MODELING FOR CLUSTER-BASED CORRELATION OF SAFETY DRIVING EVENTS WITH TIME AND LOCATION

A Dissertation

Presented to
the Faculty of the Department of Mechanical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Mechanical Engineering

by

Guoyan Cao

December 2015

MODELING FOR CLUSTER-BASED CORRELATION OF SAFETY DRIVING EVENTS WITH TIME AND LOCATION

Guoyan Cao

Approved:

Chair of the Committee
Dr. Karolos M. Grigoriadis, Professor
Mechanical Engineering

Committee Members:

Dr. Matthew A. Franchek, Professor
Mechanical Engineering

Dr. Jagannatha R. Rao, Professor
Mechanical Engineering

Dr. Gangbing Song, Professor
Mechanical Engineering

Dr. Michael Nikolaou, Professor
Chemical Engineering

Dr. Suresh K. Khator, Associate Dean
Cullen College of Engineering

Dr. Pradeep Sharma, Professor and Chair
Mechanical Engineering

ACKNOWLEDGEMENT

It is my pleasure to thank my advisor, Dr. Karolos Grigoriadis, for his guidance and for constant support and encouragement. Dr. Grigoriadis was patient with my research and he always encouraged me to learn, try and improve the scientific methods at the times when the research process was not going so well. He also gave me lots of instruction on my scientific writing. I would not be able to complete my doctoral study without his encouragement and direction. I also want to express my gratitude to Dr. Matthew Franchek, whose advice, insight and support helped me a lot on my research progress. Dr. Francheck taught me how to have a professional attitude on my scientific research and work. I really cherish the time I worked with Dr. Grigoriadis and Dr. Franchek and appreciate their financial support.

Both the substance and exposition of the results in this dissertation have benefitted from the cooperation and enlighten conversation with John Michelini, Dr. Behrouz Ebrhimi, and Dr. Yaw Nyanteh. I am grateful to all of them for their help and insight.

I would be remiss if I did not express my abiding gratitude to committee members, Dr. Jagannatha Rao, Dr. Michael Nikolaou, and Dr. Gangbing Song as well as Dr. Ben Jansen and Dr. Edward Kao, for their excellent courses which helped to lay the knowledge and foundation for all that has come since.

Finally, I could not have done any of this without the love and support of my friends and family: friends and elders in Dumble Gathering, who always encourage me to overcome the difficulties during my pursuit of the degree and direct me to set up the right attitude for everything I meet; and of course my two brothers and my parents, for whom words are simply not enough.

MODELING FOR CLUSTER-BASED CORRELATION OF SAFETY DRIVING EVENTS WITH TIME AND LOCATION

An Abstract
of a
A Dissertation

Presented to
the Faculty of the Department of Mechanical Engineering

University of Houston

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy

in Mechanical Engineering

by

Guoyan Cao

December 2015

Abstract

Advanced driver assistance systems (ADAS) are general systems developed to enhance safety driving of an individual vehicle. In this dissertation, a type of ADAS, named safety driving assistance system, is proposed to lower the potential driving risk caused by severe driving events to enhance the safety driving of an individual vehicle. The safety driving assistance system identifies the severe driving events and the occurrence of the events to infrastructure, notifies the events temporal and spatial concentration and variation, and models the concentration and variation in form of the event count data time series modeling to evaluate and predict the potential driving risk.

Safety driving assistance system uses a designed intelligent analyzer which is a systematic procedure to identify severe driving events occurrence correlation with time and location. The proposed procedure, which is constructed based on batch clustering and real-time clustering techniques, incorporates historical and real-time data to recognize the time and location of severe driving events and simulate the variation of severe driving events distribution and concentration with respective to time and location, respectively. Batch clustering is implemented with the combination of subtractive clustering and fuzzy c-means clustering to generate clusters representing the initial correlation patterns. Real-time clustering is then developed to create and update real-time correlation patterns on the foundation of the batch clustering using evolving Gustafson-Kessel Like (eGKL) algorithm. Historical and real-time data of operating vehicles acquired from data acquisition and wireless communication platform (DAP), constructed by Ford Motor company, are used to validate the proposed strategy. Batch clustering reveals the severe driving events distribution and concentration in geographical domain at different time. Real-time clustering provides and updates the variation of the intra-correlation and inter-correlation of different regions. Driver can be notified of the potential severe driving locations through maps showing the driving routes. Through the variation of the correlation, drivers can recognize the events occurrence at different time and location.

The variation of the correlation can be presented by events count data time series. Four models are proposed to describe time series of event count data in a region and to predict the future event count in the region. ARIMA and STARIMA modeling procedures account more on the aspect of the time series autocorreation in temporal domain and spatial domain. Generalized linear model (GLM) with Poisson distribution accounts more on the aspect of the natural distribution property of severe driving event. Hidden Markov Model (HMM) is attempted to describe and predict the event count data in a deep reasoning that the stochastic process of severe driving event occurrence in different regions is generated from different Poisson distribution components following certain transition logic. The four models are all validated by actual data and demonstrated their adequacy.

Table of Contents

List of Figures

List of Tables

Nomenclature

$\boldsymbol{X_k}$, a server driving data point.

$\boldsymbol{c}$, the number of clusters.

$\boldsymbol{N}$, the number of data points.

$\boldsymbol{n}$, the number of dimension space of a data point.

$\boldsymbol{P_k}$, the potential of data point $\boldsymbol{X_k}$ calculated in subtractive clustering.

$\boldsymbol{r_a}$, the radius defining a neighborhood when calculate the potential of a data point $\boldsymbol{X_k}$.

$\boldsymbol{X_i^*}$, the $\boldsymbol{i}$-th cluster center calculated from subtractive clustering.

$\boldsymbol{P_i^*}$, the potential of the $\boldsymbol{i}$-th cluster center $\boldsymbol{X_i^*}$.

$\boldsymbol{r_b}$, the radius defining the neighborhood updating the potentials of data points.

$\overline{\boldsymbol{\varepsilon}}$, an acceptance ratio to determine the acquirement of a new cluster center.

$\underline{\boldsymbol{\varepsilon}}$, a reject ratio to determine the abandonment of a new cluster center.

$\boldsymbol{V_i}$, the cluster center of $\boldsymbol{i}$-th cluster.

$\boldsymbol{d_{ik}}$, the distance between data point $\boldsymbol{X_k}$ with a cluster center $\boldsymbol{V_i}$.

$\boldsymbol{F_i}$, the fuzzy covariance matrix of $\boldsymbol{i}$-th cluster.

$\boldsymbol{u_{ik}}$, membership of the data point $\boldsymbol{X_k}$ to $\boldsymbol{i}$-th cluster.

$\boldsymbol{q}$, a parameter controls the degree of fuzziness in fuzzy c-means clustering.

$\boldsymbol{r_i}$, the radius of the $\boldsymbol{i}$-th cluster.

$\boldsymbol{\chi^2_{n,\beta}}$, the Chi-square distribution constant with the degree of freedom, $\boldsymbol{n}$, and false alarm, $\boldsymbol{\beta}$.

$\boldsymbol{D_{ik}}$, the unitless distance parameter between point $\boldsymbol{X_k}$ with a cluster center $\boldsymbol{V_i}$.

$\boldsymbol{M_{min}}$, the minimal number of points in a cluster.

X_{k1}, the latitude coordinates of a severe driving event data point.

X_{k2}, the longitudinal coordinates of a severe driving event data point.

T_k, the occurrence time of a severe driving event data point.

A_k, the deacceleration of a severe braking event data point.

$c_{\overline{\varepsilon}}$, the number of the clusters whose center potential is greater than $\overline{\varepsilon}$.

$c_{\underline{\varepsilon}}$, the number of the clusters whose center potential is less than $\overline{\varepsilon}$, but greater than $\underline{\varepsilon}$.

v^i, the i-th cluster from time domain clustering.

$\bar{X}_k^i$, a daily-compaction severe driving data point in the cluster v^i.

$v^{i,j}$, the j-th subbranch cluster from 1st geographical domain clustering from the i-th cluster, v^i, with $j = 1, \ldots, m_i$, and m_i is the number of clusters resulting from the cluster, v^i.

$\bar{X}_k^{i,j}$, a daily-compaction severe driving data point in the cluster $v^{i,j}$.

$v^{i,j,h}$, the h-th subbranch cluster from 2nd geographical domain clustering from j-th subbranch cluster, $v^{i,j}$, from the i-th cluster, v^i, where $h = 1, \ldots, w_{i,j}$, and $w_{i,j}$ is the number of clusters resulting from the cluster, $v^{i,j}$.

$\bar{X}_k^{i,j,h}$, a daily-compaction severe driving data point in the cluster $v^{i,j,h}$.

$V_{i,j}$, the cluster center of cluster $v^{i,j}$.

$V_{i,j,h}$, the cluster center of cluster $v^{i,j,h}$.

Z_t, the time series of severe driving event count data.

μ_t, the mean function of a time series Z_t.

$\gamma_{t,s}$, the autocovariance function of the time series Z_t.

$\rho_{t,s}$, the autocorrelation function of the time series Z_t.

ϕ_{kk}, the partial autocorrelation function of the time series Z_t.

$Z_t^{(7)}$, denotes the stochastic process to generate the time series of severe braking events count data in region 7 from 10/01/2013 to 02/25/2014.

$L(\boldsymbol{\theta}|Z_t)$,
the likelihood function with unknown parameter $\boldsymbol{\theta}$ and the observed Z_t.

$L^{(l)}$, denotes the spatial lag operator of spatial order l.

$\gamma_{lk}(s)$, denotes space-time covariance between l^{th} and k^{th} order neighbors at time lag s.

$\rho_{lk}(s)$, denotes space-time autocorrelation between l^{th} and k^{th} order neighbors at time lag s.

Chapter 1

Introduction and Motivation

Advanced driver assistance system (ADAS) has been widely used in vehicles in enhancing the vehicles' safety driving feature. In the past decade, the ADAS is one of the most remarkable elements and fastest-growing segments in automotive design. The traditional ADAS technologies are based on vision or camera systems, sensor networks and active control, such as, automate lighting, automate braking, GPS or traffic warning, alerting driver to keep driving in the correcting lane, and showing the cars in blind spots to assist the driver to avoid potential risks of collisions and accidents. With the development of cloud computing and wireless communication, the vehicle data analysis with artificial intelligent procedures is able to upgrade the ADAS and supply a more powerful real-time assistance.

This dissertation proposes a safety driving assistance system, which is a developing form of ADAS. The safety driving assistance system notifies potential risk caused by severe driving events to enhance the safe driving. The safety driving assistance system is designed based on the technology development of real-time communication between vehicles, database, and a cloud computing platform: vehicle real-time driving data are transmitted to the cloud database through wireless communication; in the cloud platform, intelligent algorithms are implemented to extract safety driving information by cloud computation; then the information is transmitted back and reported to the drivers. Here, the cloud platform can be a remote computing center or a computer-unit embedded in each vehicle, where the safety driving assistance system is installed and implemented.

Thus, the safety driving assistance system can be regarded as a management system for processing the loop of severe driving data. It is essentially composed by four major steps in terms of data/information flow: relevant safety driving data acquisition from database, data analysis to extract information for driving assistance, driving assistance implementation, and transmission driving assistance information to database for sharing with other vehicles. The core part of the data flow is data analysis. In this dissertation, we focus on the data analysis step. We propose a systematic correlation identification procedure to extract valuable information for assisting drivers.

The remainder of this chapter introduces the background of the development of advanced driver assistance systems and the relative technology in intelligent transportation system (section 1.1). Followed by that, the basic architecture of the safety driving assistance system and the relative severe driving events are described. At last, the structure of the dissertation is presented as well as the specific contributions presented at last.

1.1 Background

Nowadays, with more and more vehicles used daily around the world, safe driving becomes a major concern. Many measures have been taken to improve drivers' safety and traffic condition in accordance with technical methods and governmental regulations [1], [2], [3]. On March 31st, 2014, the U.S. Department of Transportation's National Highway Traffic Safety Administration (NHTSA) announced that it will require all new vehicles under 1000 pounds to be equipped with rear view cameras by May 2018 [4]. It is an example of mandatorily requiring the usage of advanced driver assistance system (ADAS) in latest vehicles to assure drivers' safety driving. ADAS has been gradually used since early 1990s [5]. For example, automotive companies started using camera-assisted detection technologies in ADAS to alert drivers about potential driving dangers, such as, lane-keeping support system, driver monitoring system, and blind spot monitoring system. Later on, radar-assisted technologies employed in ADAS were used in vehicles to reduce the risk of driving at different surroundings, such as automotive night vision, collision avoidance system, and adaptive cruise control. The traditional ADAS was based on hardware usage in individual vehicles and did not pay much attention on vehicle driving data. With the development of wireless communication technology and cloud storage and computing technology, vehicle driving data potentially plays a significant role in exploring and mining useful information to upgrade the safety driving. Recently, the technology with combination of Automotive Navigation System (ANS) and Mobile Millennium System (MMS) provided powerful capability to navigate drivers and provide the real-time traffic indication, where ANS is supported by map database and vehicle GPS data, which are from high-accuracy tracking of GPS devices placed in vehicles. In addition, Vehicular Communication System (VCS) and the Intelligent Transportation System (ITS) are proposed to integrate communication between vehicles and fixed roadside units to achieve safety driving assistance [6], [7], [8]. Those systems transform great amount of traffic data “visible” to help drivers make safe driving decision. In 2012, Ford Motor Company Research and Development Center constructed a platform called Data Acquisition and communications Platform (DAP). DAP provides a mechanism for storage, analysis and reporting of operating vehicles data by interfacing to the GPS, telematics server (via various wireless communication methods), vehicle CAN networks [9] (Figure 1.1). Cloud platform can access DAP, do data analysis, and communicate with telematics server to share the information with other vehicles. The cloud platform provides great potential in vehicle driving data analysis and, therefore, safety driving assistance.

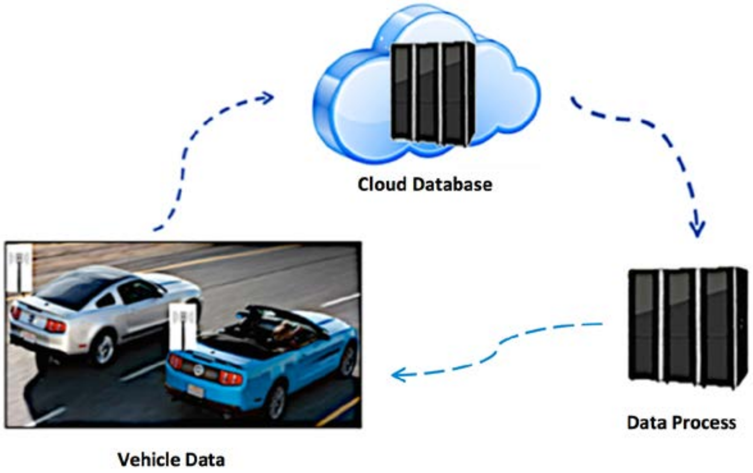


Figure 1.1: DAP data flow diagram.

1.2 Severe Driving Events and Safety Driving Assistance System

Among the vast data provided by DAP, two types of data highly relative to safety driving are severe braking data and handling limit minder data. The severe braking data and handling limit data are generated from the two vehicle driving events, severe driving event and handling limit event. A severe braking event refers to the situation in which a driver drastically reduces vehicle speed through sudden braking. A handling limit event is the event generated at the limit point of losing control of a vehicle due to hard manipulation.

Severe driving events pose potential dangers for drivers' safety and are known as a source of traffic flow instability. Analyzing the occurrence and distribution of events can help to avoid or prevent this kind of unsafe and unstable effects on traffic. Although severe driving events may be due to a large variety of random reasons, they statistically and associatively occur at a particular time and location. For example, during rush hour the possibilities of severe driving events are higher than at other times. Similarly, a high concentration of severe driving events occurrence exists at some locations with a special infrastructure, construction or other road situation.

Identification of the correlation of severe driving events with time and location is helpful for knowing the events distribution and forecasting the potential risks. However, there is a limited research effort taken to explore such driving events. In this work, we will employ clustering methods to explore severe driving event data to obtain the correlation between the occurrence of severe driving events associated with time and location [10], [11]. The correlation of severe driving events with time and location is presented as clusters of severe driving events in time and location domain. Combined with map database, real-time correlation can be notified to the drivers by screen interface in vehicles. The continuous variations of the cluster-based correlations with respect to time in areas are formulated as events count time series, which are modeled by different models accounting for event occurrence description and prediction.

The correlation identification procedure can be constructed as an intelligent analyzer installed in a vehicle computer-unit. With real-time communication with DAP and cloud platform, the intelligent analyzer identifies real-time correlation and shares the real-time correlation with DAP and other vehicles (Figure 1.1). Therefore, the system combining intelligent analyzer operation with DAP data storage and cloud platform data process is called as severe driving assistance system; the system assists drivers to recognize the risks caused by severe driving events and avoid the risks caused by severe driving events.

1.3 Contribution

To summarize the context of this work, a new form of advanced driver assistance system (ADAS), known as severe driving assistance system (SDAS), is designed in this work. The SDAS specifically assists drivers to recognize the potential risks caused by severe driving events and help drivers avoid the risks. The new ADAS uses intelligent analyzer employing clustering method to identify the events correlation with time and location. Based on the identified cluster-correlation, the statistical and probabilistically models are proposed in this dissertation for modeling the count of severe braking events occurrence at times and sites. The primary contribution is that we use clustering algorithms to recognize risk of the severe driving events at different times and locations, and construct different models for describing and predicting the risk.

The rest of the dissertation is organized as follows. Chapter 2 presents the background knowledge of the clustering algorithms used in the intelligent analyzer. Chapter 3 elaborates on the correlation identification procedure which uses batch clustering algorithm and real-time clustering algorithm for off-line and on-line implement, respectively. In Chapter 4, severe braking events are used as an example in the proposed correlation identification procedure to examine the correlations of the severe braking events with times and locations. In Chapter 5, continuing with the example of the severe braking events, we demonstrate the statistical time series modeling procedure for the variation of severe driving events occurrence with time at different locations to estimate and predict the severity of the potential risks in the future. In Chapter 6, we expand the modeling with probabilistic model and Hidden Markov model to describe the severe braking events stochastic process.

Chapter 2

Literature Review and Clustering Preliminary

Space-time scan statistics is a widely used method to detect and surveil event in time and space. In [12 - 15] researchers have used the method to identify incident disease clusters temporal trends and geographical patterns. It is possible to apply space-time scan statistics method for detecting and surveilling severe driving events with time and location. However, due to great variability of traffic flow and mutability of vehicle driving, a method with strong driving data dependency and driving data-learning ability is required to recognize the severe driving events occurrence in time and space. Clustering methods is such data-based pattern recognition method which intelligently provides instant clusters analysis.

Clustering is a method of partitioning and grouping objects into clusters where the objects in the same cluster share common characteristics [16]. The common characteristics can be interpreted as the correlations of the objects with the features upon which the clustering is applied [17], [18], [19]. Cluster algorithms have been widely used in various applications. In artificial vision or face recognition area, k-means clustering and k-nearest neighbor (K-NN) clustering are sought to compress visual data to generate clusters for finding facial objects and body objects [20], [21]. In speech recognition or acoustic inspection, hierarchical clustering and leader-follower clustering are employed to find adaptive clusters and detect abnormal clusters to identify different data sources [22], [23]. In medical data process, such as, syntactic EEG analysis, evolving Gustafson-Kessel Like (eGKL) and fuzzy-c means meaning clustering are used to grouping the

EGG data to examine EEG correlation to genetic predisposition to alcoholism [24]. Among different clustering algorithms, fuzzy c-means (FCM) is widely used for the process of partitioning. Although FCM has some drawbacks, e.g., noise sensitivity [25], it is useful in the beginning process of clustering while it can be improved in the sequel by using other clustering algorithms [24]. When using FCM, the number of clusters needs to be known a priori. Previous work has proposed criteria and methods on the number of clusters, such as rate distortion theory [26], gap statistic [27], [28], and weighted gap statistic [29], [30]. Subtractive clustering is another method capable of generating the number of clusters based on the criterion of potential reduction of the cluster centers [31]. While subtractive clustering is on the one hand fast to implement, on the other hand the number of clusters is dependent on the density and concentration of the data points. Thus, when examining a batch of data, the combination of subtractive clustering and FCM can efficiently produce appropriate clusters. However, it fails to implement on real time due to their batch clustering property. To circumvent this shortcoming, in this work we will employ evolving clusters matching with the dynamic performance of data stream to construct a real-time clustering algorithm [32], [33]. The evolving Gustafson-Kessel like (eGKL) algorithm is specifically used for online learning and evolving the pattern of clusters [24]**,** [32]. In this paper, FCM will be initially used to recognize the correlation patterns of severe driving events with their features, time and location. Subsequently, eGKL clustering is implemented to update and improve the correlation patterns in a real-time framework. In [20] and [34], the k-means clustering has shown to provide higher reliability and better run-time than FCM clustering and subtractive clustering. However, we use FCM clustering, as it is the only method to match the eGKL algorithm and calculate the memberships of data points to clusters. This cannot be accomplished by k-means clustering since it is a crisp clustering algorithm. Moreover, we use FCM in the initial stage of the correlation identification procedure and real-time clustering will be used then to update the correlation thereafter. Hence, the run-time difference between FCM and k-means clustering will be negligible subtractive clustering to suggest an appropriate number of clusters to initial the clustering.

In the following of this chapter, the background knowledge of the clustering algorithms used in the intelligent analyzer for the correlation of the severe driving events with time and location is presented. The algorithms include subtractive clustering, FCM and eGKL algorithm.

2.1 Subtractive Clustering

Subtractive clustering is a fast, one-pass algorithm to estimate the number of clusters. The number of clusters, c, is estimated based on the calculated potentials of data points. Considering a data set $X = X_1, \dots, X_N$ in an n dimension space, the potential of the data point X_k is calculated as

$$P_k = \sum_{j=1}^{N} e^{-\frac{4}{r_a^2}\left\|X_k - X_j\right\|^2}, \tag{2.1}$$

where r_a is the radius defining a neighborhood. A data point with many neighboring data points will therefore have high potential of forming a cluster. After the potential calculation for every data point, the data point with the highest potential is selected as the first cluster center and the potential of all the other data points are updated as

$$P_k \Leftarrow P_k - P_1^* e^{-\frac{4}{r_b^2}\|X_k - X_1^*\|^2}, \tag{2.2}$$

where X_1^* denotes the first cluster center and P_1^* is its potential value. A new neighborhood r_b is defined to update the clustering potential of the other data points. During the updating process, the potentials of the data points far away from the first cluster center are minimally affected. To avoid obtaining closely spaced cluster centers, r_b is set to be $r_a < r_b < 2r_a$. After updating the clustering potential, the data point with the highest remaining potential is selected as the second cluster center. Further update of the potentials of the remaining data points is carried out according to their distance to the second cluster center. Following this rule, after obtaining the i-th cluster center, the potential of a remaining data point X_k is obtained as

$$P_k \Leftarrow P_k - P_i^* e^{-\frac{4}{r_b^2}\left\|X_k - X_i^*\right\|^2}, \tag{2. 3}$$

where X_i^* is denoted the i-th cluster center, P_i^* is i-th cluster center potential value and the superscript i is also equal to the number of updates.

To obtain the appropriate number of clusters, an acceptance ratio ($\overline{\varepsilon}$) and a rejection ratio ($\underline{\varepsilon}$) are introduced to determine the acquirement or abandonment of a new cluster center. If $P_i^* > \overline{\varepsilon}\, P_1^*$, then X_k^* is accepted as the i-th cluster center; otherwise if, $P_k^* < \underline{\varepsilon}\, P_1^*$, then X_i^* is rejected and the subtractive clustering is over. However, if $\underline{\varepsilon}\, P_1^* < P_k^* < \overline{\varepsilon}\, P_1^*$, the potential of X_i^* drops into a grey area. Chiu [31] has used an assistance variable d_{min}, which is the shortest distance between X_k^* and existing cluster centers. The variable helps to check the inequality $\frac{d_{min}}{r_a} + \frac{P_i^*}{P_1^*} \geq 1$. If the inequality is valid, the point is accepted as a new center, if not, it is rejected.

2.2 Fuzzy c-means Clustering

Fuzzy c-means clustering (FCM) [10], [11] is derived from minimizing an objective function J that represents the fitting error of the clusters regarding the data,

$$J(V,U) = \sum_{i=1}^{c} \sum_{k=1}^{N} (u_{ik})^m \, {d_{ik}}^2, \tag{2.4}$$

where N is the number of data points, c is the number of clusters given in advance, u_{ik} is denoted as the membership degree of data point k to cluster i, and the distance d_{ik} is measured between the $k-th$ data point $X_k = [x_{k1}, \ldots, x_{kn}]$, and the $i-th$ cluster centers $V_i = [v_{i1}, \ldots, v_{in}]$, $i = 1, \ldots c$; n is the number of selected features describing each data point $X_k \in \mathbb{R}^n$. The parameter $q \in [1, \infty)$ controls the degree of fuzziness. If q is unity, the membership is regarded as crisp, and while q increases, the membership becomes fuzzier. Usually, the Euclidean norm is selected for the calculation the distance between centers and the points that measure their similarities,

$${d_{ik}}^2 = (X_k - V_i)(X - V_i)^T = {\|X_k - V_i\|_2}^2. \tag{2.5}$$

In addition, the following constrains are introduced to avoid trivial solution,

$$u_{ik} \in [0,1], 1 \le i \le c, and\ 1 \le k \le N, \tag{2.6a}$$

$$\sum_{i=1}^{c} u_{ik} = 1, 1 \le k \le N, \text{ and} \tag{2.6b}$$

$$0 \le \sum_{i=1}^{N} u_{ik} \le N, 1 \le i \le c. \tag{2.6c}$$

Lagrange multiplier method is used to solve the minimization problem with constrains. Initialize the membership matrix $U_{c \times N} = U^0$, and iteratively solve the membership u_{ik} and cluster centers V_i. The membership u_{ik} and cluster centers V_i are updated by Eq. (2.7) and Eq. (2.8),

$$u_{ik} = \frac{1}{\sum_{j=1}^{c}(\frac{d_{ik}}{d_{jk}})^{1/(q-1)}} \text{ and} \tag{2.7}$$

$$V_i = \frac{\sum_{k=1}^{N}(u_{ik})^q X_k}{\sum_{k=1}^{N}(u_{ik})^q}. \tag{2.8}$$

This iteration will stop until membership matrix convergent, which means $\|\Delta U\| < \epsilon$, and ϵ is a small termination value.

2.3 Evolving Gustafson-Kessel Like Algorithm

Fuzzy c-means (FCM) clustering uses Euclidean norm to calculate the distance of data points to cluster centers, and the formulated clusters are circle shape. The Gustafson-Kessel algorithm (GK) [35] is developed to identify ellipsoidal clusters by employing an adaptive distance norm (Mahalanobis norm), where the distance are calculated as

$$d_{ik}{}^2 = (X_k - V_i)S_i(X_k - V_i)^T = \|X_k - V_i\|_{S_i}{}^2, \tag{2.9}$$

in Eq. (2.9) V_i is the i-th cluster center and S_i is a positive definite symmetric matrix dependent on a covariance matrix F_i as

$$S_i = \rho_i |F_i|^{1/n} F_i^{-1}, \tag{2.10}$$

where ρ_i is the cluster volume and F_i is the fuzzy covariance matrix of i-th cluster and is calculated as

$$F_i = \frac{\sum_{k=1}^{N}(u_{ik})^q (X_k - V_i)^T (X_k - V_i)}{\sum_{k=1}^{N}(u_{ik})^q}. \tag{2.11}$$

In Eq. (2.11), u_{ik} is calculated as Eq. (2.7). The parameter q does not only control the degree of fuzziness, but also is a weighting exponent that determines how much the clusters may overlap.

The evolving Gustafson-Kessel Like (eGKL) algorithm uses the similarity between the stream of data through monitoring the process in statistical process control (SPC) and the stream of data in evolving clustering to determine the radius of the ellipsoidal clusters. The radius of the cluster is determined by the Chi-square distribution constant $\chi^2_{n,\beta}$ and the determinant of the covariance matrix as

$$r_i^2 = \chi^2_{n,\beta}|F_i|^{\frac{1}{n}}, \tag{2.12}$$

where n is the dimension of data space and β is the possibility of false alarm. The similarity between X_k and each of the existing clusters is evaluated by checking the similarity relation,

$${D_{ik}}^2 < \chi^2_{n,\beta}, i = [1, c], \tag{2.13}$$

where the unit-less distance parameter D_{ik} is determined as

$${D_{ik}}^2 = (X_k - V_i){F_i}^{-1}(X_k - V_i)^T. \tag{2.14}$$

The following steps present the eGKL algorithm implementation in real time.

Step 1. Choose the probability of a false alarm β and define the $\chi^2_{n,\beta}$. Usually, the probability of false alarm β is chosen as 0.0455 in accordance with the 2σ (σ is the *variance* of the distribution) of the process control band in the single variable SPC.

Step 2. Based on the results of fuzzy c-means, calculate the inverse covariance matrices F_i^{-1} and the cluster centers V_i.

Step 3. Choose the minimum number of points in the cluster as a function of the data point dimension n as

$$M_{min} = \frac{n(n+1)}{2}. \tag{2.15}$$

Step 4. Read the new data point X_k, calculate the membership $u_{ik}$of X_k to all credible clusters and the distance to all cluster centers by Eqs. (2.9) and (2.11).

Step 5. Check the similarity of X_k to the existing clusters, i.e. Eq. (2.14), and identify the closest p-th cluster,

$$p = \arg min_i(D_{ik}),\, i = 1, \ldots, c. \tag{2.16}$$

Step 6(a). If ${D_{pk}}^2 < {\chi_{n,\beta}}^2$ or the number of points in the p-th cluster is $M_p < M_{min}$ then update p-th cluster as follows.

- Update the parameters associated with the $p - th$ cluster,

$$V_{p,new} = V_{p,old} + \alpha\left(X_k - V_{p,old}\right), \tag{2.17}$$

$$F_{p,new}^{-1} = \frac{1}{1-\alpha}(1 - G_p)F_{p,old}^{-1}, \tag{2.18}$$

$$\left|F_{p,new}\right| = \Gamma(1-\alpha)^{n-1}\alpha\left|F_{p,old}\right|, \text{ and} \tag{2.19}$$

$$M_{p,new} = M_{p,old} + 1, \tag{2.20}$$

where α is the learning rate, $G_p = \alpha\Gamma_p^{-1}\left(X_k - V_{p,old}\right)^T\left(X_k - V_{p,old}\right)F_{p,old}^{-1}$, and $\Gamma_p = 1 - \alpha + \alpha\left(X_k - V_{p,old}\right)F_{p,old}^{-1}\left(X_k - V_{p,old}\right)^T$.

- Update the centers of the remaining clusters accordingly as

$$V_{i,new} = V_{i,old} - \alpha\left(X_k - V_{i,old}\right), i \in \{1, \ldots, c\}, i \neq p. \tag{2.21}$$

Step 6(b). If none of the conditions ${D_{pk}}^2 < {\chi_{n,\beta}}^2$ or $M_p < M_{min}$ satisfied then apply the following.

- Increase the number of clusters by one, i.e., $c = c + 1$.
- Initialize the new cluster with the associated parameters as

$$V_c = X_k;\ F_{c,new}^{-1} = F_0^{-1}; \left|F_{c,new}\right| = |F_0|; M_c = 1, \tag{2.22}$$

where F_0 is an initial estimate of the fuzzy covariance matrix. Usually, the matrix is initialized as a diagonal matrix $F_0^{-1} = \gamma I$, where γ is sufficiently large positive number.

Chapter 3

Correlation Identification Procedure

In last chapter, we reviewed the clustering algorithms which would be employed in the following proposed correlation identification procedure. In this chapter, the correlation identification procedure will be elaborated to construct as an intelligent analyzer using clustering algorithms. The clusters recognized from the procedure represent the correlations of events data points with the features time and location. At the beginning of the procedure, initial cluster-based correlations identified from clustering for a batch of historical severe driving events data with the features time and location reveal the historical correlation of the driving events with time and location. The, the correlations are continuously developed with time reflected by the variation of the modes of the recognized clusters with clustering process on the real-time data stream. Thus, the proposed correlation identification procedure is presented as two parts: namely, batch clustering and real-time clustering.

3.1 Batch clustering

The batch clustering structure is presented in the left side of the flowchart in Figure 3.1. It includes data acquisition, data transformation, time domain clustering and two-stage geographical domain clustering. The right hand side of the flowchart represents the batch clustering procedure implementation exemplified by severe braking event.

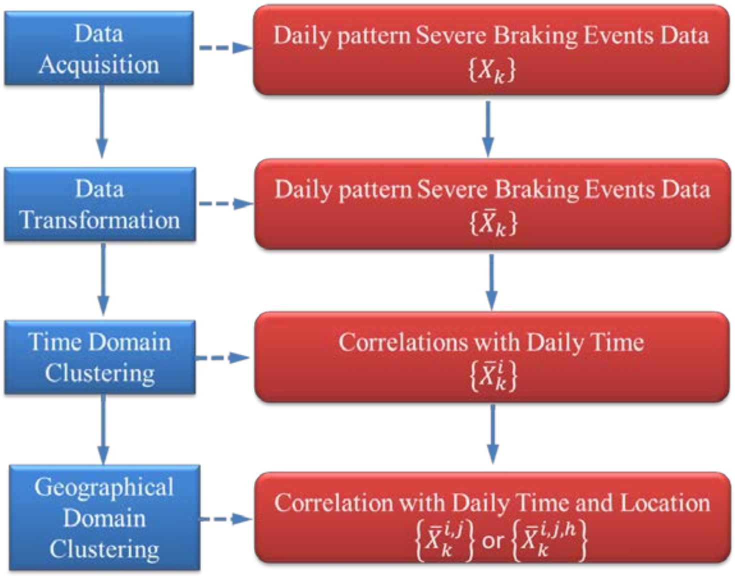

Data Acquisition
Daily pattern Severe Braking Events Data $\{X_k\}$
Data Transformation
Daily pattern Severe Braking Events Data $\{\bar{X}_k\}$
Time Domain Clustering
Correlations with Daily Time $\{\bar{X}_k^i\}$
Geographical Domain Clustering
Correlation with Daily Time and Location $\{\bar{X}_k^{i,j}\}$ or $\{\bar{X}_k^{i,j,h}\}$

Figure 3.1: Batch clustering procedure.

3.1.1 Data Acquisition

Data acquisition involves the process of acquiring historical data from Data Acquisition and communications Platform (DAP). The severe driving events (severe braking events or handling limit minder (HLM) events) contain location data, event time data and event indicator data. Location data includes the location coordinates at the respective time data, which is the time when the events are invoked. Events indicator data indicate the characteristics of the events. For example, the location data of severe braking events data are denoted as $[X_{k1}, X_{k2}, T_k]$, where X_{k1} and X_{k2} are the sampled latitude and longitude coordinates of the driving vehicle at the corresponding time data T_k. The severe braking event indicator data are denoted as $[A_k]$, where A_k is the vehicle deceleration at the corresponding time T_k. Therefore, a severe braking event data point is denoted as $X_k = [X_{k1}, X_{k2}, T_k, A_k]$. Analogously, a HLM event data contains the same feature data, location data, time data, and event indicator data. The event indicator data of HLM of severe curving and severe bumping are chosen as the vehicle centrifugal acceleration and the ratio of vehicle jolting magnitude versus jolt frequency, respectively. Other event indicator data of HLM resulting from other reasons can be presented by other certain measures.

3.1.2 Data Transformation

Data transformation is the preparation step prior to clustering process. In this step, a pre-process called "contraction" is implemented on the severe driving events data. The severe driving events occur at different times and different dates. To identify the correlation of severe driving events with time, severe driving events data are compacted into one day. After the contraction, the severe driving events are transformed into daily-pattern events data. For example, severe driving events are transformed into $\bar{X}_k = [X_{k1}, X_{k2}, \bar{T}_k, A_k]$.

3.1.3 Time Domain Clustering

Time domain clustering is the clustering process applied on the time of the events to partition the events data points into different clusters where each of them shares the common time characteristics [28]. The clustering process is composed of the subtractive clustering and the fuzzy c-means clustering (FCM). The subtractive clustering suggests an appropriate number of clusters for the FCM clustering, which produces the common time characteristics clusters. When implementing subtractive clustering, parameters, such as, the neighborhood radius r_a, update radius r_b, and acceptance ratio $\overline{\varepsilon}$ are required to supplied firstly (since we expect as many clusters as possible, we choose rejection ratio $\underline{\varepsilon}$ to be zero). We propose a potential-based objective function for the selection of the three parameters. The objective function measures the goodness of "potential" of the generated cluster centers. Good potential is defined by the following criteria: i) the mean value of the cluster centers potentials (P_i^*) is large; ii) few of the potentials of cluster centers fall into the gray areas ($\underline{\varepsilon}\, P_1^* < P_i^* < \overline{\varepsilon}\, P_1^*$); iii) if some potential inevitably fall into the gray area, they should be as close as possible to the acceptance ratio $\overline{\varepsilon}$. We expect that the potential of cluster centers are as great and far away from the gray area as possible and, at the same time, as many clusters as possible. This leads to the maximization of the objective function with a subject constrain described by Eq. (3.1) and Eq. (3.2) as

$$\text{Min} \quad J = a \cdot e^{(-r_a * c)} + b \cdot \frac{c_{\overline{\varepsilon}}}{c} \cdot \left(\frac{1}{c_{\overline{\varepsilon}}} \sum_{i=1}^{c_{\overline{\varepsilon}}} P_i - \left(\overline{\varepsilon} - \frac{1}{c_{\underline{\varepsilon}}} \sum_{j=1}^{c_{\underline{\varepsilon}}} P_j \right) \right) \text{ and} \tag{3.1}$$

$$s.t.\ a + b = 1, \tag{3.2}$$

where a and b are the balance ratios, c is the number of clusters, $c_{\overline{\varepsilon}}$ is the number of such clusters whose center potentials are greater than $\overline{\varepsilon}$, and $c_{\underline{\varepsilon}}$ is the number of the rest clusters whose center potential are less than $\overline{\varepsilon}$, but greater than $\underline{\varepsilon}$. Thus, $c = c_{\overline{\varepsilon}} + c_{\underline{\varepsilon}}$.

Generally, chosen the clustering parameters in the ranges $0 < r_a < 0.5$, $0.5 \le \overline{\varepsilon} \le 0.85$, and $r_a \le r_b \le 2r_a$, one can solve this maximization by a searching algorithm as shown in the following pseudo code.

```
For $r_b = \{1.25, 1.5, 2\} r_a$
    For $r_a = 0.05{:}\,0.5$
        For $\overline{\varepsilon} = 0.5{:}\,0.85$
            Maximize $J$ with $a = 0.4$ and $b = 0.6$
        End
    End
End
```

Utilizing this searching algorithm, the optimal combination of the three parameters is obtained. Then, the subtractive clustering algorithm is implemented to suggest the number of clusters for FCM. For severe driving events, the events data points $\{\bar{X}_k\}$ are partitioned into c time-based clusters $v^i = \{\bar{X}_k^i\}$, where $i = 1, \ldots, c$.

3.1.4 Geographical Domain Clustering

Geographical domain clustering is the process of distance-based re-clustering on the feature location to obtain further correlations of the severe driving events with location. The clustering process is implemented on latitude and longitude of the i-th time-based cluster v^i. The clustering process, similar to that of time domain clustering, consists of subtractive clustering and FCM clustering. However, the parameters used in this subtractive clustering are not solved from the optimization problem (Eqs. (3.1) and (3.2)), but they are inferred from that how large area is expected to be covered by a resulting cluster. For example, at Detroit ($42.3^\circ N, 83^\circ W$), if we require the size of the resulting cluster around $11km$ by $8km$ (latitudinal by longitudinal), the parameter r_a is chosen to be 0.1 on the latitudinal and longitudinal coordinates. If we require the size of the resulting cluster around $1km$ by $1km$ (latitudinal by longitudinal), the parameter r_a is chosen to be 0.01 on latitudinal and longitudinal coordinates (at $42.3^\circ N$ and $83^\circ W$, 0.1 $degree$ in the latitudinal and longitudinal directions is $11km$ and $8km$, respectively, and $0.01\,degree$ are around $1km$ and $1km$, respectively). Apparently, r_a plays a significant role in determining the number and size of the clusters.

In this dissertation, we construct a two-stage geographical domain clustering using two different value for r_a. In the first stage, by using $r_a = 0.1$, FCM generates clusters, which represent general correlation of events with location in a greater area, $11km \times 8km$. We will call these cluster “general clusters”. In the second stage, by using $r_a = 0.01$, FCM generates clusters, called “accurate clusters”, which represent accurate correlation of events with specific locations in a smaller area, $1km \times 1km$. In the safety driving assistance system, more than two stages of geographical domain can be implemented depending on the coverage area that drivers interest. When more accurate correlations of severe driving events with location are needed, a smaller value of r_a is provided with further stage of geographical domain clustering implementation.

Performing the first-stage geographical clustering results in clusters $v^{i,j} = \{\bar{X}_k^{i,j}\}$, which are the j-th subbranch cluster from the cluster v^i with $j = 1, \dots, m_i$, and m_i is the number of clusters resulting from the cluster v^i. The clusters $v^{i,j}$ refer to the general correlations of the severe driving events with time and area. By running the second-stage geographical clustering, clusters $v^{i,j,h} = \{\bar{X}_k^{i,j,h}\}$ are obtained. Following the same rule, $v^{i,j,h}$ is the h-th subbranch cluster from the subbranch cluster $v^{i,j}$, where $h = 1, \dots, w_{i,j}$, and $w_{i,j}$ is the number of clusters from the subbranch cluster $v^{i,j}$. The accurate clusters $v^{i,j,h}$ refer to accurate correlations of severe driving events with time and location.

3.2 Real-time Clustering

After the implementation of the batch clustering, numbers of cluster are generated representing the correlations between the severe driving events and time and location. However, the batch clustering fails to supply two major aspects of useful information: firstly, we cannot get knowledge of the very recent correlations of the severe driving events with time and location, and secondly, we cannot get knowledge of how the correlations are developed or transformed in a certain time span. In other words, batch clustering fails to provide dynamic information of the correlation between the events and time and location. To circumvent this problem, we will consider the real-time clustering.

3.2.1 Real-time Data Acquisition

In this step, the severe driving data are queried in real time. When a severe driving event occurs, the corresponding data will be acquired at the same time, and the new data point X_k is sent into the following separation step.

3.2.2 Time Domain Separation

Time-based clusters have been generated from the time domain clustering step in the batch clustering. In this step, the time of the new data point is recognized and used to compare with the boundary time of the clusters, v^i, $i = 1, \dots, c$, to determine which cluster the new data point belongs to.

3.2.3 Geographical Domain Separation

In the geographical domain separation, the new data point X_k either belongs to an existing cluster, $v^{i,j}$ or $v^{i,j,h}$, when the distance between the point and a cluster center is less than an admittance distance, or it defines a new cluster otherwise. According to evolving Gustfson-Kessel Like (eGKL) algorithm, the boundary of a cluster is determinant by the cluster covariance matrix and the probability of false alarm β parameter according to Eq. (2.12). Accordingly, the distance between the new data point and each of the existing clusters is evaluated by Eq. (2.5). The eGKL algorithm proposed in Chapter 2.3 is implemented in real time to update existing clusters $v^{i,j}$ or $v^{i,j,h}$ or define new clusters.

Chapter 4

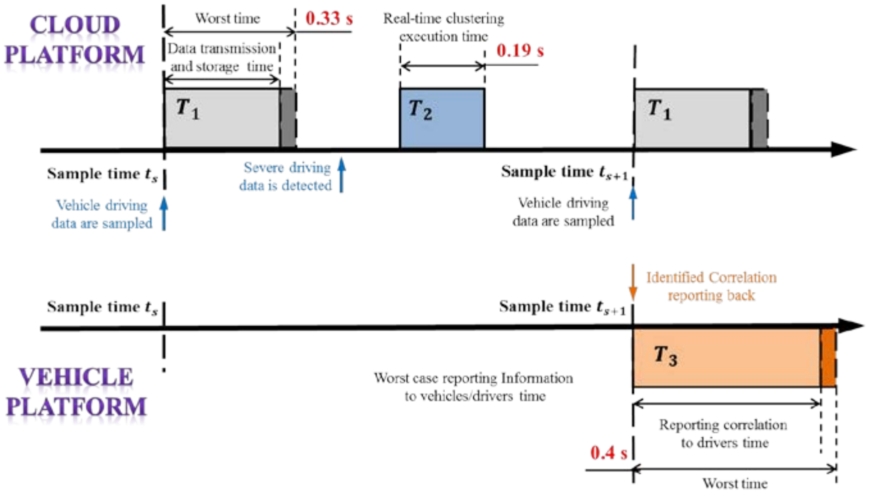
Example of the Implementation of Correlation Identification Procedure and Results

In this section, we take the severe braking event as example of the general severe driving events to describe the implement the correlation identification procedure. After the procedure, the corresponding correlations of severe braking events with time and location are discussed to exemplify the general correlation of severe driving events with time and location from safety driving assistance system.

The historical driving data of seven vehicles from 04/01/2013 to 07/09/2013 were acquired from Data Acquisition and Communication Platform (DAP) for batch clustering. Based on the batch clustering result, the real-time clustering was then implemented on the driving data of the vehicles from 10/01/2013 to 02/25/2014 in real time framework.

The real-time correlation identification procedure is implemented as presented in Figure 4.1. The vehicle driving data have been transmitted from vehicle to DAP with 1 Hz frequency. The data transmission and storage time is taken less than 0.33 second by the database management system. Once the severe driving event data have been detected, the real-time clustering computation (Eqs. (2.9) – (2.22)) is implemented in cloud platform. The real-time clustering processing time with 2.8 GHz commercial processor is obtained as 0.19 second. Then at the next sampling time, the updated real-time correlations are queried and reported to vehicles. The reporting time on vehicles to drivers takes less than 0.4 second. Generally, the process from event data transmission and storage, to real-time clustering computation, to correlation notification to drivers takes less than 1 second based on experiments on the testing vehicles."

The *August* data and *September* data are not available due to DAP system upgrade. In the following, the details of the correlation identification procedure are presented.



CLOUD PLATFORM
Worst time
0.33 s
Data transmission and storage time
Real-time clustering execution time
0.19 s
T_1
T_2
T_1
Sample time t_s
Severe driving data is detected
Sample time t_{s+1}
Vehicle driving data are sampled
Vehicle driving data are sampled
Identified Correlation reporting back
Sample time t_s
Sample time t_{s+1}
T_3
VEHICLE PLATFORM
Worst case reporting Information to vehicles/drivers time
Reporting correlation to drivers time
0.4 s
Worst time

Figure 4.1: Correlation identification procedure real-time implementation.

4.1 Batch Clustering

The batch clustering process is implemented on the severe braking events. It includes data acquisition, data transformation, time domain clustering and geographical domain clustering (see the right-hand side of the flowchart in Figure 3.1).

4.1.1 Data Acquisition

The historical driving data available in DAP were from seven testing vehicles of Ford Motor Company. The relevant data of severe braking data are GPS data and braking data. GPS data are gathered from GPS tracking devices placed in vehicles. Braking data are gathered from accelerometer sensor recording the speed change per second at the moment of brake start. Severe braking is defined as the braking with large deacceleration value and greater than a defined threshold value. The threshold value is calculated from the Eq. (4.1),

$$A_{threshold} = |\bar{A} + 1.5 \cdot std(A)|, \tag{4.1}$$

where A is the historical braking deacceleration data set and $std(A)$ is the standard deviation of the data set. $A_{threshold}$ in this testing case is $3.7\ mph/s$. After the relevant data acquisition, the severe braking event data points are formed by the combination of the GPS data and severe braking indicator data according to the severe braking events time, $X_k = [X_{k1}, X_{k2}, T_k, A_k]$. Dotted points in Figure 4.2 represent the location data of the seven testing vehicles in the longitude-latitude graph. Figure 4.3 shows the severe braking indicator data of the seven testing vehicles with respect to the occurrence time.

4.1.2 Data Transformation

The contraction procedure compacts the severe braking events data, $\{X_k\}$, based on their daily times to form daily-pattern severe braking events data, $\{\bar{X}_k\}$. In Figure 4.4, the left-hand side shows the distribution of the events of seven testing vehicles with respective to time and date, and the right side shows the daily-pattern distribution of compacted severe braking events. In next step, time domain clustering is implemented on the compacted daily-pattern severe braking events data, $\{\bar{X}_k\}$.

4.1.3 Time Domain Clustering

The optimal parameters for subtractive clustering solved from the maximization problem proposed in Chapter 2.1.3 are $r_a = 0.3$, $r_b = 1.5r_a$, and $\bar{\varepsilon} = 0.50$. With these parameters, subtractive clustering suggests two clusters for fuzzy c-means (FCM) clustering. Solving the FCM clustering, 2 clusters, i.e. v^i, $i = 1, 2$ with the clusters centers $V_1 = 0.5196$ (around time 12:28) and $V_2 = 0.8529$ (around time 20:28), we name the first cluster as “daytime” cluster and second cluster as “nighttime” cluster. The boundary between the two clusters is $\bar{T} = 0.7362$ (around 17:40).

4.1.4 Geographical Domain Clustering

The first-stage geographical domain clustering generated numbers of $m_1 = 14$ and $m_2 = 20$ from FCM clustering on the two time-based clusters, v^1 and v^2, respectively. The resulting clusters were denoted as $v^{1,j}$, where $j = 1, \dots, 14$ and $v^{2,j}$ with $j = 1, \dots, 20$, which are general clusters whose centers are denoted as $V_{i,j}$ representing the general correlation of the severe braking events with time and location. Since the general clusters fail to provide accurate correlation of the events with time and location, a second-stage geographical domain clustering is implemented on each of the general clusters $v^{i,j}$ to generate $w_{i,j}$ subbranch clusters. The number of subbranch clusters generated from general clusters $v^{i,j}$, when $i = 1$ and $j = 1, \dots, 14$ are $w_{1,j} = 24, 17, 16,19, 32, 11, 19, 3, 9, 25, 4, 21, 21,$ and 1; when $i = 2$ and $j = 1, \dots, 20$, $w_{2,j} = 2,13,32,1,3,13,20,14,13,10,17,3,10,14,16,17,9,13,15,$ and 4. This branch-branching clustering structure is illustrated in Figure 4.5. The subbranch clusters are called accurate clusters denoted as $v^{i,j,h}$, where $h = 1, \dots, w_{i,j}$. The respective centers of the accurate clusters are denoted as $V_{i,j,h}$, presenting accurate correlation of the severe braking events with time and location.

4.1.5 Batch Clustering Results

From the batch clustering, 165 qualified clusters were identified if the minimum number of data points in a cluster is 3 from Eq. (2.15). The clusters statistics is summarized in Table 4.1. $\mathcal{C}^{\mathcal{N}}$ denotes the clusters containing $\mathcal{N}$ data points ($\mathcal{C}^{\mathcal{N}}$,

$\mathcal{N} = 13$, denotes the clusters contain more than 12 data points). “$\mathfrak{D}$” and “$\mathfrak{N}$” denotes the clusters in daytime and in night-time, respectively.

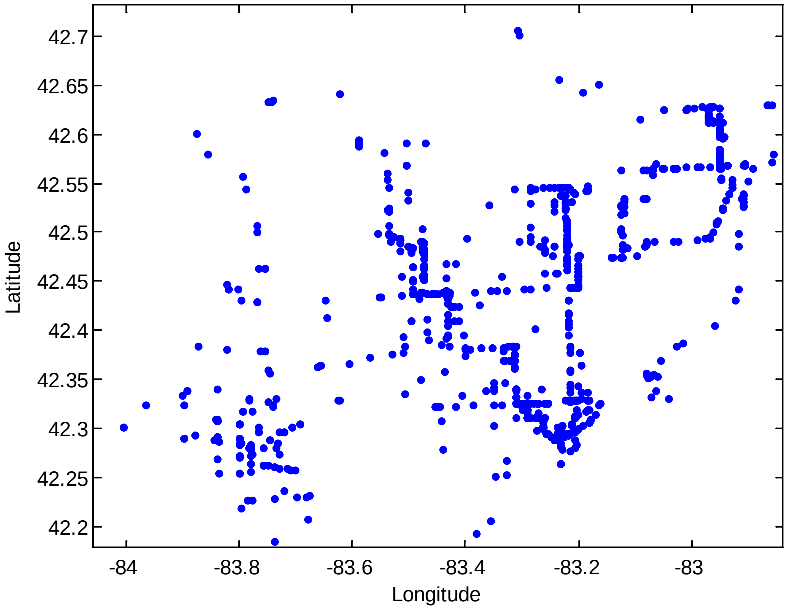


Figure 4.2: Location of the severe braking event from 04/01/2013 to 07/09/2013.

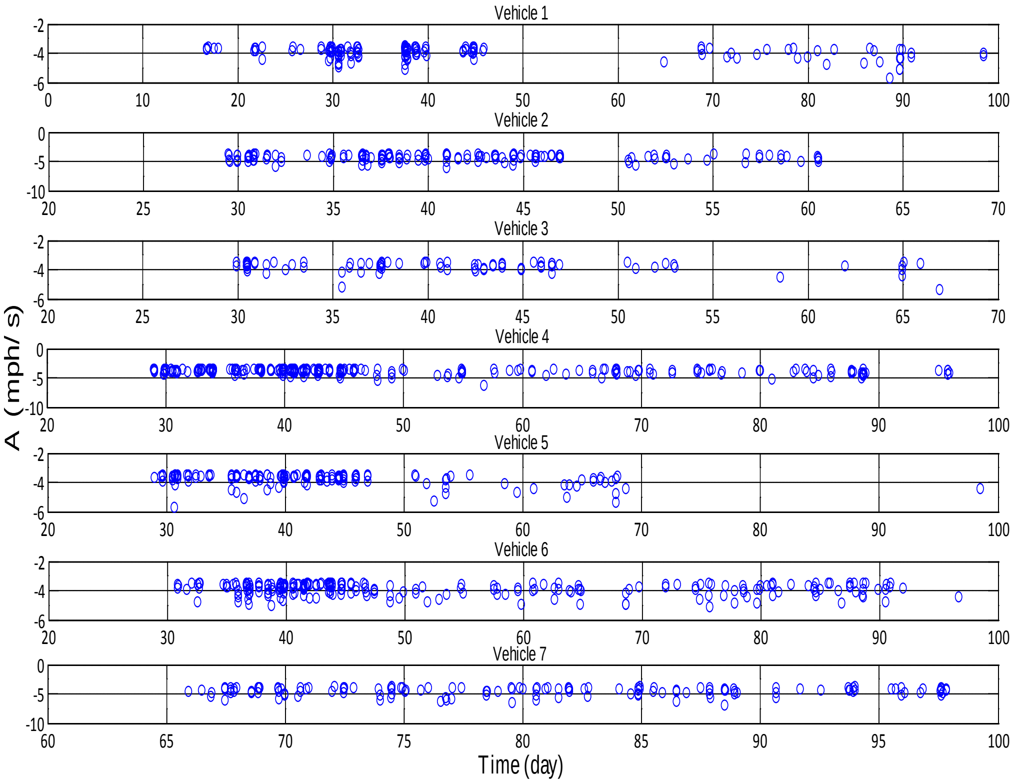


Figure 4.3: Severe braking events indicator data from 04/01/2013 to 07/04/2013.

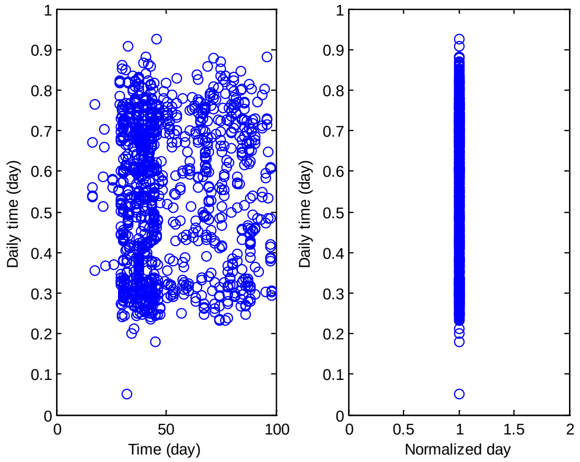


Figure 4.4: Sever braking events time distribution and daily-pattern distribtuion.

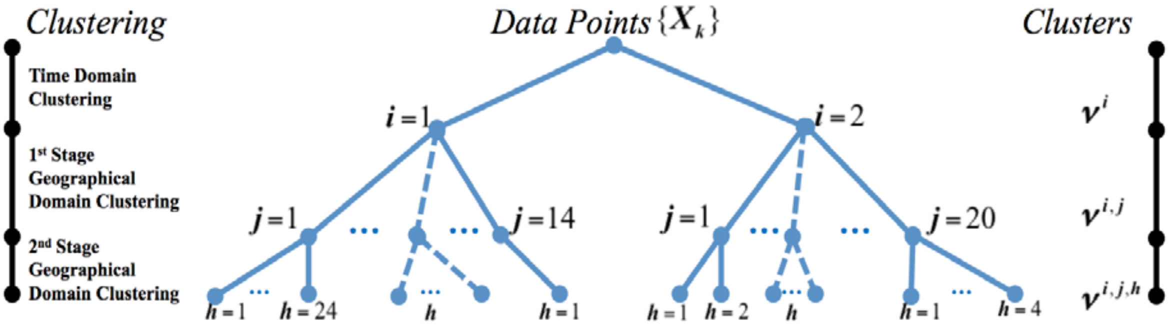


Figure 4.5: Structure of batching clustering process.

Table 4.1: Effective clusters statistics

	C^3	$\mathcal{C}^4$	$\mathcal{C}^5$	$\mathcal{C}^6$	$\mathcal{C}^7$	$\mathcal{C}^8$	$\mathcal{C}^9$	$\mathcal{C}^{10}$	$\mathcal{C}^{11}$	$\mathcal{C}^{12}$	$\mathcal{C}^{\mathcal{N}>12}$
No.	56	36	20	14	16	4	5	6	3	1	4
$\mathfrak{D}$	33	11	7	8	6	0	3	2	0	0	3
$\mathfrak{N}$	23	25	13	6	10	4	2	4	3	1	1

Among the 165 clusters, nearly 70% clusters are $\mathcal{C}^{3-5}$, and 10% clusters are $\mathcal{C}^{10-13}$. We further make three groups of clusters according to the concentration of a cluster:

- Sparse concentration cluster: the number of data points in a cluster is smaller than 5.

- Dense concentration cluster: the number of points in a cluster is between 6 and 9.
- Super-dense concentration cluster: the number of data points in a cluster is equal or greater than 10.

Since each cluster center is in essence a prototypical data point that represents the characteristic behavior of the cluster, in the following we use cluster centers to analyze the distributions and concentrations of different cluster groups. Nearly 70% clusters belong to sparse concentration clusters. We plotted the centers of those clusters as “×” markers in Figure 4.6 to show the primary distribution of the severe braking events at daytime (blue marker) and nighttime (red marker). Density-based scan algorithm with noise (DBSCAN) clustering is applied on the cluster centers to find the coverage regions to illustrate the distribution of the coverage regions (regions formed by lines in Figure 4.6). It is noted that two of the regions at daytime were overlapped with two of the regions at nighttime. It means that the severe braking events constantly occur daytime and nighttime in these regions. For the other regions, severe braking events either occur at daytime or nighttime. Here, this distribution knowledge is useful for directing drivers to differentiate the regions that are persistently excited by severe braking events all day long.

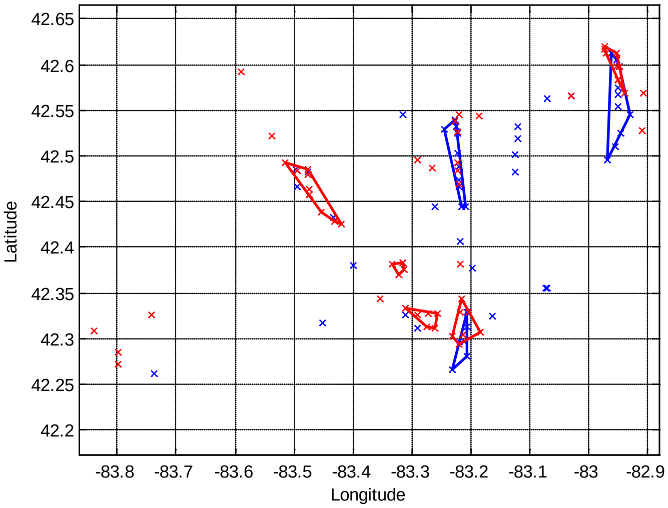


Figure 4.6: The distribution of sparse concentration clusters and the respective coverage regions. Blue markers represent daytime cluster centers, red markers represent nighttime cluster centers, blue lines form the coverage regions at daytime, and red lines form the coverage regions at nighttime.

A super-dense concentration cluster contains many more data points than sparse dense concentration clusters; however, the number of such clusters is very limited. The super-dense concentrated clusters indicate specific locations where the severe braking event occurs with a high possibility. Four examples of the locations are displayed by employing Google Maps API to display the events data points in Figure 4.7. Via the usage of Google Maps, road infrastructures with the events data points are provided for drivers. Drivers can avoid the potential dangers by choosing the alternative paths.

The thresholds differentiating the types of clusters (sparse concentration clusters, dense concentration clusters, and super-dense concentration clusters) are based on the percentage ratio of the number of each type of clusters to total number clusters, which is selected by designer. We use the percentage ratio as 70%:20%:10%. This percentage ratio is from the sense that most of clusters are not very concentrated. However, in the revised version of the paper, it is shown that the use of threshold 80% instead of 70% for the sparse concentration clusters is not. Super-dense concentration clusters denote specific locations which are notified on maps. Designers give the threshold for viewing the specific locations with high potential of occurrence of severe driving events. We choose the 10% clusters as the super-dense concentration clusters; other designers can choose other percentage of clusters as super-dense concentration clusters to view them on maps. There is no necessary to discuss the sensitivity for this percentage threshold.

So far, the analysis on the severe braking events distribution and concentration with respect to location and time is static. In the next section, we implement real-time clustering to update the clusters coordination to investigate the variation of the correlation of the severe braking events with the time and location.

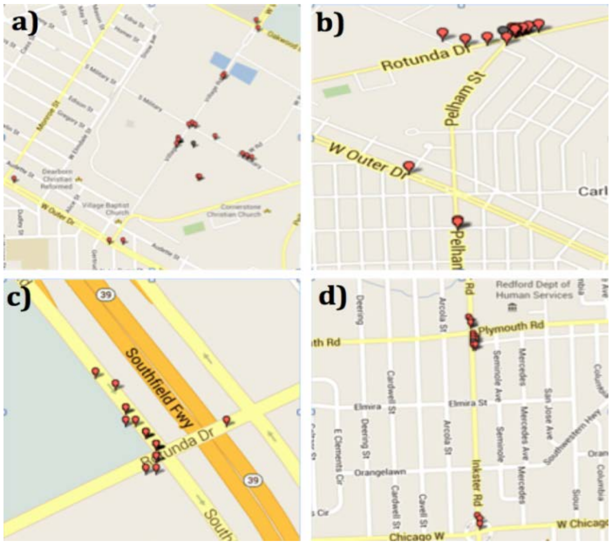


Figure 4.7: Concentrated clusters with more than 12 points: a) $v^{1,6,2}$; b) $v^{1,6,11}$; c) $v^{1,7,7}$; d) $v^{2,10,1}$. The red points represent the severe braking events locations, and the black point is the cluster center.

4.2 Real-time Clustering

Initial correlation of severe braking events with time and location were identified from batch clustering represented by recognized clusters. Real-time clustering, on the basis of the clusters, modifies and updates the clusters with real-time events data acquisition. The variation of the models of the clusters represents the development of the events correlation with time and location.

4.2.1 Real-time Data Acquisition

In this step, the severe braking data are queried in real time. Real-time clustering is implemented with the severe braking events data of the testing vehicles queried from 10/01/2013 to 02/25/2014 in real time. The clustering is implemented based on the results of batch clustering.

4.2.2 Time Domain Separation

When a new severe braking event data (X_k) is queried and recognized in real time, it goes to time domain separation. The data goes to one of the two time-based clusters, v^1 and v^2 by comparing the daily time of X_k to the boundary between the two clusters, which are resulted from time domain clustering in batch clustering.

4.2.3 Geographical Domain Separation

The eGKL algorithm is implemented in the geographical domain separation. The new severe braking event data point (X_k) goes to one of the existing clusters, if Eq. (2.13) is satisfied. Otherwise, it defines a new cluster. The radius of the existing clusters can be calculated from Eq. (2.12) with a predetermined probability of false alarm (β). Since the determination of β is similar to determining quality process control band in a single variable statistical process control, we initially choose β to be 0.68 to mimic 1σ process control band. Thus, at the beginning of the real-time clustering, the red ellipses in Figure 4.8 and Figure 4.9 define the boundaries of the general clusters $v^{i,j}$ and accurate clusters $v^{i,j,h}$, respectively. When implementing the real-time clustering, as the new data point goes into a cluster, the covariance matrix of the cluster is changed, and the boundary of the cluster is changed and updated (Eq. (2.12)). It is apparent that the determinant of the covariance matrix is sensitive to the number of data points in the cluster and dropping position of the new data point with respect to the boundary of the cluster. In one case, if the new data point drops close to the boundary of the cluster, it will affect greatly the value of the determinant of the covariance matrix of the cluster. In this case, the new data point will more significantly affect the pattern of the cluster when the number of data points in the cluster is small than when the number of data points in the cluster is large; and it will more increase the radius when the number of data points in the cluster is large than when the number of data points in the cluster is small. Based on this observation, in order to stabilize patterns of clusters and avoid continuously increasing or decreasing the sizes of clusters, we use the varying β to balance the effect of determinants of clusters on radius. The β is made experientially decrease from 0.68 to 0.3 with respect to the number points in the cluster. It means we gradually shrink force on the core data points close to the center of the cluster. In another case, if the new data point drops close to the center and far away from the boundary, it has little effect on the value of the determinant of the cluster, and the variation of β has little effect on the size of the cluster. In a third case, if the coming data point does not belong to any existing clusters, it will define a new cluster, and a circle pattern with a default radius is used to define its boundary. In Figure 4.8 and Figure 4.9, the blue lines define the boundaries of the general clusters and accurate clusters, respectively, at the end of the real-time clustering. One could easy identify the variation of the sizes of the general clusters in Figure 4.8, and observe that the number of accurate clusters is large and their sizes are small in Figure 4.9.

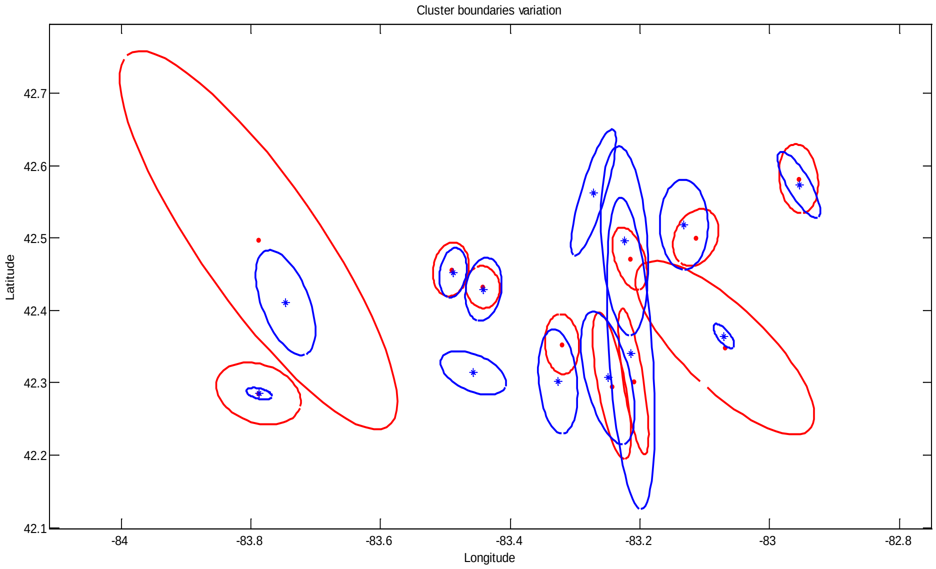


Figure 4.8: Comparison between the general clusters boundaries in the beginning of the real time clustering and at the end of the real time clustering.

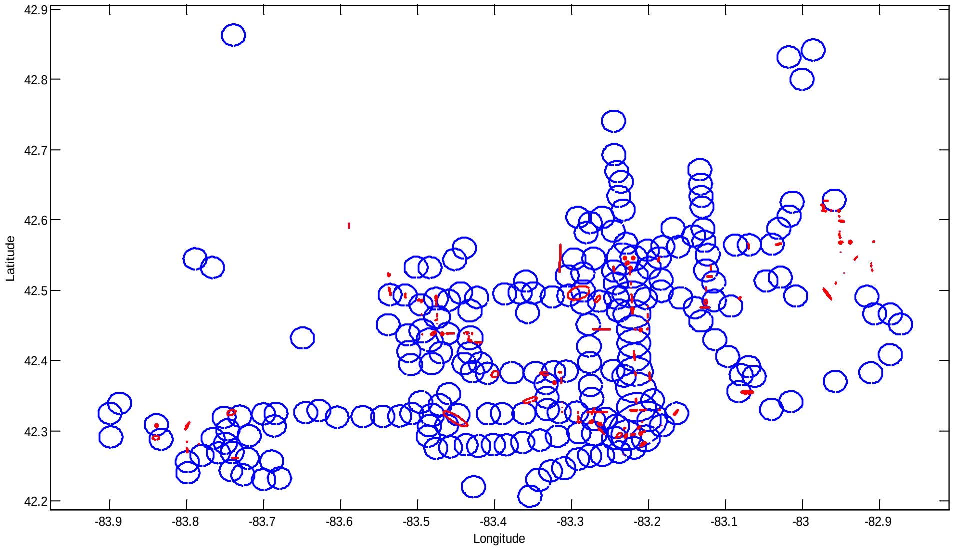


Figure 4.9: Comparison between the accurate clusters boundaries in the beginning of the real time clustering and at the end of the real time clustering.

4.2.4 Real-time Clustering Results

There are two issues need to be considered when generating accurate clusters during the real-time clustering. The first issue is the credibility of generated clusters. A credible cluster should have at least 3 data points as mentioned in Chapter 2.3. However, in Figure 4.9, any new event data point would define a new cluster once it does not belong to any of the existing clusters. It is not unusual that a new event point, once defining a new cluster, could be noisy data point. In other words, there will be no more or very low possibility of another occurrence of the event in the area for a long time. Thus, the clusters generated by the noisy data point should be ignored. The second issue is that some credible clusters could be inactive, i.e., no data point dropping in for a long time. From dynamic perspective, the noisy clusters and inactive clusters should be gradually vanished by using evolving-forgetting mechanism. Therefore, we define the following mechanism using the lifetime terminology and forgetting logic regarding the activity of clusters. We call a cluster is active means there are at least one event data point dropping into the cluster. Oppositely, a cluster is inactive means there is not event data point dropping into the cluster.

i) *Temporary cluster*: It is the cluster whose number of data points is less than 3 all its life time. We consider the lifetime of such clusters to be 10 days. If it is still a temporary cluster after 10 days, it will be neglected. Otherwise, it will be evolved to be a short-term cluster.

ii) *Short-term cluster*: It is the cluster growing from temporary cluster and its lifetime is 10 days too. If, within 10 days, it keeps inactive then it will be moved out. If it keeps active more than 20 days, it will be evolved to be a mid-term cluster. Otherwise, it is still a short-term cluster.

iii) *Mid-term cluster*: It is the cluster created from short-term clusters. Its lifetime is considered as 1 month. If, within 10 days, it is inactive, it will be discarded. If it keeps active more than 2 months, it will be evolved to be a long-term cluster. Otherwise, it is still a medium-term cluster.

iv) *Long-term cluster*: It is the cluster generated by mid-term cluster. If within 10 days it keeps inactive, it will canceled out. Otherwise, it is still a long term cluster.

We implemented the real-time clustering and applied the evolving-forgetting mechanism for the accurate clusters [3]. Figure 4.10 demonstrates the results of 09/15/2013, 11/19/2013 and 01/09/2013. In the figures, temporary clusters, short-term clusters, mid-term clusters and long-term clusters are denoted by blue, green, yellow and red ellipses, respectively. By means of differentiating clusters according to the lifetime of the clusters, drivers are informed the spots where the severe braking events mostly take place.

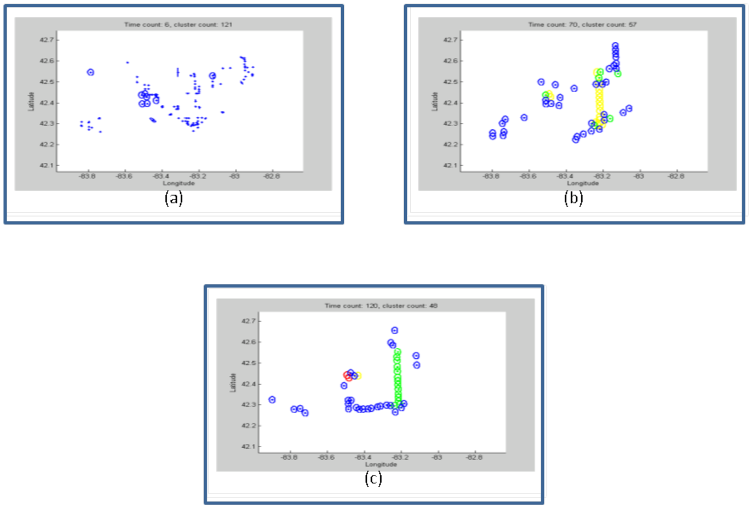


Figure 4.10: Three samples of real time clustering. (Top left part is sampled at 09/15/2013, top right part is sampled at 09/19/2013, and bottom part is sampled at 01/09/2014).

The defined lifetime thresholds differentiate the types of clusters' activities. The lifetime thresholds are chosen by the designers. However, a sensitivity analysis was examined for the four types of clusters by changing the lifetime thresholds. ±20% variations of the defined lifetime thresholds (7 days, 23 days, and 60 days) were considered, respectively. The robustness of the defined lifetime thresholds has been illustrated in the added Figure 4.11.

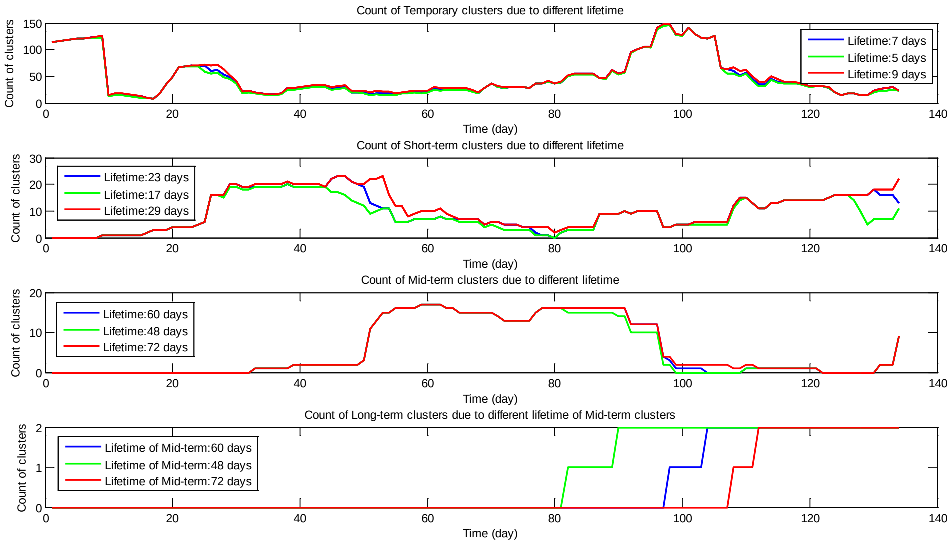


Figure 4.11: Sensitivity analysis of the lifetime thresholds of different types of clusters.

The variation of the severe braking events with time and regions can be presented by the variations of the general clusters over a certain time span, which can be indicated by events data points activities in the clusters over the period. The events data points activities over the period formulate the time series of the events count of the severe braking events occurrence. If we number the general clusters as in Figure 4.12, the activities of the severe braking events from 10/01/2013 to 02/25/2014 in the region denoted by cluster 7 formulate a time series of the events count data shown as Figure 4.13. With the events count data time series, one could construct time series models to describe the activities of the events in different. The time series models of multiple regions are capable of providing spatial-time models of multi-variables, which can be used in analyzing and forecasting the severe braking events correlation between the different regions [36], [37].

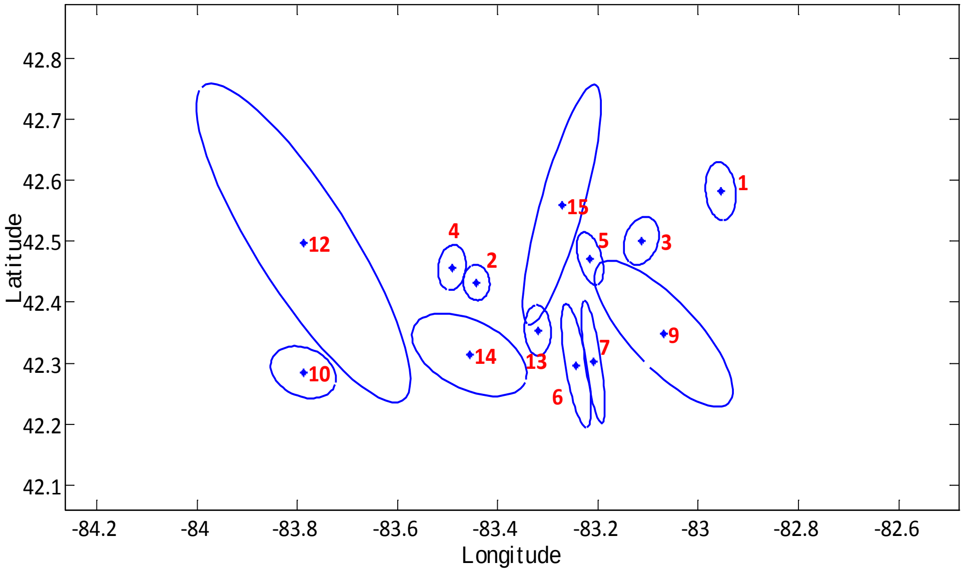


Figure 4.12: Number the recognized general clusters (Cluster 8 and Cluster 11 are not considered as they are outside of the window region).

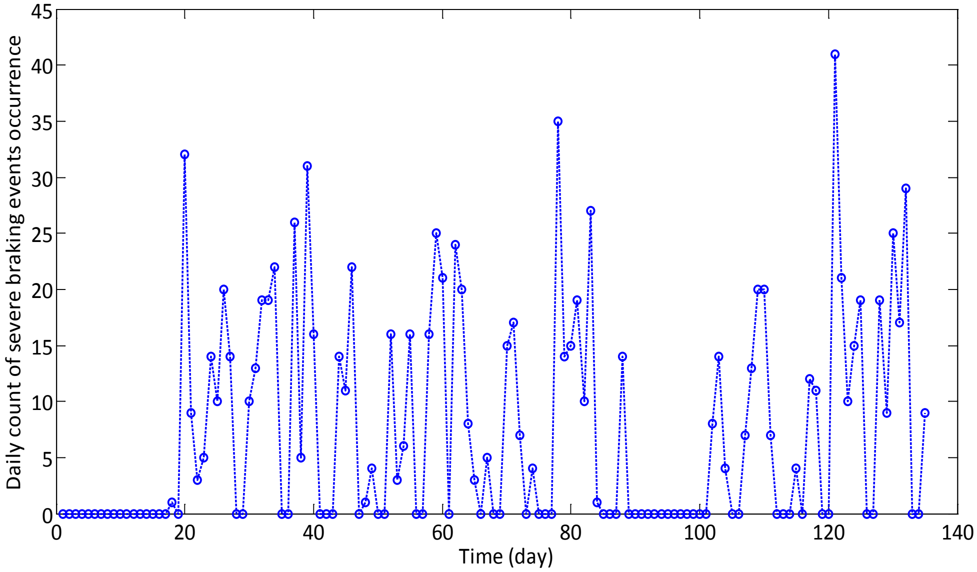


Figure 4.13: Time series of cluster 7 from 10/01/2013 to 02/25/2014

The correlation of two time series can be simply measured by

$$\rho_{z_1 z_2} = \frac{\sum_{t=1}^{N}[(z_1-\mu_{z_1})(z_2-\mu_{z_2})]}{(N-1)\sigma_{z_1}\sigma_{z_2}}, \tag{4.2}$$

where $z_1$and z_2 denote the time series of two neighbor clusters, μ_{z_1} and μ_{z_2} are the means of the two time series, and $\sigma_{z_1}$and σ_{z_2} are the covariance of the two time series. Taking cluster 7 as an example, we examine the correlation between Cluster 7 with its neighbor clusters. The correlation indexes are calculated by Eq. (4.1) and presented in Table 4.2. We can conclude that there is a stronger correlation between Cluster 7 with Cluster 5 than thereof other clusters. It is a little counterintuitive that the correlation between clusters is not strongly dependent on the distance of clusters, observing that the cluster 6 is much closer to the cluster 7 than cluster 5.

Table 4.2: Correlation of cluster 7 with its neighbor clusters

Cluster 7 and Neighbor Clusters	Correlation Index
Cluster 7 – Cluster 5	0.8017
Cluster 7 – Cluster 6	0.4625
Cluster 7 – Cluster 9	0.0422
Cluster 7 – Cluster 13	0.4148
Cluster 7 – Cluster 15	0.2420

4.3 Correlation Identification Procedure Concludion

The identification of severe driving events correlation with location and time procedure can be built as an intelligent analyzer embedded in vehicle to support severe driving events avoidance assistance. In Chapter 3, the correlation identification procedure which is composed of batch clustering and real-time clustering is presented. In Chapter 4, we take severe braking events as example of severe driving events to illustration the implementation of the procedure. The procedure working on historical vehicle driving data provides initial correlation pattern of severe driving events with time and location; working on real-time vehicle driving data provides the variation of the correlation. Such correlations can alert the drivers the time and specific location the severe driving events take place. It helps enhance the drivers awareness to potential driving risk.

Another side of valuable information supplied by correlation identification procedure is that it can extract time series of severe driving events occurrence for any interested regions. The time series of events occurrence in a region is an index of measuring the correlation variation in the region represented by the general cluster variation (Figure 4.12 and Figure 4.13). The time series can be daily count, hourly count, fifteen minutes count, or any periodical count of the events occurrence depending on the timespan we use. Analyzing the events time series of a region is measuring the correlation of the events with the region during the timespan; forecasting the events time series in a region is predicting the future correlation of the events with the region in certain time. At the same time, besides of intra-regional correlation, the inter-regional correlation can be measured by adequate time series models to reveal the correlation among different regions. In addition, anther factors, such as weather, climate, or vehicle speed and other vehicle data, can be considered for modeling the correlation with the events. Essentially, with the help of the time series models, we could take measure to avoid and reduce the events to improve safety driving. It is potentially supplies a method of reducing the severe driving events. Therefore, besides the intelligent analyzer, correlation identification procedure, we continually propose the time series modeling methods for severe braking events count data. The modeling procedure is designed as another module of the safety driving assistance system.

In next two chapters, we provide time series modeling procedure and suggest different time series models. Similarly, we take the time series of severe braking events in the region denoted by Cluster 7 from 10/01/2013 to 02/25/2014 (as Figure 4.13) as example to demonstrate the modeling procedure.

Chapter 5

Time Series Modeling for Severe Driving Events Count Data

A set of successive outcomes of a game, an experiment, or some natural phenomenon is regarded as a sample realization from all possible outcomes that could have been generated by stochastic process. A stochastic process is a rule that maps every possible outcome (e) to a function $Z(t,e)$ [38], [39]. The individual outcomes denoted by e are called *elementary events*. The set of all possible elementary events is called the *sure event* and is denoted by Ω. Usually, we denote a subset of Ω as B. If we observe the outcome e is in B, we say that B has occurred. Intuitively, we specify $P(B)$ as the probability that B will occur. Thus, let (Ω, P) be a probability space.

Define T to be a time index set. A real valued time series is a real valued function $Z(t,e)$ defined on $T \times \Omega$. The function $Z(t,e)$ is often written Z_t for fixed event e. A time series can be considered as a collection of observations $\{Z_t : t \in T\}$ of a random variable indexed sequentially over several time points $t = 1,2, \dots, T$ [40].

In this and next chapter, we will use severe braking events as examples to investigate modeling for severe driving events count data. Here, the random variable, Z_t, represents the severe braking events count data. The sequence over several times points of Z_t represents for the time series of severe braking events. We specifically use the severe braking events count data in Region 7 (the Region denoted by Clusters shown as in Figure 4.12) from 10/01/2013 to 02/25/2014 shown as Figure 4.13 as example to investigate the statistical time series modeling. The goal of the statistical time series modeling is to find a compact representation of the severe driving event count data generating process and, based on the past observations on the variable, to predict the future count values.

Before start the statistical time series modeling, we briefly review the following definitions for a stochastic process $\{Z_t: t = 0,1,2, \ldots\}$ which are necessary for the statistical time series modeling [41].

Definition 1: The *mean* function is defined by

$$\mu_t = E(Z_t), \text{ for } t = 1,2, \ldots, \tag{5.1}$$

where μ_t is the expected value of the process at time t.

Definition 2: The *autocovariance* function, $\gamma_{t,s}$, is defined as

$$\gamma_{t,s} = Cov(Z_t, Z_s) = E[(Z_t - \mu_t)(Z_s - \mu_s)], \text{ for } t, s = 1,2, \ldots. \tag{5.2}$$

Definition 3: The *autocorrelation* function, $\rho_{t,s}$, is given by

$$\rho_{t,s} = Corr(Z_t, Z_s) = \frac{E[(Z_t - \mu_t)(Z_s - \mu_s)]}{\sqrt{Var(Z_t)Var(Z_s)}} = \frac{\gamma_{t,s}}{\sqrt{\gamma_{t,t}\gamma_{s,s}}}. \tag{5.3}$$

Definition 4: The *sampled autocorrelation* function, R_k, at lag k is defined as

$$R_k = \frac{\sum_{t=k+1}^{N}(Z_t - \bar{Z})(Z_{t-k} - \bar{Z})}{\sum_{t=1}^{N}(Z_t - \bar{Z})^2}, \text{ for } k = 1,2, \ldots. \tag{5.4}$$

Definition 5: The *partial autocorrelation* function, ϕ_{kk}, at lag k is defined to be the correlation between the prediction errors; that is

$$\phi_{kk} = Corr(Z_t, Z_{t-k} | Z_{t-1}, \ldots, Z_{t-k+1}). \tag{5.5}$$

Definition 6: *Autoregressive (AR) processes* are processes that the current value of the series is a linear combination of the p most recent past values of itself plus an “innovation” term e_t. Specifically, a p-th autoregressive process satisfies the equations

$$\tilde{Z}_t = \phi_1 \tilde{Z}_1 + \phi_2 \tilde{Z}_{t-2} + \cdots + \phi_p \tilde{Z}_{t-p} + e_t, \text{ where} \tag{5.6}$$

$$Z_t = \tilde{Z}_t + \mu_t. \tag{5.7}$$

Definition 7: *Moving Average (MA) processes* are processes that the current value of the series is a weighted linear combination of present and past white noise. A moving average process of order q satisfies the equation

$$\tilde{Z}_t = e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_l e_{t-q}. \tag{5.7}$$

The AR(p) process and MA(q) process can be conveniently interpreted as AR(p) model and MA(q) model, respectively. We can obtain a quite general time series model if the series is partly autoregressive and partly moving average. That is

$$\tilde{Z}_t = \phi_1 \tilde{Z}_{t-1} + \phi_2 \tilde{Z}_{t-2} + \cdots + \phi_p \tilde{Z}_{t-p} + e_t + \theta_1 e_{t-1} + \theta_2 e_{t-2} + \cdots + \theta_q e_{t-l}. \tag{5.8}$$

Definition 8: We call this model a mixed autoregressive moving average process of orders p and l, abbreviated as *ARMA*(p, q).

ARMA(p, q) model provides a powerful ability for describing stationary time series, whose mean and variance are constant over time. For nonstationary time series, we always take difference operation for the nonstationary time series to make it close to stationary. The first difference operation of Z_t is $\nabla Z_t = Z_t - Z_{t-1}$, and d-th difference operation is presented as $\nabla^d Z_t$. Combining with difference operation, autoregressive integrated moving average (ARIMA) model can be used to describe both stationary and nonstationary time series. Usually, we use ARIMA(p, d, q) denoting p-th order AR part, d-th order difference, and q-th order MA part.

5.1 ARIMA Modeling for Severe Braking Events Count Data Time Series in Region 7

Autoregressive integrated moving average (ARIMA) models family has been used as a very useful methodology in developing the statistical time series modeling. There have been paid great efforts on the ARIMA modeling for time series in different areas from economics [42], [43], social science [44], [45] to different engineering fields [3], [46], [47]. The ARIMA models also known as the Box-Jenkins models do not involve independent variables in their construction; but they make use of the information in the series itself to generate the models. Thus, the ARIMA models rely heavily on their autocorrelation patterns in the data.

The Box-Jenkins methodology is an interactive three-step process for identifying, selecting, and assessing conditional models. The three-step process can be summarized as: model identification, model estimation and model diagnosis. In the first step, the original time series is pre-processed to establish the stationary time series. Based on the stationary time series, we then identify a stationary conditional mean model for the data. In the second step, we specify the model, and estimate the model parameters. In the third step, we implement residual analysis to ensure the model describes the data adequately. If the model describes the data adequately and not too complex, we could use it to forecast the data over a future time horizon.

In the following, we use the Box-Jenkins methodology to build ARIMA models for the example time series of severe braking events daily count data in Region 7 from 10/01/2013 to 02/25/2014. Due to vehicle maintenance, the data from *Day 1* to *Day 16* and *Day 88* to *Day 100* are missed (we define *Day 1* as 10/01/2013). Thus, we could use the first continuously time series (*Day 17 – Day 89*) for building ARIMA models, and employ the second continuously times (*Day 101 – Day 134*) for validating the proposed models.

5.1.1 ARIMA Modeling: Model Identification

We use the variable $Z_t^{(7)}$ to denote the stochastic process to generate the time series of severe braking events count data in region 7 from 10/01/2013 to 02/25/2014 shown as in Figure 5.1. The time series of $Z_t^{(7)}$ is not stationary, which means that the mean and covariance function is not constant over time. One operation for the time series $Z_t^{(7)}$ to make it stationary is first-difference operation. Figure 5.2 displays the plot of the first-order difference of the time series $Z_t^{(7)}$.

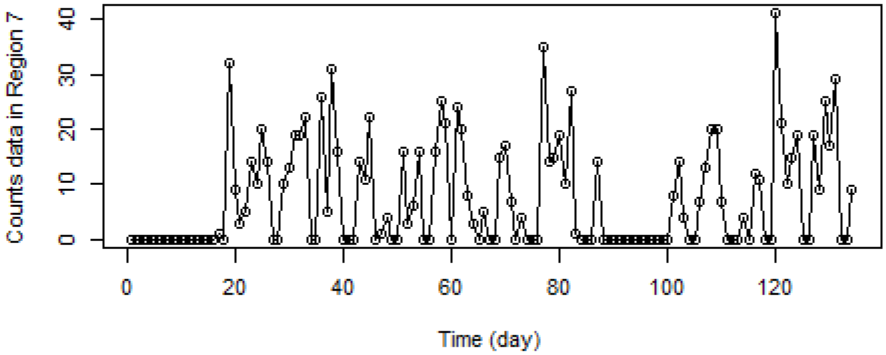


Figure 5.1: Time series of $Z_t^{(7)}$ from 10/01/2013 to 02/25/2014.

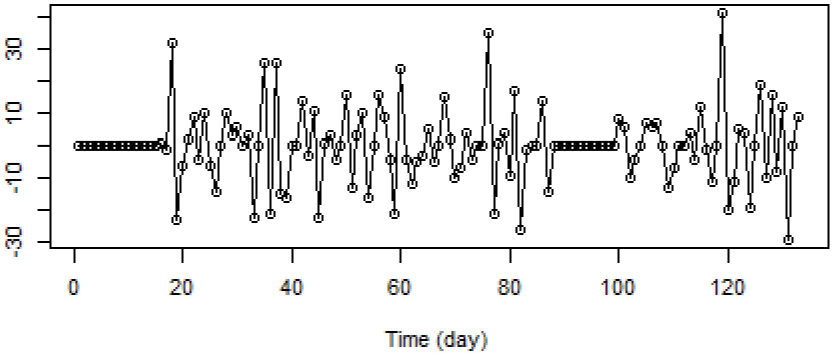


Figure 5.2: First-order difference of the time series $Z_t^{(7)}$ from 10/01/2013 to 02/25/2014.

After the first-difference operation, we can start with the ARIMA model structure:

$$\nabla^1 Z_t^{(7)} = \phi_1 \nabla^1 Z_{t-1}^{(7)} +\cdots +\phi_p \nabla^1 Z_{t-p}^{(7)} + e_t - \theta_1 e_{t-1} -\cdots -\theta_q e_{t-q} \text{ and} \tag{5.9}$$

$$\tilde{Z}_t^{(7)} = \phi_1 \tilde{Z}_t^{(7)} +\cdots +\phi_p \tilde{Z}_{t-p}^{(7)} + e_t - \theta_1 e_{t-1} -\cdots -\theta_q e_{t-q}. \tag{5.10}$$

5.1.2 ARIMA Modeling: Model Estimation

Model estimation is to specify a class of model out of ARIMA, suggest the orders for the model and estimate the parameters in the model. The autocorrelation (ACF) and partial autocorrelation (PACF) functions help to specify the class of model and decide the order for the model. Once we have a candidate model, many methods, such as the method of moment, the (recursive) least squares estimation, maximum likelihood estimation, etc., can be employed to estimate the parameters. We choose the maximum likelihood estimation (MLE) method here and afterward.

Theoretical autocorrelation and partial autocorrelation for conditional mean model AR, MA, and ARMA are quite different for each model. The following table summarizes the autocorrelation function (ACF) and partial autocorrelation function (PACF) behavior for these models. In Figure 5.3, the autocorrelation and partial correlation of the time series of $\tilde{Z}_t^{(7)}$ from *Day 16* to *Day 87* with different time lags are presented. In the ACF and PACF plot, the cut value shown as dotted line in the plots is decided by that how much confidence level of accepting the examined series as white noise. The cut value is calculated as $\pm 1.96/\sqrt{N}$, where N is number of samples and in this example $N = 72$, and 1.96 comes from $F(x = 1.96) = 0.95$, if $x \sim N(0,1)$. We say we are 95% confident that the series is not white noise and autocorrelated if ACF values are not within the within the interval (the stems of ACF are bounded by the two dotted lines).

Table 5.1: ACF and PACF behavior for different classes of ARMA models.

Conditional Mean Model	ACF	PACF
$AR(p)$	Decay exponentially with time lags	Cuts off after p lags
$MA(q)$	Cuts off after q lags	Decay exponentially with time lags
$ARMA(p,q)$	Decay exponentially with time lags	Decay exponentially with time lags

PACF and ACF plots suggest that an $ARMA(1{,}1)$ model.

$$\textbf{Model 1: } \tilde{Z}_t^{(7)} = \phi_1 \tilde{Z}_{t-1}^{(7)} + e_t - \theta_1 e_{t-1}. \tag{5.11}$$

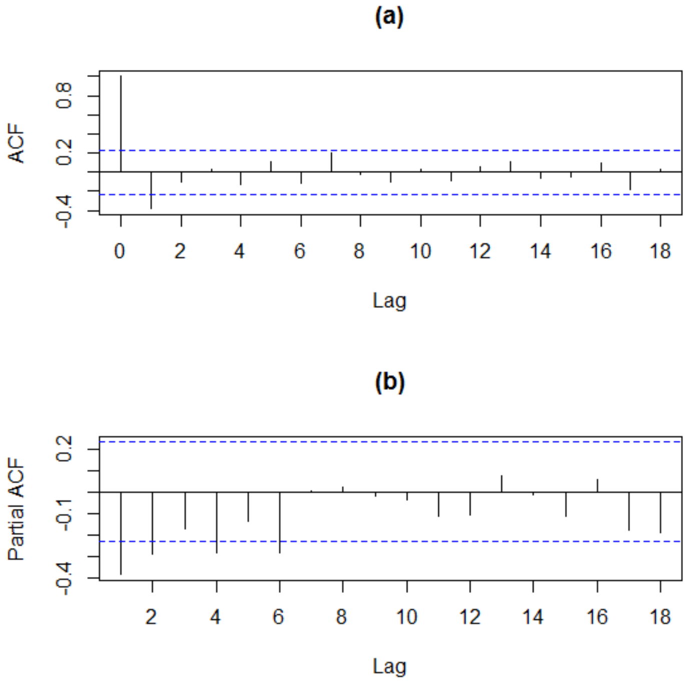


Figure 5.3: ACF and PACF of $\tilde{Z}_t^{(7)}$ from Day 16 to Day 87.

The model parameters ϕs and θs can be estimated by maximum likelihood estimation (MLE) method. The unobserved error e_t can be rewrite as,

$$e_t = \tilde{Z}_t^{(7)} - \phi_1 \tilde{Z}_{t-1}^{(7)} + \theta_1 e_{t-1}. \tag{5.12}$$

We assume the errors, e_t, is normal distribution with mean 0 and variance equal to σ_e^2, which means that the probability density function (PDF) of each e_t is

$$f(e_t|\sigma_e) = (2\pi\sigma_e^2)^{-\frac{1}{2}} \exp\left(-\frac{e_t^2}{2\sigma_e^2}\right). \tag{5.13}$$

Then the joint PDF for $e_1, e_2, \dots, e_n$ is

$$f(e_1, e_2 \dots, e_N|\sigma_e) = (2\pi\sigma_e^2)^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma_e^2}\sum_{t=1}^{n} e_t^2\right). \tag{5.14}$$

Likelihood function is a function to estimate the probability parameters or parameters of a statistical model to fit a specific event outcome that has already occurred. It is a “reverse” sense of probability function: probability function is a function of the unknown data Z for the given parameters μ and σ, whereas the likelihood is a function of the unknown parameters μ and σ for the given data Z [48]. Therefore, likelihood is event outcomes oriented method to find out the probability function or statistical model. Then the likelihood function can written as the follows when using the observed Z_t instead,

$$L(\phi, \theta, \sigma_e|Z_1, Z_2 \dots, Z_N) = (2\pi\sigma_e^2)^{-\frac{1}{2}} \exp\left(-\frac{1}{2\sigma_e^2}\ S(\phi, \theta)\right), \tag{5.15}$$

where,

$$S(\phi, \theta) = \sum_{t=1}^{n} e_t^2. \tag{5.16}$$

Maximum likelihood estimation (MLE) is the method to identify the values of the parameters of probability function or statistical model which most likely generates the specific event outcome [41]. Therefore, the following maximization is formulated,

$$\max_{\phi,\theta}: J = L(\phi, \theta, \sigma_e|Z_1, Z_2 \dots, Z_N), \tag{5.17}$$

or equivalently, we transform the maximization to be minimization with log-likelihood function,

$$\min_{\phi,\theta}: J = -\log L(\phi, \theta, \sigma_e|Z_1, Z_2 \dots, Z_N). \tag{5.18}$$

We can solve the unconstrained optimization problem by well-known Newton-

Raphson algorithm in MATLAB or R [49], [50], [51], [52]. In Table 5.2, the parameters of Model 1 are solved by MLE implemented by "*nlm*" command in R.

Table 5.2: Model 1 parameters estimation from MLE by R.

Maximum Likelihood Estimates for ARIMA(1,1,1)			
Coefficients:	ϕ_1	θ_1	intercept
	-0.1418	-1	-0.0263
Standard errors	0.1186	0.037	0.0619
Sigma^2 estimated as 92.71: log likelihood = -267.23, AIC = 542.46			

Any estimated parameter that proved to be statistically insignificant should be removed from the model. We usually use the ratio $\left|\frac{coefficient\ value}{coefficient\ standard\ error}\right|$ to check if the coefficient is significant. If the ratio is larger than 1.96 ($F(x = 1.96) = 0.95$, if $x \sim N(0,1)$), we say we are 95% confident that the true value of the coefficient is in the confidence interval and the coefficient is acceptable at the 0.05 level. The lag 1 autocorrelation term is insignificant ($|-0.1418/0.1186| = 1.196 < 1.96$). Thus, we drop this insignificant term and reduce the Model 1 to a subset model MA(1) model, Model 2, and recalculate the estimation for θ_1 in Table 5.3,

$$\textbf{Model 2: } \tilde{Z}_t^{(7)} = e_t - \theta_1 e_{t-1}. \tag{5.19}$$

Table 5.3: Model 2 parameters estimation from MLE by R.

Maximum Likelihood Estimates for ARIMA(0,1,1)		
Coefficients:	θ_1	intercept
	-1	-0.0263
Standard errors	0.0382	0.0539
Sigma^2 estimated as 94.19: log likelihood = -267.94, AIC = 541.88		

Sigma^2 is the variance of the errors, e_t. Akaik information criterion (AIC) is a measure of the relative quality of a statistical model for a given set of data from the perspective of information entropy, as measured by Kullback-Leibler divergence. If we denote $\log L(\theta)$ as the value of the maximized likelihood objective function for a model with k parameters fit to N data points, then AIC for the model is $-2\log L(\theta) + 2k.$ AIC is commonly used for model selection when comparing candidate models: the smaller value of AIC of the model, the better interpretation of the model to the actual data. When do the comparisons of AIC between the Model 1 and Model 2, the value is almost same. Thus, we choose the Model 2 as the selected model for diagnosis. Model 2 indicates that the time series of $\tilde{Z}_t^{(7)}$ is mutually independent and it is more like to come from linear combination from white noise.

5.1.3 ARIMA Modeling: Model Diagnosis

After a candidate model has been found, the model should be diagnostic checking to determine if the model adequately represents the time series. The candidate model can be inadequacy with a significant correlation among the residuals, or with the estimated parameters proved to be statistically insignificant. The residuals from an adequate fitted model should be closed to white noise, i.e. should be distributed normally with mean zero and variance equal to σ^2. In Figure 5.4, the ACF and PACF plots of the residuals of Model 2 prove the residuals to be insignificant correlated. However, in Figure 5.5, the histogram of the residuals of $Z_t^{(7)}$ to Model 2 simulation rejects the residuals to be white noise.

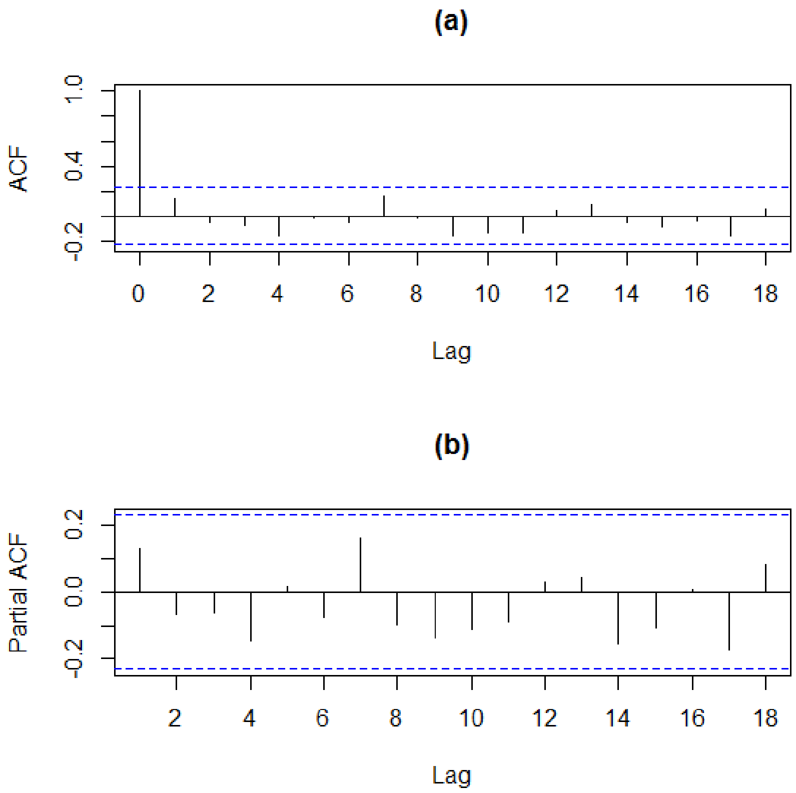


Figure 5.4: ACF and PACF of the residuals of Model 2.

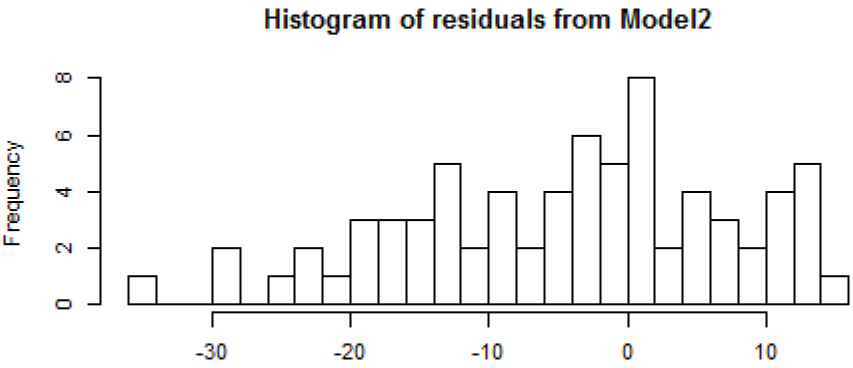


Figure 5.5: Histogram of the residuals of Model 2.

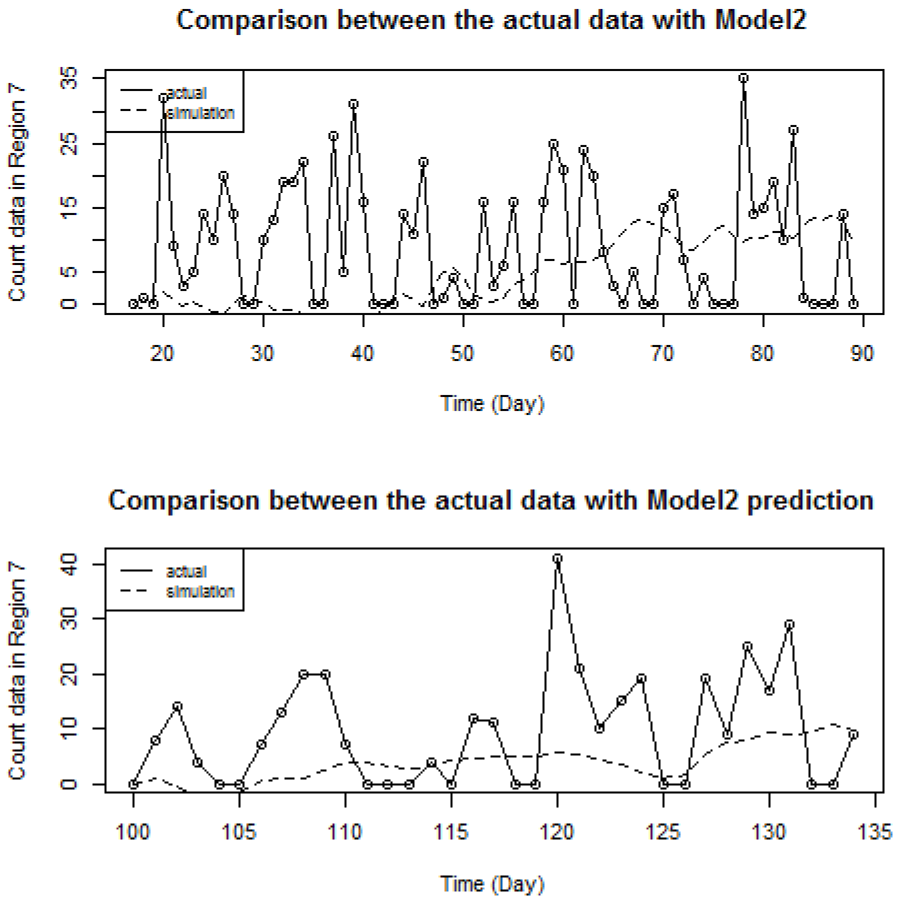


Figure 5.6: Comparison between actual data with simulation results from Model 2.

The histogram of the residuals of Model 2 does not perform in normal distribution. The model is not fitted well with the actual data observed illustrated by Figure 5.6 as well. Noticed in Table 5.3, the standard variation of the model residual is very large. Therefore, we conclude that the ARIMA modeling does not provide a good fitting for this time series only using the information of the time series itself. However, it does not mean the ARIMA modeling is meaningless. ARIMA modeling can be possible valuable for other time series data with different sampling time in the region or other regions. It is always chosen as the first step for building statistical time series models for the interested regions.

In next section, we will consider an important factor, the count time series of neighbor regions, to show how big the affectedness of other regions to the interested region and improve the ARIMA modeling.

5.2 Space-Time ARIMA modeling for Severe Braking Events Count Data Time Series in Region 7

Single time series ARIMA modeling only consider the time series of itself and unavoidably miss other information, e.g., spatial effect. Space-time time series modeling expresses the observation at time t and zone i, $Z_{(i)}(t)$ or $Z_t^{(i)}$, as linear combination of past observations at zone i and the neighboring zones. The advantage of spatial-time series modeling is that it correlates spatial factor to consider multi-regional time series as a necessary part of the model to improve the modeling adequacy for single region.

Space-time ARIMA time series modeling proposed by Pfeifer and Deutsch in 1980s provides powerful methodology to develop STARIMA models ranging from environmental to epidemiological and econometric [53], [36], [54], [55]. The STARIMA modeling used in traffic flow modeling starts in early 2000s [37], [58], [57]. Similar to ARIMA modeling, the STARIMA modeling is a three-stage procedure as well: identification, estimation and diagnostic checking. In the following, we will build STARIMA models for severe braking event count data time series in Region 7 as example to illustrate the three-stage procedure.

To assist in the formulation of this STARIMA model, the definition of the spatial lag operator is provided.

Definition 9: Analogous to the lag operator in the time domain, define $L^{(l)}$ be the spatial lag operator of spatial order l, and the operation rule is:

$$L^{(0)}Z_i(t) = Z_i(t), L^{(l)}Z_i(t) = w_{ij}^{(l)}Z_j(t), \tag{5.20}$$

where $w_{ij}^{(l)}$ are a set of weights with $\sum_{j=1}^{m} w_{ij}^{(l)} = 1$ and i stands for the region of interest for analyzing. The spatial order l in $w_{ij}^{(l)}$ reflects the distance level between the region of interest i to a neighbor region j. $l = 0$ reflects the neighbor region j is the region of interest; when l =1, it reflects a neighbor region j that is closest to the region of interest; when l =2, it reflects a neighbors j that is further away from the region of interest than the neighbors reflected by $l = 1$, but closer than the neighbors reflected by $l = 3$. The value of $w_{ij}^{(l)}$ is dependent on the number of neighbors with the same spatial order l to the site of interest. For example, there are two neighboring regions with spatial order $l = 2$ to region i, then, for each of the two regions j, $w_{ij}^{(2)} = \frac{1}{2}$.

Based on the identified 13 Regions in Figure 4.12, we have the following distance level table (Table 5.4) and weight matrices.

Table 5.4: Distance level between the regions in Figure 4.12.

Order	1	2	3
Region 1			
Region 2	4	14	13,15
Region 3			5,9,15
Region 4	2	14	
Region 5		7,15	3,6,9
Region 6	7	13	5,14,15
Region 7	6	5	9,13,15
Region 8			11
Region 9			3,5,7
Region 10			12
Region 11			8
Region 12			10
Region 13		6,15	2,7,14
Region 14		2,4	6,13
Region 15		5,13	2,3,6,7

Therefore, we deduce the weighting matrices for the regions with order 1, 2 and 3, respectively.

$$W_1 = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&1&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&1&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&1&0&0&0&0&0&0 \\ 0&0&0&0&0&1&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \qquad W_1 = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0.5&0&0&0&0&0&0.5 \\ 0&0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&1&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0.5&0&0&0&0&0&0&0.5 \\ 0&0.5&0&0.5&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0.5&0&0&0&0&0&0.5&0&0 \end{bmatrix}$$

$$W_1 = \begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&0&0.5&0&0.5 \\ 0&0&0&0&1/3&0&0&1/3&0&0&0&0&1/3 \\ 0&0&0&0&0&0&0&0&0&0&0&0&0 \\ 0&0&1/3&0&0&1/3&0&1/3&0&0&0&0&0 \\ 0&0&0&0&1/3&0&0&0&0&0&0&1/3&1/3 \\ 0&0&0&0&0&0&0&1/3&0&0&1/3&0&1/3 \\ 0&0&1/3&0&1/3&0&1/3&0&0&0&0&0&0 \\ 0&0&0&0&0&0&0&0&0&1&0&0&0 \\ 0&0&0&0&0&0&0&0&1&0&0&0&0 \\ 0&1/3&0&0&0&0&1/3&0&0&0&0&1/3&0 \\ 0&0&0&0&0&0.5&0&0&0&0&0.5&0&0 \\ 0&0.25&0.25&0&0&0.25&0.25&0&0&0&0&0&0 \end{bmatrix}$$

In general, the STARMA model can be presented as

$$Z_{(i)}(t) = \sum_{k=1}^{p} \sum_{l=0}^{\lambda_k} \phi_{kl} L^{(l)}\, Z_{(i)}(t-k) - \sum_{k=1}^{q} \sum_{l=0}^{m_k} \theta_{kl} L^{(l)}\, \epsilon_{(i)}(t-k) + \epsilon_{(i)}(t), \quad (5.21)$$

where p is the autoregressive order, q is the moving average order, λ_k is the spatial order of k^{th} autoregressive term, m_k is the spatial order of k^{th} moving average term, ϕ_{kl} and θ_{kl} are parameters, and $\epsilon_i(t)$ are random normal errors.

5.2.1 STARIMA Modeling: Model Identification

In univariate time series modeling, autocorrelation and partial autocorrelation are employed as primary tools in model identification. Similarly, we use space-time autocorrelation and partial autocorrelation to identify an adequate model, where the space-time autocorrelation and partial autocorrelation are deduced from autocovariance and Yule-Walker equation.

Definition 10: *Space-time covariance* between l^{th} and k^{th} order neighbors at time lag s is defined as

$$\gamma_{lk}(s) = E\left\{\sum_{i=1}^{N} \frac{\left[L^{(l)} z_i(t)\right]^T \left[L^{(k)} z_i(t+s)\right]}{N}\right\} = E\left\{\sum_{i=1}^{N} \frac{[W_l z_i(t)]^T [W_k z_i(t+s)]}{N}\right\}. \tag{5.22}$$

Therefore, space-time autocorrelation function (STACF) between l^{th} and k^{th} order neighbors at time lag s is,

$$\rho_{lk}(s) = \frac{\gamma_{lk}(s)}{[\gamma_{ll}(0)\gamma_{kk}(0)]^{1/2}}. \tag{5.23}$$

The space-time partial autocorrelation function (STPAF) between l^{th} and k^{th} order neighbors at time lag s is solved from the Yule-Walker equations,

$$\left[\begin{array}{c} \gamma_{00}(1) \\ \vdots \\ \gamma_{\lambda 0}(1) \\ \hline \gamma_{00}(2) \\ \vdots \\ \gamma_{\lambda 0}(2) \\ \hline \vdots \\ \hline \gamma_{00}(k) \\ \vdots \\ \gamma_{\lambda 0}(k) \end{array}\right] = \left[\begin{array}{ccc|ccc|c|ccc} \gamma_{00}(0) & \cdots & \gamma_{0\lambda}(0) & \gamma_{00}(-1) & \cdots & \gamma_{0\lambda}(-1) & \cdots & \gamma_{00}(1-k) & \cdots & \gamma_{0\lambda}(1-k) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma_{\lambda 0}(0) & \cdots & \gamma_{\lambda\lambda}(0) & \gamma_{\lambda 0}(-1) & \cdots & \gamma_{\lambda\lambda}(-1) & \cdots & \gamma_{\lambda 0}(1-k) & \cdots & \gamma_{\lambda\lambda}(1-k) \\ \hline \gamma_{00}(1) & \cdots & \gamma_{0\lambda}(1) & \gamma_{00}(0) & \cdots & \gamma_{0\lambda}(0) & \cdots & \gamma_{00}(2-k) & \cdots & \gamma_{0\lambda}(2-k) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma_{\lambda 0}(1) & \cdots & \gamma_{\lambda\lambda}(1) & \gamma_{\lambda 0}(0) & \cdots & \gamma_{\lambda\lambda}(0) & \cdots & \gamma_{\lambda 0}(2-k) & \cdots & \gamma_{\lambda\lambda}(2-k) \\ \hline \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \hline \gamma_{00}(k-1) & \cdots & \gamma_{0\lambda}(k-1) & \gamma_{00}(k-2) & \cdots & \gamma_{0\lambda}(k-2) & \cdots & \gamma_{00}(0) & \cdots & \gamma_{0\lambda}(0) \\ \vdots & \ddots & \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \ddots & \vdots \\ \gamma_{\lambda 0}(k-1) & \cdots & \gamma_{\lambda\lambda}(k-1) & \gamma_{\lambda 0}(k-2) & \cdots & \gamma_{\lambda\lambda}(k-2) & \cdots & \gamma_{\lambda 0}(0) & \cdots & \gamma_{\lambda\lambda}(0) \end{array}\right] \left[\begin{array}{c} \phi_{10} \\ \vdots \\ \phi_{1\lambda} \\ \hline \phi_{20} \\ \vdots \\ \phi_{2\lambda} \\ \hline \vdots \\ \hline \phi_{30} \\ \vdots \\ \phi_{3\lambda} \end{array}\right]. \tag{5.24}$$

The coefficients ϕ_{kl} solved from the above Yule-Walker equation as $l = 0,1,\ldots,\lambda$ and $k = 0,1,\ldots$ are called *the space-time partial correlation function* of spatial order λ. The spatial order λ is at least as large as the maximum spatial order of hypothesized model.

When choose a class of ARMA models for a stationary time series, we have the autocorrelation function (ACF) and partial autocorrelation function (PACF) behavior suggestion for the candidate model. Similarly, when choose a class of STARMA models for a stationary time series, we have the similar STACF and STPACF behavior suggestion table for a candidate STARIMA model. In the following, the space-time autocorrelation and partial correlation are calculated for site 7, respectively.

Table 5.5: STACF and STPACF behavior for different classes of STARIMA models.

Model	STACF	STPACF
$STAR(p_\lambda)$	Decay exponentially with space and time	Cuts off after p lags in time and λ_p lags in space
$STMA(q_m)$	Cuts off after q lags in time and m lags in spatial lags	Decay exponentially with space and time
$STARMA(p_\lambda, q_m)$	Decay exponentially with space and time	Decay exponentially with space and time

Table 5.6: Space-Time autocorrelation $\boldsymbol{k=0}$, $\boldsymbol{l=0-3}$, $\boldsymbol{T=10}$.

Time lag	Site Lag 0	Lag 1	Lag 2	Lag 3
1	-0.3934	0.0806	-0.0970	-0.0273
2	-0.0959	-0.0689	-0.0798	-0.0109
3	0.0033	0.0104	-0.0383	-0.0107
4	-0.0091	-0.0390	0.0288	-0.0267
5	-0.0571	-0.0019	-0.0581	0.0194
6	-0.0122	0.0007	0.0152	-0.0268
7	0.1313	0.0403	0.0731	0.0328
8	-0.0173	0.0066	-0.0013	-0.0016
9	-0.0586	-0.0455	-0.0805	-0.0173
10	-0.0035	0.0569	0.0605	0.0442

Table 5.7: Space-Time partial autocorrelation $\boldsymbol{k=0}$, $\boldsymbol{l=0-3}$, $\boldsymbol{T=25}$.

Time lag	Site Lag 0	Lag 1	Lag 2	Lag 3
1	-0.9030	0.1623	0.0967	-0.0554
2	-0.9522	0.1318	0.1193	-0.0370
3	-0.9602	0.1325	0.0539	-0.1090
4	-0.9826	0.1514	0.1177	-0.1125

Table 5.7 (continued): Space-Time partial autocorrelation $\boldsymbol{k = 0, l = 0 - 3, T = 25}$.

Time lag	Site Lag 0	Lag 1	Lag 2	Lag 3
5	-1.0354	0.1806	0.0982	-0.1198
6	-1.0358	0.1746	0.1198	-0.2272
7	-0.9049	0.2860	0.2113	-0.1879
8	-0.9326	0.2669	0.2321	-0.1905
9	-0.9778	0.2441	0.2029	-0.2399
10	-0.9742	0.3410	0.3330	-0.1771
11	-0.9652	0.2971	0.2980	-0.2618
12	-0.9469	0.2222	0.1813	-0.2375
13	-0.8582	0.2620	0.2299	-0.1911
14	-0.8717	0.2045	0.1380	-0.2466
15	-0.8378	0.1856	0.1830	-0.2767
16	-0.7783	0.1721	0.1793	-0.1639
17	-0.8305	0.0463	0.0813	-0.1903
18	-0.7377	0.0265	0.1537	-0.2023
19	-0.5959	0.0934	0.2979	-0.0964
20	-0.5222	0.0743	0.2830	-0.1073
21	-0.4012	0.1082	0.2705	-0.0037
22	-0.3551	0.1134	0.2837	0.0010
23	-0.2790	0.1173	0.1795	-0.0437
24	-0.1870	0.1217	0.1967	0.0083
25	-0.0895	0.1073	0.0571	0.0807

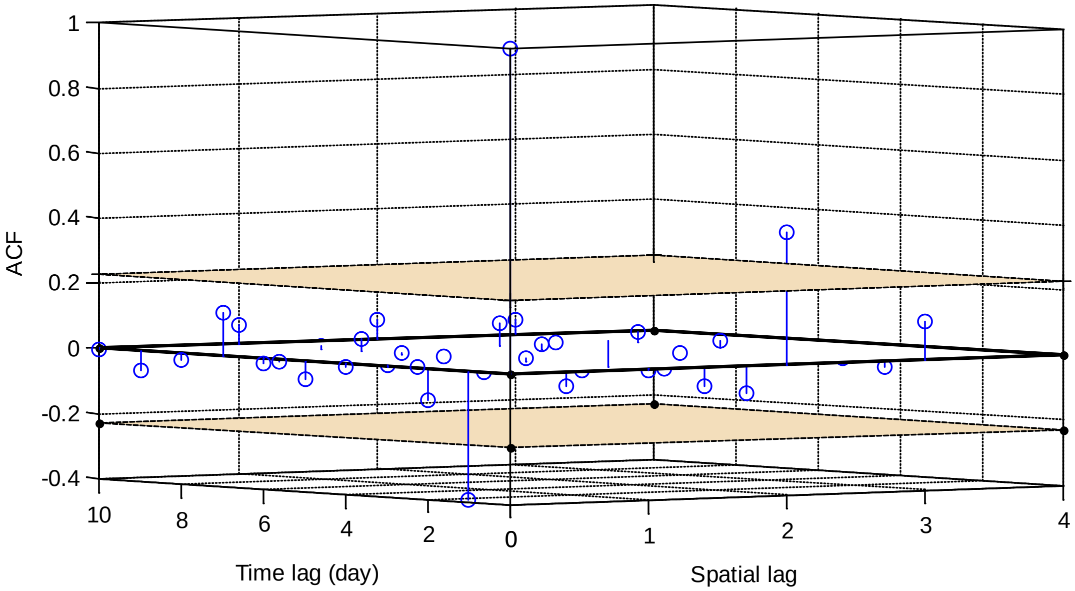


Figure 5.7: Spatial-time autocorrelation plot.

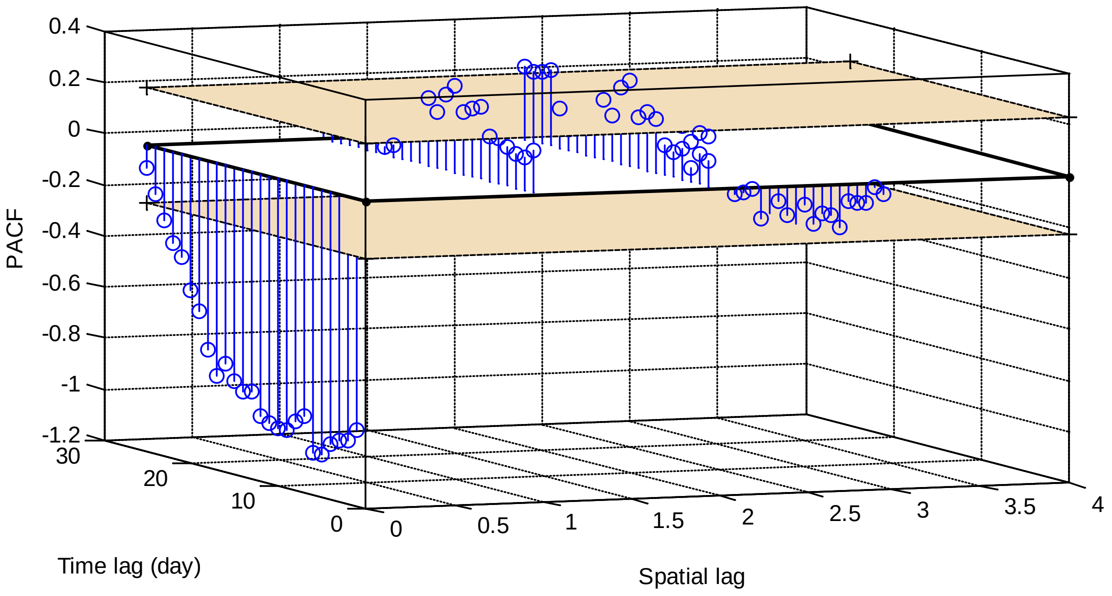


Figure 5.8: Spatial-time partial autocorrelation plot.

From the above calculation and plots, we notice that the lag $l = 2$ is significant in ACF plot. For Region 7, the region with the spatial lag 2 is the Region 5 from the Table 5.4. Thus, we can conclude that the Region 5 and Region 7 are more correlational than other neighbor regions, which is convinced by Table 4.2. Therefore, we can imply a STARIMA model as:

$$\tilde{Z}_t^{(7)} = \phi_{02} W_2 \tilde{Z}_t^{(7)} + e_t - \theta_{10} e_{t-1} - \theta_{11} W_1 e_{t-1} - \theta_{12} W_2 e_{t-1}. \tag{5.25}$$

If we combine the e_{t-1} terms, then the model is changed to be

$$\textbf{Model 3: } \tilde{Z}_t^{(7)} = \phi_{02} W_2 \tilde{Z}_t^{(7)} + e_t - \hat{\theta}_{10} e_{t-1}. \tag{5.26}$$

Since $W_2 \tilde{Z}_t^{(7)} = \tilde{Z}_t^{(5)}$, we can rewrite Model 3 as

$$\textbf{Model 3: } \tilde{Z}_t^{(7)} = \phi_{02} \tilde{Z}_t^{(5)} + e_t - \hat{\theta}_{10} e_{t-1}, \tag{5.27}$$

where $\tilde{Z}_t^{(5)} = \nabla^1 Z_t^{(5)}$. The time series of $Z_t^{(5)}$ and its first-order difference time series from 10/01/2013 to 02/25/2014 are plotted as Figure 5.9 and Figure 5.10, respectively.

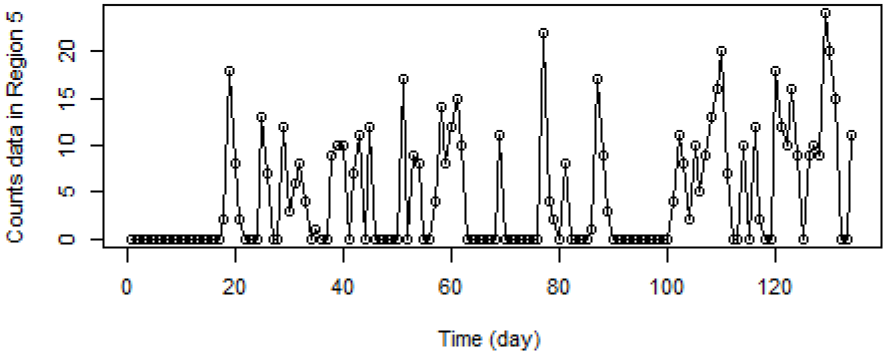


Figure 5.9: Time series of $Z_t^{(5)}$ from 10/01/2013 to 02/25/2014.

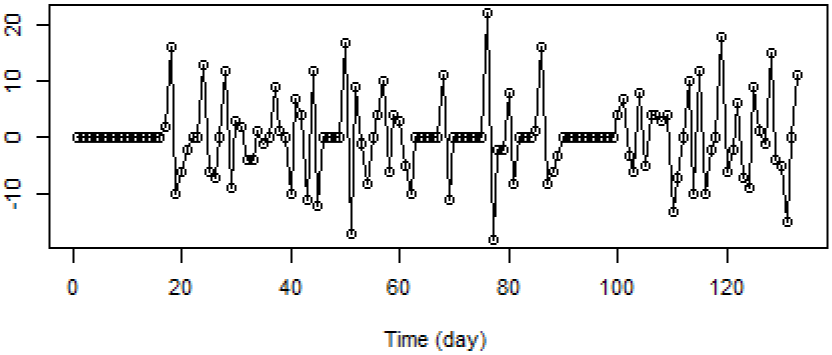


Figure 5.10: First-order difference of the time series $Z_t^{(5)}$ from 10/01/2013 to 02/25/2014.

5.2.2 STARIMA Modeling: Model Estimation

After the candidate STARMA Model 3 has been selected from the identification phase, we will estimate the parameters of the candidate model. Similar to the parameter estimation in ARIMA modeling, we use maximum likelihood estimation (MLE) method as well.

The model is rewritten as

$$e_t - \hat{\theta}_{10} e_{t-1} = \tilde{Z}_t^{(7)} - \phi_{02} W_2 \tilde{Z}_t^{(7)}. \tag{5.28}$$

If we assume the error terms is from same normal distribution process, the error distribution function can be presented as

$$f\left(e_1, e_2, \dots, e_n \middle| \phi, \theta, \sigma_e, \tilde{Z}_t^{(7)}\right) = (2\pi\sigma_e^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma_e^2} S(\phi, \theta, \sigma_e, Z_t)\right\}, \tag{5.29}$$

where

$$S\left(\phi, \theta, \sigma_e, \tilde{Z}_t^{(7)}\right) = \sum_{t=1}^{n} e_t^2 = \sum_{t=1}^{n} \left[\frac{\tilde{Z}_t^{(7)} - \phi_{02} W_2 \tilde{Z}_t^{(7)}}{1 - \hat{\theta}_{10}}\right]^2. \tag{5.30}$$

The function $S(\phi, \theta, \sigma_e, Z_t)$ is called the *conditional sum-of-squares function* of the candidate model. The corresponding likelihood function is then deduced as

$$L\left(\phi, \theta, \sigma_e \middle| \tilde{Z}_t^{(7)}\right) = (2\pi\sigma_e^2)^{-n/2} \exp\left\{-\frac{1}{2\sigma_e^2} S(\phi, \theta, \sigma_e, Z_t)\right\}. \tag{5.31}$$

The parameters ϕ, θ and σ_e can then be identified by maximize the likelihood under the observation of $\tilde{Z}_t^{(7)}$ and $\tilde{Z}_t^{(5)}$. To make the MLE convenient, we proceed with the likelihood function to be log-likelihood function for optimization, as the logarithm of the likelihood function is more convenient to work with. The log-likelihood function denoted as $\mathcal{L}(\phi, \theta, \sigma_e)$ is given by

$$\mathcal{L}(\phi, \theta, \sigma_e) = -\frac{n}{2}\log(2\pi) - \frac{n}{2}\log(\sigma_e^2) - \frac{1}{2\sigma_e^2} S(\phi, \theta, \sigma_e). \tag{5.32}$$

Then we can formulate the minimization problem as Eq. (5.33),

$$\min_{\phi,\theta}: J = -\mathcal{L}(\phi, \theta, \sigma_e). \tag{5.33}$$

The minimization problem is solved with unconstrained minimization algorithm and the parameters of the STARIMA model is solved in Table 5.8.

Table 5.8: Model 3 parameters estimation from MLE by unconstrained minimization algorithm.

Maximum Likelihood Estimates for STARIMA(1_2,1,1_0)				
Coefficients:	ϕ_{01}	ϕ_{02}	$\hat{\theta}_{10}$	intercept
	0	1.0338	0.9	0
Sigma^2 estimated as 25.3830: log likelihood = -218.5904, AIC = 441.1809				

Therefore, the STARIMA model is formulated as

$$\tilde{Z}_t^{(7)} = 1.03338\,\tilde{Z}_t^{(5)} + e_t - 0.9e_{t-1}. \tag{5.34}$$

All the parameters estimated in Model 3 are assumed to be significant. Compared with Model 2, the $MA(1)$ part is still kept and additional term is the spatial factor of Region 5. The AIC of Model 3 and standard deviation of model error are all less than those indices of Model 2. In next section, we implement diagnostic checking with residuals and prediction of Model 3 to demonstrate the adequacy of Model 3.

5.2.3 STARIMA Modeling: Model Diagnosis

The initial diagnosis for residuals of the Model 3 to actual data is to check whether the residuals are significantly autocorrelated. Figure 5.11 shows the residuals are not significantly autocorrelated. The histogram plot of the residuals of Model 3 to actual data is then presented in Figure 5.12. The histogram is close to normal distribution shape. However, the histogram has a positive displacement away from zero and presents a short and wide shape, which means the standard deviation of the errors is large.

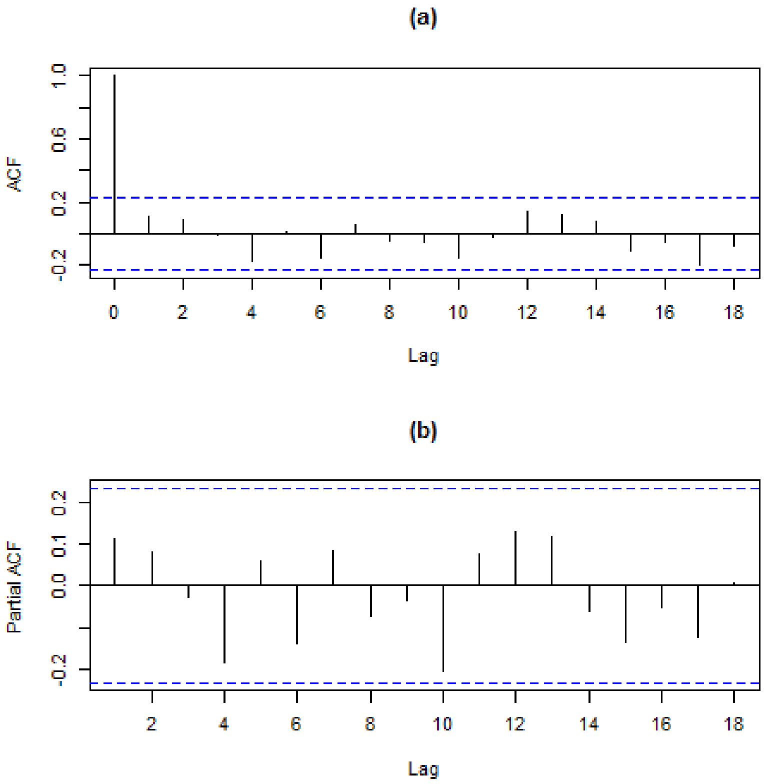


Figure 5.11: ACF and PACF of the residuals of Model 3.

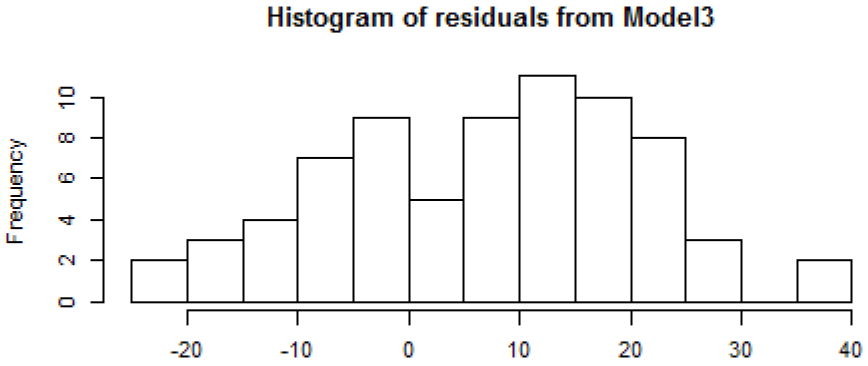


Figure 5.12: Histogram of the residuals of Model 3.

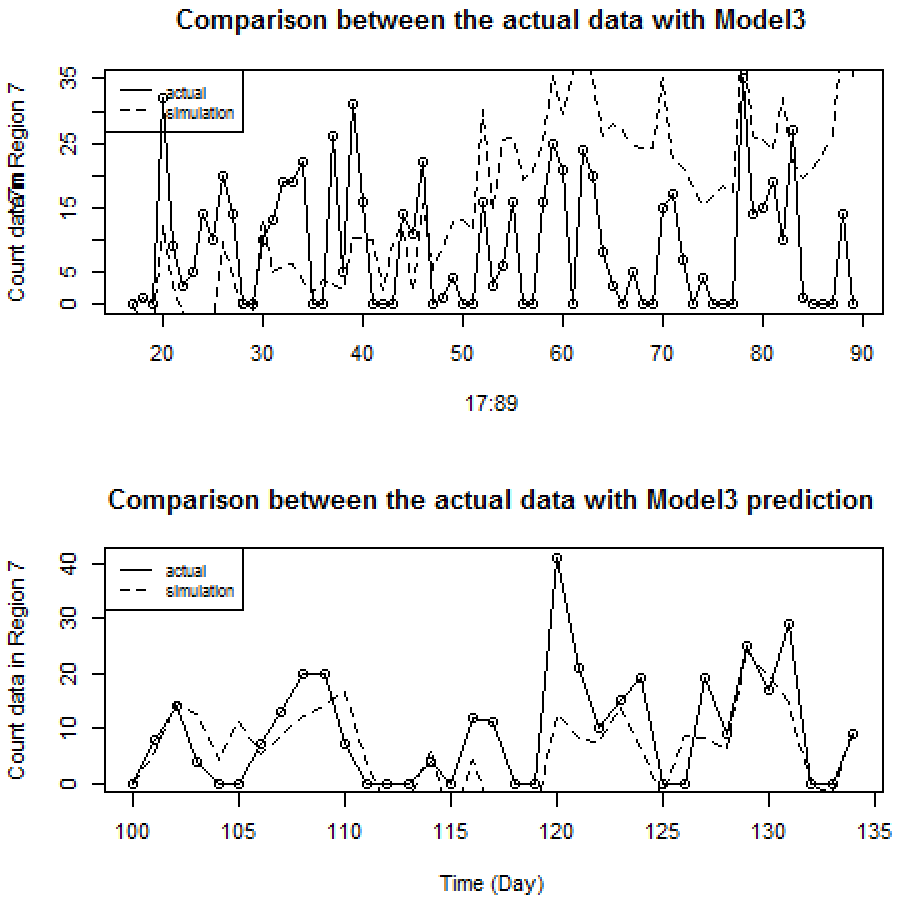


Figure 5.13: Comparison between actual data with simulation results from Model 3.

Figure 5.13 illustrate the fitting of model simulation and prediction to actual data. A random walk affect is gradually obvious and hence leads to a trend noticed from the comparison between actual data to Model 3 simulation. This is the reason to cause the positive displace of the histogram plot. However, when compare the prediction of Model 3 to actual data, the Model 3 is proved to be advantage. Also, it is notices that the starting 30 days of simulation of Model 3 is close to actual data. Therefore, we conclude Model 3 is specifically adequate for describing the severe braking events count data in shot time space. For large time space, Model 3 needs to be modified with certain offset.

5.3 Statistical Time Series Modeling Conclusion

In this chapter, time series modeling procedure for severe driving events count data in a region is investigated. ARIMA modeling for univariate time series modeling procedure and STARIMA modeling for multivariate time series modeling procedure are proposed respectively. The sever braking event count data in Region 7 denoted by Cluster 7 from 10/01/2013 to 02/25/2014 is utilized as an example to illustrate the implementation of the procedures. Time series from *Day 17* to *Day 89* is used for building adequate statistical models, and time series of *Day 101* to *Day 134* is used to validate the models from the procedures. Improvement of adequacy from ARIMA modeling to STARIMA modeling demonstrates a stronger correlation of spatial factor than daily temporal factor for this exemplified severe braking event count time series.

In next chapter, we will continue the time series modeling and give other two modeling procedures which are more focusing on the nature property of event occurrence and event count patterns.

Chapter 6

Generalized Linear Time Series Modeling and Nonparametric Time Series Modeling Expansion

In last chapter, the ARIMA and STARIMA methodologies are investigated for modeling the time series of the severe braking events count data in Region 7. The modeling procedures use only the self-information of the time series. They give adequate data description and supply embracive future data prediction. However, when using ARIMA or STARIMA methodology, the time series variable is always assumed to be normal distributed, which is conflict to the fact that the count data are relative to the Poisson distribution. The original count data is not appropriate for ARIMA and STARIMA modeling and, thus, we employ first difference operation processing the count data time series in ARIMA and STARIMA procedure. This difference operation for count data results in approximate normal distributed data, and the modeling procedures are essentially avoid discussing the natural property of the count data. In this chapter, we will continue to investigate the appropriate models for describing the count data time series and predicting future count data based on the Poisson distribution property. Since the spatial factor is proved to be a very influential factor in Section 5.2, the models in this chapter use the spatial factor as explanatory variable for examining the relations of the region of interest with its neighbors, and improve the count data description and prediction based on the spatial relations.

In Section 6.1, in the beginning, the Poisson distribution property is briefly reviewed. Then we formulate the generalized linear model with Poisson distribution for examining the Poisson distribution property of severe driving events count data in Region 7. In Section 6.2, nonparametric models are developed to provide better description of the severe braking events count stochastic process and to give better interpretation on the limit of the generalized Poisson model. The nonparametric models are inspired by Poisson Hidden Markov Modeling and by observed Poisson distribution patterns of the count data. The generalized linear model and the nonparametric model are developed with count data time series from *Day 17* to *Day 89*, and validated by the count data time series from *Day 101* to *Day 134*, respectively.

6.1 Generalized Linear Model with Poisson Distribution

Before we formulate the generalized linear model with Poisson distribution for the severe braking event count data time series, let us briefly review the Poisson distribution property in advance.

6.1.1 The Poisson Distribution [48]

A random variable Y is said to have a Poisson distribution with parameter μ if it takes integer values $y = 0,1,2, ...$ with probability

$$\Pr\{Y = y\} = \frac{e^{-\mu}\mu^{y}}{y!}, \tag{6.1}$$

for $\mu > 0$. The mean and variance of the this distribution can be shown to be

$$E(Y) = Var(Y) = \mu. \tag{6.2}$$

6.1.2 Generalized Linear Models Formulation

Generalized linear models extend linear models by allowing some non-linearity in the model structure; this extension is much more flexible in the specification of the distribution of the response variable Y [58]. Analogous to linear model structure,

$y_i = X_i\beta + \epsilon_i$, where $\epsilon_i \sim N(0, \sigma^2)$, the generalized linear model structure is presented as $\mathbb{E}(y_i) = X_i\beta$, where $\mathbb{E}(\cdot)$ is the smooth monotonic function. Cited from Nelder and Wedderburn [59], a generalized linear model (GLM) consists of three components:

1. A random component for the response named response variable, Y, which has a distribution following the exponential family.
2. A linear systematic component relating the linear predictor, $\eta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k$, to the product of explanatory variable X_i and the parameters β_i.
3. A smooth and invertible linearizing link function $g(\cdot)$, which transforms the expectation of the response variable, $\mu = E(Y)$, to the linear predictor:

$$g(\mu) = \eta = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_k X_k. \tag{6.3}$$

Generalized linear modeling (GLM) used in transportation field focuses mostly on traffic accidents data. In [60 - 63], the authors illustrated that generalized linear models are suitable models for determining relationships between accidents and characteristics of traffic and road geometry. In the works, the response variable is the count of accidents and the explanatory variables are chosen as the characteristic of traffic, e.g., traffic flow rate, and road geometry, such as, lane width, length of the segment, curve radius, etc.. Generalized linear models are formulated incorporating the traffic characteristics and road geometry in the form of linear predictor (Eq. (6.3)) for describing and predicting future accidents data. Goodness-of-fitting tests (GOF), such as, Pearson's $X^2$and the scaled deviance (G^2), are employed to address the adequacy of the GLMs. GOF tests use the properties of a hypothesized distribution to access whether or not observed data are generated from a given distribution [64].

Intuitively similar to accidents property, we use GLM for addressing the relationship between severe driving events with location. The response variable is the severe braking events count data in Region 7, denoted as $Y^{(7)}$ (we use $Y^{(7)}$ instead of $Z_t^{(7)}$ due to ignoring the temporal factor in GLM at this section). The explanatory variable is chosen to be the corresponding severe braking events count data in Region 5, denoted as $Y^{(5)}$. Thus, the linear predictor η will be linear combination of the count data in Region 5 (denoted as $Y^{(5)}$) and the design parameters β_k, which is $\eta = \beta_0 + \beta_1 Y^{(5)}$. For the counts data with Poisson distribution, the link function used in generalized linear modeling is always chosen as log function, which means $\eta = g(\mu) = \ln \mu$, where μ is the parameter of the Poisson distribution of the response variable $Y^{(7)}$,

$$\eta = \ln \mu = \beta_0 + \beta_1 Y^{(5)}. \tag{6.4}$$

In Figure 6.1, we reorganize the severe braking events count data time series in Region 5 in an order of increasing value and remaining the time order for the count data if the values of the daily counts are same. Corresponding to the reorganization, the severe braking events count data time series in Region 7 is also changed. From this plot we have an observation that the count data in Region 7 have a weak trend of increasing when the count data in Region 5 is strictly increase. This observation convinces the proposed Poisson model presented by Eq. 6.4.

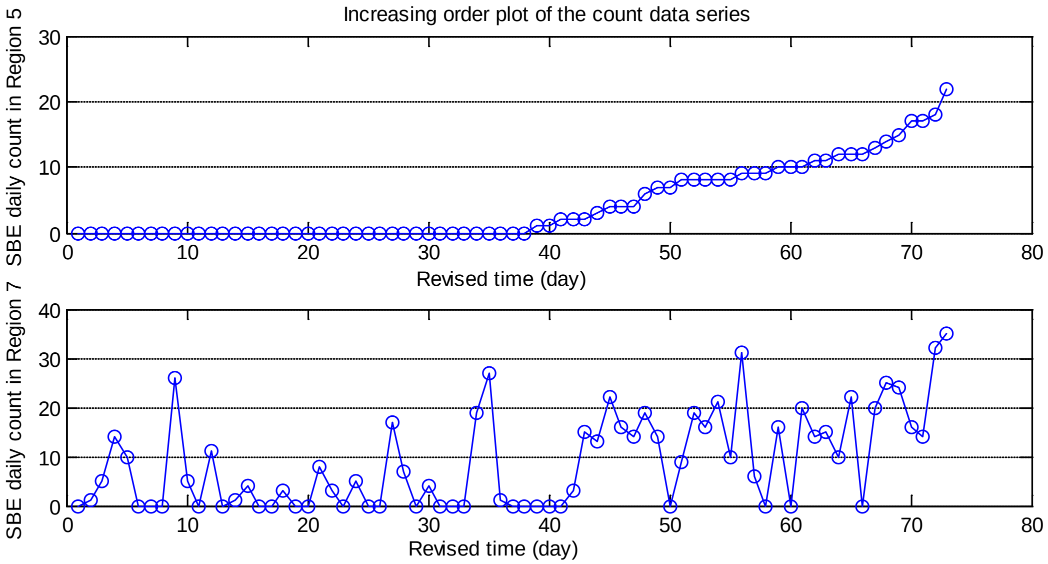


Figure 6.1: Reorganized severe braking events count series in Region 5 and Region 7 in an increasing order of count data in Region 5.

6.1.3 The Poisson Regression for Generalized Linear Poisson Model

Poisson distribution belongs to the exponential family of distributions. In [40], [41], the general form of the exponential family is provided with canonical parameter (θ). In the general form each of the y_i observation has been defined in terms of the parameters θ. The joint probability density function may be expressed as likelihood function of θ given the observations y_i.The parameter θ can be estimated by maximum likelihood function and then the coefficients of the generalized linear model is estimated. We describe this procedure in details below.

The Poisson probability density function (Eq. 6.1) can be rewritten in exponential-family form as

$$f_y(y;\mu) = \exp\{y\ln(\mu) - \mu - \ln\Gamma(y+1)\}. \tag{6.5}$$

The log-likelihood function can be abstracted from the exponential form as

$$\mathcal{L}(\mu;y) = \sum_{i=1}^{n}\{y_i\ln(\mu) - \mu - \ln\Gamma(y_i+1)\}. \tag{6.6}$$

The log-likelihood function can then be parameterized in terms of $X\beta$ through the log link function. Substituting the inverse link, $\exp(X\beta)$, for μ in the above formula,

$$\mathcal{L}(\mu; y) = \sum_{i=1}^{n}\{y_i \ln(\exp(X\beta)) - \exp(X\beta) - \ln\Gamma(y_i + 1)\}$$

$$= \sum_{i=1}^{n}\{y_i \exp(X\beta) - \ln\Gamma(y_i + 1)\}. \tag{6.7}$$

Measurement of model discrepancy (or goodness-of-fitting of model with data) in a form of logarithm of a ratio of likelihoods always is the scaled deviance (G^2). Suppose there are N observations we can fit models to the observations containing as many as N parameters. The simplest model, named the null model, has one parameter μ representing all observations [65]. At the other extreme the saturated model, has N parameters, dedicating one parameter to each observation and consequently match the data exactly [65]. The saturated model gives us a baseline for measuring the discrepancy for a proposed model. The residual deviance for a proposed GLM is

$$D_m = 2(\ln\mathcal{L}_s - \ln\mathcal{L}_m), \tag{6.8}$$

where $\mathcal{L}_m$ is the maximized likelihood of the proposed GLM and $\mathcal{L}_s$ is the maximized likelihood of a saturated model [65]. The scaled deviance (G^2) is simply the difference in the residual deviances for models. Suppose that *Model 0* ,with $k_0 + 1$ coefficients, is compared with *Model 1*, with $k_1 + 1$ coefficients. Suppose $k_0 < k_1$, in other words, *Model 0* would simply omit some of the repressors in *Model 1*. We test the null hypothesis that *Model 0* are corrected by computing the deviance

$$G_0^2 = D_0 - D_1. \tag{6.9}$$

Under the hypothesis, G_0^2 is asymptotically distributed as Chi-square with $k_1 - k_0$ degrees of freedom, $G_0^2 \sim \chi^2_{,k_1-k_0}$. The confidence interval for design parameters β_j for the hypothesis H_0: β_j is acceptable at p value can be calculated for

$$2(\ln\mathcal{L}_1 - \ln\mathcal{L}_0) \le \chi^2_{1-p,k_1-k_0}. \tag{6.10}$$

The Poisson model is formulated in Eq. (6.4) and the generalized linear model is solved by maximum likelihood estimation method. In Table 6.1, the Poisson model details are presented.

Table 6.1: Poisson model parameters estimation from MLE by R.

Poisson Model: glm(formular = y~x, family = poisson)

Deviance Residuals:

Min	1Q	Median	3Q	Max
-5.776	-3.2392	-0.5722	1.4667	6.7054

Coefficients:

	Estimate	Std. Error	z value	Pr $>|z|$
(Intercept) β_0	1.657494	0.05935	27.93	<2e-16
β_1	0.090573	0.005585	16.22	<2e-16

(Dispersion parameter for Poisson family taken to be 1)
Null deviance: 853.09 on 72 degrees of freedom
Residual deviance: 614.58 on 71 degrees of freedom
AIC: 816.98

Therefore, the Poisson model is estimated as

$$\ln \mu = 1.66 + 0.09Y^{(5)}. \tag{6.11}$$

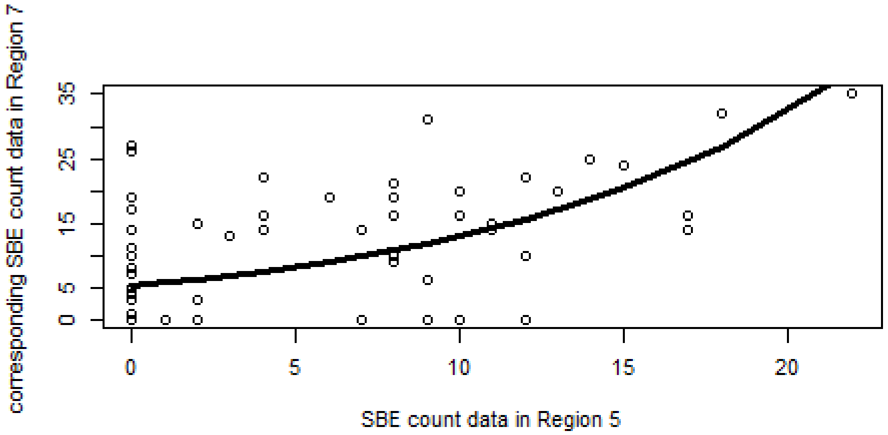


Figure 6.2: Poisson Model fitting vs actual data. (Explanatory variable is severe braking events count data in Region 5 and response variable is severe braking events count data in Region 7.)

The solved Poisson model fitting versus the actual data is presented in the Figure 6.2, whose explanatory variable is the event count data in Region 5 and the response variable is the corresponding daily count data in Region 7. The design parameters are statistically significant confirmed by the p-value ($\Pr > |z|$) which are quite near to zero. The residual deviance (deviance between saturate model with the Poisson model, $2(\ln \mathcal{L}_m - \ln \mathcal{L}_0)$) is very large. When solve p-value from Eq. 6.10, p-value is approximately zero showing that there is a significant lack of evidence to reject the Poisson model compared with the Null model.

We rewrite the Poisson model in term of time series form describing the severe braking events count data in Region 7 as

$$\hat{Z}_t^{(7)} = exp^{1.66+0.09Z_t^{(5)}}, \qquad (6.12)$$

where $\hat{Z}_t^{(7)}$ is the prediction of events count data time series in Region 7 from the observation of the events count data in Region 5. The errors are $e_t = Z_t^{(7)} - \hat{Z}_t^{(7)}$, and are convinced that they are white noise by the ACF and PACF plots in Figure 6.3. The histogram of the errors histogram is presented in Figure 6.4, which improves the fitting when compare with Model 3 on this example. However, there is a deviation from zero in the histogram. It is because that the exponential operation in Eq. (6.12) causes the minimum value is 5.26 when $Z_t^{(5)} = 0$. Therefore, we revise the Eq. (6.12) to be Eq. (6.13) to eliminate the deviation,

$$\hat{Z}_t^{(7)} = exp^{1.66+0.09Z_t^{(5)}} - 5.26. \qquad (6.13)$$

The simulation of the revised Poisson model from *Day 17* to *Day 89* and validation from *Day 100* to *Day 134* are presented in Figure 6.5, respectively.

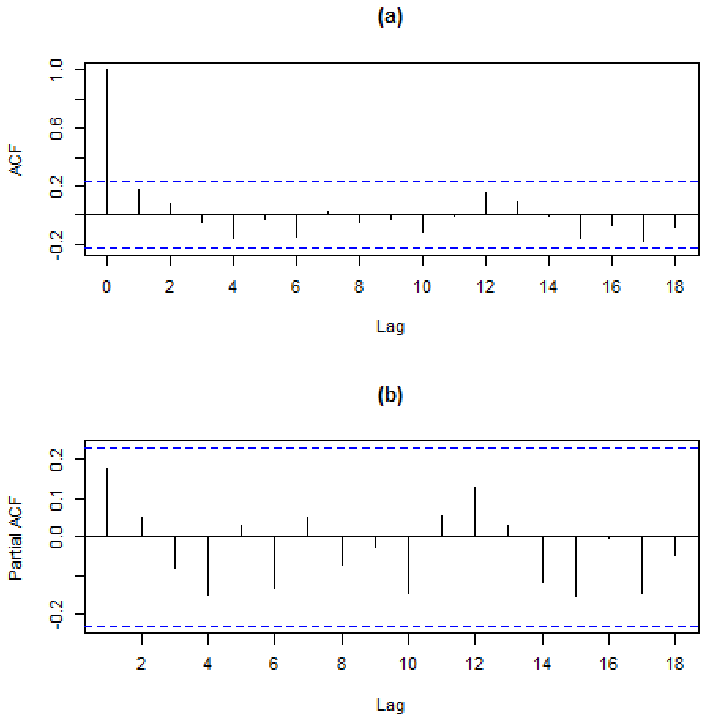


Figure 6.3: ACF and PACF of the errors between prediction from Poisson model and actual data.

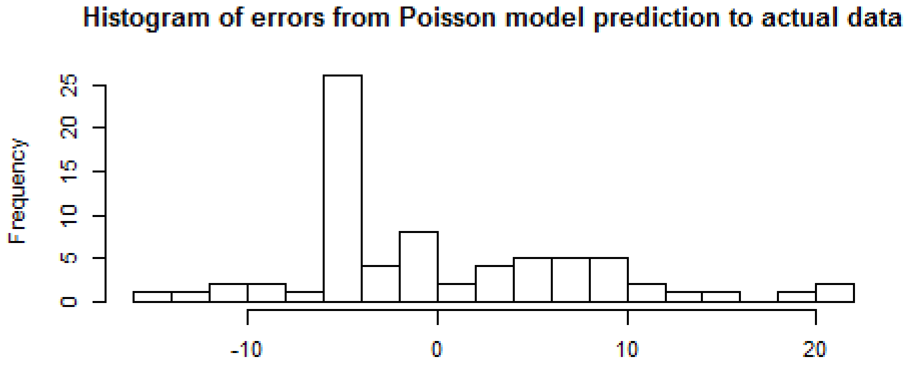


Figure 6.4: Residuals of Poisson model to actual severe braking events count series.

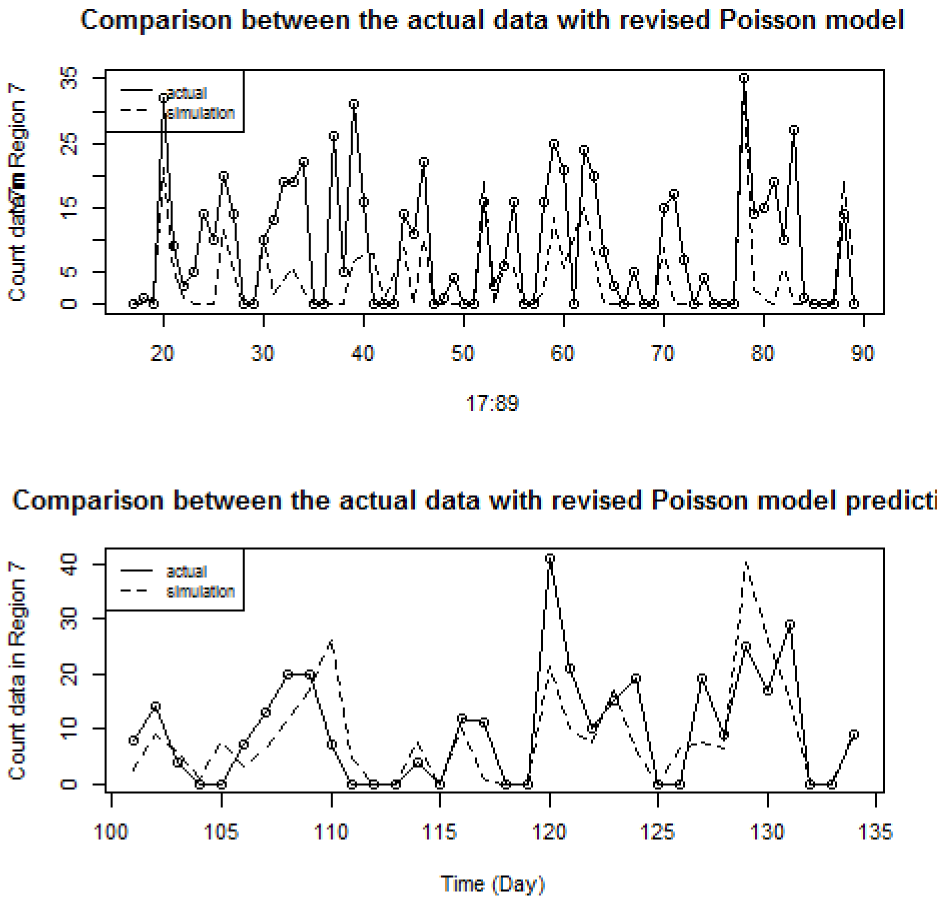


Figure 6.5: Comparison between actual data with simulation results from revised Poisson Model.

The revised Poisson model is proved to be adequately describing the actual count data and predict the future count data from Figure 6.5, respectively. However, there is a strong overdispersion phenomenon when use the Poisson model interpreting the actual data noticed in Figure 6.2. For example, the variance of $Y^{(7)}$ is quite different from the mean of $Y^{(7)}$, as its variance $\sigma_7^2 = 94.5$ and mean $\mu_7^2 = 9$, which is not consist with a single Poisson distribution property. In addition, the Poisson model cannot interpret the stochastic distribution of the response variable when $Y^{(5)} = 0$. Although the Poisson model is able to model the actual data adequately, it is unable to catch up details of the stochastic process. In next section, a nonparametric model, inspired by Poisson Hidden

Markov Modeling and by observed Poisson distribution patterns of the count data is proposed to solve the overdispersion phenomenon and the stochastic distribution phenomena.

6.2 Nonparametric Independent Mixture Poisson Model and Hidden Markov Model

One method of dealing with overdispersion is to use a mixture model of multimodal distribution. Mixture models are the models consisting unobserved groups, each having a distinct distribution for the observed variable [66]. Since the severe braking events count data, $Z_t^{(7)}$, is not strong autocorrelated proved by Model 2, we start modeling the count data supposing that each daily count data is generated from an independent mixture model. The independent mixture modeling is thus investigated in Section 6.2.1. The independent mixture model could very adequately describe the actual data; however, it is weak in perdition. As proved in the STARIMA model and Poisson model, the spatial factor is very influential in improving the prediction the future data; we use the count data series in Region 5 as hidden states which influence the count data in Region 7 to construct the final Hidden Markov Model (HMM) [67] in Section 6.2.2. The simulation and prediction results illustrate the adequacy of the model.

6.2.1 The Independent Mixture Poisson Model

The independent mixture model uses the information of the time series itself as well. We suppose that each count in Region 7, $Y^{(7)}$, is generated from an independent mixture model, which consists of a finite number, say m, of Poisson distributions with means λ_1, λ_2,…, λ_m. The probabilities assign to the different components is δ_1, δ_2,…, δ_m, respectively, and let p_1, p_2,…, p_m denote their density functions. To specify the component, we define a discrete random variable $\mathcal{C}$ which perform the mixing:

$$C = \begin{cases} 1 & with\, probability\; \delta_1 \\ 2 & with\, probability\; \delta_2 \\ \ldots & \ldots \\ m & with\, probability\; 1 - \sum_{i=1}^{m-1} \delta_i \end{cases}.$$

It is easy to show that the probability of $Y^{(7)}$ is given by

$$\Pr\left(Y^{(7)} = y\right) = \sum_{i=1}^{m} \Pr\left(Y^{(7)} = y \middle| C = i\right) \Pr(C = i) = \sum_{i=1}^{m} \delta_i p_i(y). \qquad (6.14)$$

The structure is represented by the graph in Figure 6.6.

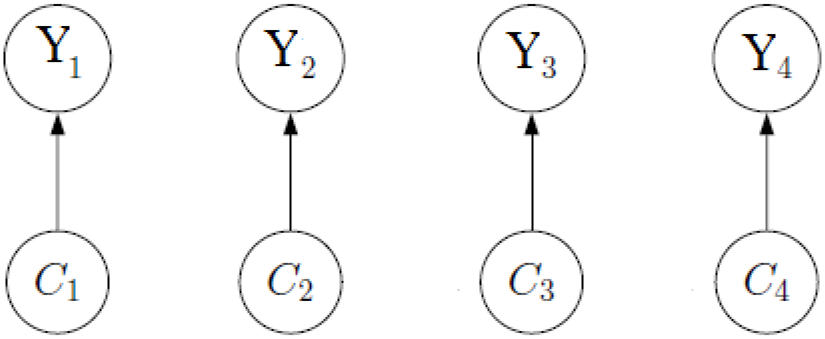


Figure 6.6: Independent mixture model structure.

Analogously, the estimation of the parameters of a mixture distribution is often identified by maximum likelihood (ML). The likelihood of the mixture model with m components is given as

$$L(\lambda_1, \ldots \lambda_m, \delta_1, \ldots, \delta_m | y_1, \ldots, y_N) = \prod_{j=1}^{N} \sum_{i=1}^{m} \delta_i p_i\left(y_j, \lambda_i\right). \qquad (6.15)$$

Then maximum likelihood can be equivalent and formulated as

$$Min\!: J = -\log(L) \text{ and} \qquad (6.16\ (a))$$

$$s.b.\ \sum_{i=1}^{m} \delta_i = 1. \qquad (6.16\ (b))$$

By minimization the likelihood function, we can identify the components of the mixture model fitted to the count data in Region 7 shown as in Table 6.2.

Table 6.2: Poisson independent mixture models fitted to the count data in Region 7.

model	i	δ_i	λ_i	$-\log L$	Mean	variance
m = 2	1	0.5	0.9	245.9616	9.1	76.4
	2	0.5	17.4			
m = 3	1	0.39	0.1	214.6215	9.1	85.5
	2	0.21	6.1			
	3	0.40	19.4			
m = 4	1	0.39	0.1	208.2795	9.1	93.2
	2	0.16	4.9			
	3	0.33	15.6			
	4	0.11	26.8			
Observations					**9.1**	**94.5**

From Table 6.2, the independent mixture model with 4 components is more appropriate describing count data series in Region 7 than other mixture models, as the mean and variance is close to the observations. Thus, we have the following model formulation

Independent Mixture Poisson Model

$$\Pr\left(Y^{(7)} = y\right) = \sum_{i=1}^{4} \Pr\left(Y^{(7)} = y \middle| C = i\right) \Pr(C = i) = \sum_{i=1}^{4} \delta_i p_i(y), \qquad (6.17\text{ (a)})$$

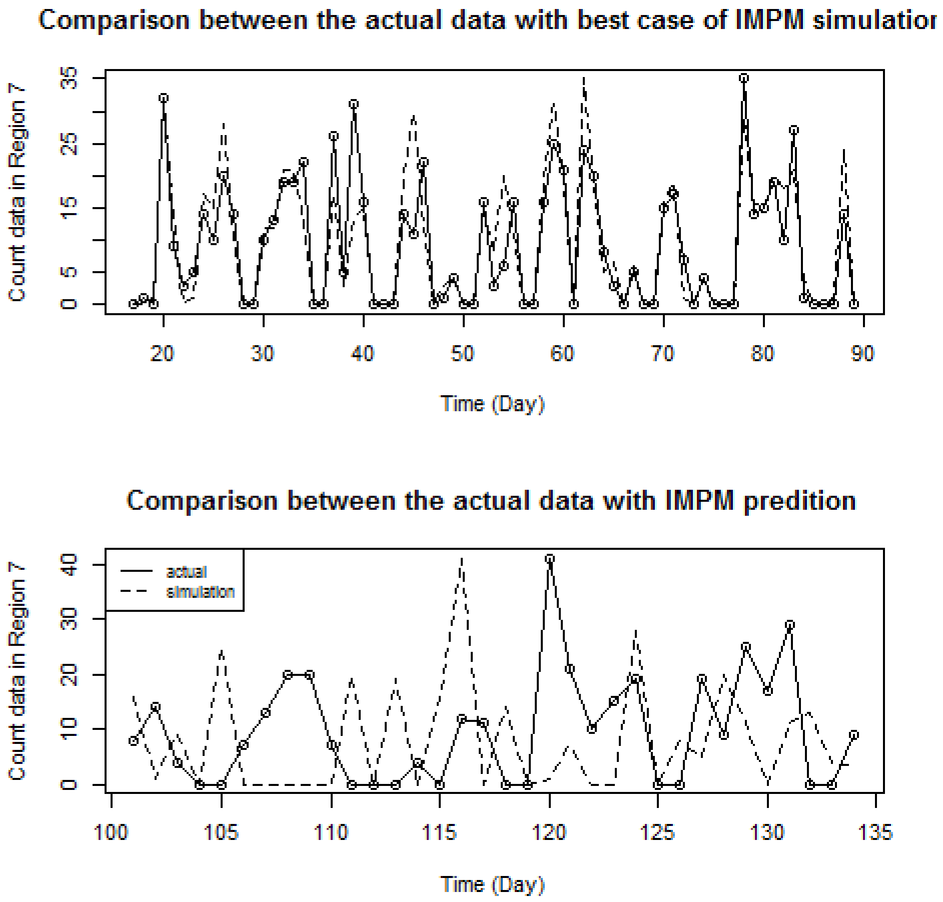
where, $p_1(y) = \frac{e^{-0.1}0.1^y}{y!}, p_2(y) = \frac{e^{-4.9}4.9^y}{y!}, p_3(y) = \frac{e^{-15.6}15.6^y}{y!}, p_4(y) = \frac{e^{-26.8}26.8^y}{y!}$, and

(6.17 (b))

$$\delta_1 = 0.39, \delta_2 = 0.16, \delta_3 = 0.33, \delta_4 = 0.31. \qquad (6.17\text{ (c)})$$

This independent mixture Poisson model is very precise in describing the severe braking events daily count stochastic process. The time series of $Z_t^{(7)}$ is one experiment from the independent mixture Poisson stochastic process. Ideally, we can simulate the independent mixture Poisson stochastic process using Monte Carlo method to obtain infinite possible sets of count series [68]. With a specified condition, the independent mixture Poisson model can generate a set of count series exactly close the time series of $Z_t^{(7)}$from *Day 17* to *Day 89*. However, the independent mixture model is hard to use for prediction, as a Poisson component is uniformly generated based on its probability. For example, Figure

6.7 shows best simulation from independent mixture Poisson model and one example set of count series prediction from the independent mixture Poisson model.


Comparison between the actual data with best case of IMPM simulatior

Comparison between the actual data with IMPM predition

Figure 6.7: Comparison between actual data with simulation results from Independent Mixture Poisson Model (IMPM).

6.2.2 The Hidden Markov Model (HMM) Discussion

The severe braking event count time series $Y^{(7)}$ is not strong autocorrelation, in other words, the count time series is not serial dependence, therefore, the independent mixture Poisson model (Eq. (6.17)) is enough to describe the count series. However, as mentioned at the end of Section 6.2.1, we cannot employ the independent mixture Poisson model for perdition, because perdition requires data serial dependence. Once we can, using a certain rule to reveal a hidden relation connecting the series to be serially dependent, we are able to predict the future count value through modeling the dependence rule.

Assuming the parameter process of the count series is serially dependent, a simple and mathematically convenient way to model this dependence is Markov chain. The resulting model for the observations is called Hidden Markov Model.

A Hidden Markov model (HMM) is a mixture model in which the distribution that generates an observation depends on the state of an underlying and unobserved Markov process [66], [69]. It can accommodate both overdispersion and serial dependence, which overcomes the drawbacks of general linear models with Poisson distribution and independent mixture Poisson model.

Definition 11:

A Hidden Markov Model $\{Z_t: t \in \mathbb{N}\}$ is a particular kind of dependent mixture model, with Z_t is dependent on the state C_t. The state transition is presented as Eq. (6.18), the model output Z_t is determined dependent on the current state C_t, presented by Eq. (6.19),

$$\Pr(C_t) = \Pr(C_t|C_{t-1}),\ t = 1,2, \ldots \text{ and} \tag{6.18}$$

$$\Pr(Z_t) = \Pr(Z_t|C_t),\ t \in \mathbb{N}. \tag{6.19}$$

The model consists of two parts: firstly, an unobserved state process $\{C_t: t = 1,2, \ldots\}$ satisfying the Markov property, and secondly the ‘state-dependent process’ $\{Z_t: t = 1,2, \ldots\}$ such that, when C_t is known, the distribution of Z_t depends only on the current state C_t and not on previous states or observations [66]. This structure is represented by the diagram in Figure 6.8.

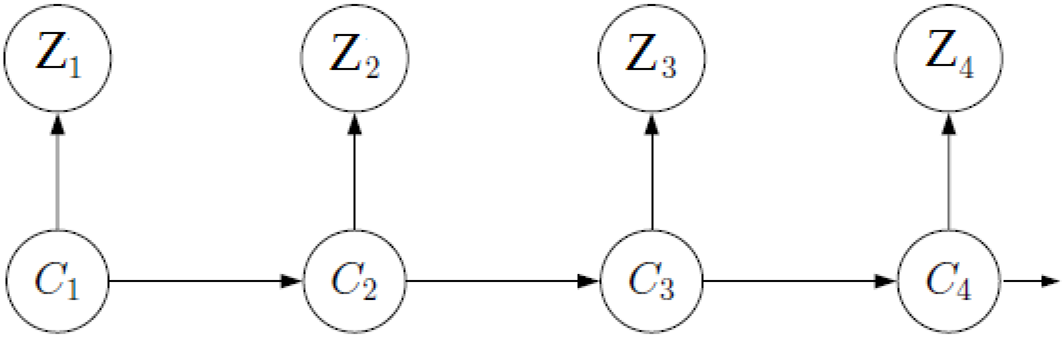


Figure 6.8: Basic Hidden Markov Model diagram.

However, when apply HMM method for the severe braking event count series in Region 7, we have to think over two questions: i) what is the unobserved state for the count series in Region 7? ii) How could we find a rule to convert the count series in Region 7 to be serially dependent?

Let's start with the first question: what is the unobserved state for the count series in Region 7. In STARIMA model and general Poisson model reveal a strong correlation between the events count series in Region 5 and it in Region 7, thus, let's start with an assumption that the unobserved state is the independent Poisson component of the event count series in Region 5. Noticed that the mean of $Y^{(5)}$ is 4.3 and the variance of $Y^{(5)}$ is 32.9, therefore, an independent mixture model is formulated to describe the count series of $Y^{(5)}$, and the Poisson components of $Y^{(5)}$ are assumed to be unobserved states in the Hidden Markov Model.

First Remark: The Poisson components of $Y^{(5)}$ are assumed to be unobserved states in a Hidden Markov Model for describing the count series of $Y^{(7)}$.

Similarly, we suppose that each count data of $Y^{(5)}$ is generated from an independent mixture model, which consists of m Poisson distributions with means λ_1, λ_2,…, λ_m. The probabilities assign to the different components are δ_1, δ_2,…, δ_m, respectively, and let p_1,

$p_2, \dots, p_m$ denote their density functions. Then we have

$$C = \begin{cases} 1 & \mathit{with\ probability}\ \delta_1 \\ 2 & \mathit{with\ probability}\ \delta_2 \\ \cdots & \cdots \\ m & \mathit{with\ probability}\ 1 - \sum_{i=1}^{m-1} \delta_i \end{cases}$$

and

$$\Pr\left(Y^{(5)} = y\right) = \sum_{i=1}^{m} \Pr\left(Y^{(5)} = y \middle| C = i\right) \Pr(C = i) = \sum_{i=1}^{m} \delta_i p_i(y). \tag{6.20}$$

Analogously, the likelihood function and maximum likelihood problem of the mixture model with m components formulated as Eqs. (6.15 – 6.16).

Solving the minimization likelihood problem, we can identify the components of the mixture model fitted to the series of $Y^{(5)}$ is presented in Table 6.3.

Table 6.3: Poisson independent mixture models fitted to the counts data in Region 5.

model	i	δ_i	λ_i	$-\log L$	**Mean**	**variance**
m = 2	1	0.52	0	171.8659	4.3	24.4
	2	0.48	8.97			
m = 3	1	0.51	0	154.5403	4.3	30.9
	2	0.13	2.4			
	3	0.36	11.1			
m = 4	1	0.51	0	152.8461	4.2	32.8
	2	0.12	2.1			
	3	0.30	9.5			
	4	0.07	16.3			
Observations					**4.3**	**32.9**

It is obvious that an independent mixture model with 4 components is more appropriate describing series of $Y^{(5)}$ than other mixture models. The independent mixture Poisson model is presented as follows,

$$\Pr\left(Y^{(5)} = y\right) = \sum_{i=1}^{4} \Pr\left(Y^{(5)} = y \middle| C = i\right) \Pr(C = i) = \sum_{i=1}^{4} \delta_i p_i(y), \tag{6.21 (a)}$$

where

$$p_1(y) = \begin{cases} 1, & if\ y = 0; \\ 0, otherwise; \end{cases}, p_2(y) = \frac{e^{-2.1}2.1^y}{y!}, p_3(y) = \frac{e^{-9.5}9.5^y}{y!}, p_4(y) = \frac{e^{-16.3}16.3^y}{y!}, \text{ and}$$

(6.21 (b))

$$\delta_1 = 0.51, \delta_2 = 0.12, \delta_3 = 0.30, \delta_4 = 0.07. \qquad (6.21\ (c))$$

We denote the four Poisson components as $C_i, i = 1,2,3, or\ 4$. According to the First Assumption, the four Poisson components of $Y^{(5)}$ are the unobserved states.

Based on the four Poisson components of $Y^{(5)}$, we can basically partition the count series $Y^{(5)}$ and $Y^{(7)}$ into four parts corresponding to the four components of the independent mixture model shown as Figure 6.8. From the perspectives of waving frequency, developing trend, and data magnitude of the count series $Y^{(7)}$, it is recognized four patterns consistent to the four different parts: i) the first pattern consistent to Part I, named as Patter I, performs to be randomized from three major Poisson components, one components with a small Poisson parameter (between 0 and 10), another one with a medium Poisson parameter (between 10 and 20) and the last one with a large Poisson parameter (larger than 20); ii) the second pattern consistent to Part II, named Pattern II, and the third pattern consistent to Prat III, named Pattern III, are similar, because they both have a “broad band” and the Poisson distribution parameter is medium, from 10 to 20. The last pattern consistent to Part IV, named Pattern IV, is strongly associated with the tread of the count series in Region 5; when the counts of $Y^{(5)}$ is increasing, the tread of the counts of $Y^{(7)}$ is also increasing. The four patterns are recognized based on the observation on the series waving frequency, developing trend, and data magnitude. Essentially, the series waving frequency is dependent on the possibilities of different Poisson components of the certain series. Relatively close possibilities of different Poisson components result in zigzag shape, for example, Pattern I; relative very different possibilities to Poisson components result in “board band”, for example, Pattern II and Pattern III. Series developing trend are dependent on the positive correlation of $Y^{(5)}$ and $Y^{(7)}$, for example, Pattern IV. Series data magnitude is dependent on the values of different Poisson components.

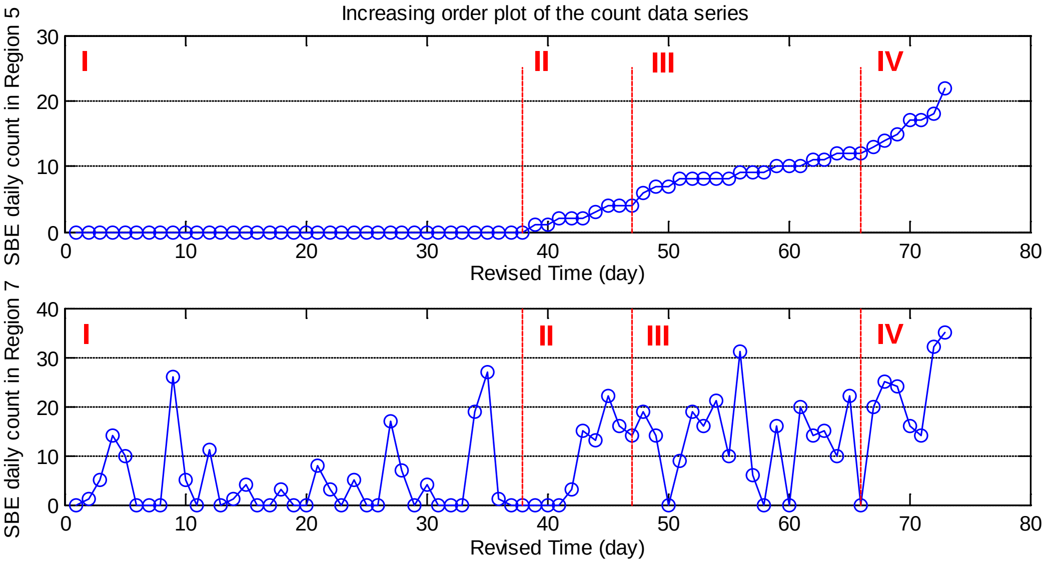


Figure 6.9: 4 partitions of $Y^{(5)}$ and $Y^{(7)}$ according to the 4 Poisson components of independent mixture model of $Y^{(5)}$.

Therefore, in the following, we use independent mixture models describing each part the count series of $Y^{(7)}$ to validate the proposed patterns and attempt to create connections between the Poisson components of $Y^{(5)}$ and $Y^{(7)}$.

The independent mixture Poisson model for Part I of $Y^{(7)}$ is

$$\Pr\left(Y^{(7)} = y\right) = \sum_{i=1}^{3} \Pr\left(Y^{(7)} = y \middle| C = i\right) \Pr(C = i) = \sum_{i=1}^{3} \delta_i p_i(y), \qquad (6.22\text{ (a)})$$

where

$$p_1(y) = \begin{cases} 1, & if\ y = 0; \\ 0, & otherwise; \end{cases}, p_2(y) = \frac{e^{-4.3} 4.3^y}{y!}, p_3(y) = \frac{e^{-18.6} 18.6^y}{y!} \text{ and} \qquad (6.22\text{ (b)})$$

$$\delta_1 = 0.50, \delta_2 = 0.34, \delta_3 = 0.16. \qquad (6.22\text{ (c)})$$

The independent mixture Poisson model for Part II of $Y^{(7)}$ is

$$\Pr\left(Y^{(7)} = y\right) = \sum_{i=1}^{2} \Pr\left(Y^{(7)} = y \middle| C = i\right) \Pr(C = i) = \sum_{i=1}^{2} \delta_i p_i(y), \qquad (6.23\text{ (a)})$$

where

$$p_1(y) = \frac{e^{-0.7} 0.7^y}{y!}, p_2(y) = \frac{e^{-16} 16^y}{y!} \text{ and} \qquad (6.23\text{ (b)})$$

$$\delta_1 = 0.44, \delta_2 = 0.56. \qquad (6.23\text{ (c)})$$

The independent mixture Poisson model for Part III of $Y^{(7)}$ is

$$\Pr\left(Y^{(7)} = y\right) = \sum_{i=1}^{2} \Pr\left(Y^{(7)} = y \middle| C = i\right) \Pr(C = i) = \sum_{i=1}^{2} \delta_i p_i(y), \qquad (6.24\text{ (a)})$$

where

$$p_1(y) = \begin{cases} 1, & if\ y = 0; \\ 0, otherwise; \end{cases}, p_2(y) = \frac{e^{-16} 16^y}{y!} \text{ and} \qquad (6.24\text{ (b)})$$

$$\delta_1 = 0.21, \delta_2 = 0.79. \qquad (6.24\text{ (c)})$$

The independent mixture Poisson model for Part IV of $Y^{(7)}$ is

$$\Pr\left(Y^{(7)} = y\right) = \sum_{i=1}^{2} \Pr\left(Y^{(7)} = y \middle| C = i\right) \Pr(C = i) = \sum_{i=1}^{2} \delta_i p_i(y), \qquad (6.25\text{ (a)})$$

where

$$p_1(y) = \frac{e^{-018.4} 18.4^y}{y!}, p_2(y) = \frac{e^{-29.6} 29.6^y}{y!} \text{ and} \qquad (6.25\text{ (b)})$$

$$\delta_1 = 0.53, \delta_2 = 0.47. \qquad (6.25\text{ (c)})$$

The independent mixture Poisson model for each part of $Y^{(7)}$ is dependent on the Poisson component of $Y^{(7)}$. Thus, we could present a compound mixture Poisson model incorporating the independent mixture Poisson models (Eq. (6.21)) of $Y^{(5)}$ together with the independent mixture Poisson models (Eqs. (6.22 – 6.25)) of each part of $Y^{(7)}$. The compound mixture Poisson model is illustrated as Figure 6.10. In the figure, we use term C_i^j to denote the j-th Poisson component of i-th Pattern of $Y^{(7)}$.

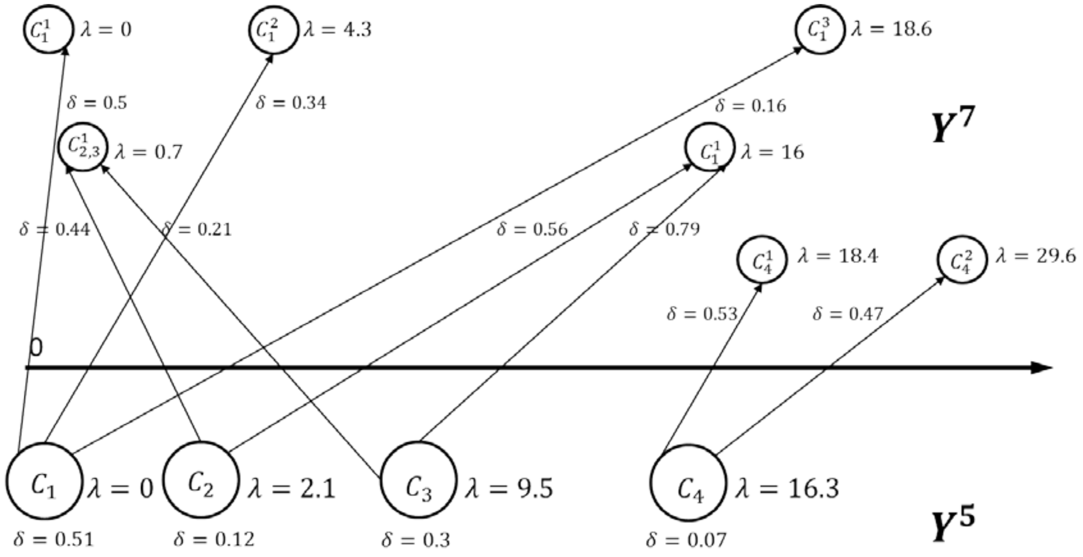


Figure 6.10: Relation of the 4 count series patterns of $Y^{(7)}$ with the 4 Poisson components of $Y^{(5)}$.

Essentially, the compound mixture model is the second part of Hidden Markov Model presented by Eq. (6.19). With the Initial Remark, we can simulate the count series in Region 7 when we have the count in Region 5, i.e., we could predict the count data in Region 7 when we obtain the count data in Region 5. The compound mixture model not only perfectly interprets the count series in Region 7 and but also it connects the relation with the count series in Region 5.

Back to the second question, how could we develop rule to reveal the relation to connect that the series to be serially dependent? There may be possible answers such as bring in a new relative factor to severe braking events, for example, a variable measuring visibility or vehicle speed. But in this dissertation, we suppose the count series of $Y^{(7)}$ to be serially dependent according to the four patterns (as Figure 6.9). Therefore, once we have count data in Region 5, we could predict the count data in Region 7 according to the four patterns. However, since the perquisite of HMM is the data series is autocorrelated, we have to confirm that the four patterns' series are autocorrelated. Figure 6.11 presents autocorrlation test for the four partners' count series.

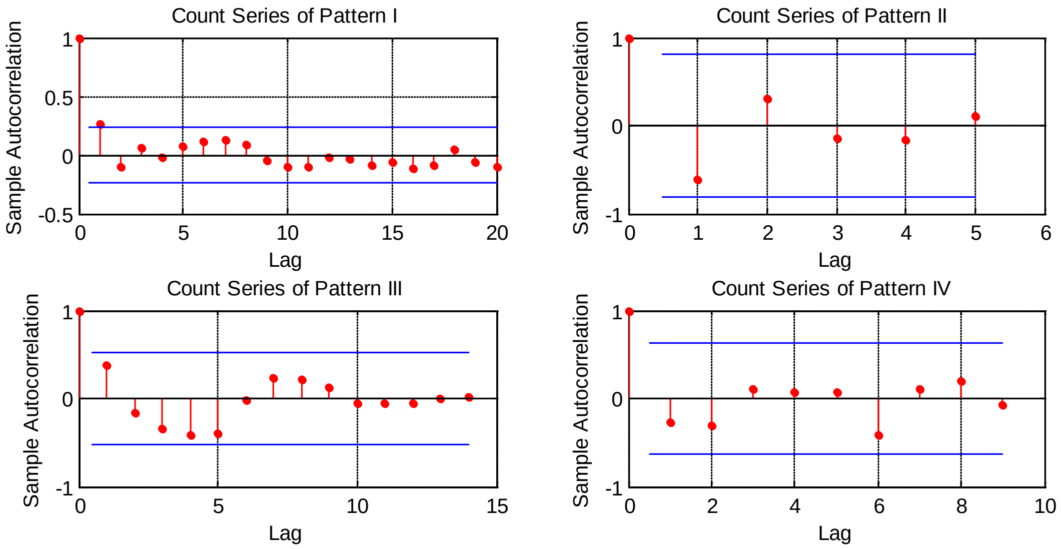


Figure 6.11: Autocorrelation plots of the four pattern series of $Y^{(7)}$.

We have to admit that except count series of Pattern I, the other three count series do not show strong autocorrelation. As we known, the cut value of the ACF is usually calculated as calculated as $\pm 1.96/\sqrt{N}$, where N is number of samples and 1.96 comes from 95% confidence interval of accepting the series is white noise if the ACF is within the cuts value. In our case, the number of count data in Patter II, III, and IV is very few causing the cut value comparably great. So let lower the confidence level to reduce the cut values are same with Pattern I, then obviously, thee autocorrelation plots support the four pattern series autocorrelated.

Second Remark: the count series of Pattern II, III, and IV of $Y^{(7)}$ is autocorrelated with a reduced confidence level.

Since the count series of the four patterns of $Y^{(7)}$ is autocorrelated, the assumption that count series of $Y^{(7)}$ is serially dependent according to the four patterns is valid. The serially dependence can be presented by Markov chain.

Markov Chain: A sequence of discrete random variable $\{C_t: t \in \mathbb{N}\}$ is said to be a (discrete-time) Markov chain (MC) if for all $t \in \mathbb{N}$ it satisfies the Markov property

$$\Pr(C_{t+1}|C_t, \dots, C_1) = \Pr(C_{t+1}|C_t). \tag{6.26}$$

The conditional probabilities for the transition between the states $C_t = i$ and $C_{s+t} = j$ associated with a Markov chain are called transition probabilities:

$$\gamma_{ij}(t) = \Pr(C_{s+t} = j|C_t = i). \tag{6.27}$$

If the states are finite, the matrix $\Gamma(t)$ is defined as the matrix with (i, j) element $\gamma_{ij}(t)$.

The count series of a pattern of $Y^{(7)}$ can be presented by Markov Chain due to the series' autocorrelation property. However, the states for the count series of a pattern of $Y^{(7)}$ is not the independent Poisson components of $Y^{(5)}$ as mentioned in First Remark, because the sequence of count data of a pattern of $Y^{(7)}$ corresponds to only one independent Poisson components of $Y^{(5)}$, C_i. Actually, each pattern series of $Y^{(7)}$ can be formulated by the sequence of the pattern's independent Poisson components, C_i^j, and a pattern series is composed by a sequence of every two neighbor independent Poisson components. In other word, the state for the count series of $Y^{(7)}$ is the transition between two neighbor independent Poisson components of the pattern series.

Hence, we have to revise the First Remark for the states in HMM.

Revised First Assumption: the unobserved states in HMM is not the Poisson components of $Y^{(5)}$, but is the transition between two neighbor independent Poisson components of the pattern series.

Therefore, in the following, we develop the Markov chain for the each pattern series transition to model the count series dependence. Firstly, let notify the state which denote the transition between two neighbor independent Poisson components of the pattern series.

For Pattern I series, when the independent Poisson components of $Y^{(5)}$ is C_1, we have a MC which has three states, ‘1’, ‘2’, and ‘3’,

$$\begin{cases} 1 \equiv C_1 \to C_1^1 \\ 2 \equiv C_1 \to C_1^2. \\ 3 \equiv C_1 \to C_1^3 \end{cases} \tag{6.28}$$

For Pattern II series, when the independent Poisson components of $Y^{(5)}$ is C_2, we have a MC which has two states, ‘4’ and ‘5’,

$$\begin{cases} 4 \equiv C_2 \to C_2^1 \\ 5 \equiv C_2 \to C_2^1 \end{cases}. \tag{6.29}$$

For Pattern III series, when the independent Poisson components of $Y^{(5)}$ is C_3, we have a MC which has two states, ‘6’ and ‘7’,

$$\begin{cases} 6 \equiv C_3 \to C_3^1 \\ 7 \equiv C_3 \to C_3^1 \end{cases}. \tag{6.30}$$

For Pattern IV, when the independent Poisson components of $Y^{(5)}$ is C_4, we have a MC which has two states, ‘8’ and ‘9’,

$$\begin{cases} 8 \equiv C_4 \to C_4^1 \\ 9 \equiv C_4 \to C_4^2 \end{cases}. \tag{6.31}$$

Therefore, the Markov Chains structure is presented as Figure 6.12.

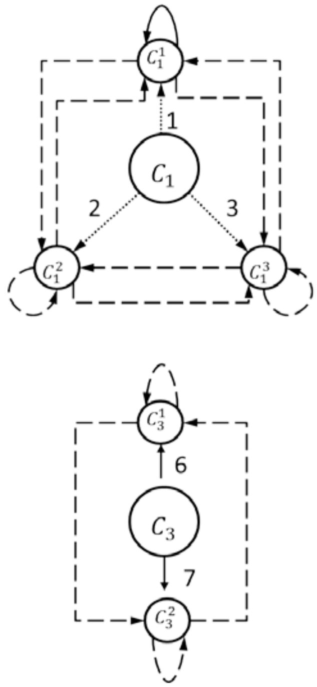


Figure 6.12: Markov chain structure for the four pattern of $Y^{(7)}$.

With these Markov chains structure, the Hidden Markov Model structure is constructed as Figure 6.13.

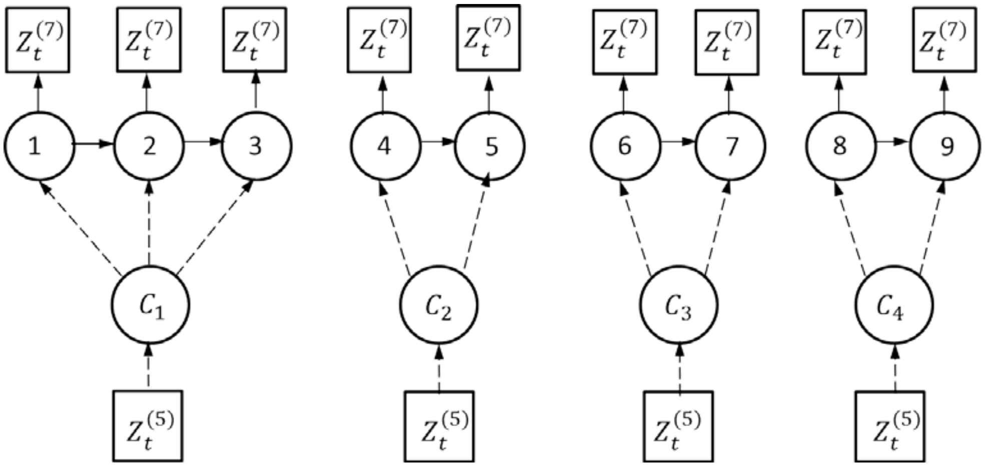


Figure 6.13: Hidden Markov Model structure for estimating the count time series of $Z_t^{(7)}$.

After we have the states for each pattern series, we could simply estimate the transition probability matrix as follows. The observed Markov chain state sequence of Pattern I from Figure 6.9 is

$$\text{“11233 11132 13122 11211 22121 13212 11133 211”},$$

then, the matrix of transition counts is developed as

$$(f_{ij}) = \begin{pmatrix} 9 & 6 & 4 \\ 8 & 2 & 1 \\ 2 & 3 & 2 \end{pmatrix},$$

where f_{ij} denotes the number of transitions observed from the state i to state j. Since the number of transitions from state 1 to state 2 is 6, and the total number of transitions from state 1 is $8 + 6 + 4 = 18$, we could deduce a transition probability matrix, denoted as Γ, and the transition probability matrix for Pattern I series is estimated as

$$\Gamma_1 = \begin{pmatrix} 9/19 & 6/19 & 4/19 \\ 8/11 & 2/11 & 1/11 \\ 2/7 & 3/7 & 2/7 \end{pmatrix}.$$

Similarly, the transition probability matrix for count series of Pattern II, Pattern III and Pattern IV is

$$\Gamma_2 = \begin{pmatrix} 1/3 & 2/3 \\ 2/5 & 3/5 \end{pmatrix}, \Gamma_3 = \begin{pmatrix} 1/3 & 2/3 \\ 3/15 & 12/15 \end{pmatrix}, \Gamma_4 = \begin{pmatrix} 1/2 & 1/2 \\ 2/4 & 2/4 \end{pmatrix}, \text{ respectively.}$$

Applying the transition probability matrices for each pattern with the observation on the count series in Region 5, we can simulate and predict the severe braking events count time series in Region 7, $Z_t^{(7)}$. The comparisons of the results of HMM with actual data are presented in Figure 6.14 and Figure 6.15.

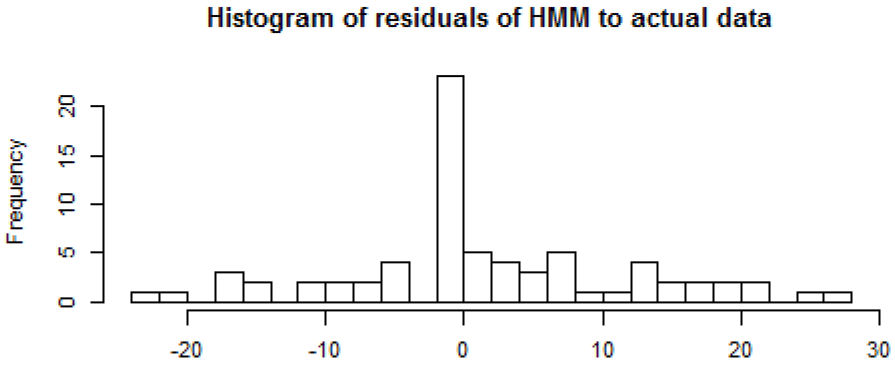


Figure 6.14: Residuals of Hidden Markov Model to actual count data of $Z_t^{(7)}$.

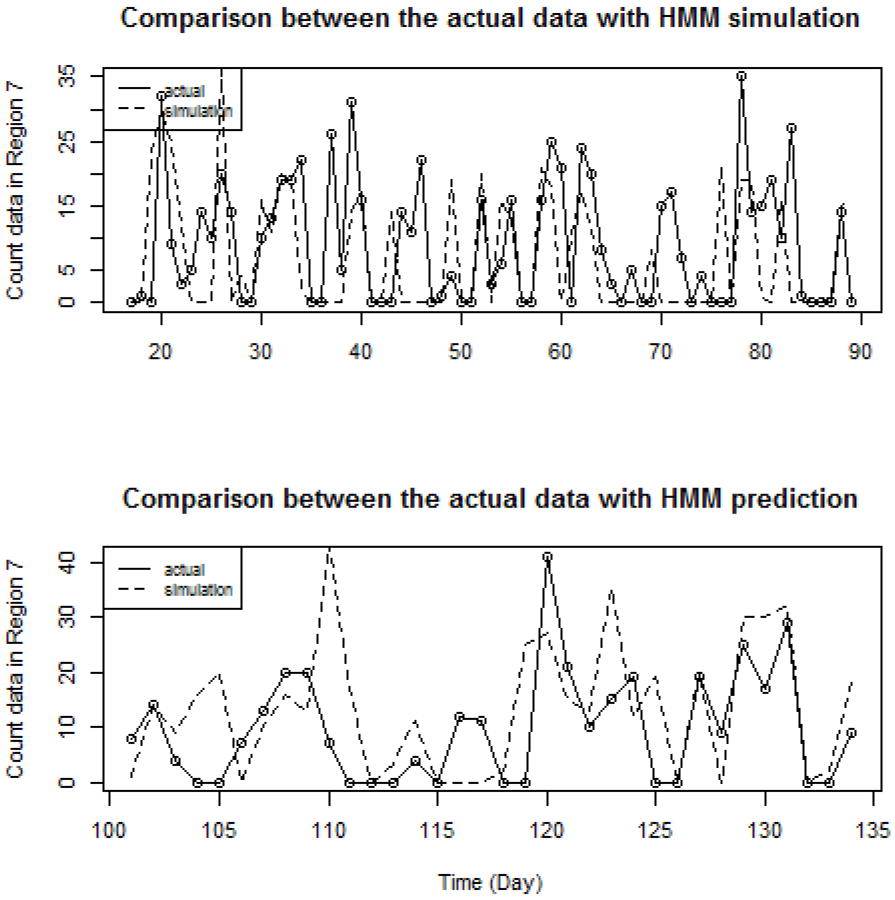


Figure 6.15: Comparison between actual data with simulation results from HMM.

The prediction of HMM is much improved compared with the prediction of IMPM.

However, the simulation from Day 17 to Day 89 is not as good as best case of the simulation of IMPM. The simulation and prediction can further be improved by use more pattern state transition or use a developing transition probability matrix or real-time updating transition probability matrix by observer [69 – 70].

Chapter 7

Conclusion and Further Work

This dissertation presents the correlation identification of severe driving events with time and location. Clustering methods extract the cluster-based correlation. Based on the cluster-based correlation, we further model the temporal and spatial correlation variation and development in term of modeling the time series of the daily counts of the events occurrence in regions. The two functions consists the thought of constructing safety driving assistance system aiming to help drivers recognize the severe driving events, predict the risk of driving, and reduce the driving risk caused by these events. Severe braking events are particularly used as example to illustrate the two system functions: i) identify the events cluster-based correlation with spatial and temporal factors and monitor the variation and development of the cluster-based correlation, ii) measure and predict the correlation variation and development through different proposed mathematical models. Four different mathematical models are proposed and verified with example severe braking events count series data in a certain region. Each model regards to modeling different aspect information of the count series data. ARIMA model is specifically proposed to examine the temporal correlation; STARIMA model, on the basis of ARIMA model, incorporates the spatial correlation to do both sides examination. General linear model (GLM) with Poisson distribution focuses on the spatial correlation of different regions in the point view of the events occurrence property. Hidden Markov model (HMM) is tempted to describe and predict the event count data in a deep reasoning with Poisson components transition between different regions. The four models are all validated by actual data of Region 7 and Region 5 and demonstrated their adequacy. The modeling procedures are developed schematically to be applied in the safety driving assistance system, hence, used for other interested regions to examine the severe driving events occurrence phenomena and employed for other severe driving events, for example, Handling limit minder (HLM) events. In realization of the safety driving assistance system, some places needs to consider for the use in real vehicles to assure a safer and more reliable driving assistance.

7.1 Safety Driving Assistance System Discussion

The core part of safety driving assistance system is the intelligent analyzer, which used clustering algorithms to identify the cluster-based correlation of severe driving events with time and location. The intelligent analyzer architecture shown as in Figure 1.1, replies on the technology of wireless communication and cloud computing. Wireless communication technology supports the data flow loop: acquire vehicle data (e.g. driving events data) or relative external data (e.g., GPS data, weather data), feedback useful data for sharing useful information. Cloud computing, either the computer unit embedded in individual vehicles or computation facilities in cloud, supports the data process and intelligent analyzer implementation. The wireless communication frequency, speed and stability decide the assistance system performance; the intelligent analyzer procedure implement speed and algorithms stability determine the assistance system reliability. In the side of hardware or facility usage, these two places are critical to be considered. As the Ford Motor Company’s Data acquisition and communication platform (DAP) is prerequisite for the safety driving assistance system, the database management system is directly relevant to the wireless communication speed and frequency. Therefore, powerful database management system is also required to satisfy millions of vehicles data acquisition and feedback at the same time.

Another core part to the safety driving assistance system is the interface design. The interface is the part of interaction between driver and vehicle. In the safety driving assistance system, it requires to presents the two core functions, cluster-based correlation identification and time series modelling. In detail, the interface should enables driver to choose the parameters relative to the clustering, the expected time clusters and geographical clusters size, should notify the locations of the severe driving events in Maps, should monitor the variation of the severe driving events' variation in Maps, and should alert the risk of driving in a certain region or place though different models simulation and prediction. The safety driving assistance system can be designed with Graphical user interface (GUI). GUI is a type of interface containing controls called components that enable users to interact with electronic devices through graphical icons and visual indicators. With capacitive touch screen in vehicles, drivers are convenient to perform interactive communication with safety driving assistance system.

7.2 Future Work Discussion

So far, we designed the major two functions for the safety driving assistance system, identify the cluster-based correlation of severe driving events with time and location and model the count data time series of the severe driving events in regions for future correlation prediction. The correlation identification is identified by data clustering and modeling the count time series are four basic models. Further development of the system design should be focus on other modeling methods and more designed functions.

On the model construction side, one place we need to improve is to used more appropriate models to describe actual data in different regions. We specifically discuss the count time series of severe braking events in large regions ($11km$ by $8\,km$). Although the STARIMA methods can be expanded for small regions, but Bayesian Vector Autoregressive (BVA) method should be more appropriate according to the introduction by [37]. Hence, we should develop a mechanism in the system design to suggest an appropriate model according to the size of a region. In addition, we have to notice that some assumption is not appropriate. For example, $Y^{(5)}$ is assumed to determined series not statistical probability distributed as explanatory variable in generalized linear modeling. This assumption needs to be loose by using Bayesian inference in the future GLM.

On the model forecasting side, future work can be the extension on upgrade the accuracy of the proposed models. The parametric models can be developed with incorporating Kalman Filter and nonparametric model can be developed with Bayesian inference to update priori and posteriori distribution to improve the future event occurrence prediction and models' self-adaptiveness [69 - 70].

On the clustering methods side, the results presented in Chapter 4 are depending on the threshold choices, e.g. different kinds of clusters on concentration and lifetime duration. Thus, the sensitive of the thresholds to the resulting clusters are concerned. How the change of threshold affects the cluster concentration or cluster lifetime? At the same time the clustering accuracy, robustness and efficiency are also concerned to ensure the safety driving assistance system high level performance.

On the database side, the frequency and speed of data querying decide the instantness of the safety driving assistance system. How long vehicles querying vehicle data for real-time correlation identification and in ongoing traffic? The issue is strong relevant to database structure. Traditional object-oriented database cannot provide fast speed for the tens of thousands of vehicle querying data at the same time. Thus, new generation of post-relational databases need to be upgraded for very fast data query. In addition, appropriate database management system (DBM) is also very important. A DBM is computer software applications that interact with user, other applications, and database itself, and is designed to allow the definition, creation, querying, update, and administration of databases. A powerful DBM will increase the efficiency of data querying.

References

[1] National Highway Traffic Safety Administration, Overview of National Highway Traffic Safety Administration's Driver Distraction Program, Report: DOT-HS-811-299, U.S. Department of Transportation, 2010.

[2] A. B. Ellison, S. Greaves, and M. Bliemer, "Examining Heterogeneity of Driver Behavior with Temporal and Spatial Factors," Transportation Research Record: Journal of the Transportation Research Board, vol. 2386, pp. 158 - 167, 2014.

[3] R. J. Herring, "Real-Time Traffic Modeling and Estimation with Streaming Probe Data using Machine Learning," Dissertation, University of California, Berkeley, 2010.

[4] NHTSA Announces Final Rule Requiring Rear Visibility Technology, National High Way Traffic Safety Administration (NHTSA), NHTSA.gov. 2014-03-31.

[5] I. Riches, "Strategy Analytics: Automotive Ethernet: Market Growth Outlook," IEEE SA: Ethernet & IP, Keynote Speech, 2014.

[6] J. Xia and M. Chen, "Defining Traffic Flow Phase Using Intelligent Transportation Systems-Generated Data," Journal of Intelligent Transportation Systems, vol. 11, no. 1, pp. 15 - 24, 2007.

[7] Y. Fan, A. J. Khattak, and E. Shay, "Intelligent Transportation Systems: What Do Publications and Patents Tell Us?," Journal of Intelligent Transportation Systems, vol. 11, no. 2, pp. 91 - 103, 2007.

[8] Telematics Database Access, Technical report, Ford Motor Company, 2011.

[9] C. Oh, E. Jeong, K. Kang, and Y. Kang, "Hazardous Driving Event Detection and Analysis System in Vehicular Networks: Methodology and Field Implementation," Transportation Research Record: Journal of the Transportation Research Board, vol.

2381, pp. 9 - 19, 2013.

[10] G. Gan, C. Ma, and J. Wu, “Data Clustering: Theory, Algorithms, and Applications,” Society for Industrial and Applied Mathematics, 2007.

[11] A. K. Jain, M. N. Murty, and P. J. Flynn, “Data Clustering: A Review,” ACM Comput. Surv., vol. 31, no. 3, pp. 264 - 323, 1999.

[12] M. Kulldorff, F. Mostashari, L. Duczmal, K. Yih, K. Kleinman, and R. Platt, “Multivariate Scan Statistics for Disease Surveillance,” *Statistics in Medicine*, vol. 26, no. 8, pp. 1824 - 1833, 2007.

[13] M. Kulldorff, “Prospective Time Periodic Geographic Disease Surveillance Using a Scan Statistic,” *Journal of Royal Statistical Society (Series A)*, vol. 164, pp. 61 - 72, 2001.

[14] K. P. Kleinman, A. M. Abrams, M. Kulldorff, and R. Platt, “A Model-adjusted Space-time Scan Statistic with an Application to Syndromic Surveillance,” Epidemiology Infection, vol. 133, pp. 409 - 419, 2005.

[15] D. B. Neill, “Expectation-based Scan Statistics for Monitoring Spatial Time Series Data,” International Journal of Forecasting, vol. 25, pp. 498 - 517, 2009.

[16] C. Li, E. Chang, H. Garcia-Molina, and G. Wiederhold, “Clustering for Approximate Similarity Search in High-Dimensional Spaces,” IEEE Trans. Knowl. Data Eng., vol. 14, no. 4, pp. 792 - 808, 2002.

[17] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, “Multitask Spectral Clustering by Exploring Intertask Correlation,” IEEE Trans. on Cybernetics, vol. 99, 2014.

[18] E. Achtert, C. Bohm, H. P. Kriegel, P. Kroger, and A. Zimek “On Exploring Complex Relationships of Correlation Clusters,” Scientific and Statistical Database

Management, pp. 7 - 16, 2007.

[19] N. Bansal, A. Blum, and S. Chawla, “Correlation Clustering,” 43rd Annual IEEE Symposium on Foundations of Computer Science, pp. 238 - 250, 2002.

[20] M. M. Romera, M. A. S. Vazquez, and J. C. G. Garcia, “Comparing Improved Versions of ‘K-Means’ and ‘Subtractive’ Clustering in a Tracking Application,” Computer Aided System Theory, vol. 4739, pp. 717 – 724, 2007.

[21] J. P. Jose, “A Novel Method for Color Face Recognition Using KNN Classifier,” International Conference on Computing, Communication, and Application, pp. 1 – 3, 2012.

[22] H. Jin, F. Kubala, and R. Schwartz, “Automatic Speaker Clustering,” Proceeding of the DARPA Speech Recognition Workshop, pp. 108 – 111, 1997.

[23] D. Liu and F. Kubala, “Online speaker clustering,” in Proceeding of IEEE Conference on Acoustic, Speech, and Signal Processing, vol. 1, pp. 333 – 336, 2004.

[24] P. Angelov, D. P. Filev, and N. Kasabov, “Evolving Intelligent Systems: Methodology and Applications,” 1st ed., Wiley-IEEE, 2010.

[25] D. L. Pham, “Spatial Model for Fuzzy Clustering,” Computer Vision and Image Understanding, vol. 84, pp. 285 - 297, 2001.

[26] J. C. Dunn, “A Fuzzy Relative of the ISODATA Process and Its Use in Detecting Compact Well-Separated Clusters,” Journal of Cybernetics, vol. 3, no. 3, pp. 32 - 57, 1973.

[27] C. A. Sugar and G. M. James, “Finding the Number of Clusters in a Data set: An Information Theoretic Approach,” Journal of the American Statistical Association, vol. 98, no. 463, pp. 750 - 763, 2003.

[28] R. Guo and Y. Zhang, “Identifying Time-of-Day Breakpoints Based on Nonintrusive Data Collection Platforms,” Journal of Intelligent Transportation Systems, vol. 18, pp. 164 - 174, 2014.

[29] R. Tibshirani, G. Walther, and T. Hastie, “Estimating the Number of Clusters in a Data set via the Gap Statistic,” Journal of the Royal Statistical Society: Series B, vol. 63, part 2, pp. 441 - 423, 2001.

[30] M. Yan, “Methods of Determining the Number of Clusters in a Data set and a New Clustering Criterion,” Dissertation, Virginia Polytechnic Institute and State University, Blacksburg, 2005.

[31] S. L. Chiu, “Fuzzy Model Identification Based on Cluster Estimation,” Journal of Intelligent and Fuzzy Systems, vol. 2, pp. 267 - 278, 1994.

[32] L. Serir, E. Ramasso, and N. Zerhouni, “Evidential Evolving Gustafson-Kessel Algorithm for Online Data Streams Partitioning Using Belief Function Theory,” International Journal of Approximate Reasoning, vol. 53, no. 5, pp. 747 - 768, 2012.

[33] W. Yu and X. Li, “On-line Fuzzy Modeling via Clustering and Support Vector Machines,” Information Science, vol. 178, no. 22, pp. 4264 - 4279, 2008.

[34] K. Hammouda and F. Karray, “A Comparative Study of Data Clustering Techniques,” SYDE 625: Tools of Intelligent System Design, Course Project, 2000.

[35] D. E. Gustafson and W. C. Kessel, “Fuzzy Clustering with a Fuzzy Covariance Matrix,” IEEE Conference on Decision and Control, pp. 761 - 766, 1978.

[36] S. J. Deutsch and P. E. Pfeifer, “Space-Time ARMA Modeling with Contemporaneously Correlated Innovations,” Technometrics, vol. 23, no. 4, pp. 401 - 409, 1981.

[37] Y. Kamarianakis and P. Prastacos, “Space-Time Modeling of Traffic Flow,” Computers & Geosciences, vol. 31, no. 2, pp. 119 - 133, 2005.

[38] G. E. P. Box and G. M. Jenkins, Time Series Analysis: Forecasting and Control, Wiley, 2008.

[39] W. A. Fuller, Introduction to Statistical Time Series, Wiley-Interscience, 1995.

[40] MATLAB, Economics Toolbox^{TM}: User’s Guide R2014, The Math Works, Inc., 2014.

[41] J. D. Cryer and K. Chan, Time Series Analysis: with Application in R, Springer, 2010.

[42] J. D. Hamilton, Time Sereis Analysis, Princeton University, 1994.

[43] R. J. Hodrick and E. C. Prescott, “Postwar U.S. Business Cycles: An Empirical Investigation,” Journal of Money, Credit, and Banking, vol. 29, no. 1, pp. 1 – 16, 1997.

[44] J. M. Box-Steffensmeier and J. R. Freeman, Time Series Analysis for the Social Science, Cambridge University, 2014.

[45] J. Tayman, S. K. Smith, and J. Lin, “Precision, bias, and uncertainty for state population forecasts: an exploratory analysis of time series models,” Popul. Res. Policy Rev, vol. 26, pp. 347 – 369, 2007.

[46] J. M. Arnsparger, B. C. Mcinnis, and J. R. G, “Adaptive Control of Blood Pressure”, IEEE Trans. On Biomedical Engineering, no. 30, pp 168 – 176, 1983.

[47] B. M. Williams and L. A. Hoel, “Modeling and Forecasting Vehicular Traffic Flow as a seasonal ARIMA Process: Theoretical Basis and Empirical Results,” Journal of Transportation Engineering, vol. 129, no. 6, pp. 664 – 672, 2003.

[48] J. W. Hardin and J. M. Hilbe, Generalized Linear Models and Extensions, Stata,

College Station, 2007.

[49] J. E. Dennis and R. B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations, Prentice-Hall, Englewood Cliffs, 1983.

[50] MATLAB, Optimization Toolbox^{TM}: User’s Guide R2014, The Math Works, Inc., 2014.

[51] J. Pinheiro, D. Bates, S. DebRoy, and D. Sarkar, Package ‘nlme’, Linear and Nonlinear Mixed Effects Models, 2014.

[52] R Core Team, R: A Language and Environment for Statistical Computing, R Foundation for Statistical Computing, Vienna, Austria, 2013.

[53] P. E. Pfeifer and S. J. Deutsch, “A Three-Stage Iterative Procedure for Space-Time Modeling,” Technometrics, vol. 22, no. 1, pp. 35 – 47, 1980.

[54] S. J. Deutsch and P. E. Pfeifer, “Space-Time ARMA Modleing with Contemporaneously Correlated Innovations,” Technometrics, vol. 23, no. 4, pp. 401 – 409, 1981.

[55] S. J Deutsch and P. E. Pfeifer, “Variance of the Sample Space-Time Correlation Function of Contemporaneously Correlated Variables,” SIAM Journal on Applied Mathematics, vol. 40, no. 1, pp. 133 – 136, 1981.

[56] I. Kamarianakis and P. Prastacos, “Forecasting Traffic Flow Conditions in an Urban Network: A Comparison of Univariate and Multivariate Procedures,” the 82nd Transportation Research Board Annual Convention, 03-4318, 2003.

[57] X. Min, J. Hu, Q. Chen, T. Zhang, and Y. Zhang, “Short-Term Traffic Flow Forecastings of urban Network Based on Dynamic STARIMA Model,” 12th International IEEE Conference on Intelligent Transportation Systems, St. Louis, 2009.

[58] P. McCullagh and J. A. Nelder, Generalized Linear Models, Chapman and Hall, London, 1989.

[59] J. A. Nelder and R. W. M. Wedderburn, "Generalized Linear Models," Journal of the Royal Statistical Society, vol. 135, no. 3, pp. 370 – 384, 1972.

[60] M. J. Maher and I. Summersgill, "A Comprehensive Methodology for The Fitting of Predictive Accident Models," Accident Analysis and Prevention, vol. 28, no. 3, pp. 281 – 296, 1996.

[61] G. R. Wood, "Generalized linear accident models and goodness-of-fit testing," Accident Analysis and Prevention, vol. 34, no. 4, pp. 417 – 427, 2002.

[62] P. Greibe, "Accident prediction models for urban roads," Accident Analysis and Prevention, vol. 35, no. 2, pp. 273 – 285, 2003.

[63] A. Fernandes and J. Neves, "An Approach to Accidents modeling Based on Compounds Road Environments," Accident Analysis and Prevention, vol. 53, pp. 29 – 45, 2013.

[64] Z. Ye, Y. Zhang, and D. Lord, "Goodness-of-fit Testing for Accident Models with Low Means," Accident Analysis and Prevention, vol. 61, pp. 78 – 86, 2013.

[65] J. Fox, Applied Regression Analysis and Generalized Linear Models, SAGE, 2008.

[66] W. Zucchini and I. L. Donald, Hidden Markov Models for Time Series: An Introduction Using R, Chapman and Hall/CRC, 2009.

[67] D. Barber, A. T. Cemgil, and S. Chiappa, Bayesian Time Series Models, Cambridge University Press, Cambridge, UK. (Monta Calro), 2011.

[68] C. P. Robeert and G. Casella, Monte Carlo Statistical Methods, Springer, 2004.

[69] S. L Scott, "Bayesian Methods for Hidden Markov Models: Recursive Computing in

the 21st Century," Journal of the American Statistical Association, vol. 97, no. 457, 2002.

[70] M. J. Beal, Z. Ghahramani, and C. E. Rasmussen, "The Infinite Hidden Markov Model," In Advances in Neural Information Processing Systems 17: Proceedings, pp. 577 – 584, 2002.