東京大学大学院新領域創成科学研究科
社会文化環境学専攻

2017年度

修士論文

GPS軌跡による都市規模の鉄道交通分析システム
A System for Citywide Railway Traffic Analysis Through GPS trajectories

2017年7月18日提出

指導教員　柴崎　亮介　　教授

夏 天琦
Xia, Tianqi

Abstract

Urban railway transit is of great significance in the daily lives of Metropolitan residents as a large amount of residents choose to travel by train. Therefore understanding and analyzing railway traffic is fundamental to urban planning as well as the building of intelligent transportation system (ITS). With the development of big data technology, a lot of researches have been conducted on analyzing railway traffic through several types of sensors while few of them focus on traffic analysis with GPS data. Therefore, in this research an intelligent system is presented for efficiently representing and understanding railway traffic as well as finding out railway traffic anomalies through big heterogeneous data. The system is validated through the official statistic data and its feasibility of detecting anomalies is tested on some typical abnormal events.

Keywords: traffic analysis, GPS trajectory, railway network, anomaly detection

Acknowledgements

Foremost, I would like to express my sincere gratitude to my advisers Prof. Shibasaki, Prof. Song and Dr. Kusakabe. I would like to give my gratitude to Shibasaki-sensei for the continuous support of my master study and research. Prof. Shibasaki always listens to my reports carefully and gives very valuable information for my researches. I would like to specially thank Prof. Song for his devotion in the guidance and instruction to me throughout my master years. Prof. Song is the leader of IPUC group which I affiliated to. He teaches me a lot of methodologies of analyzing human mobility and instructs me on doing experiments and writing papers. My special sincere thanks also goes to Dr.Kusakabe for his advice and guidance of my master thesis. I learn a lot from Kusakabe-sensei about the methods of railway traffic analysis.

In addition, I thank my labmates in Shibasaki Lab: thanks the IPUC group members including the senior students Zipei Fan, Satoshi Miyazawa,Quanqun Chen and Renhe Jiang for teaching me a lot on my research and helping me to get used to study and live in Japan. Besides, the thank also goes to the junior students Dou Huang and Shuzhe Huang in IPUC group for their support and encouragement on my researches. Thanks the group members of data infrastructures especially Researcher Kanasugi for sharing the railway data and his source code for people flow projects on which I modify and develop the tools I need for my research. I have a good time to cooperate with Kanasugi-san for processing the railway data and I learn a lot of skills of data processing from him. Thanks Zhiling Guo with whom I take the master course together for giving me very useful information and motivation for my researches. Thanks Prof. Shao, Dr. Xu, Dr. Chen for their helps in my daily live in Japan.

Last but not the least, I would like to thank my family especially my parents for giving birth to me at the first place and supporting me spiritually throughout my life.

Contents

Abstract iii

Acknowledgements iv

List of Figures vii

List of Tables ix

1 Introduction 1

2 Related Works 5

2.1 Research on Railway Transportation 5

2.2 Research on Traffic and Human Behavior Analysis Through GPS Data 6

2.3 Research on Traffic Anomaly 7

3 Methodology 9

3.1 Concept and Framework 9

3.2 Map Matching 11

3.2.1 Review of Map Matching Methods 11

3.2.1.1 Categories of Traditional Map Matching Methods 11

3.2.1.2 Feasibility of Traditional Map Matching Methods 12

3.2.2 Map Matching with Hidden Markov Model 13

3.2.2.1 Hidden Markov Model 13

3.2.2.2 HMM for Map Matching 13

3.2.2.3 Output Probability and Transition Probability in Map Matching 14

3.3 Network Kernel Density Estimation 15

3.3.1 Preliminary 15

3.3.2 Estimation Process 16

3.4 Anomaly Detection 17

3.4.1 Review of Anomaly Detection 17

3.4.2 PCA Based Anomaly Detection 18

3.4.2.1 Preliminary 18

3.4.2.2 Procedures of Anomaly Detection with PCA 19

3.4.3 Prediction Model Based Anomaly Detection 20

3.4.3.1 Preliminary 20

3.4.3.2 Procedures of Anomaly Detection with Prediction Models . . 23

4 Experiment **25**
4.1 Heterogeneous Data Sources . . . 25
4.1.1 Railway Network . . . 25
4.1.2 GPS Trajectories . . . 25
4.1.3 Daily Passenger Number of each Station . . . 26
4.1.4 Test Area . . . 27
4.2 Experiment of Preprocessing Module . . . 27
4.2.1 Map Matching . . . 28
4.2.2 Data Interpolation . . . 28
4.2.3 Evaluation and Statistical Analysis of Map Matching Results . . . 30
4.2.3.1 Evaluation of Analyzing Railway Traffic Through GPS Data 30
4.2.3.2 Statistical Analysis of Map Matching Results . . . 30
4.3 Experiments of Network Kernel Density Estimation . . . 31
4.3.1 Parameter Settings . . . 31
4.3.2 Visualization and Discussion . . . 31
4.4 Anomaly Detection . . . 33
4.4.1 Anomaly Data and Training Datasets for Feasibility Test . . . 33
4.4.2 Anomaly Detection with PCA . . . 34
4.4.2.1 Parameter Setting and Anomaly Detection Procedures . . . 34
4.4.2.2 Anomaly Detection Results and Discussions . . . 35
4.4.3 Experiment on Prediction Model Based Anomaly Detection . . . 36
4.4.3.1 Parameter Settings and Model Selection . . . 37
4.4.3.2 Anomaly Detection Results and Discussions . . . 39
4.4.3.3 Causes of Failures: A Further Exploration . . . 39

5 Conclusions and Future Work **41**
5.1 Conclusions and Discussions . . . 41
5.2 Future Work . . . 42

A Additional Specification for Data Preprocessing **43**
A.1 Tools developed for this research . . . 43
A.2 Matching Data from NLNI to our data set . . . 44

Bibliography **45**

List of Figures

1.1 Graph of annual traffic delay counts from Ministry of Land, Infrastructure, Transport and Tourism (MLIT) (http://www.mlit.go.jp/common/001097991.pdf) 2
1.2 Graph of annual traffic accident counts from MLIT 2

3.1 Distance and projection specification 10
3.2 System Overview: the intelligent system utilize heterogeneous data to analyze railway traffic and detect anomalies with three main modules 11
3.3 An example of the problem that use only geometry or topology information for railway map matching. Here Point 2 is closer to Line B than Line A 12
3.4 A visualization of HMM and Viterbi: in Viterbi algorithm, only the largest probability of current state will be saved to finding the best path. 14
3.5 An example of the function of PCA 19
3.6 The comparison of two prediction types. In this figure, the data size for in-sample prediction is 500 with the prediction size 30, while the data size for out-of sample prediction is 470. 23

4.1 Visualization of the railway network, station data, and a sample of GPS points . 27
4.2 An example of map matching and data interpolation 29
4.3 An comparison of the raw data and preprocessed data 29
4.4 Regression plot of GPS data and passenger numbers 30
4.5 Histogram of trajectory number, line number and O-D number per user 31
4.6 Four kernel density maps to visualize the kernel density values at different time 32
4.7 Kernel Density of the lixel near Kanamachi and Ueno in a normal week 33
4.8 Comparison of statistics 33
4.9 Variance ratio of each principle component 35
4.10 Comparison of abnormal and normal principle components (left: abnormal, right: normal) 35
4.11 PCA Anomaly detection results of four datasets 36
4.12 ACF and PACF of differenced sequences 38
4.13 Seasonal ARIMA Anomaly detection results of four datasets 38

A.1 Visualization of the number of passengers in Ueno. The table indicate that the number of passengers is calculated only once for the lines in the same company. 44

List of Tables

1.1 Traffic mode share in railway of different metropolises (Data source: Land Transport Authority of Singapore [4]) 1

4.1 Data dictionary of railway links 26
4.2 Data dictionary railway stations 26
4.3 Data dictionary of GPS trajectory 27
4.4 A test data set for different kinds of anomalies 34
4.5 Comparison of Different Models 37

Chapter 1

Introduction

Rail transportation system plays an important role in urban development as it is an energy-efficient and relatively punctual way for transporting passengers and freights. According to the survey of Person Trip conducted by Japanese government as well as other countries [4], railway transportation takes up a significant proportion of travel modes which the citizens of metropolises choose all over the world. Table 1.1 shows the proportion in some world wide famous metropolises. Among the selected cities in this table, Tokyo has the maximum ratio of rail transit which is up to 48%, showing that almost half of the citizens prefer to travel by train or metro service. Therefore, understanding railway traffic is of great significance to build intelligent transportation system and smart city.

TABLE 1.1: Traffic mode share in railway of different metropolises (Data source: Land Transport Authority of Singapore [4])

Metropolis	Population(Million)	Area(km^2)	Traffic Mode Share in Railway
Beijing	12.3	1368	17%
Hong Kong	7.2	1104	30%
London	8.4	1595	12%
New York	8.4	784	12%
Singapore	5.5	718	21%
Tokyo(23-ward)	9.1	623	**48%**

A serious issue in railway traffic analysis is the detection of abnormal events. Over the past few decades, the anomalies result in enormous financial losses to the society. Figure 1.1 demonstrates the statistic results of traffic disorder cases from 1988 to 2014.

Anomaly of railway transportation can be concluded to two types. On one hand, there are a lot of regular events which attracts a lot of visitors such as The Sumida River Fireworks Festival

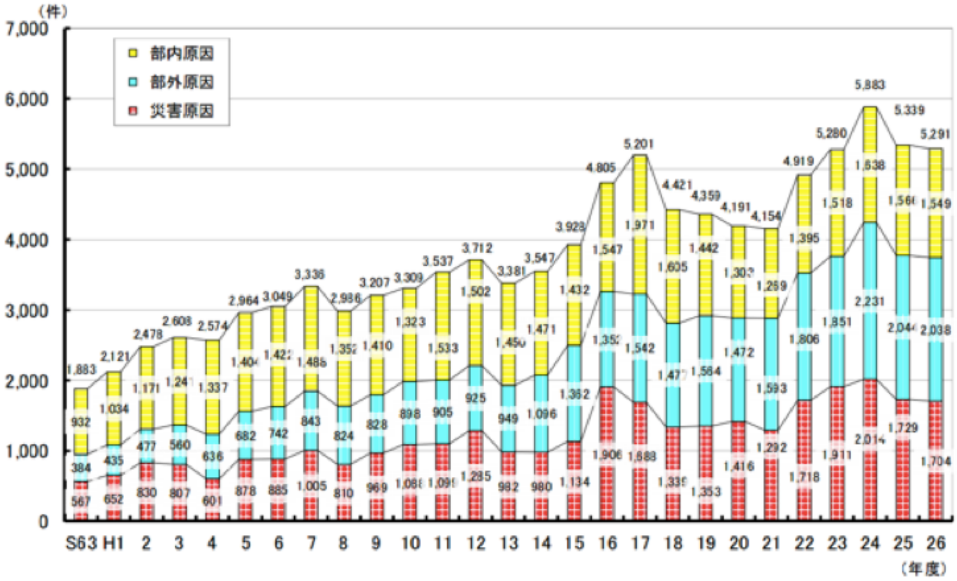


FIGURE 1.1: Graph of annual traffic delay counts from Ministry of Land, Infrastructure, Transport and Tourism (MLIT) (http://www.mlit.go.jp/common/001097991.pdf)

and Comiket. These big events cause the crowdedness in the nearby railway systems. On the other hand, railway transportation is influenced by natural disasters as well as human accidents, which may cause delay or suspension of railway transportation service. Figure 1.2 is shows the annually traffic accident numbers.

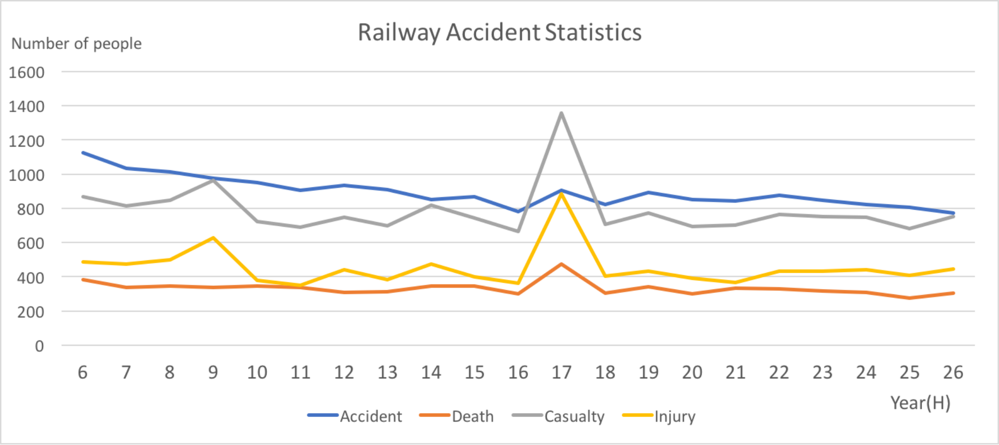


FIGURE 1.2: Graph of annual traffic accident counts from MLIT

With the popularization of smartphones and the development of location based services (LBS), huge amounts of GPS data generated by these LBS applications become available for traffic analysis. Over the past decades, though a lot of researches have been focused on understanding human behaviors or detecting anomalies of road network using GPS data, few of them pay attention railway network as analysis of railway traffic using GPS trajectory data is challenging due to the following factors:

1. Separating railway GPS trajectories with other trajectories is usually a challenging data preprocessing task.
2. The topology of rail lines is more complex than roads as transferring from one line to another requires extra walking distance, waiting time and sometimes traveling cost.
3. Citywide Railway traffic have spatial and temporal dependencies which are related to rail lines and timetable. For example, an accident happens in a rail line may have larger effect on passengers in the same line with a long distance than passengers in other lines closed to the accident.

With the given background and motivation above, the purpose of this research is to develop a system for understanding railway traffic through GPS trajectory, which include the traffic volume as well as the traffic anomalies. The remaining of this thesis is structured as follows: firstly, some related works concerning railway traffic analysis, human behavior analysis through GPS data as well as traffic anomaly analysis are introduced in Chapter 2. Then the methodology of this research including the framework and some important railway traffic analysis methods are introduced in Chapter 3. Chapter 4 shows the procedure and results of the experiments and Chapter 5 concludes the whole paper with the research proposal in the next steps and some points to be improved in this research.

Chapter 2

Related Works

2.1 Research on Railway Transportation

Research of railway transportation can be categorized into two classes based on their research objects. On one hand, the train itself has the information like position, velocity, acceleration and weight acquired by the sensors. These information can be used for detect the status of the train. Rabatel et al. [33] collects the data from different sensors and extract the anomaly patterns with data mining technologies. Lu and Schnieder [22] utilize GNSS receiver as well as other sensors of the train for localization and evaluate the performance by comparing with standards.

On the other hand, railway traffic can be represented by the number of passengers in stations and trains. With the development of technology, the traffic volume data, which used to be acquired through questionnaire or ticket count only, can be collected much more precise and convenient through a lot of sensors. Over the past years, many researches have focused on utilizing tap-in and tap-out information to analyze railway traffic as it provides the information of origin and destination (OD) which can be directly used to estimate the traffic volume in each station Asakura et al. [3], Pelletier et al. [29]. However, since train is separate from the smart card system, utilizing smart card cannot acquire the route or train information directly. Kusakabe et al. [19] develop a methodology to use smart card data to extract the exact railway travel information of passengers including waiting time, timetable information, transfer information and the service the user choose. Sun et al. [35] proposes an framework using Bayesian inference method to assign passengers to railway network with OD pairs and cost information.

In addition, there are some sensors that can indirectly reflect railway traffic at real time. Cameras are the most widely used sensors in stations and trains that can be used to analyze the people flows Prassler et al. [30] and laser-range scanner is proved to be a valid tool for tracking pedestrians in the work of Zhao and Shibasaki [43]. Recently, East Japan Railway(JR East) Company provide an application for detecting the crowdedness at real time through weight measuring of each car, marking that detecting traffic volume with these sensors can be applied to people's daily life.

2.2 Research on Traffic and Human Behavior Analysis Through GPS Data

Though there are few research utilizing GPS data to analyze railway traffic, GPS data has been widely used for analyzing road traffic as well as human behaviors.

An important field of traffic analysis using GPS data is road traffic prediction. The research concerning traffic prediction can be divided into short-term prediction and long-term prediction based on the temporal granularity of the data. The long term analysis of railway traffic is more close to the issues of urban plan and always take other factors into consideration such as the land use, population and economic status Mitchell and Rapkin [24]. On the other hand, short-term traffic analysis is conducted through the analysis of traffic variances through some regression or auto correlation models such as ARIMA Hamed et al. [15]. In recent years, with the development of deep learning technology, some deep learning methods such as RNN, LSTM and CNN are proved to be valid in dealing with sequential data. Zhang et al. [42] propose a grid-based method for predicting in-flow and out-flow with deep spatial-temporal residential networks.

In addition, some research focuses on analyzing human behavior through GPS data. In these kinds of research, the models are built by the training samples of each trajectory to find out the patterns or mutual behaviors of the individuals. Fan et al. [12] propose a machine learning model to predict the crowd behavior in the citywide level. Song et al. [34] utilize a deep learning model to predict the location as well as traffic mode of the individuals.

2.3 Research on Traffic Anomaly

The researches of traffic anomaly mainly focus on the road traffic. As these research methods and themes can be applied to railway traffic to some extent, they are briefly introduced it in this section.

The anomaly of road network can also be categorized into traffic congestion and traffic accidents. There are a lot of researches concerning traffic congestion. Wen et al. [37] utilize the GPS log data of taxis to statistically analyze traffic congestion changes around the Olympic games in Beijing. Kaklij [18] propose a system to use some clustering and classification methods to detect traffic congestion. In recent years,some researches also utilize deep learning methods to analyze traffic congestion. Ma et al. [23] generalize the congestion problem as a binary classification problem and predicts traffic congestion on each road link with the energy-based deep model RNN-RBM.

On the other hand, traffic accident analysis deals with traffic accident data as well as the factors that are considered to cause the accidents. Abdel-Aty and Radwan [1] models the traffic accidents with different factors by a binomial probability distribution and estimate the parameters through the accident cases. Chen et al. [8] utilize big GPS data as well as the accident data to infer the accident risks of the area through a deep network.

Chapter 3

Methodology

3.1 Concept and Framework

The objective of this research is to use GPS data to represent railway traffic and detect the anomalies of the number of railway passengers. In order to achieve this goal, we firstly generalize GPS trajectories and railway information to the data stored in computer with some specific data structures, then an an intelligent system is proposed by using these big and heterogeneous data to analyze railway traffic.

The data structures utilized in this research can be defined as follows:

Definition 1 (Trajectory Dataset and Trajectory Data): A trajectory dataset can be represented by T = $\{tr_1$, tr_2, ..., $tr_n\}$ where tr_i refers to a trajectory record for one specific passenger. A trajectory is made up of several GPS points which can be denoted by l_i $=$ $(uid, tid, pid, longitude, latitude, timestamp)$ where uid denotes the user id, tid denotes the trajectory and pid is the sequence number of the point in this trajectory.

Definition 2 (Railway Network): A railway network can be denoted by a graph $G(V, E)$ where V is the set of nodes which represent railway stations and E is the set of links which represent railway segments between railway stations. In this research, the railway link set is also denoted by L with each link in denoted by L_i.

Comparing to road network, railway network is sparse and the origin destination (OD) is fixed to the station points. However, railway network has more information such as rail lines, service types (local, rapid, express) and transfer information.

Definition 3 (Projection): In this research, the projection of one GPS point l_i to the link L_i can be denoted by the point p_i on L_i where the distance from point l_i to p_i is the shortest among all the points L_i. Thus the distance from l_i to L_i can be calculated by the euclidean or great circle distance from l_i to p_i as is shown in Figure 3.1.

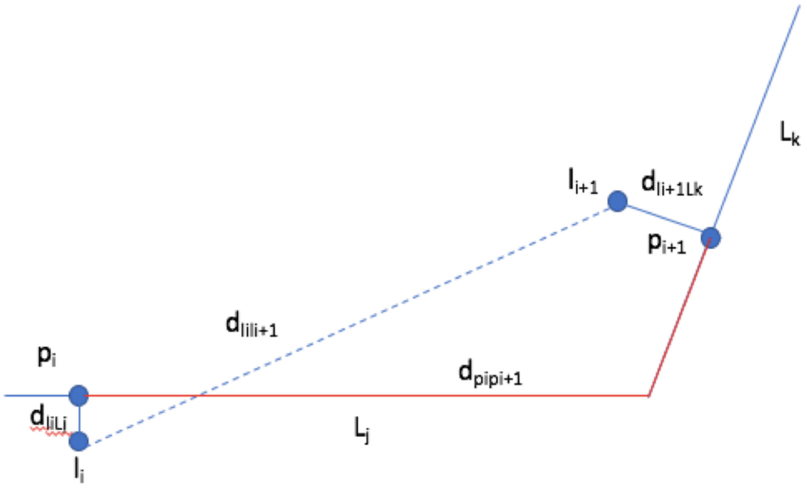


FIGURE 3.1: Distance and projection specification

Definition 4 (Railway Network Distance): In graph theory, the shortest path of two nodes in a network can be calculated by some graph algorithms such as Dijkstra Algorithm Dijkstra [10]. In this research, since railway network has a lot of semantic information, the shortest path of two stations is computed through a multi-criteria Dijkstra shortest-path algorithm proposed by Disser et al. [11] is adapted and utilized for computing the shortest path railway network so as to take transfer and company information into consideration. The railway network distance of two GPS points l_i and $l_i + 1$ is regarded as the network distance of two projected points on the network which can be calculated based on the shortest past between their nearest stations.

The framework of this system is demonstrated in Figure 3.2. The system has three main modules. The data preprocessing module matches the trajectories of railway passengers to the railway network and interpolates GPS data into five-minute intervals. The pattern extraction module utilizes a network kernel density estimation method to calculate the kernel density of passengers on railway network to represent the crowdedness. The anomaly detection module mainly utilizes machine learning and statistic methods to find out the segments with abnormal traffic density.

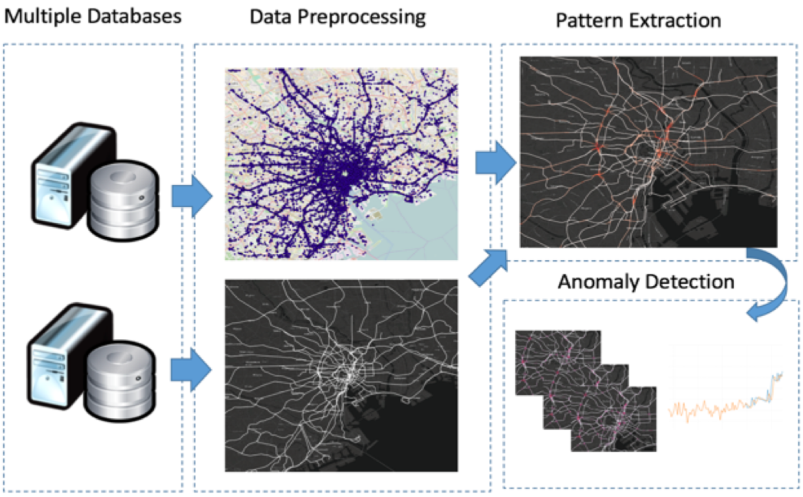


FIGURE 3.2: System Overview: the intelligent system utilize heterogeneous data to analyze railway traffic and detect anomalies with three main modules

Three key techniques utilized in this research, which are respectively map matching, network kernel density estimation and anomaly detection, will be introduced in the remaining of this chapter.

3.2 Map Matching

3.2.1 Review of Map Matching Methods

Map matching is the problem of matching trajectory points into road networks. In this research, map matching is the key component in data preprocessing module to acquiring railway information like stations and rail line names from trajectories as well as reduce the accuracy error of GPS.

3.2.1.1 Categories of Traditional Map Matching Methods

Traditional map matching methods are categorized into four classes in the work of Quddus et al. [32]. Geometric analysis based map matching methods utilize geometry information to match points into segments or nodes without taking topology into consideration Bernstein and Kornhauser [5], Greenfeld [13]. The topological based map matching methods take topological features into consideration and can better discover the spatial patterns of road connectivity and region containment Yu [41]. The probability map matching methods Ochieng et al. [26] represent the errors of each GPS point with probability function and the GPS points are matched to the road links with the maximum probability. Besides, other advanced approaches such as Kalman filter and its extensions Yang et al. [40], particle filter Gustafsson et al. [14] and Fuzzy theories Quddus et al. [31] have been developed. The advanced map matching methods aim at get a more precise positioning output at the real time.

3.2.1.2 Feasibility of Traditional Map Matching Methods

In this research, as the GPS data is sparse and in low precision while the accessible route in railway network is limited from origin to destination. It is much more important and feasible to get the route information than the precise location. Thus the advanced map matching technology is not suitable for this research. On the other hand, using geometry or topology information only will cause some problems that the route choice may not be realistic for railway passengers which is shown in Figure 3.3. In this figure, there is a trajectory contains three GPS points and there are two candidate route for this trajectory which are respectively R_1= $lineA$, or R_2 = $LineA$,$LineB$, $LineC$, $LineA$. If the GPS points are matched to the nearest points, route R_2 will be chosen as $Point2$ is closer to $LineB$ than $LineA$. However, it is obvious that the trajectory should be matched to R_1 as it is less likely for a passenger to transfer twice to the same line. Thus, the map matching model for GPS trajectory data on railway should have a higher probability to match the points based on their previous rail lines choice.

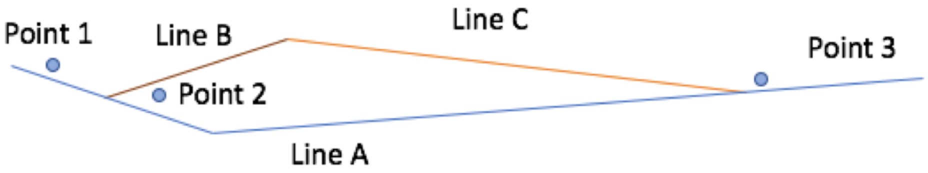


FIGURE 3.3: An example of the problem that use only geometry or topology information for railway map matching. Here Point 2 is closer to Line B than Line A

In order to make map matching more accurate in sparse dataset and solve the problem mentioned above, this research utilizes a Hidden Markov Model based map matching method which chooses the candidate links through the probabilities in section 3.2.2.

3.2.2 Map Matching with Hidden Markov Model

Recently, Hidden Markov Model (HMM) is proved to be a valid approach for matching sparse and low precision GPS data Lou et al. [21], Newson and Krumm [25]. In this section, we will go through the concepts about HMM and the procedures of applying HMM on GPS trajectory.

3.2.2.1 Hidden Markov Model

Given Markov process is a stochastic process in which the outcomes at any stage are determined by the outcomes of the previous stage, the Hidden Markov Model is a model in which the observed data is organized as a Markov process with hidden states. Thus in HMM, the two basic assumptions are as follows: on one hand, the observed data at any stage is assumed to be dependent on the hidden state in that stage with an output probability distribution. On the other hand, the hidden state at one stage is determined by the previous hidden state by a transition probability. Therefore, given the initial state denoted as $P(S_1)$, the transition probability $P(S_t|S_{t-1})$ for any hidden state S_t acquiring from the transition matrix as well as the output probability $P(Y_t|S_t)$, the probability distribution of observations over the sequences can be determined.

3.2.2.2 HMM for Map Matching

For trajectory map matching using HMM, the visible output is the trajectory point, while the hidden state is the road which the trajectory point belongs to. Thus the output probability can be measured by the distance from the trajectory point to its projection in the railway line and the transition probability can be measured by the network distance of two adjacent trajectory points. The most probable route which the trajectory is matched to is the hidden state sequence with the maximum product values of output probabilities and transition probabilities, which is usually calculated by Viterbi algorithmViterbi [36].

A brief procedure for railway network map matching is as follows and is shown in Figure 3.4:

1 For each GPS point l_i in a railway trajectory, loop the railway network to find out the candidate rail lines L_1,L_2...L_n within the searching radius.

2 For each candidate line, Calculate the output probability of l_i matched to the line.

3 From the first GPS point to the last GPS point. Calculating the transition probability of the hidden states between the adjacent GPS points. For each state, the total probability with the path from the first state to this state is the product of all transition and output probabilities. Here, since the number of paths might be very large in some cases, to reduce computation time, the Viterbi algorithm is iterated in each state to find out the best path.

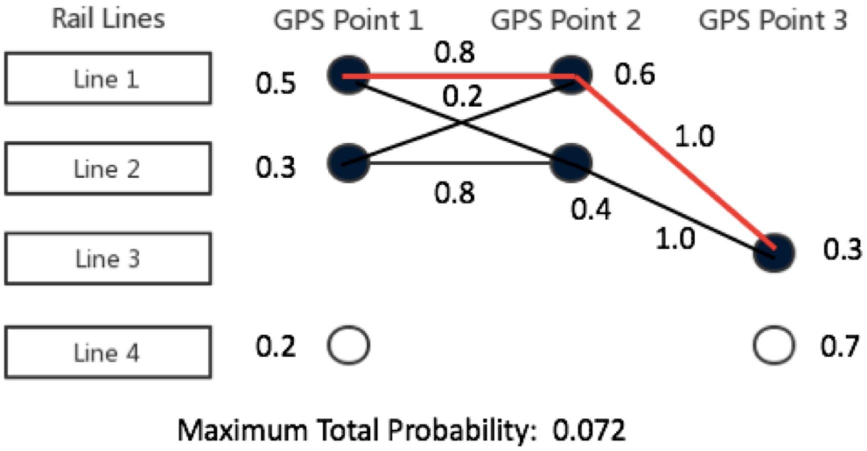


FIGURE 3.4: A visualization of HMM and Viterbi: in Viterbi algorithm, only the largest probability of current state will be saved to finding the best path.

3.2.2.3 Output Probability and Transition Probability in Map Matching

Output and transition probability are two fundamental factors to determine the map matching result. In this work, the definition of these two probabilities are adapted from the work of Newson and Krumm [25] with the following rules:

1 The output probability should be inversely proportional to the distance between the GPS point and the potential rail lines it belongs to.

2 The transition probability between two adjacent hidden states should be proportional to the accessibility of these two rail links.

In this research, the output probability of a trajectory point l_i matched to a railway link L_j is calculated through a Gaussian distribution as is shown in equation 3.1.

$$p(l_i|L_j) = \begin{cases} \frac{1}{\sqrt{2\pi}\sigma} e^{-0.5(\frac{d_{liLj}}{\sigma})^2} & \text{when} \quad 0 < d_{liLj} \leqslant r \\ 0 & \text{when} \quad d_{liLj} > r \end{cases} \tag{3.1}$$

Where d_{liLj} is the vertical distance from the trajectory point to the railway link and σ is the standard deviation of GPS measurements.

The transit probability from one hidden state $s_{l_i L_j}$ to the adjacent hidden state $s_{l_(i+1) L_k}$ is calculated through a normal score with the following conditions:

1 If L_i and L_k are in the same railway, the score will be set to 1.

2 if L_i and L_k are in different railways, the transfer cost c_jk will be calculated which is proportional to the number of transfer times and transfer companies, the length of transfer distance and the ratio of route distance. $d_{p_i p_i+1}$ and euclidean distance $d_{l_i l_i+1}$ where the distance is introduced in 3.1.

3.3 Network Kernel Density Estimation

3.3.1 Preliminary

Network kernel density estimation(KDE) is an adaption of standard kernel density estimation which is used for calculating the density of point events over network space. Network KDE has been widely used in the analysis of traffic accidentsXie and Yan [39]. In this research, as GPS data are matched to railway network, trajectories can be regarded as an activity through the network, thus network kernel density can be utilized to analyze the distribution of the GPS points.

Definition 1 (Lixel): Lixel is the basic linear unit for density estimation and prediction in our system. A specific railway segment e_i in the link set can be divided into several lixels denoted by $\{l_{i1}$,l_{i2},...,$l_{in}\}$. These lixels share the same length except for the last lixel with the remaining length. In network KDE, all point events are assigned to their nearest lixel and each lixel can be represented as a tuple of $(id, pointcount, density, geometry, lixelcenter)$ where $lixelcenter$ is the centroid of each lixel and is used for distance calculation.

Definition 2 (Distance and Bandwidth): Instead of Euclidean distance, shortest-path distance is measured in network KDE. The distance d_{is} from lixel l_i to lixel l_s in network KDE refers to the short-path length between these two $lixelcenters$ and bandwidth is the maximum searching distance.

Definition 3 (Kernel Function): Kernel function in network KDE is used for measuring distance decay effect. In this paper we choose Gaussian kernel function which is defined as:

$$k(\frac{d_{is}}{r}) = \begin{cases} \frac{1}{\sqrt{2\pi}} \exp(-\frac{d_{is}^2}{2r^2}) & \text{when} \quad 0 < d_{is} \leqslant r \\ 0 & \text{when} \quad d_{is} > r \end{cases} \tag{3.2}$$

Where r is the bandwidth.

Definition 4 (Kernel Density Estimator): An estimator is a function for estimating the kernel density of point events. For network KDE, the estimated density at any point s can be calculated by the following function:

$$\lambda(s) = \sum_{i=1}^{n} \frac{1}{r} k(\frac{d_{is}}{r}) \tag{3.3}$$

This value is proportional to the sum of point numbers multiplied by kernel function in all lixels within the bandwidth of the target lixel.

3.3.2 Estimation Process

Given the definition mentioned above, we calculate network kernel density value within a time period with the following steps:

1) Create a dataset of trajectory points by iterating each trajectory to find out the points within the time period.

2) Divide the railway links into lixels and create a network of these lixels.

3) For each point in the dataset, iterate the lixel network to find out its nearest lixel. Count the numbers of nearest trajectory points in each lixel as the value of $pointcount$ attribute. The lixel with one or more trajectory points are regarded as source lixels.

4) For each source lixel, calculate the shortest distance form the $lixelcenter$ of source lixel to the $lixelcenter$ of its neighboring lixels within the bandwidth r.

5) For each source lixel as well as its neighboring lixels, calculate a density value based on the numbers of trajectory points, the network distances and Gaussian kernel function.

6) For each lixel within the searching bandwidth of any source lixels, sum the total density values from different source lixels.

3.4 Anomaly Detection

In this research, the railway traffic density is assumed to focus on some regular observed patterns. The anomalies (or sometimes called outliers) are the observation values which differ from other observation values thus cannot fit the patterns. The Anomaly detection module aims to extract the anomalies with kernel density values and explain the factors that cause these anomalies.

In anomaly detection module, the input data can be denoted as a $size(t) * size(m)$ matrix Y where t denotes the time span of the dataset while m denotes the total lixel links generated in network KDE module. Then the output of the anomalies should be a subset of $t \times m$ where $Y_{tm} - Y_{tm_normal} > \delta$ in which δ denotes the threshold.

3.4.1 Review of Anomaly Detection

Anomaly detection on graph can be divided by two categories based on the input data format. The origin-destination (OD) based anomaly detection methods aggregate the traffic flows into each OD pairs while the link based method calculate traffic volume of each link. In this research, link based anomaly detection methods are applied in anomaly detection module as network kernel density is represented by the value in each link.

Patcha and Park [28] summarizes the technologies utilized in anomaly detection which are respectively statistic based methods, machine-learning-based methods and data-mining-based methods. In machine-learning-based methods, the author introduces some widely used models such as PCA, Bayesian network and HMM while in data-mining based methods while the data-mining-based methods mainly aim at finding the associated patterns and clusters to extract the outliers. Huang et al. [17] classifies the anomaly detection of network traffic methods into three representative classes based on the theory and the process which are PCA-based, sketch-based(prediction-model-based) and wavelet-based methods. The author describes the prediction-model-based method as a kind of anomaly detection which utilizes some sequential prediction models such as ARIMA or seasonal ARIMA to predict the traffic on links and compare it to the ground truth value to find out the anomalies while wavelet-based method regard the network traffic streams as signals and using signal processing methods to decompose the traffic streams to different bands to find out the anomaly patterns. Besides, some researches try to use heterogeneous data sources to extract and explain the outliers. Pan et al. [27] develops an system utilizing GPS data to extract anomalies and mining the semantic meanings of the anomalies through the geo-tagged SNS data.

In this research, two anomaly detection methods are utilized to extract the outliers which are respectively based on PCA and prediction models. In the remaining of this section we will introduce these two approaches in details.

3.4.2 PCA Based Anomaly Detection

3.4.2.1 Preliminary

PCA (Principal Component Analysis) is the procedure to project the observation data onto a lower dimensional linear space (also named as principle axes or principle components) with the least projection cost Bishop [6].

In PCA, the projection to each principal component is determined by the maximum variance. For the first principle component, the projection is determined by the maximum variance of the whole data as is shown in equation.

$$v_1 = arg\ max \left\| Yv \right\| \tag{3.4}$$

For the kth principle component, it is determined by the maximum variance of the residual, which is the differences of

$$v_k = arg\ max \left\| (Y - \sum_{1}^{k-1} Y v_i v_i^T) \right\| \tag{3.5}$$

After conducting PCA to the dataset, the data will be projected to the new axes as the Figure 3.5 shows:

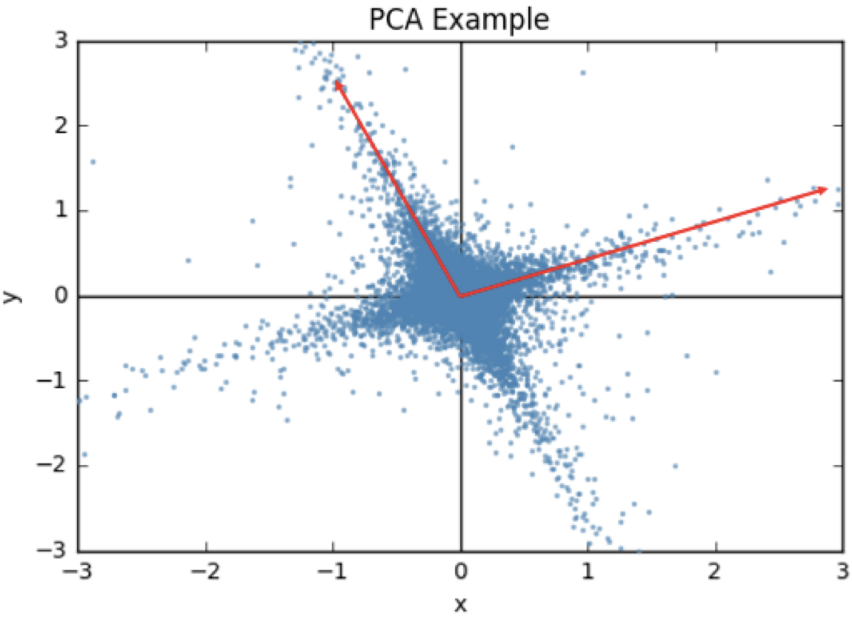


FIGURE 3.5: An example of the function of PCA

3.4.2.2 Procedures of Anomaly Detection with PCA

The procedure of applying PCA to network anomaly detection is introduced in the work of Lakhina et al. [20] which can be divided into the following steps.

1 Decompose the dataset into m principle components (axes).

2 Divide the axes into the components of normal traffic and abnormal traffic based on threshold. The division process goes as follows: First, the process examines the projection on each principal axis in order. If the projection of an axis contains a value which is larger than 3σ from the mean where σ denotes the value of standard deviation, then this component as well as the following components are marked as the abnormal components.

3 Inversely transform the projected values to the original dimensions and compare with the observed value. This step generates a matrix of estimated normal railway traffic matrix Y_n.

4 Compare the estimated matrix with the observed data, filter out the data where the difference is larger than δ

3.4.3 Prediction Model Based Anomaly Detection

Prediction-model-based methods utilize the prediction model to predict new data through the past observed data. In this research, the time series prediction model Seasonal ARIMA, which is an enhanced sequence prediction model for data with seasonal features, will be applied to the dataset for anomaly detection. ARIMA and Seasonal ARIMA as well as their usage on anomaly detection will be introduced in the remaining part of this section.

3.4.3.1 Preliminary

Auto-regressive integrated moving average (ARIMA) is a regression model wildly used in sequential data forecast Box et al. [7]. In ARIMA, the forecast equation is built by three factors, which are respectively auto-regressive(AR) factor representing the lags of the stationarized series for prediction, moving average(MA) factor representing the lags of the prediction errors, and integrated(I) factor representing the rank of the difference to acquire a stable time series. Thus an ARIMA model can be denoted by $ARIMA(p,d,q)$ where p is the number of auto-regressive terms, d is the number of difference orders to acquire a stable sequence while q is the number of prediction errors in the forecast equation.

For a time series of $Y_1, Y_2, .., Y_t, ...Y_n$, for $d = 1$, the difference $y_t^{(1)}$ is calculated by $Y_t - Y_{t-1}$ and when the difference rank is d, then the value of $y_t^{(d)}$ is calculated by $y_t^{(d-1)} - y_{t-1}^{(d-1)}$.

For a differentiated time series, the equation can be denoted by the following equation:

$$\widehat{y_t^d} = \mu + \phi_1 y_{t-1}^d + ... + \phi_p y_{t-p}^d - \theta_1 e_{t-1} - ... - \theta_q e_{t-q} \tag{3.6}$$

Where μ,ϕ and θ are parameters to be estimated and e_t is the error term which is sampled from a normal distribution with zero mean.

The identifying of the values of p, d, q follows Box-Jenkins approach Box et al. [7] with several rules which can be listed as follows:

1 The order of difference in an ARIMA model (d) depends on the autocorrelation result and the standard deviation values. if the series has a positive value of autocorrelation out to a high number of lags, then it is more likely that the model needs a higher order

of difference. When the autocorrelation is zero or negative, then there is no need for a higher order. The optimal value of order is the order at which the standard deviation of the sequence is the lowest.

2 The value of p and q should be inferred from the auto-correlation function (ACF) and partial auto-correlation plots (PACF). ACF can be denoted by the equation

$$R(r) = \frac{E[(y_t - \mu)(y_{t-r} - \mu)]}{\sigma^2} = \frac{\text{Covariance}(y_t, y_{t-r})}{\text{Variance}(y_t)} \tag{3.7}$$

where r represents the lag, μ is the mean of the sequence while σ is the standard deviation of the sequence. PACF is defined by as the conditional correlation between y_t and y_{t-r} on the values between these lags and can be denoted by the equation

$$R(r) = \frac{\text{Covariance}(y_t, y_{t-r}|y_{t-1}, y_{t-2}...y_{t-r+1})}{\text{Variance}(y_t|y_{t-1}, y_{t-2}...y_{t-r+1})(\text{Variance}(y_{t-r}|y_{t-1}, y_{t-2}...y_{t-r+1}))} \tag{3.8}$$

The rules of determining p and q goes as follows: if the PACF of the differenced series displays a sharp cutoff then the indicated value of p should be the lag at which the PACF cuts off, while if the ACF of the differenced series displays a sharp cutoff then the indicated value of q should be the lag at which the ACF cuts off.

Seasonal ARIMA is an enhanced ARIMA model which takes the periodical information into consideration. In SARIMA, the model is divided into two parts which represents respectively the seasonal features and non seasonal features. Given the period of the dataset as s, then the seasonal ARIMA model can be denoted as $ARIMA(p, d, q) \times (P, D, Q)_s$ where P, D, Q denotes the seasonal AR, I, MA factors respectively. Seasonal difference data is similar to the non-seasonal one, except for that $t - 1$ should be changed to $t - s$.

The identifying of seasonal parameters can be listed as follows:

1 The order of seasonal difference D should be set to one if the seasonal pattern of the dataset is stable, however, the D should be no more than one and the total value of $D + d$ should be no more than two.

2 P term should be added If the autocorrelation of the appropriately differenced series is positive at lag s while the Q term should be added if the autocorrelation of the differenced series is negative at lag s.

The evaluation of the order choice can be done by comparing AIC values of the models in different models. AIC stands for Akaike information criterion, which is a measure of the relative quality of statistical models Akaike [2] and can be represented by the following equation.

$$AIC = 2k - 2ln(\hat{L}) \tag{3.9}$$

Where k is the number of estimated parameters and $\hat{L}$ is the maximum likelihood value of the likelihood function for the model. The model with minimum AIC value has the least information loss thus is better than other models on fitting the data and keeping the simplicity of parameters.

Another important method for evaluating the model is LjungBox Q test. Ljung-Box Q test is used to measure whether any groups of autocorrelations in a time series are different from zero, which can be detonated by the equation 3.10

$$Q(m) = n(n+2)\sum_{j=1}^{m}\frac{r_j^2}{n-j} \tag{3.10}$$

where n is the sample size, r_j is the autocorrelation value at lag j and m is the total number of lags. Since the models of ARIMA and seasonal ARIMA assume that the errors that the model fitting data should be identically and independently distributed from each other (also called white noise), thus the probability of the error distribution should accept the null hypothesis of Ljung-Box Q test.

In a sequence prediction model there are two types of prediction distinguished by the prediction data which are respectively in-sample prediction and out-of-sample prediction. In-sample prediction predicts a sub sequence of the total sequence that trains the model while out-sample prediction predicts the value of the different sequence. For in-sample prediction, the observed data can be used to correct the predicted data dynamically, which make the prediction much more precise. Figure 3.6 shows the prediction result of the same sequence with in-sample and out-of-sample prediction.

The performance of prediction model can be evaluated through Root Mean Square Error(RMSE) which is denoted by the following equation:

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(Y_i - \hat{Y_i})^2} \tag{3.11}$$

Where $\hat{Y_i}$ is the predicted value and $\hat{Y_i}$ is the ground truth data.

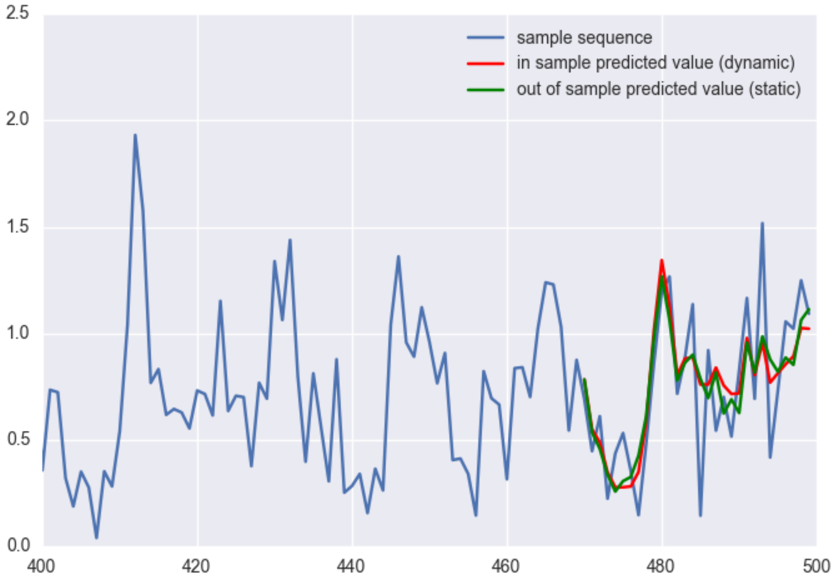


FIGURE 3.6: The comparison of two prediction types. In this figure, the data size for in-sample prediction is 500 with the prediction size 30, while the data size for out-of sample prediction is 470.

3.4.3.2 Procedures of Anomaly Detection with Prediction Models

When applying the prediction models to anomaly detection. At first the parameters of the model should be fit by the training dataset. Then the model will be used to predict the subsequent valuesof the sequence data. The anomaly is detected by determining whether the observed value is in the prediction interval. The data out of the prediction interval will be regarded as the anomalies.

Chapter 4

Experiment

4.1 Heterogeneous Data Sources

In this research, data from heterogeneous data sources are collected for analyze railway traffic which include the railway network, GPS trajectories as well as daily passenger number of each station which will be introduced in details in the remaining of this section.

4.1.1 Railway Network

Railway network is collected from National Land Numerical Information (NLNI) and adapted by Kanasugi et.al in the work of Hiroshi et al. [16]. The adaptation includes simplifying railway network and adding topology information of transfer stations. The data of railway network links and nodes are stored in the same table of a PostgreSQL database with the spatial extension of PostGIS. The data dictionaries of railway links and railway stations are respectively shown in Table 4.2 and 4.2. The total number of links in the table is 13472 with 3281 transfer lines and the total number of railway stations is 10791.

4.1.2 GPS Trajectories

GPS trajectory data are collected based on a huge volume of raw GPS records from August 1, 2010 to July 31, 2013 provided by a mobile operator and a private company. The total number of GPS records is about 30 billions. The total data size of the GPS trajectory is 1.5 TB and is stored as files in a portable HDD and sorted by users.

TABLE 4.1: Data dictionary of railway links

Name	Specification
linkid	Primary Key, the unique identification of the railway network table
comp_code	The company code of the railway lines. Null for transfer line. ID is the same as NLNI
comp_name	The name of railway companies, a combination of two companies for transfer line
line_code	The code to identify each railway line, -1 for transfer line
line_name	The name of railway lines, a combination of two companies for transfer line
source_station_code	The origin (tail) station of each link, the code is the foreign key of railway station table
target_station_code	The target (tail) station of each link, the code is the foreign key of railway station table
length	The length of the link
geom	The geometry information of railway links

TABLE 4.2: Data dictionary railway stations

Name	Specification
station_code	Primary Key, the unique identification of the railway stations
comp_code	The company code of the railway stations
line_code	The code of the railway line the station belongs to
station_name	The name of the railway station
station_group	A column crated for storing topology information, the stations in the same group can be transferred to each other
geom	The geometry information of railway stations(points)

In the GPS files, each row represents an GPS trajectory and a preprocssing is conducted by the the work of Witayangkurn et al. [38] to utilize buffer and speed limitation to classify a GPS trajectory into one traffic mode among walk, train, bus, car and stay. The accuracy of the trajectory marked as train is up to 97.72%. The data dictionary of GPS trajectory is shown in Table 4.3.

4.1.3 Daily Passenger Number of each Station

This data is utilized to validate GPS data and map matching method. The data is acquired from NLNI in the format of shapefile. In this data, the passenger volume is counted by each station of different companies. The detailed specification of utilizing this data is introduced in the appendix.

TABLE 4.3: Data dictionary of GPS trajectory

Name	Specification
user_id	The code to identify each user
trajectory_id	The trajectory id of each user
date	The date of the trajectory
traffic_mode	The traffic mode generated by the work of Witayangkurn et al. [38]
trajectory	The GPS points of a trajectory

4.1.4 Test Area

In this research, the experiments are conducted among a subset which contains the GPS trajectory data in the whole year of 2012 in great Tokyo area.

The visualization of the railway network, station data, and a sample of GPS points in the test area is shown in Figure 4.1.

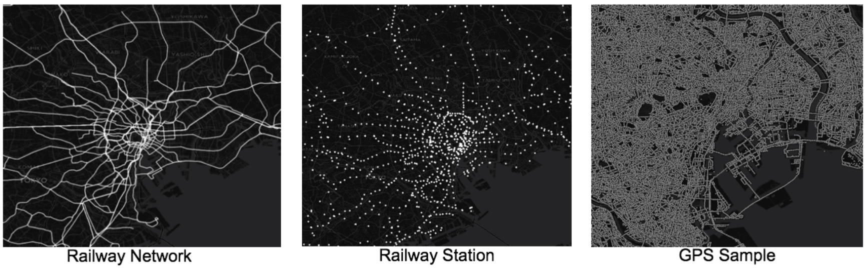


FIGURE 4.1: Visualization of the railway network, station data, and a sample of GPS points

4.2 Experiment of Preprocessing Module

In the module of preprocessing. The data is preprocessed by mainly three steps. Firstly the data is filtered and pruned to reduce the computation volume and improve data precision. Then the HMM map matching method is conducted to the data to acquire the railway information as well as project the points to the rail lines. Finally, the lost data in each trajectory are interpolated to make the trajectory more completed.

Data Filtering and Pruning As the whole GPS trajectory is too much to analyze and the work of Witayangkurn et al. [38] shows a high accuracy of identifying railway trajectory. In this experiment, the trajectory data is firstly filtered by the traffic mode to keep only the GPS trajectory with the mode of train.

As human movement is consequent. A trajectory of train may include some redundant GPS data before or after the railway travel. In the experiment, these data are filtered by the estimated speed and the direction information with the adjacent GPS points.

4.2.1 Map Matching

The map matching code is written in Java with multi-thread to improve the the map matching speed. The output of map matching model consists of three parts: the route on railway network, the origin, destination and transfer information and the new GPS location projected to railway network.

The total success rate of the map matching method is around 97.86% which is near to the estimated accuracy of 97.72% in the work of Witayangkurn et al. [38]. Some failures can be attributed to three main factors:

1 The adjacent GPS points are too far from each other without intermediate points. This often happens when the origin and destination are near the airport which indicates that the user travels by air instead of by train.

2 The adjacent GPS points belong to different railway system and there is no optimal route between it. This often happens when the sequential points belong to Shinkansen as well as the normal trains. It is also a wrong classification of other traffic modes like cars or bikes.

3 The topology information is wrong or lost in the railway network. Though the railway network data has been improved in these years and I have fixed some topology problems by myself including deleting and adding lines and modify the network, it still need to be improved in the future.

4.2.2 Data Interpolation

As GPS data suffer from signal loss, in order to acquire more precised trajectory data, data interpolation is conducted to simulate the lost data.

In this experiment a linear interpolation method through railway network is utilized on the route data generated from map matching. After interpolation, the time span between two adjacent GPS point in all trajectories is kept to around five minutes. Figure 4.2 and 4.3 show an example of map matching and data interpolation as well as comparison between raw data and the preprocessed data.

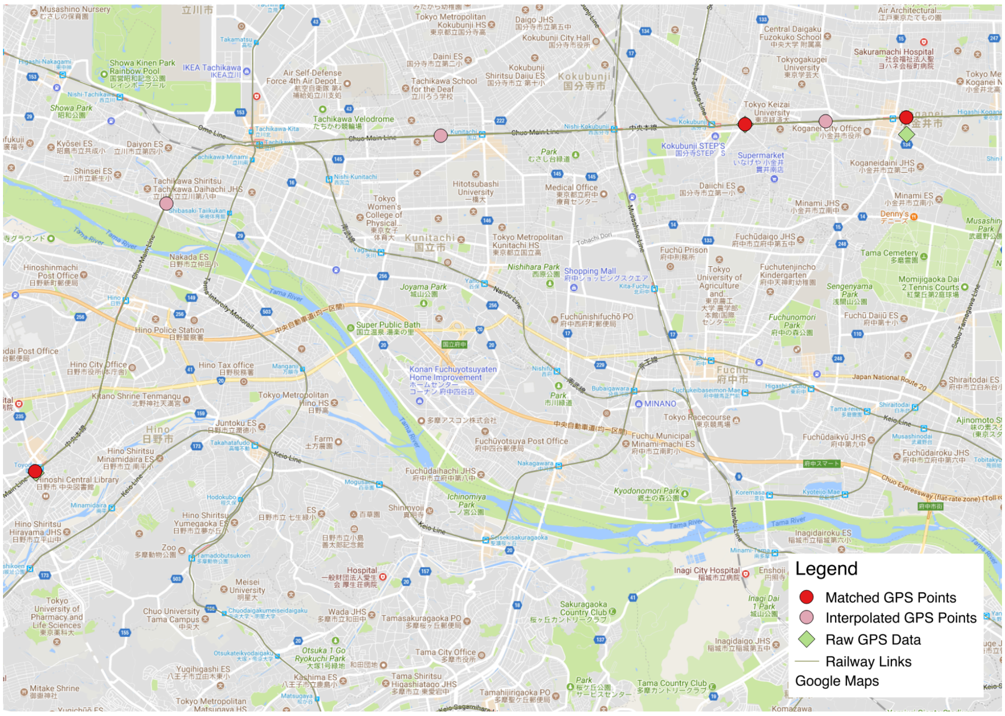
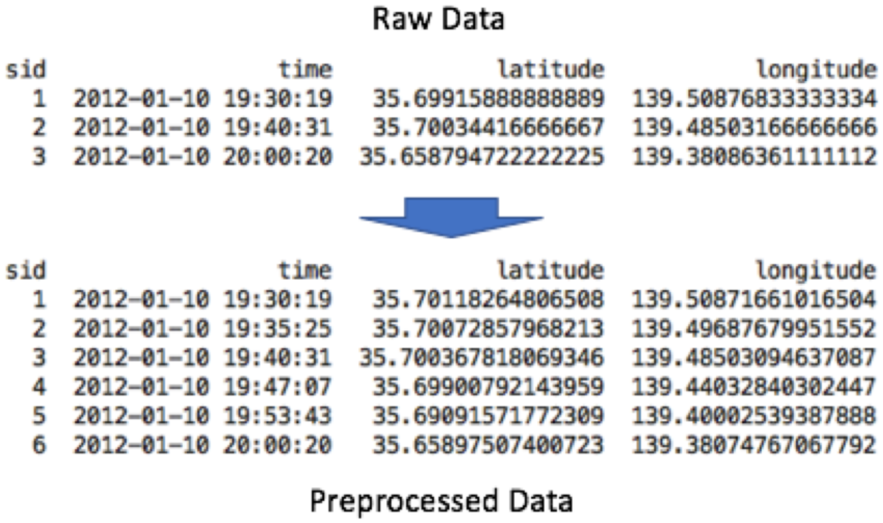


FIGURE 4.2: An example of map matching and data interpolation


Raw Data

```
sid                time           latitude        longitude
  1 2012-01-10 19:30:19   35.69915888888889 139.50876833333334
  2 2012-01-10 19:40:31   35.70034416666667 139.48503166666666
  3 2012-01-10 20:00:20  35.658794722222225 139.38086361111112
```

```
sid                time           latitude        longitude
  1 2012-01-10 19:30:19   35.70118264806508 139.5087166101650
  2 2012-01-10 19:35:25   35.70072857968213 139.49687679951552
  3 2012-01-10 19:40:31  35.700367818069346 139.48503094637087
  4 2012-01-10 19:47:07   35.69900792143959 139.44032840302447
  5 2012-01-10 19:53:43   35.69091571772309 139.40002539387888
  6 2012-01-10 20:00:20   35.65897507400723 139.38074767067792
```

Preprocessed Data

FIGURE 4.3: An comparison of the raw data and preprocessed data

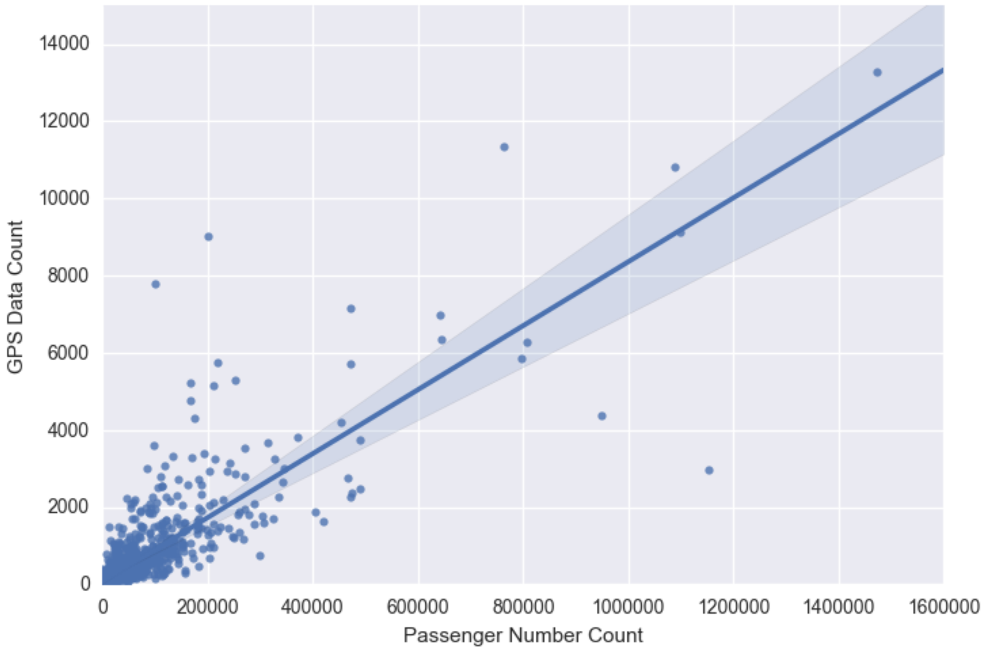


FIGURE 4.4: Regression plot of GPS data and passenger numbers

4.2.3 Evaluation and Statistical Analysis of Map Matching Results

4.2.3.1 Evaluation of Analyzing Railway Traffic Through GPS Data

In the research, as the kernel density estimation and anomaly detection is conducted GPS data, it is necessary to validate that the GPS data and map matching result can be utilized for traffic analysis. Here we randomly choosing five days and count the average daily numbers of each station that is a origin, destination or transfer station in one trajectory. The result is compared with the daily passenger number from NLNI.

Figure 4.4 shows the regression plot of these two data. The correlation coefficient of these two data is 0.832, the rank correlation coefficient is 0.908 and both of the p-values are less than 0.01, which shows that these two data are highly correlated, thus GPS data can be applied to analyzing railway traffic.

4.2.3.2 Statistical Analysis of Map Matching Results

A brief statistical analysis is conducted on the result of map matching. During 2012, the map matching extracts 580,207 users that can be regarded as the railway passengers in the data set and the total trajectory number of railway passengers is around 53.7 million. For each user,

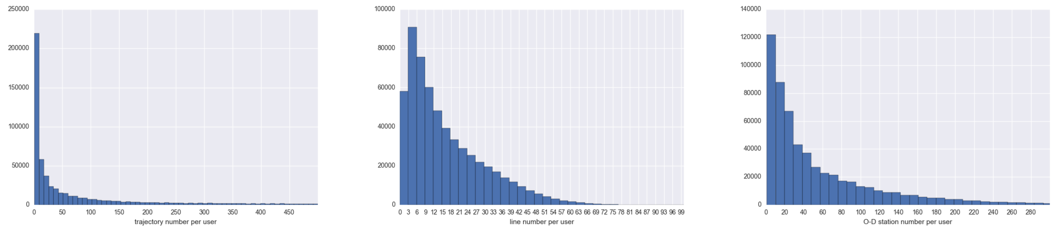


FIGURE 4.5: Histogram of trajectory number, line number and O-D number per user

the trajectory numbers, line numbers and station numbers are calculated. Figure 4.5 shows the histograms of the trajectory numbers, route numbers and OD station numbers. From the plots, it can be concluded that the trajectory numbers of most of the ZDC users are less than 10 and the numbers of lines and OD for each users also tend to be sparse in the whole datasets. Which indicates that the traffic routes for each railway passenger tend to be fixed in their daily lives. In addition, the percentage of trajectory on weekdays is 77.1%, which is larger than the ratio in uniform distribution (71.4%), showing that human behavior of railway transportation is more frequent in weekdays than in weekends and one possible explanation is that railway transportation plays an important role in people's commuting in Tokyo.

4.3 Experiments of Network Kernel Density Estimation

4.3.1 Parameter Settings

In the experiment, since the interval of GPS point is five minutes, network kernel density is calculated every five minutes to make sure that each GPS point of one GPS trajectory is calculated once in kernel density estimation. After testing some candidates of the parameters, the lixel size is set to 300 meters while the bandwidth is set to 1000 meters. To smooth the result of kernel density and simplify the result. the average kernel density of each thirty minutes is calculated in each day from 6:00 to 23:00.

4.3.2 Visualization and Discussion

Figure 4.6 shows four kernel density maps of central Tokyo which are respectively from 6:00 to 6:30, 9:00 to 9:30, 14:00 to 14:30 and 22:30 to 23:00 from the upper left to the lower right. The figure indicates that the links with high kernel density values are mainly near the transportation junctions, such as Shinjuku, Shibuya and Tokyo, which seem to always have high kernel density values. However, compare the density map around 6:00 with the map around 9:00, the kernel density maps obviously describe the movement of the commuters who live in outer Tokyo that the stations with highly density values transferred from the large stations in outer Tokyo to that in inner Tokyo.

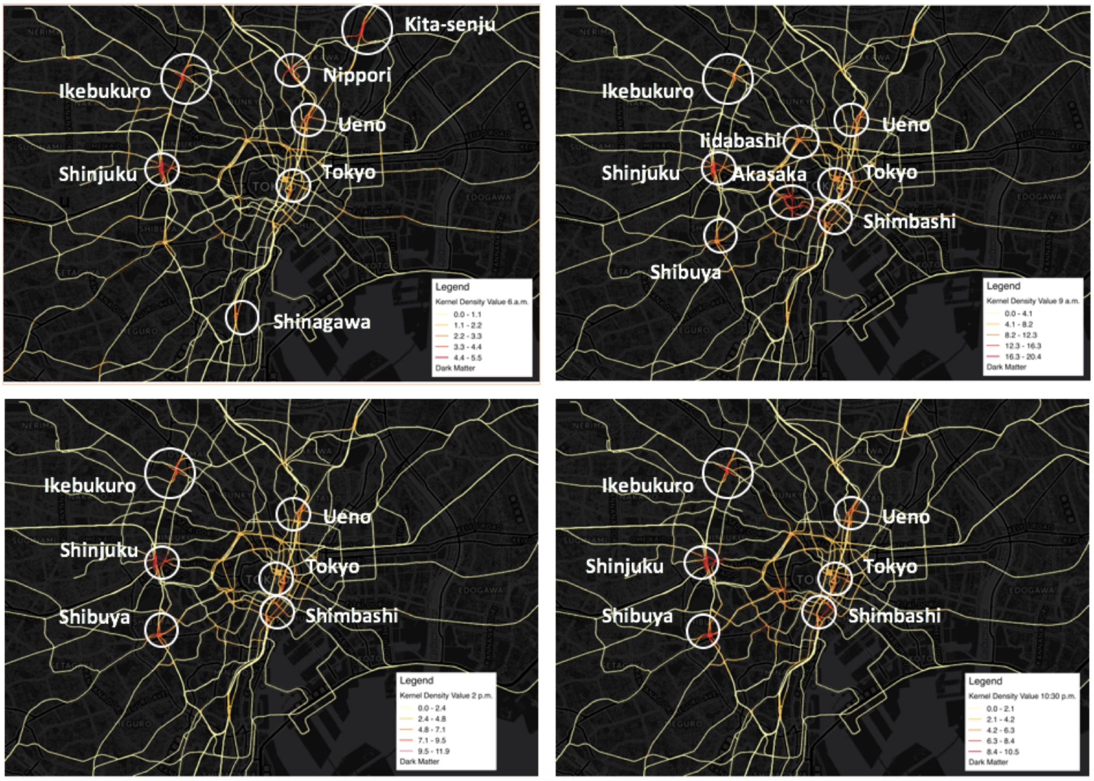


FIGURE 4.6: Four kernel density maps to visualize the kernel density values at different time

Figure 4.7 compares the line charts of kernel density values in a whole week in two links. The links chosen in this research is Kanamachi and Ueno, which are respectively represent the residential area and the transportation junction. As the figure indicates, both of the density sequences show the periodical patterns and it is clear that the kernel density pattern varies from weekdays and weekends in both of the stations. However, in residential area, there are no obvious peaks in weekends and the density value seems to be more fluctuated from time to time and the total volume decreases drastically, while in transportation junction, the density value is more stable at weekends and shows a higher volume of traffic in daytime than the weekdays except for the rush hours. In addition, at weekends, though the seqential patterns are similar on Saturday and Sunday, the traffic volume on Saturday is larger than Sunday.

In addition, the kernel density value of these two links in a holiday are tested on the golden week

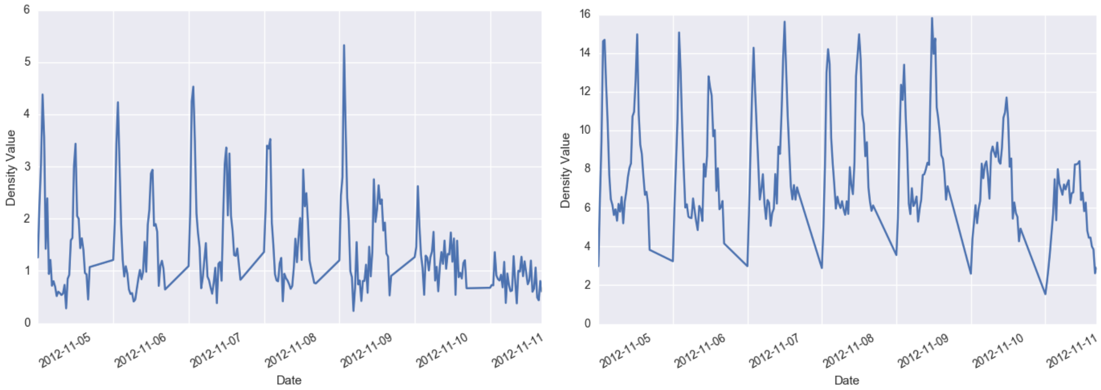


FIGURE 4.7: Kernel Density of the lixel near Kanamachi and Ueno in a normal week

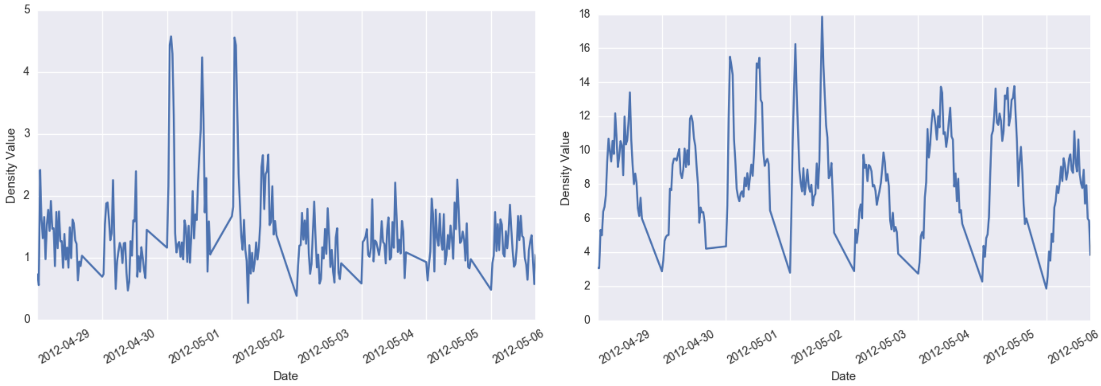


FIGURE 4.8: Comparison of statistics

and the result is shown in Figure 4.8. The figure indicates that except for May 1st and May 2nd which are not statutory holidays and the kernel density patterns are more like weekdays, other days during the golden week share the same sequential patterns with the weekends.

4.4 Anomaly Detection

In this research, the feasibility of the three anomaly detection methods are tested by different types of anomalies.

4.4.1 Anomaly Data and Training Datasets for Feasibility Test

As the anomaly can be divided by events and accidents on weekdays or weekends (holidays). In this experiment, several different types of anomalies listed in Table 4.4 are used to test the anomaly detection methods. As the previous research shows that the feature of kernel density

TABLE 4.4: A test data set for different kinds of anomalies

Date	Station Location	Cause	Severity	Train Dataset(PCA and Prediction)
2012-08-10 Fri (whole day)	Tokyo Big Site	Comiket	Around 200,000 visitors per day	2012-7-20∼2012-8-20(weekdays)
2012-04-02 Mon (around 21:05)	Nishi-Nippori	Human Accident	Around 37,000 passengers get influenced	2012-03-10∼2012-04-10(weekdays)
2012-06-19 Tue (whole day)	Whole Tokyo	Typhoon	Unknown	2012-06-05∼2012-07-05(weekdays)
2012-07-28 Sat (18:00 ∼ 20:00)	Near Sumida River	Firework Festival	Around 500,000 visitors per day	2012-06-01∼2012-09-01(weekends)

volume varies in weekdays and weekends and holidays share the similar features with weekends, in this research, the dates are divided into two types by weekdays and weekends for training anomaly detection models.

Different kinds of datasets are provided for different anomaly detection approaches. For statistic based approach, the training dataset $D(i,t)$ for each link i and time t can be represented by $d(i,t)_{day1}, d(i,t)_{day2}, d(i,t)_{dayn}$ for weekdays and weekends(include holidays) respectively. For PCA and prediction model based approaches, the training dataset is the whole data of the adjacent 20 days with the same type.

4.4.2 Anomaly Detection with PCA

4.4.2.1 Parameter Setting and Anomaly Detection Procedures

In PCA, the component number is defined by the explained variance ratio. Figure 4.9 shows the variance ratio of the test samples.

As is shown in the result, 94.3% of the variance can be explained in the first eight dimensions. Among these dimensions, the dimensions represent normal traffic and abnormal ones can be divided through the condition that if in the dimension there are data which meets that , then the following components are all represent anomalies. Figure 4.10 shows the graphs of the normal dimension and abnormal one.

After inversely transforming the normally projected data to the original axes, the estimated data is compared to the raw data and the data with differences more than δ is extracted. In this experiment, the threshold δ is defined by $3 \times \sigma$ where σ is the standard deviation of the

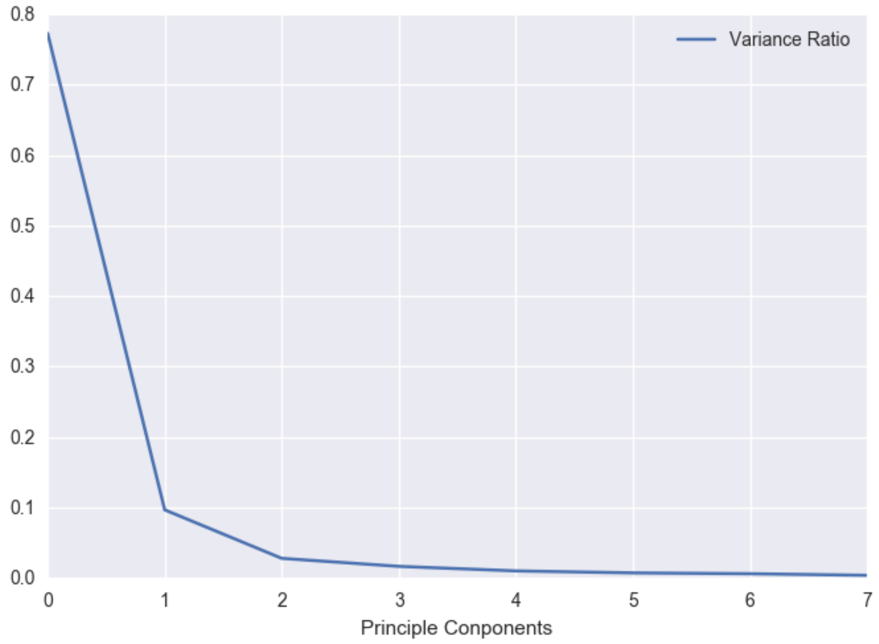


FIGURE 4.9: Variance ratio of each principle component

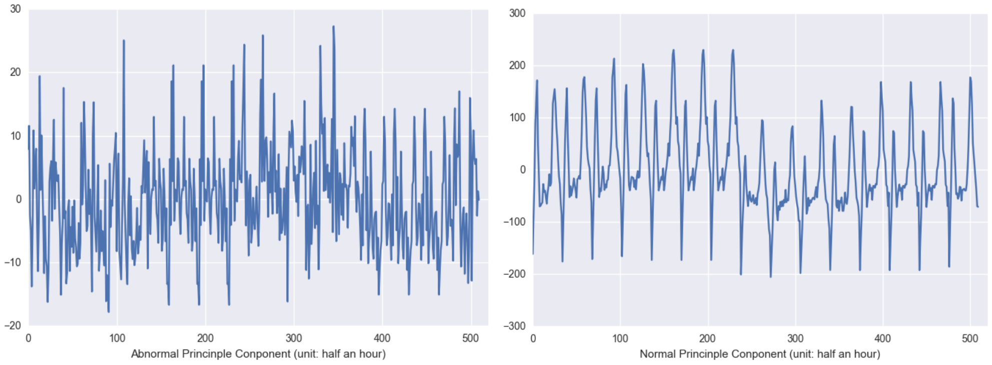


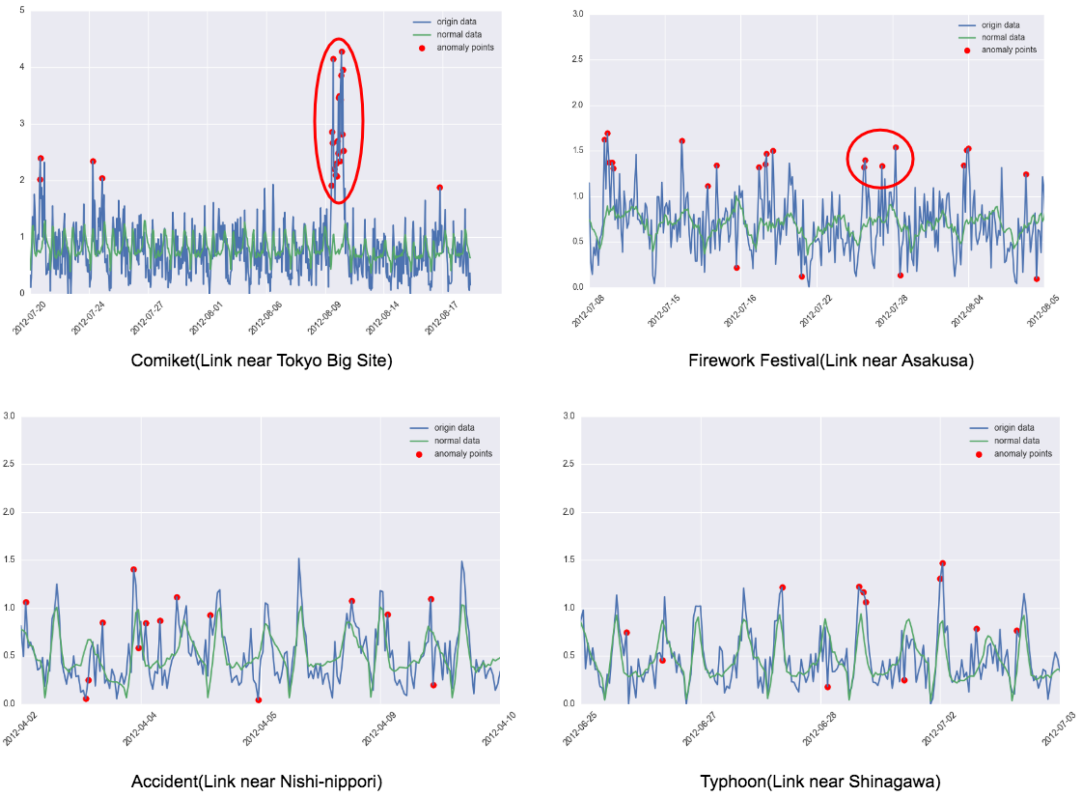
FIGURE 4.10: Comparison of abnormal and normal principle components (left: abnormal, right: normal)

differences of each link. Then the index of every anomaly data is acquired by looping over the whole data to compare their differences with the normal data.

4.4.2.2 Anomaly Detection Results and Discussions

The anomalies can be grouped and visualized by links which is shown in Figure 4.11.

The figure indicates that PCA can find out the anomalies in different datasets. It has a good performance to detect the big events such as Comiket and firework festival. However, it fails to detect the anomaly of Typhoon and railway accident, the failure of detecting typhoon may have the possible explanation on that the anomaly of Typhoon chosen in our dataset has little



Comiket(Link near Tokyo Big Site)

Firework Festival(Link near Asakusa)

Accident(Link near Nishi-nippori)

Typhoon(Link near Shinagawa)

FIGURE 4.11: PCA Anomaly detection results of four datasets

influence on railway traffic (there is nearly no information about railway delay in that day as a matter of fact) or PCA is difficult to find out the citywide anomalies that last for a long time. While the failure of detecting a sever railway accident may lie in that the number of passengers is not large enough or the time span of one kernel density value is too large to analyze an accident. In addition, the result of anomalies in each dataset shows that PCA tends to simulate the normal density value with a low peak, thus in some datasets, the peak kernel density values are extracted in several days even if these peaks seem to be regular ones.

4.4.3 Experiment on Prediction Model Based Anomaly Detection

In this section, we test different parameters on seasonal ARIMA model and build an optimal model with the result of evaluation. Then the optimal model is utilized for anomaly detection.

TABLE 4.5: Comparison of Different Models

Parameters	AIC	RMSE	Q test Probability
$(0,1,0)\times(0,1,1)_{34}$	505.192	0.388	<0.01
$(0,1,0)\times(1,1,0)_{34}$	606.071	0.424	<0.01
$(1,0,0)\times(0,1,1)_{34}$	285.137	0.279	0.01
$(1,0,0)\times(1,1,1)_{34}$	286.982	0.276	0.13
$(3,0,0)\times(0,1,1)_{34}$	**259.267**	**0.265**	**0.78**
$(4,0,0)\times(0,1,1)_{34}$	261.247	0.267	**0.78**

4.4.3.1 Parameter Settings and Model Selection

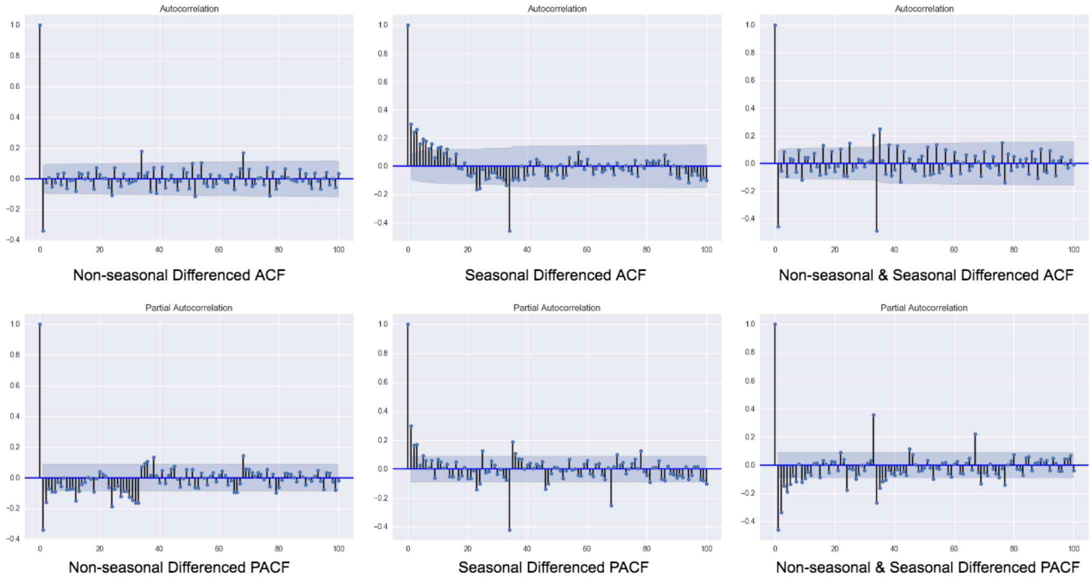
As is introduced in Chapter 3, seasonal ARIMA model can be represented by $ARIMA(p, d, q)\times(P, D, Q)_s$. The parameters in these model can be listed as follows:

1 As for each day, there are 34 kernel density values in each link for anomaly detection, so the period denoted by s for the dataset is 34.

2 The difference order d and D should be set to ensure the stationarization of the sequence. Here, the d and D can be defined by visualization through the data and the measurement of stationarization can be realized through the Dickey-Fuller test Dickey and Fuller [9]. In this experiment, we test three kinds of diffrencing orders, which are respectively one order difference $(0,1,0)\times(0,0,0)$, one order seasonal difference $(0,0,0)\times(0,1,0)$ as well as one order difference with one order seasonal difference $(0,1,0)\times(0,1,0)$. The Dickey-Fuller test result of all three sequences show a p-value less than 0.01, which indicates that all of the sequences are stationarized.

3 The terms of AR and SAR (p and P) is identified by PACF and the terms of MA and SMA (q and Q) is identified by ACF. Figure 4.12 shows the ACF and PACF of the three sequences. The parameters should be set to the lag where a sharp cutoff happens.

After judging from the figures, different tuples of parameters are tested to build the model, then the models with different parameters are utilized to predict the kernel density values. In this experiment, the test sequence data is randomly chosen in some links from the total kernel density values. The Table 4.5 shows the model performance under the indicators of AIC, RMSE and Ljung-Box Q test in different parameters. The result shows that $ARIMA(3,0,0)\times(0,1,1)_{34}$ model is the optimal model for predicting density values.



Non-seasonal Differenced ACF

Seasonal Differenced ACF

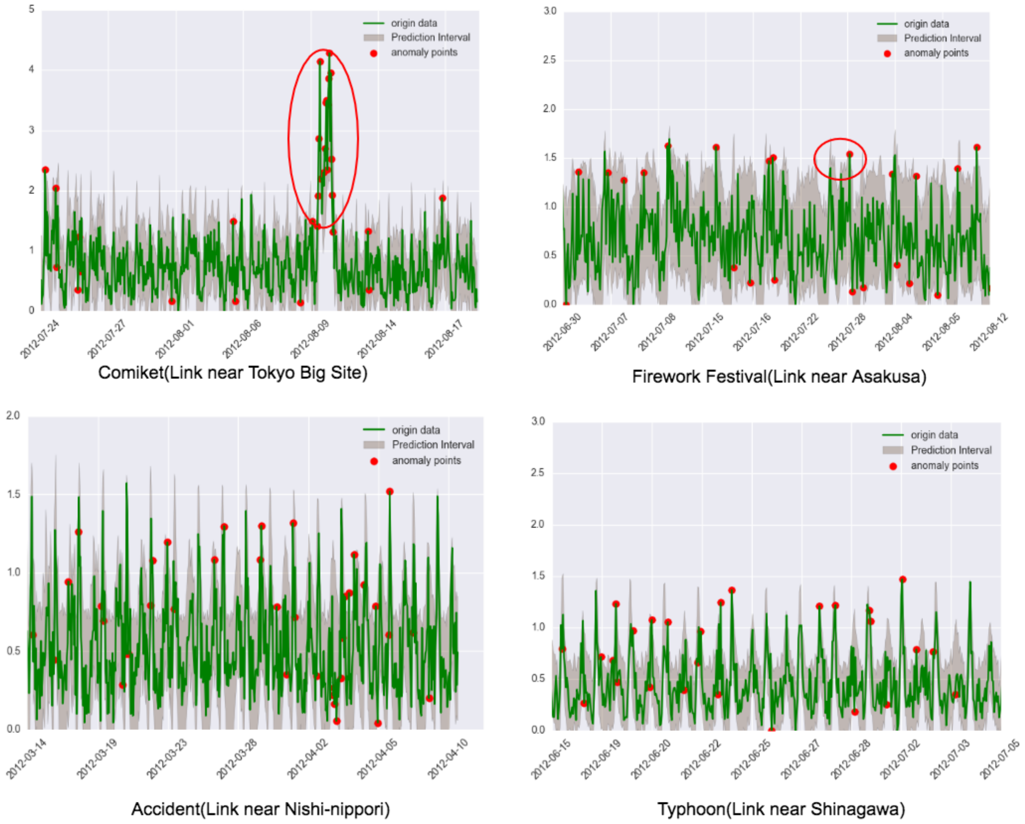
Non-seasonal & Seasonal Differenced ACF

Non-seasonal Differenced PACF

Seasonal Differenced PACF

Non-seasonal & Seasonal Differenced PACF

FIGURE 4.12: ACF and PACF of differenced sequences



Comiket(Link near Tokyo Big Site)

Firework Festival(Link near Asakusa)

Accident(Link near Nishi-nippori)

Typhoon(Link near Shinagawa)

FIGURE 4.13: Seasonal ARIMA Anomaly detection results of four datasets

4.4.3.2 Anomaly Detection Results and Discussions

The result of anomaly detection via seasonal ARIMA is demonstrated in Figure 4.13. The result of anomaly detection with seasonal ARIMA is similar to the PCA result as it detects the anomalies in Comiket and firework festival as well and fails to extract the patterns representing a railway accident or typhoon. Comparing to PCA anomaly detection, seasonal ARIMA model marks more kernel density values as anomalies especially those with a smaller value than the observed data and is more sensitive to a sharp curve of value change, which can be proved by the prediction interval during Comiket where the bound of the interval is more closed to the observation data than the normal value in PCA model. This is because that seasonal ARIMA not only takes the periodical patterns into consideration but also predict the value using the values from nearby lags.

4.4.3.3 Causes of Failures: A Further Exploration

In order to make out whether the failure for detecting railway accidents is by chance or not, we test our model in other human accident cases. Unfortunately, till the submission of this thesis, we still cannot find out an accident case that can be successfully detected through our model. The main factor lies in that there is little difference between kernel density value during those accidents and the normal ones. This may result from a lot of factors. First of all, the kernel density is represented by an average value of each 30 minutes, which reduces the significance of the value. Besides, the data volume of GPS data might be not enough for detecting such kinds of anomalies. Though from the reports we can find that one case of railway accident may influence 30,000 passengers or more, however since railway transportation system is very complicated, the 30,000 passengers may not be gathered in the station where the accident happens but distributed sparsely among the railway network, which may not be reflected from the traffic volume of one station or one railway link (the case of an accident is totally different from that of some big events where people are gathered in some specific stations). In addition, since this research doesn't track each car, it cannot distinguish the kernel density of the running car in one link to the ones that stopped in the same link due to an accident, thus make it difficult to be applied to the anomaly detection of some short time emergencies.

Chapter 5

Conclusions and Future Work

5.1 Conclusions and Discussions

In this research, a system is proposed to utilize GPS data to analyze railway traffic. The result of the experiment indicates that the system is valid to analyze railway traffic and is feasible to detect some of the anomalies. Comparing to other research, the main contribution of this research is as follows:

- Big and heterogeneous data: this system utilizes heterogeneous data from multiple data sources which includes the railway network data and GPS trajectories.
- Network based data analysis: this research is conducted through railway network, which is a more precise description of the real world than grid-based researches.

However, there are some aspects where the researches should be improved, which can be listed as follows:

1. Timetable is not available in this research while it is very important for railway traffic analysis.
2. The evaluation of map matching lacks the ground truth data thus the percentage of recall is not calculated in this research even though it is a very important indicator.
3. In anomaly detection module, only the feasibility of anomaly detection models are tested, while the detection precision and accuracy is not measured as the ground truth data is not available.

5.2 Future Work

The future work of this research will focus on two aspects. On one hand, the data generated in this research such as the origin and destination information, the statistical result of the railway network as well as the interpolated GPS data will be utilized in the future research such as the analysis of passengers' route choices and the subsequent behavior after getting off the train. On the other hand, the current methods used in this research are required to be improved. Firstly, the prediction model would be improved to use some machine learning and deep learning technologies. Besides, the pattern of the anomalies will be extracted to further understand the the abnormal situations.

Appendix A

Additional Specification for Data Preprocessing

A.1 Tools developed for this research

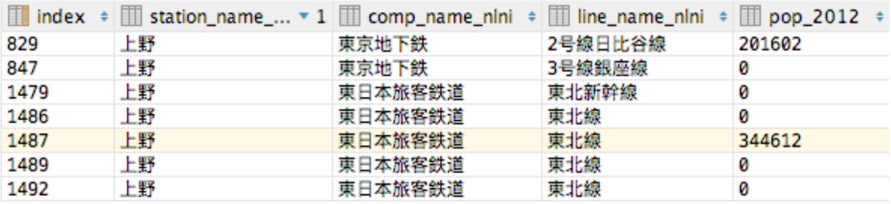
Manual interpretation tool: In this research, as there are some trajectories that are failed to mark as railway passengers, to identify these points, a simple browser tool is created to visualize a trajectory on AWS server. The URL is `http://s3-ap-northeast-1.amazonaws.com/xiageodata/showPts.html`.

API for real-time GPS map matching: Since the code of map matching and kernel density estimation is written in Java while the code for some data preprocessing and anomaly detection is written in Python. Here an API is created for map matching GPS data with the trajectory inputs. This server application is built by Java with the application of Glassfish. The URL format of this API is as follows:

```
https://hostname:portname/test_gf_war_exploded/ipuc?track={track_input}
```

An request With the parameters of GPS trajectory returns a matched trajectory and the railway information.

These tools will be improved in the future works too.



index	station_name_... 1	comp_name_nlni	line_name_nlni	pop_2012
829	上野	東京地下鉄	2号線日比谷線	201602
847	上野	東京地下鉄	3号線銀座線	0
1479	上野	東日本旅客鉄道	東北新幹線	0
1486	上野	東日本旅客鉄道	東北線	0
1487	上野	東日本旅客鉄道	東北線	344612
1489	上野	東日本旅客鉄道	東北線	0
1492	上野	東日本旅客鉄道	東北線	0

FIGURE A.1: Visualization of the number of passengers in Ueno. The table indicate that the number of passengers is calculated only once for the lines in the same company.

A.2 Matching Data from NLNI to our data set

This work is conducted for evaluating GPS points and map matching result with the passenger number data received from NLNI. As NLNI and our railway data set have different naming and code system, it is difficult to match these two data set by the attributes. To overcome this problem, spatial join method is used to get the nearest railway station point in our data set for each station data of NLNI. In Tokyo area, the number of station points is 1828, and the spatial matching accuracy is around 92.7%. The failure lies in that some stations are too close to each other to distinguish.

In NLNI data, the number of each station of the same company is calculated once, which means that the passenger number of some stations in some stations are marked with with zero if there are more than one line transferred in this station. As is shown in Figure. Besides, in Japan, some rail lines have the system of direct service and sometimes several companies share the same ticket gate, thus in some stations the passenger number calculated only once even in different companies.

Here to solve this problem. The topology information of the station is utilized to group the same station of different companies. For both data from NLNI and GPS trajectory, the statistical analysis is conducted on a station group level.

Bibliography

[1] Abdel-Aty, M. A. and Radwan, A. E. (2000). Modeling traffic accident occurrence and involvement. *Accident Analysis & Prevention*, 32(5):633–642.

[2] Akaike, H. (1974). A new look at the statistical model identification. *IEEE transactions on automatic control*, 19(6):716–723.

[3] Asakura, Y., Iryo, T., Nakajima, Y., Kusakabe, T., Takagi, Y., and Kashiwadani, M. (2008). Behavioural analysis of railway passengers using smart card data. *WIT Transactions on The Built Environment*, 101:599–608.

[4] Authority, L. T. et al. (2014). Passenger transport mode shares in world cities. *Journeys*, pages 54–64.

[5] Bernstein, D. and Kornhauser, A. (1998). An introduction to map matching for personal navigation assistants.

[6] Bishop, C. M. (2006). *Pattern recognition and machine learning*. springer.

[7] Box, G. E., Jenkins, G. M., Reinsel, G. C., and Ljung, G. M. (2015). *Time series analysis: forecasting and control*. John Wiley & Sons.

[8] Chen, Q., Song, X., Yamada, H., and Shibasaki, R. (2016). Learning deep representation from big and heterogeneous data for traffic accident inference. In *AAAI*, pages 338–344.

[9] Dickey, D. A. and Fuller, W. A. (1979). Distribution of the estimators for autoregressive time series with a unit root. *Journal of the American statistical association*, 74(366a):427–431.

[10] Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271.

[11] Disser, Y., Müller-Hannemann, M., and Schnee, M. (2008). Multi-criteria shortest paths in time-dependent train networks. In *International Workshop on Experimental and Efficient Algorithms*, pages 347–361. Springer.

[12] Fan, Z., Song, X., Shibasaki, R., and Adachi, R. (2015). Citymomentum: an online approach for crowd behavior prediction at a citywide level. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 559–569. ACM.

[13] Greenfeld, J. S. (2002). Matching gps observations to locations on a digital map. In *81th annual meeting of the transportation research board*, volume 1, pages 164–173.

[14] Gustafsson, F., Gunnarsson, F., Bergman, N., Forssell, U., Jansson, J., Karlsson, R., and Nordlund, P.-J. (2002). Particle filters for positioning, navigation, and tracking. *IEEE Transactions on signal processing*, 50(2):425–437.

[15] Hamed, M. M., Al-Masaeid, H. R., and Said, Z. M. B. (1995). Short-term prediction of traffic volume in urban arterials. *Journal of Transportation Engineering*, 121(3):249–254.

[16] Hiroshi, K., Yoshihide, S., and Takehiro, K. (2013). Development of open railway dataset towards people flow reconstruction. In *The 22th conference on GIS Association of Japan*.

[17] Huang, H., Al-Azzawi, H., and Brani, H. (2014). Network Traffic Anomaly Detection. *ArXiv e-prints*.

[18] Kaklij, S. (2015). Mining gps data for traffic congestion detection and prediction. *International Journal of Science and Research (IJSR)*, pages 876–880.

[19] Kusakabe, T., Iryo, T., and Asakura, Y. (2010). Estimation method for railway passengers train choice behavior with smart card transaction data. *Transportation*, 37(5):731–749.

[20] Lakhina, A., Crovella, M., and Diot, C. (2004). Diagnosing network-wide traffic anomalies. In *ACM SIGCOMM Computer Communication Review*, volume 34, pages 219–230. ACM.

[21] Lou, Y., Zhang, C., Zheng, Y., Xie, X., Wang, W., and Huang, Y. (2009). Map-matching for low-sampling-rate gps trajectories. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 352–361. ACM.

[22] Lu, D. and Schnieder, E. (2015). Performance evaluation of gnss for train localization. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):1054–1059.

[23] Ma, X., Yu, H., Wang, Y., and Wang, Y. (2015). Large-scale transportation network congestion evolution prediction using deep learning theory. *PloS one*, 10(3):e0119044.

[24] Mitchell, R. B. and Rapkin, C. (1954). Urban traffic–a function of land use.

[25] Newson, P. and Krumm, J. (2009). Hidden markov map matching through noise and sparseness. In *Proceedings of the 17th ACM SIGSPATIAL international conference on advances in geographic information systems*, pages 336–343. ACM.

[26] Ochieng, W. Y., Quddus, M., and Noland, R. B. (2003). Map-matching in complex urban road networks. *Revista Brasileira de Cartografia*, 2(55).

[27] Pan, B., Zheng, Y., Wilkie, D., and Shahabi, C. (2013). Crowd sensing of traffic anomalies based on human mobility and social media. In *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, pages 344–353. ACM.

[28] Patcha, A. and Park, J.-M. (2007). An overview of anomaly detection techniques: Existing solutions and latest technological trends. *Computer networks*, 51(12):3448–3470.

[29] Pelletier, M.-P., Trépanier, M., and Morency, C. (2011). Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557–568.

[30] Prassler, E., Scholz, J., and Elfes, A. (1999). Tracking people in a railway station during rush-hour. In *International Conference on Computer Vision Systems*, pages 162–179. Springer.

[31] Quddus, M. A., Noland, R. B., and Ochieng, W. Y. (2005). Validation of map matching algorithms using high precision positioning with gps. *The Journal of Navigation*, 58(2):257–271.

[32] Quddus, M. A., Ochieng, W. Y., and Noland, R. B. (2007). Current map-matching algorithms for transport applications: State-of-the art and future research directions. *Transportation research part c: Emerging technologies*, 15(5):312–328.

[33] Rabatel, J., Bringay, S., and Poncelet, P. (2009). So_mad: Sensor mining for anomaly detection in railway data. In *Industrial Conference on Data Mining*, pages 191–205. Springer.

[34] Song, X., Zhang, Q., Sekimoto, Y., Shibasaki, R., Yuan, N. J., and Xie, X. (2016). Prediction and simulation of human mobility following natural disasters. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 8(2):29.

[35] Sun, L., Lu, Y., Jin, J. G., Lee, D.-H., and Axhausen, K. W. (2015). An integrated bayesian approach for passenger flow assignment in metro networks. *Transportation Research Part C: Emerging Technologies*, 52:116–131.

[36] Viterbi, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *IEEE transactions on Information Theory*, 13(2):260–269.

[37] Wen, H., Hu, Z., Guo, J., Zhu, L., and Sun, J. (2008). Operational analysis on beijing road network during the olympic games. *Journal of Transportation Systems Engineering and Information Technology*, 8(6):32–37.

[38] Witayangkurn, A., Horanont, T., Ono, N., Sekimoto, Y., and Shibasaki, R. (2013). Trip reconstruction and transportation mode extraction on low data rate gps data from mobile phone. In *Proceedings of the international conference on computers in urban planning and urban management (CUPUM 2013)*, pages 1–19.

[39] Xie, Z. and Yan, J. (2008). Kernel density estimation of traffic accidents in a network space. *Computers, Environment and Urban Systems*, 32(5):396–406.

[40] Yang, D., Cai, B., and Yuan, Y. (2003). An improved map-matching algorithm used in vehicle navigation system. In *Intelligent Transportation Systems, 2003. Proceedings. 2003 IEEE*, volume 2, pages 1246–1250. IEEE.

[41] Yu, M. (2006). *Improved positioning of land vehicle in ITS using digital map and other accessory information*. PhD thesis, The Hong Kong Polytechnic University.

[42] Zhang, J., Zheng, Y., and Qi, D. (2016). Deep spatio-temporal residual networks for citywide crowd flows prediction. *arXiv preprint arXiv:1610.00081*.

[43] Zhao, H. and Shibasaki, R. (2005). A novel system for tracking pedestrians using multiple single-row laser-range scanners. *IEEE Transactions on systems, man, and cybernetics-Part A: systems and humans*, 35(2):283–291.

２０１７年度　修士論文　**GPS 軌跡による都市規模の鉄道交通分析システム**

夏　天琦