learning system that uses a hypothesis function with given features to optimize the margin between instances with different categories. For a binary linear SVM, we have a real-valued function (f:X $\subset \mathbb{R}^n \rightarrow \mathbb{R}, f(x) = \overrightarrow{w} \cdot \overrightarrow{x} + b$, where ($\overrightarrow{x}$ is an instance, ($\overrightarrow{w}$ is the weight, and (y is the actual category. If (f (x) ≥ 0, the input belongs to positive category, and otherwise to negative category. In order to classify instances that are not linearly separable, we can use soft-margin with hinge loss function: ($\max\{0, 1 - y_i(\overrightarrow{w} \cdot \overrightarrow{x} + b)\}$, if ($\overrightarrow{x_i}$ lies in the correct field, loss function is zero, and otherwise the loss function value is

proportional to the distance between wrong instance to the margin. Therefore, SVM are aimed to minimize Equation 6.6.

$$\frac{1}{n}\sum_{i=1}^{n} max\{0, 1 - y_i(\overrightarrow{w} \cdot \overrightarrow{x_i} - b)\} + \lambda \left\| \overrightarrow{w} \right\|^2 \tag{6.6}$$

However, it can be transferred to dual problem by maximizing Equation 6.7.

$$f(c_1, ..., c_n) = \sum_{i=1}^{n} c_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} c_i \psi(\overrightarrow{x_i}) \cdot \psi(\overrightarrow{x_j}) y_j c_j \tag{6.7}$$

with $(\sum_{i=1}^{n} c_i y_i = 0$, and $(0 \leq c_i \leq \frac{1}{2n\lambda}, \forall i$. After maximization, the weight and bias could be calculated as $(\overrightarrow{w} = \sum_{i=1}^{n} c_i y_i \psi(\overrightarrow{x_i})$ and (b = $1\frac{}{n\sum_{i=1}^{n}(\overrightarrow{w} \cdot \psi(\overrightarrow{x_i}) - y_i)}$. The category of new instance could be calculated as Equation 6.8.

$$\overrightarrow{z} \rightarrow sgn(\overrightarrow{w} \cdot \psi(\overrightarrow{z}) + b) \tag{6.8}$$

In this method, we chose rbf kernel shown in Equation 6.9:

$$rbf(\overrightarrow{x_i}, \overrightarrow{x_j}) = \exp(-\gamma \left\| \overrightarrow{x_i} - \overrightarrow{x_j} \right\|^2) \tag{6.9}$$

In order to deal with multi-class classification, the system has used one-vs-one schema using $k(k-1)/2$ classifiers to classify k categories, where each classifier was able to classify two of k categories. After computations, each classifier would output one category, where the most voted category would be the predicted category [36].

Overall, those classification methods can serve as base classifiers for evaluating the model. For certain tasks, they have outstanding performance. There is not much contribution for enhancing those algorithm therefore this section only briefly goes through those methods without detail explain. In the Evaluation part, we are going to compare them with Deep Learning method and we can see their performance.

6.3 Recurrent Neural Net

Recurrent Neural Net (RNN) [1] is introduced to resolve the time-oriented data. Such data is the vector that every dimension would have dependency on the previous one. In this research, RNN is utilized for the prediction task. With the help of dynamic architecture of RNN, this algorithm can handle the sequence data in an efficient way. In this section, I will explain my design choice of the architecture and the one methodology that solve the vanish gradient problem called Long-short Term Memory.

6.3.1 Architecture

The feature of RNN is that this network contains feed-back connection, therefore activation can flow round in a loop which makes it able to do the temporal processing and learn sequences. Such feature guarantees that RNN can have excellent performance in time series data. In our research, we utilize Three-hidden-layers Architecture for RNN which is a classic architecture of choice and each hidden layer manages loops that can achieve dynamic updated and training based on timestamp t_i.

Like other neural nets, RNN has forward activation and back propagation for training. Specifically, forward activation in RNN is using dynamic feed-forward process that result output from neural depends on the sequence of data ordering with t_0 to t_i.

$$h_t = \theta\phi(h_{t-1}) + \theta_x x_t \tag{6.10}$$

$$y_t = \theta_y \phi(h_t) \tag{6.11}$$

where θ represent for the weights associated and ϕ the **Tanh** forward activation is taken as our activation function ϕ in our method. Also back-propagation in RNN takes the time with essential parameters and we called it **Back-Propagation through time**, the formula of it can be presented as follow:

$$E_{total}(t_0, t_1) = \sigma E_{sse/ce}(t) \tag{6.12}$$

$$\Delta w_{ij} = -\eta\sigma - \frac{\delta E_{sse/ce}(t)}{\delta w_{ij}} \tag{6.13}$$

The usage of RNN in this task is similar to apply RNN in the Temporal Context problem. Therefore in that case, the length of time interval will have strong impact for predicting. Theoretically longer time period will give more information for recurrent network. That is to say with the lager timestamp we will get larger set of training data and relatively higher accuracy (i.e. accuracy will be improved with lager timestamp.) To reveal how time period affect the accuracy we analyse the prediction accuracy among different timestamps.

6.3.2 Long-Short Term Memory

Long Short-Term Memory (LSTM) [1] is a technique to solve the **vanishing gradient problem** which occurs as time steps and each layer of architecture related to each other through multiplication, derivatives would have potentially vanish. The work flow of LSTM is illustrated with Figure 6.3. LSTM sets up the output gating ,output squashing, memorizing, forgetting, input gating and squashing. It would be useful to avoid such error that can be backpropagated through time and layers. They allow recurrent nets to continue to learn over many time steps without the potential vanished occurring by keeping a more stable error.

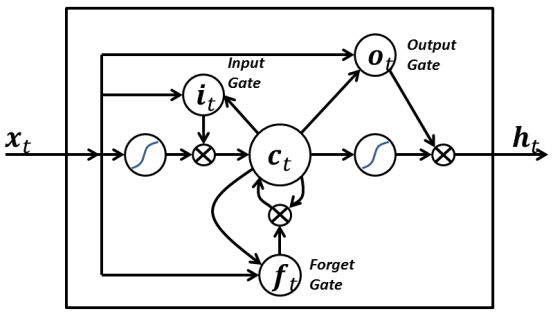

Figure 6.3: Architecture of Long-short Term Memory [1] with input gate, output gate and forget gate.

6.4 Experiment and Evaluation

To obtain predictor model, we input the time-oriented data for training. Such model would be used to predict upcoming location for the next time stamp. Specifically, for every single visitor, once next stay point is predicted, we are able to visualize the check-in pattern in the next time period. It is crucial to understand where people is likely to check-in such as which station user will use in the next hour, which attraction most visitors will attend in next timestamp and which path most traffic will travel through. Taking park management as an example, by knowing which location that majority customers is about to move in the next timestamp, more safety management can be done with corresponding hot spot. For instance, the manager can assign more security guards to corresponding location for guiding and helping attendees. Another example can be the mobile station usage, if we are able to predict the usage of station in coming timestamp, the advertisements can be put onto the corresponding station for commercial promotion and more maintaining works can be concerned on to avoid the technical problems occurring.

In this section, we are giving three use cases to conduct the prediction empirical experiments for the travel flow prediction for VAST 2015 data, station usage prediction for mobile data and traffic forecast for Vast 2017. There are three main results we want to see in the experiments. First of all, the result evaluation for RNN and other methods. Secondly, the change of accuracy among different training time. Lastly, the visualization of prediction result rendered using visualization tools such as flow map.

In terms of empirical experiment, we have set up the classic classification methodology including Decision Tree, Random Forest, K-Nearest Neighborhood and Support Vector Machine. The most frequency and uniform are took as the baseline. We set the same parameter for each classifier. Specifically, for KNN, we set the K to be 10. For random Forest the number of estimators (trees) is set to be 100 and the max depth is set to be 2. Depth of Decision Tree is 5. We also use Mutinomial Method for Naive Bayes as the value (label) of sequence vector is discrete. For Support Vector Machine, we choose radial basis (rbf) function kernel (Guassian Kernel) as Kernel Method. The design choices can be seen as following table.

Classifier	Parameter
KNN	K = 10
SVM	Kernel = RBF
DT	max Depth = 5
RF	max Depth = 2, n_{tree} = 100
NB	method = multinomial

Table 6.1: Evaluation Classifier Parameter

6.4.1 Travel Flow Prediction for Visual Analytics Science and Technology 2015

The first experiment is performed with the data from VAST 2015. In this section, we will evaluate the accuracy of RNN with other methods and visualize the prediction result of RNN as well as others. Specifically, the location that people would visit in the upcoming timestamp would be visualized. By comparing different methods, we find the RNN is the most optimal solution. Figure 6.4 shows that RNN achieves the highest average prediction accuracy with 46% while the second highest score 39.4% achieved by Most Frequent predictor.

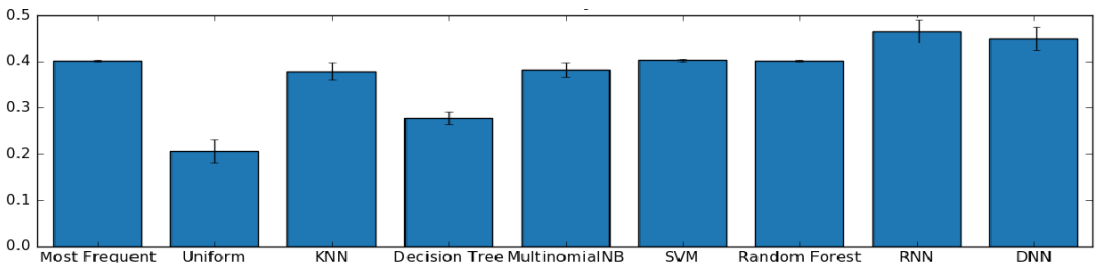

Figure 6.4: The prediction evaluation for VAST 2015, y-axis is accuracy, x-axis is method. The accuracy values from left to right are: 39.4%, 19.2%, 38.1%, 28.2%, 38.5%, 39.2%, 37.9%, 46.0% (Recurrent Neural Net), 42.3% (Deep Neural Net)

Except the average accuracy, we also analyze the prediction accuracy among different timestamp. As we utilize the movement data before certain timestamp T as training set to predict movements at T. Therefore, theoretically, with the lager timestamp we will obtain larger set of training data. In other words, ideally, the accuracy will be improved with lager timestamp. Figure 6.5 shows how different methods perform prediction with different timestamp. RNN has gained almost all highest score among all

the time, and increases significantly particularly after 20:00 pm.

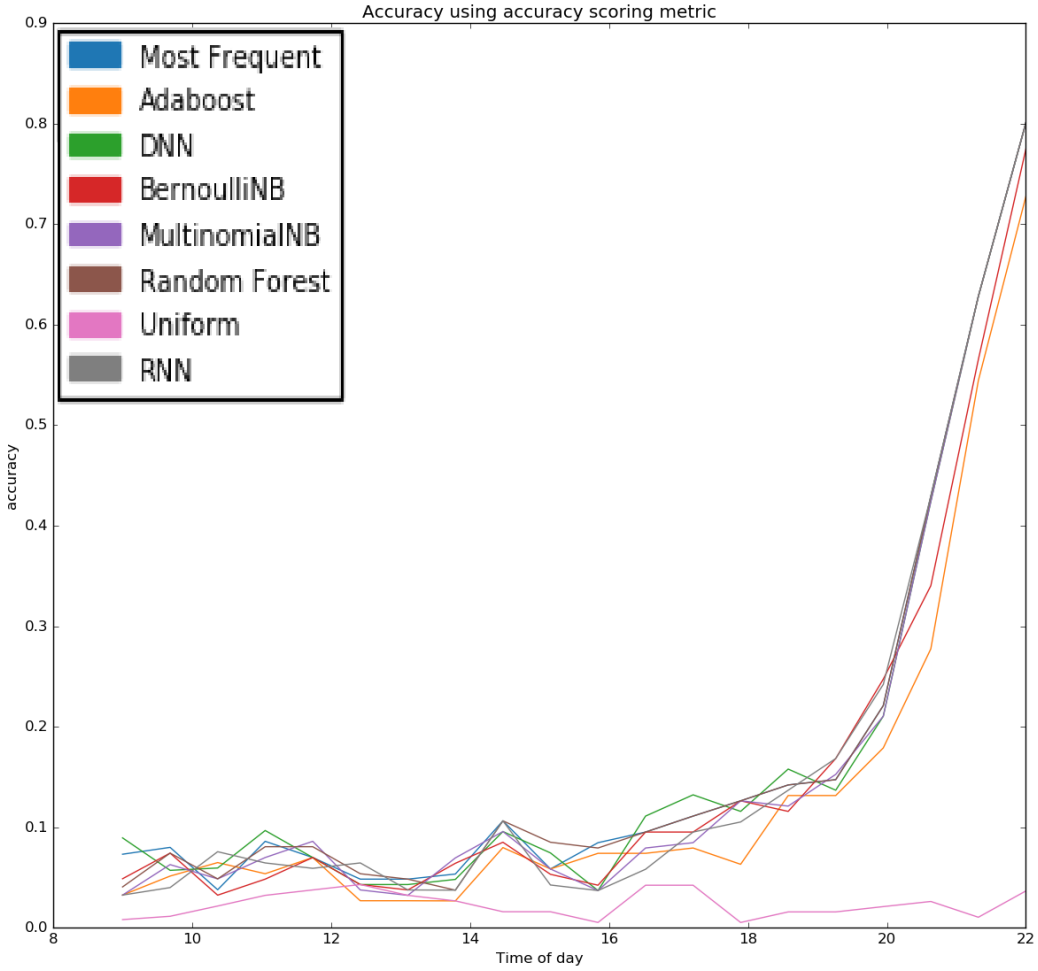

Figure 6.5: The prediction accuracy evaluation among time for VAST 2015, x-axis is predicted time set divided by 2 hours, from 8:00 am, 10:00 am, 12:00 am, 14:00 pm, 16:00 pm, 18:00 pm, 20:00 pm and 22:00 pm. Y-axis is the accuracy.

Figure 6.6 is the actual movement of VAST 2015 data from 13:00pm to 14:00 pm Saturday, 08 June 2014 while Figure 6.7 is the visualization result of RNN prediction of movement in the same period. From that two plots, it is quite clear to see that RNN is able to outline the majority movement of visitors. In this example, RNN can predict the trajectory from attraction 13 to attraction 81 which is the most important movement among this time period. In addition for other movements, there are some slight errors but mostly we can see the prediction is very successful.

Other prediction results are also revealed in the Figure 6.8. From those results, we can see the prediction visualizations of other methods do not really perform well in this experiment. They show a lot of difference from the actual movements.

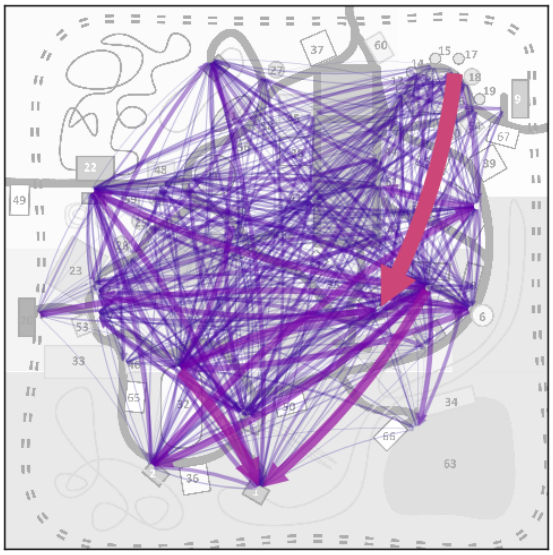

Figure 6.6: The actual movements for VAST 2015 from 13:00 pm to 14:00 pm, Sat, 07 June 2014, arrows indicate movements in one hour.

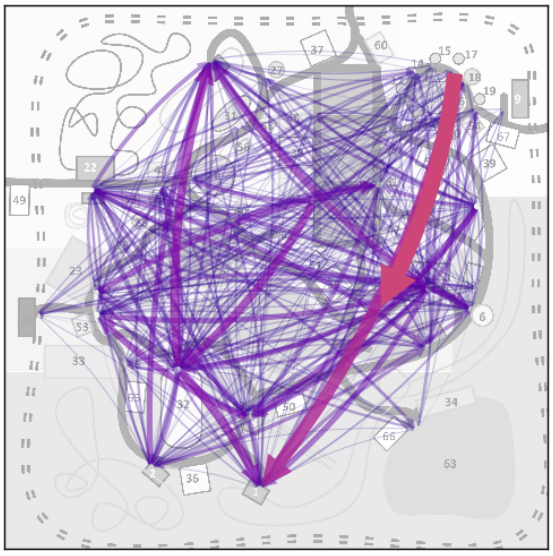

Figure 6.7: The RNN prediction movements for VAST 2015 from 13:00 pm to 14:00 pm, Sat, 07 June 2014, arrows indicate predicted movements in one hour.

The accurate prediction for is essential for the safety-oriented park management. By

knowing which attraction could possibly be crowded and most visitors will move to, the corresponding security forces can be assigned to the path and attraction to help attendees in case any accidents happen. In this case, as illustrated from prediction result, park manager should ask the security guards to the path along attraction 13 to attraction 81 to help customers as this path will be traveled through by most visitors.

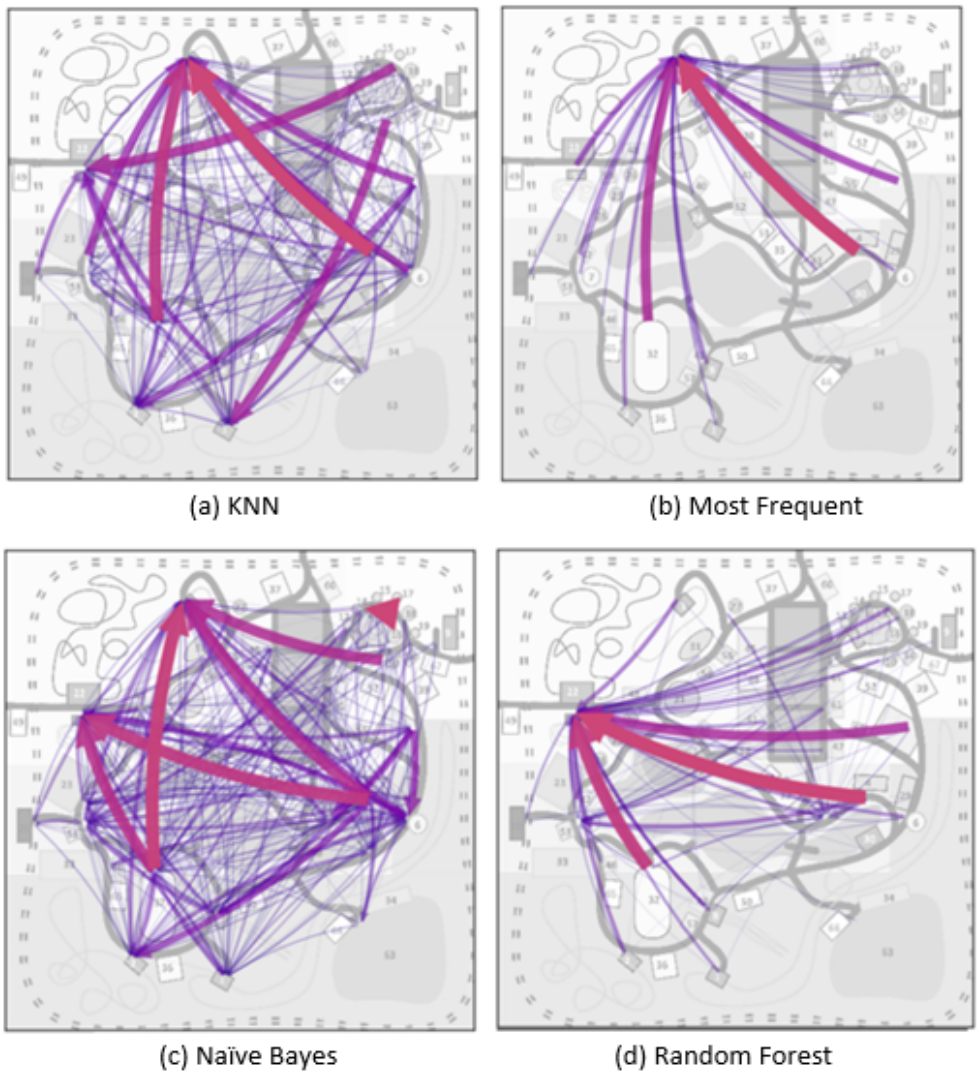

Figure 6.8: Other prediction for VAST 2015, at 14:00 pm Sat, 07 June 2014.

6.4.2 Station Usage Prediction for Moblie Data

In this experiment, we use our method to predict the usages mobile station in the upcoming timestamp and then visualize it. Data set represent that users use mobile stations to visit the IP address.

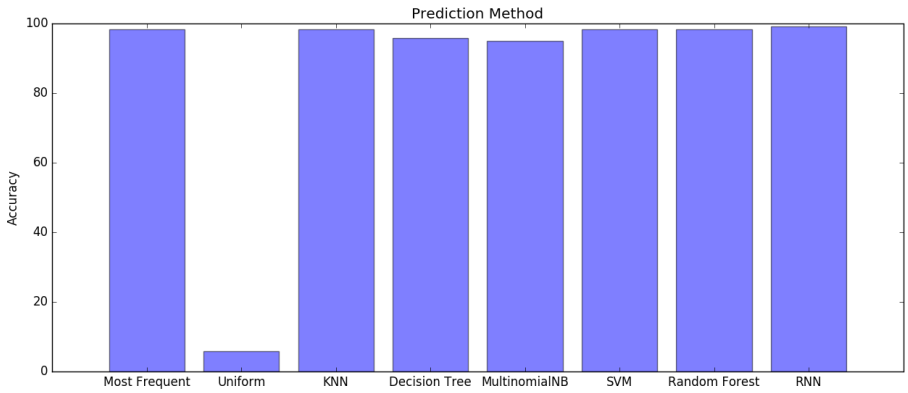

Figure 6.9: Evaluation for mobile data, y-axis is the accuracy. The accuracy values from left to right are: 97.2%, 6%, 97.5%, 93.2%, 92.4%, 90.5%, 96.2%, 96.4%, 98%

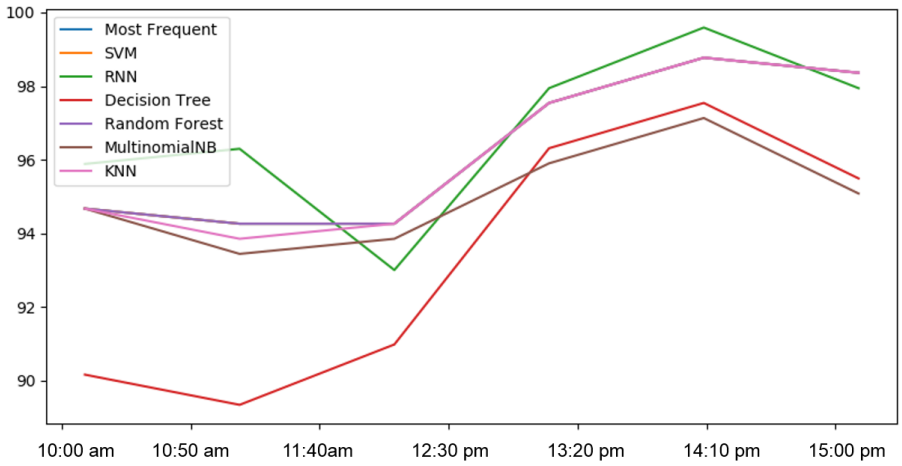

Figure 6.10: Prediction Evaluation for mobile data through time, x-axis is divided by 50 minutes, starting from 10:00 am ending with 15:00 pm (10:00 am, 10:50 am, 11:40 am, 12:30 pm, 13:20 pm, 14:10 pm, 15:00 pm), y-axis is accuracy.

Figure 6.9 shows the prediction accuracy among the different methods. From the result, except the uniform method, other methods achieve very outstanding performance. The highest one is still obtained by RNN with 98% following by KNN, most frequency, RF and SVM with 97.5%, 97.2%, 96.4% and 96.2%. It is also interesting to see that, in this data set, most methods can obtain excellent performance.

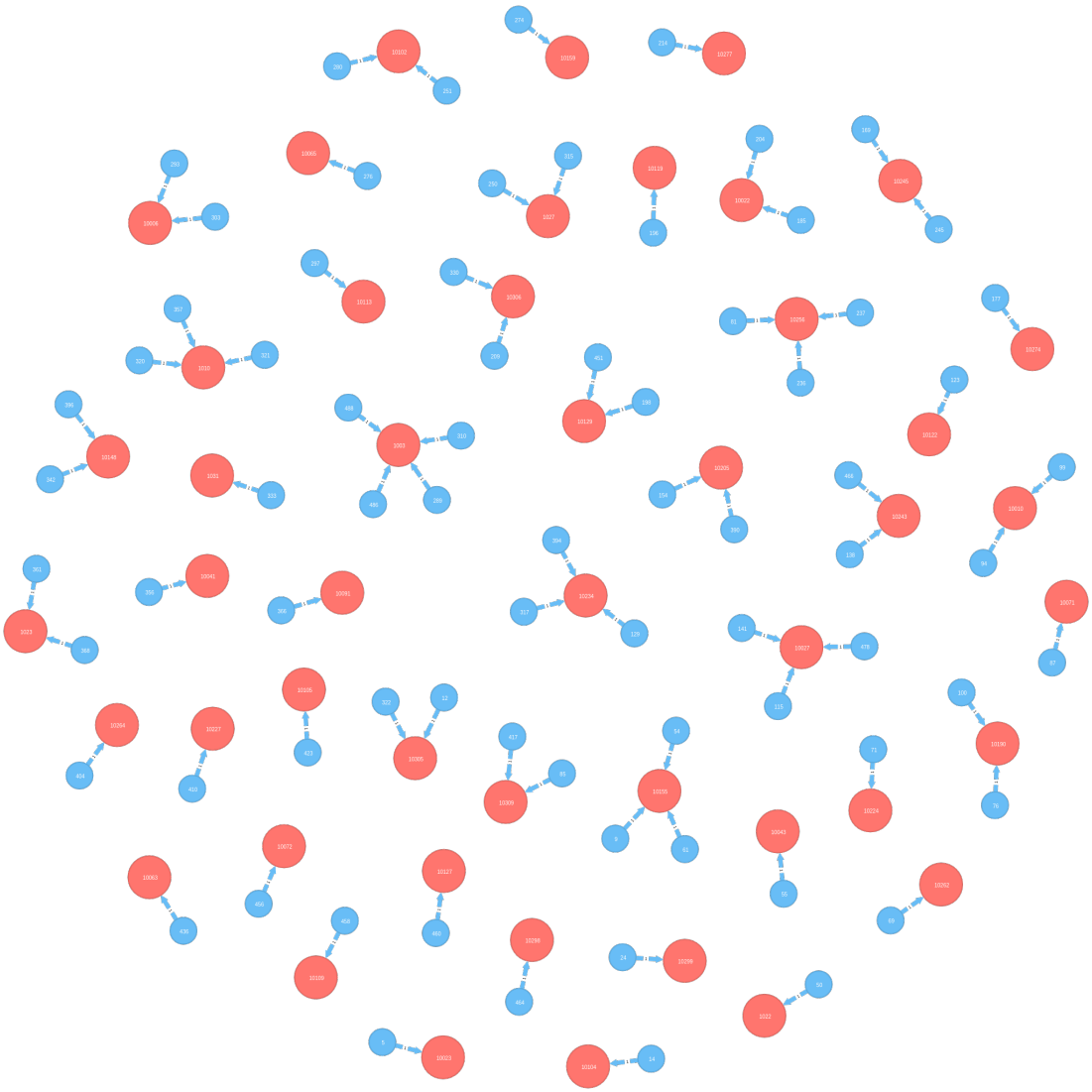

Figure 6.11: Actual usage of mobile station at 17:00:00 pm, 14-May-2015. This pattern represents user (blue) uses station (red)

In Figure 6.10, we can see the accuracy changes through time. RNN (indicated by green line) achieves mostly the highest result among all the time. Especially, when the

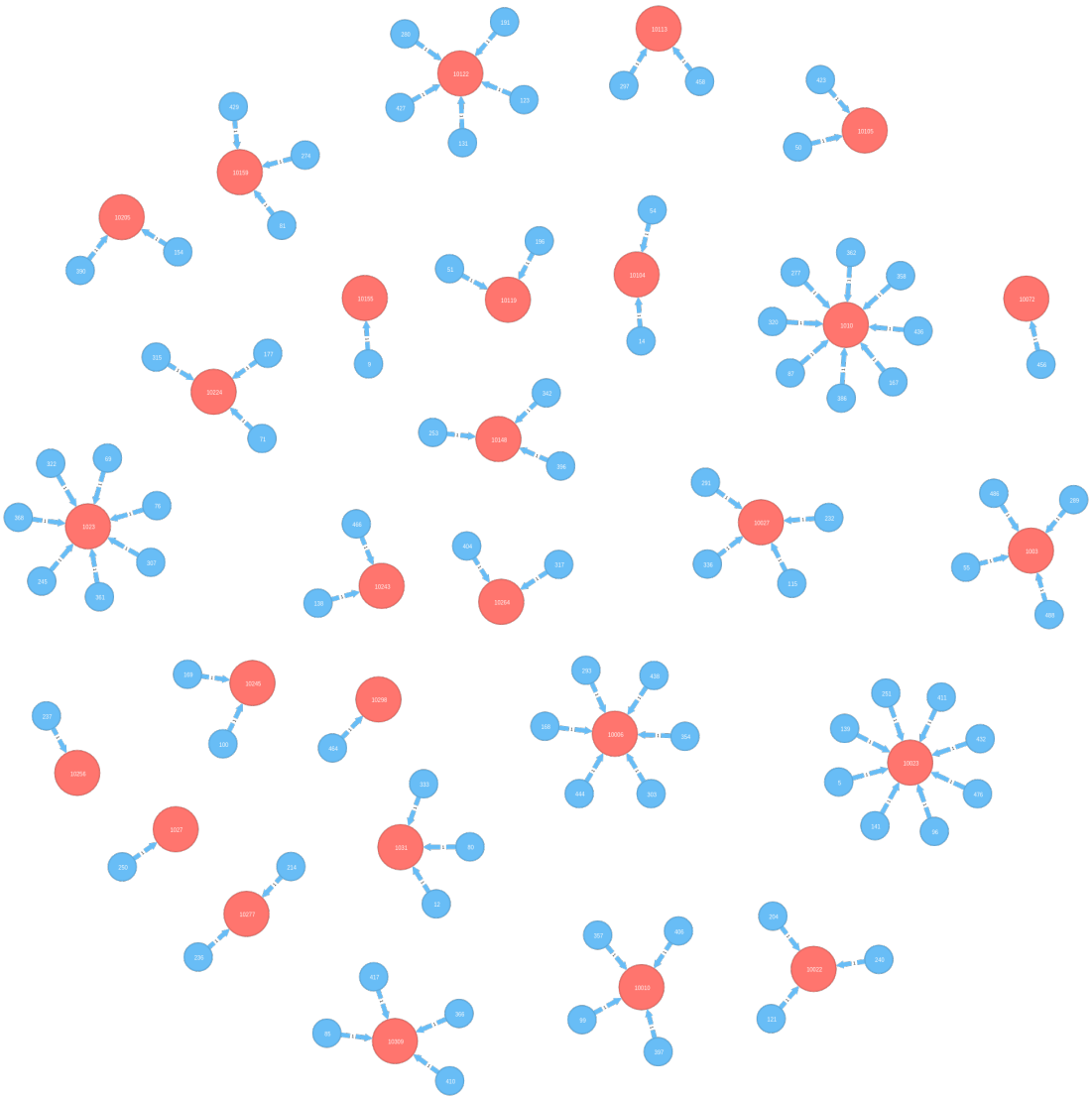

Figure 6.12: Predicted usage of mobile station at 17:00:00 pm, 14-May-2015. This pattern represents user (blue) uses station (red).

training set is 300 minutes (5 hours), RNN becomes the best predictor. Also besides RNN, we can see that Decision Tree is also a excellent predictor in this task.

Figure 6.11 shows the actual usage of mobile stations at 17:00 pm, 14 May, 2015 while Figure 6.12 shows the predicted usage pattern of stations at the same time. From these two figures, we can see that RNN can mostly forecast the usage even with some slight mistakes. In this task, by successfully predicting the usage pattern of stations, we can provide suggestions to the managers. For instance, which station may have a large

number of users in next hour, therefore more technical support should be concerned on. For the commercial promotion sides, once we know which station will be busy in next hour, commercial company can put advertisements onto that station.

6.4.3 Traffic Forecast for Visual Analytics Science and Technology 2017

In this experiment, we use RNN prediction to forecast the traffic movement for VAST Challenge 2017. Similar to previous two tasks, we will have two experiment evaluations that are the evaluation for average prediction accuracy result and the accuracy through time respectively. In addition, the prediction result visualization of RNN and other methods will be shown.

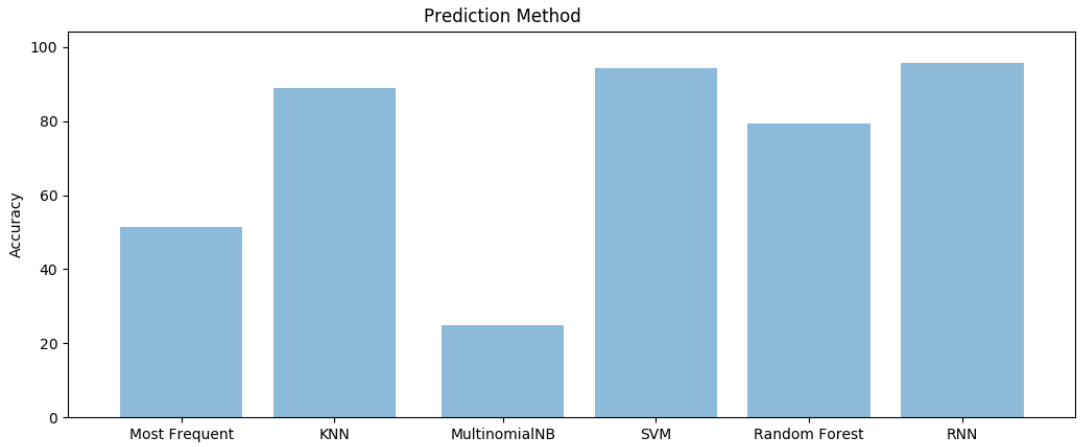

Figure 6.13: Prediction evaluation for VAST 2017, y-axis is accuracy. The accuracy values from left to right are: 52.3%, 84.2%, 21.2%, 87.6&, 79.0% and 92.3%

From Figure 6.13, we see that RNN get the most outstanding performance with 92.3%, other three methods SVM, KNN and Random Forest get 87.6%, 84.2% and 79% respectively for the second, third and fourth outstanding performance. Overall, for this data set, the average accuracy for most methods are quite good.

The result of accuracy through time (Figure 6.14) is quite interesting. Ideally, we assume that, with larger training set, the accuracy should increasing. Though in this

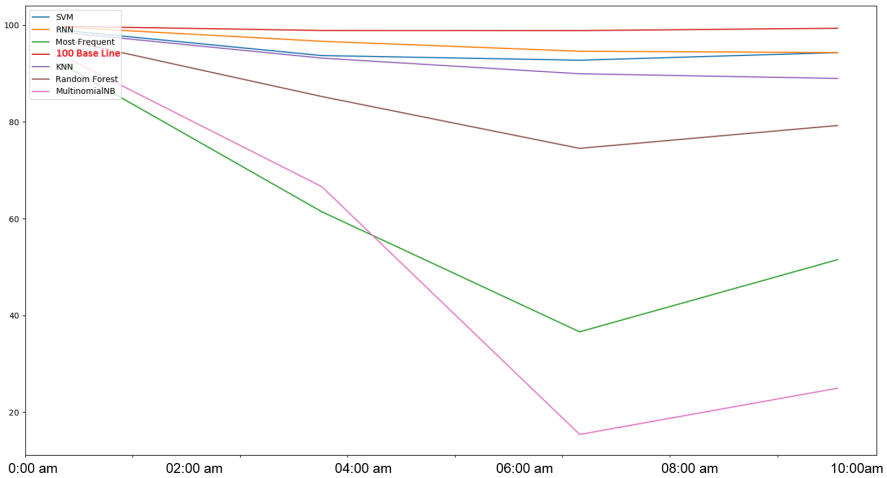

Figure 6.14: Prediction accuracy through time for VAST 2017, x-axis is divided by 120 mins from 0:00 am to 10:00 am. They are 0:00 am, 2:00 am, 4:00 am, 6:00 am, 8:00 am and 10:00 am. Y-axis is accuracy.

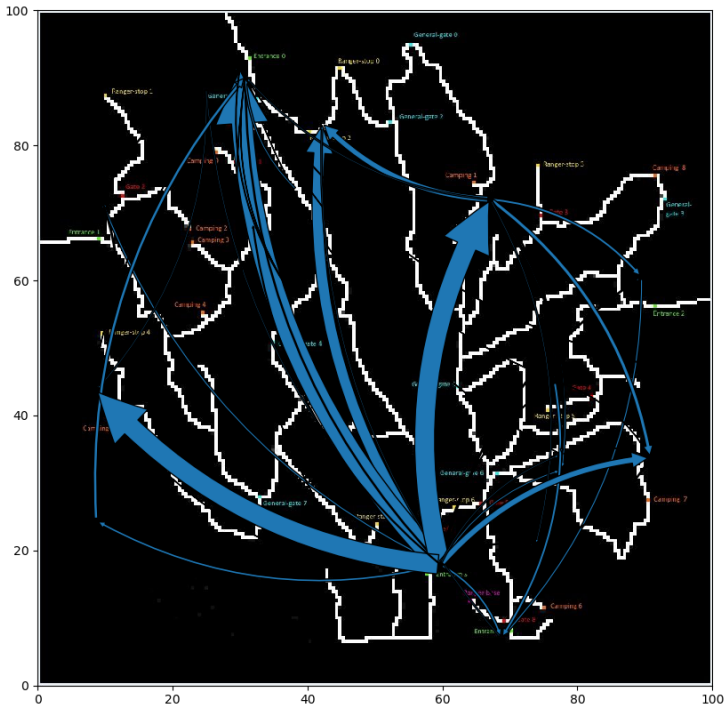

Figure 6.15: The actual movement for VAST 2017 from 19:00 pm to 20:00 pm from 01 Jan 2016 to 29 May 2016, arrows indicate movement in 1 hour.

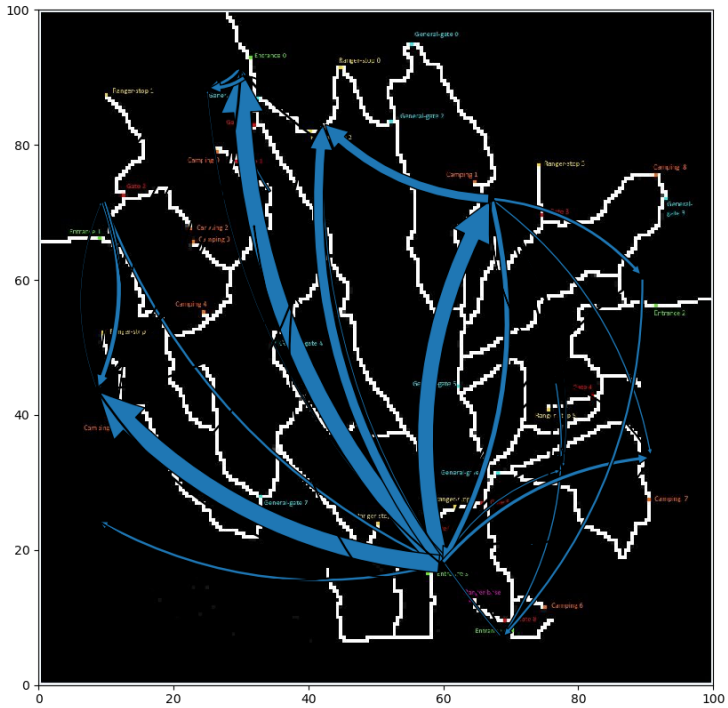

Figure 6.16: The RNN predicted movement for VAST 2017 from 19:00 pm to 20:00 pm from 01 Jan 2016 to 29 May 2016, arrows indicate movement in 1 hour.

result, it illustrates an opposite consequence that the accuracy actually decreases with the increasing size of training data. Particularly for Random Forest, Multi-Normial Naive Bayes and Most Frequent method, they all get a significant decrease. Even though RNN get slightly decrease among the time as well from the graph, we can see it still serves as the best method with the highest accuracy among all time.

Figure 6.15 shows the actual traffic movement from 19:00 pm to 20:00 pm in period from 01 Jan 2016 to 29 May 2016 while Figure 6.16 shows the RNN predicted movement in the same time period. In this task, I use the data set from 01 May 2015 to 31 Dec 2015 from 0:00 am to 20:00 pm movement as training data to predict the attendance at 20:00 pm from 01 Jan 2016 to 29 May 2016 . In this task, RNN can successfully outline the four majority trajectories that all start from General Gate 6 but to Camping 5, Ranger Stop 2, General Gate 1 and Camping 1 respectively. Not just these four majority parts, we can see that RNN is also able to predict the most of trajectories in the right way from the result.

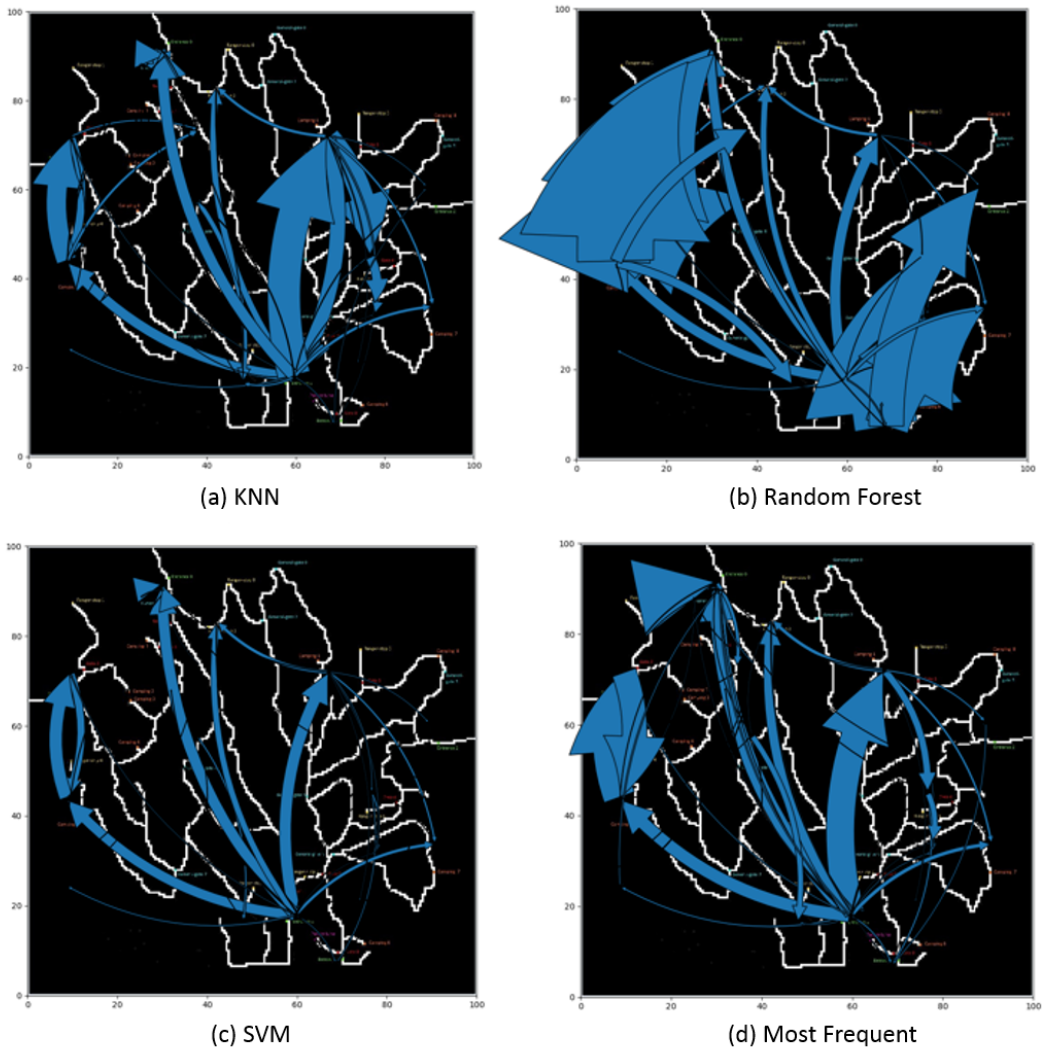

Figure 6.17: Other predictions for VAST 2017 at 20:00 am.

For evaluation, we also plot the trajectory prediction visualizations for the methods obtaining the top four best results that are KNN, Random Forest, SVM and Most Frequent. From the results, we can see that all of these four methods are able to predict the four majority paths. But for other paths, we see that KNN and SVM can have very good results but Random Forest and Most Frequent obviously fail the task.

In this task, we try to forecast the traffic movements that would probably have effects on wildlife therefore we can provide suggestions to the nature preserve manager. For

instance, in this case, we know the wildlife in following four paths among Gate 6 to Camping 5, Generate gate 1, Ranger Stop 2 and Camping 1 respectively will get most effects from 19:00 pm to 20:00 pm. As it is the night time, so that the night time wildlife in those four paths are going get most effects based on our prediction. According to our discover, we will suggest the manager to take corresponding activities to help the wildlife among these four paths so that mitigate the effects as much as possible.

6.5 Method Summary

Deep Learning-based prediction method is proposed to forecast the visiting pattern in upcoming timestamp. The main predictor algorithm is Recurrent Neural Net.

Taking the advantage of dynamic algorithm of RNN, this method is efficient for time-series data type. The performance of approach can be seem from our empirical experiments for all three data set. RNN achieves highest average accuracy for all three data. In addition, for performance among time, RNN shows outstanding results comparing to other classification methods. Therefore, it approves that RNN is a very good approach for prediction part for our framework.

Our works mainly focus on visualizing the prediction results and apply them on different areas such as urban management and commercial behavior. Hence, even though we have utilized the methods including KNN, SVM, Decision Tree, Random Forest and so on, we do not try to contribute on modifying those algorithms. Instead, we optimize parameters for better using those methods. Particularly for RNN, we choose the three-layer architecture and Tanh as activation function. From those excellent results, we can see those optimization and choices of design contribute strongly to prediction performance . Overall, Deep Learning-based prediction serves as an useful approach.

Chapter 7

Conclusion

We will close this thesis by summarizing the framework, result and future work of this method. In this chapter, we will firstly recap the whole framework of this approach, which covers three main points of Data Model (Chapter 4), SOM clustering (Chapter 5) and RNN Prediction (Chapter 6). And then list the advantages of them. Next part is to summarize the experiments we have preformed with VAST 2015, Mobile data in China and VAST 2017 data set. Finally a we will foresee some future works that can be further implied to enhance this method.

7.1 Approach

This section will recap the whole framework we set up for this method. It includes three main parts that back-end data model, two important feature extraction algorithms and the front-end visual analytical interface. Data model is based on the static relational model which stores all data set. Once user query the time and locations from static data model, the dynamic database will be produced. Such temporal data will be piped into the SOM and RNN algorithms for clustering and prediction respectively. After clustering result U-Matrix and and prediction result flow map are generated, user can further interpret those results with different visual analytics tools such as heat map, 3D cube and so forth.

The most significant advantage of this framework over conventional methods is that

this method focuses more on the feature extraction that are clustering and prediction. Such results will indicate essential information to user such as behavior pattern. Moreover, using neural net rather than simply constructing a visual analytics system, users do not have to conduct expensive experiments and suffer from massive information because of large time-spatial data set. In other words, this approach would be especially efficient to deal with massive data set.

In terms of feature extraction algorithms, we applied two neural nets SOM and RNN for clustering and prediction respectively. In terms of SOM, the contribution is mainly on the post-training visual analytics. Most of previous works only render the clustering result without any deep study of those clustering patterns though they could be essential. In our work, we concern on analysis of those outstanding clusters to study their patterns in order to perform tasks such as abnormal visitors detection. In addition, for RNN, we use this dynamic Neural Network to complete our prediction task. This method is inspired from deep learning prediction method [9, 23]. In our research, rather than simply predict the upcoming locations, we use this method to visualize the future movement flow. Such visualization can be applied for many fileds including commercial promotion tasks. Overall, the methods proposed from this thesis are approved that can used in the experiments, which I will give a detail in the next section.

7.2 Experiment

In our experiment, we tested three different data sets that are VAST 2015 challenge data, mobility data in China and VAST 2017 challenge data. For VAST 2015 challenge and 2017 challenge data, they are both public set with clear tasks and aims. Therefore, the results are clear with the detail of each identity information. However regarding mobility data, as the confidential reasons, only some of information we found out are selected to be shown in this thesis.

In terms of result, we successfully detect the crime groups in the VAST 2015 Challenge and predict the traveling flow in a relatively high accuracy. With the mobility data, we can visualize and analyze to understand check-in user patterns via the clustering. Those

results would be helpful for promotion to certain users. While by predicting the usage of stations for coming timestamp, company managers would know which station would be busy and have related activities. For instance, they can put commercial advertisements onto such busy station so that many users can see. The abnormal patterns of VAST 2017 can also be detected using SOM-based clustering. Not only in-regular patterns but also the popular patterns can be found out using the same method. Prediction of movement flow also shows the advantages with high accuracy from experiment results. Generally, the experiment results of all three temporal context data sets significantly indicate the excellent usability of this approach.

7.3 Future Work

The future work can be concerned on mainly two divisions. For one, our prediction approach still gets a lot of space to be further enhanced. For instance, different architectures and design choices of LSTM-RNN might have strong impacts on the prediction accuracy. In our case, we only attempt three-layers architecture. Even though our result is good. But we believe more attempts with different architectures of RNN should be tried in the future.

The other work can be conducted is that beyond these two features extraction that clustering and prediction, there are still many important features and embedded information can be mined from the time-spatial data. Therefore, more approaches together with neural net can be applied in this research.

Overall, future work will be concerned on the how to enhance current method and what new methods should be introduced to resolve the temporal-spatial data visual analytics.

Bibliography

[1] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] Zhenghao Chen, "Neural Net-Based and Safety-Oriented Visual Analytics for Time-Spatial Data," *Inter-national Joint Conference on Neural Networks*, 2017.

[3] Tatiana von Landesberger, Felix Brodkorb, Philipp Roskosch, Natalia Andrienko, Gennady Andrienko, and Andreas Kerren, "Mobilitygraphs: Visual analysis of mass mobility dynamics via spatio-temporal graphs and clustering," *IEEE transactions on visualization and computer graphics*, vol. 22, no. 1, pp. 11–20, 2016.

[4] R. S. Varga, *Matrix Iterative Analysis*, Prentice-Hall, Englewood Cliffs, NJ, 1st Edition, 1962.

[5] Yu Zheng, Lizhu Zhang, Xing Xie, and Wei-Ying Ma, "Mining interesting locations and travel sequences from GPS trajectories," in *Proceedings of the 18th international conference on World wide web*. ACM, 2009, pp. 791–800.

[6] Natalia Andrienko and Gennady Andrienko, "Visual analytics of movement: An overview of methods, tools and procedures," *Information Visualization*, vol. 12, no. 1, pp. 3–24, 2013.

[7] Teuvo Kohonen, "The self-organizing map," *Neurocomputing*, vol. 21, no. 1, pp. 1–6, 1998.

[8] Sanjiv Kumar Shukla, Sourabh Rungta, and Lokesh Kumar Sharma, "Self-Organizing Map based clustering approach for trajectory data," *International Journal of Computer Trends and Technology (IJCTT)*, vol. 3, no. 3, pp. 321–324, 2012.

[9] Qiang Liu, Shu Wu, Liang Wang, and Tieniu Tan, "Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts.," in *AAAI*, 2016, pp. 194–200.

[10] LR Medsker and LC Jain, "Recurrent neural networks," *Design and Applications*, vol. 5, 2001.

[11] Yu Zheng, "Trajectory data mining: an overview," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 6, no. 3, pp. 29, 2015.

[12] Zeqian Shen and Kwan-Liu Ma, "Mobivis: A visualization system for exploring mobile data," in *Visualization Symposium, 2008. PacificVIS'08. IEEE Pacific*. IEEE, 2008, pp. 175–182.

[13] Luca Chittaro, "Visualizing information on mobile devices," *Computer*, vol. 39, no. 3, pp. 40–45, 2006.

[14] Quannan Li, Yu Zheng, Xing Xie, Yukun Chen, Wenyu Liu, and Wei-Ying Ma, "Mining user similarity based on location history," in *Proceedings of the 16th ACM SIGSPATIAL international conference on Advances in geographic information systems*. ACM, 2008, p. 34.

[15] Min Lu, Chufan Lai, Tangzhi Ye, Jie Liang, and Xiaoru Yuan, "Visual analysis of route choice behaviour based on GPS trajectories," in *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*. IEEE, 2015, pp. 203–204.

[16] Raghu Ramakrishnan, "Database Management Systems . pdf," 2000.

[17] Jie Bao, Ruiyuan Li, Xiuwen Yi, and Yu Zheng, "Managing massive trajectories on the cloud," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2016, p. 41.

[18] Julian Hagenauer, Marco Helbich, and Michael Leitner, "Visualization of crime trajectories with self-organizing maps: a case study on evaluating the impact of hurricanes on spatio-temporal crime hotspots," in *Proceedings of the 25th conference of the International Cartographic Association, Paris*, 2011.

[19] S Wayne Martin, Pascal Michel, Dean Middleton, John Holt, Jeff Wilson, et al., "Investigation of clusters of giardiasis using GIS and a spatial scan statistic," *International Journal of Health Geographics*, vol. 3, no. 1, pp. 11, 2004.

[20] Martin Kulldorff, "A spatial scan statistic," *Communications in Statistics-Theory and methods*, vol. 26, no. 6, pp. 1481–1496, 1997.

[21] Natalia Adrienko and Gennady Adrienko, "Spatial generalization and aggregation of massive movement data," *IEEE Transactions on visualization and computer graphics*, vol. 17, no. 2, pp. 205–219, 2011.

[22] Scott Gaffney and Padhraic Smyth, "Trajectory clustering with mixtures of regression models," in *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999, pp. 63–72.

[23] Junbo Zhang, Yu Zheng, and Dekang Qi, "Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction," *arXiv preprint arXiv:1610.00081*, 2016.

[24] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[25] Mark Whiting, Kristin Cook, Georges Grinstein, John Fallon, Kristen Liggett, Diane Staheli, and Jordan Crouser, "VAST Challenge 2015: Mayhem at dinofun world," in *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*. IEEE, 2015, pp. 113–118.

[26] Michael Steptoe, Robert Krueger, Yifan Zhang, Xing Liang, Rolando Garcia, Sagarika Kadambi, Wei Luo, Thomas Ertl, and Ross Maciejewski, "VAST Challenge 2015: Grand Challenge-Team VADER/VIS Award for Outstanding Comprehensive Submission," in *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*. IEEE, 2015, pp. 119–120.

[27] Eric Zivot and Jiahui Wang, *Modeling financial time series with S-Plus®*, vol. 191, Springer Science & Business Media, 2007.

[28] Pavel Senin, "Dynamic time warping algorithm review," *Information and Computer Science Department University of Hawaii at Manoa Honolulu, USA*, vol. 855, pp. 1–23, 2008.

[29] Jieqiong Zhao, Guizhen Wang, Junghoon Chae, Hanye Xu, Siqaio Chen, William Hatton, Sherry Towers, Mahesh Babu Gorantla, Benjamin Ahlbrand, Jiawei Zhang, et al., "ParkAnalyzer: Characterizing the movement patterns of visitors VAST 2015 Mini-Challenge 1," in *Visual Analytics Science and Technology (VAST), 2015 IEEE Conference on*. IEEE, 2015, pp. 179–180.

[30] Shyam Varan Nath, "Crime pattern detection using data mining," in *Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops. 2006 IEEE/WIC/ACM International Conference on*. IEEE, 2006, pp. 41–44.

[31] Pascal Soucy and Guy W Mineau, "A simple KNN algorithm for text categorization," in *Data Mining, 2001. ICDM 2001, Proceedings IEEE International Conference on*. IEEE, 2001, pp. 647–648.

[32] Gunnar Rätsch, Takashi Onoda, and K-R Müller, "Soft margins for AdaBoost," *Machine learning*, vol. 42, no. 3, pp. 287–320, 2001.

[33] Andrew McCallum, Kamal Nigam, et al., "A comparison of event models for naive bayes text classification," in *AAAI-98 workshop on learning for text categorization*. Citeseer, 1998, vol. 752, pp. 41–48.

[34] S Rasoul Safavian and David Landgrebe, "A survey of decision tree classifier methodology," *IEEE transactions on systems, man, and cybernetics*, vol. 21, no. 3, pp. 660–674, 1991.

[35] Andy Liaw and Matthew Wiener, "Classification and regression by randomForest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[36] Hirotoshi Taira and Masahiko Haruno, "Feature selection in SVM text categorization," in *AAAI/IAAI*, 1999, pp. 480–486.