Traffic pattern prediction in cellular networks.

Zhang, Kejing

TRAFFIC PATTERN PREDICTION IN CELLULAR NETWORKS

Kejing Zhang

Supervisor: Laurie Cuthbert

School of EE and CS

Queen Mary, University of London

November, 2011

ABSTRACT

Increasing numbers of users together with a more use of high bit-rate services complicate radio resource management in 3G systems. In order to improve the system capacity and guarantee the QoS, a large amount of research had been carried out on radio resource management. One viable approach reported is to use semi-smart antennas to dynamically change the radiation pattern of target cells to reduce congestion.

One key factor of the semi-smart antenna techniques is the algorithm to adjust the beam pattern to cooperatively control the size and shape of each radio cell. Methods described in the literature determine the optimum radiation patterns according to the current observed congestion. By using machine learning methods, it is possible to detect the upcoming change of the traffic patterns at an early stage and then carry out beamforming optimization to alleviate the reduction in network performance.

Inspired from the research carried out in the vehicle mobility prediction field, this work learns the movement patterns of mobile users with three different learning models by analysing the movement patterns captured locally. Three different mobility models are introduced to mimic the real-life movement of mobile users and provide analysable data for learning.

The simulation results shows that the error rates of predictions on the geographic distribution of mobile users are low and it is feasible to use the proposed learning models to predict future traffic patterns. Being able to predict these patterns mean that the optimized beam patterns could be calculated according to the predicted traffic patterns and loaded to the relevant base stations in advance.

ACKNOWLEDGEMENT

First and foremost, I would like to express my sincere appreciation to my supervisor, Professor Laurie Cuthbert, for his supervision and friendly support. His wide knowledge, guidance and continuous encouragements have been a great help throughout my PhD study.

I would also like to thank the staff of the Department of Electronic Engineering of Queen Mary for their help during my studies. I would like to thank all the friends in Queen Mary for our friendship.

Finally, special thanks go to my parents (idols in my life) for their generous support and endless love; and Ms Ruijie Dong for her faith in me.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

2G	Second Generation (mobile network)
3G	Third Generation (mobile network)
3GPP	3G Partnership Project
AMR	Adaptive Multi-Rate
AP	Access Point
APC	Access Point Centric
AUDG	Arc-Shaped UDG
BS	Base Station
BoD	Bandwidth on Demand
CA	Cellular Automaton
CBR	Case Based Reasoning
CP	Centralized Prediction
DP	Distributed Prediction
DS-CDMA	Direct-Sequence Code Division Method Access
DTPL	Dynamic Traffic Pattern Learning
FDD	Frequency Division Duplex
FOMA	Frontier of Mobile Multimedia Access
GGE	Geographic Grid Element
GSM	Global System for Mobile Communication
LG	Learning Grid
MBMS	Multimedia Broadcast Multicast Service
MHC	Mobile Host Centric
MLCT	Multi-Lane City Traffic
MMPA	Markov-based Mobility Prediction Algorithm
MMS	Multimedia Messaging
MMP	Multi-Movement Pattern
NaSch	Nagel-Schreckenberg

RNC	Radio Network Controller
RNG	Random Number Generator
RNS	Radio Network Sub-system
RRM	Radio Resource Management
RWP	Random Walking Point
SMP	Single Movement Pattern
STPL	Static Traffic Pattern Learning
SUDG	Square-shaped UDG
TD-SCDMA	Time Division Synchronous Code Division Multiple Access
TDD	Time Division Duplex
QoS	Quality of Service
UDG	User Distribution Grid
UMP	User Mobility Profile
UMTS	Universal Mobile Telecommunication System
USIM	UMTS Subscriber Identity Module
UTRA	UMTS Terrestrial Radio Access
UTRAN	UMTS Terrestrial Radio Access Network
VoIP	Voice-over-IP
WCDMA	Wideband Code Division Multiple Access
WMA	Weighted Moving Average

1 INTRODUCTION

With the development of the 3rd Generation (3G) mobile communication techniques, this new generation of mobile networks began to flourish and soon became the mainstream in Europe since 2005. In order to provide stable and high quality services, advanced load balancing schemes, such as applying semi-smart antennas, are introduced to meet the demands of 3G networks. With the purpose of taking the advantage of semi-smart antennas to alleviate network congestion in a predictive way, this research proposes a novel approach to simulate, learn and predict traffic patterns in 3G networks.

1.1 Research Motivation

The first successful launch of the mass market commercial 3G services was by 3 in Europe in 2003. With the development and popularization of the 3G networks, 3G services have been introduced to more and more people’s daily life. The European Union Council even suggested that the 3G operators should cover 80% of the European national populations by the end of 2005. So far, most European 3G networks are built based on the Wideband Code Division Multiple Access (WCDMA) air interface standard.

WCDMA is a wideband spread-spectrum mobile air interface that uses the Code Division Multiple Access to achieve higher transmission speeds and larger network capacity. As a complete reflection of 3G requirements, WCDMA technology has emerged as the most widely utilized 3G radio air interface. However, the variety of data services in WCDMA networks, such as surfing the internet, playing online games, etc., need advanced radio resource management algorithms to guarantee QoS and optimized system capacity. One viable approach is to use smart antenna techniques [Chry00] to dynamically change the radiation pattern of target cells to reduce congestion.

The concept of using semi-smart antennas in 3G networks is described in [CRTBB01]. One key factor of the smart antenna techniques is the beam pattern adjustment algorithm that is used to cooperatively control the size and shapes of the radio coverage. In [DBC04], Du indicated that a semi-smart antenna with six sectors and four elements for each sector could significantly increase system capacity by using the bubble oscillation algorithm. Subsequently, Yao showed that the computational workload and processing time of [DBC04] could be considerably reduced by using CBR matching to change the shape of the antenna patterns [YC05]. Moreover, [YC05] also tried to predict the consequent congestion pattern by matching a sequence of congestion solutions. This thesis describes research that extends these techniques.

When the network is considered congested, both the above two approaches work out the optimized radiation pattern in accordance with the current traffic snapshot. As described in [YC06], motions of mobile terminals are not completely random in real life, since all the motions should be constrained at least by road topologies. Thus, movement trajectories of mobile terminals should follow similar patterns in a specific area. If the local traffic patterns could be learnt and the future traffic status could be predicted in advance, proper beamforming patterns would be loaded on the relevant antennas before the exacerbation of congestion leads to a decrease in network performance.

In order to provide more accurate upcoming traffic information for the existing beamforming optimization schemes mentioned in [DBC04] and [YC05], this research proposes a novel approach to simulate, learn and predict the traffic patterns of mobile terminals in cellular networks.

In this thesis, three mobility models and three traffic pattern learning models are presented. The first two mobility models can be considered as semi-microscopic models, which are built based on the traditional Random Walking model with more advanced movement rules to reproduce the realistic inter-cell and intra-cell traffic patterns of mobile terminals. Then, a microscopic model, built based on the Cellular Automaton model, is applied to mimic traffic patterns of vehicular mobile users with inter-user interactions. On the other hand, in order to learn and make predictions of the intra-cell traffic patterns, an Order-*k* Markov model, a Static Traffic Pattern Learning model and a Dynamic Traffic Pattern Learning model are proposed and the feasibility of performing traffic pattern prediction with these models are evaluated.

1.2 Research Scope

The aim of this research is to find a general way to learn local traffic patterns and to make predictions on the geographic distribution of mobile terminals, so as to provide more accurate upcoming traffic information for the beamforming adjustment module of semi-smart antennas in cellular networks.

The previous work described in [Du04] used three load balancing schemes to perform the radiation pattern adjustment for semi-smart antennas. Each optimization was made according to the users' geographic distribution in a traffic snapshot obtained from a macroscopic mobility model that had 50,000 users distributed with predefined pattern in an area with 100 BSs. It was proved that the network performance could be significantly improved.

With the purpose of reducing the large computational overhead, a CBR-matching approach was proposed in [Yao07] to match a given congestion distribution to a best-fit one in the case library that was built up with results from the simulator used in [Du04]. Then, the optimal radiation pattern of the matched distribution was loaded to alleviate the given congestion. Notice that only the congested areas were taken into account in [Yao07]. It was proved that the network performance could be improved, and the execution time for adjusting the radiation pattern with the CBR-matching approach was much shorter then applying methods proposed in [Du04]. Besides, [Yao07] also investigated the practicality of making predictions with the

CBR-matching approach. The results showed that the accuracy of predicting upcoming patterns of growing hotspots are acceptable, but a large error might occur when predicting the congestion patterns for moving hotspots.

Based on the prior research reported in [Du04] and [Yao07], one main task of this research is to propose feasible approaches to make accurate predictions for the upcoming user distributions, for both congested and uncongested areas, and both growing mobile clusters and moving mobile clusters. [Du04] calculated the optimal radiation patterns for semi-smart antennas based on not only the number of users covered by target antennas but also the users' geographic distribution patterns in the target antennas' coverage areas as shown in Figure 1.1. If the geographic distributions of mobile terminals can be predicted accurately, load balancing schemes in [Du04] can be applied to work out optimised radiation patterns for the expected traffic distribution.

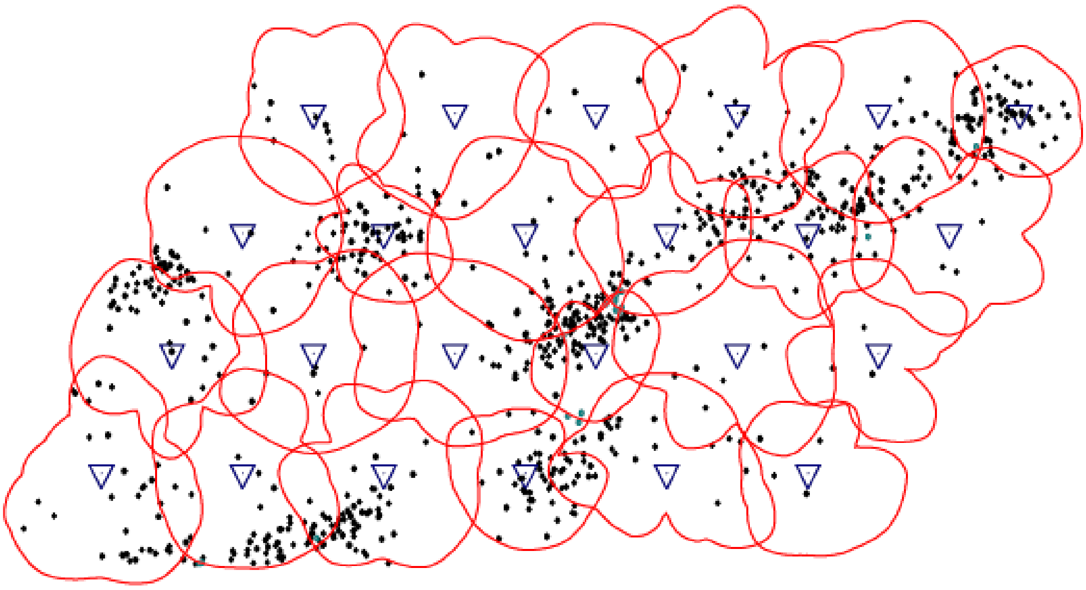
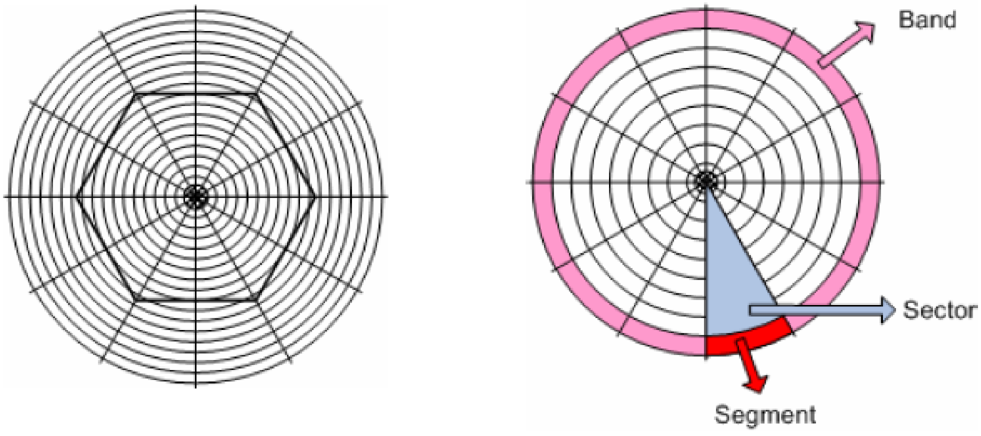


Figure 1.1 Optimised cellular coverage for hot-spots traffic (Figure 3.6 from [Du04])

[Yao07] divided each radiation coverage area of a BS into bands, sectors and segments, as shown in Figure 1.2. Then, the number of users in each segment in a given area was used to match a best-fit case in the case library with similarity functions. If the distribution of mobile clusters in each segment can be predicted precisely, load balancing schemes in [Yao07] can be used to find the proper beamforming patterns from the case library to alleviate the forthcoming congestion.



a) Cell model

b)Illustration of *band, sector* and *segment*

Figure 1.2 Illustration of cell model (Figure 3.9 from [Yao07])

Besides, the mobility model involved in [Du04] and [Yao07] could be considered as a simplified version of the RWP model with limited information of users' movement behaviours within a radio cell or between different cells. The lack of movement behaviour information might be a significant reason that [Yao07] failed to make predictions for moving hotspots. Thus, this research attempts to build mobility models that include movement behaviour information and geographic impact to reflect the real-life movement patterns of mobile users.

Therefore, this research focuses on building advanced mobility models to generate realistic cell transition trajectories, and evaluating the feasibility of using proposed pattern learning models to learn local movement patterns and to make predictions on the upcoming geographic distributions of mobile terminals and mobile clusters by mining the captured cell transition trajectories.

1.3 Research Contributions

The work presented in this thesis is novel. The main contributions are:

- Performing traffic pattern predictions for the intra-cell distribution of mobile users and mobile clusters, which is novel in the location management and mobility prediction field.

- Introducing two novel semi-microscopic mobility models and one novel microscopic mobility model to mimic the movement of mobile users in a realistic way.

- Proposing three pattern learning models, the Order-*k* Markov model, the Static Traffic Pattern Learning (STPL) model and the Dynamic Traffic Pattern Learning (DTPL) model, to learn the geographic movement patterns from the cell transition data and to make predictions on the upcoming traffic patterns of mobile users and future positions of mobile clusters. The remaining two pattern learning models are original from this research.

- Introducing the concept of letting traffic pattern prediction cooperate with the load balancing schemes of semi-smart antennas, which extends the existing research of [Du04] and [Yao07].

1.4 Author's Publications

1. K. Zhang and L. Cuthbert, "Traffic Congestion Forecasting in WCDMA Networks," in Proceedings of the 3rd International Conference on Communication and Networking in China (ChinaCom2008), pp. 992-996, Aug. 2008.

2. K. Zhang and L. Cuthbert, "Performing Traffic Pattern Prediction in WCDMA Networks," in Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR08), pp. 832-837, Aug. 2008.

3. K. Zhang and L. Cuthbert, "Random Traffic Prediction in WCDMA Networks," in Proceedings of the 4th International Conference on Wireless Communication,

Networking and Mobile Computing (WICOM08), Oct. 2008

4. K. Zhang and L. Cuthbert, “Traffic Pattern Prediction in Cellular Networks,” in Proceedings of the 11th IEEE Singapore International Conference on Communication Systems (ICCS2008), pp. 549-553, Nov. 2008.

5. K. Zhang and L. Cuthbert. “Predicting the Distribution of Mobile Users in Cellular Networks,” in Proceedings of the IEEE International Conference on Communication Technology and Application (ICCTA2009), Oct. 2009.

1.5 Organisation of the Thesis

The remainder of the thesis is organised as follows.

Chapter 2 provides the state-of-the-art relevant to this research. It starts with an overview of the development of the mobile networks before moving on to introduce WCDMA and its relevant radio resource management methods. Then, smart antennas and their operational mechanisms are described to explain how network performance can be improved by using semi-smart antennas when congestion occurs. Finally, the ideas that have been proposed to tackle traffic prediction are discussed.

Chapter 3 gives a detailed introduction of the basic mobility model and the first two pattern learning models applied in this research. The concept and the components of the basic mobility model are introduced first. Then, the Order-k Markov model and the original STPL model are presented. Finally, result evaluation methods are introduced.

Chapter 4 presents the intra-cell traffic pattern prediction proposed in this research. To start with, an overview of traffic pattern prediction is given. Then the configuration of the mobility model and the pattern learning models are introduced, followed by the description of the predefined scenarios. Finally, the simulation results are presented and discussed.

Chapter 5 presents the traffic pattern prediction carried out in more sophisticated ways. An enhanced mobility model is presented first. Then, an improved pattern learning model, the original DTPL model, is proposed. Finally, the simulation results are discussed and the accuracy of the prediction is evaluated.

Chapter 6 introduces a cellular automaton model to mimic the vehicular traffic patterns with inter-user interactions. Then, the traffic pattern prediction is carried out with the Order-k Markov model. Finally the simulation results are presented and discussed.

Chapter 7 presents the conclusions and future work of this research.

2 BACKGROUND

This chapter provides the state-of-the-art relevant to this research. It starts with an overview of the development of the mobile networks before moving on to introduce WCDMA and its relevant radio resource management methods. Then, smart antennas and their working mechanisms are described to explain how network performance can be improved by using semi-smart antenna when congestion occurs. Finally, the ideas that have been proposed to tackle traffic prediction are discussed.

2.1 Mobile Networks

Universal Mobile Telecommunications System (UMTS), which has a dominant role in Europe, is one of the widely utilized third generation (3G) mobile telecommunications technologies. It is part of the ITU IMT-2000 standard and is specified by 3GPP. As 3G systems, UMTS networks are designed for flexible delivery of a variety of services based on high bit rates.

Generally speaking, services supported by UMTS networks can be categorized into person-to-person services, content-to-person services and business connectivity [HT04]. The person-to-person service type includes Adaptive Multirate (AMR) speech service, video telephony, multimedia messaging (MMS), Voice-over-IP (VoIP), multiplayer games, etc. These services mentioned above can be further divided into person-to-person circuit switched services and person-to-person packet switched services. For the content-to-person service type, web browsing, audio and video streaming, content downloading and Multimedia Broadcast Multicast Service (MBMS) are the main components. The business connectivity service type considers access to corporate intranet or to Internet services using laptops.

Being a complete network system, UMTS specifies the UMTS Terrestrial Radio Access Network (UTRAN) that handles all radio related functionality, the core network that takes charge of switching and routing calls and data connections to external networks, and the authentication of users via the UMTS Subscriber Identity Module (USIM). Figure 2.1 illustrates an overview of the UMTS system architecture.

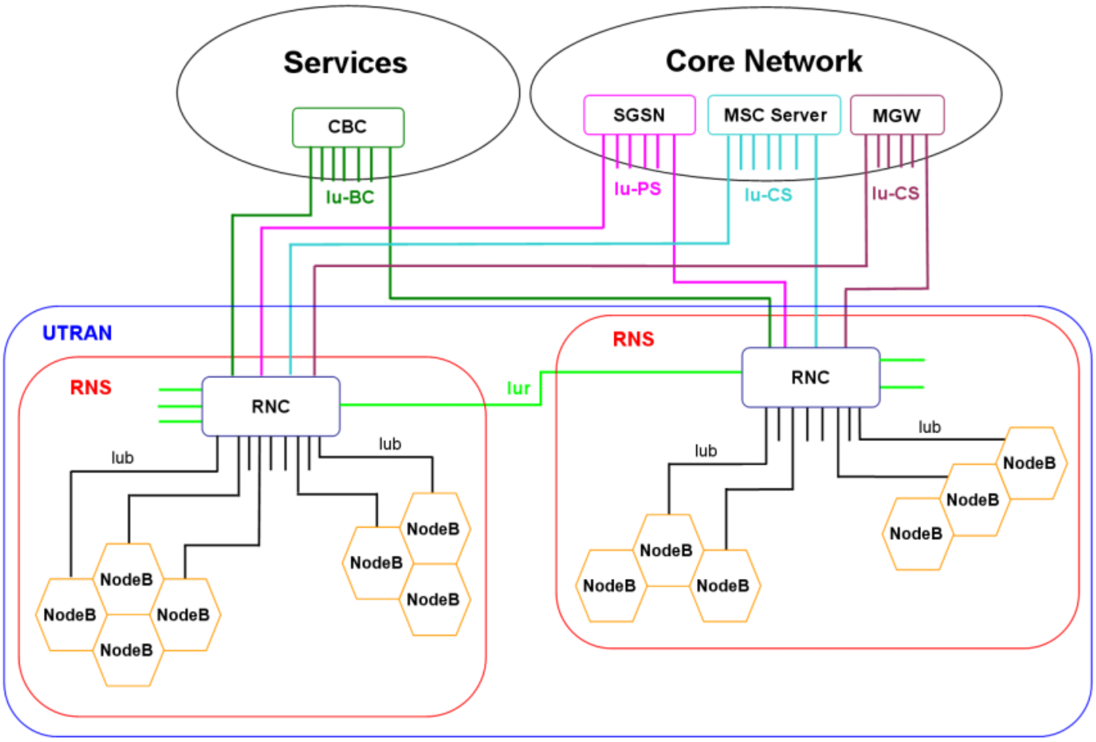


Figure 2.1 UMTS network architecture

In UMTS networks, the Node B, more often known as the Base Station (BS), is responsible for providing the radio link for mobile terminals and managing the radio resources within its coverage area. The Radio Network Controller (RNC) manages a group of radio cells, and owns and controls the radio resources in its domain. The sub-network composed by one RNC and at least one Node B is called a Radio Network Sub-system (RNS). As shown in Figure 2.1, a number of cooperatively working RNS can be considered as a UTRAN.

One important part of IMT-2000 is the specification of air interface options. For the UMTS networks, the UMTS Terrestrial Radio Access (UTRA) includes three main air interface standards, WCDMA, TD-CDMA and TD-SCDMA. Since WCDMA is dominant in Europe, and is relevant to this research, its features are discussed in the next section.

2.2 WCDMA

Wideband Code Division Multiple Access (WCDMA) is a wideband spread-spectrum mobile air interface standard. It was first developed by NTT DoCoMo for the early 3G network Frontier of Mobile Multimedia Access (FOMA). Then, WCDMA was selected as an air interface for UMTS, and was specified as part of the 3G standard in IMT-2000.

2.2.1 Technical features

WCDMA uses the Direct-Sequence Code Division Method Access (DS-CDMA) technology to transmit on a pair of 5MHz-wide radio channels. It supports the concept of Bandwidth on Demand (BoD) whereby the data capacity allocated for a user might vary from frame to frame according to the real time traffic demand. Frequency Division Duplex (FDD) and Time Division Duplex (TDD) are the two duplexing methods supported by WCDMA. In the FDD mode, both the uplink and downlink can have 5MHz carrier bandwidth for transmission use. By contrast, only one 5MHz bandwidth is shared between the uplink and downlink in the TDD mode. Table 2.1 summarizes the main parameters of WCDMA air interface.

Table 2.1 Main WCDMA parameters [Table 3.1 in HT04]

Multiple access method	DS-CDMA
Duplexing method	FDD / TDD
Base station synchronization	Asynchronous operation
Chip rate	3.84 Mcps (5 MHz carrier bandwidth)
Frame length	10 ms
Service multiplexing	Multiple services with different QoS requirements multiplexed on

	one connection
Multi-rate concept	Variable spreading factor and multi-code
Detection	Coherent using pilot symbols or common pilot
Multi-user detection, smart antennas	Supported by the standard, optional implementation

WCDMA is designed to enable 3G networks to achieve higher transmission speed and larger network capacity. Using WCDMA, UMTS is able to support maximum theoretical data transfer rates of 21Mbit/s, which is also a key factor for UMTS to support a variety of mobile data services, such as web browsing, file downloading, etc. In order to guarantee the QoS of these services as well as increase the system capacity, more advanced radio resource management approaches should be involved in WCDMA networks.

2.2.2 Radio resource management in WCDMA

Radio resource management (RRM) can be considered as a system level control that is responsible for efficient utilization of the limited radio spectrum resources and radio network infrastructure. Generally speaking, the RRM algorithms used in WCDMA networks can be divided into power control, handover control, admission control, load control and packet scheduling functionalities [HT04].

Power control is utilized to minimize the interference level and to provide the required QoS. In CDMA systems, power control is especially important since interference comes from other mobile terminals. Without proper controls, a single overpowered mobile terminal could block a whole cell. Two power control algorithms are mainly used in WCDMA networks, the closed loop power control [SC01] and the outer loop power control [OP98].

Handover control is adopted to enable mobile terminals to work seamlessly when moving across cell boundaries. In WCDMA networks, the softer handover and soft handover are introduced to improve the performance of the hard handover. When a mobile terminal is in the overlapping cell coverage area of two adjacent sectors of one base station, the softer handover is triggered. During softer handover, each sector allocates a channel for the terminal to communication with the base station. When a mobile terminal is in the overlapping cell coverage area of two sectors belonging to different base stations, the soft handover is triggered. During soft handover, the mobile terminal concurrently communicates with both base stations from two air interface channels provided by each base station separately. Figure 2.2 shows the above two scenarios. According to [HT04], softer and soft handovers typically occur in about 5-15% and 20-40% of connections respectively.

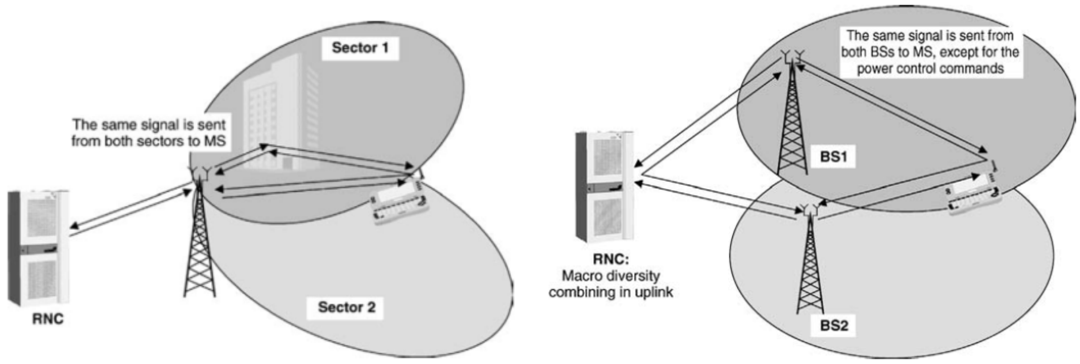

Sector 1
The same signal is sent from both sectors to MS
RNC
Sector 2
The same signal is sent from both BSs to MS, except for the power control commands
BS1
RNC:
Macro diversity combining in uplink
BS2

Figure 2.2 Softer and soft handover scenarios (Figure 3.11 and 3.12 from [HT04])

For admission control, load control and packet scheduling, they are required to provide the required QoS and maximize the system throughput. Since the radio resources within a radio cell are limited, it is impossible for a base station to keep admitting connection request without considering the consumption of the radio resources. In a WCDMA network, admission control is involved to determine the impact of allowing an incoming connection request from the perspectives of system capacity and call dropping probability. Moreover, packet scheduling is used to support packet switched services and analyse their performance for the sake of guaranteeing QoS of these services. Proper admission control and packet scheduling methods could ensure that the network is in a stable state without being overloaded. When the system is overloaded, load control is triggered to return the system back to the expected states. Existing load control methods currently used in the WCDMA networks are [HT04]:

- Downlink fast load control;
- Uplink fast load control;
- Reduce the throughput of packet data traffic;
- Handover to another W-CDMA carrier;
- Handover to GSM;
- Decrease bit rates of real time applications;
- Drop low priority calls in a controlled fashion.

The characteristics of WCDMA air interface enable advanced CDMA receiver concepts, such as multiuser detection and smart antennas, to be realized, which makes it possible to introduce advanced load control approaches to increase system capacity and optimize the radiation pattern. This research intends to carry out the load control in a predictable way to extend the existing load control schemes of semi-smart antennas.

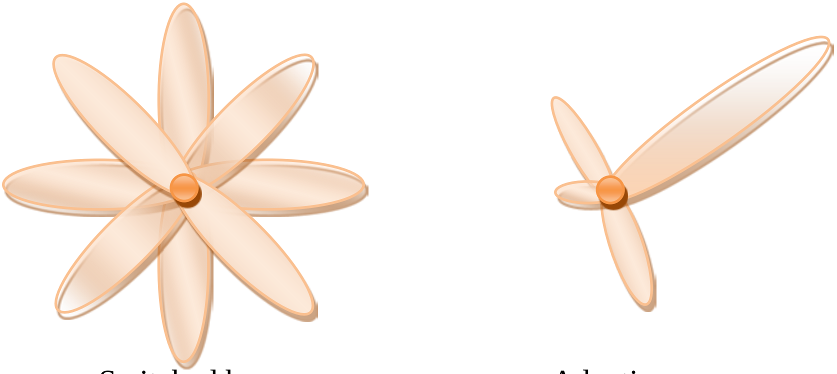
2.3 Using Smart Antenna in 3G Networks

Smart antennas have been adopted in areas such as radars, satellite communications, and remote sensing. Generally speaking, a smart antenna refers to a system of antenna arrays arranged in a certain distributed configuration with a specialized signal processor. So far, a considerable amount of work, such as [Chry00], [GL04], [PCAB06] and [TAP00], has proved that using smart antennas in cellular networks can significantly improve the system performance and capacity. In this section, a brief introduction of smart antenna technology is presented, followed by a discussion of the load balancing schemes relevant to this research.

2.3.1 Smart antenna technology

A smart antenna system has a number of antenna arrays with signal processing capability to transmit and receive in an adaptive way. Using a smart antenna, the radiation pattern of the transmitter can be customized to match the traffic conditions. In real 3G networks, although various technologies offer higher data rates and double voice capacity compared with their 2G counterparts, their actual performance is still susceptible to interference and adverse channel conditions created by multipath propagation and system loading. Therefore, smart antenna techniques are introduced as an enhancement to 3G systems to improve network performance and capacity by alleviating the degradation caused by co-channel interference and other adverse factors.

According to [Chry00], smart antenna systems are commonly classified into two categories, switched beam system and adaptive array system. Figure 2.3 illustrates these two kinds of smart antenna.



Switched beam

Adaptive array

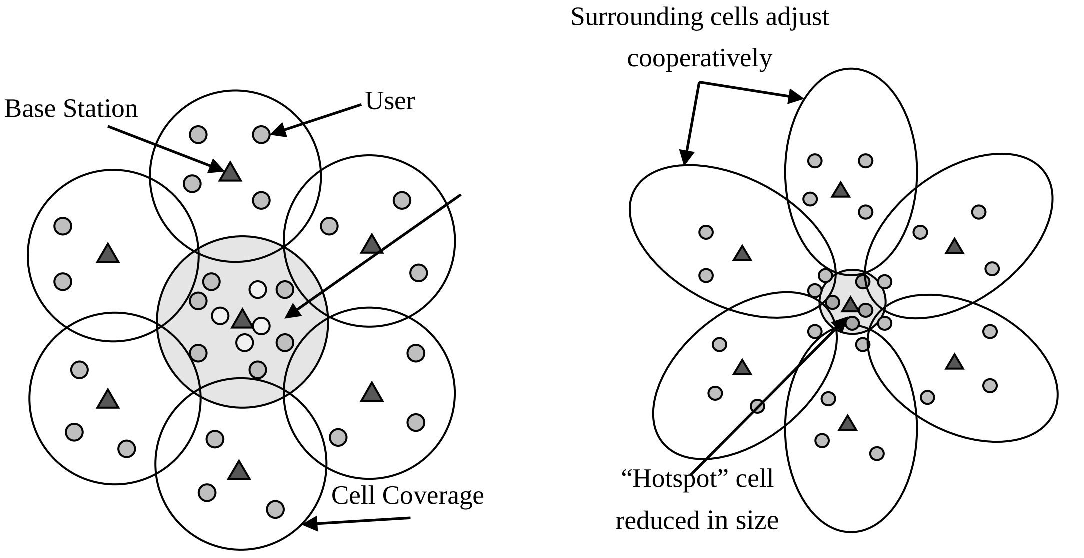
Figure 2.3 Illustration of smart antenna types

Briefly speaking, the most significant difference between these two types is the radiation pattern adjusting scheme. For the switched beam system, several available fixed beam patterns are recorded to support the adjustment of beamforming shapes. All the patterns loaded on the antenna are chosen from this predefined pattern set. On the other hand, adaptive array system allows the antenna to fine-tune the beam pattern in real time according to the real traffic demand. Although both systems attempt to increase gain in the direction of the user, only the adaptive array system offers optimal gain, while simultaneously identifying, tracking, and minimizing interfering signals [Comp88].

2.3.2 Semi-smart antenna system

Admittedly, using smart antennas can increase the utility of radio resources, providing both enhanced data rates and improved coverage and spectral utilization. However, the cost on modifying the existing cellular system architecture and the difficulties of maintaining and managing such systems cannot be neglected. Even building a dedicated network using smart antennas, for example the TD-SCDMA network in China, shows much more unexpected difficulties. Therefore, a mutation of smart antenna with lower complexity, called semi-smart antenna, was introduced to overcome the limitations identified in existing smart antenna systems.

The aim of developing such a system is to introduce a much simpler scheme that meets the industry needs of accommodating traffic hotspots dynamically, but with minimum changes to the existing infrastructure [PCAB06]. The basic principle behind the load balancing scheme used in the semi-smart antenna system is to shape the radiation pattern of each cell in real time according to the traffic needs. Congestion in a heavily loaded cell is decreased to match the guaranteed QoS by contracting the radiation pattern around the hotspot, while adjacent cells expand their radiation patterns to compensate for coverage loss, as illustrated in Figure 2.4.


Base Station
User
Cell Coverage
Surrounding cells adjust cooperatively
"Hotspot" cell reduced in size

(a) Conventional fixed radiation patterns (b) Cells adjust shape to handle congestion

Figure 2.4 Semi-smart antenna system for geographic load balancing in a cellular network (Figure 2.7 from [Yao07])

Compared to other approaches used by smart antenna systems, semi-smart antenna systems can accomplish the load control only with the knowledge of the location of mobile clusters, rather than the location of individual mobile terminals. Since the movement of mobile clusters is much slower than the movement of individual mobiles, the system can afford a much longer update period between two pattern loadings. According to [PCAB06], this update period can be 30 seconds.

With proper load balancing algorithms to work out the optimal radiation patterns for the current traffic, semi-smart antenna system can significantly improve the network performance and capacity with much lower complexity and cost. Two load balancing algorithms relevant to this research are discussed in the rest of this section.

2.3.3 Bubble oscillation algorithm

Having got antennas that are capable of adjusting beamforming shapes, it is necessary to design load balancing algorithms that can work out the optimal radiation patterns according to the traffic demands. In the literature [DBCNP03] [DBC03] [DBC04], three different load balancing algorithms were proposed to guide the process of radiation pattern adjustment according to the geographic traffic distribution. These algorithms were the cooperative negotiation approach, the utility-based approach, and the bubble oscillation approach. Then, it was proved in [Du04] that using the bubble oscillation algorithm can effectively and efficiently improve the system capacity and decrease the call blocking rate in WCDMA networks. Thus, the bubble oscillation algorithm is described below in detail.

The main concept of this approach is to use an analogy with air bubbles. The local coverage scheme is treated as an air bubble; the local traffic load is treated as the air within the bubble; the un-served traffic is treated as a vacuum between adjacent bubbles [DBC04]. Figure 2.5 illustrates the comparison between the natural bubble oscillation and the radio cell oscillation.

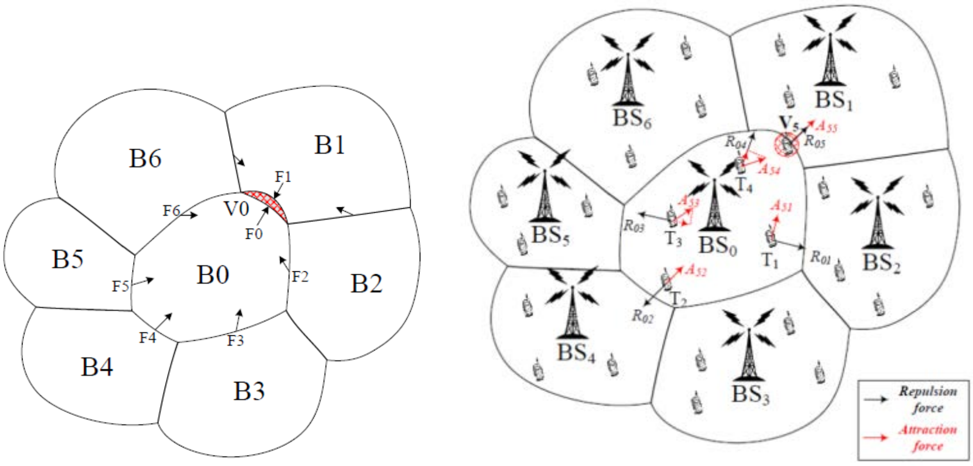


Figure 2.5 Comparison between the oscillation of natural bubbles and the radio cells (Figure 6.1 and 6.4 from [Du04])

When running the bubble oscillation algorithm, the congested areas are considered as “temporary vacuum area”, which causes the coverage shape oscillation of every cell to minimize these vacuum areas. In each iteration, the attraction forces generated by the un-served mobiles are calculated to update the repulsion forces of the relevant mobiles. Then, the local optimizations are performed with the updated repulsion forces. The result converges when the number of un-served mobiles remains constant or becomes zero. The flowchart of the bubble oscillation algorithm is shown in Figure 2.6.

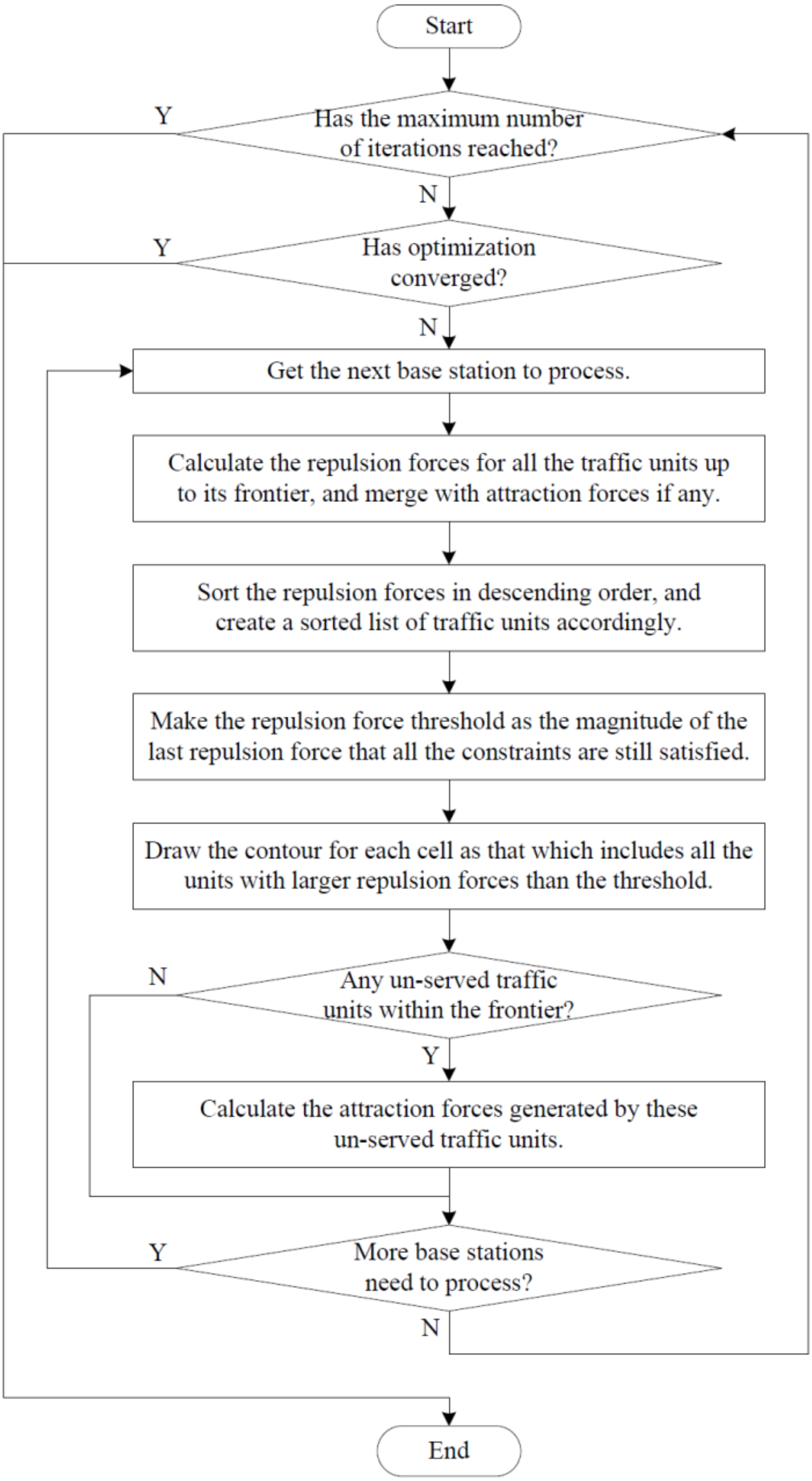

Start
Has the maximum number of iterations reached?
Y
N
Has optimization converged?
Y
N
Get the next base station to process.
Calculate the repulsion forces for all the traffic units up to its frontier, and merge with attraction forces if any.
Sort the repulsion forces in descending order, and create a sorted list of traffic units accordingly.
Make the repulsion force threshold as the magnitude of the last repulsion force that all the constraints are still satisfied.
Draw the contour for each cell as that which includes all the units with larger repulsion forces than the threshold.
Any un-served traffic units within the frontier?
N
Y
Calculate the attraction forces generated by these un-served traffic units.
More base stations need to process?
Y
N
End

Figure 2.6 Flowchart for the process of the bubble oscillation algorithm

(Figure 6.3 from [Du04])

Figure 2.7 shows the simulation results of adopting the bubble oscillation algorithm in the WCDMA network to assist semi-smart antennas in radiation pattern optimization. It had been proved in [Du04] that the bubble oscillation algorithm can obviously improve the network performance.

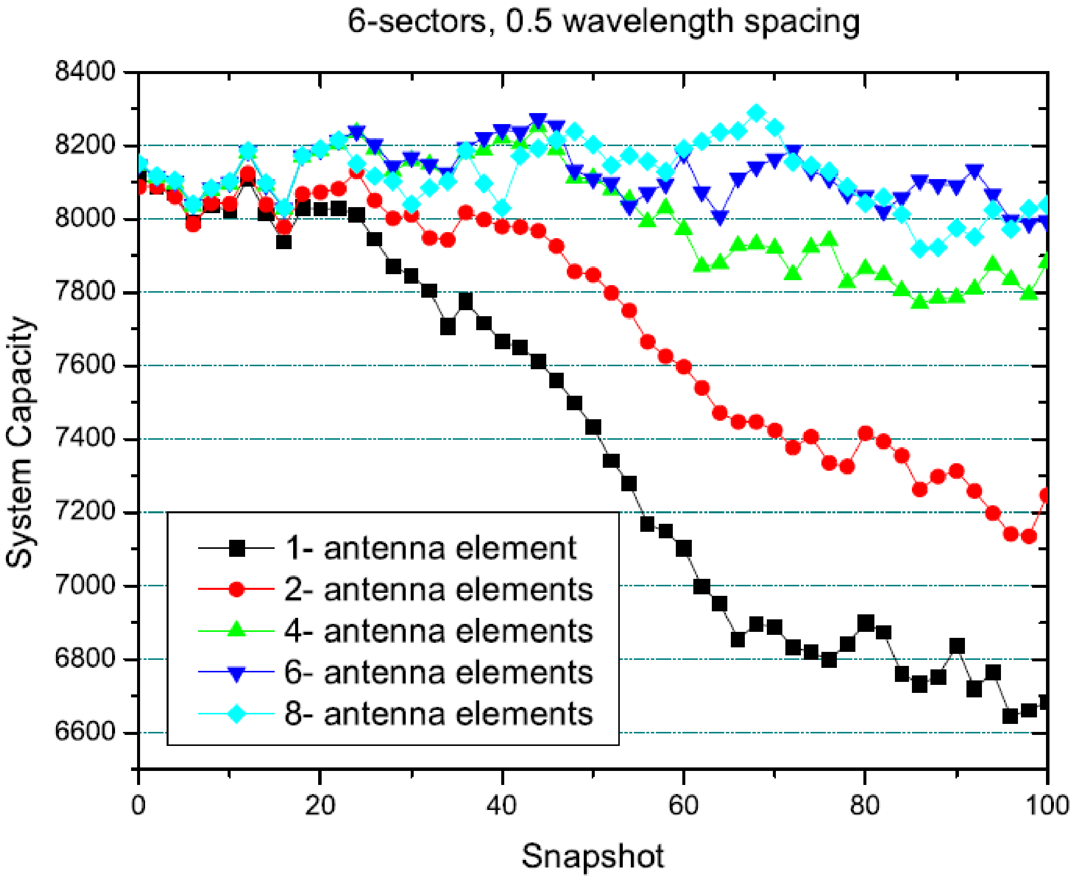


Figure 2.7 Simulation results of the bubble oscillation algorithm

(Figure 7.19 from [Du04])

The load balancing schemes described in [Du04] adjust the cell size and shape according to the geographic traffic distribution. The geographic traffic distribution has the same meaning as the term “traffic pattern” used in this research. In Du’s simulations, the optimized radiation patterns were generated based on the traffic pattern captured in each snapshot. The time interval between every two continuous snapshots is 60 s. For each snapshot, the traffic pattern was analysed with the conventional cell shape. If congestion areas are found in one snapshot, then the load balancing scheme is invoked to work out the suitable radiation pattern for the current situation.

The concept of applying the bubble oscillation algorithm to alleviate the network congestion can be considered as the workflow that is illustrated in Figure 2.8.

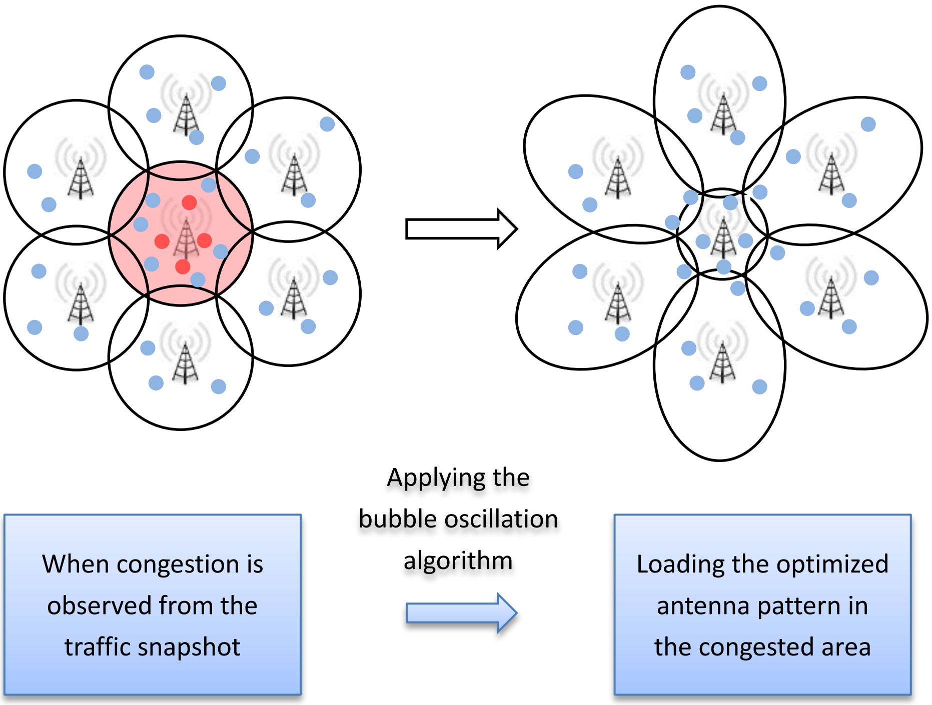

When congestion is observed from the traffic snapshot
Applying the bubble oscillation algorithm
Loading the optimized antenna pattern in the congested area

Figure 2.8 The workflow when apply the bubble oscillation algorithm

As indicated in the above figure, the bubble oscillation algorithm is applied when traffic congestion is observed from the snapshot. Regarding the mobile traffic pattern, or in other words the distribution of mobile users, in the congested areas, the antenna pattern optimization is carried out with the bubble oscillation algorithm. By adjusting the beamforming shape of the semi-smart antenna with congestion and semi-smart antennas in the neighbouring areas, the traffic burden in the congested cell can be eased by the extended coverage of the neighbouring cells. When the calculation is accomplished, the optimized antenna patterns are loaded on the relevant semi-smart antennas to improve the network performance.

When using the bubble oscillation algorithm, the optimization problem is a NP-hard problem with considerably heavy calculation overhead. Although the network performance can be improved significantly with the bubble oscillation algorithm, the computational workload for every optimization cannot be neglected.

2.3.4 CBR-matching algorithm

With the purpose of decreasing the computational workload and processing time for the bubble oscillation algorithm, a Case Based Reasoning (CBR) matching approach was introduced in [YC05] [YC07] and [Yao07]. The principle is that congestion often follows certain patterns, which means that the optimal radiation pattern found for one congestion case can be reloaded when a similar situation reoccurs.

Different from the work described in [Du04] which takes the geographic traffic distribution into account to adjust the radiation pattern, the approach taken in [Yao07] is based on congestion patterns, rather than traffic patterns. Figure 2.9 illustrates how a congestion pattern is determined.

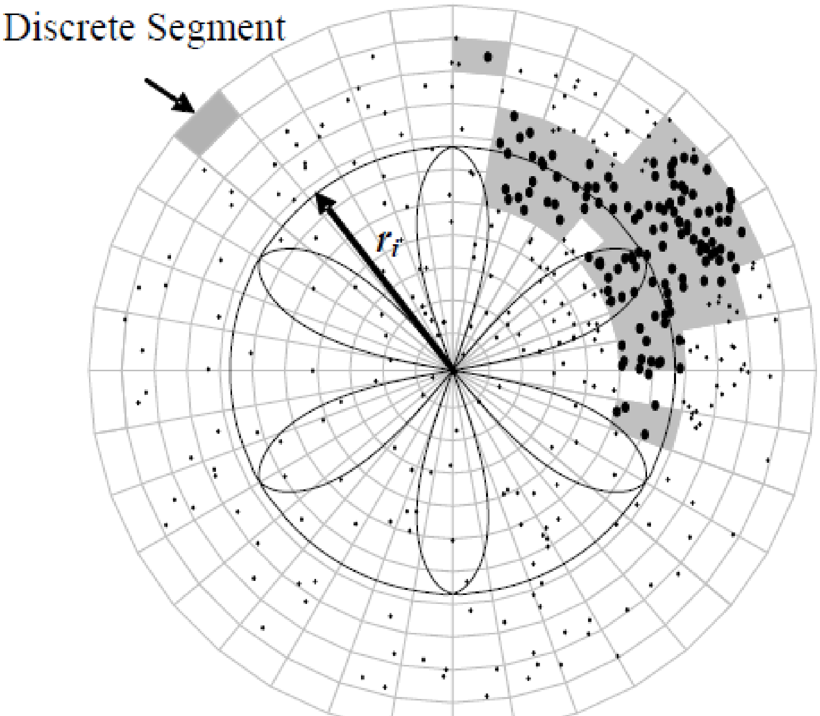


Figure 2.9 Illustration of radiation pattern and congestion pattern

(Figure 5.1 from [Yao07])

In Figure 2.9, the conventional coverage area of a cell is divided into a number of segments. The circle represents the antenna coverage with equally-excited radiation sectors; the black dots represent mobile terminals located in this area; the shadow areas represent the segments with one or more blocked traffic units. The blocked units can be determined by inputting the traffic snapshot into the network simulator used in [Du04]. Then, one segment is considered as a blocked segment if the number of blocked units is larger than a predefined threshold. Finally, the congestion pattern of one cell is defined as the distribution of the blocked segments in that cell. The reason of using congestion patterns instead of traffic patterns is to reduce the amount of data to be matched and to concentrate on the areas where congestion occurs.

In order to carry out the CBR algorithm, a case library is built to record historical data. In the case library, each case has two parts, the situation description part and the solution part [Yao07]. Every component in the situation part contains the description of a congestion pattern and the relevant radiation pattern when that congestion occurs in one cell. The corresponding solution part records the adjusted radiation pattern in that cell and the surrounding six cell's radiation patterns.

When applying matching, the existing antenna and congestion patterns are compared with the corresponding values in the library by using similarity functions. The best-matched case, which has the lowest similarity value, is retrieved from the library, and the related solution is loaded by the antenna in that cell. Figure 2.10 shows the system performance improvement after involving CBR matching mechanism in the network. Figure 2.11 shows the execution time of the bubble oscillation approach and the CBR matching approach under same scenarios. It can be seen that the system performance is obviously improved; and the execution time of the CBR matching is much less than the bubble oscillation algorithm. Moreover, [Yao07] had proved that both growing hotspots and moving hotspots could be handled by the CBR matching algorithm in the matching phase. However, problems come about in the prediction phase, which is explained in the following section.

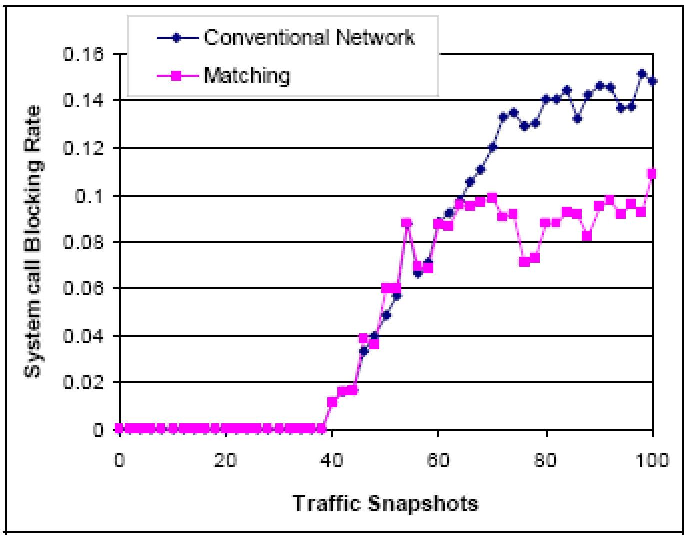


Figure 2.10 Improvement in system blocking rate by applying CBR approach (Figure 4.8 from [Yao07])

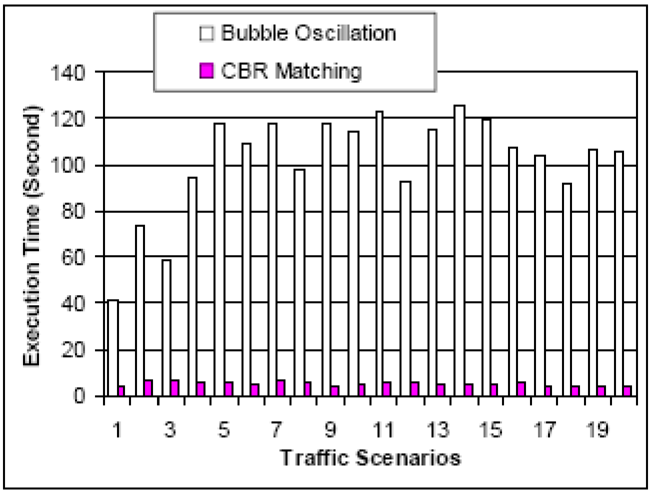


Figure 2.11 Comparison of the execution time (Figure 4.9 from [Yao07])

The concept of using CBR matching algorithm to ease the traffic congestion can be considered as the workflow that is shown in Figure 2.12.

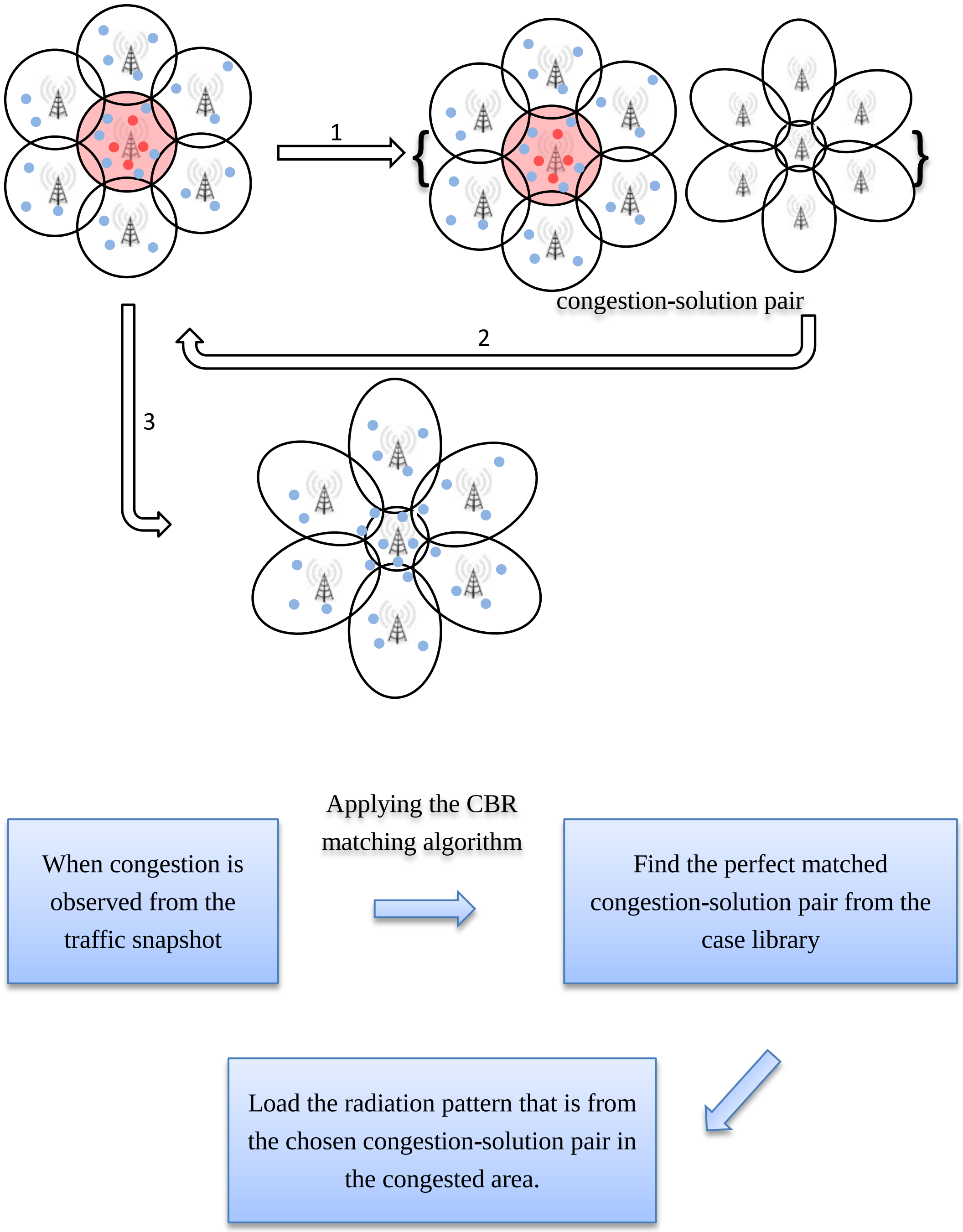

1
congestion-solution pair
2
3
When congestion is observed from the traffic snapshot
Applying the CBR matching algorithm
Find the perfect matched congestion-solution pair from the case library
Load the radiation pattern that is from the chosen congestion-solution pair in the congested area.

Figure 2.12 The workflow when apply the CBR matching algorithm

In [Yao07], the CBR approach was also used to carry out congestion pattern predictions. In the prediction phase, future congestion patterns were predicted if previous matching results can match a continuous sequence of historical congestion patterns. All the predictions took place based on the information from the case library and from the congestion patterns without using any information from the geographical distribution and movement of the mobile terminals. The simulation results in [Yao07] revealed that the CBR scheme does have the potential to predict future congestion patterns, but mainly for growing hotspots rather than moving hotspots.

2.3.5 Extending the existing research

The work presented in the previous two sections can be regarded as the foundation for this research. [Du04] aimed at implementing an intelligent load balancing scheme to dynamically optimise the cellular coverage patterns according to the geographic traffic pattern. Letting the bubble oscillation algorithm work cooperatively with the semi-smart antenna system, the network capacity can be improved considerably. The main drawback of this scheme is the heavy computational workload for each optimisation. In order to realize a swift load balancing, [Yao07] introduced the CBR matching algorithm only considering the congestion pattern instead of the entire traffic pattern in each snapshot. When determining the optimised radiation pattern by matching the current congestion pattern to those captured ones in the library, it is proved that the capacity improvement is still significant and the processing time is much faster; this works on both the growing hotspots and the moving hotspots. When making predictions on future radiation patterns based on the information from the case library and from the congestion patterns without using any prediction information from the geographic traffic distributions, it turns out that the antenna reshaping cost can be reduced mainly for growing hotspots rather than moving ones. As the existing approaches do not pay enough attention to the upcoming traffic status, this research introduces traffic pattern prediction to fill the gap.

When making predictions on the future traffic patterns of a growing mobile cluster, the aim is to determine the size and scale of the target cluster and the affected areas in the upcoming observations. On the other hand, making predictions for a moving mobile cluster aims at finding out its movement or dispersal tendency and areas that might be affected by the mobile cluster in the following snapshots. If the upcoming traffic patterns could be predicted, proper radiation patterns would be calculated and loaded on the relevant BS to alleviate the impact caused by the congestion.

According to the prediction results for moving hotspots in [Yao07], it can be seen that only matching the current congestion pattern to the historical ones cannot obtain enough information to picture upcoming situations. In real life, there is no completely random movement of a mobile terminal, since movements should be restricted by the intention and the habits of the owner, the local road topology and other factors. Thus, it is reasonable to believe that the generalized local movement features could be extracted from the relevant segments of the cell transition trajectories. Inspired from the mobility prediction field, the intention and movement trends of one terminal owner could be learnt by analysing the historical movement data of a specific terminal; the local road topology of an area could be learnt by analysing the historical movement date of terminals passing by. With the learnt local movement features, future traffic patterns could be predicted by estimating the upcoming distribution of each mobile terminal based on its current location and relevant local features.

2.4 Mobility Prediction

In this section, the existing research in the mobility prediction field that is relevant to this research is presented. First, the necessity of performing mobility prediction is discussed. Then, approaches that inspire this research are described.

2.4.1 Performing mobility prediction

Mobile devices, such as mobile phones, PDAs and laptops, are very common and appear to be becoming more widespread in recent years. This situation forces the wireless network providers to find ways to increase the capacity of their networks. In big cities, an often used solution for cellular networks is to decrease the antenna coverage and increase the number of base stations so as to increase the frequency reuse. Nevertheless, this method causes more frequent handovers and increasing paging costs, which might lead to a serious QoS degradation [CZDTZ04]. One possible way to mitigate these problems is to guess the next access point (AP) of a mobile terminal so as to take pro-active actions before handover happens or incoming calls arrive, which is known as the mobility prediction. Many studies have been carried out to manage the mobility prediction in various wireless networks, such as WiFi, ad hoc and GSM networks.

Generally speaking, the research goal in the mobility prediction field is to forecast the next possible locations of a mobile terminal during its movement according to the known historical data. In most research, the next possible locations refer to areas covered by the specific APs in Ad Hoc networks or BSs in cellular networks. Besides, the prediction methods used in the mobility prediction field can be classified into two categories, as mentioned in [FL07]. These two categories are the Mobile Host Centric (MHC) approach and the Access Point Centric (APC) approach. For the MHC, each mobile terminal tries to train a movement pattern learning model by the historical data carried by itself and perform the prediction individually. Thus, the MHC prediction process can be considered as a distributed prediction process. For APC, each AP builds one or more pattern learning models and trains each model by the local motion patterns that have been captured by the AP. In the prediction phase, predictions are activated and made by the relevant AP. Thus, the APC approach can be considered as a centralized prediction process. In order to realise an intelligent location management, different modelling techniques have been applied for mobility prediction.

In [BD99] and [BD02], the data compression algorithm was utilized to learn the characteristic of movements between radio cells. The authors of [BD99] and [BD02] believe that good compressors usually are good predictors since recent research in computational learning theory shows that prediction is synonymous with generalization and data compression. Every access point in [BD99] is named by a unique symbol from a finite alphabet. Thus, the location history of every user can be listed as a string of symbols. Then, the LeZi-update compression algorithm is used to make predictions on the next access point of a mobile user to reduce the paging cost. The authors of [YL02] extended the prediction model used in [BD99] by introducing the factor of channel holding time. This model can predict not only the cell to which a mobile will handoff but also when the handoff will occur.

In [AW04] [Quin05] and [BBK11], a user mobility profile (UMP) framework was proposed to record many important factors associated with a mobile terminal's behaviour. Then, this UMP was used to predict the possible neighbours which a mobile terminal might move into and the possible service type in the next observation.

In [LBC98] [PSJ04] and [ZM05], variants of the Kalman filter were used to perform the mobility prediction. In these works, the mobility state consisted of the position, velocity, and the acceleration of a mobile user, and the movement of a mobile user was considered as the transition between different states. In [LBC98], future mobility states were estimated by using a modified Kalman filter with observations taken from there independent pilot signal strength measurements from three different base stations. [ZM05] improved the algorithm in [LBC98] by employing a pre-filter to obtain coarse position estimates prior to the application of the Kalman filter. In [PSJ04], a Robust Extended Kalman Filter was proposed to estimate the next cell a mobile would visit. These approaches can give good results in the case of regular movements. However, poor prediction results can be seen in irregular mobility conditions.

In [AZ01] [SVA06] and [ITV09], probabilistic modelling approaches were applied. [AZ01] made use of the mobility history and the radio signal strength measures to indicate a group of radio cells, called the shadow cluster. The probability of visiting cells in the shadow cluster by the target mobile user should be larger than a threshold. [ITV09] extended the work in [AZ01] by introducing a Predictive User Mobility Behaviour algorithm to improve the threshold for choosing shadow clusters. [SVA06] proposed an activity based mobility prediction scheme to determine the target user's location of interest, preference and the probable entry and exit times.

Artificial neural networks and Markov chains remain the most widely used techniques in mobility prediction. When applying neural networks, one of the key factors that can influence the prediction performance is the training sets used to train the neural networks. Different types of movement data were taken into account as input parameters in [AS07a] [AS07b] [BEJ97] [CB04] [CBCS02] [Chak02] [JHSRP10] and [Quin05] to uncover the movement characteristics. When building learning models with Markov chains in [BK10] [BBK11] [FL07] [GV06] [SB07] [SK04] and [SWYS10], a number of variants of Markov chain were applied and several additional mobility data were taken into account. The applications of artificial neural networks and Markov chains will be discussed in the following sections.

2.4.2 Mobility prediction with neural networks

The neural network algorithm is one of the most famous artificial intelligence models that are widely used in the data mining field. It is an interconnected assembly of simple processing nodes, whose functionality is loosely based on the animal neuron, as shown in Figure 2.13. The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to a set of training patterns [Gurn03]. With the variants of neural network, both the linear regression and the non-linear regression can be carried out. Due to the strong capability of regression, [AS07a] [AS07b] [BEJ97] [CB04] [CBCS02] [Chak02] [JHSRP10] and [Quin05] carried out the mobility prediction by building the pattern learning models with different variants of neural networks, and creating the training sets with different types of mobility data to train the models. When using neural networks to predict future states, the structure of the neural networks and the quality of the training sets (composed by input patterns and expected output patterns) directly affect the prediction performance.

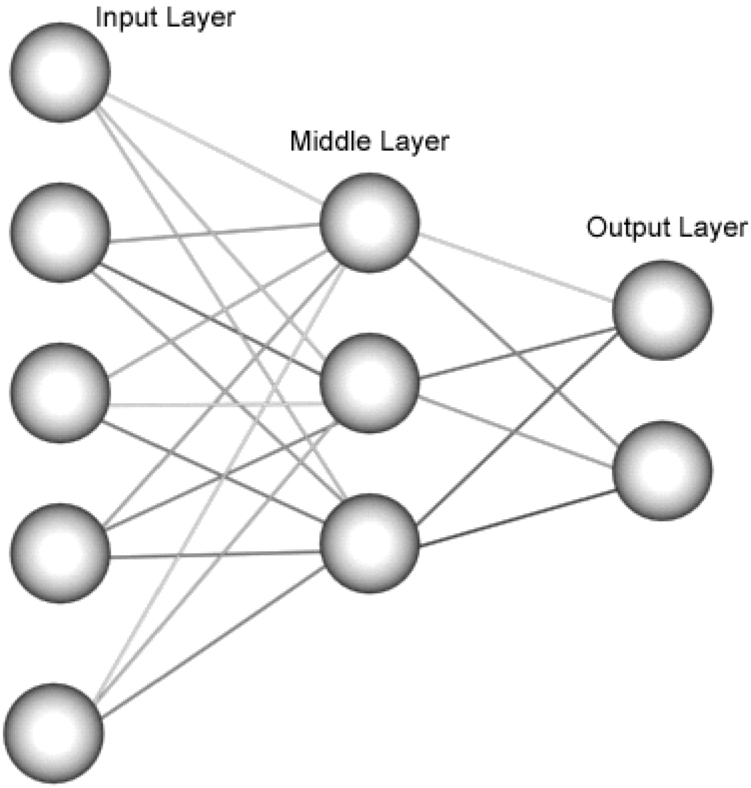


Figure 2.13 A simple model of neural network

In [AS07a] [AS07b] [BEJ97] [CBCS02] [Chak02] [JHSRP10] and [Quin05], MHC approaches are used to carry out the mobility prediction. The general argument when using the MHC in the above works is that although the mobility patterns seen in the networks as a whole are complex, these patterns become much simpler and more regular when viewed on a per-user basis.

In [BEJ97] [CBCS02] [Chak02] and [Quin05], user mobility profiles were used to generate the training sets. An example of the daily MH mobility profile is shown in Figure 2.14. The authors believe that, for a majority of mobile terminals for most of the time, the mobility profile repeats on a day-to-day basis. The next movement of a mobile user strongly depends on the present location and the time of the day. A feedback neural network was used in [BEJ97] and feed-forward neural networks were used in [CBCS02] [Chak02] and [Quin05]. Simulation results show that the above approaches work fine only when the movement pattern repeats almost unchanged day by day.

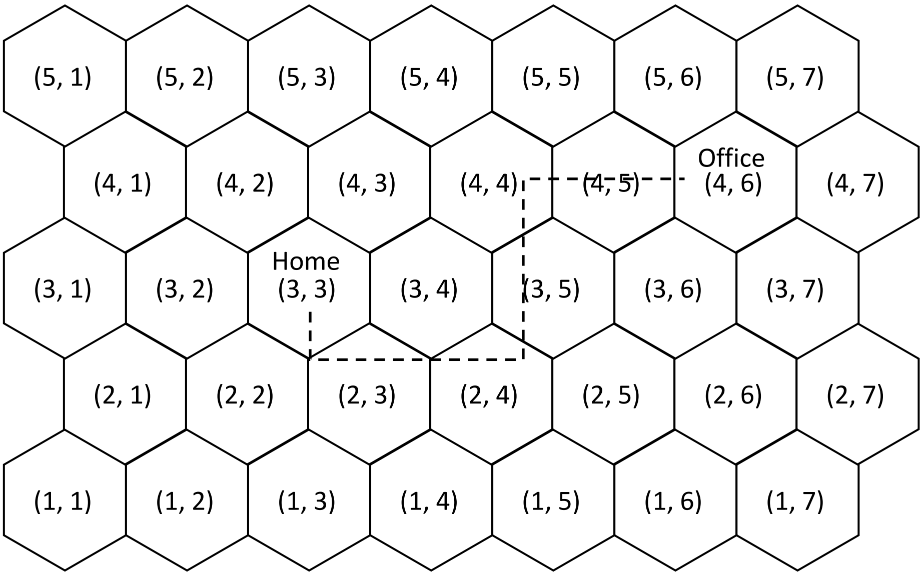


Figure 2.14 An example of the daily MH mobility profile

In [JHSRP10], a feed-forward back-propagation neural network was used to predict the movement speed of the mobile user. Training sets were generated with three variables, the simulation time, the mobility speed and the desired speed. This approach can give good results in the highway scenario. Poor results can be seen if the movement direction varies frequently.

In [AS07a] and [AS07b], the author used a Bayesian neural network to predict the next locations the users would probably be given their past movements. In these works, the input vector is composed by Cell ID, Cell History, Start Hour, Start Minute and Day of week; the output value is the next Cell ID. According to the author, the major benefit of the Bayesian neural network is that the resulting prediction is an average prediction of possible traditional neural network solutions weighted by their probability. The results show that the proposed model works well on the regular movements but not the random ones. One contribution of [AS07a] and [AS07b], which inspired this research, is that the proposed method is topology-independent.

In [CB04], the APC approach was used to carry out the mobility prediction. Figure 2.15 shows the APC mobile tracking proposed in [CB04]. The author believed that the mobility patterns would be influence in a significant way by the geographic conditions present at the location of an AP. Two types of information about the user's mobility behaviour, the movement direction and the movement speed, were included in the training sets. Then, a back-propagation neural network was used to learn the movement characteristic. The simulation results show that the proposed model can give good results in the urban scenario as enough mobility data can be captured in the observed area in the low-speed movement scenario. The feature inspiring this research from the above work is that the APC approach is better able to handle erratic behaviour by a single user.

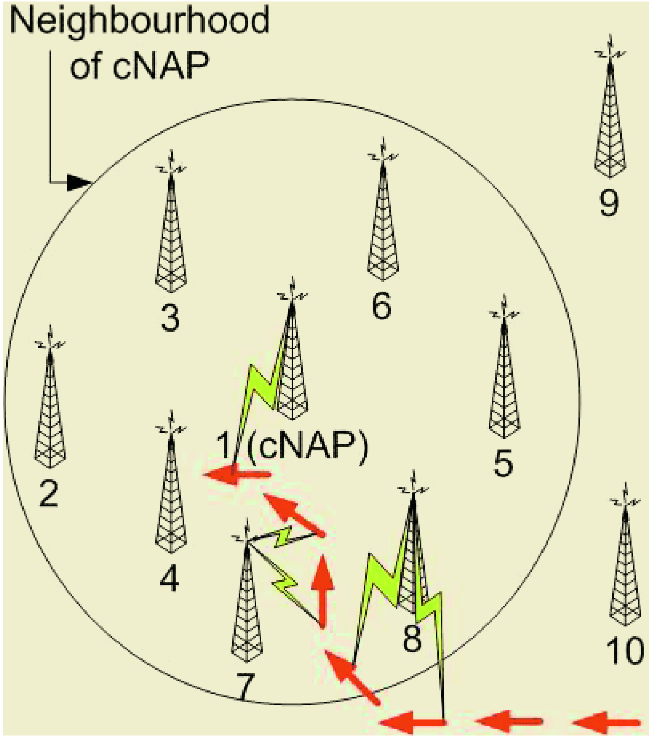


Figure 2.15 An example of the APC mobile tracking (Figure 3 from [CB04])

2.4.3 Mobility prediction with Markovian models

Generally speaking, the Order-*k* Markov predictors, widely used in the mobility prediction field, attempt to predict the next location from a given *k*-length movement context comprised of the k most recent observations in the mobility history. In [SKJH04], the author made a comparison between Markov-based and compression-based predictors. It was proved that low-order Markov predictors perform better than the more complex and more space-consuming compression-based predictors. Due to the low complexity and good performance of the Order-*k* Markov predictor, [BK10] [BBK11] [FL07] [GV06] [SB07] [SK04] and [SWYS10] applied a number of variants of Markov chain with several additional mobility data to carry out the mobility prediction.

The authors of [FL07] believe that the sequence of APs crossed by each mobile terminal is an interesting source of information. Because the movements are always constrained by the geographical conditions and a mobile user usually has a precise purpose for a specific movement, it is possible to extract the movement characteristics from the captured AP sequences. In [FL07]], two mobility prediction methods are studied, the MHC method and the APC method. For the MHC method, each user trains a learning model using its own movement historical data. For the APC method, each AP learns the local movement characteristic based on the captured movement historical data of mobiles that has visited that AP. Figure 2.16 indicates the learning process for order-2 Markov models. Each time a terminal moves, its new AP is appended to its mobility trace; a special “OFF” AP is added when the user is disconnected from the network. According to the results in [FL07], the measured accuracy difference between APC and MHC is only a few percent (typically 55% vs. 59%). Moreover, the authors indicate that building models specific to certain periods of time (week / weekend, morning / afternoon) does not bring any improvement.

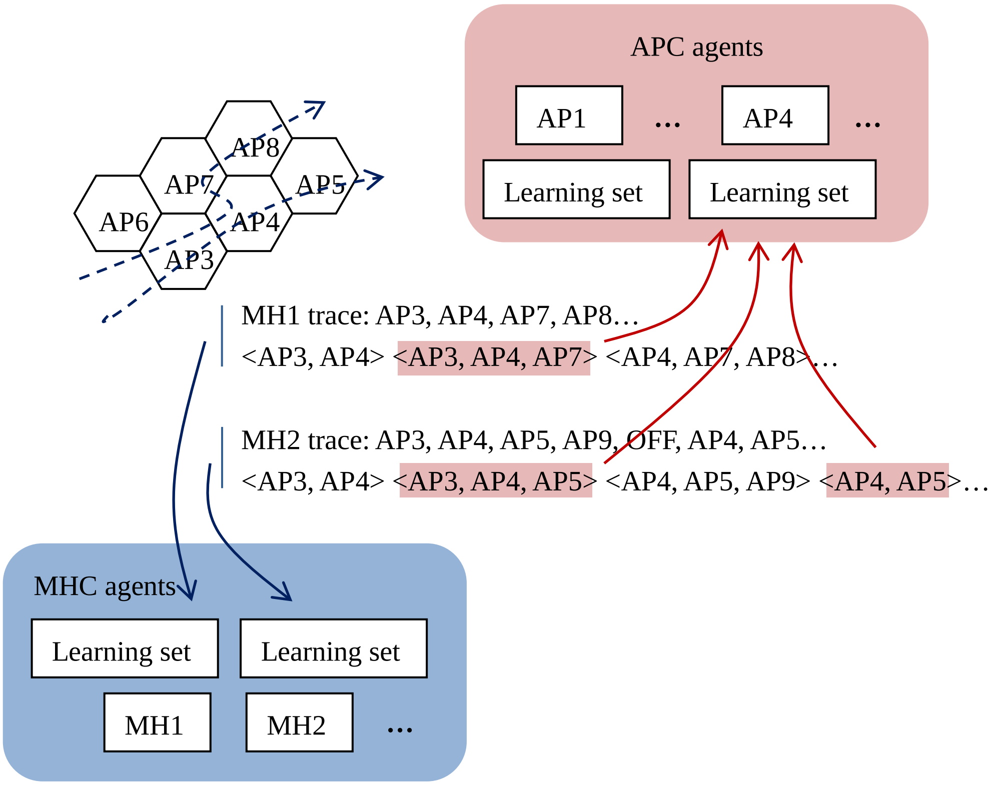

APC agents
AP1
...
AP4
...
Learning set
Learning set
AP8
AP7
AP5
AP6
AP4
AP3
MH1 trace: AP3, AP4, AP7, AP8…
<AP3, AP4> <AP3, AP4, AP7> <AP4, AP7, AP8>…
MH2 trace: AP3, AP4, AP5, AP9, OFF, AP4, AP5…
<AP3, AP4> <AP3, AP4, AP5> <AP4, AP5, AP9> <AP4, AP5>…
MHC agents
Learning set
Learning set
MH1
MH2
...

Figure 2.16 Overview of the learning process for order 2 Markovian models

In [SK04], the author performed the mobility prediction with an order-2 Markov predictor and other additional data, such as road topology, cell geography and the average road-crossing time. In order to create the training sets for the Markov model, roads in the observed area were divided into segments, as shown in Figure 2.17. According to the simulation results, the author believed that incorporating road topology information into the prediction technique could potentially yield better prediction accuracy for mobile users in vehicles. Inspired from the work in [SK04], this research treats the geographic layout of the observed area as a key factor when learning the local movement patterns.

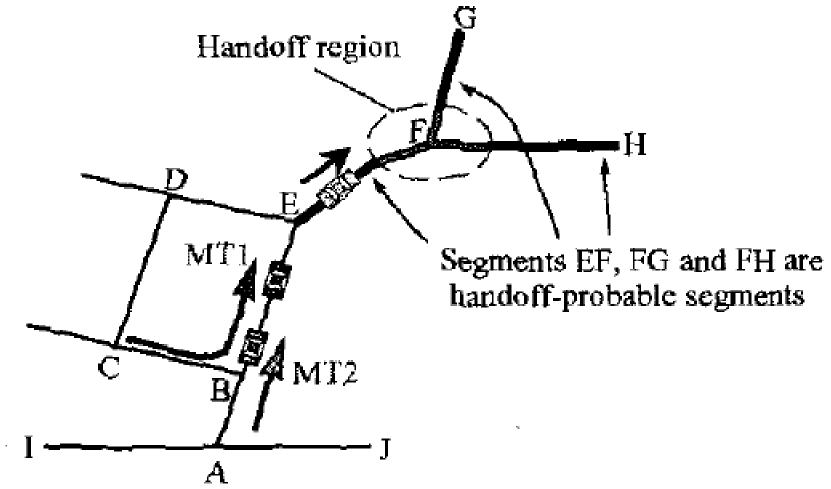


Figure 2.17 Utilizing road topology information for mobility prediction (Figure 1 from [SK04])

In [BK10] and [BBK11], the author presented an APC approach and a MHC approach respectively. For the APC predictor in [BK10], a Markov-based Mobility Prediction Algorithm (MMPA) was proposed. The MMPA algorithm was build based on the order-2 Markov model with two additional enhancements, the visit frequency generator and the ping pong handover eliminator. The visit frequency generator will be invoked if the mobility history data are not enough to support the Markov predictor; the ping pong handover eliminator is used to alleviate the influence from the ping pong handover phenomena. In [BBK11], the user mobility profile is used cooperatively with an order-2 Markov model to make predictions for individual users. Inspired from [BK10], the local visit frequency is taken into account when learning the generalized movement patterns.

In [GV06] and [SWYS10], hidden Markov models are used for mobility prediction. The hidden Markov model can be interesting whenever movement observations can be obtained rather than the exact mobility states due to measurement errors and other noise. However, this model introduces complex computations.

2.5 Summary

This chapter starts with an overview of the development of the mobile networks before moving on to introduce WCDMA and its relevant radio resource management methods. Then, the smart antenna concept and its working mechanisms are described to explain how network performance can be improved by using semi-smart antenna when congestion occurs. Finally, the ideas that have been proposed to tackle traffic prediction are discussed.

Notice that most of the mobility prediction methods described in this chapter aim at location prediction of an individual user. Just the next AP or BS's ID of each moving user is not enough for the semi-smart antenna system to optimize the coverage pattern of each antenna sector using the existing optimization methods, for example the Bubble Oscillation method and the CBR matching method mentioned above. In order to take the advantage of semi-smart antennas to improve the network capacity, future geographic distributions of mobile terminals and mobile clusters are needed by the load balancing module of the semi-smart antenna system.

Thus, this research attempts to make predictions on the geographic distributions of a number of mobile users in specific areas in advance. The term "geographic distribution" includes the number of users in sub-areas that compose the coverage area of each AP, as well as the location of each mobile user and mobile cluster in the observed area. In the next chapter, the traffic pattern prediction carried out in this research will be explained in detail.

3 MOBILITY MODELLING AND TRAFFIC PATTERN LEARNING

The main task of this research is to predict the future traffic distribution of mobile users, which can be used by the semi-smart antenna system to load proper beamforming patterns to alleviate the network performance decrease caused by the upcoming traffic congestion. In order to generate movement data for traffic pattern learning, a mobility model that reflects features of the real-life traffic of mobile users is needed. Then, pattern learning modules are needed to learn the movement pattern of mobile users and make predictions of upcoming patterns in specific areas. In order to evaluate the performance of the learning modules, error rate calculation methods are involved to indicate the prediction performance.

3.1 The Overview of Traffic Pattern Prediction

In real life, the movements of mobile users are rarely completely random, since all the motions should be constrained by many factors, such as the road topology the intention of the movement, etc. The location information of a moving mobile terminal that can be easily obtained is the relative position of that terminal with reference to the BS connected by it. As all the motions should obey the same road restriction in the same area, it is reasonable to believe that the road topology as well as the movement characteristic of a specific area can be learnt by analysing the intra-cell transition data captured in this area. Besides, the movement trajectories of different mobile users with similar intentions usually look alike, for example the routes of pedestrians from the stadium to the nearest tube station after a football match. Thus, the individual movement habit recorded in each terminal could be used to indicate the mobile trend more accurately based on the learnt local movement characteristic.

The traffic pattern prediction carried out in this research is similar to the user mobility prediction that is mentioned in the previous chapter. The similarity between them is that both attempt to guess the future location of moving mobile terminals. In order to develop the traffic prediction method used by the semi-smart antenna systems, traffic patterns with local geographic details should be forecasted to support the radiation pattern adjustment. Thus, the traffic pattern prediction is interested in predicting the locations of upcoming or moving mobile clusters and the distribution status of the mobile users within each BS rather than just the number of users within each BS. Knowing the distribution of mobile terminals, it can be easily discovered which base stations and which parts of coverage areas of these base stations are heavily loaded. Figure 3.1 (a) and (b) shows how the traffic pattern prediction module would fit within the semi-smart antenna system.

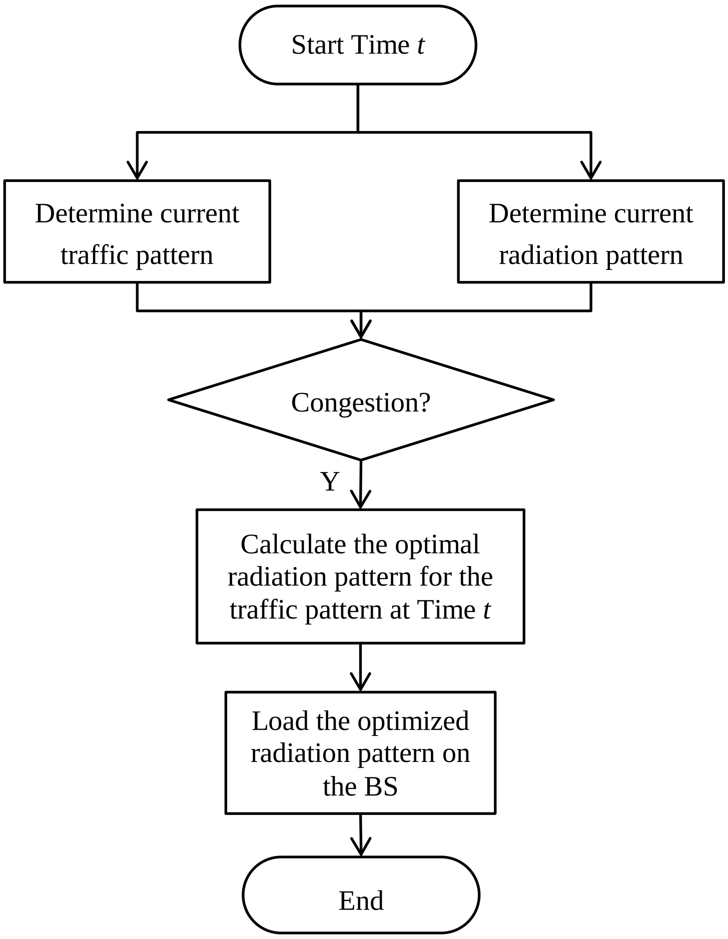

Start Time *t*
Determine current traffic pattern
Determine current radiation pattern
Congestion?
Y
Calculate the optimal radiation pattern for the traffic pattern at Time *t*
Load the optimized radiation pattern on the BS
End

Figure 3.1 (a) The flowchart of a semi-smart antenna system

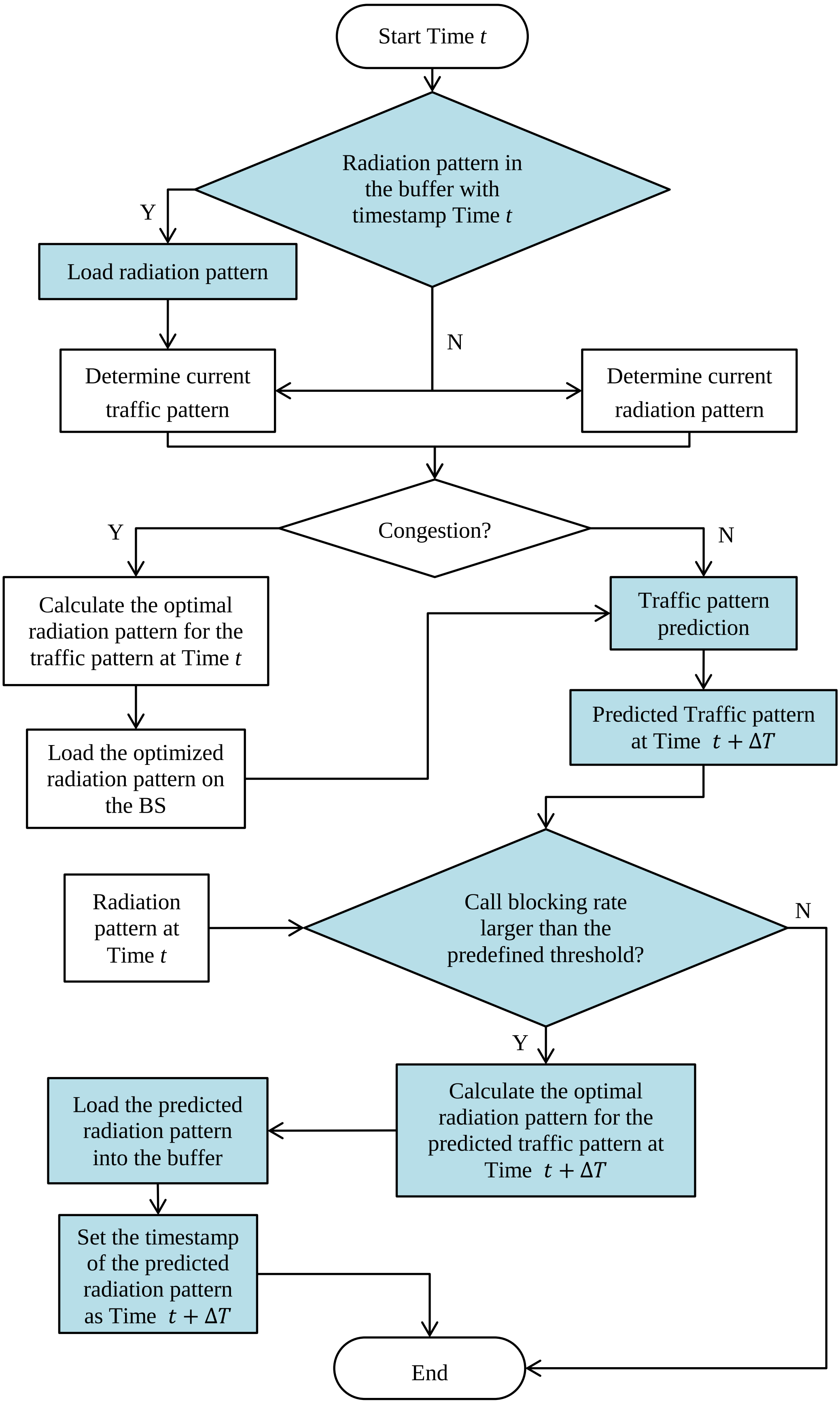

Start Time *t*
Radiation pattern in the buffer with timestamp Time *t*
Y
Load radiation pattern
N
Determine current traffic pattern
Determine current radiation pattern
Congestion?
Y
N
Calculate the optimal radiation pattern for the traffic pattern at Time *t*
Traffic pattern prediction
Load the optimized radiation pattern on the BS
Predicted Traffic pattern at Time $t + \Delta T$
Radiation pattern at Time *t*
Call blocking rate larger than the predefined threshold?
N
Y
Load the predicted radiation pattern into the buffer
Calculate the optimal radiation pattern for the predicted traffic pattern at Time $t + \Delta T$
Set the timestamp of the predicted radiation pattern as Time $t + \Delta T$
End

Figure 3.1 (b) The flowchart of a semi-smart antenna system with the traffic pattern prediction module

Figure 3.1 (a) shows the flowchart of a normal semi-smart antenna system; Figure 3.1 (b) shows the flowchart of a semi-smart antenna system with the traffic pattern prediction module. The blue areas represent the steps introduced by the traffic pattern prediction module. With the normal semi-smart antenna system, an optimized radiation pattern will be loaded on the relevant BS to improve the network performance when network congestion is found. With the traffic pattern prediction module, the upcoming traffic pattern can be estimated and the relevant optimized radiation pattern can be calculated in advance. The variable ΔT in Figure 3.1 is the time interval between two continuous changes of radiation patterns. According to [YWZCX11], there is no need for rapid changes of radiation patterns in semi-smart antenna systems; and a fast algorithm for beamforming optimization is not required as the time interval ΔT can be set up to 100s. Thus, additional steps introduced by the traffic pattern prediction module can be tolerated by semi-smart antenna systems.

Similar to the classification of prediction methods in the mobility prediction area, approaches used in this research are classified into centralized prediction (CP) and distributed prediction (DP).

- CP approach: Each BS is responsible for building and maintaining pattern learning models. Learning models are trained with the local movement data that have been captured by the BS to uncover the local movement regularly. When making predictions, each BS plays a leading role in the process.

- DP approach: Each observed mobile terminal records its own movement habit and trains the pattern learning models to reveal the individual behaviour characteristics. Predictions are activated and made by each terminal in the prediction phase.

It can be seen that the CP approach is used to obtain the general behaviour features in an area. The main benefit of the CP approach is that this scheme can quickly get an estimation of the learnt model since it improves every time a mobile goes by. On the other hand, the DP approach intends to indicate the individual feature during the observation. One reason of using DP approach is that the behaviour of a particular mobile cannot be simplified to the mean behaviour of all the users moving in the same area. However, the DP approach does not work when a mobile terminal enters the observation area for the first time, since there is no historical data related to this area. Thus, combining the CP and DP approaches together may improve the performance of the traffic pattern prediction. In this research, CP learning models are applied to reveal the geographic features of an observed area. Then, individual movement features are taken into account in the pattern learning phase to improve the accuracy of the prediction.

3.2 Existing Mobility Models

When carrying out telecommunication network simulations, mobility models, such as models proposed in [AHMES09], [AKAET09], [CST96], [HMA06] and [SK99], are always involved to mimic the movement of a number of real mobile terminals in a simulation area. According to [CBD02], current mobility models can be classified into two types, traces and synthetic models. Traces are those mobility patterns that are captured in real life systems. Information extracted from traces can more accurately reflect the real traffic situations. However, the expense and time needed for collecting traces from the real network sometimes cannot be afforded. Thus, synthetic models are introduced in this field. As implied by the name, synthetic models attempt to realistically represent the behaviours of the moving mobile users without the use of traces. During the simulation, speed and direction of the user must be changed in reasonable time slots. In terms of the correlation among the moving mobile users, the synthetic model can be further classified into entity mobility models and group mobility models. Mobile users move independently in the entity mobility model. By contrast, users’ decisions on movement might depend upon others in the group mobility model.

According to [CBD02] and [CJS04], the most widely used synthetic entity models include: Random Walk Mobility Model and its probabilistic version, Random Waypoint Mobility Model, Boundless Simulation Area Mobility Model and Gauss-Markov Mobility Model. On the other hand, the most widely used synthetic group models includes: Exponential Correlated Random Mobility Model, Colum Mobility Model, Nomadic Community Mobility Model, Pursue Mobility Model and Reference Point Group Mobility Model. The mobility models that inspire this research are the Random Walk Model, the Gauss-Markov Model, the Nomadic Community Model, and the Boundless Simulation Area model. The relation of those models to this research will be discussed in the following sections.

3.2.1 Random Walk Model

The Random Walk Model was first proposed mathematically by Einstein in 1926 and was widely used (e.g. [BKM94], [GM99], [RC97] and [ZD97]) to simulate the extremely unpredictable movement of mobile entities. This model is sometimes referred to as Brownian Motion Model. The moving rules of mobile terminals in this model are very simple. At the beginning of each simulation, a number of mobile terminals are uniformly distributed in a simulation area. Each mobile terminal moves from its current location to the next location by randomly choosing a movement direction and a movement speed with the following moving rules at time *n*.

$$\begin{gathered} \theta_n = uniform\_rand(0, 2\pi) \\ v_n = uniform\_rand(v_{min}, v_{max}) \\ \begin{cases} x_{n+1} = x_n + v_n \cdot \cos\theta_n \\ y_{n+1} = y_n + v_n \cdot \sin\theta_n \end{cases} \end{gathered} \tag{3.1}$$

where $uniform\_rand(a, b)$ is the function that returns a value uniformly distributed between a and b; (x_{n+1}, y_{n+1}) and (x_n, y_n) are the location of this mobile terminal at time $(n + 1)$ and n respectively; θ_n and v_n are the movement direction and speed the mobile terminal chooses at time *n*. Figure 3.2 illustrates the movement pattern Random Walk Model with a single mobile terminal.

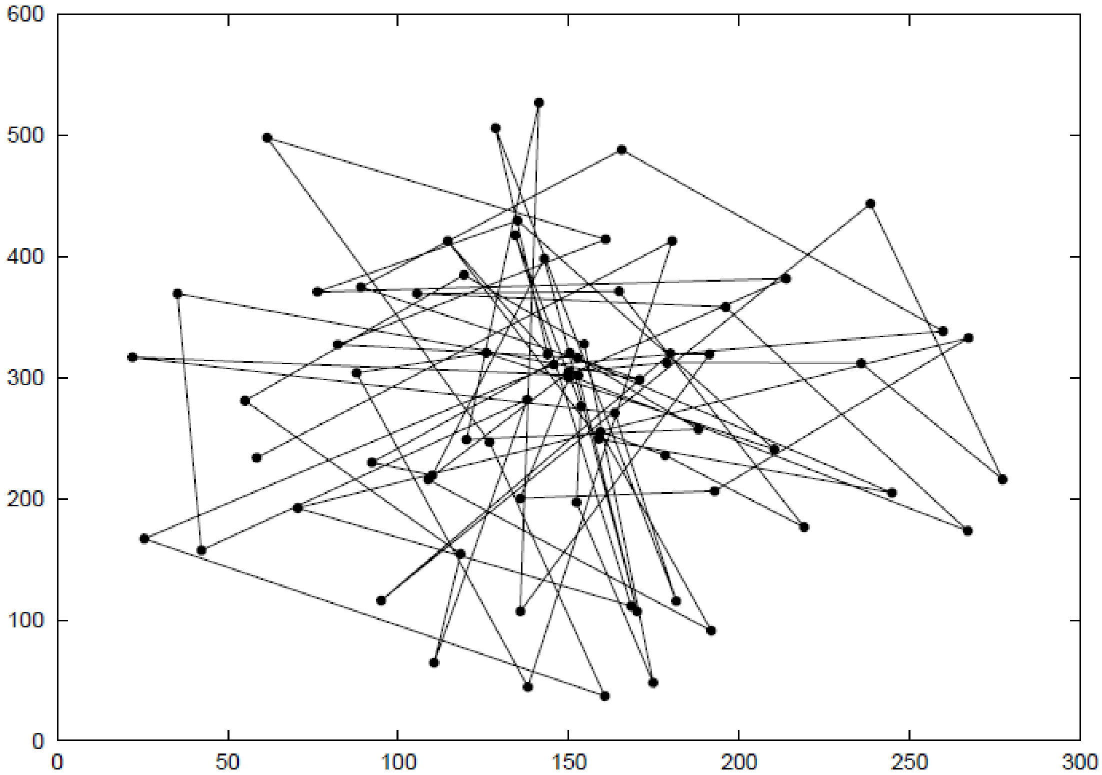


Figure 3.2 Movement pattern of a mobile terminal in the Random Walk Model

(Figure 1 from [CBD02])

In this research, the walking mechanism of the mobile terminal in the mobility model derives from the Random Walk Model. At each time interval, the mobile terminal needs to choose a direction and a speed, which are used together with the current coordinates to determine its location at the next time interval.

3.2.2 Gauss-Markov Model

The Gauss-Markov Model was first proposed by Liang and Haas in [LH99]. The main aim of this model is to introduce different levels of randomness for the movement of mobile terminals. Similar to the Random Walk Model, the movement direction and speed are chosen by mobile terminals at each time interval to determine the movement trajectory in the following time interval. Instead of using uniformly distributed direction and speed, these two walking parameters at time *n* are determined using the following equations.

$$\theta_n = \alpha\theta_{n-1} + (1-\alpha)\bar{\theta} + \sqrt{(1-\alpha^2)}\theta_{x_{n-1}}$$
$$v_n = \alpha v_{n-1} + (1-\alpha)\bar{v} + \sqrt{(1-\alpha^2)}v_{x_{n-1}} \tag{3.2}$$

where θ_n and v_n are the direction and speed determined by the mobile terminal at time interval n; $\alpha(0 < \alpha < 1)$ is the randomness coefficient that is used to vary the randomness of θ_n and v_n; $\bar{\theta}$ and $\bar{v}$ are two constants representing the mean value of speed and direction as $n \to \infty$; and $\theta_{x_{n-1}}$ and $v_{x_{n-1}}$ are random variables from the Gaussian distributions with mean values $\bar{\theta}$ and $\bar{v}$ respectively. When $\alpha = 1$, the values of θ_n and v_n are deterministic and equals to the values of θ and v in previous time intervals; When $\alpha = 0$, the values of θ_n and v_n are completely random and follow the Gaussian distribution. When $0 < \alpha < 1$, the intermediate level of randomness can be obtained. Figure 3.3 illustrates the movement pattern of the Gauss-Markov Model with a single mobile terminal.

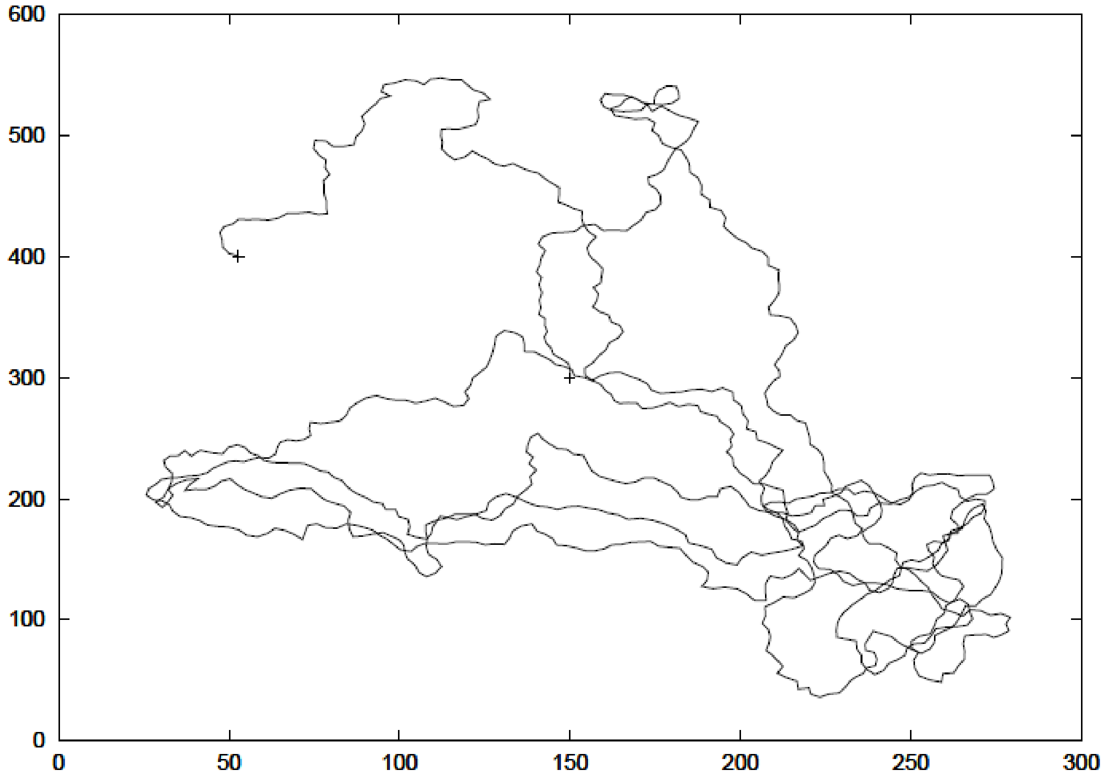


Figure 3.3 Movement pattern of a mobile terminal in the Gauss-Markov Model

(Figure 10 from [CBD02])

In this research, the method of determining the walking parameters θ and v derives from the Gauss-Markov Model. Intermediate level of randomness is introduced when determine the movement speed; θ is chosen according to the road topology.

3.2.3 Nomadic Community Model

In real life, the movement of each pedestrian could be independent if the density of pedestrians is low in the observed area. Along with the increase of pedestrian density, the possibility of interactions among different pedestrians increases as well. In an area with high pedestrian density, the movement speed and direction decision made by an individual might be greatly affected by the nearby pedestrians. The movement speed and direction of an individual pedestrian might be very close to the average speed of the pedestrian cluster that the individual belongs to. For vehicles, interference from nearby vehicles always exists even when the density of vehicles is not high, due to safety considerations. Thus, with the purpose of investigating the influence of a group of mobile users, the synthetic group mobility models are introduced into the mobility management field.

As the name implies, the Nomadic Community Model is applied to mimic the movement of groups of mobile terminals from one point to another [CBD02]. In the Nomadic Community Model, each mobile terminal uses an entity mobility model to roam around a given reference point. When the reference point starts moving, all the terminals in this group move following the trajectory of the reference point. Figure 3.4 illustrates the movement pattern of the Nomadic Community Model with seven mobile terminals and one reference point.

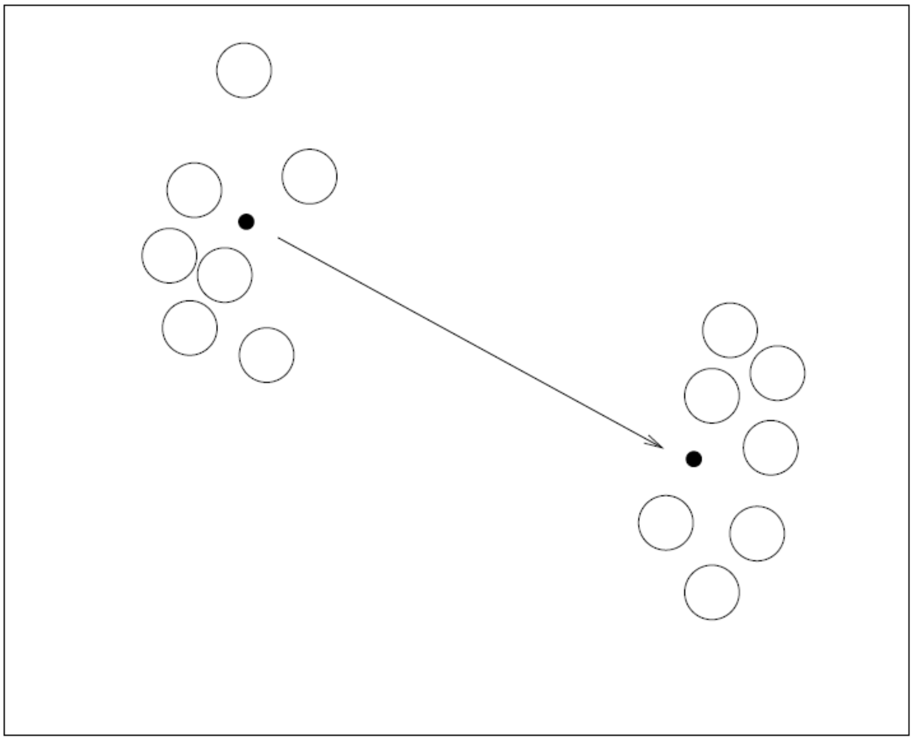


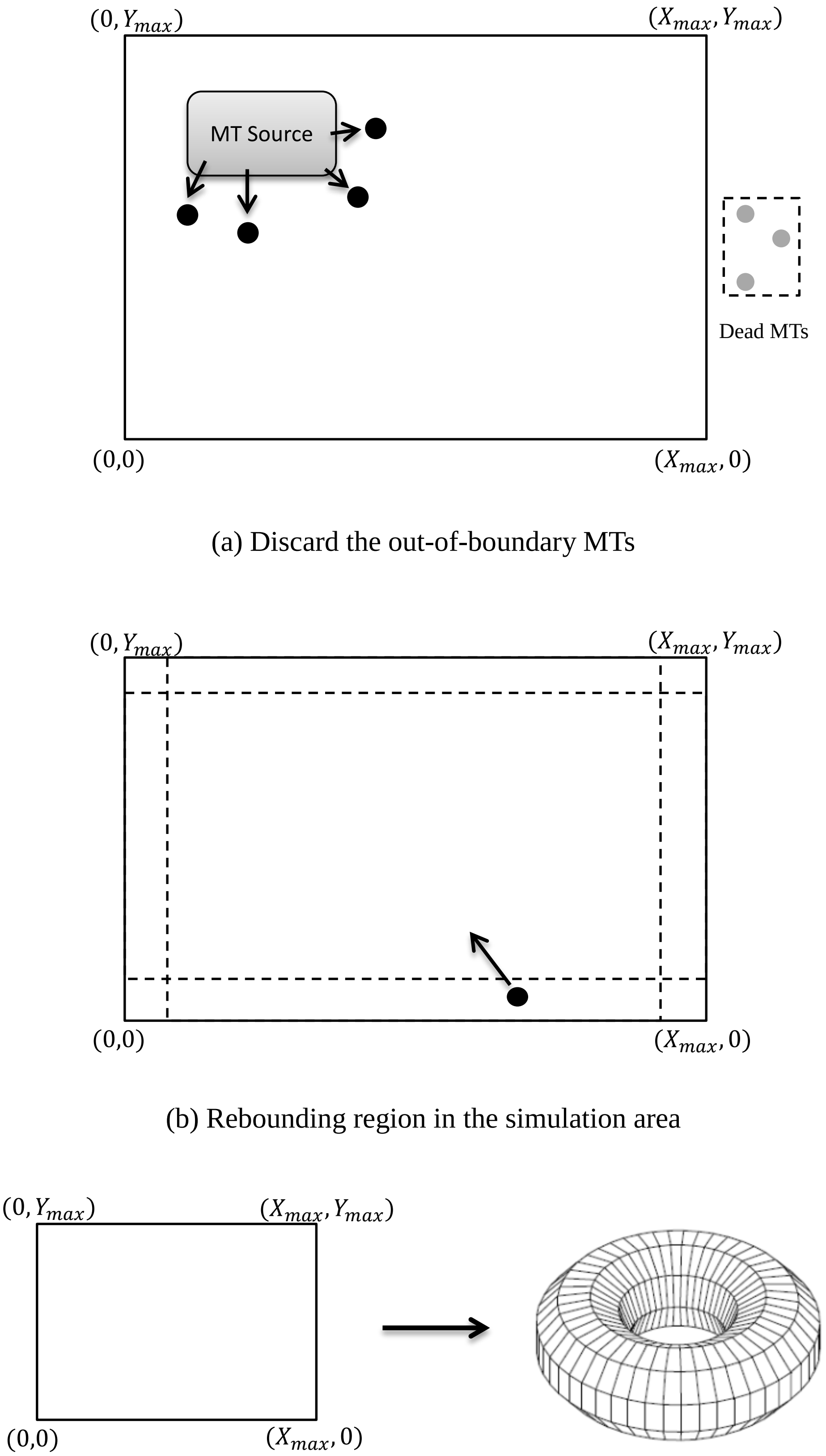
Figure 3.4 Movement pattern of a group of users in Nomadic Community Model (Figure 16 from [CBD02])

In this research, one main task is to learn the local traffic pattern and make predictions on the future geographic distribution of mobile clusters. The concept of the Nomadic Community Model is applied to guide the movement of mobile clusters in [Yao07]. In real life, people in a moving cluster can hardly maintain similar movement patterns for a long time, but disperse during the movement according to the geographic layout. Thus, a more realistic walking rule is applied for the moving clusters in this research, which will be presented in the next chapter.

3.2.4 Boundless Simulation Area

When simulating the movement of mobile users with mobility models, one circumstance that mobile users face is the possibility of moving out of the boundary of the simulation area which might happen quite often. Thus, it is necessary to extend to moving rule to include the solutions for out-of-boundary situations. Three solutions that are widely used in mobility models are discussed next.

The first solution is to discard those mobile users that are no longer in the simulation area and add new mobile users into the simulation area after each ΔT or the total number of the users is less than a threshold value, as shown in Figure 3.5(a). The second solution is to prevent the mobile users getting too close to the boundary. When the distance between one mobile user and the boundary is shorter than a predefine threshold value, the mobile terminal will be forced to move towards the centre of the simulation area, as shown in Figure 3.5(b). The third solution is to treat the simulation area as a boundless plane. Mobile users that reach one side of the simulation area continue travelling and reappear on the opposite side of the simulation area. This solution creates a torus-shaped simulation area allowing the mobile users to travel unobstructed, as shown in Figure 3.5(c).



(a) Discard the out-of-boundary MTs

(b) Rebounding region in the simulation area

(c) Boundless simulation area

Figure 3.5 Three boundary rules for mobility models

In this research, one key task is to learn the local traffic pattern so as to predict the future geographic distribution of mobile users. The mobility model used in this research is required to reflect real-life traffic patterns. For the first two boundary handling solutions, movements in areas with the Mobile Terminal (MT) source or in the rebounding region are deterministic and unrealistic. With the boundless simulation area solution, the total number of the mobile terminals can be kept constant; and the predefined movement rules are not affected when a mobile terminal gets close to the boundary. Thus, the concept of the boundless simulation area is applied when building the mobility model in this research. Equations below are used when a mobile user is out of the boundary at time interval *n*.

$$
\begin{aligned}
x_n &= \begin{cases} x_n + (x_{max} - x_{min}), x_n < x_{min} \\ x_n - (x_{max} - x_{min}), x_n > x_{max} \end{cases} \\
y_n &= \begin{cases} y_n + (y_{max} - y_{min}), y_n < y_{min} \\ y_n - (y_{max} - y_{min}), y_n > y_{max} \end{cases}
\end{aligned}
\tag{3.3}
$$

where x_{min}, x_{max}, y_{min} and y_{max} are the boundary values of the simulation area in the Cartesian coordinate system.

3.3 The SMP Mobility Model

In this section, the Single-Movement-Pattern (SMP) mobility model used in this research is presented. The main goal of this research is to learn the local traffic pattern with historical intra-cell transition data and make predictions on the future local distribution of mobile users. The ideal historical data for this research are the daily movement trajectories observed by base stations in real life. These trajectories provide accurate information that can reflect the behaviour habits of users and the geographic layout of the observed area. However, collecting a large set of mobile users' movement trajectories is very time consuming and expensive. For the existing mobility models, they are either generating completely random trajectories as shown in [AWS06] or generating probabilistic cell transition history as shown in [AHMES09] and [SMBS10]. Thus, the SMP mobility models are designed and used in this research to generate comparative realistic movement trajectories of mobile users for local traffic pattern learning. One novelty of the proposed SMP model is that factors influencing the movement patterns in real life, such as the road topology, the traffic condition, the road throughput capability, are introduced when building the mobility model.

In order to enable semi-smart antennas to optimise their beamforming patterns with load balancing schemes, such as the bubble oscillation algorithm and the CBR matching algorithm, for upcoming traffic, both the inter-cell traffic patterns and the intra-cell traffic patterns need to be presented in the prediction results of the mobile traffic. The inter-cell traffic pattern can be simply indicated by the number of mobile users in the coverage area of the target cell. On the other hand, the intra-cell traffic pattern needs to show the geographic distribution of mobile terminals so as to let the semi-smart antenna know which part of its coverage area is heavily loaded. Thus, another novelty of the SMP model is that it generates both the inter-cell transition data and the intra-cell movement trajectories.

3.3.1 Random Walking Point

In real life, most movements are not completely random but with explicit intent. Those real-life movements also need to be restricted by road topology and traffic rules. If there is no sudden stop during the movement, the movement trajectory of a mobile user can be considered as a continuous and smooth curve. If the moving duration is divided into *n* small time intervals, the movement trajectory then can be decomposed into *n* sub-curves, which can be treated as short lines when the time interval is small. This approximation is shown in Figure 3.6.

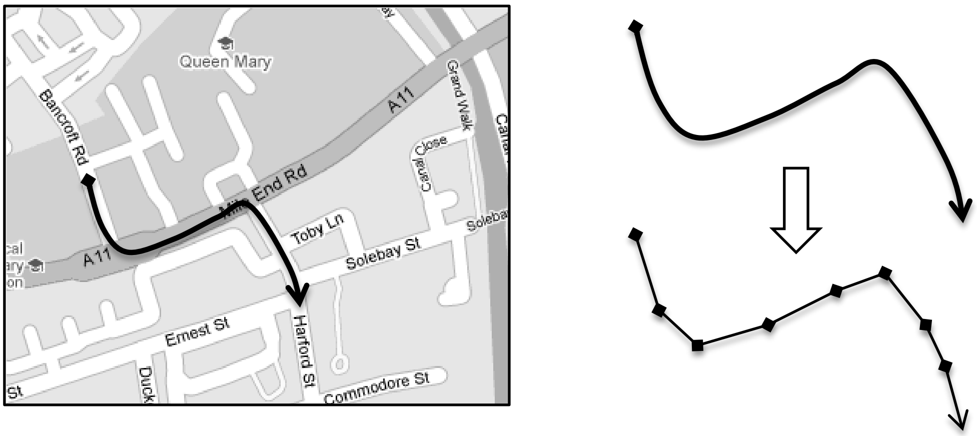


Figure 3.6 User trajectory approximation

With this approximation, each mobile user could be treated as a moving point with constant speed and direction in each time interval ΔT. Thus, the Random Walking Point (RWP) is introduced in this research to mimic the moving mobile user in real life.

The RWP is defined as a continuously moving point that randomly determines its location at every step according to the predefined walking rules. The term “step”, with the similar meaning of “time interval” mentioned in other research, is used in this research. The time interval ΔT between two steps is a constant. At each step, the coordinates of all the RWPs in the simulation area are captured as a traffic snapshot to present the instantaneous geographic distribution of the mobile users. Besides, a number of consecutive coordinates of one RWP can be used to generate the movement trajectory to reflect the user’s behaviour habits and the local area topology.

In this research, a pair of variables, v and θ, are introduced and named as walking parameters for the RWP. At each step, the RWP generates the walking parameter values to determine its coordinates for the next step. As illustrated in Figure 3.7, v describes the walking speed and θ indicates the walking direction. In this simulator, Cartesian coordinates are used to represent and calculate the position of RWPs. The new coordinates of the RWP can be obtained by using the following equations:

$$\begin{cases} x_{(i+1)j} = x_{ij} + v_{ij} \cdot \cos\theta_{ij} \\ y_{(i+1)j} = y_{ij} + v_{ij} \cdot \sin\theta_{ij} \end{cases} \tag{3.4}$$

where $P_j\left(x_{ij}, y_{ij}\right)$ are the coordinates of RWP P_j at step i; $P_j'\left(x_{(i+1)j}, y_{(i+1)j}\right)$ are the next coordinates of P_j at step $(i+1)$. v_{ij} and θ_{ij} are the walking parameters generated by P_j at step i.

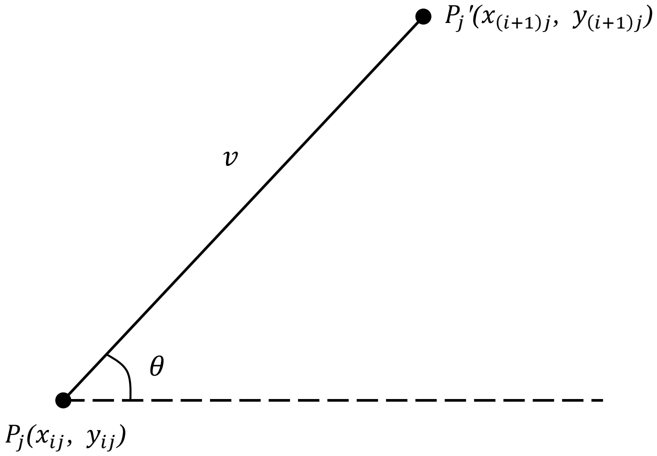


Figure 3.7 Walking parameters of the RWP

In order to reduce the modelling complexity, all the RWPs in this research are treated as activated users and there is no “OFF” state for the RWP in the simulation.

3.3.2 Simulation area

In order to let the RWPs move in a realistic way, a simulation area with real-life geographic features is needed to guide to movements of RWPs. In real life, the possible movement directions of mobile users in a specific area are restricted by the local road topology. Although the intents of moving mobile users are not the same, their movement trajectories in the same local area should follow a limited number of movement patterns, which are restricted by the local geographic layout. Thus, in order to generate realistic traffic flows, local geographic details are introduced in the simulation area used in this research.

In real life, people could always use “east”, “west”, “north”, “south”, “northeast”, “northwest”, “southeast” and “southwest” to describe the current direction they are moving towards. If a specific area is divided into a number of square-shaped grids, then directions all the roads in a grid could lead to are the directions from the current grid to its eight neighbours, as shown in Figure 3.8.

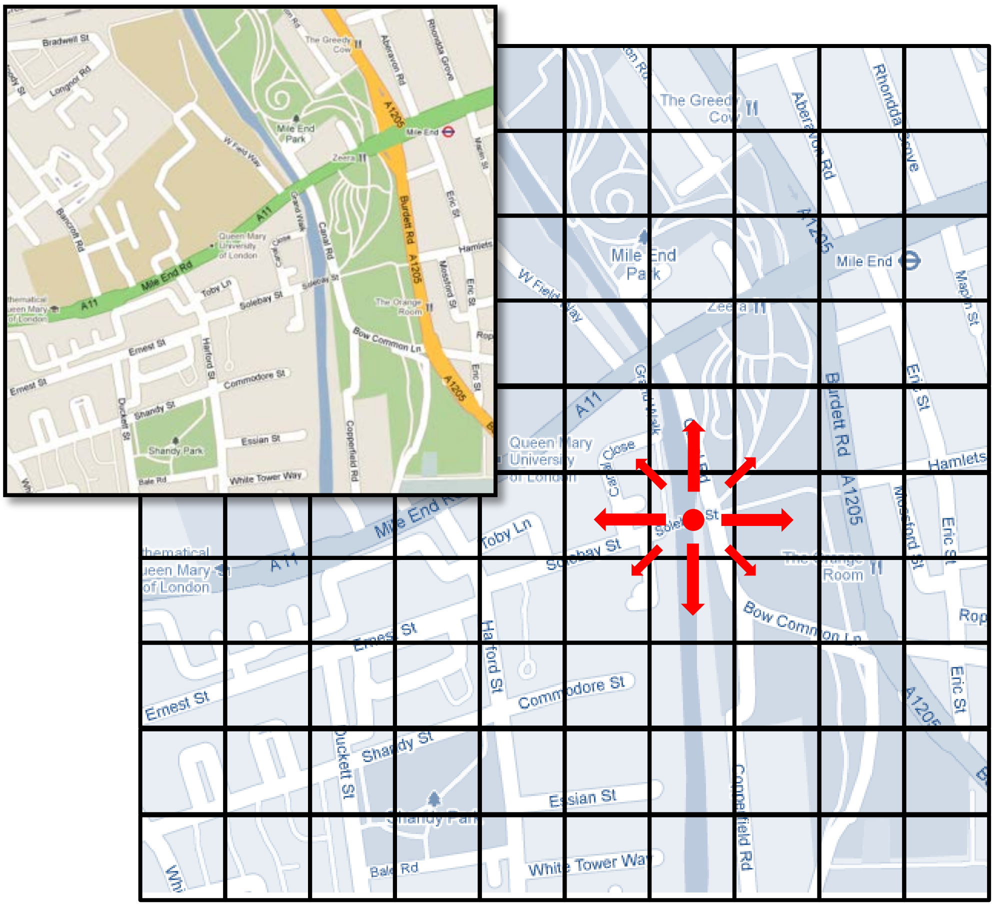
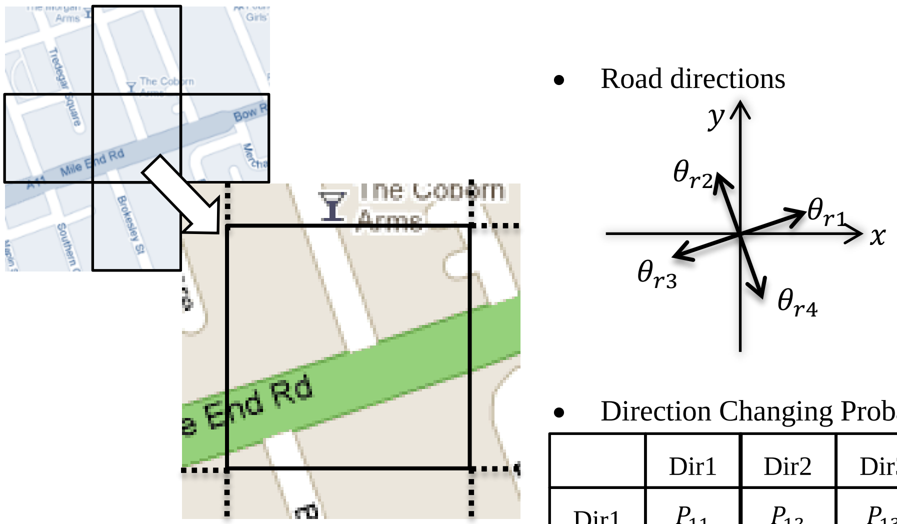


Figure 3.8 Geographic grid elements in the simulation area

With the gridded areas as shown in Figure 3.8, the local geographic layout in one grid square could be described as the main directions of the traffic flow, which reflect the directions of the main roads in that area; and the possibilities transiting from the current gird square to its neighbours, which reflect the local movement habits and preferences. Therefore, the grids, as shown in Figure 3.8, are defined as geographic grid elements (GGEs) in this research, which are used to build the simulation area and to provide local geographic details. In order to simplify the simulation, only four directions, “east”, “west”, “north” and “south”, are taken into account in this research. Figure 3.9 shows an example of the geographic grid element.



- Road directions

- Direction Changing Probability

	Dir1	Dir2	Dir3	Dir4
Dir1	P_{11}	P_{12}	P_{13}	P_{14}
Dir2	P_{21}	P_{22}	P_{23}	P_{24}
Dir3	P_{31}	P_{32}	P_{33}	P_{34}
Dir4	P_{41}	P_{42}	P_{43}	P_{44}

- Road throughputs

Dir	1	2	3	4
TP	TP_1	TP_2	TP_3	TP_4

Figure 3.9 An example of the geographic grid element

As shown in Figure 3.9, the geographic details carried by a GGE can be classified into the following types:

- Road directions
- Transition matrix for direction changing
- Road throughput capabilities

Road directions $\theta_i (i = 1, 2, 3, 4)$ are used to indicate the directions of the main roads that restrict the local traffic flow in a GGE. As mentioned above, four main directions are taken into account in this research, “θ_{r1} - East", “θ_{r2} - North", “θ_{r3} - West" and

“θ_{r4} - South". In the rest of the thesis, direction 1, direction 2, direction 3 and direction 4 are mapped to “East”, “North”, “West” and “South”, respectively. The value ranges are$\theta_{r1}[\frac{-\pi}{4},\frac{\pi}{4})$, $\theta_{r2}[\frac{\pi}{4},\frac{3\pi}{4})$, $\theta_{r3}[\frac{3\pi}{4},\frac{5\pi}{4})$ and $\theta_{r4}[\frac{5\pi}{4},\frac{7\pi}{4})$. When initiating the road topology, each GGE randomly generates the four local traffic flow directions according to the value range above for each direction. Then these directions are used to restrict the movement pattern of MTs in the local area.

The probability P_{ij} $(i,j = 1,2,3,4)$ in the transition matrix for direction changing is used to indicate the inclination of mobile users when travelling in a local area from a specific direction. i is the direction the mobile user comes from in the previous step; j is the direction this user moves towards in the current step. Notice that the movement inclination mentioned here is duo to the features of the geographic layout. For instance, pedestrians are inclined to turn to the direction of the bus stop or the tube station at a fork during rush hour. When initiating the road topology, the probabilities in the transition matrix are generated randomly or based on the predefined local geographic layout. Then this matrix needs to be taken into account by MTs when making decisions for the next movement direction.

Road throughput capabilities $TP_i(i = 1,2,3,4)$ are used to indicate the traffic saturation threshold value of each direction. If the number of MTs moving toward direction i is larger than the predefined TP_i, then the movement speed of those MTs will be affected by the traffic saturation status, and the movement speed of an individual MT approaches the average speed in the saturated direction. When initiating the road topology, the road throughput capabilities are generated randomly or based on the predefined local geographic layout.

In this research, the simulation area is composed of a number of GGEs. As introduced above, the main intent of using the GGE is to imitate the influences on the moving mobile users from the local geographic layout in real life. Therefore, the movement patterns generated by RWPs will be more realistic rather than random moving or moving straight forward from the initial position to the predefined destination. Another benefit of using GGEs to build the simulation area is that the whole layout of the simulation area can be easily adjusted to create different scenarios only by adjusting the local layouts of the relevant GGEs.

3.3.3 Walking rules

In this research, the historical movement trajectories of mobile users are generated by the mobility model for movement pattern learning sake. As discussed in the previous two sections, the RWPs are used to mimic the moving users and the GGEs are used to build the simulation area. Then, RWPs are distributed in the simulation area and start to move according to the predefined walking rules. This section presents the walking rules applied for the RWPs walking over each GGE.

At each step, RWPs generate the walking parameters θ and v according to the information provided by the GGE they visit. When determining the value of θ in each step, the first decision a RWP needs to make is which direction it will move toward in this step. The probability of direction changing is calculated based on the transition matrix in the visited GGE. Notice that each RWP has a limited number of chances to change its direction in one visit of the same GGE, since it is unrealistic to keep changing movement direction in the same area in real life. When the movement direction is decided, the base value of the movement direction θ_{base} is fixed, as θ_{base} is the same as the relevant road direction θ_{ri} of the visited GGE. In order to introduce the randomness factor into each movement, a random bias θ_{bias} is generated followed the Gaussian distribution and added to θ_{base} to determine the value of θ in this step. The generating rules of θ can be described as follows:

$$\theta_{base} \in (\theta_{r1}, \theta_{r2}, \theta_{r3}, \theta_{r4})$$
$$\theta_{bias} = rand\_normal(\mu_{bias}, \sigma_{bias}, \theta_{bias\_min}, \theta_{bias\_max}) \tag{3.5}$$
$$\theta = \theta_{base} + \omega \cdot \theta_{bias}$$

where $rand\_normal(w, x, y, z)$ is the function that generates a random number that follows the normal distribution $N(w, x^2)$ within the boundary $[y, z]$. $\omega (0 < \omega < 1)$ is the randomness level coefficient that is used to vary the randomness level of θ_{bias}. As discussed above, the value range of θ_{ri} is $[\frac{-\pi}{2}, \frac{7\pi}{2})$. This value range is also applicable for θ. Besides, θ_{bias_min} and θ_{bias_max} are set to $\frac{-1}{6}\pi$ and $\frac{1}{6}\pi$ respectively in this model to introduce a small variation of θ_{base}.

The movement speed v is determined after the RWP knows which direction it travels in the current step. When initiating the RWP at the beginning of the simulation, two parameters v_{normal} and v_{max} are preconfigured. v_{normal} can be considered as the recommended movement speed for the RWP; v_{max} is the maximum value of the movement speed the RWP can reach. If there is no internal and external influence, the RWP is supposed to move with speed v_{normal}. However, due to the influences from the external environment and the random internal impacts, the actual movement speed v_i might vary in each step. The generating rules of v_i at step i can be described as follows:

$$\alpha = rand\_uniform(v_{bias\_min}, v_{bias\_max})$$
$$\beta = \mathcal{F}(\rho_{r_i}) \tag{3.6}$$
$$v_i = (\omega \cdot v_{i-1} + (1 - \omega) v_{normal}) \cdot \alpha \cdot \beta$$
$$v_i = argmin(v_i, v_{max})$$

where $rand\_uniform(x, y)$ is the function that generates a random number follows the uniform distribution between x and y. $\mathcal{F}(x)$ is the function that determines the congestion coefficient according to the number of users facing the relevant direction in a GGE. $\rho_{r_i}(i = 1, 2, 3, 4)$ indicates the number of uses with the same movement direction in a GGE. $\alpha\ (0.9 < \alpha < 1.1)$ is the internal random influence factor, which represents the speed variation caused by the RWP itself. $\beta\ (0 < \beta < 1)$ is the traffic congestion factor, which represents the external influence from the environment and other RWPs. $\omega(0 < \omega < 1)$ controls the extent of the dependence between the movement speed v_i at step i and the movement speed v_{i-1} at step $i-1$. $argmin(x, y)$ is the function that retrieving the argument with the smaller value. Figure 3.10 shows an example of the simulation results of the SMP model.

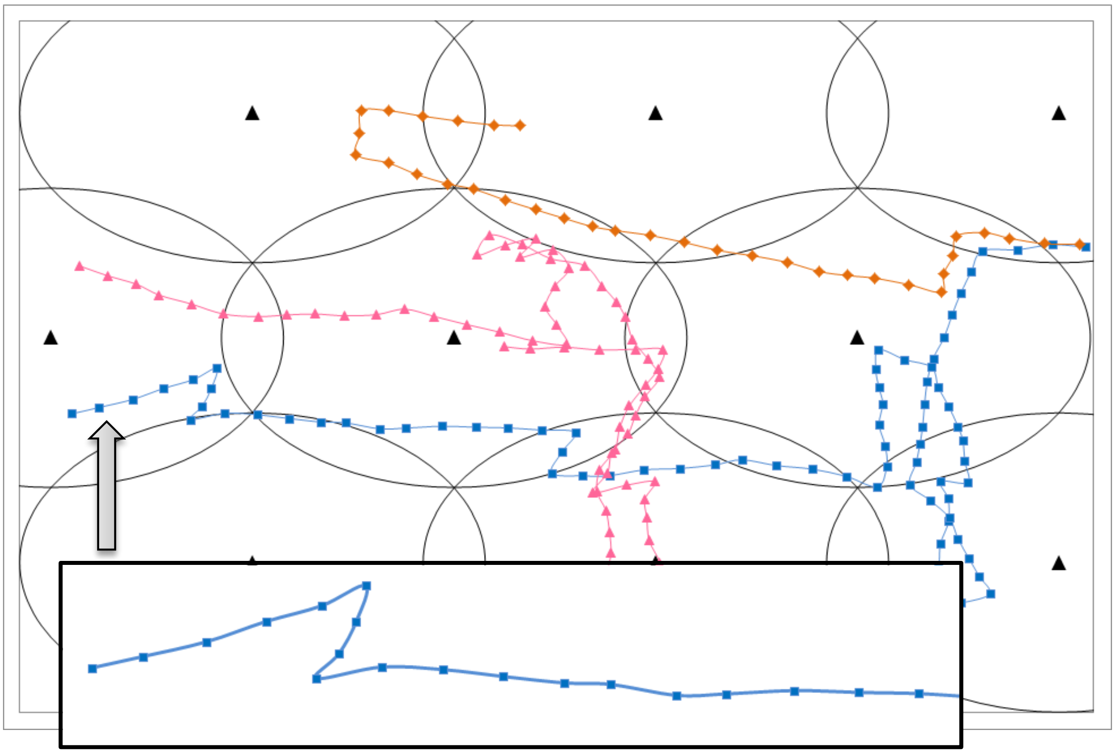


Figure 3.10 An example of movement trajectories

3.4 Traffic Pattern Learning Methods

The traffic pattern learning methods involved in this research are inspired from the approaches proposed in the mobility prediction field. As mentioned in the previous chapter, the main task in the mobility prediction field is to figure out the next possible cells a mobile terminal might visit. The simulation area used in most of the research is divided into hexangular grids to mimic the radio cells. Thus, the movement history of a mobile terminal could be presented as a sequence of cell IDs from a finite set.

After referring to the methods used in the mobility prediction field, the learning grids (LG) are introduced in this research to capture the local movement characteristics and provide information for prediction use. The learning grid is defined as a square-shaped grid that is much smaller than the radio cell. The number of the learning grid contained in the coverage area of a conventional radio cell is a constant. Each learning grid is given a unique ID, so the movement trajectory of a mobile terminal within a cell can also be presented as a sequence of IDs from a finite set. The two CP approaches introduced in this section are both based on the usage of a learning grid.

3.4.1 The Order-k Markov learning model

In the mobility prediction field, both [HKA04] and [SKJH04] show that simple Markovian models perform nearly as well as other more complex methods when learning the movement features the motion trajectories captured in the Wi-Fi networks. Thus, the Order-*k* Markov learning model with a self-degradation function is used in this research. As the existing learning models proposed for predicting the intra-cell user distribution are rare, the prediction performance of the Order-*k* Markov learning model is used as a baseline when carrying out the results evaluation.

A Markov chain is a sequence of random variables X_1, X_2, X_3,… with the Markov property. Having the Markov property means that, given the present state, future states and past states are independent. A Markov chain of order *k* means that the next state only depends on the last *k* states. The order-*k* Markov chain can be represented by the following equation:

$$P(X_{n+1} = x \mid X_n = x_n, \dots, X_2 = x_2, X_1 = x_1) = \\ P(X_{n+1} = x \mid X_n = x_n, \dots, X_{n-k+1} = x_{n-k+1}) \tag{3.7}$$

where $P(A|B)$ denotes the conditional probability of A given B; x_i $(x_i \in \vartheta)$ denotes the value at state X_i. Moreover, a stochastic process is said to be stationary, if the following equation is satisfied for every shift s and for all $x_i \in \vartheta$.

$$P(X_n = x_n, \dots, X_2 = x_2, X_1 = x_1) = \\ P(X_{n+s} = x_n, \dots, X_{2+s} = x_2, X_{1+s} = x_1) \tag{3.8}$$

It is reasonable to believe that there are relations between the next area a mobile user would visit and the sequence of k areas this user visited most recently because the movements of mobile users are always constrained by road topologies and might have a common purpose. Thus, the sequence composed by the next location and the previous k locations an RWP visited has the Markov property and can be considered as order-k Markov chain. Moreover, similar mobility patterns always take place repeatedly in one area. If the temporal shift of the occurrence of the similar mobility patterns is treated as the shift s appearing as the increment of the subscript in Equation 3.8, then the occurrence of the k-length movement trajectory in a specific area (for instance in the same geographic grid element) can be considered a stationary process.

The main task of the Markov learning model is to predict a sequence of areas an RWP will visit according to its movement history. In order to capture the local movement characteristics, an Order-k Markov learning model is implemented based on the conclusion of Equation 3.7 and 3.8. As mentioned above, a number of learning grids are used to cover the simulation area. Since each learning grid has a unique ID, the short term motion trajectory of a mobile terminal can be represented by a series of learning grid IDs from a finite integer set.

At each step, the RWPs determine the IDs of the learning grids they are visiting. Thus, an RWP P_i should have a movement history set $L_i = \{l_0,\ l_1,\ l_2, \dots, l_{n-1},\ l_n\}$ at the end of step *n*, where l_0 is the learning grid ID related to P_i's initial position and l_n is the learning gird ID related to the coordinates of P_i at step *n*. Then, this RWP generates a "context, next location" pair $\{l_{n-k}, l_{n-k+1}, \dots, l_{n-1};\ l_n\}$ that will be sent to the Order-*k* Markov model that belongs to learning grid with l_{n-1} as its ID. The first *k* elements denote the *k* grids this user visited most recently; the last element denotes the grid visited after the context. The "context, next location" pairs generated by the RWPs are used to train the learning models. With the CP scheme, an Order-*k* Markov learning model can be trained quickly because it improves every time an RWP goes by.

Assume that L_n denotes the location of an RWP at step *n*; $L_{context} = \{L_n, L_{n-1}, \dots, L_{n-k+1}\}$ denotes the *k*-length location context used for prediction; $\mathbb{L}$ denotes the historical sample set of "context, next location" pairs the predictor has already recorded. It is worth noting that the length of each member in $\mathbb{L}$ is $k+1$. If one $L_{context}$ is seen and captured before by a learning model, this $L_{context}$ is the substring of at least one member of the $\mathbb{L}$ owned by that learning model. Then, the prediction rule of the Order-*k* Markov predictor can be represented by the following equation:

$$\hat{P}(L_{n+1} = l\ |L_{context}) = \frac{N((L_{context}, l);\ \mathbb{L})}{N(L_{context};\ \mathbb{L})} \tag{3.9}$$

where $N(s; \mathbb{S})$ denotes the number of times the string s occurs as a substring in the string set $\mathbb{S}$; (s_1, s_2) denotes a string with s_1 as its former part and s_2 as its latter part.

When making a prediction, an RWP sends a *k*-length context, including the last *k* learning grid IDs, to the local Order-*k* Markov model. The learning model calculates the probability of each possible *l* with equation 3.9. In this research, two different choosing rules are used to determine the next possible location. The first rule is that the predicted next location l_{pred} is chosen if it has the highest $\hat{P}(l_{pred}\,|L_{context})$; the second rule is that the l_{pred} is chosen with a probability of $\hat{P}(l_{pred}\,|L_{context})$. Only one of these rules is activated in each simulation. Notice that if the k-length context is seen for the first time, the first j IDs $(1 \leq j < k)$ will be discarded. Then, the Order-k Markov model will degenerate to an Order-$(k-j)$ Markov model and use the last $k-j$ elements in the context to make a prediction.

3.4.2 Static Traffic Pattern Learning model

In [YKUM04], a three-phase algorithm is proposed to carry out the mobility prediction. According to [YKUM04] and [WY02], traffic patterns hiding in the historical trajectories contain valuable information that could reveal the behaviour features of a target user or the movement features in a target cell. The authors use the first two phases to learn the inter-cell traffic patterns and to extract the mobility rules from the learnt patterns. Then these rules are used to make mobility prediction in the last phase.

For the inter-cell mobility prediction, the mobility rules reflect the probabilities of a target user moving from the current cell to its neighbours. When mining the intra-cell movement trajectories in this research, the aim is to discover the local geographic pattern and the movement habits of each local area covered by a base station. Inspired from the concept of learning mobility patterns from historical data in [YKUM04], this research proposes a Static Traffic Pattern Learning (STPL) model to handle the short-term traffic pattern prediction.

Notice that the real-life movement trajectories of mobile users are rarely random. When walking on the street, most people have an explicit destination rather than aimless roaming. In order to avoid obstacles, a movement trajectory can hardly be a straight line between the starting point and the end point, but be restricted by the geographic layout along the path. Besides, the routes chosen by different users should be similar in the same area, as the movement patterns in a specific area are limited. Thus, this research uses the STPL model to learn the traffic patterns from the following aspects: local geographic layout, local behaviour habit, and short-term movement speed variation.

Figure 3.11 shows movement trajectories of moving mobile users in the coverage area of a semi-smart antenna. In real life, the coverage radius of a BS in urban area is usually between 500 metres and 1500 metres. Although the movement patterns in the whole coverage area cannot be unique, trajectories in the same local area should follow the similar patterns, as those trajectories are affected by the same local features. Thus, a number of square-shaped learning grids are deployed into the target area, and each learning grid applies the STPL model locally to learn the local traffic patterns, as shown in Figure 3.12.

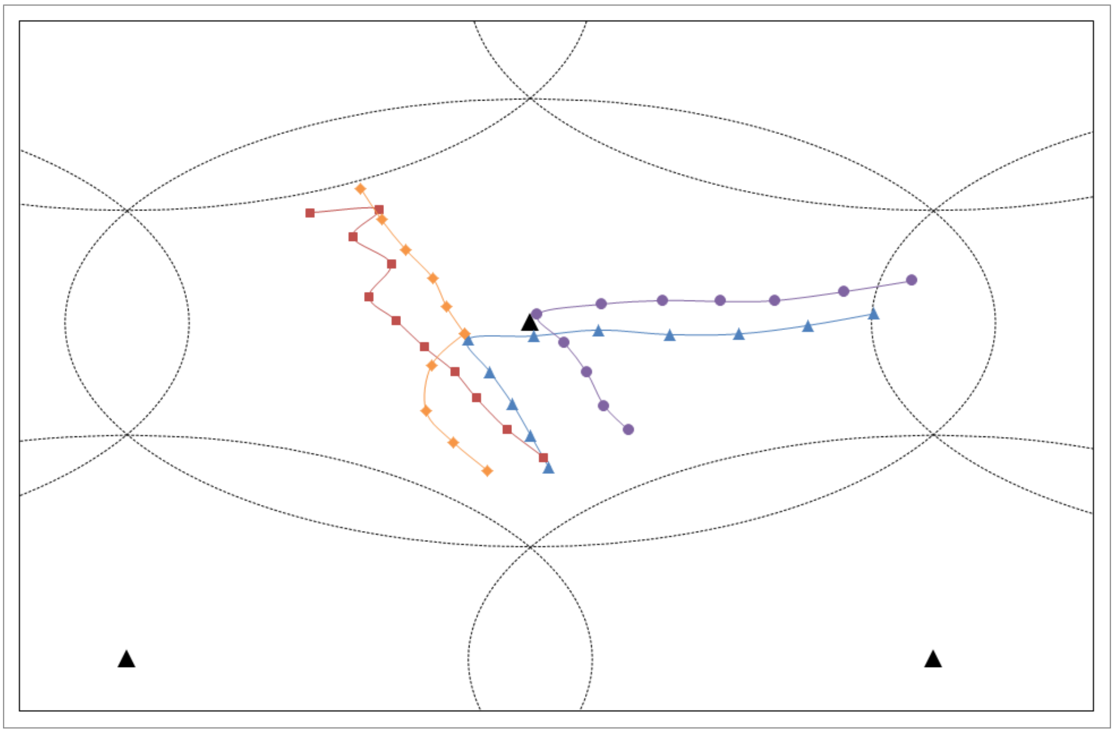


Figure 3.11 Intra-cell movement trajectories of mobile terminals

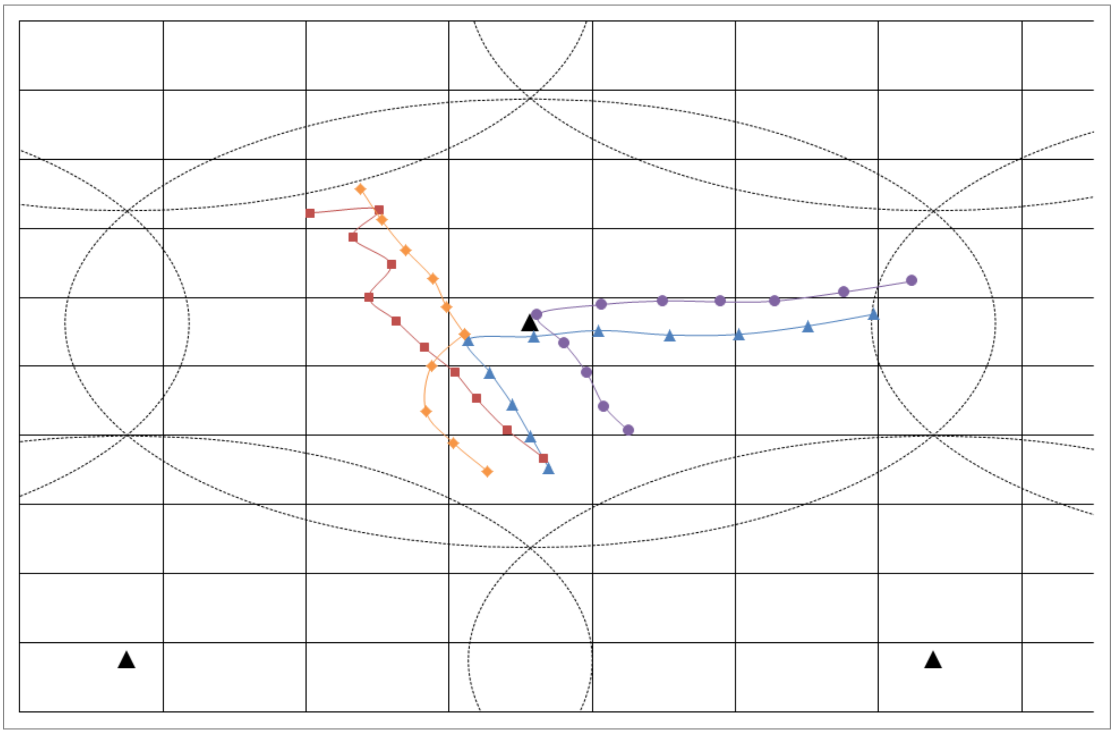


Figure 3.12 Learning grids deployed in the simulation area

When processing the historical movement data, the learning grid records each movement trajectory that goes through its coverage area as a context sequence of coordinates $C_{context} = \{C_{from}, C_1, C_2, \dots, C_n, C_{to}\}\ (n \geq 1)$, where C_{from} is the last observed coordinates before a user enters the coverage area of the learning grid; C_{to} is the first observed coordinates before a user leaves the coverage area of the learning grid; $C_1, C_2, \dots, C_n$ are n pairs of coordinates consecutively observed within the coverage area of the learning grid, as shown in Figure 3.13. Then, all the context sequences recorded by the learning grid are used to learn the local traffic pattern.

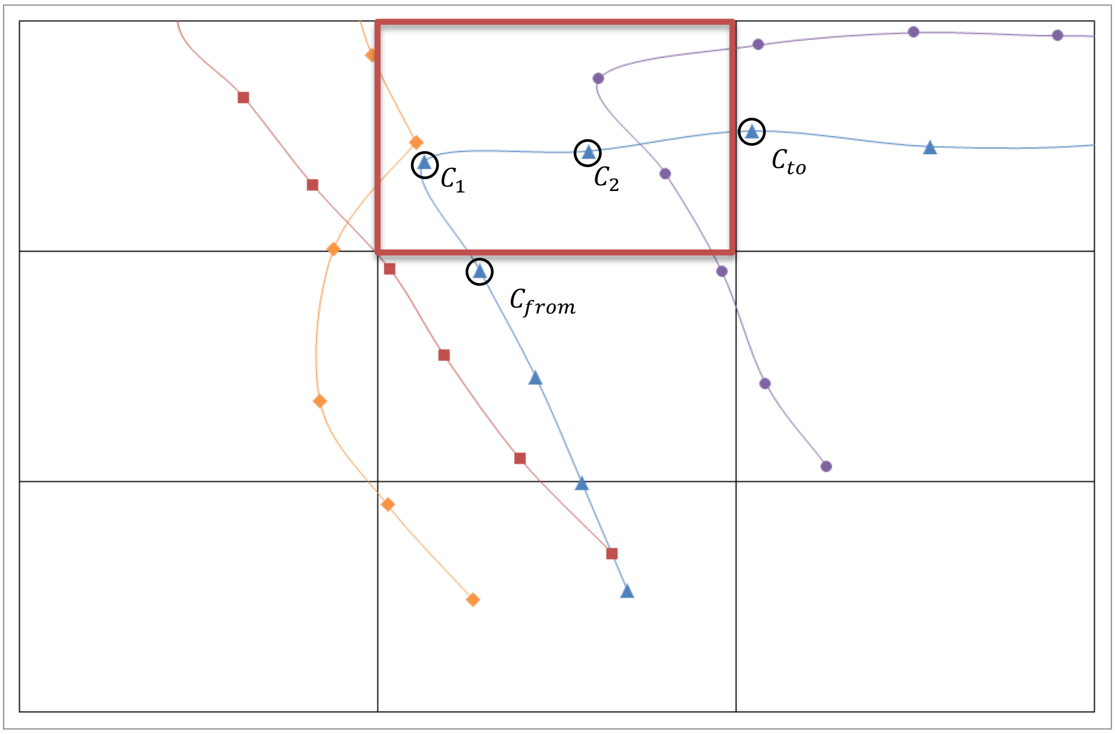


Figure 3.13 An example of the context sequence

When learning the local traffic patterns, the first task is to learn the local geographic layout. In this research, the main movement directions in the target area are treated as the direct reflection of the main roads' directions, which reveals the local geographic layout. At the beginning of the pattern learning phase, an integer number k is designated to denote the number of main directions. Then the 2π radians are averagely divided into k sectors, as shown in Figure 3.14. The sector i indicates the ith main direction with the value range $\left[\frac{(2i-3)\pi}{k}, \frac{(2i-1)\pi}{k}\right)$ $(i = 1, 2, 3, \dots, k)$.

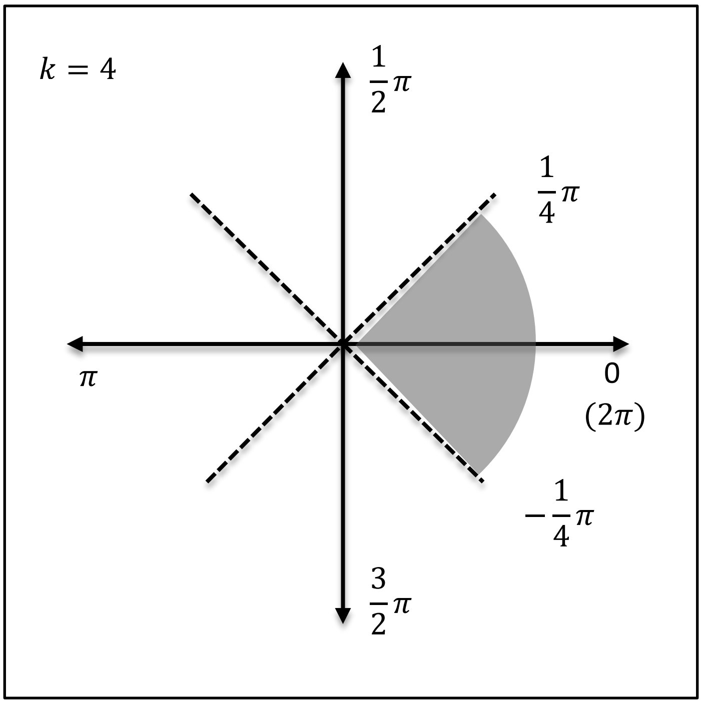


Figure 3.14 Dividing 2π radians into *k* main directions

For a context sequence $C_{context} = \{C_{from}, C_1, C_2, \dots, C_n, C_{to}\}$ $(n \geq 1)$, $n > 1$ means the observed user moves within the coverage area of the target learning grid for more than 1 time interval of the observation. $n = 1$ means the observed user moves fast in this area, only one observation captures its presence in this area. Figure 3.15 shows the above two circumstances.

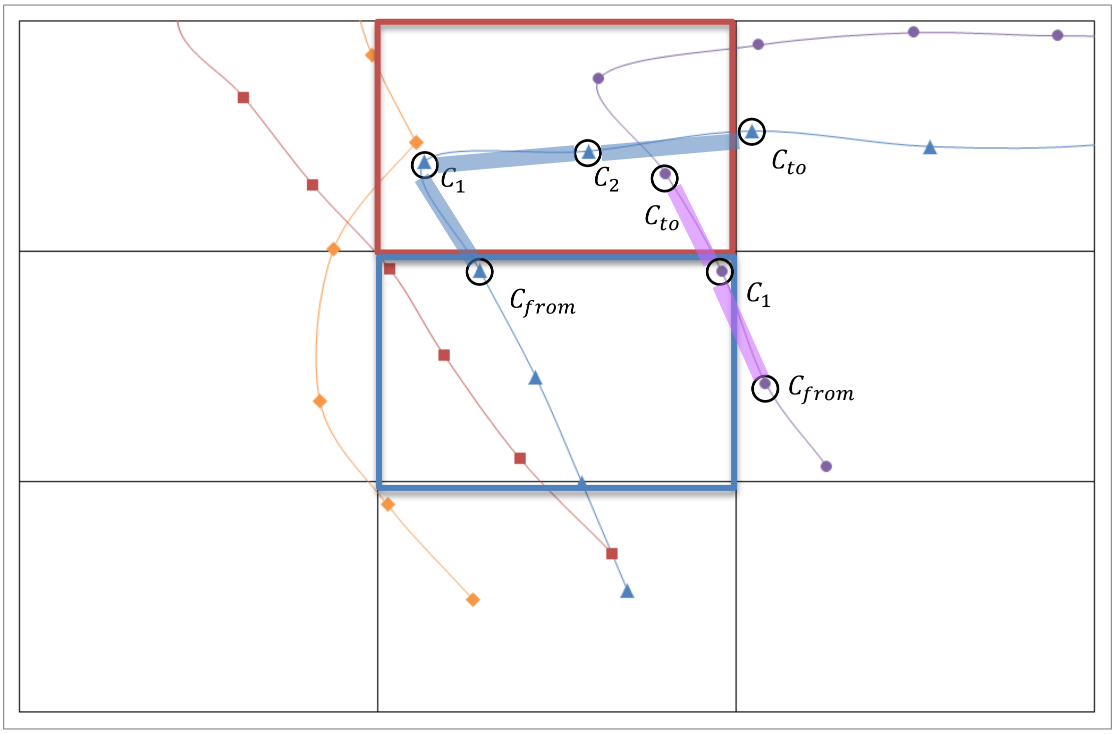


Figure 3.15 Examples of the context sequence with different length

In order to extract the sample values of the movement directions, the learning grid filters out all the context sequences with $(n \geq 1)$, and calculates the direction sample values θ with all $(C_i(x_i, y_i), C_{i+1}(x_{i+1},, y_{i+1}))$ pairs extracted from the context sequences by using the following equations.

$$
\begin{aligned}
s_{opp} &= y_{i+1} - y_i \\
s_{adj} &= x_{i+1} - x_i \\
\theta &= \mathcal{F}_{arctan}(s_{opp}, s_{adj}, k)
\end{aligned}
\tag{3.10}
$$

where s_{opp} and s_{adj} denote the length of opposite side and the length of the adjacent side respectively. Notice that the value of s_{opp} and s_{adj} can be negative. $\mathcal{F}_{arctan}(a, b, k)$ is the function to calculate the arctangent value of $\frac{a}{b}$ with the value range $[-\frac{\pi}{k}, \frac{(2k-1)\pi}{k})$. k is an integer that indicates the number of main directions taken into account in the learning phase.

Assume that $\Theta_i(i = 1, 2, \ldots, k)$ denotes the θ sample set for the *i*th interval with value range $\left[\frac{(2i-3)\pi}{k}, \frac{(2i-1)\pi}{k}\right)$ $(i = 1, 2, 3, \ldots, k)$. After each calculation with equation 3.10, the direction sample θ $(\frac{(2j-3)\pi}{k} \leq \theta < \frac{(2j-1)\pi}{k})$ is added to the sample set $\Theta_j(1 \leq j \leq k)$. With the purpose of simulating a realistic environment, the population of θ in each sample set is assumed to follow a normal distribution. Assume that the θ samples recorded in Θ_i are $\Theta_i(\theta_{i1}, \theta_{i2}, \ldots, \theta_{in})$. By assuming $\Theta_i \sim N(\mu_{\theta i}, \sigma_{\theta i}^2)$, the estimated value of $\mu_{\theta i}$ and $\sigma_{\theta i}^2$ are:

$$\begin{cases} \hat{\mu}_{\theta i} = \bar{\theta}_\iota = \sum_{j=1}^{n} \theta_{ij} / n \\ \hat{\sigma}_{\theta i}^2 = \frac{1}{n}\sum_{j=1}^{n} (\theta_{ij} - \bar{\theta}_\iota)^2 \end{cases} \quad (i = 1, 2, \ldots, k) \tag{3.11}$$

where $\hat{\mu}_{\theta i}$ is the average value of the *i*th movement direction; $\hat{\sigma}_{\theta i}$is the average deviation between $\hat{\mu}_{\theta i}$ and sample values in Θ_i . In this research, $\Theta_i \sim N\left(\mu_{\theta i}, \sigma_{\theta i}^2\right)(i = 1, 2, \ldots, k)$ is treated as the learnt geographical features of the coverage area of a learning grid.

The next task in the traffic pattern learning phase is to determine the local behaviour habits. When walking in the same local area, similar patterns can be observed as users share common intentions. As shown in Figure 3.16, mobile users have similar movement patterns when they want to reach the bus station. In this research, the local route choosing preferences are considered as the reflection of mobile users' behaviour habits.

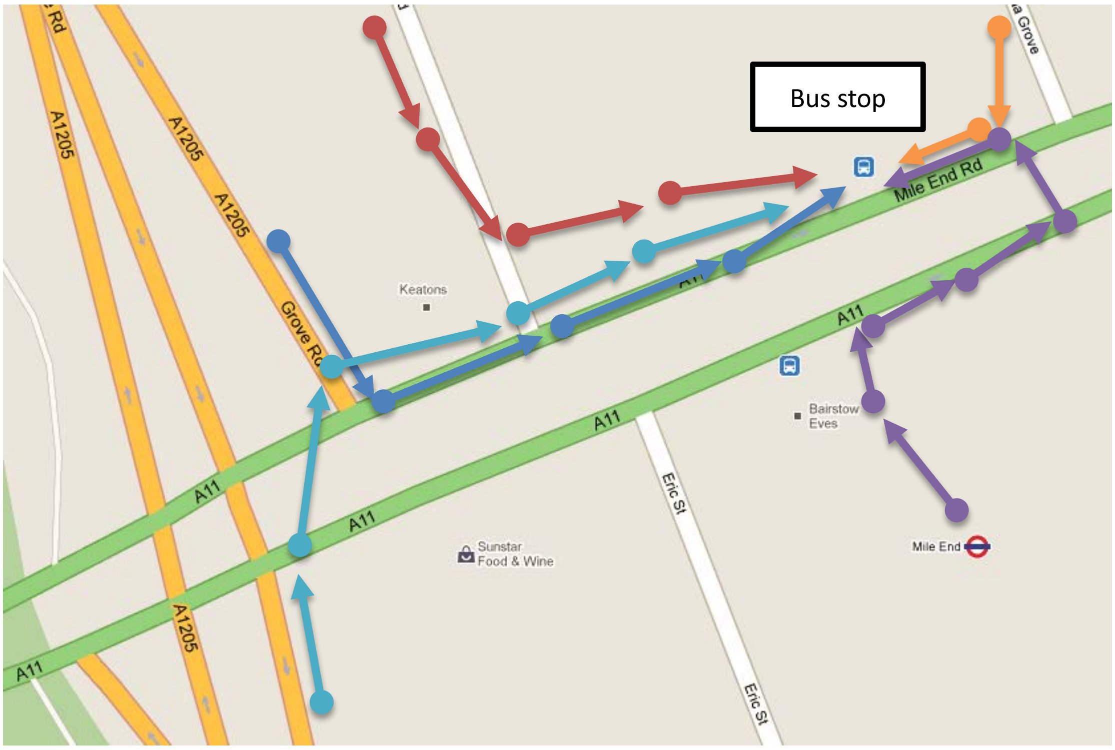


Figure 3.16 Typical movement trajectories towards the bus stop at Mile End, London

For a context sequence $C_{context} = \{C_{from}, C_1, C_2, \dots, C_n, C_{to}\}\ (n \geq 1)$, $C_i (i = 1, 2, \dots, n)$ are the observations within the learning grid; C_{from} and C_{to} are the observations in the neighbour areas of the learning grid. If C_{from} and C_{to} are substituted by C_0 and C_{n+1} respectively, the context sequence can be written as $C_{context} = \{C_0, C_1, C_2, \dots, C_n, C_{n+1}\}\ (n \geq 1)$. Then, the movement direction of the observed user at observation $C_i (i = 1, 2, \dots, n, n+1)$ can be obtained by applying equation 3.10 to (C_{i-1}, C_i). Assume that θ_i denotes the movement direction at C_i; k denotes the designated number of the main directions. As mentioned above, the 2π radians are averagely divided into k sectors, and sector j indicates the jth main direction with the value range $\left[\frac{(2j-3)\pi}{k}, \frac{(2j-1)\pi}{k}\right)$ $(j = 1, 2, 3, \dots, k)$. These k sectors are treated as k states in this part. The state of the movement direction θ_i at C_i is j, if $\frac{(2j-3)\pi}{k} \leq \theta_i < \frac{(2j-1)\pi}{k}$ exists. Assume $d(\theta_i)$ denotes the state of the movement direction at $C_i (i = 1, 2, \dots, n, n+1)$, then the context sequence of the direction states related to $C_{context} = \{C_0, C_1, C_2, \dots, C_n, C_{n+1}\}\ (n \geq 1)$ can be written as:

$$\vec{d}_{context} = [d(\theta_1), d(\theta_2), \dots, d(\theta_n), d(\theta_{n+1})]\ (n \geq 1) \tag{3.12}$$

By applying the conversion described above, all the context sequences recorded by a learning grid can be written as context sequences of direction states to indicate the local direction state transition regularities. Notice that the lengths of the direction state sequences are not the same, as the value of *n* in each $C_{context}$ varies.

As discussed in the previous section, the movements of mobile users are not memoryless, as the current pattern strongly depends on the previous L movement patterns. Let $\vec{d}_x(L)$ denote the direction state sequence of mobile user x with sequence length L. Assume that the direction state is a variable *D*, and $D(i,j)$ is a string $D_i D_{i+1} \dots D_j$ representing the sequence of values that *D* takes for $1 \leq i \leq j \leq L$, then the states of movement direction can be described as an order-L Markovian process as follows:

$$\begin{aligned} &Prob\left(D_{n+1} = a \middle| D(1,n) = \vec{d}_x(L)\right) \\ &= Prob(D_{n+1} = a | D(n - L + 1, n) = c) \\ &= Prob(D_{i+L+1} = a | D(i + 1, i + k) = c) \end{aligned} \tag{3.13}$$

Notice that the first two lines indicate that the probability depends on the most recent L states, while the latter two lines indicate the assumption of a stationary property of this process.

Each direction state sequence $\vec{d}_{context}$ recorded by the learning grid is a sub-string of a $\vec{d}_x(L)$, which reveals the direction state transitions when user *x* goes through the coverage area of the learning grid. Let $\mathbb{D}_m$ denote all the $\vec{d}_{context}$ sequences recorded by learning grid *m*; let $D_m = \{d_{m1}, d_{m2}, \dots, d_{ml}\}\ (l \geq 1)$ denote the direction state context sequence that could be observed in *m*; let $D_m(l)$ denote the *l*th states in D_m. Notice that all the states in D_m should be movement direction states that can be observed in *m*. Thus, the local direction state transition regularities in m can be presented as:

$$P_m(D_m(l+1) = a|D_m) = \frac{N((D_m, a);\ \mathbb{D})}{N(D_m; \mathbb{D})} \tag{3.14}$$

where $N(s; \mathbb{S})$ denotes the number of times the string s occurs as a substring in the string set $\mathbb{S}$; (s_1, s_2) denotes a string with s_1 as its former part and s_2 as its latter part. Notice that the Markov model above will degenerate to a lower order level, if D_m is seen for the first time.

The last task in the traffic pattern learning phase is to learn the short-term movement speed variation. In this research, the time interval between each traffic pattern observation ΔT is very short. The movement speed cannot be considered as a constant during the whole movement, as the movement speed of the mobile user might be affect by the unexpected incidents from the external environment. However, the speed in the consecutive time intervals should be similar and correlated, as the value of ΔT is small. Thus, the Weighted Moving Average (WMA) model, mentioned in [QWZW10], is applied in this research to make short-term predictions on the movement speed.

Let $v_x(t)$ denote the movement speed of mobile user *x* at step *t*; let $\hat{v}_x(t)$ denote the estimated speed at step *t*. Considering that the speed at time interval nearer to the time interval $t+1$ is more correlated with the movement speed at time interval $t+1$, a WMA model can be presented as:

$$\hat{v}_x(t+1) = \frac{\alpha_0 v_x(t) + \alpha_1 v_x(t-1) + \cdots + \alpha_k v_x(t-k)}{\alpha_0 + \alpha_1 + \cdots + \ \alpha_k} \tag{3.15}$$

where $\alpha_i (0 < \alpha_i < 1)$ denotes the weight and the contribution from $v_x(t-i)$. The values of α_i and k are designated at the beginning of the simulation.

3.5 Results Evaluation

When the predicted traffic distribution is obtained, evaluation methods need to be applied to determine the performance of the proposed pattern learning models. In this research, traffic pattern predictions are carried out to provide upcoming traffic pattern details for semi-smart antennas to adjust beamforming patterns. The existing load balancing schemes as presented in Chapter 2 do the calculation based on the local distribution of mobile users and the location of mobile clusters. Thus, the prediction results can be considered as accurate if the predicted traffic distribution in the target base station's coverage area is close to the real distribution, and the upcoming location of the mobile clusters can be correctly identified.

In this research, user distribution grids (UDGs) are introduced to evaluate the prediction performance. Two types of the UDGs are involved in the results evaluation, the square-shaped UDG (SUDG) and the arc-shaped UDG (AUDG), as shown in Figure 3.17.

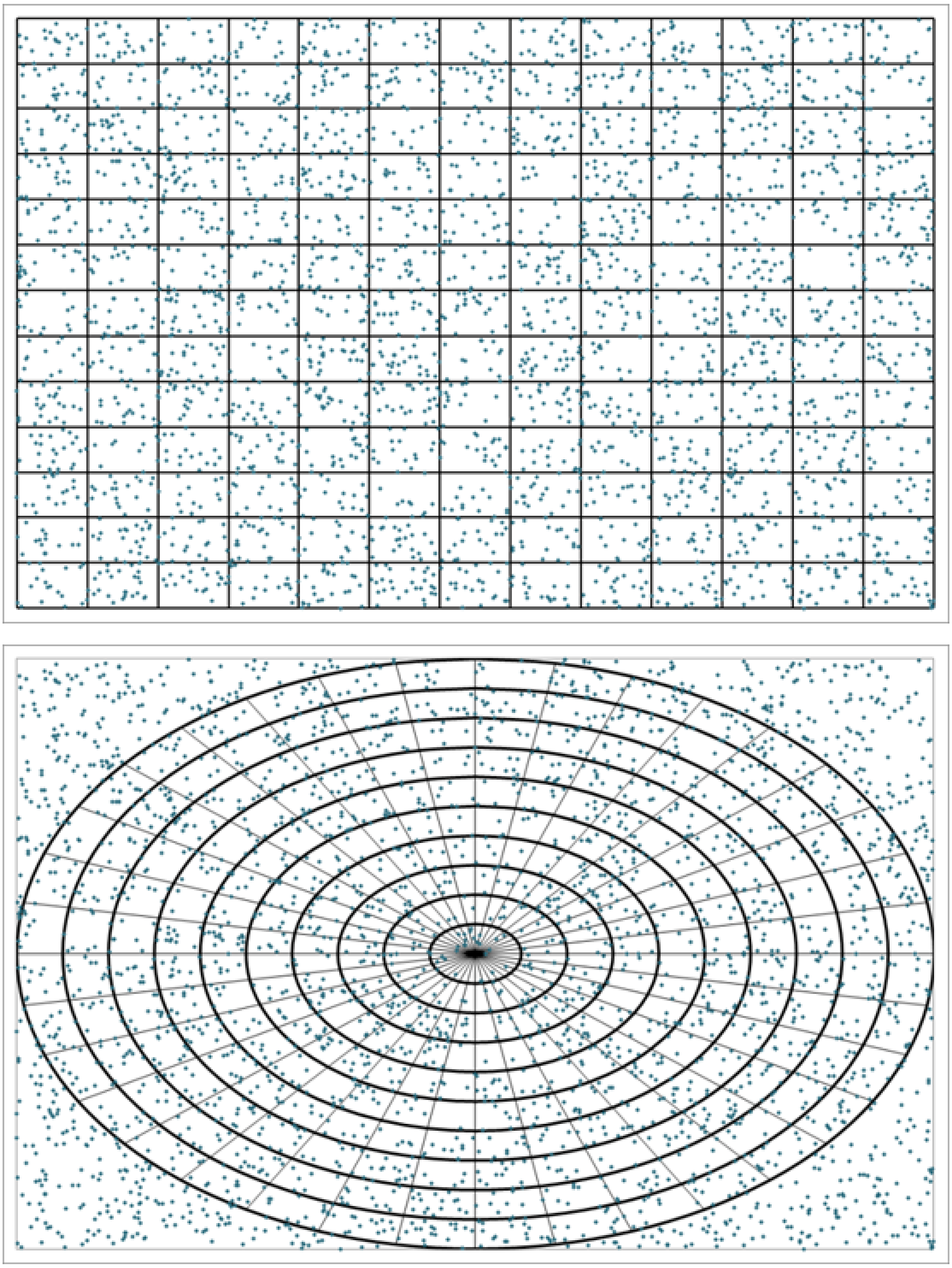


Figure 3.17 Examples of SUDGs and AUDGs

For the SUDG, the whole simulation area is divided into a number of square-shaped units with the predefined length and width. For the AUDG, the conventional coverage area of each base station is divided into sectors and bands, as shown in Figure 2.9. Then all the segments in the coverage area of a semi-smart antenna are considered as the UDG for this antenna.

When evaluating the prediction results with SUDGs, the cumulative error rate of each prediction is calculated in terms of the incorrect ratio of the population of RWPs in the SUDGs. Assume that $N_{ri}(m)$ denotes the real number of mobile users in the *m*th SUDG at step *i*. $N_{pi}(m)$ is used to denote the predicted number of mobile users in the *m*th UDG at step *i*. Thus, prediction error rate of the *m*th SUDG at step *i* can be calculated by the following equation:

$$e_{SUDG}(i,m) = \frac{N_{ri}(m) - N_{pi}(m)}{N_{ri}(m)} \tag{3.16}$$

and the cumulative error rate of the prediction at step *i* can be calculated by the following equation:

$$E_{SUDG}(i) = \sqrt{\frac{\sum_m e_{SUDG}(i,m)^2}{M}} \tag{3.17}$$

where *M* denotes the total number of the SUDG deployed in the simulation area.

When evaluating the prediction results with AUDGs, the cumulative error of each prediction is calculated in terms of difference of the population of RWPs in each AUDG. Notice that the centre point of one set of AUDGs, as shown in Figure 3.16, is the location of one BS, and AUDGs are deployed in the coverage area of this BS. When the granularity of the AUDG set is high, for example 36×20 which is used in [Yao07], the population of RWPs might be 0 in many AUDGs, which leads the denominator in equation 3.16 become invalid. Thus, the following equations are used to calculate the cumulative errors.

$$e_{AUDG}(i,m) = N_{ri}(m) - N_{pi}(m) \tag{3.18}$$

$$E_{AUDG}(i) = \sqrt{\frac{\sum_m e_{AUDG}(i,m)^2}{M}} \tag{3.19}$$

3.6 Verification and Validation

As explained in [LK97], verification is the check that the simulation model is performing as intended; validation is the check that the conceptual simulation model is an accurate representation of the system under study.

In this research, the verification of the mobility models and the learning models were carried out by detailed debugging of the codes and detailed study of printouts. Tracing information was added into each key step to track the correctness of the operations.

On the other hand, the validation of the mobility models was performed by checking the correctness of the movement trajectory against the predefined geographic layout; and the validation of the learning models was performed by checking the predictions made manually against the historical trajectory segments extracted from the relevant learning grid.

3.7 Summary

In this chapter, the main concepts of this research and the models involved are presented. First, the definition and features of the traffic pattern prediction carried out in this research is given. Then the SMP model used in this research is introduced from the perspectives of the RWP, the simulation area and the walking rules. Subsequently, the two pattern learning models and the result evaluation methods are presented.

The STP mobility model is novel, as the external influences are taken into account when generating individual movement trajectories and intra-cell traffic patterns are generated rather than the inter-cell distributions. The STPL pattern learning method is novel, as the cell transition data is used to uncover the local geographic layout and the aggregating intra-cell movement features are learnt rather than the visiting preference of neighbouring radio cells. The concept of combing the proposed traffic pattern prediction with load balancing is novel, as upcoming geographic distributions of mobile users are predicted and optimized radiation patterns are calculated accordingly to guide the beamforming adjustment rather than matching the observed congestion patterns with existing cases in the case library and choosing the radiation pattern related to the matched case ([Yao07]).

In the next chapter, the process of performing the traffic pattern prediction is described and the simulation results in different scenarios are presented and discussed.

4 PERFORMING TRAFFIC PATTERN PREDICTION

In this chapter, traffic pattern prediction is performed by using the two pattern learning models described in the previous chapter. At the beginning of this chapter, an overview of how traffic pattern prediction is performed is given. Then the configuration of the mobility model and the pattern learning models are introduced, followed by the description of the predefined scenarios. Finally, the simulation results are presented and discussed to reveal the feasibility of predicting upcoming traffic patterns.

4.1 An Overview of Performing Traffic Pattern Prediction

Generally speaking, the simulation carried out in this research is to investigate the feasibility of using proposed learning models to learn the local movement patterns and to make predictions on the upcoming distributions of mobile users and mobile clusters in cellular networks. Mobility models are used to provide historical movement patterns by simulating the real-life traffic patterns of moving mobile users. Pattern learning models are used to learn the local movement patterns according to the historical data and to make predictions on the upcoming pattern variations. Traffic patterns generated by the mobility models are called “real patterns”; Traffic patterns predicted by the learning models are called “predicted patterns”. Both the real pattern and the predicted pattern are given a time stamp to indicate the moment at which the pattern is seen or will be seen. The time stamp here is the step number, at which the pattern is observed. The performance is evaluated by making comparisons between the real and the predicted traffic patterns that have the same time stamp.

In the simulation, a number of RWPs continuously move in a predefined simulation area. The simulation area is composed of a number of geographic grid elements, which are used to mimic a variety of road topologies. All the RWPs change their speed and direction according to the restrictions in the geographic grid element. In order to learn the mean behaviour regularity in each specific area, learning grids are involved to cover the whole simulation area, capture the historical movement data, and learn the mobility features. Then, with the knowledge that has already been learnt, the future locations of the RWPs are predicted to present the distribution of the mobile terminals in future steps. Figure 4.1 shows the flowchart of the simulation described in this chapter.

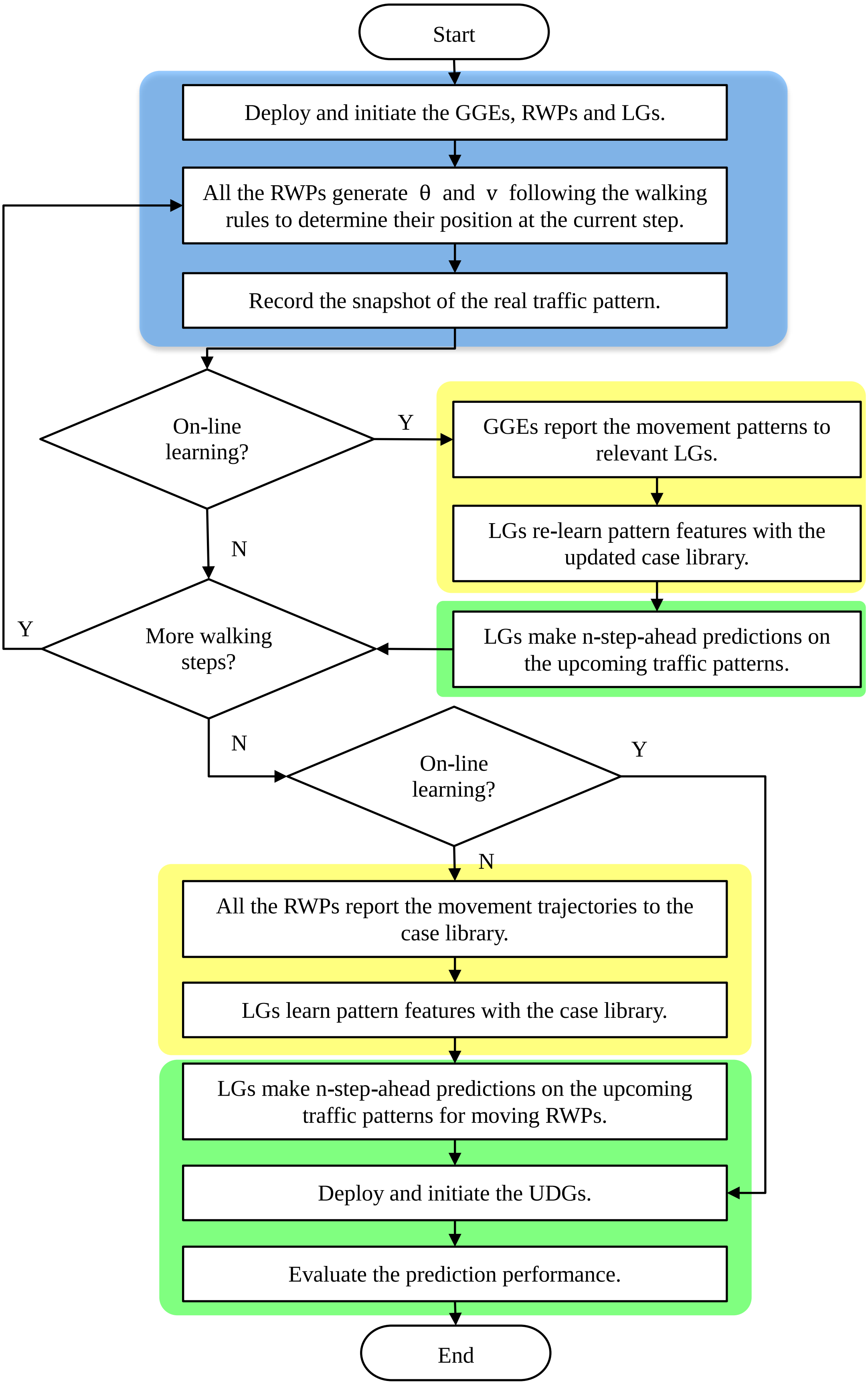

Start
Deploy and initiate the GGEs, RWPs and LGs.
All the RWPs generate θ and v following the walking rules to determine their position at the current step.
Record the snapshot of the real traffic pattern.
On-line learning?
Y
GGEs report the movement patterns to relevant LGs.
LGs re-learn pattern features with the updated case library.
N
Y
More walking steps?
LGs make n-step-ahead predictions on the upcoming traffic patterns.
N
On-line learning?
Y
N
All the RWPs report the movement trajectories to the case library.
LGs learn pattern features with the case library.
LGs make n-step-ahead predictions on the upcoming traffic patterns for moving RWPs.
Deploy and initiate the UDGs.
Evaluate the prediction performance.
End

Figure 4.1 Simulation flowchart

In this research, either the on-line pattern learning or the off-line pattern learning can be performed. With the on-line pattern learning, the pattern prediction is carried out with pattern learning at the same time; with the off-line pattern learning, the pattern prediction is performed based on the knowledge that has already been learnt from the historical data. From the perspective of tasks, the simulation process can be divided into three phases, the pattern generating phase, the pattern learning phase, and the pattern prediction phase. In Figure 4.1, blue areas represent the pattern generating phase; yellow areas represent the pattern learning phase; green areas represent the pattern prediction phase.

4.1.1 Pattern generating phase

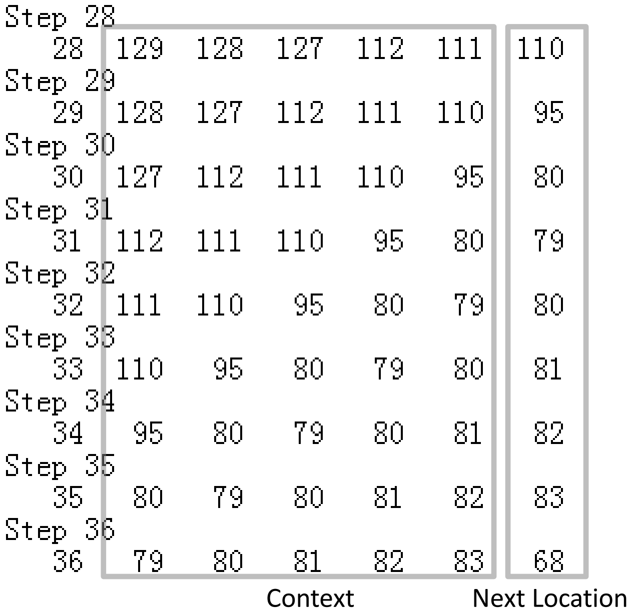
In the pattern generating phase, the main task is to generate traffic patterns that can be used as historical data by the pattern learning models. At the beginning of the simulation, all the RWPs are distributed in the simulation area uniformly or following the predefined rules. Then, the RWPs independently generate the walking parameters r and θ to carry out the movement. Notice that the r and θ values generated by an RWP are not dependent on the parameter values generated by others, but impacted by the predefined local road conditions. The higher the throughput towards one direction is, the larger the flux of that direction could be seen. Besides, in order to eliminate the unrealistic direction change, there is limited number of chances for an RWP to change its moving direction at the same visit of one geographic grid element. Since the simulation area is a boundless area, RWPs that reach one side of the simulation area continue travelling and reappear on the opposite side of the simulation area.

On the other hand, the GGEs are used to support the random walking of RWPs by reproducing the factors and phenomena that can be seen in real life. First, the road topology is reproduced by different direction changing probabilities predefined in each grid element. If the direction changing probability of each direction is similar, that area probably is an intersection. If the direction changing probability of “east” and “west” is much higher than other probabilities, it means that there probably is an east-west major road in this area. Second, the throughput capacity of different roads is reproduced by the throughput capability parameters. Major roads have larger throughput capability parameters. Besides, the throughput capability parameter is also used to calculate the congestion coefficient when the RWP determines the movement speed at each step. The value of the congestion coefficient in one direction is inversely proportional to the user density of that direction. The term “user density” is defined as the number of RWPs that moving towards the same direction in a specific area. Third, different traffic scenarios are created by adjusting the values of direction changing probability and the road throughput capability. The mobile clusters can be easily formed by decreasing the value of the road throughput capability, as happens in real life.

4.1.2 Pattern learning phase

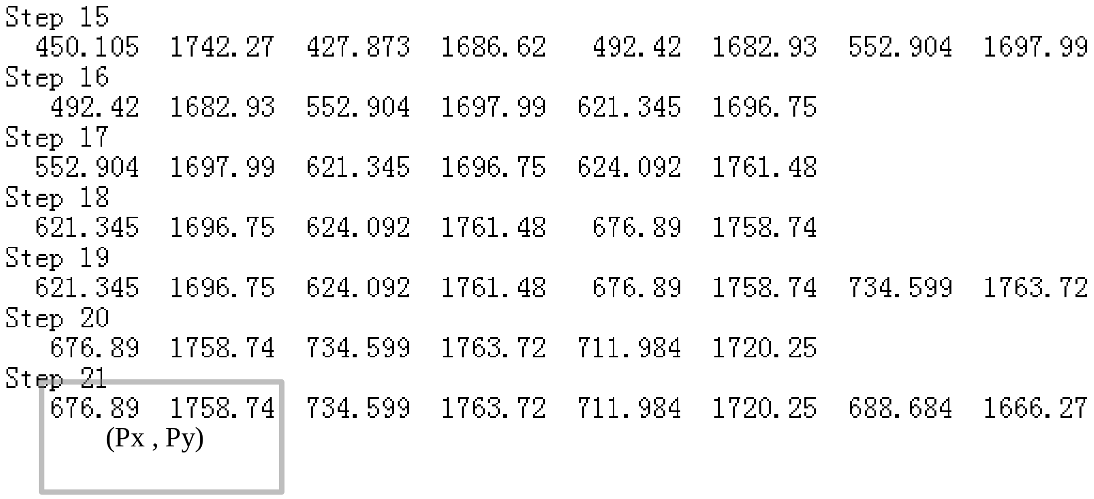
The main task of this phase is to record the local movement pattern and learn the local behaviour regularity. The order-k Markov learning model and the STPL model are used in this section to illustrate the process of extracting the local traffic patterns. In this research, either the on-line pattern learning or the off-line pattern learning can be employed.

When applying the on-line pattern learning mode, the RWPs need to report the movement trajectory segments to the relevant learning grids at the end of each step, which allows the latest traffic patterns to be taken into account. With a proper predefined time window, abnormal traffic patterns can be noticed right after they are captured by the learning grids. The reported trajectory segment for the order-*k* Markov model is the most recent $k + 1$ length movement trajectory represented by a string of IDs of the related learning girds. The reported trajectory segment for the STPL model is the movement trajectory from the last observed location before visiting the target learning grid to the first observed location after moving out of this learning grid. This reported trajectory segment is represented by a string of coordinates. Examples of these two kinds of trajectory segments are shown in Figure 4.2 (a) and Figure 4.2 (b). One drawback of using the on-line mode is that pattern generation, pattern learning and pattern prediction are carried out in parallel, which might lead to a slow operating speed.



```
Step 28
   28  129  128  127  112  111   110
Step 29
   29  128  127  112  111  110    95
Step 30
   30  127  112  111  110   95    80
Step 31
   31  112  111  110   95   80    79
Step 32
   32  111  110   95   80   79    80
Step 33
   33  110   95   80   79   80    81
Step 34
   34   95   80   79   80   81    82
Step 35
   35   80   79   80   81   82    83
Step 36
   36   79   80   81   82   83    68
```

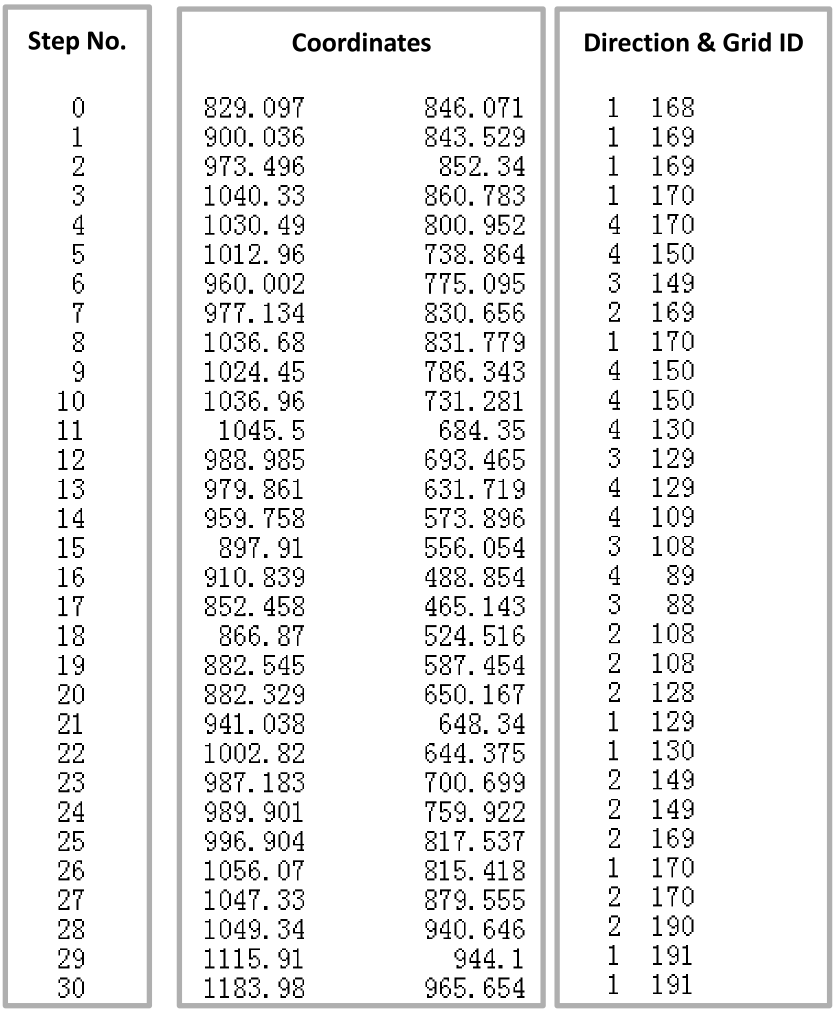
Figure 4.2 (a) Context-Next Location pairs for Order-5 Markov model



```
Step 15
  450.105  1742.27  427.873  1686.62   492.42  1682.93  552.904  1697.99
Step 16
   492.42  1682.93  552.904  1697.99  621.345  1696.75
Step 17
  552.904  1697.99  621.345  1696.75  624.092  1761.48
Step 18
  621.345  1696.75  624.092  1761.48   676.89  1758.74
Step 19
  621.345  1696.75  624.092  1761.48   676.89  1758.74  734.599  1763.72
Step 20
   676.89  1758.74  734.599  1763.72  711.984  1720.25
Step 21
   676.89  1758.74  734.599  1763.72  711.984  1720.25  688.684  1666.27
```

Figure 4.2 (b) Coordinates strings for STPL model

When applying the off-line pattern learning mode, the complete movement trajectory of each mobile terminal is recorded instead of trajectory segments. The recorded movement trajectory contains the coordinates, the moving direction, and the related learning grid’s ID of each observation, as shown in Figure 4.3. When collecting the pattern data, each movement trajectory is written in an individual text file, and the file name is reported to the learning grids that appear in the trajectory to expedite the generation of learning patterns. Before starting to learn the pattern features, local traffic patterns that are suitable for the applied learning model should be extracted from the recorded trajectories by each learning grid. Although the pattern learning phase is more complicated and time consuming in the off-line mode, predictions can be made much faster with the knowledge learnt off-line. One drawback of using the off-line mode is that response time to real-time traffic pattern changes is long.



Step No.	Coordinates		Direction & Grid ID	
0	829.097	846.071	1	168
1	900.036	843.529	1	169
2	973.496	852.34	1	169
3	1040.33	860.783	1	170
4	1030.49	800.952	4	170
5	1012.96	738.864	4	150
6	960.002	775.095	3	149
7	977.134	830.656	2	169
8	1036.68	831.779	1	170
9	1024.45	786.343	4	150
10	1036.96	731.281	4	150
11	1045.5	684.35	4	130
12	988.985	693.465	3	129
13	979.861	631.719	4	129
14	959.758	573.896	4	109
15	897.91	556.054	3	108
16	910.839	488.854	4	89
17	852.458	465.143	3	88
18	866.87	524.516	2	108
19	882.545	587.454	2	108
20	882.329	650.167	2	128
21	941.038	648.34	1	129
22	1002.82	644.375	1	130
23	987.183	700.699	2	149
24	989.901	759.922	2	149
25	996.904	817.537	2	169
26	1056.07	815.418	1	170
27	1047.33	879.555	2	170
28	1049.34	940.646	2	190
29	1115.91	944.1	1	191
30	1183.98	965.654	1	191

Figure 4.3 Movement trajectory recorded in the off-line mode

Either applying the on-line learning mode or the off-line learning mode, learning patterns that are suitable for the applied learning model are needed before carrying out the pattern learning phase. With the on-line mode, all the patterns recorded by the learning girds are ready to use. With the off-line mode, learning patterns, with the same format as patterns captured in the on-line mode, need to be extracted from the recorded trajectories. When performing the pattern learning algorithm, there is no difference between the on-line mode and the off-line mode.

With the proposed order-k Markov model, $k+1$ length "context, next location" pairs $\{l_{n-k}, l_{n-k+1}, \dots, l_{n-1};\ l_n\}$ are used by each learning gird to reveal the local traffic features. With Equation 3.9, the conditional probability of the next possible location $\hat{P}(L_{n+1} = l \,|L_{context})$ can be calculated by given a k length observed movement trajectory $L_{context} = \{L_n, L_{n-1}, \dots, L_{n-k+1}\}$. In the learning phase, related "context, next location" pairs are loaded into the buffer of the learning grid for prediction use, rather than doing calculation.

With the proposed STPL model, the context sequences of coordinates $C_{context} = \{C_{from}, C_1, C_2, \dots, C_n, C_{to}\}\ (n \geq 1)$ are used by each learning gird to discover the local geographic layout and the local behaviour habits. Assume 2π radians are averagely divided into k sectors that represent the possible movement directions. With Equation 3.10 and 3.11, the regularities of each possible movement direction in the local area can be expressed as a probability distribution. With Equation 3.10, a coordinate sequence can be converted to a context sequence of the direction states $\vec{d}_{context} = [d(\theta_1), d(\theta_2), \dots, d(\theta_n), d(\theta_{n+1})]\ (n \geq 1)$. The converted direction state sequences are loaded into the learning grid's buffer for calculating the conditional probability of the next moving direction in the prediction phase.

4.1.3 Pattern prediction phase

The main task of this phase is to estimate the distributions of mobile terminals in the upcoming observation steps. As mentioned above, the location of a moving mobile terminal at step $t + 1$ is correlated with the movement patterns from step $t - k + 1$ to step t. With the most recent k observations, the next location of the moving mobile terminal can be estimated by the pattern learning models according to the local traffic features learnt from the historical data. In this research, predictions are made for individual mobile terminals. Then, the predicted distributions of all the mobile terminals are compared to the real observed traffic distributions to determine the prediction accuracy.

When applying the order-k Markov model, future locations are represented by the IDs of the learning grid. When making a prediction, an RWP sends a k length historical trajectory including k most recent IDs of the visited learning grid to the current learning grid where this RWP located. By using Equation 3.9, the probability of the possible next locations can be obtained. Two decision rules can be chosen here to decide the next location. The first rule is to choose the location with the highest probability as the next location; the second rule is to choose the location ID according to the probability distribution. When the prediction for one step is finished, all the RWPs with the same location ID are uniformly distributed in that area.

When applying the STPL model, future locations are represented by the coordinates of the RWP in the simulation area. When making a prediction, the walking parameters are generated for the visiting RWP by the learning grid to predict next location of the RWP. In order to provide the context information, a sequence of coordinates $\left\{C_{from}, C_1, C_2, \ldots, C_n\right\} (n \geq 1)$, including the coordinates from the most recent observation back to the last observation before visiting the current learning grid, is reported to the currently visited learning grid. With Equation 3.10, the context sequence can be converted to a sequence of movement direction states $\{d(\theta_1), d(\theta_2), \ldots, d(\theta_n)\} (n \geq 1)$. By applying the direction state sequence into Equation 3.14, the probability of the possible movement directions in the next step can be obtained. Similarly, two decision rules are available here, to choose the direction with the highest probability or to decide the moving direction according to the obtained probability distribution. When the next moving direction is determined, the estimated walking parameter $\hat{\theta}$ can be generated by using the learnt probability density function of the value of θ for that direction. By applying Equation 3.15, the walking parameter $\hat{v}$ can be estimated. Finally, Equation 3.4 is used to calculate the predicted coordinates of this RWP at the next step.

When predicting the future traffic patterns, all the learning grids cooperatively carry out a continued movement prediction for n steps. This is called the n-step-ahead prediction in the simulation. In the n-step-ahead prediction, only the first prediction uses the information from the real movement history. In the remaining $n-1$ predictions, the previous prediction results are used as the latest movement history when calculating the next coordinates or generating new contexts. Assume $\{C_{i-l+1}, C_{i-l+2}, \ldots, C_i\}$ is the l-length movement context; $\hat{C}_{i+1}$ is the 1-step-ahead prediction result. Thus, for the 2-step-ahead prediction, $\left\{C_{i-l+2}, \ldots, C_i, \hat{C}_{i+1}\right\}$ is used as the movement context to estimate $\hat{C}_{i+2}$, and the rest may be deduced by analogy.

When evaluating the prediction performance, the predicted traffic distribution of each step is compared to the real pattern that has the same time stamp as the predicted one. As mentioned before, a time stamp is given to every real traffic pattern that is generated by the mobility model. This time stamp is used to indicate the order of each traffic pattern in the temporal axis. Assume that there is a traffic pattern generated by the mobility model at step k. The time stamp of this real pattern is k; the time stamps of the next n continuous patterns are $k+1, k+2, \ldots, k+n$. When making the n-step-ahead prediction at step k, the prediction results are the estimated traffic pattern at step $k+1, k+2, \ldots, k+n$ with the time stamp $k+1, k+2, \ldots, k+n$ respectively. Then, the real and predicted patterns at $k+i$ $(i=1,2,\ldots,n)$ are compared to reveal how close the predicted pattern is to the real one.

4.2 Simulation Environment

The main goal in this research is to test the feasibility of making predictions on the upcoming geographic distributions of mobile terminals by mining the historical cell transition trajectories. If the accuracy of the prediction is acceptable, then predicted future traffic patterns could be used by the load balancing algorithms, such as the Bubble Oscillation Algorithm, to help semi-smart antennas adjusting the radiation patterns.

When calculating the optimal radiation patterns with the Bubble Oscillation Algorithm, the geographic distribution pattern of the mobile users is needed, especially the distribution of the active users. With the predicted geographic distribution pattern, the Bubble Oscillation Algorithm can work out a set of the proper radiation patterns to alleviate the upcoming negative effects on the network performance. In order to reduce the modelling complexity, all the RWPs in the simulation area are treated as active mobile users.

In the following section, other configurations of the simulator and the predefined scenarios are presented.

4.2.1 Initialization of simulation area

In the simulations carried out in this research, the simulation area was defined as a 3 kilometres long and 3 kilometres wide square-shaped area. The GGE, which was used to reflect the local geographic patterns, was defined as a 0.5 kilometres long and 0.5 kilometres wide square-shaped grid. Thus, 36 GGEs were deployed in the simulation area to reproduce the road topology. Figure 4.4 illustrates the configurations of the simulation area and the GGEs. The map used in Figure 4.4 is a real map of the Mile End area in London. The length and width of the grid use the same scale as the map. It can be seen that most of the major roads in the same $0.25km^2$ area have similar directions. Thus, the proposed size of the GGE is an acceptable value.

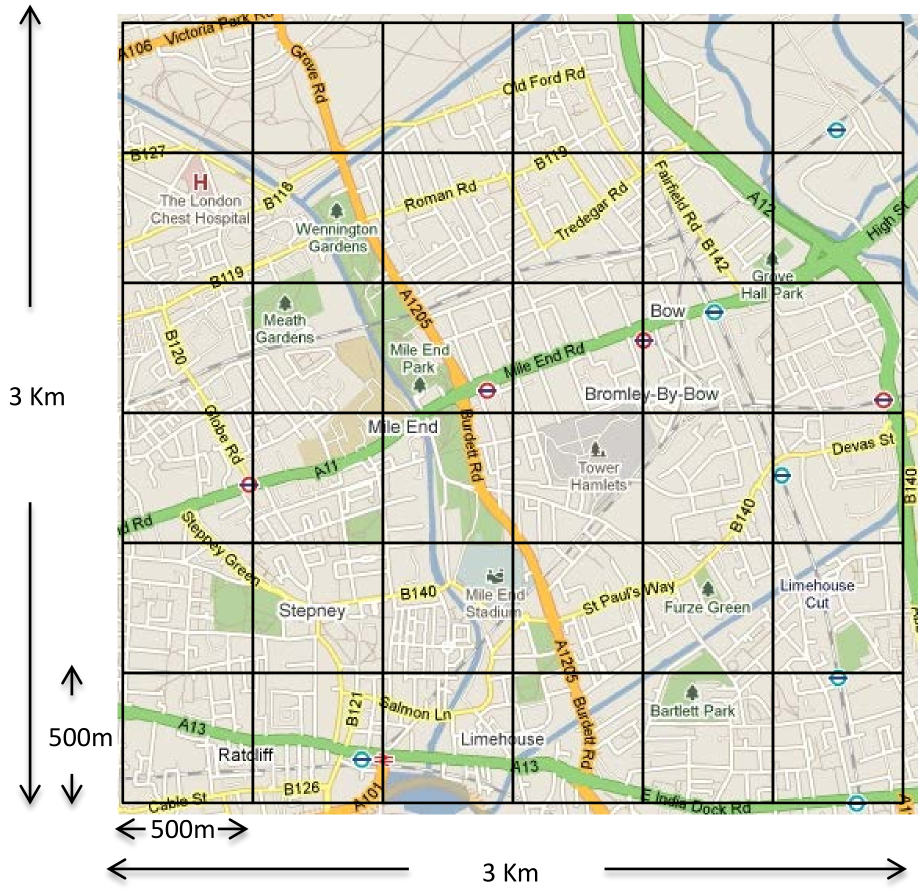


Figure 4.4 An example of the simulation area with deployed GGEs

In this simulation, “East”, “North”, “West” and “South” were the four main road directions, and were mapped to direction 1, direction 2, direction 3 and direction 4 respectively. In the initialization of the deployed GGEs, the road directions, the road throughput capability of each direction and the direction transition matrix need to be determined. In real life, most of the roads are bidirectional. Thus, θ_{r13} and TP_{13} were used to represent the road direction and the throughput capability of East and West respectively; θ_{r24} and TP_{24} were used to represent the road direction and the throughput capability of North and South respectively. Five levels ($TP_{Lv} = 1,2,..5$) were used to rank the throughput capability. The higher the TP_{Lv} is, the larger throughput capability the road could have. In real life, the width and direction of a road varies smoothly. Thus, the values of θ_{r13} and TP_{Lv_EW} were allowed to change slightly or remain constant in the East-West neighbouring GGEs; the values of θ_{r24}

and TP_{Lv_NS} were allowed to change slightly or remain constant in the North-South neighbouring GGEs. The values of θ_{r13}, θ_{r24}, TP_{Lv_EW} and TP_{Lv_NS} were generated randomly for the GGEs in the bottom and on the left side. Then, other GGEs determined values of θ_{r13}, θ_{r24}, TP_{Lv_EW} and TP_{Lv_NS} accordingly to avoid sudden changes of the local road topology. The road directions can be determined by the following equations.

$$\begin{cases} \theta_{r1} = \theta_{r13} \\ \theta_{r2} = \theta_{r24} \\ \theta_{r3} = \theta_{r13} + \pi \\ \theta_{r4} = \theta_{r24} + \pi \end{cases} \tag{4.1}$$

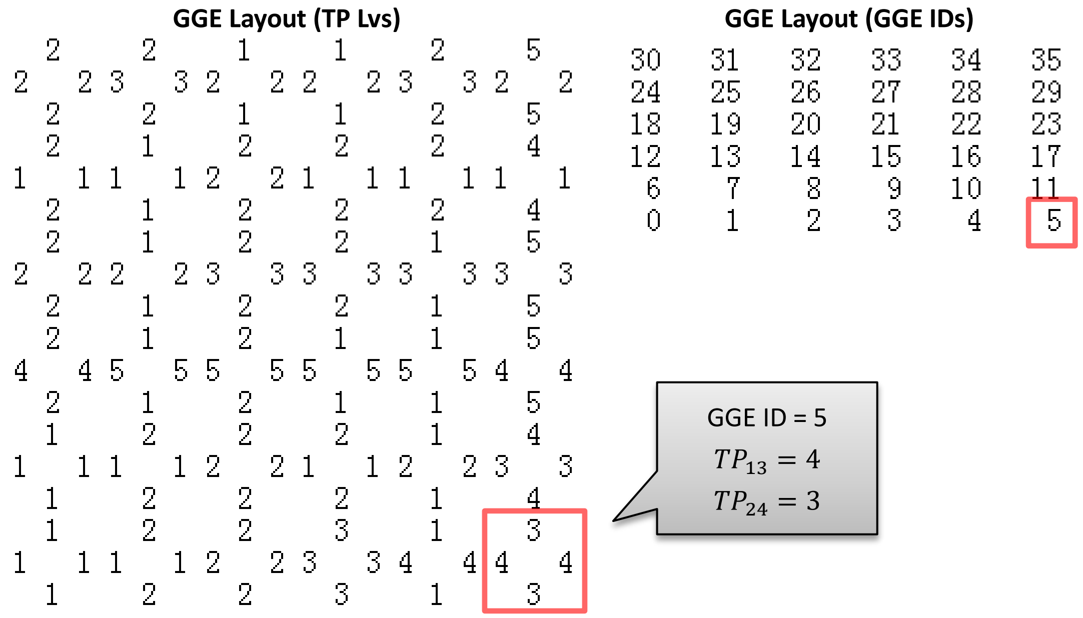
The throughput capability of each direction of a GGE can be determined by the following equations.

$$\begin{gathered} TP_{13} = u + u \cdot (TP_{Lv\_EW} - 1) \cdot \delta \\ TP_{24} = u + u \cdot (TP_{Lv\_NS} - 1) \cdot \delta \\ TP_1 = TP_3 = TP_{13} \\ TP_2 = TP_4 = TP_{24} \end{gathered} \tag{4.2}$$

where u is the unit number of users, which was set to 100 in the simulation; δ is the incremental factor, which was set to 0.5 in the simulation to avoid significant difference of the throughput capability between roads with the adjacent throughput level. In real life, people prefer to choose major roads for their journey to save travel time. The probability of choosing the road with stronger throughput capability should be higher than the probability of choosing roads with low throughput level. Thus, the direction transition matrix is initialized with the following equation.

$$P_{ij} = \begin{cases} \dfrac{TP_j}{\sum_k TP_k} & (k = 1,2,3,4 \; and \; |k - i| \neq 2 \; and \; |j - i| \neq 2) \\ 0 & (|j - i| = 2) \end{cases} \tag{4.3}$$

Notice that Equation 4.3 is only used to generate the initial value of P_{ij}, and it can be seen that U-turns are forbidden in the simulation area. The value of P_{ij} could be changed by activated events during the simulation. Figure 4.5 (a) (b) and (c) show examples of the initialized GGEs.



GGE Layout (TP Lvs)

```
  2     2     1     1     2     5
2   2 3   3 2   2 2   2 3   3 2   2
  2     2     1     1     2     5
  2     1     2     2     2     4
1   1 1   1 2   2 1   1 1   1 1   1
  2     1     2     2     2     4
  2     1     2     2     1     5
2   2 2   2 3   3 3   3 3   3 3   3
  2     1     2     2     1     5
  2     1     2     1     1     5
4   4 5   5 5   5 5   5 5   5 4   4
  2     1     2     1     1     5
  1     2     2     2     1     4
1   1 1   1 2   2 1   1 2   2 3   3
  1     2     2     2     1     4
  1     2     2     3     1     3
1   1 1   1 2   2 3   3 4   4 4   4
  1     2     2     3     1     3
```

GGE Layout (GGE IDs)

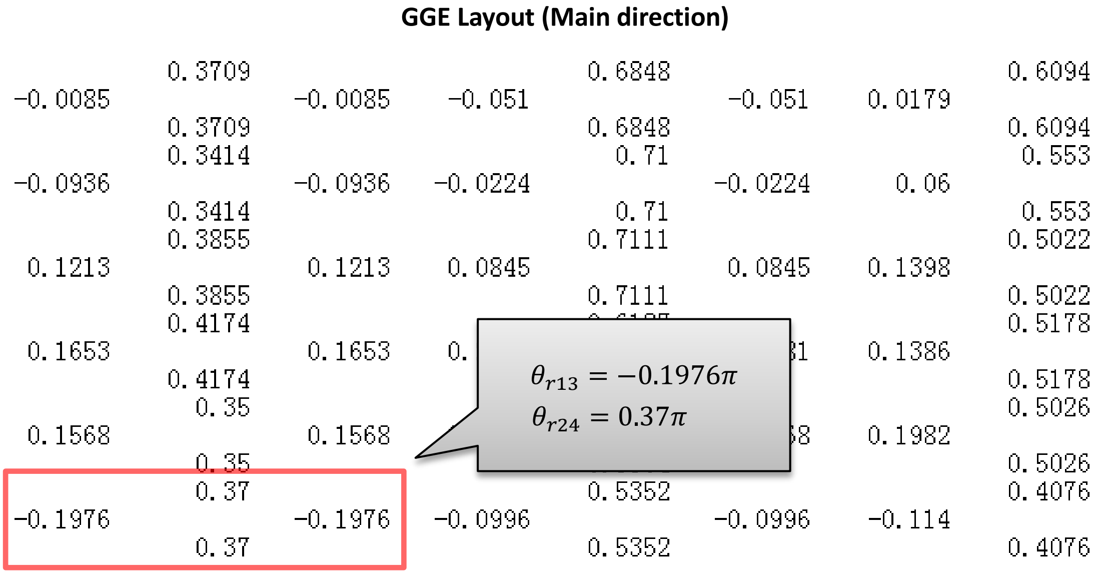
```
  30  31  32  33  34  35
  24  25  26  27  28  29
  18  19  20  21  22  23
  12  13  14  15  16  17
   6   7   8   9  10  11
   0   1   2   3   4   5
```

GGE ID = 5

$TP_{13} = 4$

$TP_{24} = 3$

Figure 4.5 (a) The geographic layout of GGEs represented by IDs and TP Levels



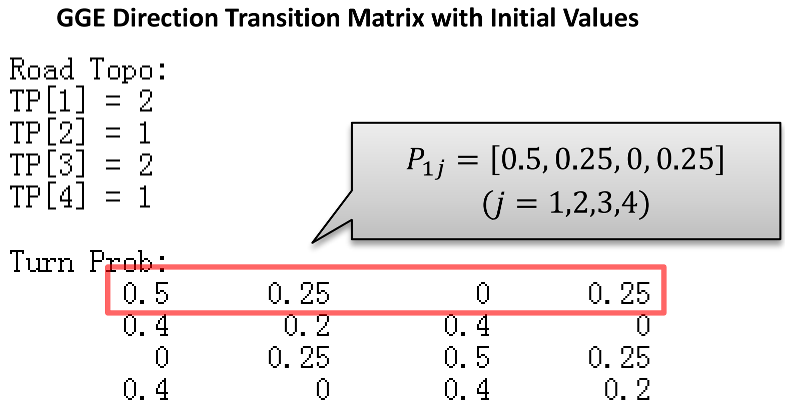
GGE Layout (Main direction)

```
            0.3709                     0.6848                     0.6094
  -0.0085           -0.0085   -0.051            -0.051   0.0179
            0.3709                     0.6848                     0.6094
            0.3414                       0.71                      0.553
  -0.0936           -0.0936  -0.0224           -0.0224     0.06
            0.3414                       0.71                      0.553
            0.3855                     0.7111                     0.5022
   0.1213            0.1213   0.0845            0.0845   0.1398
            0.3855                     0.7111                     0.5022
            0.4174                                                0.5178
   0.1653            0.1653   0.                     1   0.1386
            0.4174                                                0.5178
              0.35                                                0.5026
   0.1568            0.1568                          8   0.1982
              0.35                                                0.5026
              0.37                     0.5352                     0.4076
  -0.1976           -0.1976  -0.0996           -0.0996   -0.114
              0.37                     0.5352                     0.4076
```

$\theta_{r13} = -0.1976\pi$

$\theta_{r24} = 0.37\pi$

Figure 4.5 (b) The geographic layout of GGEs represented by road directions



GGE Direction Transition Matrix with Initial Values

```
Road Topo:
TP[1] = 2
TP[2] = 1
TP[3] = 2
TP[4] = 1

Turn Prob:
         0.5        0.25          0        0.25
         0.4         0.2        0.4           0
           0        0.25        0.5        0.25
         0.4           0        0.4         0.2
```

$P_{1j} = [0.5, 0.25, 0, 0.25]$
$(j = 1,2,3,4)$

Figure 4.5 (c) An example of the direction transition matrix

4.2.2 Initialization of RWPs

In the simulation, 15000 RWPs were created and distributed in the simulation area. The walking rules presented in Section 3.3.3 were applied by the RPWs to determine their coordinates at each step. The time interval between two consecutive steps was 60 seconds. In order to simplify the simulation, only one type of moving users, i.e. pedestrians, was taken into account. According to the average walking speed of pedestrians in real life, v_{normal} and v_{max} of the RWP were set to 80 m/min and 100 m/min. The speed dependence controller ω defined in Equation 3.5 was set to 0.5, which means the previous moving speed has a moderate effect on the current speed.

4.2.3 Simulation scenarios

In order to test the performance of the learning model for predicting a variety of traffic patterns, three simulation scenarios are defined, the normal walking scenario, the mobile cluster forming scenario, and the mobile cluster dispersing scenario.

In the normal walking scenario, all the RWPs are distributed uniformly in the simulation area and all the connatural parameters of each GEE are kept constant in the simulation. This scenario is used to test the performance of the proposed learning models with the normal traffic pattern.

In the mobile cluster forming scenario, mobile clusters are formed intentionally by two methods. The first method is to degrade the throughput level of a number of selected roads to a very low value. By degrading the throughput level, the movement speed of the mobile terminals in those areas is affected, which makes the mobile terminals stay longer in those areas. The second method is to modify the direction transition matrix of a number of selected GGEs to divert the traffic to the direction with a low throughput level. In the simulation, the above two methods are used cooperatively to form the mobile clusters in a very short time. This scenario is used to mimic a traffic jam caused by traffic accident or short-term traffic restriction in real life. The performance and reaction speed of the learning models could be evaluated in this scenario.

In the mobile cluster dispersing scenario, a number of mobile clusters are distributed initially in the simulation area. When starting the simulation, the mobile terminals in the mobile clusters begin to move according to the walking rules in section 3.3.3. The relevant direction transition matrix can be modified to divert the movement of the mobile cluster intentionally. On the other hand, the dispersion of the mobile clusters can be observed if the parameters of the GGEs are not changed. This scenario is used to mimic the traffic pattern around a stadium after a football match or a concert in real life. The performance of the learning models on predicting the movement of mobile clusters are evaluated in this scenario.

4.2.4 Initialization of learning models

In the simulation, the order-*k* Markov model and the STPL model were applied to learning the pattern features and to make predictions on the upcoming traffic pattern with the given trajectory context. As it has been proved that the second decision rule can lead to better performance, both the order-*k* Markov model and the STPL model used the second decision rule in the simulation.

The off-line learning mode was applied for both learning models. A case library was built with 50000 movement trajectory samples, which were generated with the same geographic topology used in the simulation. The learning patterns were extracted by the two learning models from the case library for pattern learning use.

4.3 Simulation Results

In this section, the simulations results are presented to reveal the feasibility of making predictions on future traffic with proposed learning models. Simulations were carried out in three scenarios respectively. Different numbers of learning grids and UDGs were deployed to test the granularity influences.

4.3.1 Simulation result in the normal walking scenario

In the simulation, all the RWPs were uniformly distributed in the simulation area. Figure 4.6 (a), (b) and (c) shows the real traffic pattern at step 0 (initial pattern), step 50, and step 100 respectively. Both the x-axis and y-axis represent the length of the simulation area. The black triangle represents the location of the BS; the black circle represents the coverage area of the BS. The square-shaped grids are the deployed GGEs. The blue dots are the RWPs.

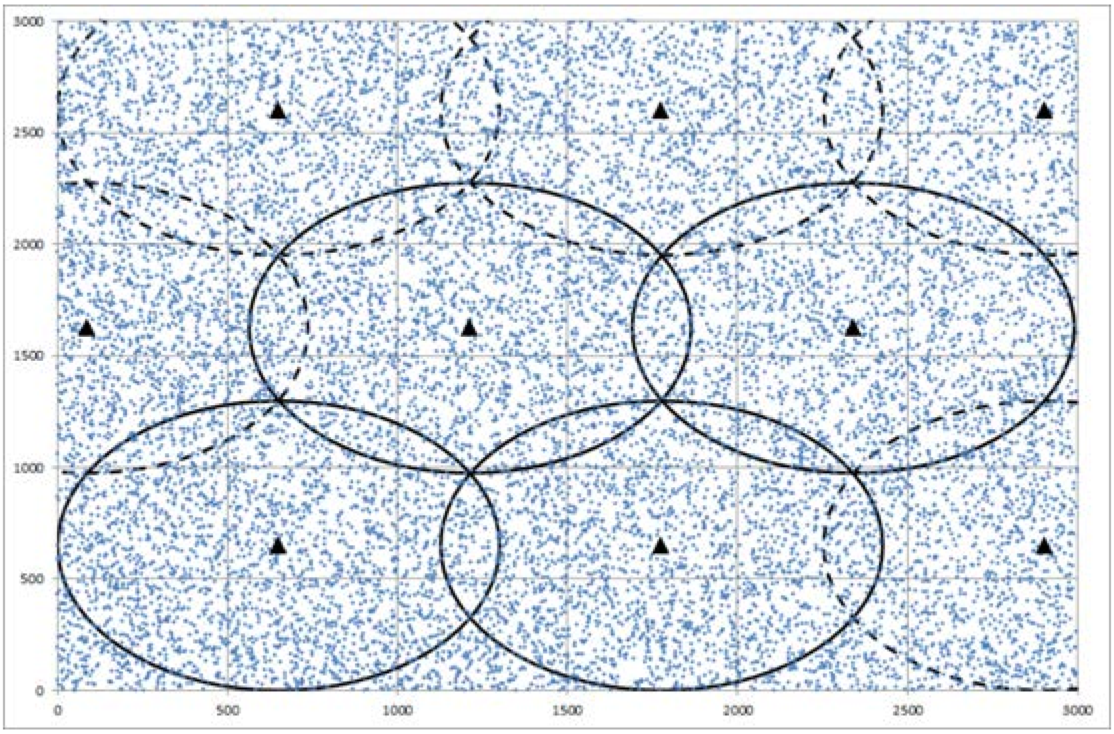


Figure 4.6 (a) The initial distribution of the mobile terminals

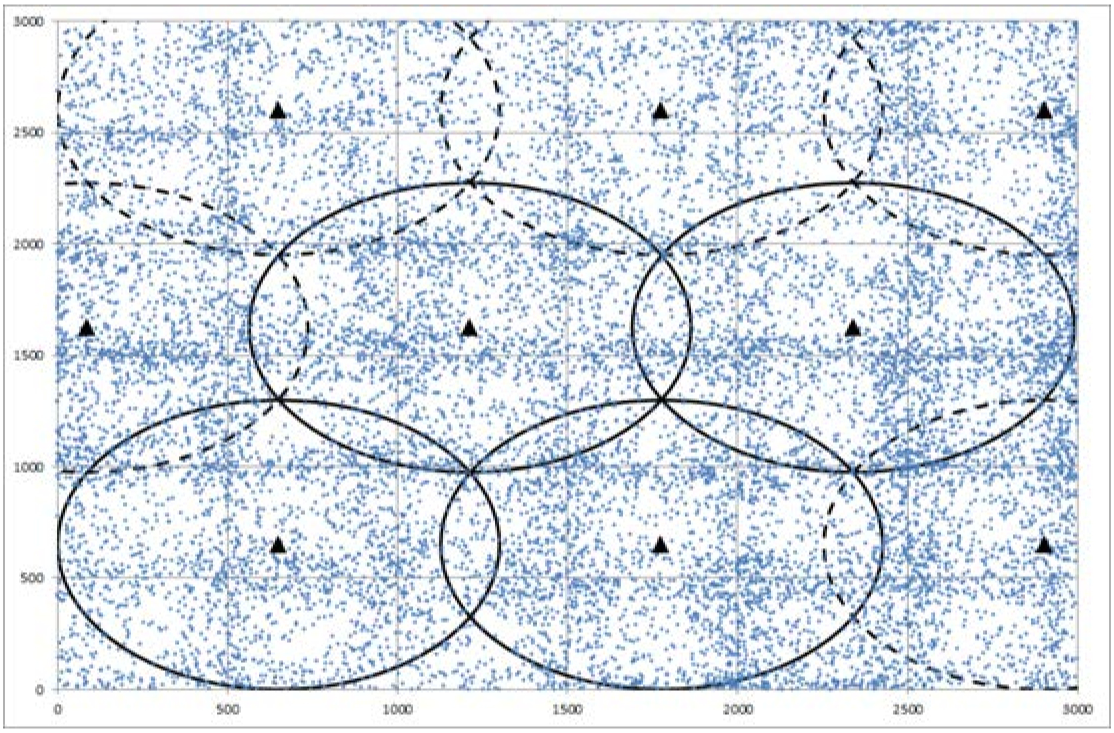


Figure 4.6 (b) The distribution of mobile terminals at Step 50

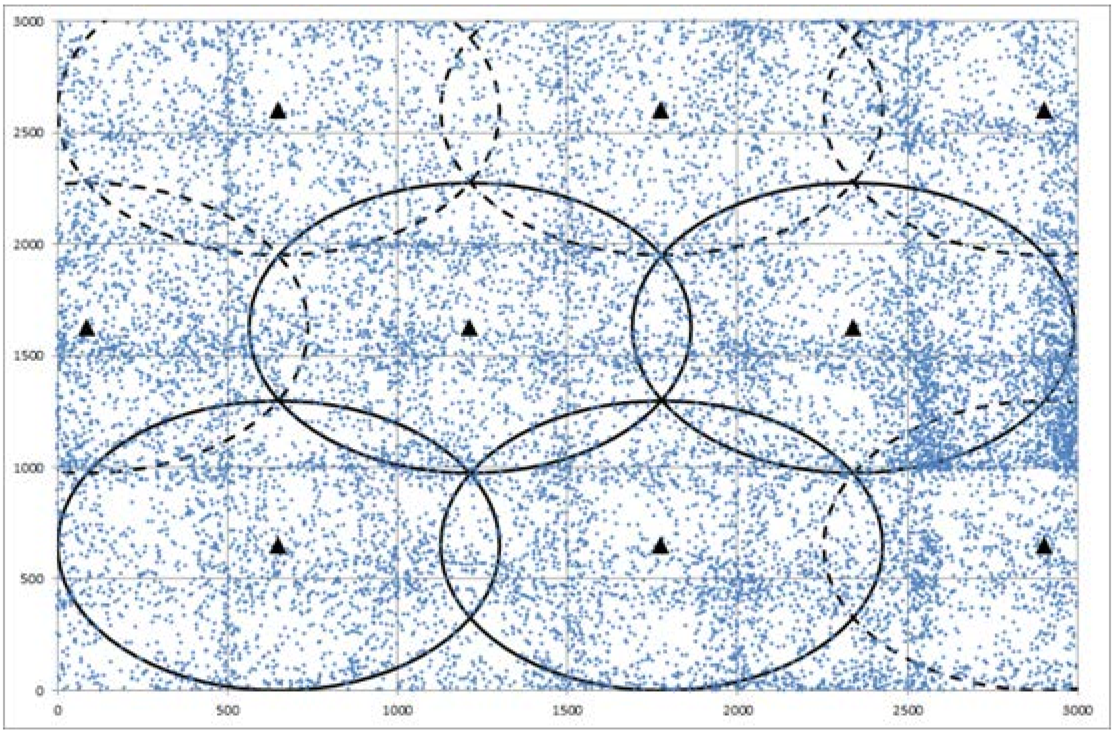


Figure 4.6 (c) The distribution of mobile terminals at Step 100

It can be seen in Figure 4.6 (a) that RWPs were distributed uniformly at the beginning of the simulation. Clusters of RWPs can be seen in Figure 4.6 (b) and 4.6 (c) without any triggered events, which was caused by the bottleneck points of the geographic layout.

In the simulation, a number of BSs were deployed in the simulation area. Considering the configuration of antennas in the urban area in real life, the coverage radius of the BS is set to 650 metres and the distance between two BSs is set to 1.1 kilometres, as shown in Figure 4.6. With the above settings, the coverage area of four BSs could be contained completely by the simulation area. Those coverage areas are represented by real-line circles in Figure 4.6. When evaluating the prediction performance with AUDGs, four sets of AUDGs were deployed in the area covered by the real-line circles, and the predicted user distributions in that area were analysed. The rest part of the simulation area was not taken into account when evaluating the prediction performance with AUDGs.

Figure 4.7 shows the simulation results of the one-step-ahead prediction with the Order-5 Markov learning model and the STPL model when the number of learning grids was set to 225 $(200m \times 200m)$ and the number of the SUDG was set to 900 $(100m \times 100m)$. Figure 4.8 shows the simulation results evaluated by 4 sets of AUDGs $(36\, sectors \times 20\, bands)$, and other configurations are the same as in Figure 4.8. In order to let the RWPs have enough movement context, predictions were carried out from Step 10.

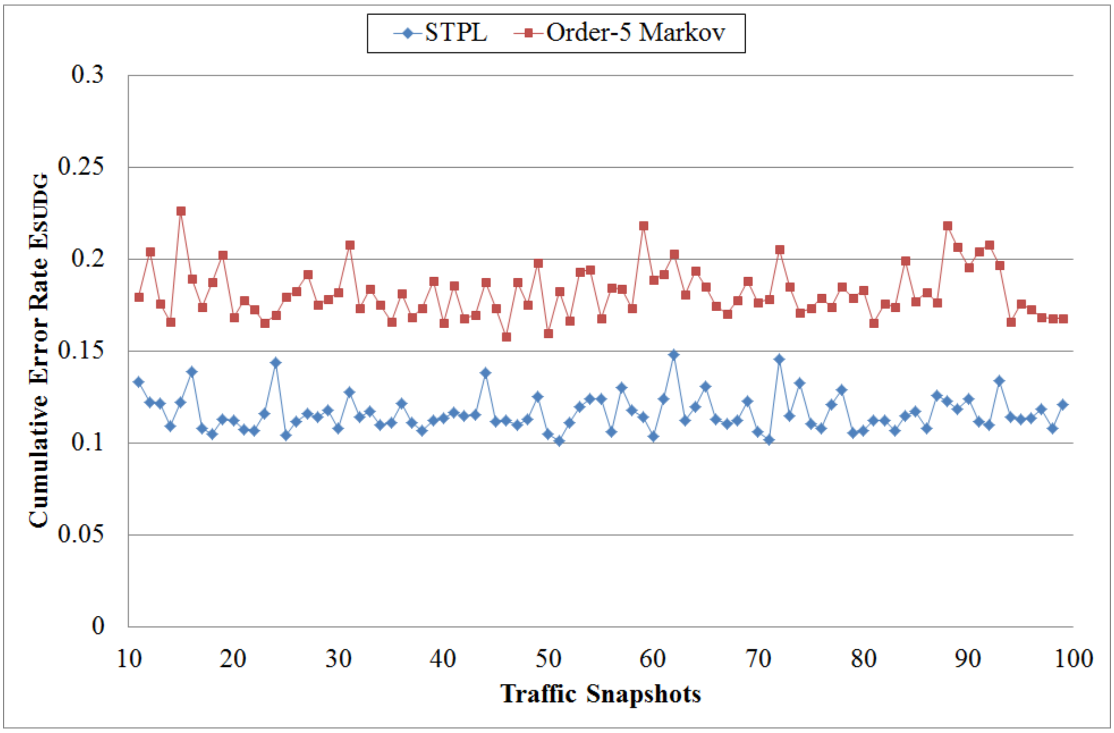


Figure 4.7 The simulation result of one-step-ahead prediction with 225 (15×15) LGs, evaluated by 900 (30×30) SUDGs

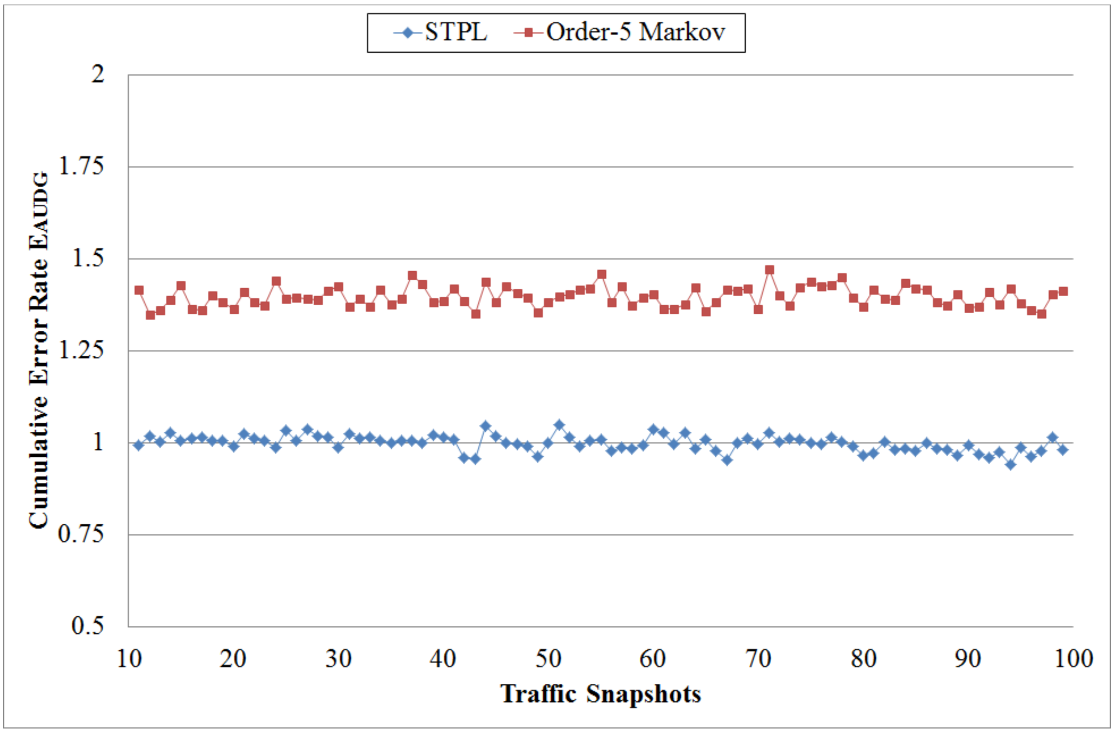


Figure 4.8 The simulation result of one-step-ahead prediction with 225 (15×15) LGs, evaluated by 4 sets of AUDGs (36×20)

As proved in [HKA04] and [SKJH04], simple Markov models perform nearly as well as other more complex methods when learning the movement features the motion trajectories captured in the cellular networks. In this research, not only the performance of using Order-k Markov model to predict intra-cell traffic patterns was evaluated, the performance of the Order-k Markov model was also used as a benchmark when evaluating other models' performance.

In the above two figures, the cumulative error rate of the STPL model is presented by the blue line; the cumulative error rate of the Order-5 Markov predictor is presented by the red line. The *x*-axis represents the identifier of the snapshot; the *y*-axis represents the value of the cumulative error rate. Notice that the error rate used in this research is the Root Mean Square Error (RMSE) rate, as the RMSE is measured in the same units as the data. In the simulation, $E_{SUDG}(i) = 0.1$ means the predicted local user density in a SUDG is on average 10% more or less than the observed value; $E_{AUDG}(i) = 1.0$ means the difference between the predicted population of RWPs and the observed one in an AUDG is on average 1.

It can be seen in Figure 4.7 that the prediction error rates of the STPL model are below 0.15 and the error rates of the Order-5 Markov model are below 0.25 for all the predicted traffic patterns when using SUDGs to compare the predicted local distributions with the observed ones. The average error rates of the STPL model and the Order-5 Markov model for the one-step-ahead prediction are 0.1165 and 0.1826 respectively. When using AUDGs to compare the predicted distributions with the real ones, it can be seen that the prediction error of both the STPL model and the Order-5 Markov model are below 1.5, with the average prediction error 0.9985 and 1.3969 respectively for the next-step prediction.

When carrying out the traffic pattern prediction, one aim is to forecast the location of mobile cluster in the upcoming steps. In the simulation, a SUDG or AUDG was considered as congested containing a mobile cluster, if the population of RWPs in the same SUDG or AUDG is larger than a predefined threshold. Notice that distribution of congested UDGs represents the distribution of mobile clusters. Figure 4.9 and Figure 4.10 show the prediction accuracy in terms of the number of congested UDGs with the STPL model; Figure 4.11 and Figure 4.12 show the prediction accuracy in terms of the number of congested UDGs with the Order-5 Markov model.

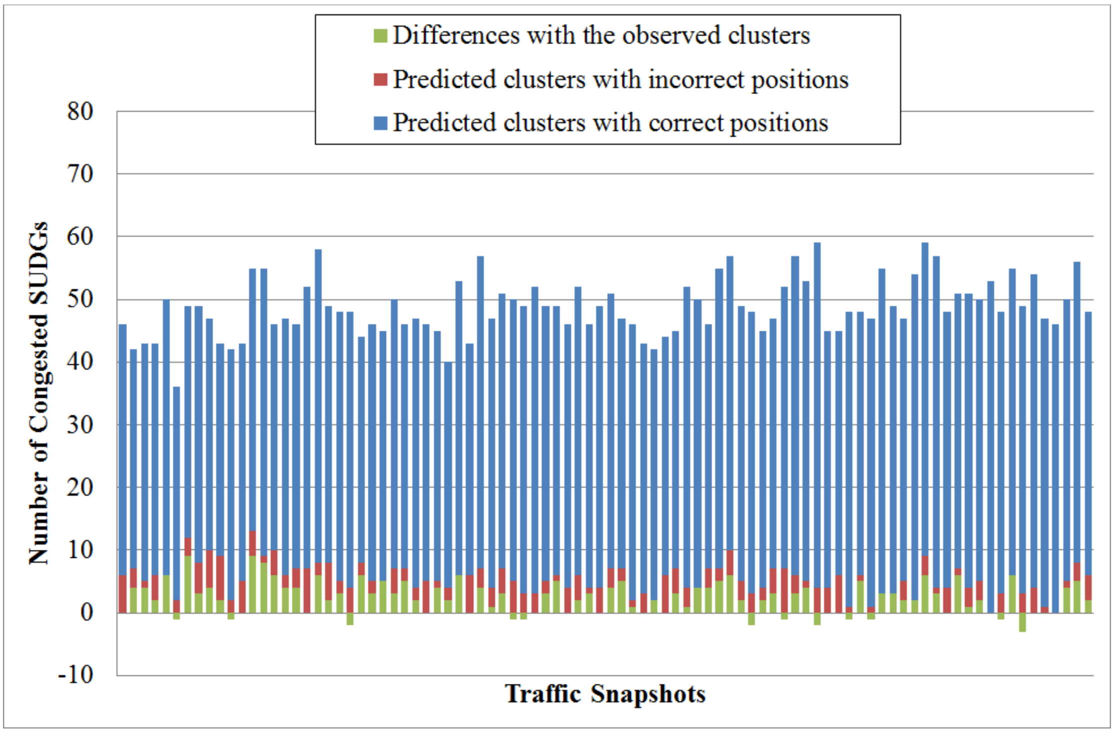


Figure 4.9 Predicted number of congested SUDGs with STPL, evaluated with 900 SUDGs (30 × 30)

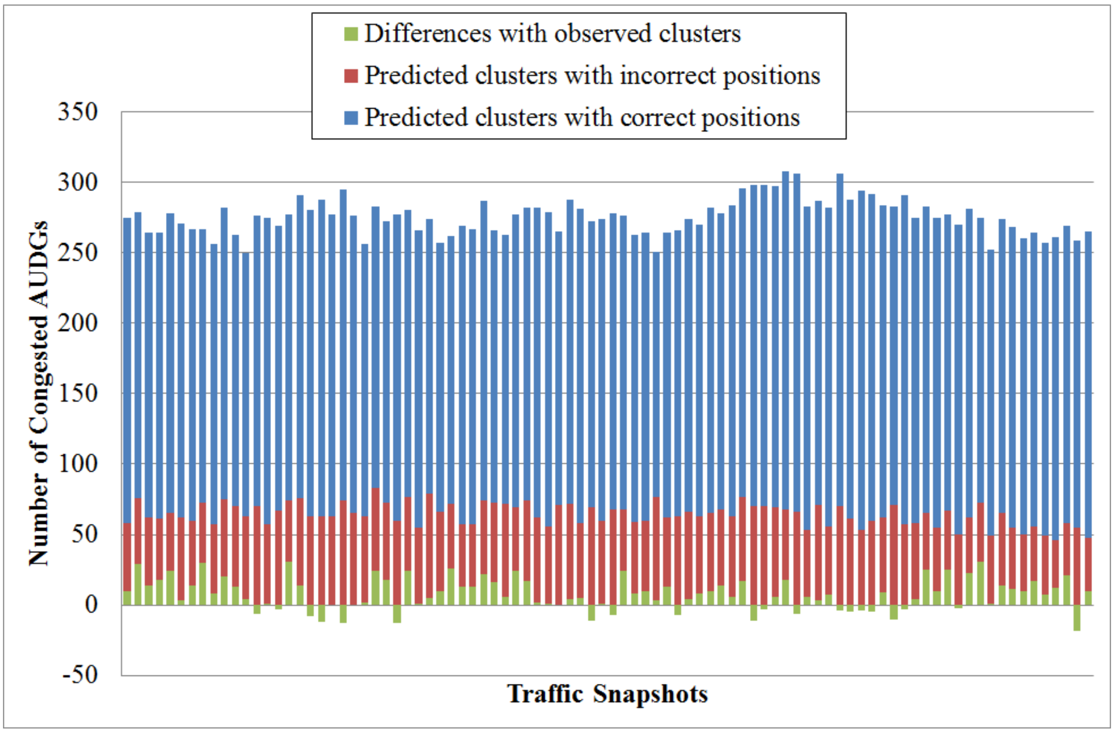


Figure 4.10 Predicted number of congested AUDGs with STPL, evaluated with 4 sets of AUDGs (36 × 20)

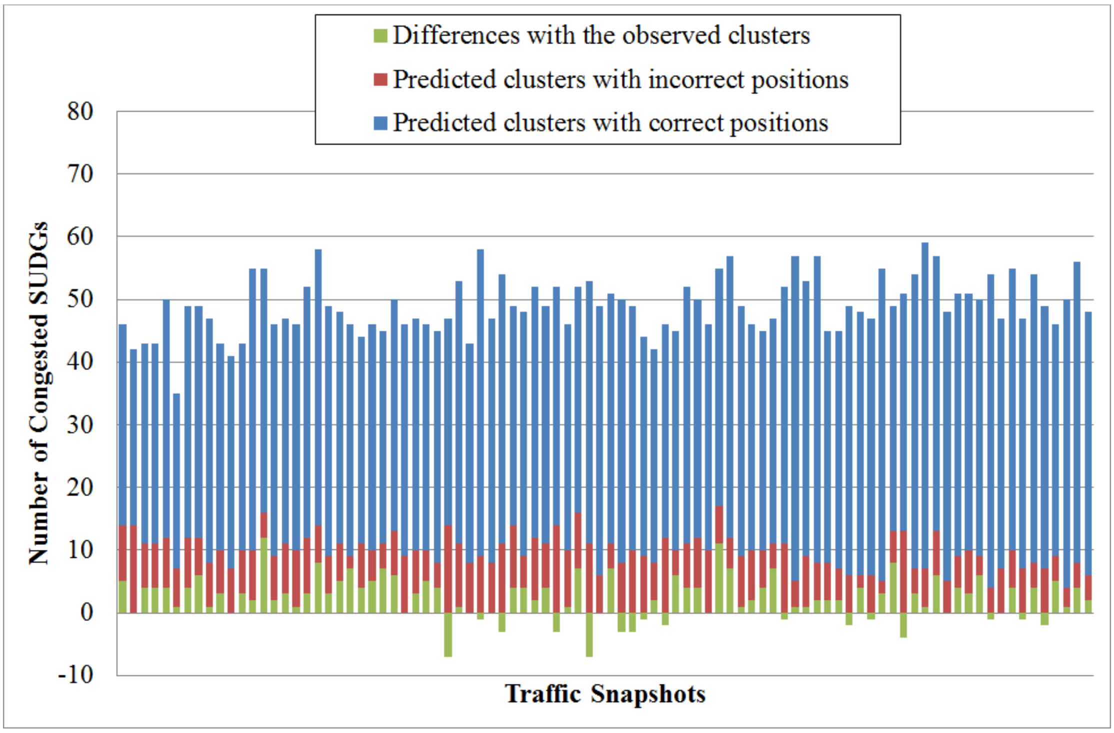


Figure 4.11 Predicted number of congested SUDGs with Order-5 Markov Model, evaluated with 900 SUDGs (30 × 30)

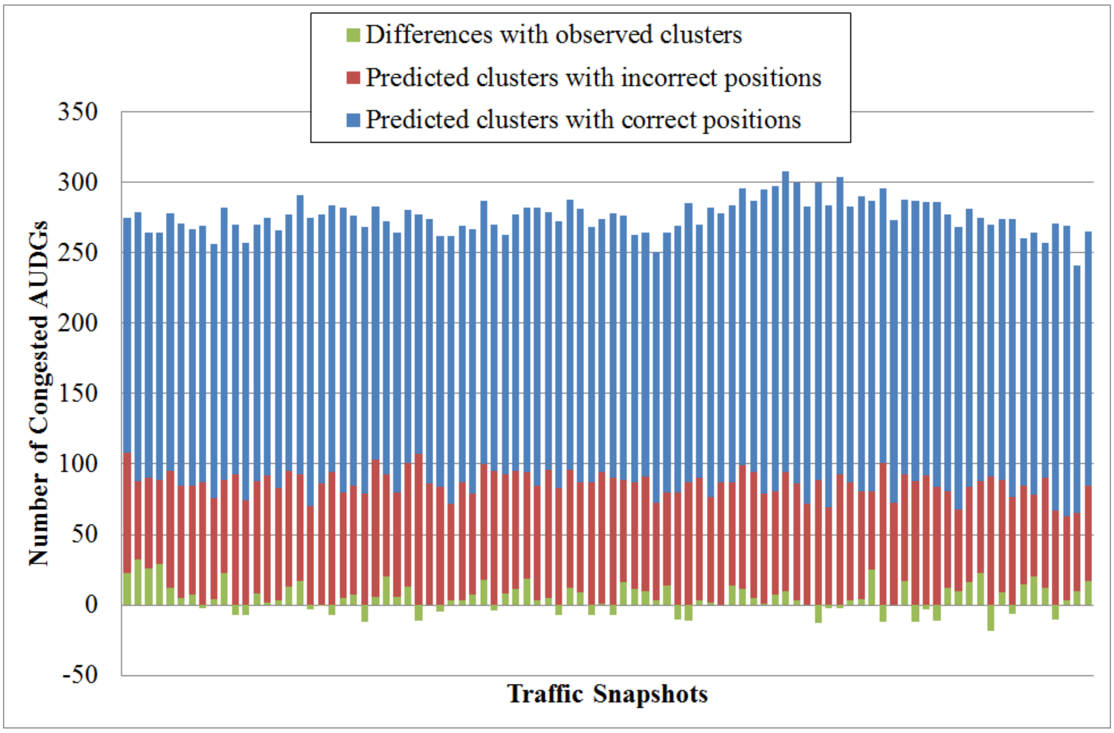


Figure 4.12 Predicted number of congested AUDGs with Order-5 Markov Model, evaluated with 4 sets of AUDGs (36 × 20)

In the above four bar charts, the blue bars represent the number of congested UDGs whose locations are predicted correctly; the red bars represent the number of congested UDGs whose locations are defective in the predicted patterns; the green bars represent the number of congested UDGs that are missed in the predicted patterns. In each snapshot, the sum of the blue bars and the red bars represents the total number of congested UDGs in the prediction; the sum of the three bars represents the total number of congested UDGs in the observation. If the green bar is seen below the x-axis, it means the total number of the congested UDGs in the prediction is larger than the observed ones. With the distribution of congested UDGs, the location and scale of mobile clusters can be easily determined.

It can be seen from Figure 4.9 and Figure 4.11 that most congested UDG locations can be predicted when evaluating the results with 900 SUDGs. The average accuracies with STPL model and the Order-5 Markov model are 88.56% and 80.69% respectively. When using 4 sets of AUDGs to evaluate the results, the proportion of the red bars grows, but the average accuracy is still acceptable, as the granularity of four sets AUDGs is much higher than 900 SUDGs. The average accuracies with STPL model and the Order-5 Markov model are 77.16% and 69.28% respectively in Figure 4.10 and Figure 4.12.

According the above simulation results, it can be seen that the predictions made with the STPL model are more accurate than with the Markov model. One reason is that more details are taken into account when learning the local movement characteristics with the STPL model, such as the local road topology, the probability of movement direction changing, and the varying tendency of movement speed. By contrast, the Order-5 Markov model makes prediction only based on the movement context comprised of 5 most recent observed locations. Another reason is that prediction results with the STPL model are coordinates of mobile users, which can present the traffic distribution directly. With the Order-5 Markov model, prediction results are the next LGs mobile users would visit. Then mobile users are uniformly distributed in the estimated next location to present the traffic distribution, which affects the accuracy of the distribution.

In order to test the sensitivity of the two learning models to the granularity of the learning grid, simulations were carried out with different numbers of learning grids applied in the pattern learning phase and pattern prediction phase. Figure 4.13 and Figure 4.14 show the relevant results.

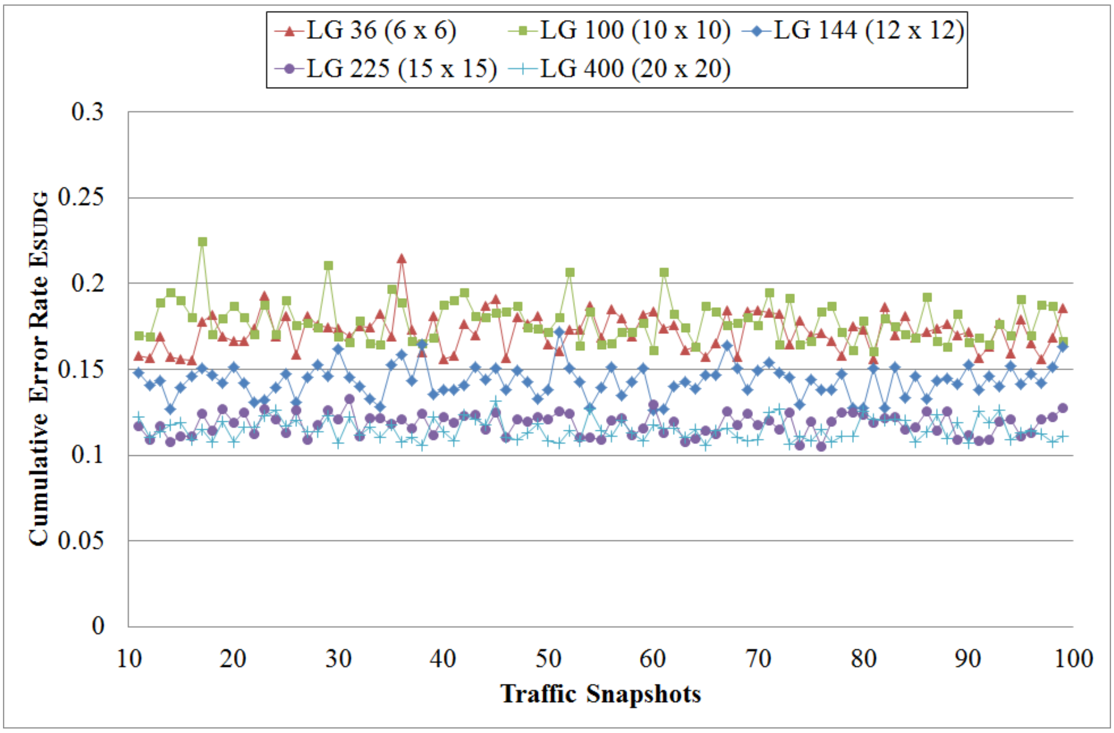


Figure 4.13 Comparing number of learning grid using STPL model

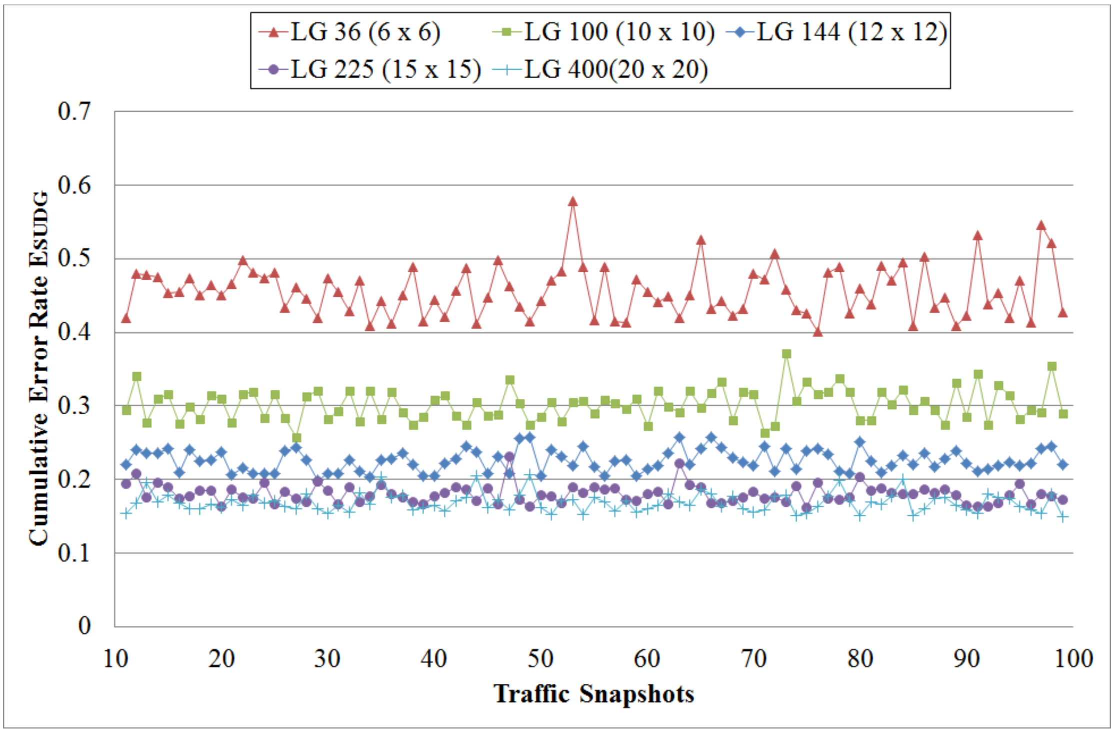


Figure 4.14 Comparing number of learning grid using Order-5 Markov model

It can be seen in Figure 4.13 that the performance differences when deploying different number of learning grids are not significant. The highest rate, which is 0.1781, can be seen when 100 learning grids were used to learn the traffic pattern; the lowest rate, which is 0.1153, is obtained when 400 learning grids were deployed. The traffic patterns learnt by the learning grid with small size contain less generalized features, which could reflect the real local movement regularity better. However, one exception can be seen when 36 learning grids are used. This is because the size of the learning grid is exactly the same as the size of the GGE, and the learning patterns extracted by each learning grid are from the same GGE. In the learning and prediction phase of the simulation, the geographic layout of the simulation area is treated as unknown information. Thus, this exception cannot be generalized.

Figure 4.14 indicates that the performance of the Order-5 Markov model is correlated with the granularity of the learning grid. The more learning grids that are deployed, the lower the error rate that can be obtained. When using the Order-*k* Markov model, both the context trajectories and the prediction results are represented by the IDs of the learning grid. The low granularity of the learning grid could affect the accuracy of the prediction. With 400 learning grids, the average error rate is 0.1689, which is still higher than the error rate when making predictions with the STPL model under the same configuration.

Figure 4.15 and Figure 4.16 show the variation of the error rate when analysing the prediction results with different granularity of the SUDGs and AUDGs.

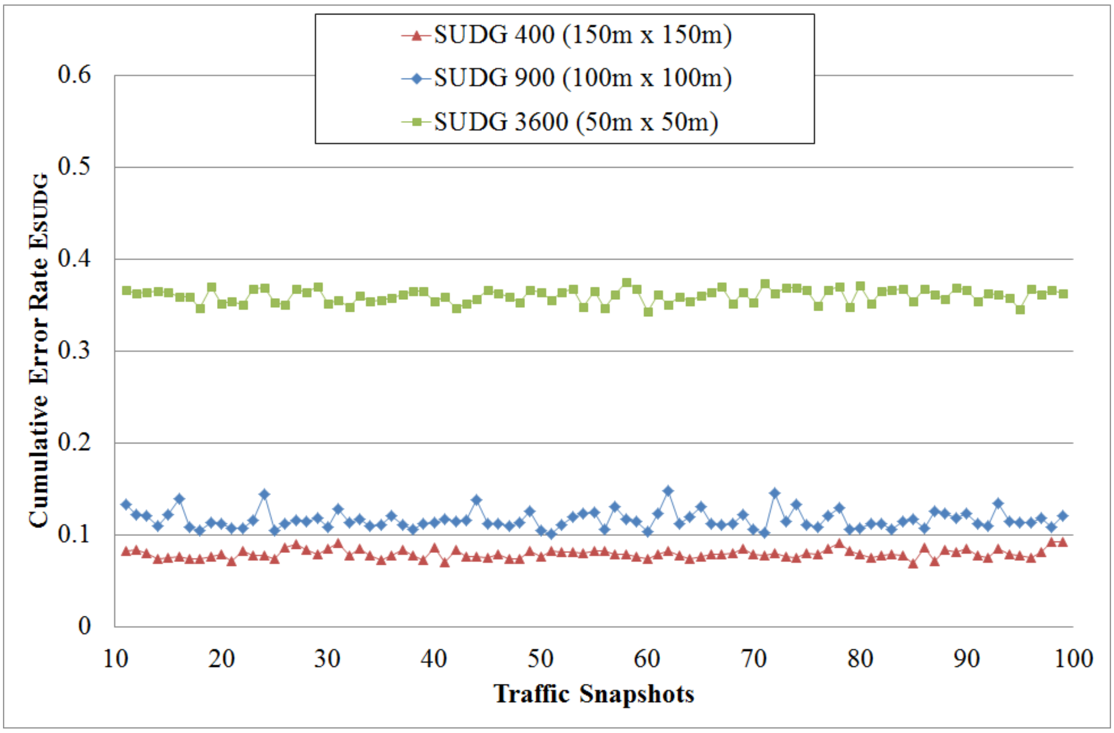


Figure 4.15 Comparing number of SUDGs

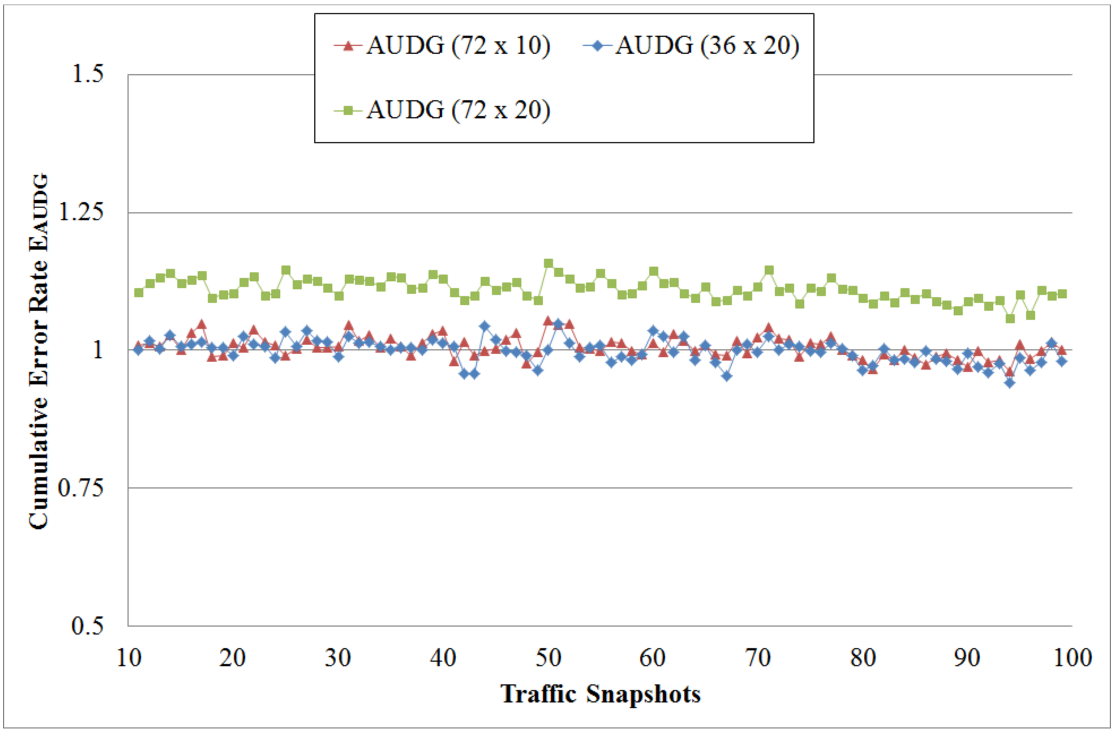


Figure 4.16 Comparing number of AUDGs

Figure 4.15 indicates that the more the UDGs are deployed, the higher the error rate that is experienced. With 3600 SUDGs, the average error rates can reach 0.3602. Figure 4.16 shows the cumulative error with high granularity of AUDG does not significantly increase. According to the result, the average population difference when applying four sets of 72×20 AUDGs is 1.1109, which is only 12% higher than the average difference when the granularity of AUDG is set to 36×20. However, this does not mean that the error rate under high granularity of AUDG is as good as the rate with low granularity. When using 72×20 AUDGs, the average population of the RWP in most grids is less than 1. Thus, one more RWP in such grids might lead to large error rate.

Notice that the same granularity of the AUDG, 72×20, 36×20 , and 72×10 were used in Yao's work [Yao07] to match the observed congestion patterns to the pattern-solution pairs recorded in the case library. As mentioned in [Yao07], the CBR matching algorithm is not sensitive to the granularity, as the performance with 36×20 granularity matching is very close to the performance with 72×20

granularity matching. According to results shown in Figure 4.16, the average population difference is 0.9985 when applying the 36×20 AUDG sets. With the population difference around 1.0, the predicted traffic pattern could be used in Yao's CBR matching model to pick out the proper antenna beamforming for upcoming traffic patterns.

Figure 4.17 shows the variation of the prediction performance with different *k* values in the Order-*k* Markov model. It can be seen that the cumulative error rate decreases along with increasing *k* value. However, there is no significant improvement in the prediction performance after the value of *k* reaches 5.

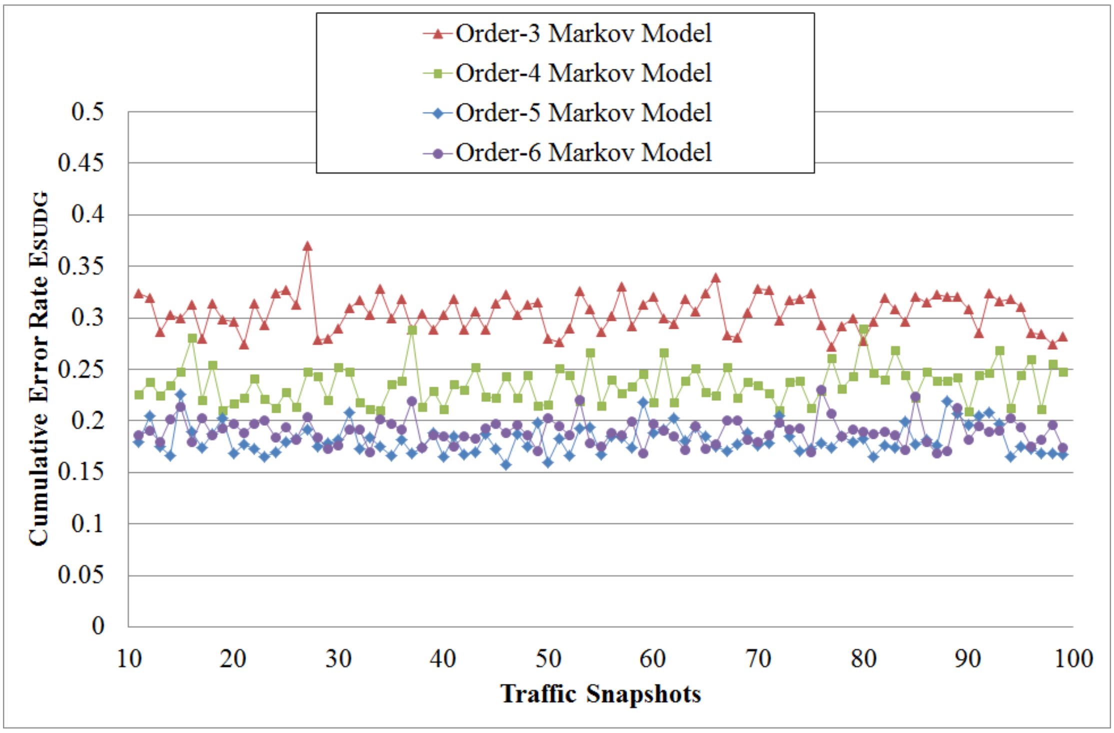


Figure 4.17 Comparing the *k* values

Figure 4.18 shows the variation of the average error rate when performing the predictions from 1-step ahead up to 6-steps ahead with the two proposed learning models.

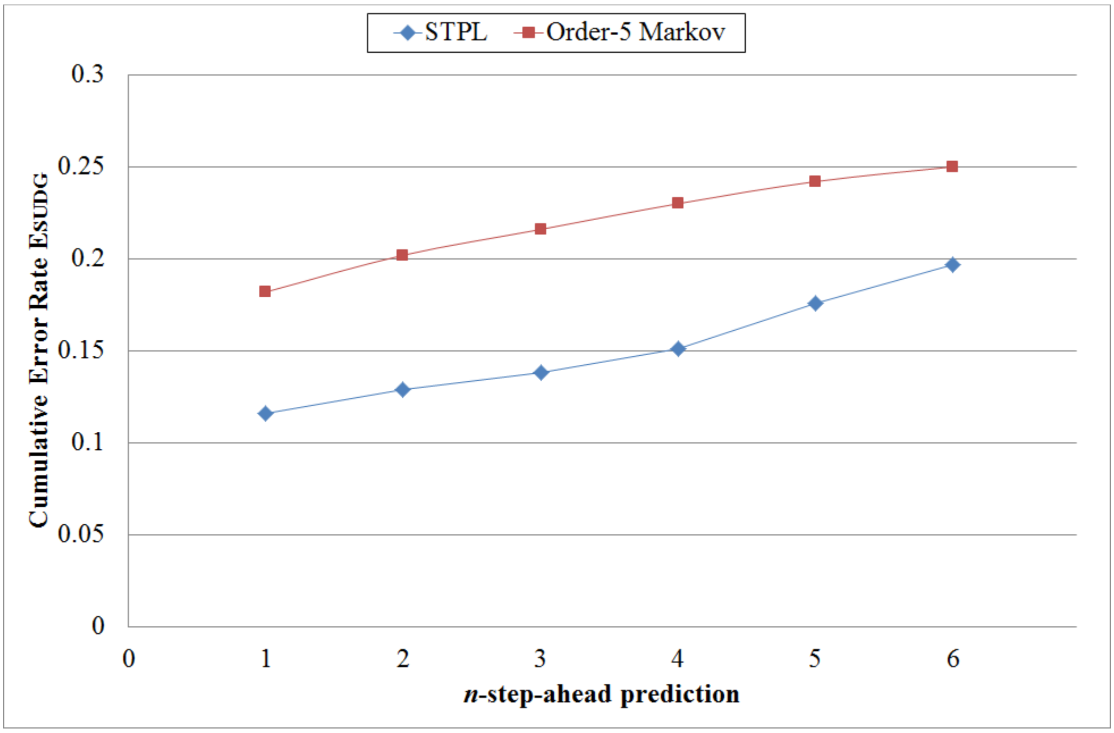


Figure 4.18 Comparing the *n* values

4.3.2 Simulation results for the mobile cluster forming scenario

In this simulation, traffic events were introduced to help form mobile clusters in the simulation area. Similar to the simulation presented in the previous section, traffic patterns were learnt from the case library by 225 learning grids before the simulation and 100 traffic snapshots were generated as the observed traffic patterns in the simulation. 15000 RWPs were deployed uniformly into the simulation area. Upcoming patterns were predicted at each step based on previous observed patterns. Six GGEs were embedded with the same traffic event which restricted the movement speed from Step 16 to Step 55. Figure 4.19 (a) (b) (c) and (d) present the observed traffic patterns at Step 25, 35, 55 and 65 respectively.

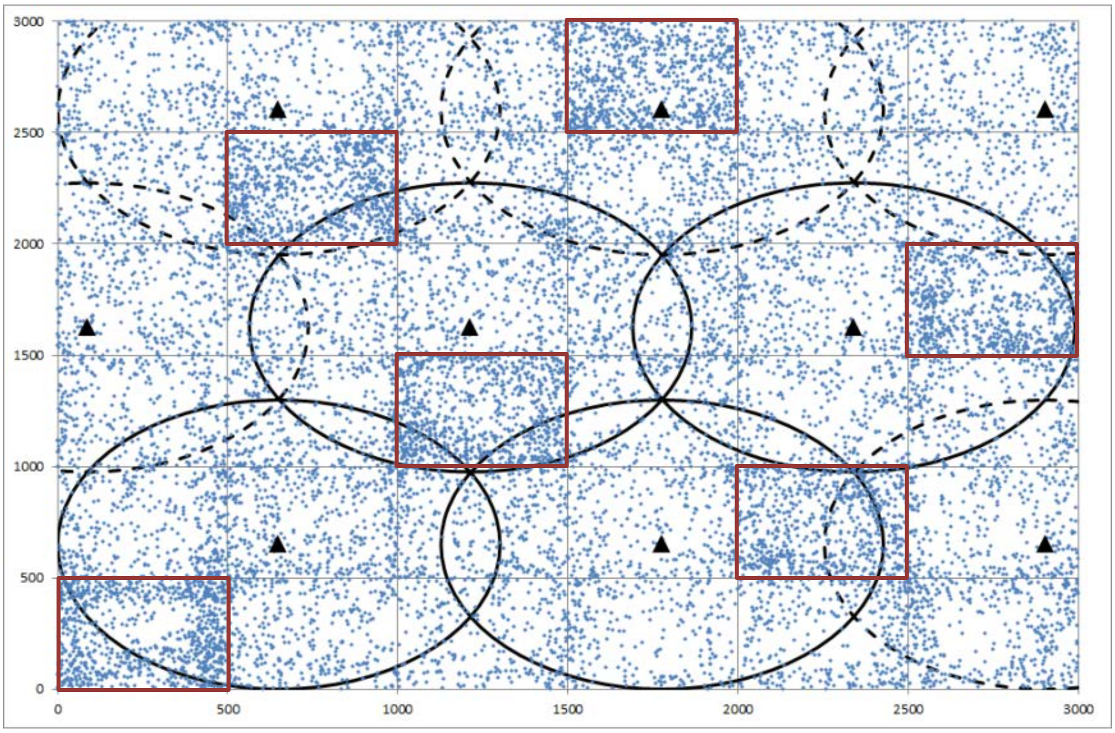


Figure 4.19 (a) Traffic pattern observation at Step 25

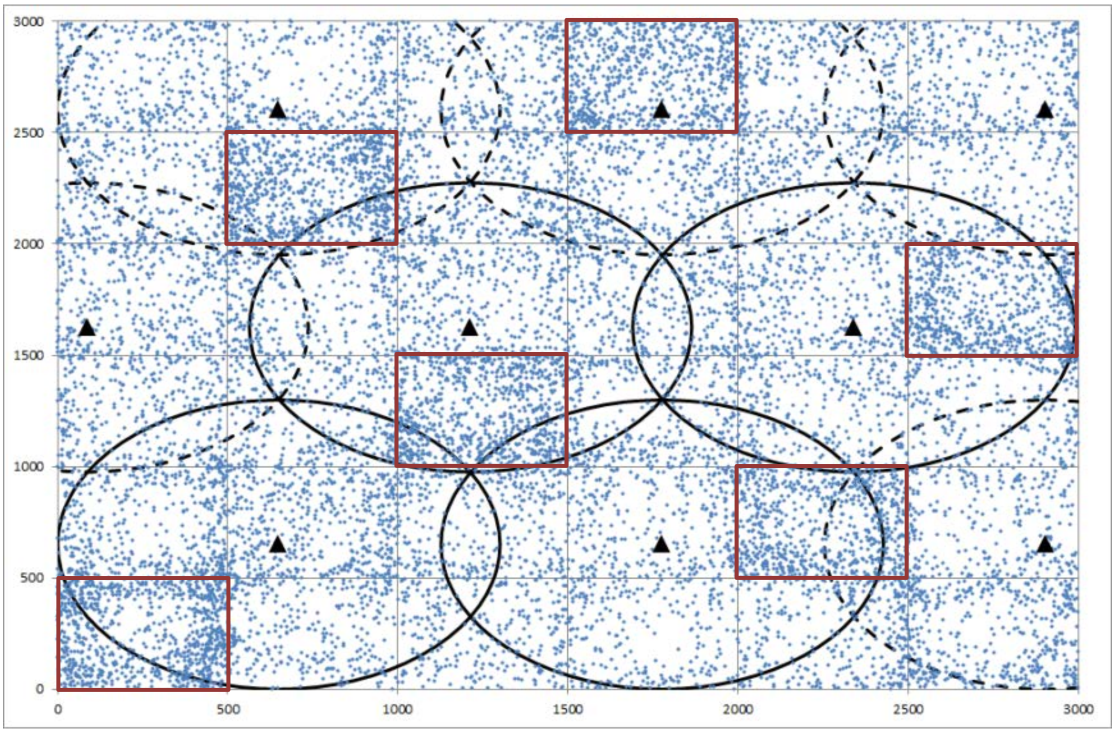


Figure 4.19 (b) Traffic pattern observation at Step 35

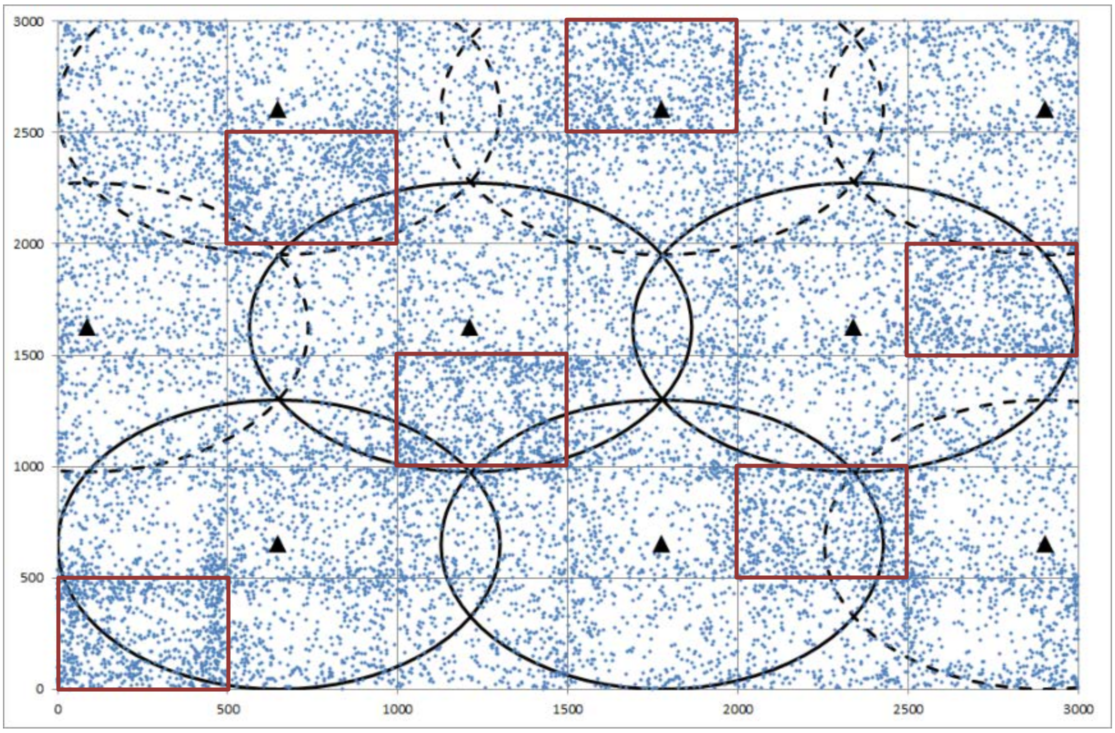


Figure 4.19 (c) Traffic pattern observation at Step 55

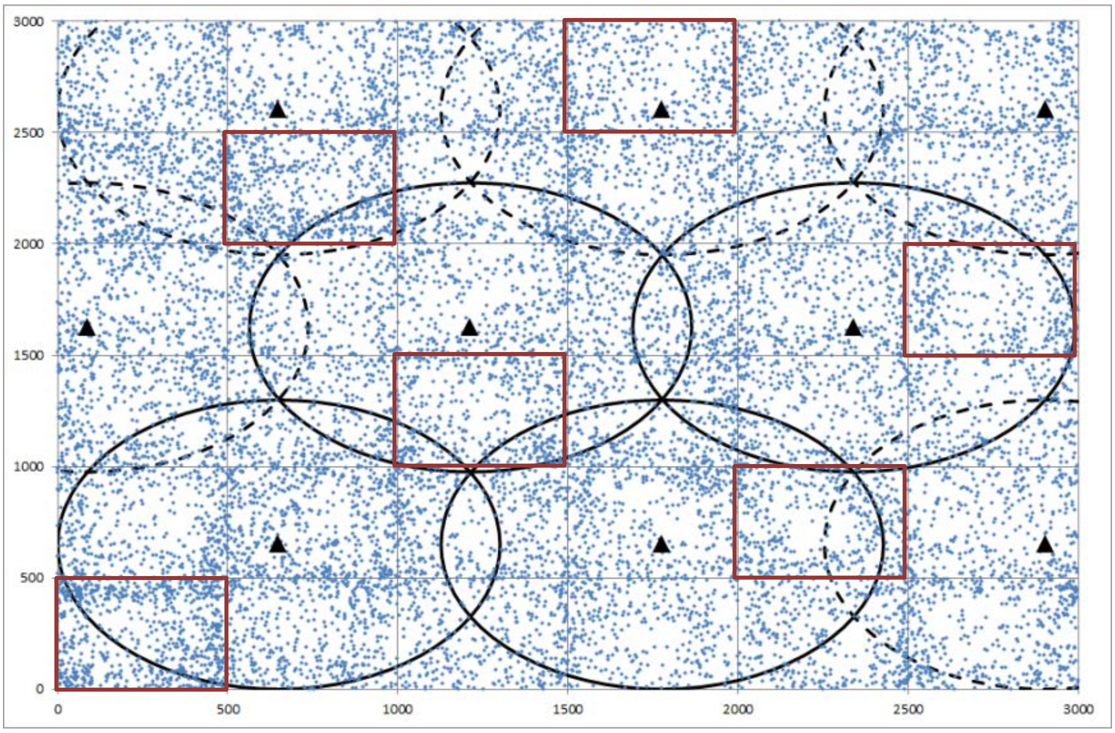


Figure 4.19 (d) Traffic pattern observation at Step 65

In the above figures, areas with the red frame indicate the areas that are affected by traffic events. It can be seen that mobile clusters were formed when events were activated. After the termination of the events, the mobiles clusters dispersed to neighbour GGEs. Figure 4.20 shows the one-step-ahead prediction performance evaluated by 900 SUDGs; Figure 4.21 shows the one-step-ahead prediction performance evaluated by 4 sets of AUDGs (36×20).

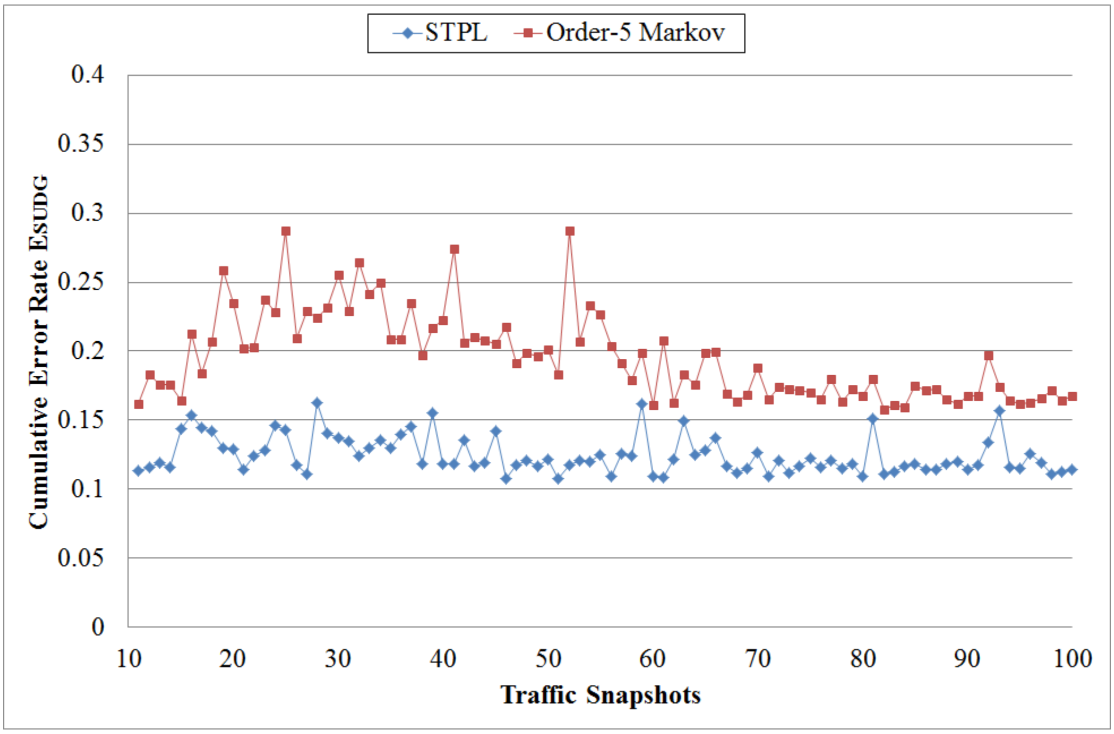


Figure 4.20 Prediction performances in the cluster forming scenario, evaluated by 900 SUDGs (30×30)

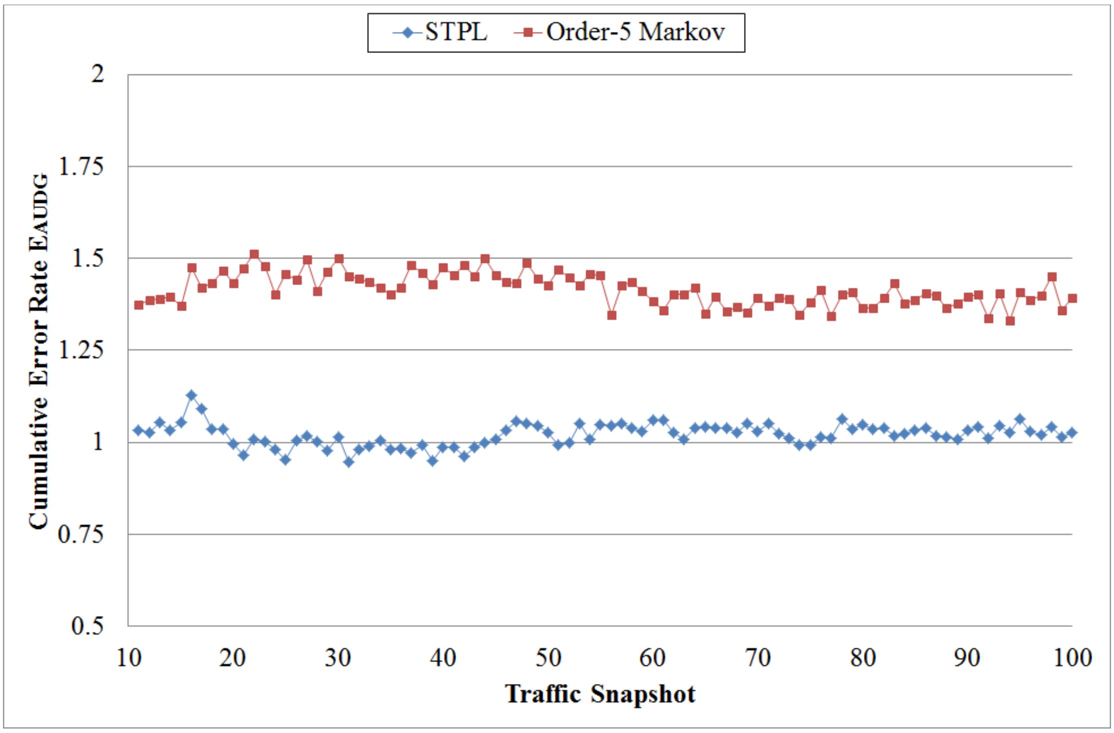


Figure 4.21 Prediction performances in the cluster forming scenario, evaluated by 4 sets of AUDGs (36 × 20)

According to the above two figures, the sudden increase of the error rate with STPL model is seen when making prediction for the traffic pattern of Step 16. Then the error rate drops back to the normal range in the following steps. As mentioned above, traffic events were activated at Step 16, which lets the movement speeds of the RWPs in the affected areas drop tremendously. However, this sudden change could not be known by the STPL model at Step 15. From Step 16, this change could be seen gradually by the STPL model when predicting the movement speed with Equation 3.15. When applying the Order-5 Markov model, the error rate remains higher than normal during the existence of the activated traffic event. If a RWP stays in the same GGE for more than four consecutive steps due to the low movement speed, the context trajectory sequence would be a sequence of same GGE IDs, which severely affects the prediction performance. Thus, the STPL model can handle the prediction in the cluster forming scenario better than the Order-5 Markov model, and the reaction speed of the STPL model for movement pattern change is much faster than for the Order-5 Markov model.

Figure 4.22 and Figure 4.23 shows the prediction accuracy in terms of the number of congested UDGs with the STPL model; Figure 4.24 and Figure 4.25 show the prediction accuracy in terms of the number of congested UDGs with the Order-5 Markov model.

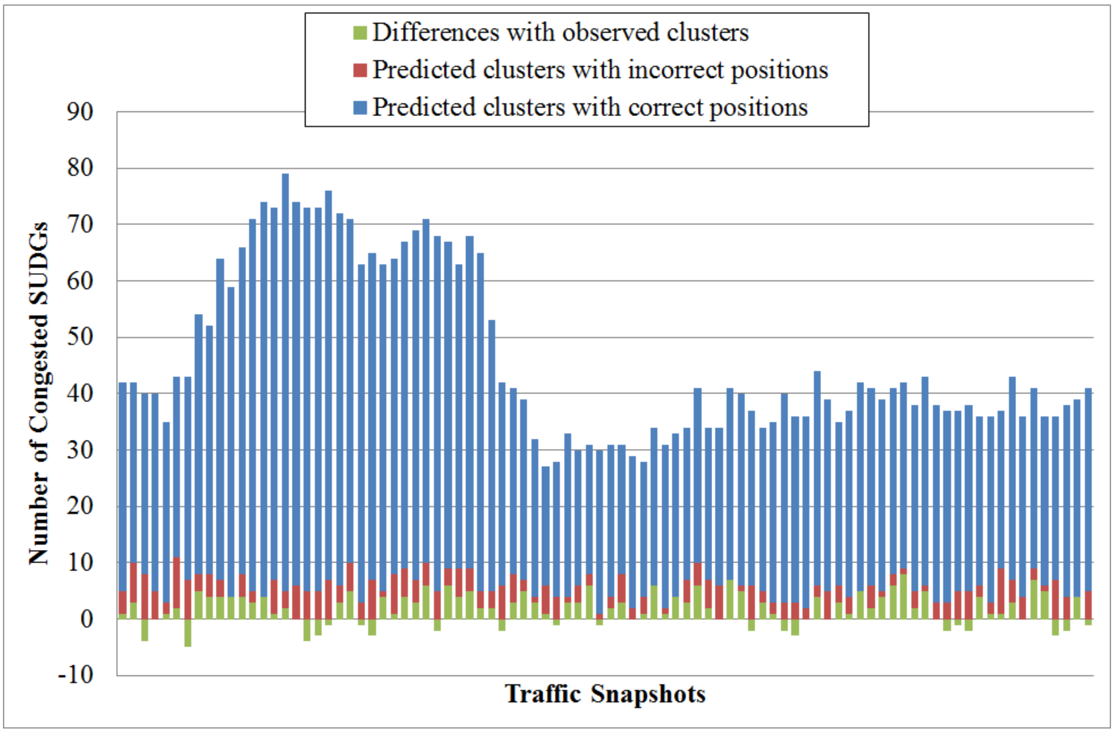


Figure 4.22 Predicted number of congested SUDGs with STPL, evaluated with 900 SUDGs (30 × 30)

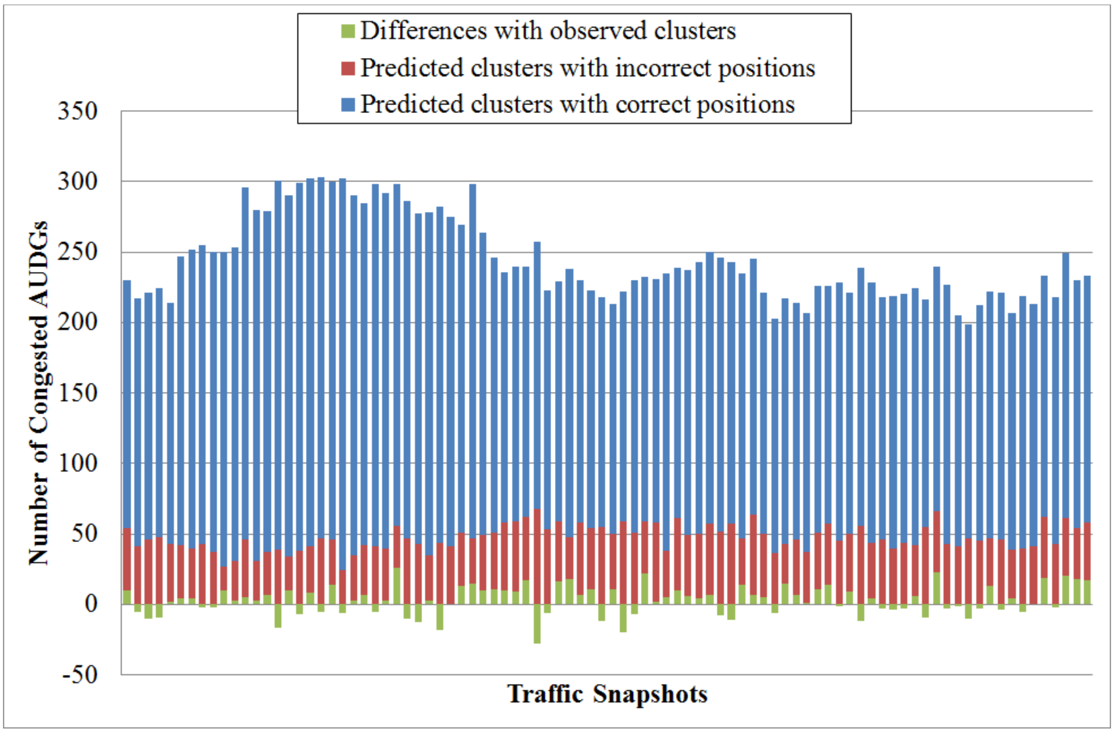


Figure 4.23 Predicted number of congested AUDGs with STPL, evaluated with 4 sets of AUDGs (36 × 20)

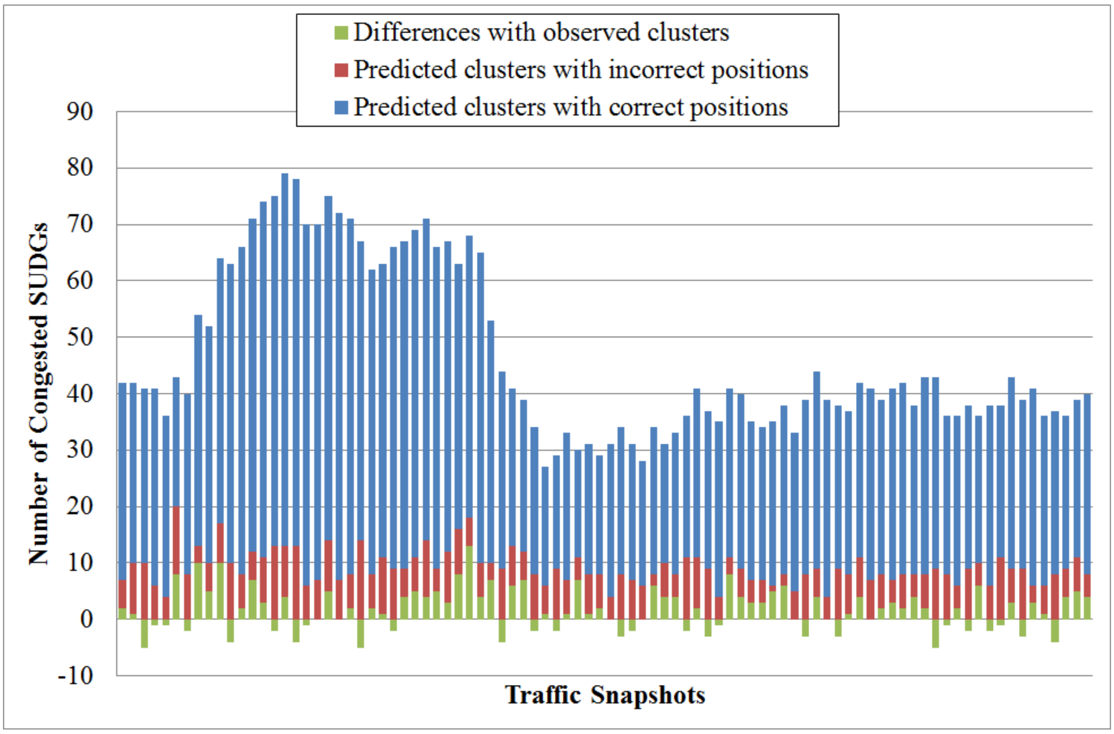


Figure 4.24 Predicted number of congested SUDGs with Order-5 Markov Model, evaluated with 900 SUDGs (30 × 30)

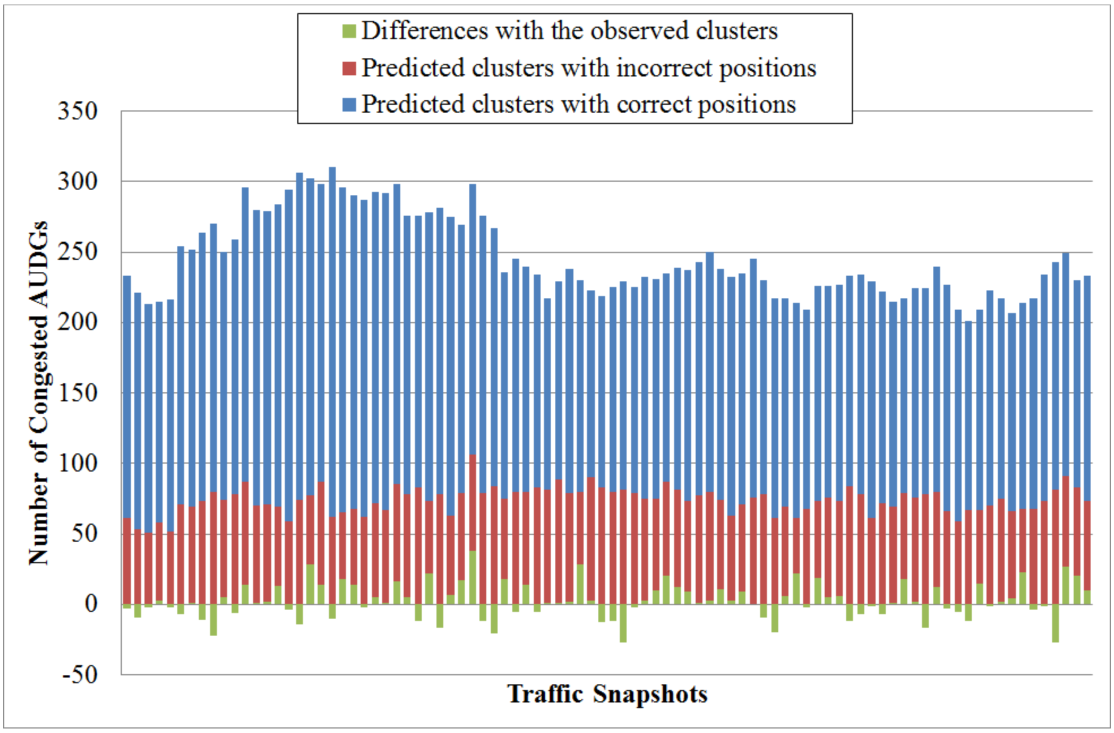


Figure 4.25 Predicted number of congested AUDGs with Order-5 Markov Model, evaluated with 4 sets of AUDGs (36 × 20)

From the above four figures, it can be seen that most of the numbers and locations of the congested UDGs are identified correctly. As mentioned above, well predicted distribution of the congested UDGs indicates the well predicted distribution of mobile clusters. In terms of the mobile cluster prediction in the cluster forming scenario, the performance of both models can be acceptable, and higher prediction accuracy can be obtained with the STPL model. Figure 4.26 (a) and (b) present an example of the observed traffic pattern and predicted traffic pattern at Step 20. The applied SUDGs and one set of AUDG (36 × 20) are deployed in the simulation area and the area with traffic event is highlighted by the red frame. Similar distributions of the observed traffic pattern and the predicted traffic pattern can be seen below.

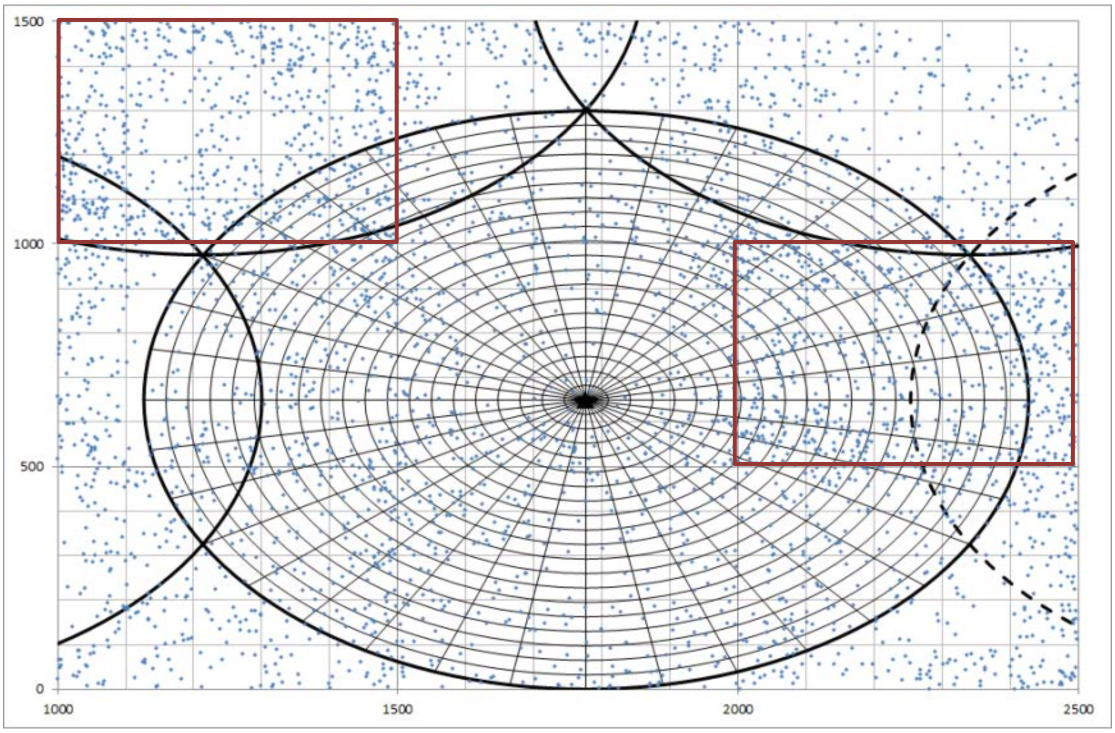


Figure 4.26 (a) Observed traffic pattern at Step 20

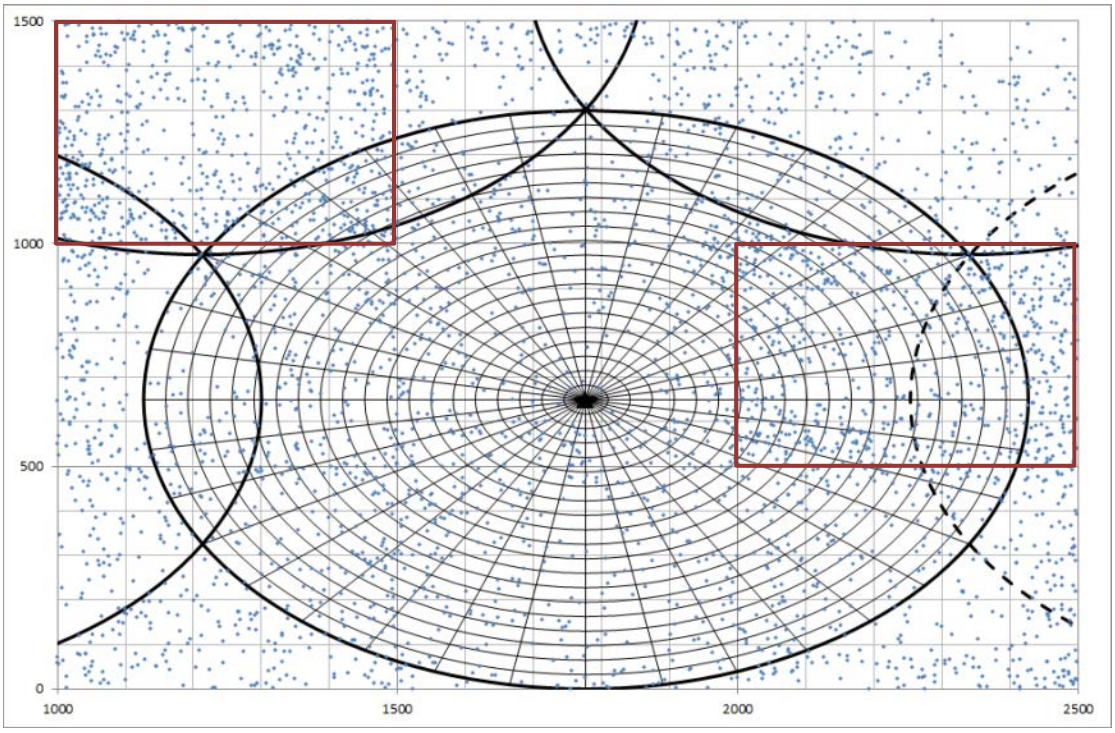


Figure 4.26 (b) Predicted traffic pattern for Step 20

4.3.3 Simulation result in the mobile cluster dispersing scenario

Instead of deploying 15000 RWPs uniformly, 2000 RWPs were uniformly distributed in the whole simulation area and 8000 RWPs were uniformly distributed in the middle of the simulation area, as shown in Figure 4.27 (a). Besides, the four GGEs in the middle were configured manually to force RWPs disperse to the surrounding areas, as shown in Figure 4.27 (b), (c) and (d). This scenario is used to mimic the traffic patterns after a football match or a concert in real life. With this scenario, the capability of the proposed learning models for predicating the movement of mobile clusters is tested. In this simulation, predictions are made from Step 1.

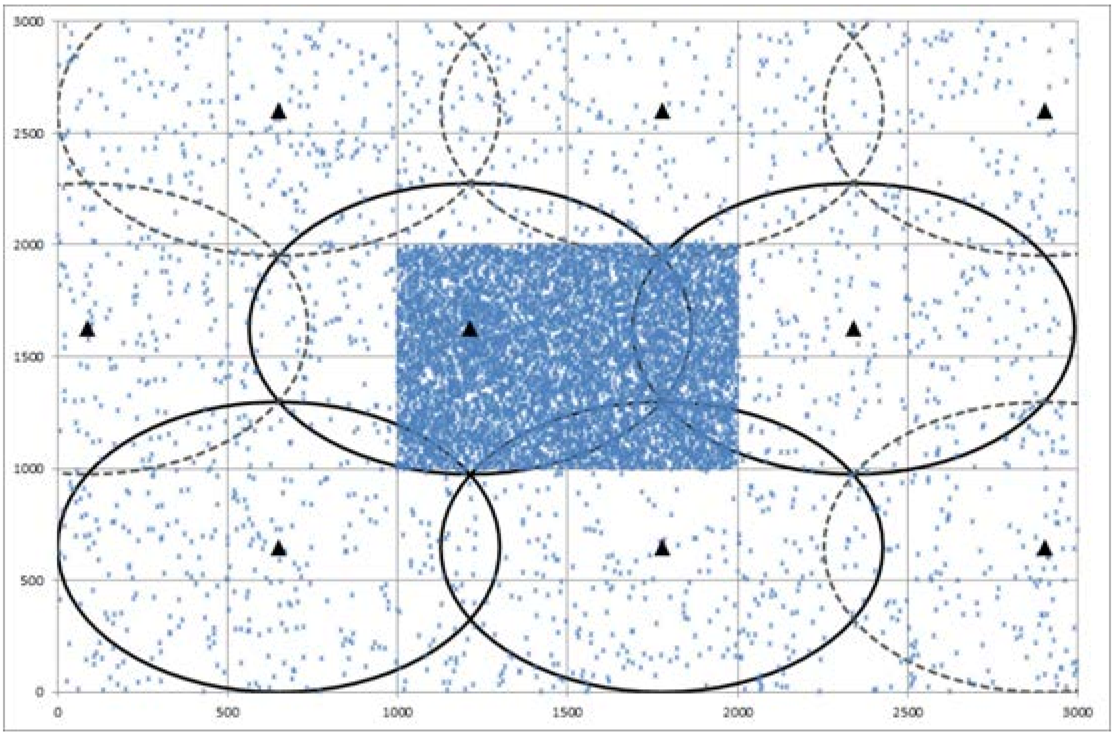


Figure 4.27 (a) The initial position of the RWPs

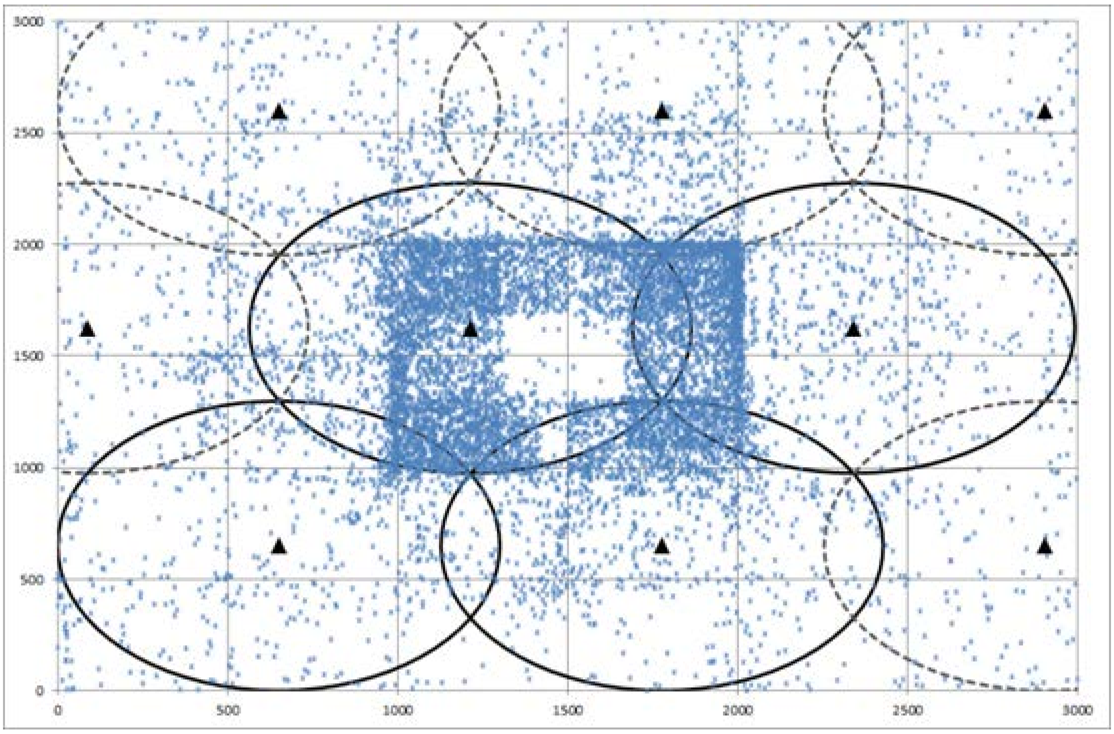


Figure 4.27 (b) Traffic pattern observation at Step 10

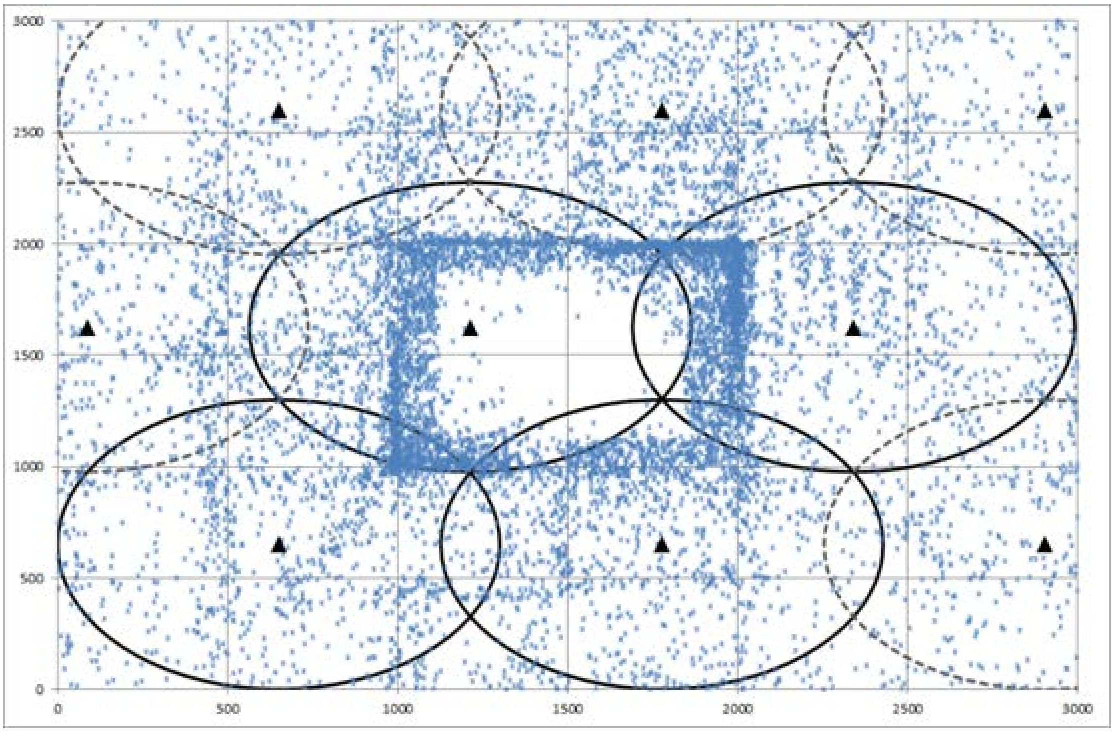


Figure 4.27 (c) Traffic pattern observation at Step 20

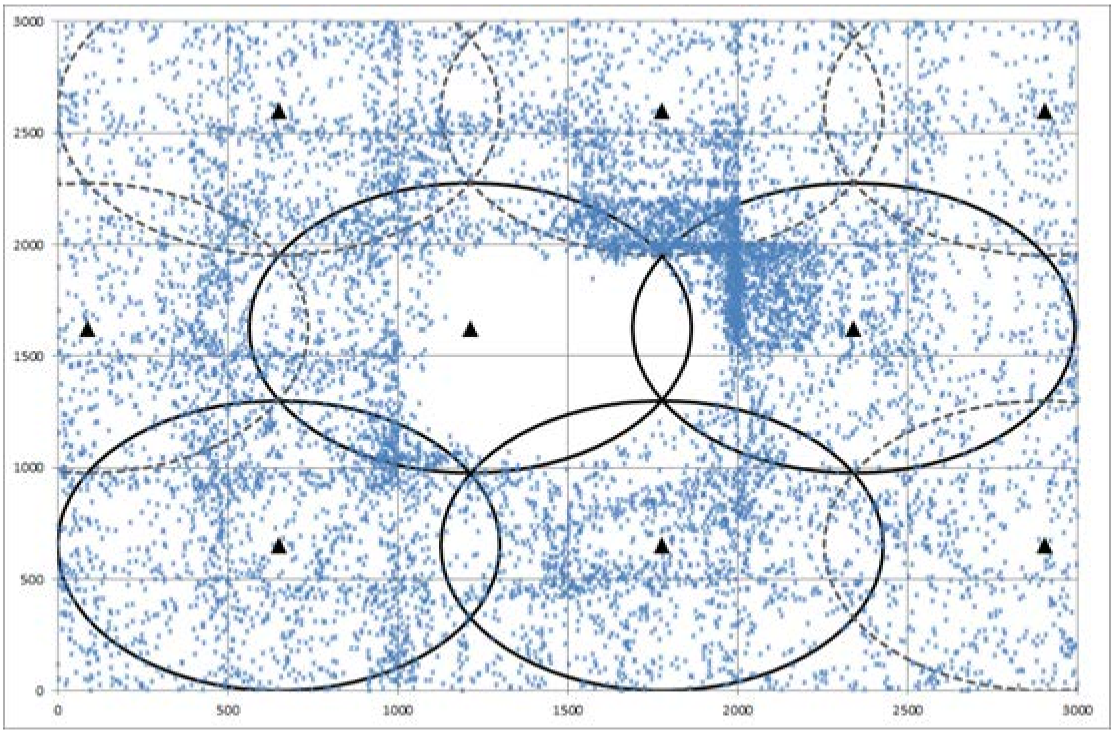


Figure 4.27 (d) Traffic pattern observation at Step 30

It is worth noticing that the movement of the mobile clusters are not the same as in Yao's work [Yao07]. In [Yao07], a number of mobile users are configured to move in the same direction, and these mobile users are treated as moving mobile clusters. The moving cluster used in [Yao07] is illustrated in Figure 4.28.

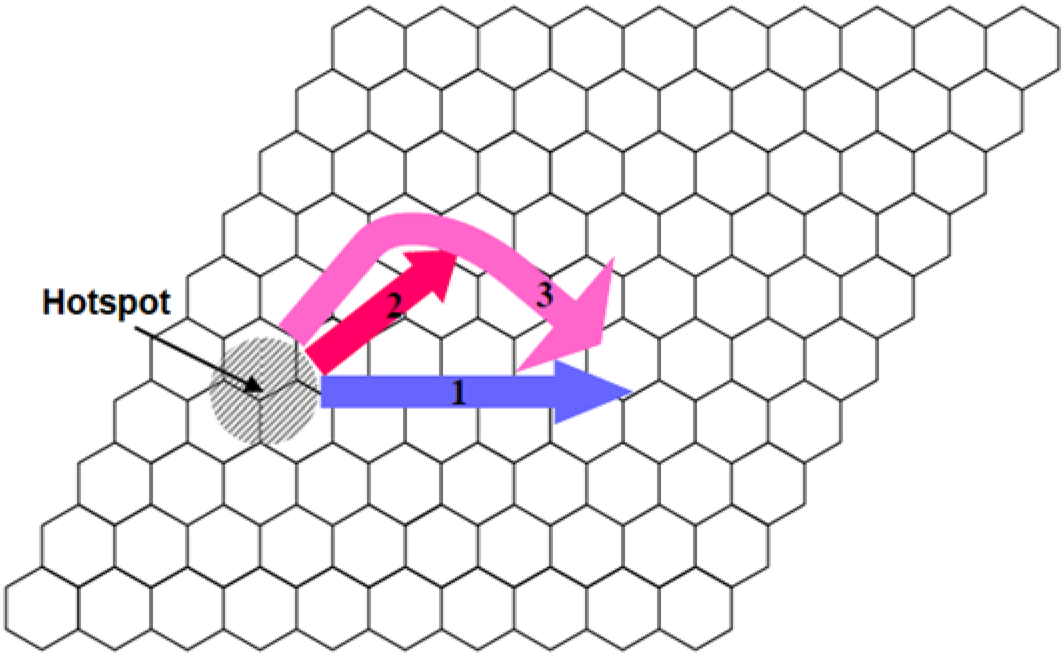


Figure 4.28 Moving hotspot (Figure 4.26 in [Yao07])

In order to introduce more realistic traffic patterns, the movement direction and speed of the mobile clusters in this research are not predefined but are generated according to the walking rules and the relevant geographic conditions at each step. The location change of the mobile clusters represented by SUDGs and AUDGs are shown in Figure 4.29. Notice that the red parts of the simulation area are the location of mobile clusters.

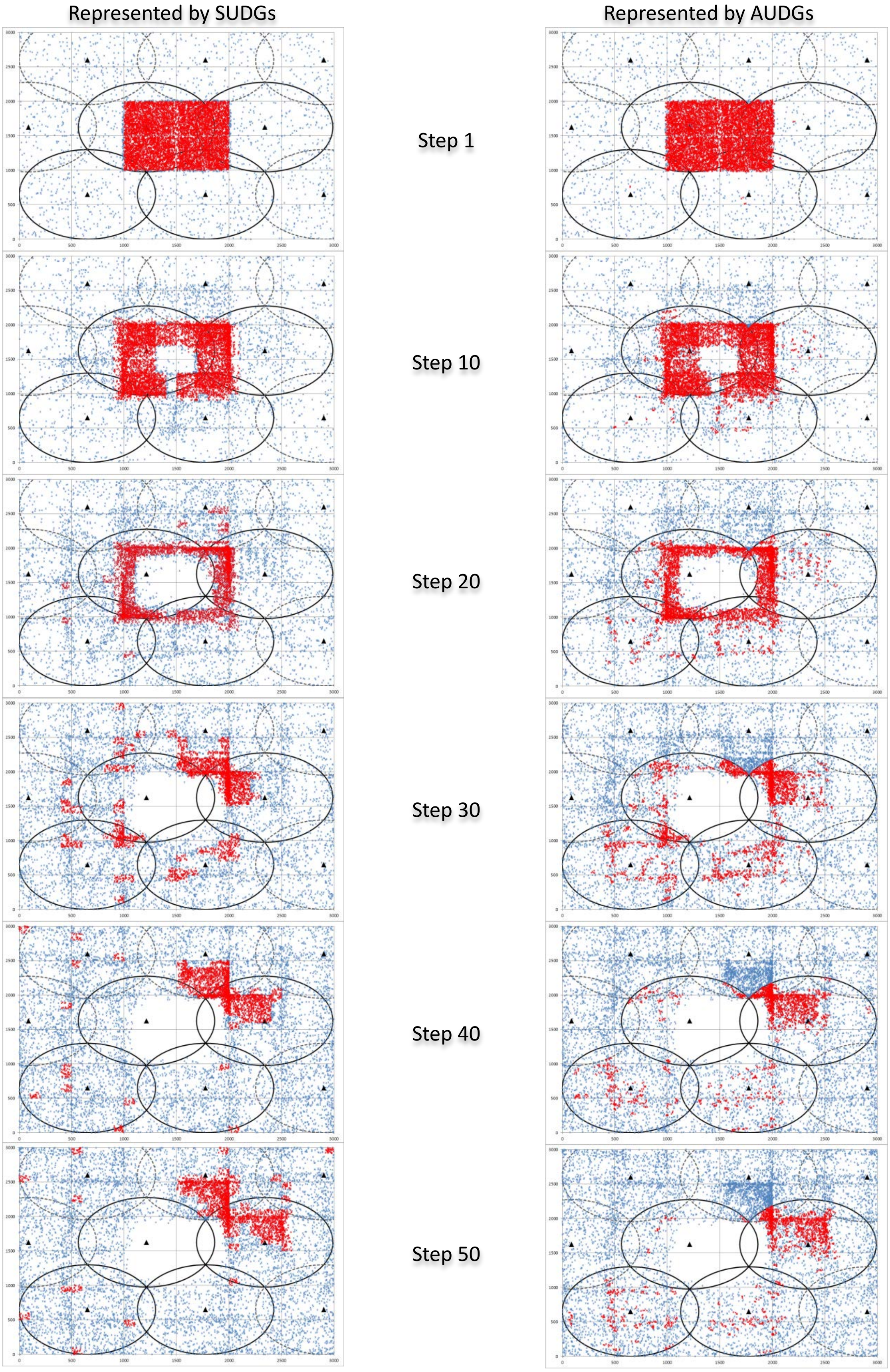


Figure 4.29 The observed disperse of mobile clusters

It can be seen that the 8000 RWPs that were deployed in the centre disperse to the rest of the simulation area in the simulation. Figure 4.30 shows the one-step-ahead prediction performance evaluated with 900 SUDGs; Figure 4.31 shows the one-step-ahead prediction performance evaluated with 4 sets of AUDGs (36×20).

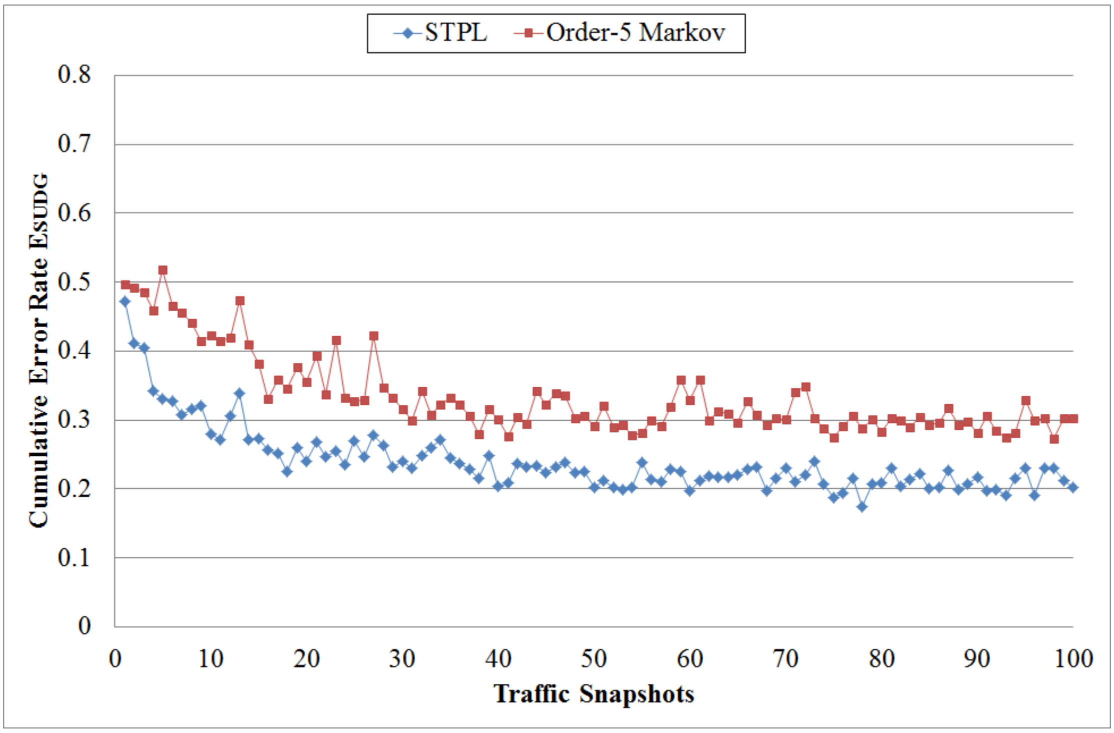


Figure 4.30 Prediction performances in the cluster dispersing scenario, evaluated by 900 SUDGs (30×30)

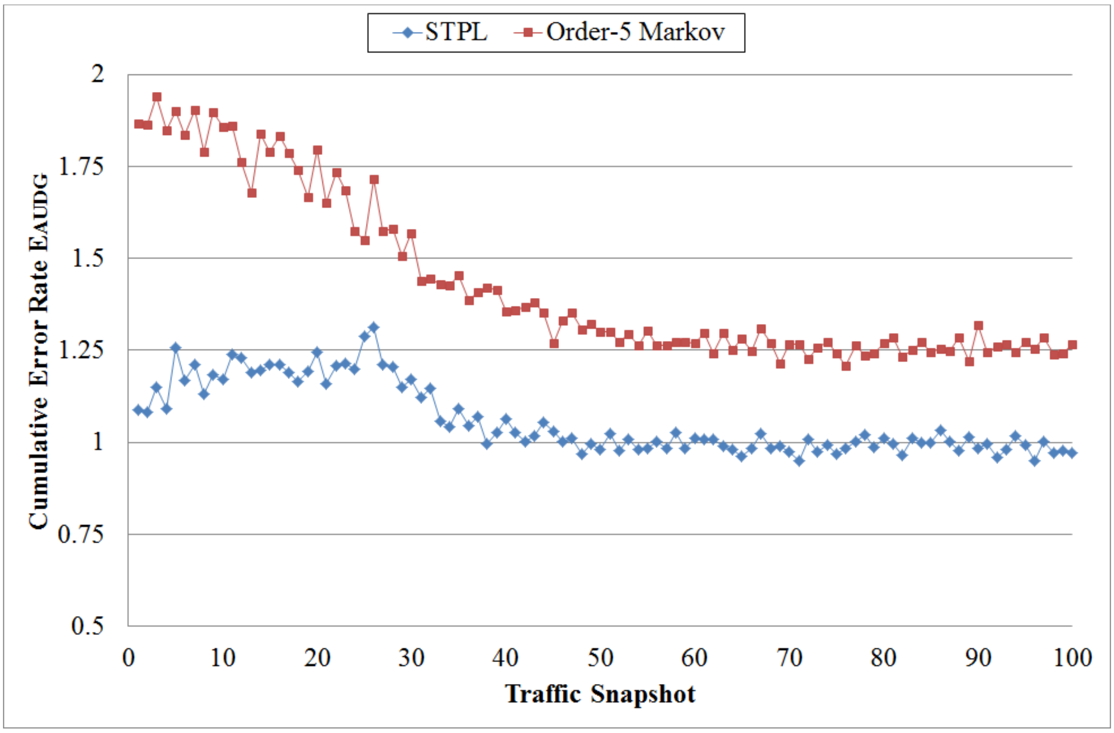


Figure 4.31 Prediction performances in the cluster dispersing scenario, evaluated by 4 sets of AUDGs (36×20)

From the above two figures, it can be seen that the error rates of both the proposed models are considerably higher when making predictions for the first 20 steps. The error rate keeps decreasing until it reaches the normal value range. Notice that the high error rates in the first 20 steps are not caused by the poor prediction performance but the small number of RWPs in most of the UDGs. For the former part of the simulation, most of the RWPs are located in the central part of the simulation area. When using UDGs to measure the local user density, the number of RWPs is less than 5 in most UDGs, as shown in Figure 4.32. Thus, even a small deviation will lead to a large error rate.

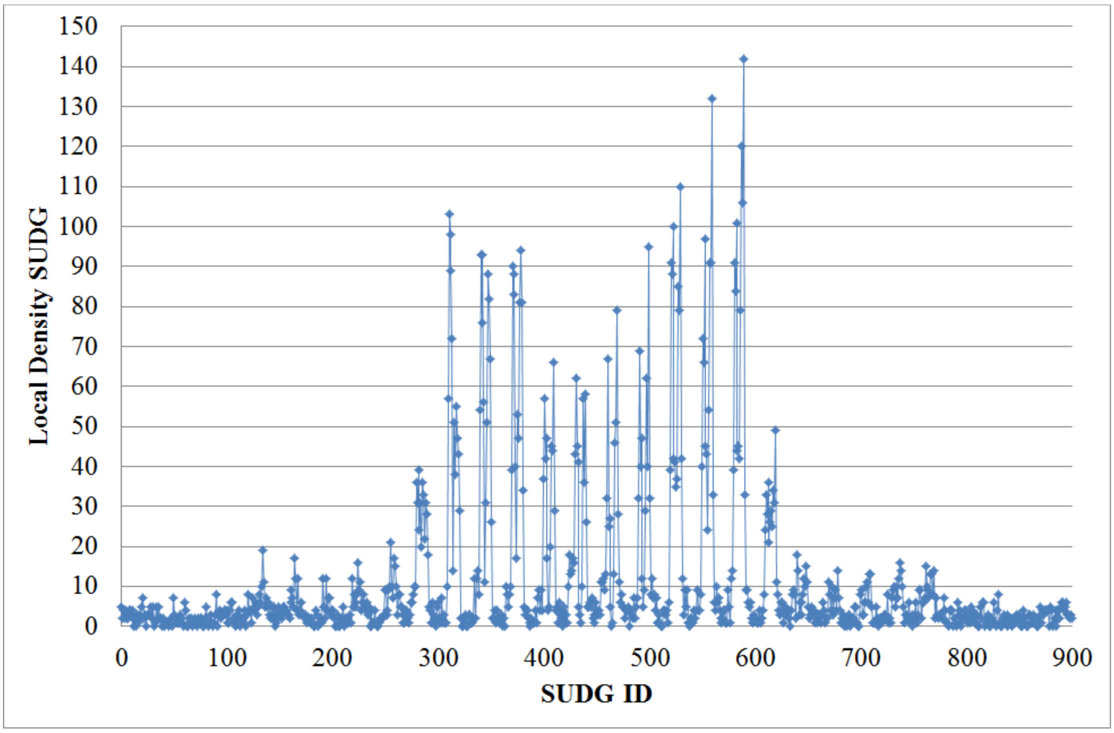


Figure 4.32 Local density of the RWPs at Step 10, evaluated by SUDGs

Figure 4.33 and Figure 4.34 show the prediction accuracy in terms of the number of congested UDGs with the STPL model; Figure 4.35 and Figure 4.36 show the prediction accuracy in terms of the number of congested UDGs with the Order-5 Markov model.

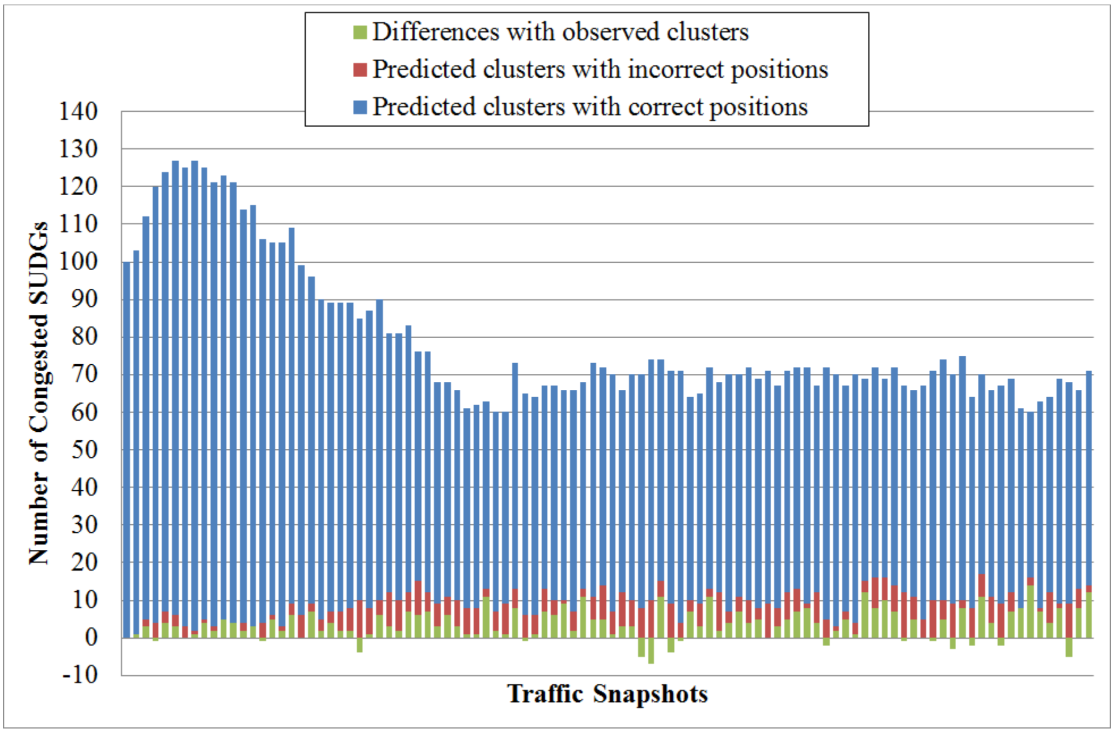


Figure 4.33 Predicted number of congested SUDGs with STPL, evaluated with 900 SUDGs (30 × 30)

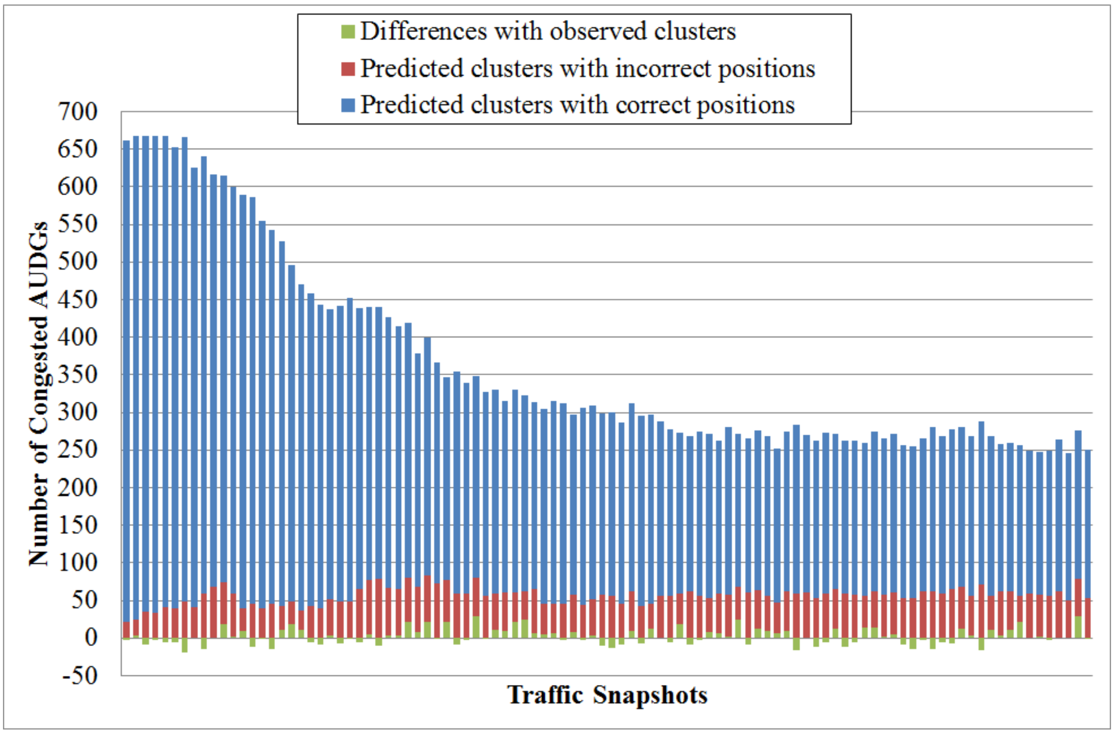


Figure 4.34 Predicted number of congested AUDGs with STPL, evaluated with 4 sets of AUDGs (36 × 20)

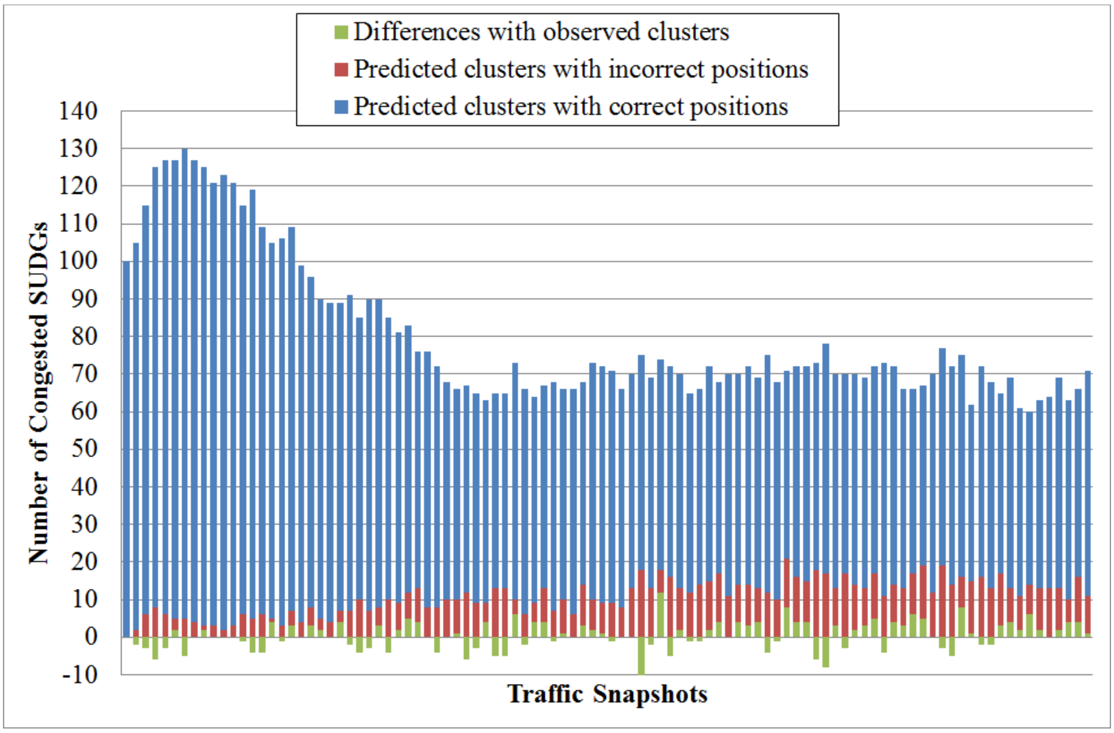


Figure 4.35 Predicted number of congested SUDGs with Order-5 Markov Model, evaluated with 900 SUDGs (30 × 30)

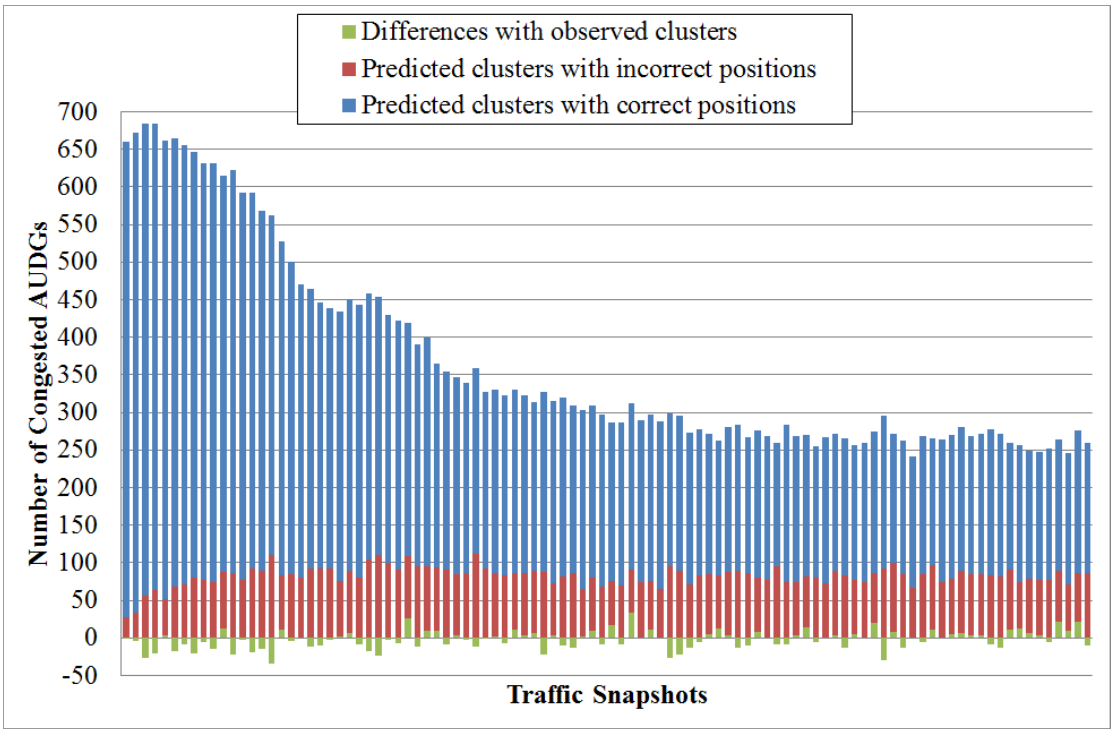


Figure 4.36 Predicted number of congested AUDGs with Order-5 Markov Model, evaluated with 4 sets of AUDGs (36 × 20)

Although the performance shown in Figure 4.30 and 4.31 does not look as good as the results in previous section from the perspective of the error rate value, the above four figures show that the performance is acceptable from the perspective of cluster movement prediction. It can be seen that most of the numbers and locations of the mobile clusters can be identified correctly, which means that the dispersion of most clusters can be tracked. Figure 4.37 shows the observed and predicted distribution of mobile clusters in five consecutive steps.

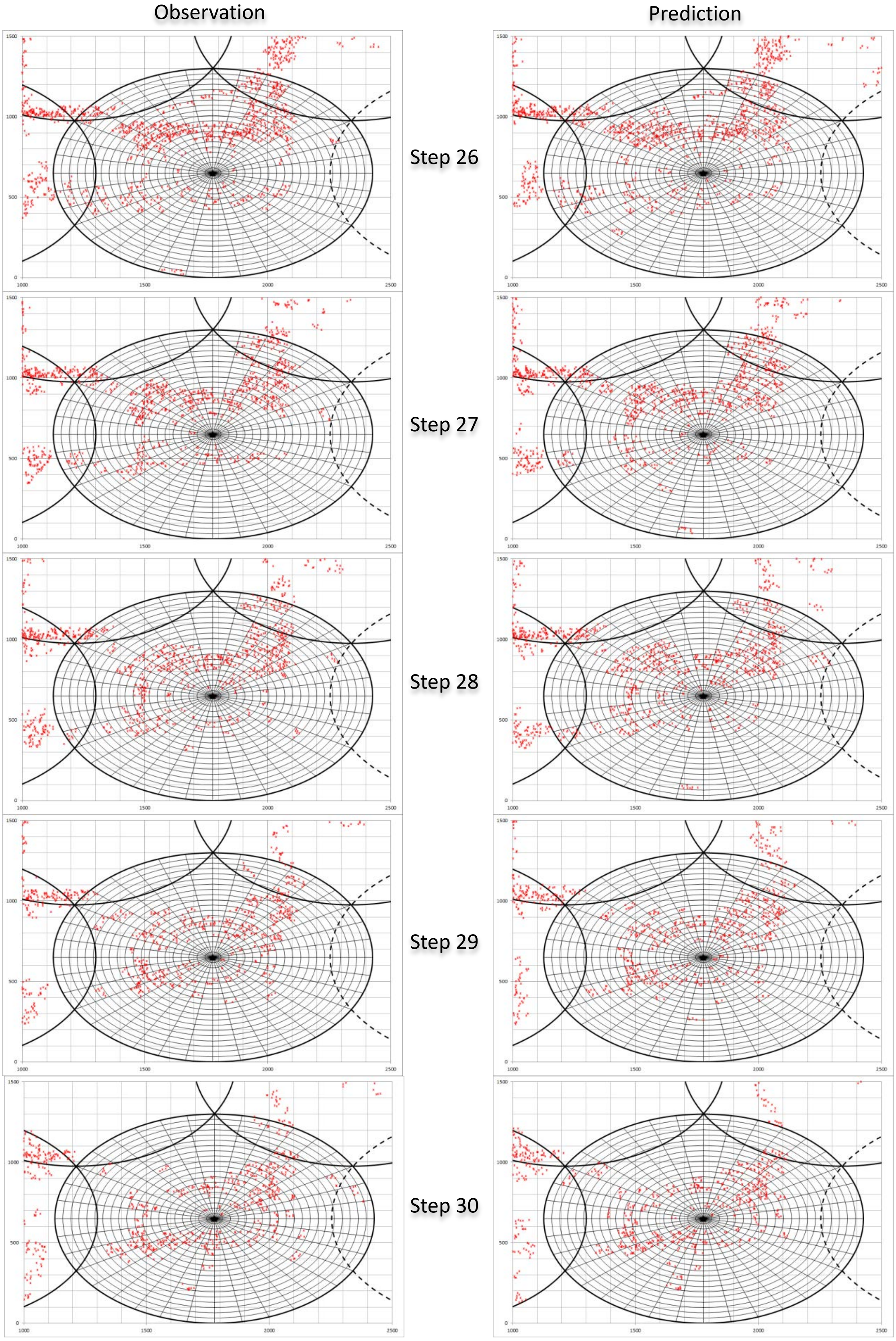


Figure 4.37 Observations and predictions of mobile clusters' distribution

4.4 Summary

In this chapter, the simulation results when applying the two proposed learning models to make predictions of the upcoming traffic patterns are presented. Three scenarios are introduced to test the performance of the learning models under different traffic situations. According to simulation results, it can be seen that both the STPL model and the Order-5 Markov model can accurately make predictions for the future traffic patterns and the distribution of mobile clusters.

With the STPL model, the local movement characteristics can be learnt from many perspectives, such as the local road topology, the probability of movement direction changing, and the varying tendency of movement speed; the future geographic traffic distribution can be well estimated; and the location of mobile clusters can be accurately identified. Besides, short convergence time can be seen when irregular traffic conditions are observed by the STPL model.

In the next chapter, a more sophisticated mobility model and an improved version of the STPL model are introduced.

5 INCREMENTAL PREDICTION

In this chapter, the traffic pattern predictions are performed in more sophisticated ways. A more complex mobility model is presented at the beginning of this chapter. Then, an improved pattern learning model is proposed. Finally, the simulation results are discussed and the accuracy of the prediction is evaluated. Both the mobility model and the pattern learning model introduced in this chapter are novel.

5.1 An Overview of the Incremental Pattern Prediction

In this research, all the local movement patterns learnt by the learning models are based on the “dummy data” generated by the mobility model. In order to evaluate the feasibility of mining cell transition data to learn the local movement patterns so as to make predictions on the upcoming user distributions, it is ideal to use the intra-cell transition data captured by BSs in real life. As the cost of collecting the real-life data is significant, mobility models are introduced in this research to help generating the historical data for pattern learning use and the individual movement context for traffic prediction use. In order to let the dummy data be close to the real ones, more real-life factors are taken into account in the mobility model.

In real life, the local geographic layout is complicated and movements of mobile users could be restricted by many obstacles. Besides, the movement patterns of mobile users are not unique. The user who travels in a car might move faster than pedestrians, but should follow the restriction of the road topology strictly. On the other hand, movements of pedestrians should not be seriously impacted by traffic jams, since the road restriction on pedestrians is mild and pedestrians can adjust their routes easily. Thus, a Multi-Movement-Pattern (MMP) mobility model is introduced to mimic more realistic traffic conditions.

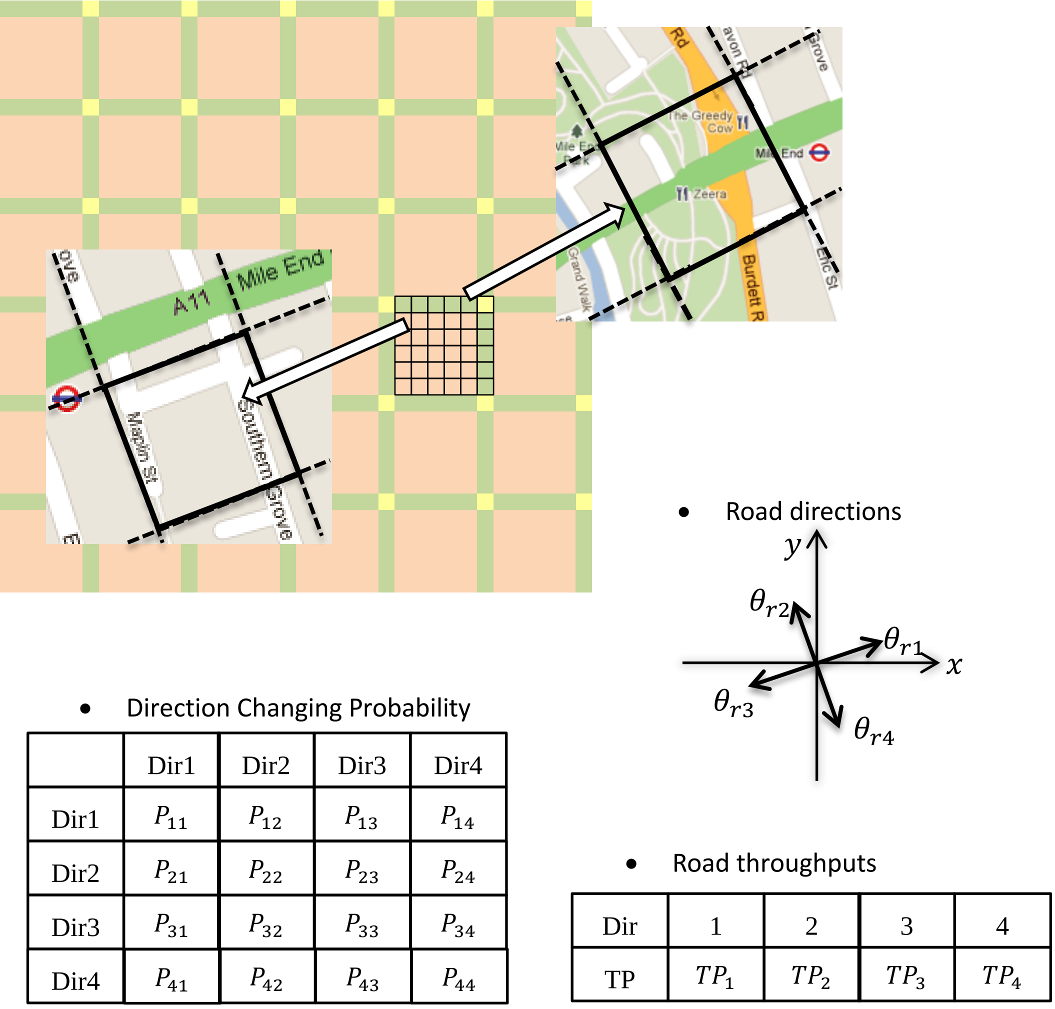
In the previous chapter, traffic pattern predictions are made with the Order-*k* Markov model and the STPL model. Local traffic patterns are learning from the learning patterns extracted by each learning grid. As only one type of the mobile user is deployed in the simulation area, all the learning patterns could be considered as reflections to the same local traffic pattern. When multiple types of mobile users are deployed in the simulation area, the local traffic learning patterns extracted from the case library might contain more than one type of movement patterns in that area. Different types of the patterns should be distinguished and chosen wisely when making predictions. Thus, the Dynamic Traffic Pattern Learning (DTPL) model is proposed to make predictions with mixed learning patterns.

5.2 The MMP Mobility Model

As the name implied, the MMP mobility model has multiple types of users deployed in the simulation area and can generate cell transition data that contains multiple movement patterns. The method of building the MMP model is similar to the process of building the basic mobility model that is presented in the previous chapters. GGEs are used to build the simulation area and RWPs are deployed to mimic the moving mobile terminals.

5.2.1 Simulation area

One advantage of using GGEs is that the local geographic features can be easily configured. In the MMP model, three types of the movement area are involved, the low speed movement area, the high speed movement area, and the intersection area. The low speed movement area is for the movement of pedestrian users; the high speed movement area is for the vehicular users; the intersection area is the place vehicular users change their movement directions. All the three types of the movement area are composed of GGEs with different geographic layout and restrictions. Figure 5.1 shows an example of the simulation area of the MMP model.



- Direction Changing Probability

	Dir1	Dir2	Dir3	Dir4
Dir1	P_{11}	P_{12}	P_{13}	P_{14}
Dir2	P_{21}	P_{22}	P_{23}	P_{24}
Dir3	P_{31}	P_{32}	P_{33}	P_{34}
Dir4	P_{41}	P_{42}	P_{43}	P_{44}

- Road throughputs

Dir	1	2	3	4
TP	TP_1	TP_2	TP_3	TP_4

Figure 5.1 An Example of the simulation area

As shown in the above figure, the same sets of parameters are applied in each GEE to represent the local geographic layout. The pink area represents the low speed movement area, in which all the settings are the same as the GEEs used in the previous simulations. The green area represents the high speed movement area, in which vehicular users move with the same direction of the road and the movement direction cannot be changed. Pedestrian users who move into the green area should keep the movement direction and move to the other side of the green area as fast as possible. The yellow area represents the intersection area, in which vehicular users can change the movement direction.

5.2.2 Moving terminals and walking rules

The RWPs are deployed into the simulation area at the beginning of the simulation to mimic the moving mobile terminals. As mentioned above, two types of user are involved in the MMP model, pedestrian users and vehicular users. For pedestrian users, all the settings are the same as the RWPs used in the previous simulations. For vehicular users, Equation 3.5 and 3.6 are used to determine the walking parameters θ and v.

As vehicular users should strictly follow the direction of the road during the movement, the value of ω is set to 0 in Equation 3.5 to discard the bias when determining the movement direction. As vehicular users should move much faster than pedestrians, a larger $\mathrm{v}_{\mathrm{normal}}$ and $\mathrm{v}_{\mathrm{max}}$ are used in Equation 3.6 when determining the movement speed. Figure 5.2 shows an example traffic snapshot of the MMP model. The blue dots represent pedestrian users and the green dots represent vehicular users.

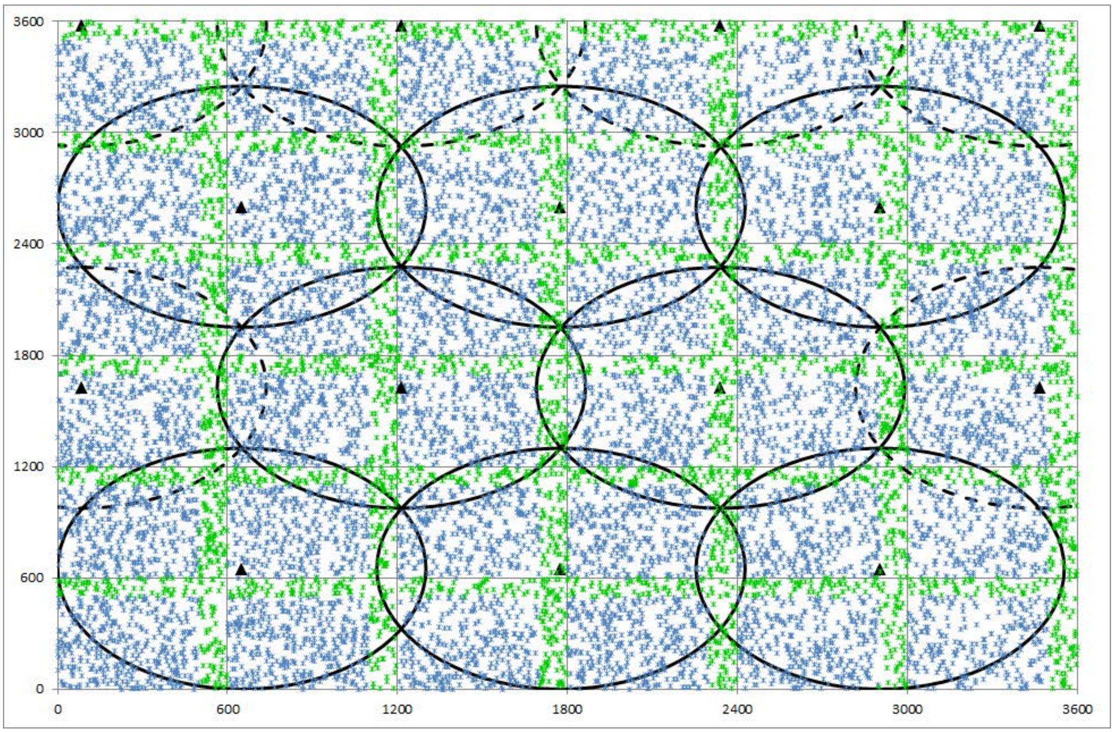


Figure 5.2 A traffic snapshot of the MMP model

5.2.3 Other configurations

One important modification in the MMP model is that the time interval between two consecutive steps is set to 30 seconds instead of 60 seconds. However, the time interval between two consecutive observations is still 60 seconds, which means the traffic snapshots will be captured in every two walking steps.

In the previous simulations, the time intervals of the RWP moving and the snapshot capturing are both set to 60 seconds. With this setting, the learning model can easily calculate the movement speed and the movement direction of one RWP with its coordinates from consecutive steps, as shown in Figure 5.3 (a). This simplifies the pattern learning process. However, the movement of the mobile terminal might not follow the same pattern between two consecutive observations in real life, as shown in Figure 5.3 (b). Thus, the MMP model gives different time interval to the RWP moving and the snapshot capturing to increase the complexity of the pattern learning task.

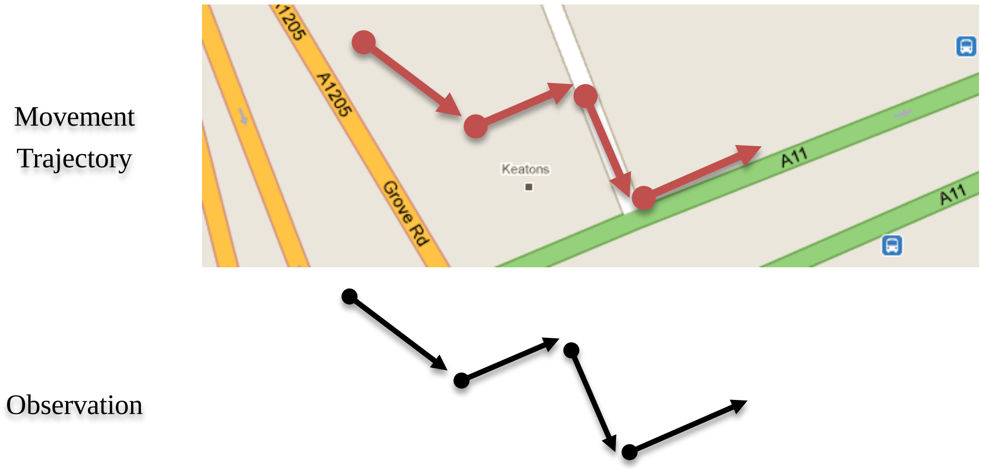


Figure 5.3 (a) Same time interval for the RWP moving and the snapshot capturing

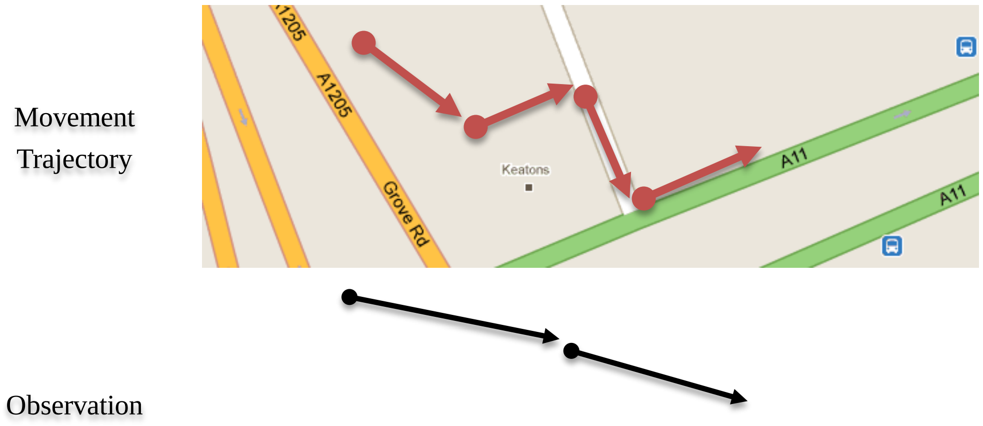


Figure 5.3 (b) Different time intervals for the RWP moving and the snapshot capturing

5.3 The DTPL Model

In order to make traffic pattern predictions based on the more generalized movement observations, the Dynamic Traffic Pattern Learning (DTPL) model is proposed in this research. The concept of the DTPL model is inspired from the *k*-Nearest Neighbour algorithm and the Hidden Markov Chain model, and can be considered as an improved version of the STPL model. A combination of the on-line learning and off-line learning is applied in the pattern learning phase and pattern prediction phase.

Similar to the pattern learning phase with the STPL model, a number of learning girds are deployed in the simulation area, and each learning grid extracts the learning patterns $C_{context} = \{C_{from}, C_1, C_2, \dots, C_n, C_{to}\}$ $(n \geq 1)$ from the case library to build the local pattern library. When using the STPL model, this step is crucial, as the predictions made by the STPL model in learning grid LG_i are all based on the learning patterns extracted by LG_i. The more learning patterns the learning grid records, the more traffic patterns it can recognize. When using the DTPL model, this step can be considered as indexing the substrings of the historical trajectories in the library. As the movement direction and speed calculated by two consecutive coordinate pairs (C_k, C_{k+1}) are not necessarily accurate, no further actions are needed in this phase with the DTPL model. When the simulation starts, all the learning grids capture the trajectories of the RWPs visiting them and record the relevant segments into their buffer to extend the local pattern library with the latest traffic conditions.

In the pattern prediction phase, predictions are made for individual mobile terminals. Assume $\{C_i, C_{i-1}, C_{i-2}, \dots, C_{i-k+1}\}$ is the most recent k-length sequence of coordinates of a RWP at step i. The value of k indicates the needed length of the context sequence, and the length of the learning patterns need to be extracted from the local pattern library for one prediction. $\mathbb{R} = \{r_1, r_2, \dots, r_k\}$ is a set of threshold values that are used to filter out the similar traffic patterns from the local library. The learning patterns are filtered by the following procedures:

- Step 1: let C_i be the centre of the coverage area and r_1 be the coverage radius. Any learning grids partially or completely covered by the above area are treated as relevant learning grids.

- Step 2: For a learning pattern $C_{context} = \{C_{from}, C_1, C_2, \dots, C_n, C_{to}\}$ recorded in a relevant learning grid, denote it as $C_{context} = \{C_0, C_1, C_2, \dots, C_n, C_{n+1}\}$ and calculate the distance $d_{i,l}$ between C_i and C_l $(l = n, n-1, \dots, 1)$ in descending order of l. When $d_{i,l} < r_1$, extract the substring $C'_{context} = \{C_0, C_1, C_2, \dots, C_l, C_{l+1}\}$ and load it into the buffer. Repeat Step 2 for all the learning patterns recorded in all the relevant learning grids.

- Step 3: For a substring $C'_{context} = \{C_0, C_1, C_2, \dots, C_{l-1}, C_l; C_{l+1}\}$ obtained from the previous step, calculate the distance $d_{i-1,l-1}$ between C_{i-1} and C_{l-1} . If $d_{i-1,l-1} < r_2$, mark the $C'_{context}$ as level-2 alive. Otherwise, mark the $C'_{context}$ as level-2 dead. Repeat Step 3 for all the substrings obtained by Step 2.

- Step 4: Repeat Step 3 to compare the C_{i-m+1} and C_{l-m+1}, if there are more than one substrings that are marked as level-$(m-1)$ alive and if there are more than one substrings whose length are equal or longer than *m*. Otherwise, terminate Step 4 and move forward to Step 5.

- Step 5: Assume that Step 4 is terminated after the comparison of C_{i-m+1} and C_{l-m+1}. Then the substrings eventually marked as level-*m* alive and level-*m* dead are both used to estimate the next position of the target RWP. Figure 5.4 illustrates the steps above.

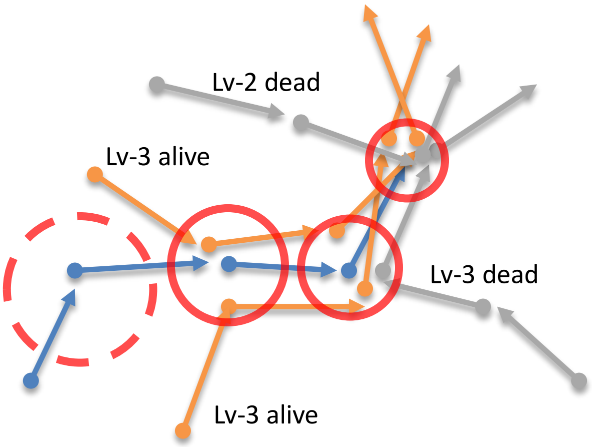


Figure 5.4 Filtering learning patterns

In the above figure, blue line represents the movement context of the target RWP; the orange lines represent the substrings marked as alive; the grey lines represent the substrings marked as dead. As the fourth filtering is failed, both the level-3 alive substrings and the level-3 dead substrings are used to estimate the next position of the target RWP in the prediction phase.

After filtering out the relevant level-*m* alive and the level-*m* dead substrings $C'_{context} = \{C_{l-m+1}, C_{l-m+2}, \dots, C_{l-1}, C_l; C_{l+1}\}$, all the possible next locations C_{l+1} are extracted and used to compose the possible next location set $\mathbb{L}$. With the restriction of the same geographic layout, elements in $\mathbb{L}$ should not be distributed randomly but in the limited number of areas. With proper classification of the elements in c, locations of those areas can be revealed and the main movement patterns in the target area can be determined. Inspired by applications of the *k*-means clustering algorithm presented in [DH10], [EB08], [FNNAK10] and [KIS09], the *k*-means clustering algorithm is applied to classify the possible next locations into proper clusters in the DTPL model.

Assume the *m* elements in $\mathbb{L}$ are classified into *k* clusters; $P_i(i = 1,2, \ldots, k)$ denotes the *i*th cluster with $n_i(i = 1,2, \ldots, k)$ pairs of the coordinates of C_{l+1}; $(x_{in}, y_{in})(i = 1,2, \ldots, k; n = 1,2, \ldots, n_i)$ denotes the coordinates of the *n*th element in the *i*th cluster; $Ce_i(i = 1,2, \ldots, k)$ denotes the centroid of the *i*th cluster with the coordinates (cx_i, cy_i). The process of applying the *k*-means clustering algorithm to classify the elements in $\mathbb{L}$ can be described in terms of the following steps:

- Step 1: Set the value of *k* and then randomly choose *k* elements from $\mathbb{L}$ as the initial centroid of each cluster.

- Step 2: Update the Euclidean distance matrix *D* with the current centroids by applying the following equations. $(x_i, y_i)(i = 1,2, \ldots, m)$ denotes the coordinates of the *i*th element in $\mathbb{L}$.

$$D = \begin{bmatrix} d_{11}, d_{12}, & \cdots & d_{1m} \\ \vdots & \ddots & \vdots \\ d_{k1}, d_{k2}, & \cdots & d_{km} \end{bmatrix}$$
$$d_{ij} = \sqrt{\left(x_j - cx_i\right)^2 - \left(y_j - cy_i\right)^2}\ (i = 1,2, \ldots, k; j = 1,2, \ldots, m) \quad (5.1)$$

- Step 3: Update the clustering status matric *S* with the following equations.

$$S = \{S_1, S_2, \ldots, S_m\}$$
$$S_i = argmin(d_{1i}, d_{2i}, \ldots, d_{ki})\ (i = 1,2, \ldots, m) \quad (5.2)$$

- Step 4: Recalculate the location of the centroid for this cluster with the following equations.

$$\begin{cases} cx_i = \dfrac{\sum_n x_{in}}{n_i} \\ cy_i = \dfrac{\sum_n y_{in}}{n_i} \end{cases} \quad (i = 1,2,\dots,k; n = 1,2,\dots,n_i) \tag{5.3}$$

- Step 5: Repeat Step 2, 3, and 4 until the clustering status matric *S* keeps stable, or all elements in the Euclidean distance are less than a predefined threshold.

- Step 6: Repeat from Step 1 if a new value of *k* needs to be configured. Figure 5.5 illustrates an example of the clustering results.

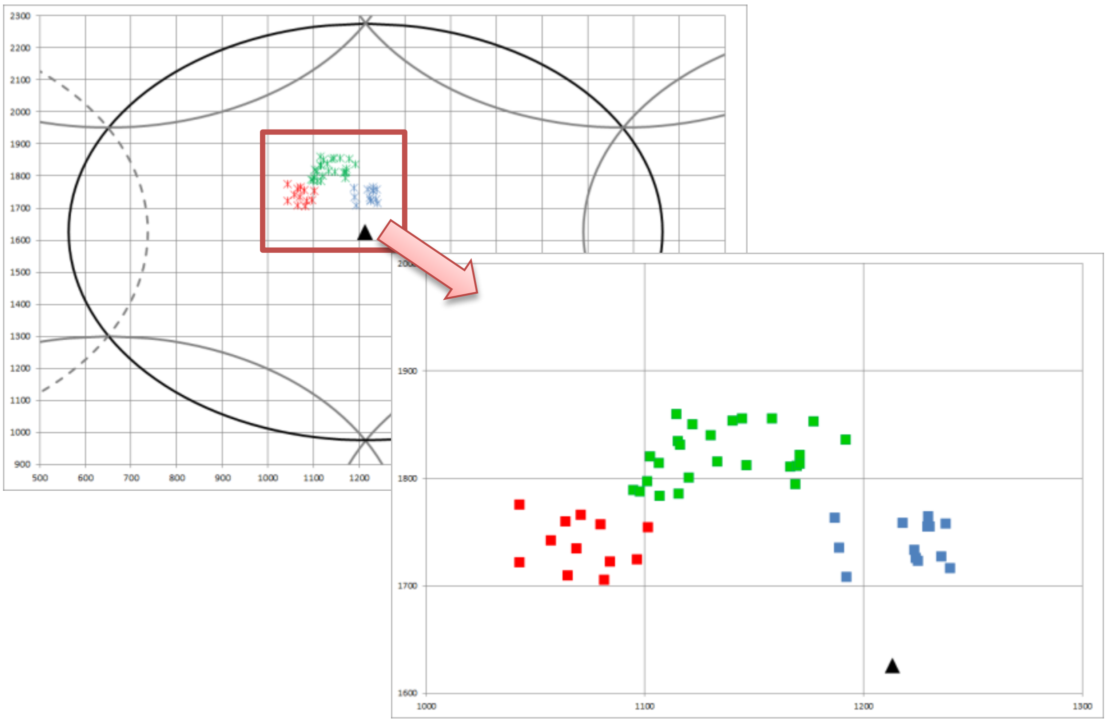


Figure 5.5 Clustering the possible next locations

Let N_i, N_{ia} and N_{id} $(i = 1,2,\dots,k)$denote the total number of possible next location, the number of possible next locations from the level-*m* alive substrings, and the number of possible next locations from the level-*m* dead alive substrings, respectively, in the *i*th cluster. Then, the probability of moving to the *i*th cluster P_i can be represented as:

$$P_i = \frac{\alpha \cdot N_{ia} + \beta \cdot N_{id}}{\sum_i (\alpha \cdot N_{ia} + \beta \cdot N_{id})} \quad (i = 1,2, \dots, k) \tag{5.4}$$

where α and β $(0 < \beta < \alpha < 1)$ are the weights of N_{ia} and N_{id} which indicate the extent of the influence of the better matched substrings in the prediction phase.

Let X_i and Y_i $(i = 1,2, \dots, k)$ denote the sample sets of the x-axis coordinates and the y-axis coordinates of the possible next locations in the *i*th cluster. With the purpose of simulating a realistic environment, the distribution of the possible next locations in each cluster is assumed to follow a normal distribution. Assume that samples in X_i and Y_i are $(x_{i1}, x_{i2}, \dots, x_{in_i})$ and $(y_{i1}, y_{i2}, \dots, y_{in_i})$, and n_i denotes the number of samples in X_i and Y_i. Thus, the distributions of $X_i \sim N(\mu_{x_i}, \sigma_{x_i}^2)$ and $Y_i \sim N(\mu_{y_i}, \sigma_{y_i}^2)$ can be estimated by the following equations:

$$\begin{cases} \hat{\mu}_{x_i} = \bar{x}_i = \sum_{j=1}^{n_i} x_{ij} \,/\, n_i \\ \hat{\sigma}_{x_i}^2 = \frac{1}{n_i} \sum_{j=1}^{n_i} (x_{ij} - \bar{x}_i)^2 \end{cases} \quad (i = 1, 2, \dots, k) \tag{5.5}$$

$$\begin{cases} \hat{\mu}_{y_i} = \bar{y}_i = \sum_{j=1}^{n_i} y_{ij} \,/\, n_i \\ \hat{\sigma}_{y_i}^2 = \frac{1}{n_i} \sum_{j=1}^{n_i} (y_{ij} - \bar{y}_i)^2 \end{cases} \quad (i = 1, 2, \dots, k) \tag{5.6}$$

where $\hat{\mu}_{x_i}$, $\hat{\sigma}_{x_i}^2$, $\hat{\mu}_{y_i}$ and $\hat{\sigma}_{y_i}^2$ are the estimated parameters of the probability distribution of in X_i and Y_i.

When making a prediction for a given context $\{C_i, C_{i-1}, C_{i-2}, \dots, C_{i-k+1}\}$, the possible next locations $\bar{C}_{i+1}$ are extracted by filtering out the relevant segments from the local case library of the learning grid. With the k-means clustering algorithm mentioned above, $\bar{C}_{i+1}$ are classified into k clusters. Then, the probability of moving to each cluster is calculated with Equation 5.4, and the location distribution in each cluster is estimated with Equation 5.5 and 5.6. Finally, the cluster the target RWP is predicted to move to is chosen according to the probability distribution of all the clusters, and the coordinates of $\bar{C}_{i+1}$ are generated with the estimated location distribution in the chosen cluster.

Compared to the STPL model, the reason this model is named as Dynamic Traffic Pattern Learning model is that only the historical data that suits the given contexts are used to make predictions. The DTPL model can let the pattern learning phase focus on the individual movement patterns rather than the generalized patterns.

5.4 Simulation Results

In the simulation, the simulation area was built with 1296 GGEs (36×36), as shown in Figure 5.6. Each block of the low speed movement area was composed of 25 GGEs (5×5) with different geographic layouts and is shown in pink. The high speed movement areas are in the middle of two adjacent low speed areas and are green. The yellow areas are the intersections, where vehicular users could change movement direction. The length and width of the simulation area are both 3.6km.

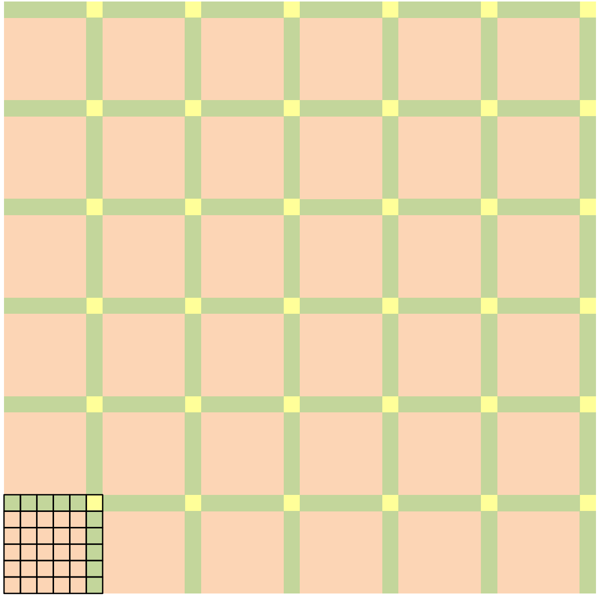


Figure 5.6 The simulation area of the MPP model

15000 RWPs were distributed in the simulation area at the beginning of the simulation. The type of the RWP, pedestrian user or vehicular user, was decided by its initial location. RWPs deployed in the pink area were set to pedestrian user type; RWPs in the green and yellow area were set to vehicular user type. During the simulation, the user type could not be changed.

Similar to the simulations presented in the previous chapter, UDGs were applied to evaluate the prediction performance. In this simulation, 1600 SUDGs (40×40) and 8 sets of AUDGs (36×20) were deployed, as shown in Figure 5.7. The grey square-sharped grids are the SUDGs; the arc-shaped grids are the AUDGs. Each BS with the real-line coverage area owns a set of AUDGs.

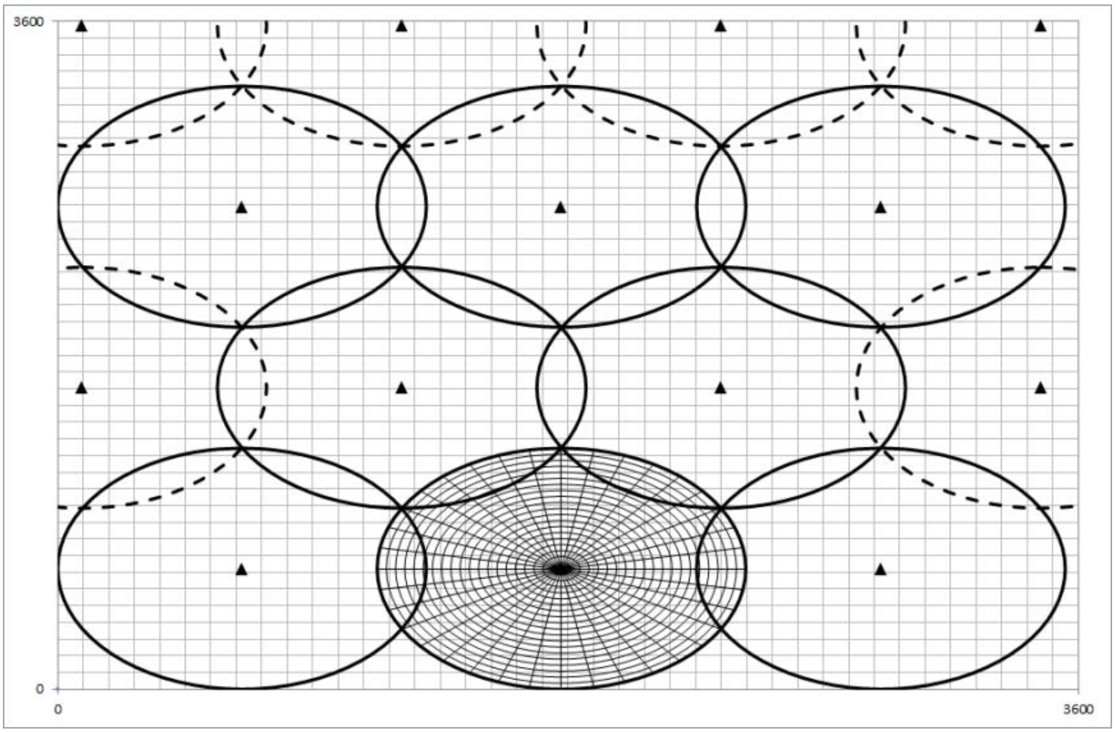


Figure 5.7 SUDGs and AUDGs deployed in the simulation of the MPP model

Two scenarios were included in the simulation, the normal walking scenario and the cluster forming scenario. The cluster dispersing scenario was not included in the simulation individually, as similar traffic patterns could be seen when the traffic restriction was finished in the cluster forming scenario. 200 snapshots were generated by the MMP model, but only the snapshots with an even order number are considered as valid observations. 324 learning grids (18×18) were deployed in the simulation area to capture and learn the traffic patterns. Notice that the size of the learning gird was the same as the one used in the previous simulations. The DTPL model, the STPL model and the order-5 Markov model were applied in the simulation to make predictions for the upcoming traffic patterns. In the simulation, all the models started to make predictions from the 5th observation.

Figure 5.8 shows the one-step-ahead prediction performance evaluated by 1600 SUDGs; Figure 5.9 shows the one-step-ahead prediction performance evaluated by 8 sets of AUDGs (36×20). Predictions evaluated here are made based on the observations from the normal walking scenario.

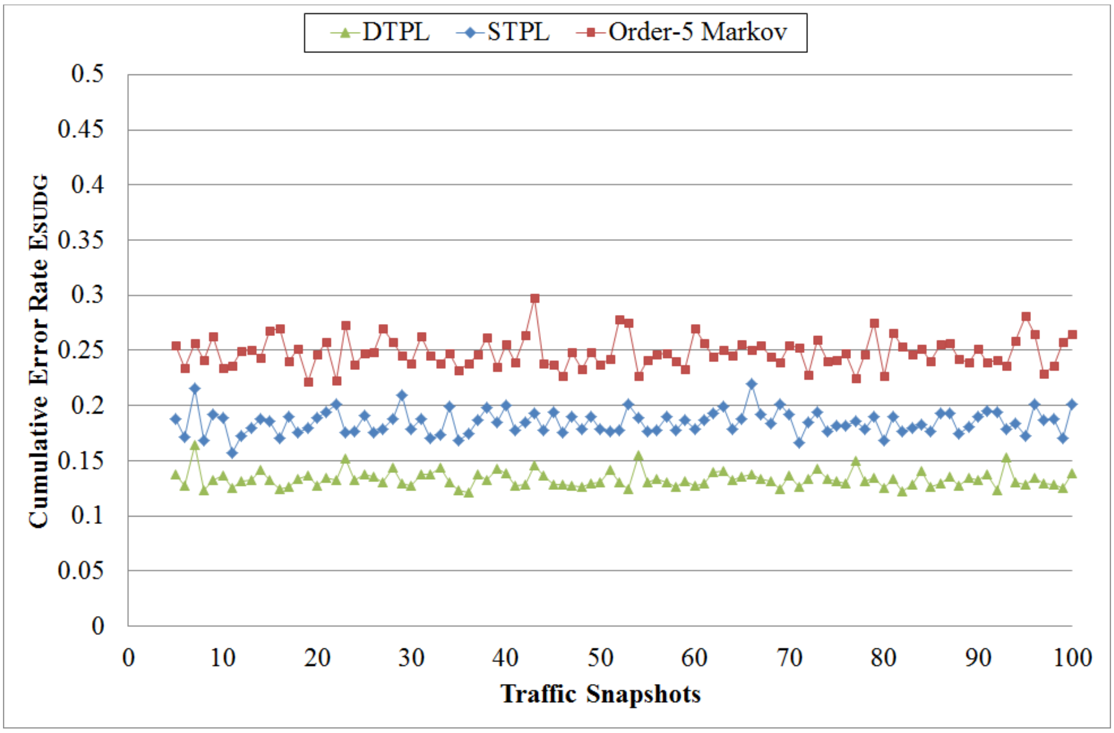


Figure 5.8 Prediction performances in the normal walking scenario, evaluated with 1600 SUDGs（40 × 40）

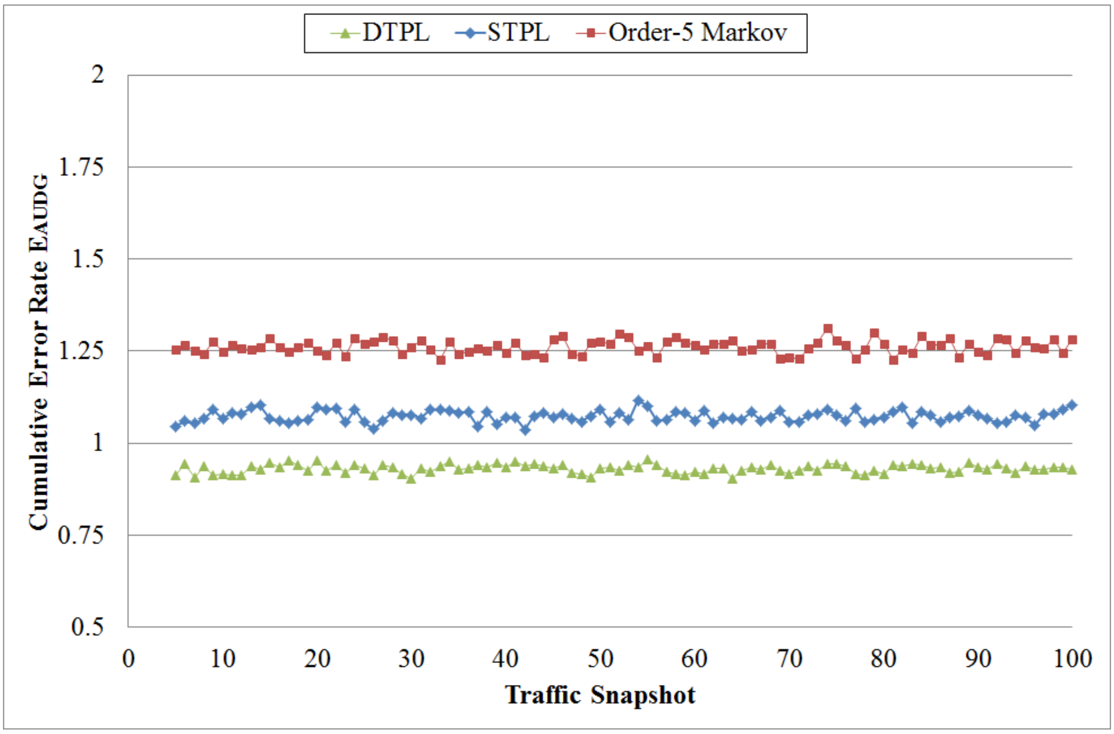


Figure 5.9 Prediction performances in the normal walking scenario, evaluated with 8 sets of AUDGs（36 × 20）

According to the above two figures, it can be seen that the cumulative error rate E_{SUDG} of the DTPL model is around 5% lower than the E_{SUDG} error rate of the STPL model, and is around 12% lower than the E_{SUDG} error rate of the Order-5 Markov model. When evaluating the prediction results with AUDG, the average difference between the predicted population of RWPs and the observation in every AUDG is less than 1 with the DTPL model. This average population difference is around 1.1 with the STPL model, and 1.45 with the order-5 Markov model.

Compared to the prediction performance shown in Figure 4.7 and 4.8, it also can be seen that the prediction accuracy decreases when using the STPL model and the order-5 Markov model make predictions based on the observations from the MMP model. This is caused by the mixed movement patterns in the observed traffic snapshots, and the asynchronous frequency between the traffic snapshot observation and the traffic pattern generation.

Figure 5.10 and Figure 5.11 show the prediction accuracy in terms of the number of congested UDGs with the DTPL model; Figure 5.12 and Figure 5.13 show the prediction accuracy in terms of the number of congested UDGs with the STPL model; Figure 5.14 and Figure 5.15 show the prediction accuracy in terms of the number of congested UDGs with the Order-5 Markov model.

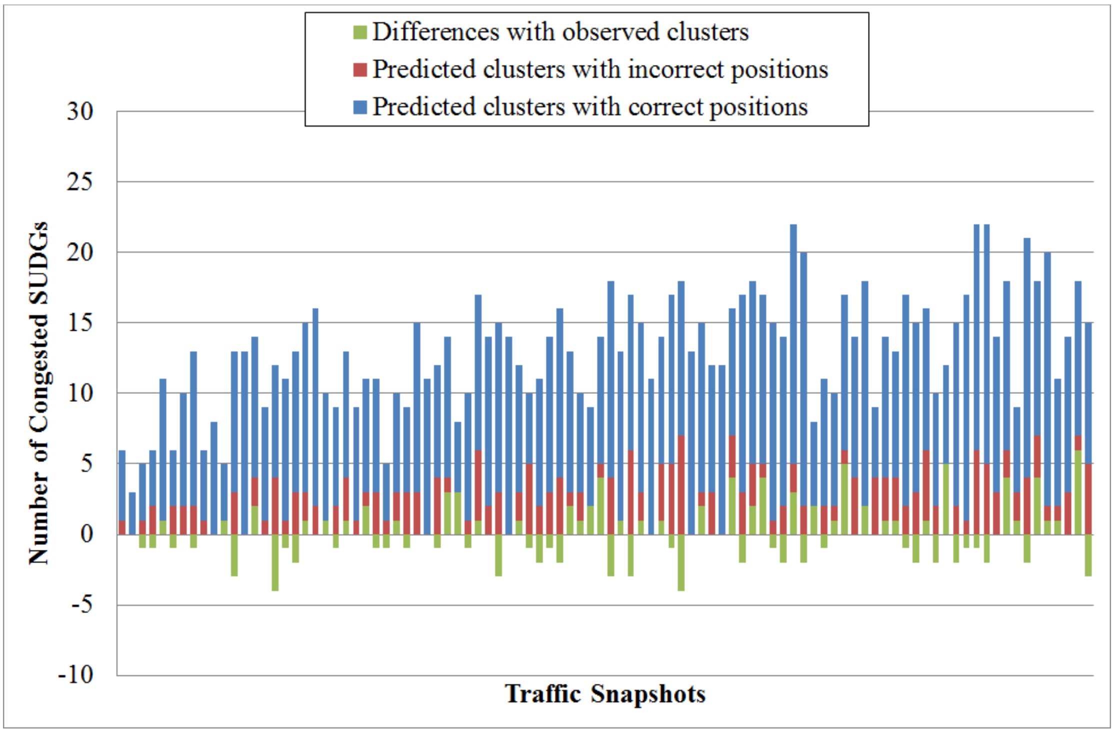


Figure 5.10 Predicted number of congested SUDGs with DTPL, evaluated with 1600 SUDGs (40 × 40)

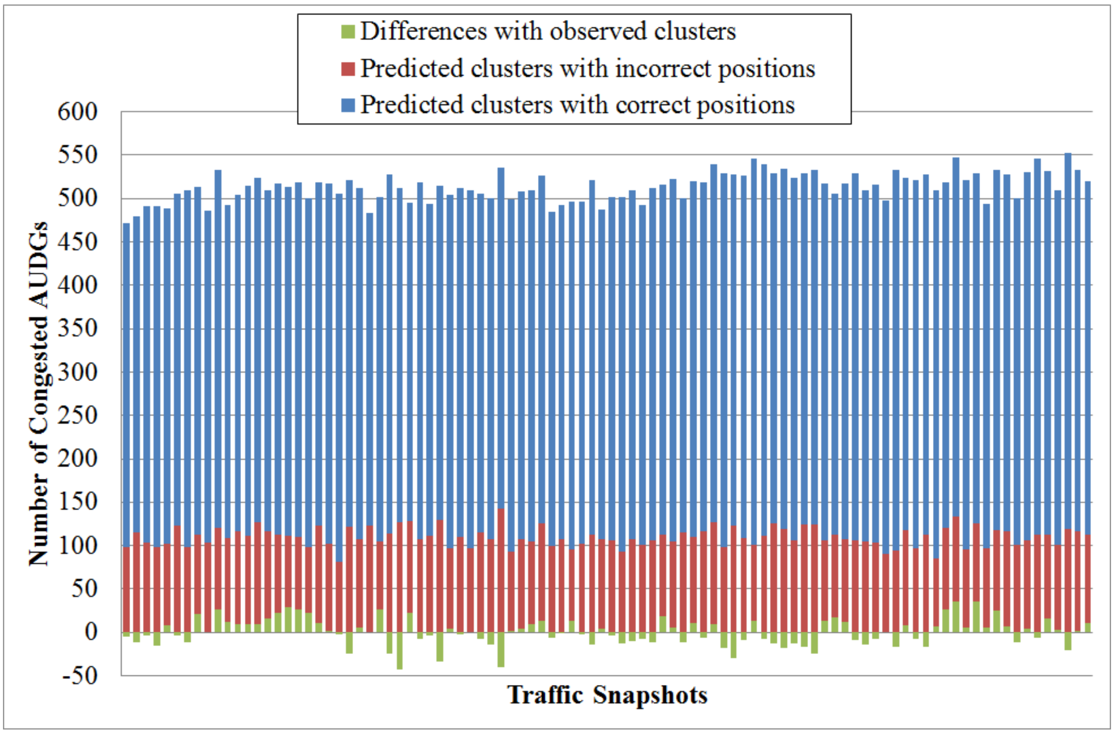


Figure 5.11 Predicted number of congested AUDGs with DTPL, evaluated with 8 sets of AUDGs (36 × 20)

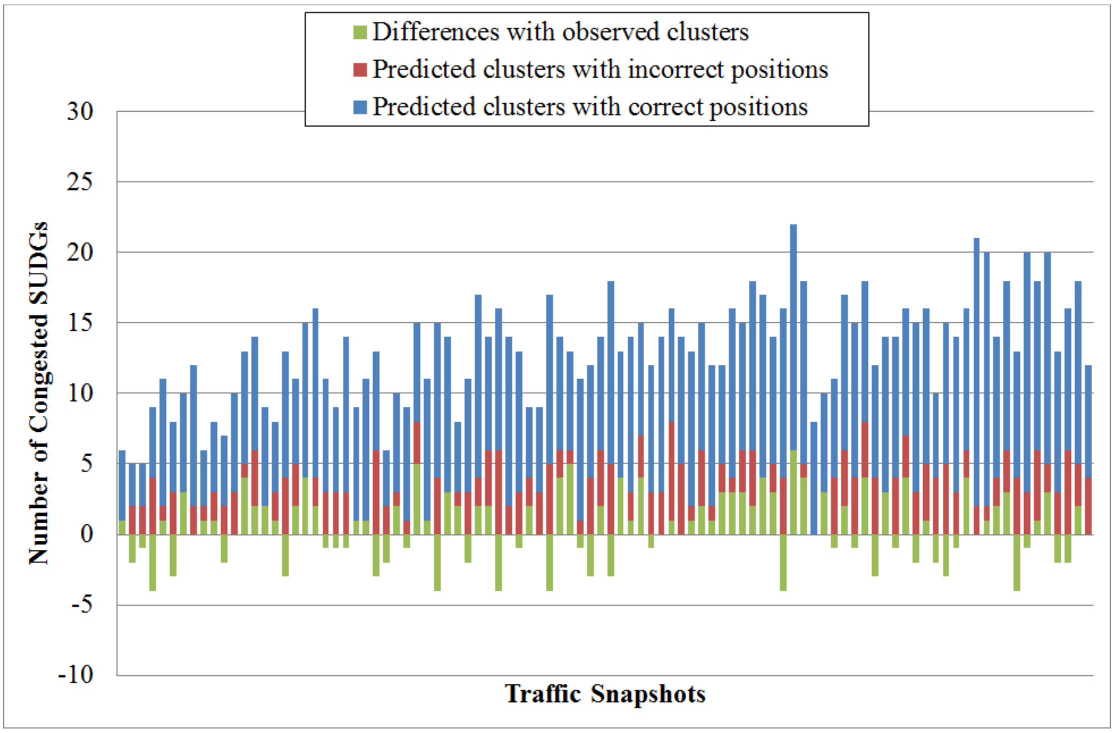


Figure 5.12 Predicted number of congested SUDGs with STPL, evaluated with 1600 SUDGs (40 × 40)

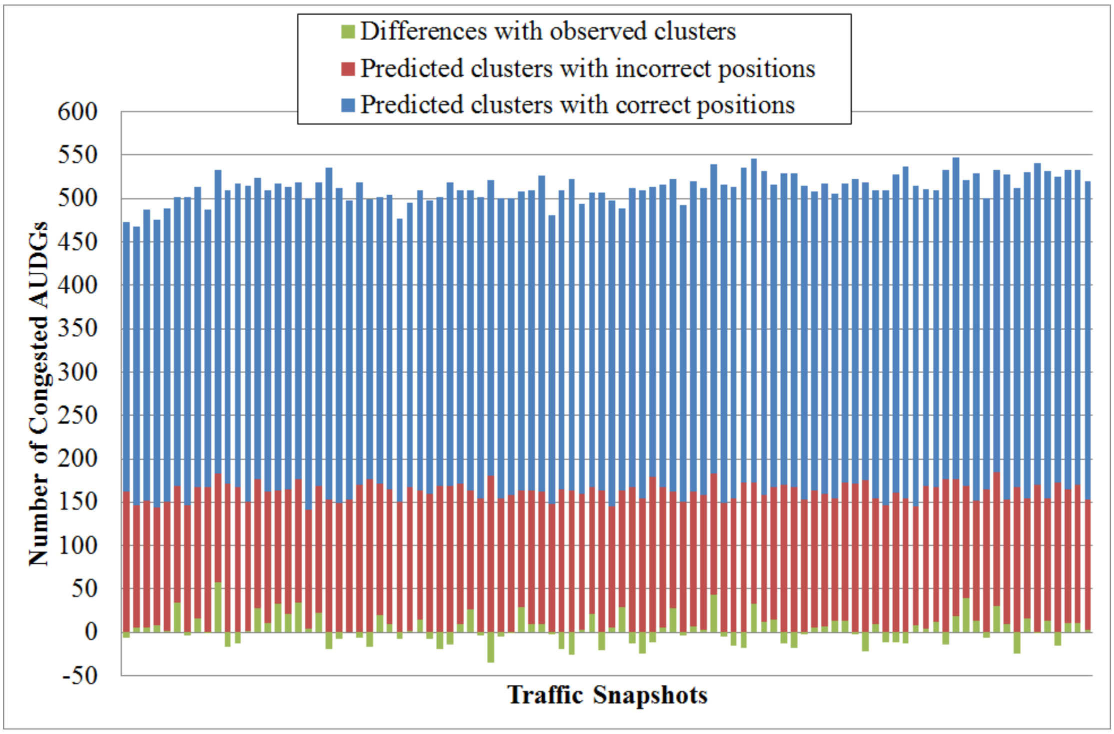


Figure 5.13 Predicted number of congested AUDGs with STPL, evaluated with 8 sets of AUDGs (36 × 20)

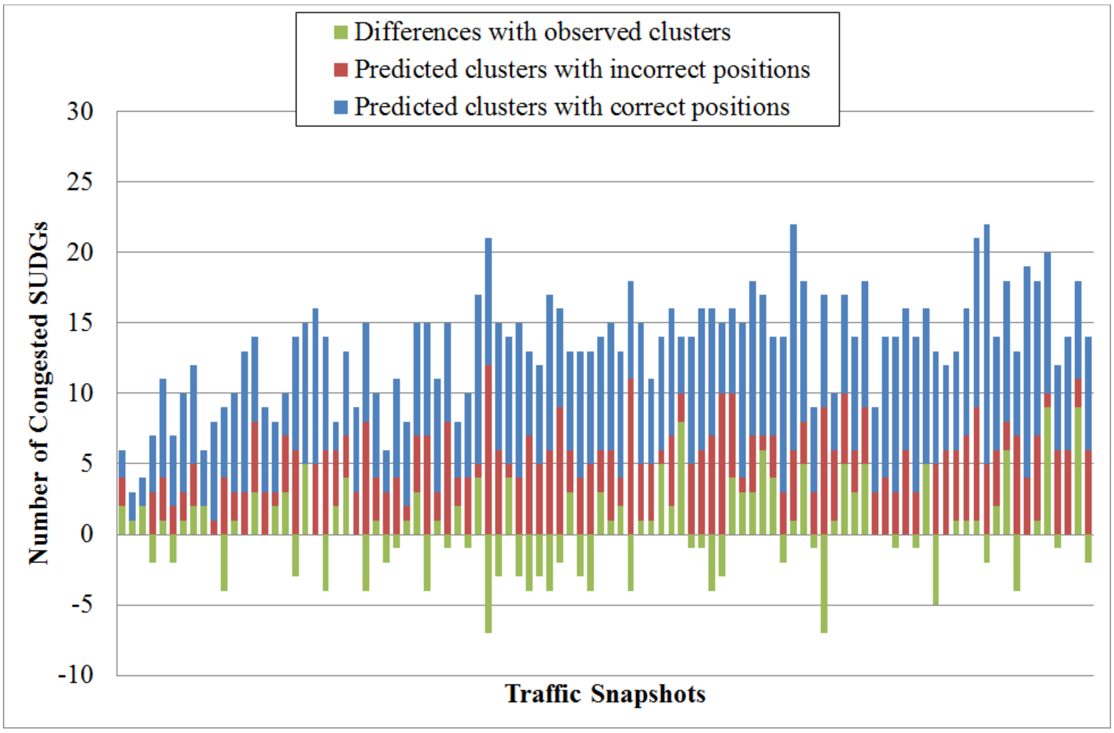


Figure 5.14 Predicted number of congested SUDGs with Order-5 Markov model, evaluated with 1600 SUDGs (40 × 40)

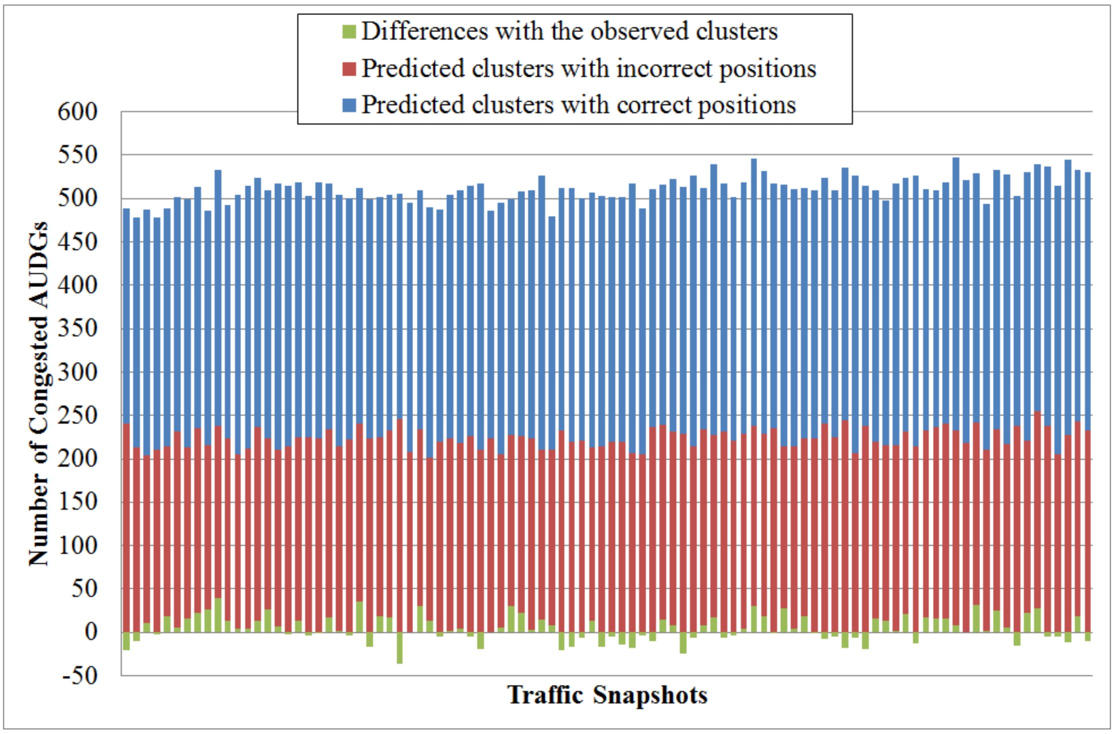


Figure 5.15 Predicted number of congested AUDGs with Order-5 Markov model, evaluated with 8 sets of AUDGs (36 × 20)

In the above six figures, the best performance can be seen when predictions are made with the DTPL model. With DTPL model in the normal walking scenario, the predicted number of mobile clusters is very close to the observed number. When evaluating the performance with 8 sets of AUDGs, around 80% of the predicted mobile clusters are in the same location as the observed ones. Compared to the performance presented in the previous chapter, the prediction accuracies of the STPL model and the Order-5 Markov model decrease, which is mainly caused by the mixed movement patterns in the observed traffic snapshots.

Figure 5.16 shows the one-step-ahead prediction performance evaluated by 1600 SUDGs when the simulation is carried out in the cluster forming scenario; Figure 5.17 shows the one-step-ahead prediction performance evaluated by 8 sets of AUDGs (36×20) in the cluster forming scenario. Traffic events are triggered at Step 30, and can be seen from the 15th traffic observation. All the traffic events end at Step 80.

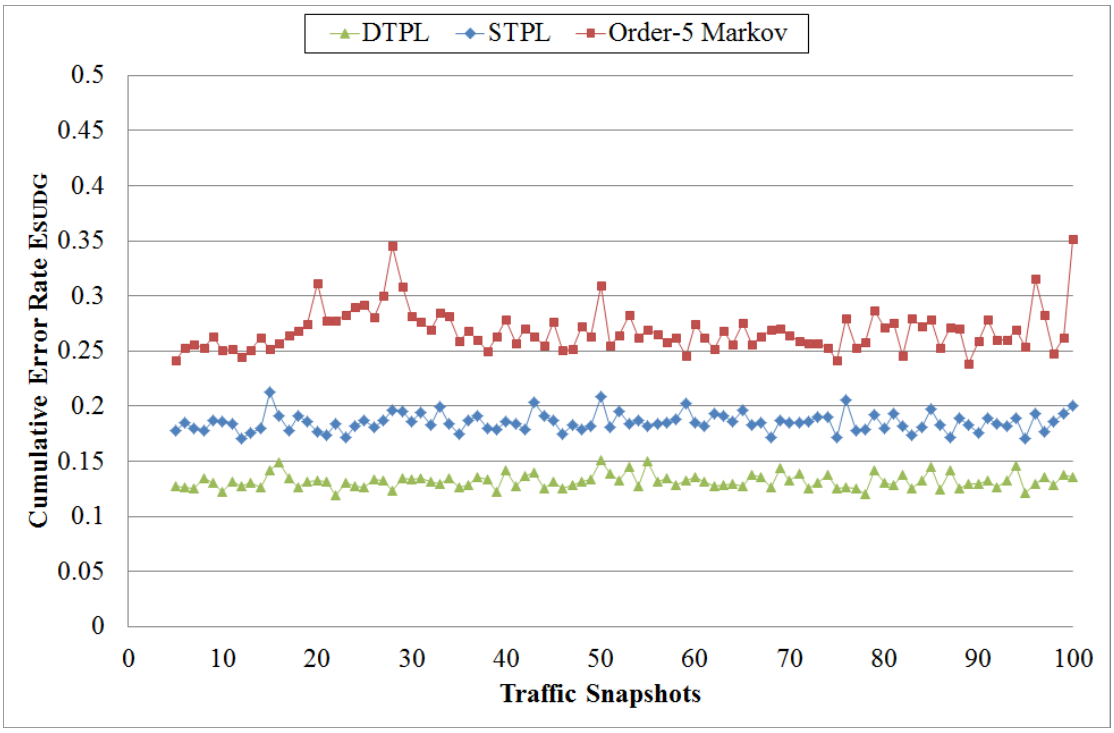


Figure 5.16 Prediction performances in the cluster forming scenario, evaluated with 1600 SUDGs (40×40)

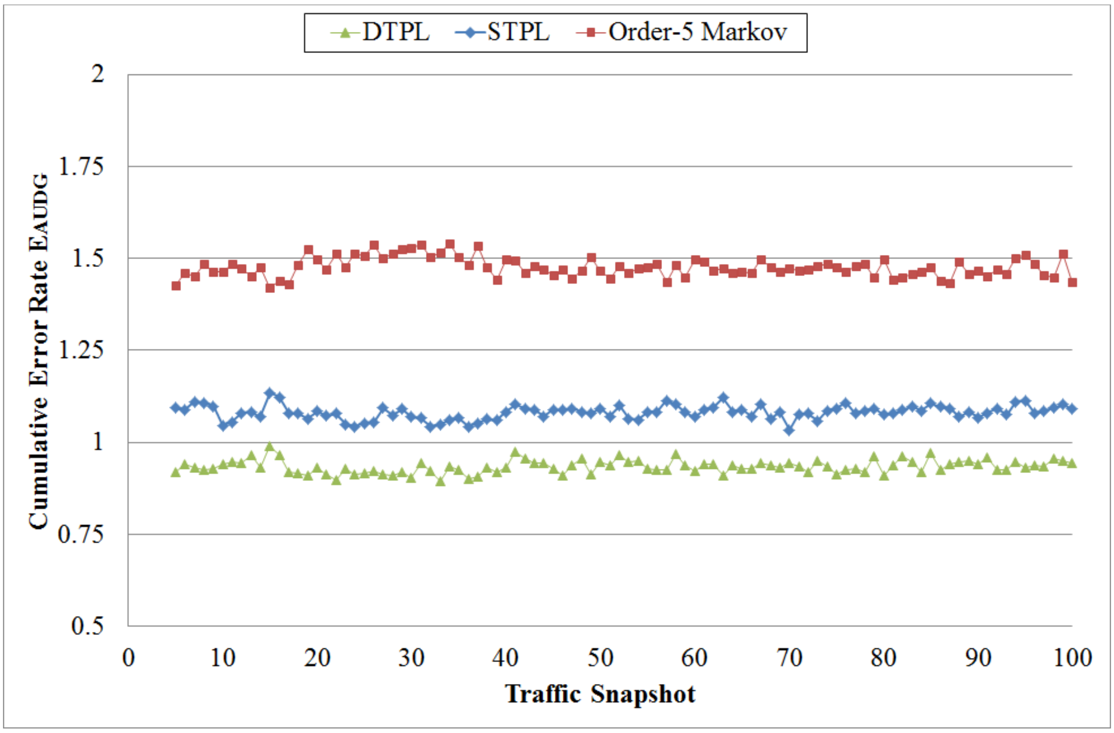


Figure 5.17 Prediction performances in the cluster forming scenario, evaluated with 8 sets of AUDGs (36×20)

According to the above two figures, the lowest error rates E_{SUDG} and E_{AUDG} are obtained when applying the DTPL model, with average errors of 0.1320 and 0.9339 respectively. For both the DTPL model and the STPL model, E_{SUDG} and E_{AUDG} increase suddenly at the 15th step. This is caused by the change of the movement patterns in the areas affected by the traffic events. The error rates decrease to the normal range after the pattern changes are known to the DTPL and STPL model. For the Order-5 Markov model, the influence from the traffic events lasts longer, due to the comparative simple structure of the model.

Figure 5.18 and Figure 5.19 show the prediction accuracy in terms of the number of congested UDGs with the DTPL model; Figure 5.20 and Figure 5.21 show the prediction accuracy in terms of the number of congested UDGs with the STPL model; Figure 5.22 and Figure 5.23 show the prediction accuracy in terms of the number of congested UDGs with the Order-5 Markov model.

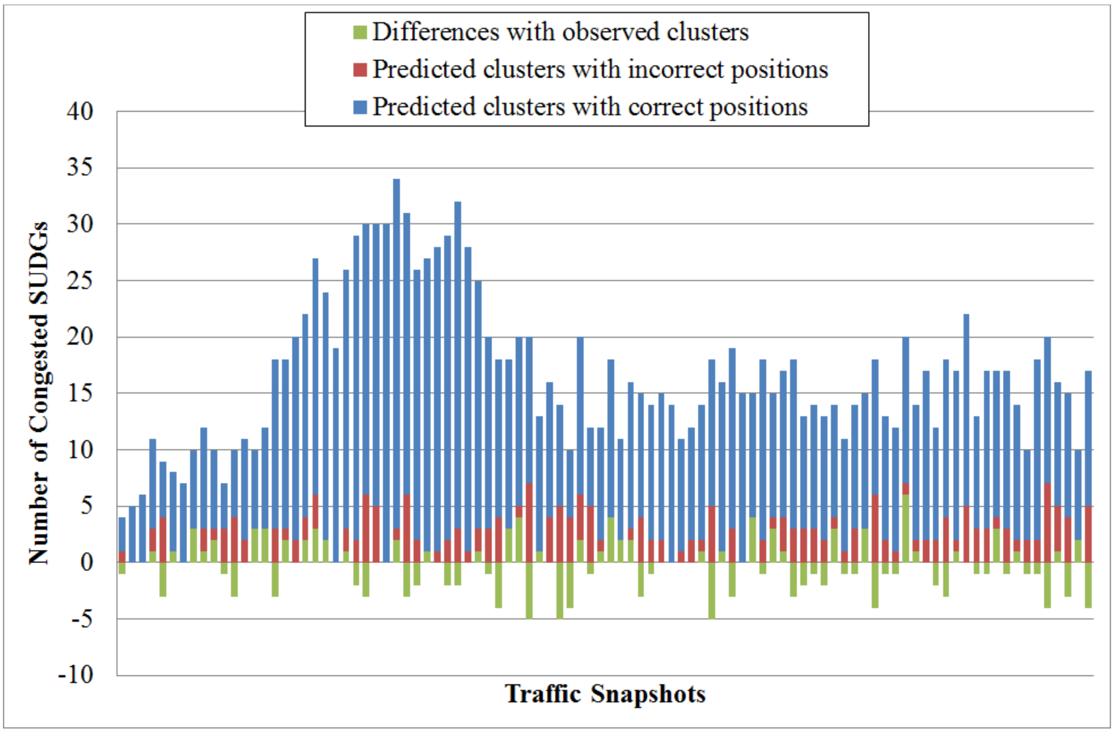


Figure 5.18 Predicted number of congested SUDGs with DTPL, evaluated with 1600 SUDGs (40 × 40)

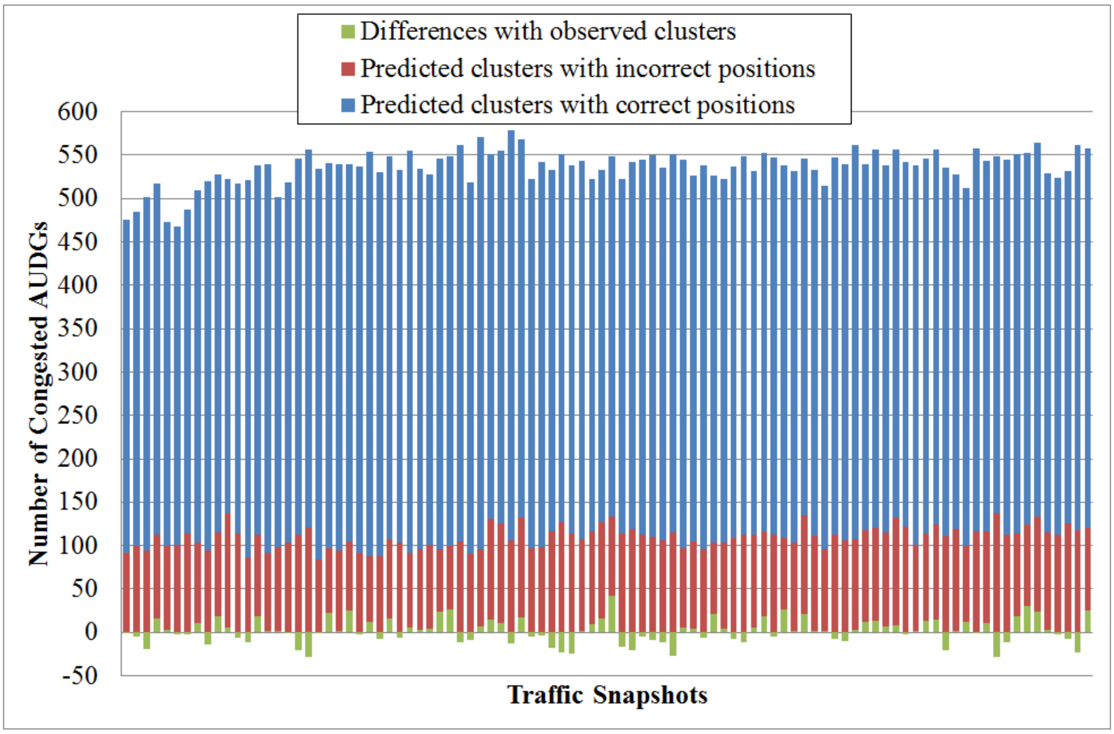


Figure 5.19 Predicted number of congested AUDGs with DTPL, evaluated with 8 sets of AUDGs (36 × 20)

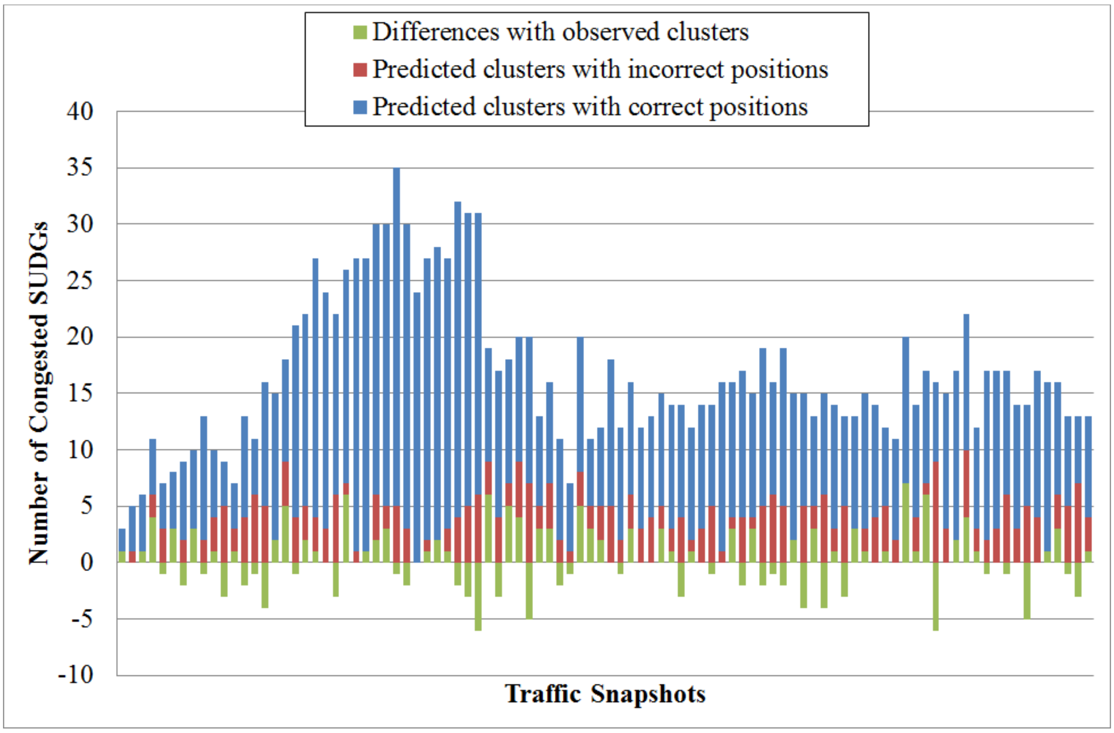


Figure 5.20 Predicted number of congested SUDGs with STPL, evaluated with 1600 SUDGs (40 × 40)

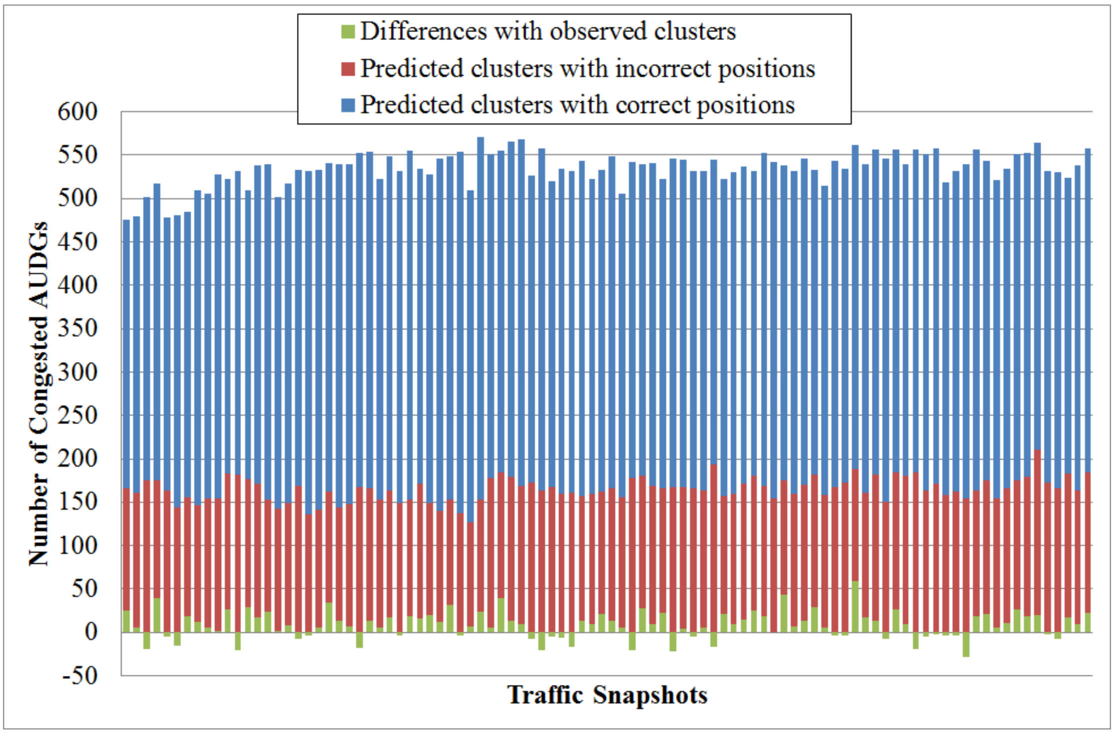


Figure 5.21 Predicted number of congested AUDGs with STPL, evaluated with 8 sets of AUDGs (36 × 20)

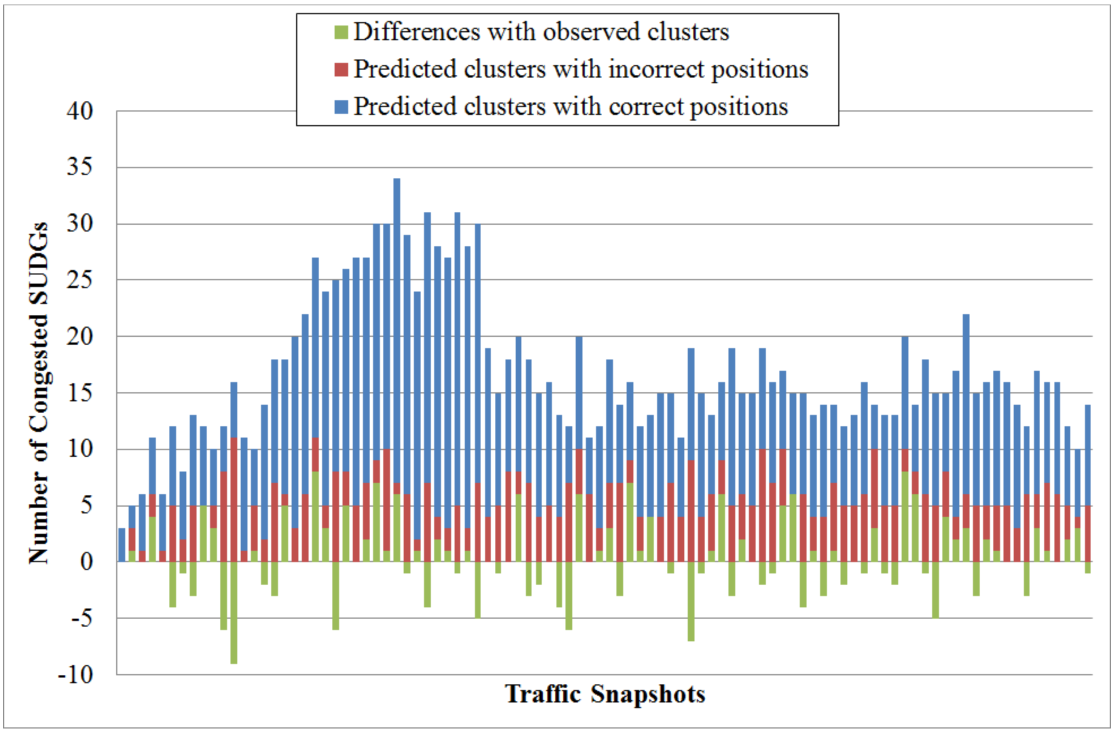


Figure 5.22 Predicted number of congested SUDGs with Order-5 Markov model, evaluated with 1600 SUDGs (40 × 40)

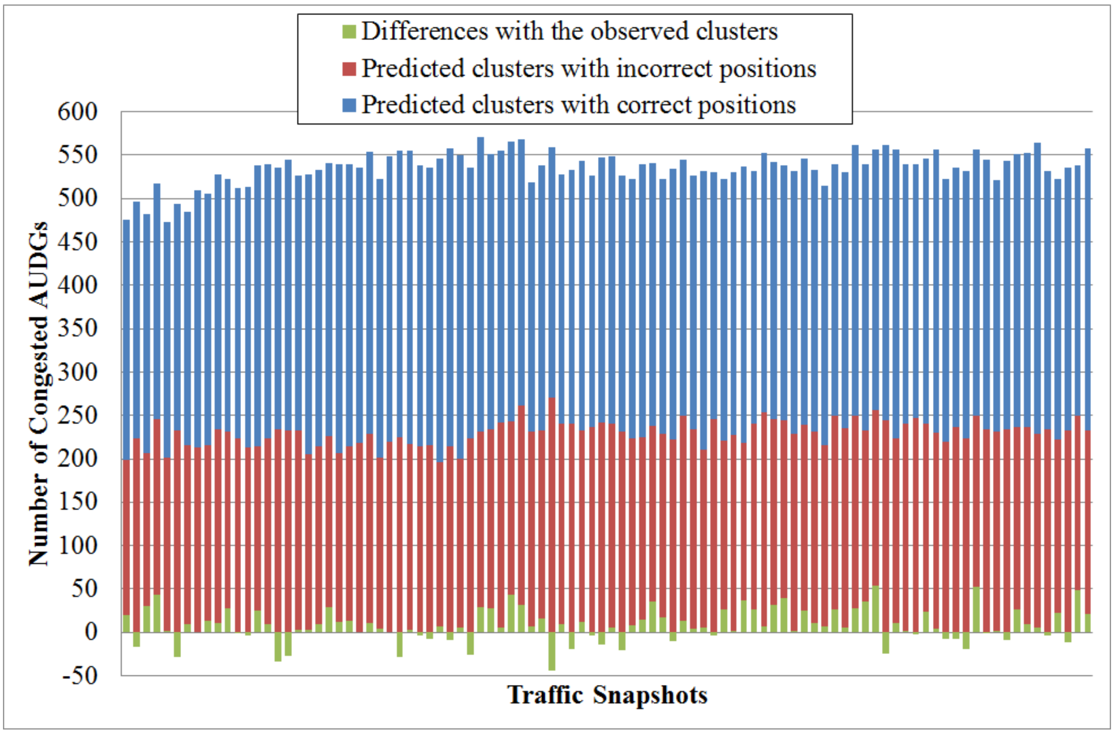


Figure 5.23 Predicted number of congested AUDGs with Order-5 Markov model, evaluated with 8 sets of AUDGs (36 × 20)

Similar to the performances obtained in the normal walking scenario, the best performance can be seen when predictions are made with the DTPL model. With the DTPL model, most of the mobile clusters can be located precisely in terms of the evaluations with SUDGs and AUDGs. Notice that the AUDGs are deployed according to the coverage of the semi-smart antennas, and the applied granularity (36×20) is enough to perform CBR-matching for the traffic patterns, which has been proved by Yao in [Yao07]. With the slight deviation of the population in each segment (0.9339 per segment), the prediction results of the DTPL model could be used in the along with CBR-matching for radiation pattern optimisation. Figure 5.24 shows the comparison between five consecutive observed mobile clusters and the relevant predictions.

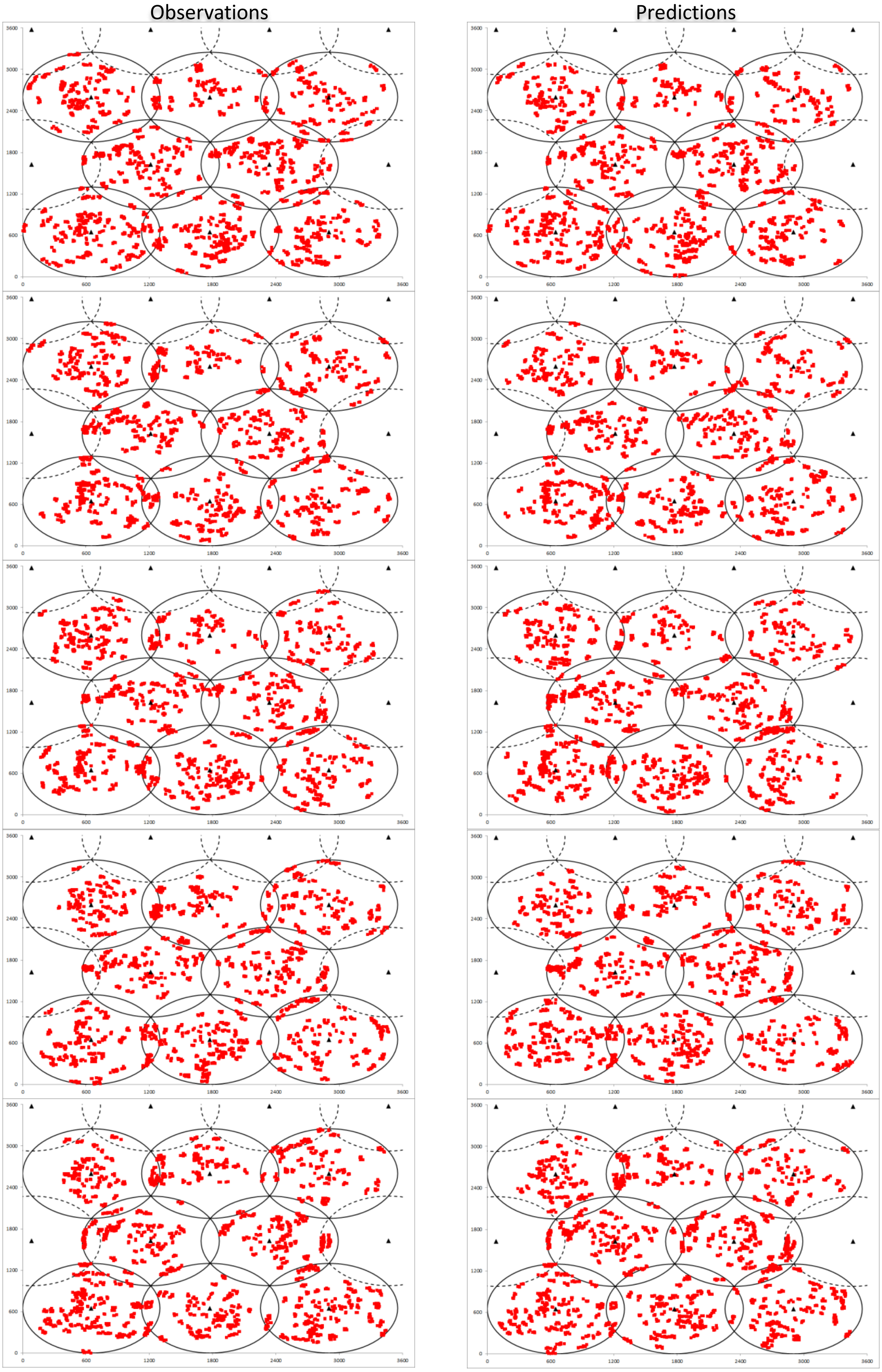


Figure 5.24 Comparison between observed and predicted locations of mobile clusters

5.5 Summary

In order to carry out the traffic pattern prediction in a more realistic environment, the MMP mobility model is proposed to provide observations of mixed traffic patterns. Besides, observations are captured asynchronously relative to the pattern generation to increase the uncertainty and complexity of the prediction task. Then, the DTPL model, an improved version of the STPL model, is proposed to learn traffic patterns and make predictions with fewer requirements in terms of historical data. Finally, simulation results are presented and discussed. According to the results, it can be concluded that the performance of the DTPL is comparatively stable and the accuracy of the prediction can be acceptable.

So far, the applied mobility model can be considered as a semi-microscopic model, as there is no interaction between two mobile terminals. Without inter-user interactions, it is especially difficult to reproduce the real-life traffic patterns for vehicular users. Thus, an advanced cellular automaton model, which is widely used in statistical physics field, is introduced to regenerate the traffic behaviour of vehicles.

6 SIMULATING VEHICULAR TRAFFIC PATTERN WITH A CELLULAR AUTOMATON MODEL

In the mobility prediction field, most research focuses on mining the inter-cell transition data. Generally speaking, inter-cell transition data consists of a sequence of transition events, which are pairs of cell identifiers and transition times. Thus, the main mobility models used for mobility prediction, as mentioned in [CBD02], can be considered as macroscopic models, which just concentrate on the inter-cell location changes of each mobile user, but not the intra-cell location change. In this research, microscopic models are needed to simulation both the inter-cell movement process and the intra-cell movement process. With microscopic models, more detailed traffic patterns can be captured to learn the local movement characteristics.

When designing a mobility model, one difficulty is to let the model reproduce phenomena that happen in real life without excessively increasing the complexity of the model. So far, two mobility models that are built based on the Random Walk Model have been proposed to simulate real life traffic patterns. From a microscopic perspective, the Random Walk Model might be suitable for mimicking the movement of pedestrians, since the size of a walking user could be negligible. However, it is unrealistic if dozens of moving cars gather in a small area. In order to simulate the movement of vehicle users in the city area or in the highway, the cellular automaton model, which is widely used in the field of statistical physics, is introduced to reproduce the traffic behaviour of vehicles.

6.1 An overview of the cellular automaton models

From the perspective of statistical physics, there are two main conceptual frameworks for modelling vehicular traffic in real life [CSS00]. They are the fluid-dynamical description and the microscopic model. In the fluid-dynamical description, traffic is viewed as a compressible fluid formed by the vehicles but these individual vehicles do not appear explicitly in the model. In the microscopic model, attention is explicitly focused on individual vehicles each of which is represented by a non-negligible particle. The movement influence among vehicles is simulated by the interactions among these particles. In the statistical physics field, the cellular automaton (CA) model is the most widely used framework to implement the microscopic models for vehicular traffic. Besides, the CA model also has been applied to mimic the movement of pedestrians, as shown in [BSS98], [ES97], [BA98] and [KMWS00].

In general, the CA models are an idealization of physical systems in which both space and time are assumed to be discrete and each of the interacting units can have only a finite number of discrete states [CSS00]. The concept of CA was introduced in the 1950s by Von Neumann and popularized in the 1960s by Wolfram [Codd68]. Since then the concept of CA has been applied to model a wide variety of systems. The first CA model for vehicular traffic was introduced by Cremer and Ludwig in 1986 [CSS00].

One of the most famous CA models used to simulate vehicular traffic is the Nagel-Schreckenberg (NaSch) model, which is first proposed in [NS92]. In the NaSch model, the positon, speed, acceleration and time are treated as discrete variables. Since this model is built for freeway traffic originally, the vehicle lane is represented by a one-dimensional lattice with *L* sites. Each site may either be occupied by one vehicle, or it may be empty. Nagel and Schreckenberg propose that each car occupies about 7.5m of space in a complete jam. Thus, 7.5m is used as the length of each site. Besides, they also choose 1 second to be the duration between each time stamp. Figure 6.1 illustrate the NaSch model. The number in the right up corner represents the speed of the residing vehicle.

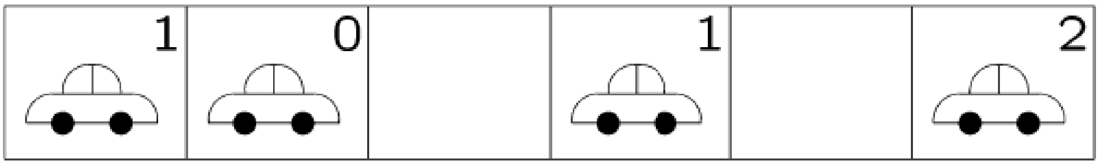


Figure 6.1 An example of the NaSch model

In the NaSch model, the speed v of each vehicle can take one of the $v_{max}+1$ allowed integer values $v=0,1,\dots,v_{max}$. Suppose x_n and v_n denote the position and speed of the *n*th vehicle, then $d_n=x_{n+1}-x_n$ is the gap in between the *n*th vehicle and the vehicle in front of it at time *t*. At each time step $t\rightarrow t+1$, the arrangement of the *N* vehicles on a finite lattice of length L is updated in parallel according to the following rules[NS92]:

- Step 1: Acceleration. If $v_n<v_{max}$, the speed of the nth vehicle is increased by one, but v_n remains unaltered if $v_n=v_{max}$.

$$v_n\rightarrow min(v_n+1,v_{max}) \qquad (6.1)$$

- Step 2: Deceleration. If $d_n\leq v_n$, the speed of the *n*th vehicle is reduced to d_n-1.

$$v_n\rightarrow min(v_n,d_n-1) \qquad (6.2)$$

- Step 3: Randomization. If $v_n>0$, the speed of the *n*th vehicle is decreased randomly by unity with probability *p*.

$$v_n\rightarrow max(v_n-1,0)\ \text{ with probability } p \qquad (6.3)$$

- Step 4: Vehicle movement. Each vehicle is moved forward according to its new velocity determined in Steps 1-3.

$$x_n\rightarrow x_n+v_n \qquad (6.4)$$

As mentioned in [NS92], the NaSch model is a minimal model, since all the above four steps are necessary to reproduce the basic characteristics of the real vehicular traffic. Step 1 reflects the general tendency of the drivers to drive as fast as possible without exceeding the maximum speed limit. Step 2 is used to avoid collision between vehicles. Step 3 takes into account the random behaviour of drivers, such as nondeterministic acceleration and overreaction while slowing down. Step 4 performs the vehicle movement according to the parameters determined in the preceding steps.

Although the NaSch model is a minimal model to mimic only basic freeway traffic features, it is convenient to extend this model to a sophisticated one by appending additional rules and adjusting the order of each rule to reproduce more traffic phenomena. As mentioned in [CSS00], many extensions of the NaSch model have been proposed in recent years, such as the slow-to-start model, the multi-lane highway models, bidirectional traffic model, etc. Inspired by the existing multi-lane models proposed in [RWH05], [CG10], [RNSL96], [KSSS02] and [LJGJ06], this research introduces a Multi-Lane City Traffic (MLCT) model, which can be considered as another extension of the NaSch model.

6.2 Performing traffic pattern prediction in the MLCT model

In this research, a MLCT model, extended from the city traffic model described in [CSS00], is introduced to mimic the movement of mobile users in vehicles. The original city traffic model is a combination of the NaSch model and the Biham-Middleton-Levine (BML) model. In this original model, there is only one traffic lane in each road and the possible movement directions are “east” and “north”. In this model, the number of traffic lanes is increased to two for each direction and four movement directions, “east”, “west”, “north” and “south”, are available for the vehicle users. A detailed description of the MLCT model is given in the following part.

Figure 6.2 illustrates part of the MLCT model. By using the concept of the Manhattan model, the simulation area is divided into M regions. Each region consists of one intersection and four lanes in each direction that are connected with neighbour regions. Between every two intersections, there are D sites in each traffic lane to model the segment of the streets in the same manner as modelling highways in the NaSch model. Thus, the methods used in the NaSch model for describing the speeds, accelerations, and interactions among vehicles can be used here. In this model, two parallel lanes are available for each direction, which means the lane changing rules should be taken into account. Besides, a traffic signal light is put in each intersection to control the traffic flow.

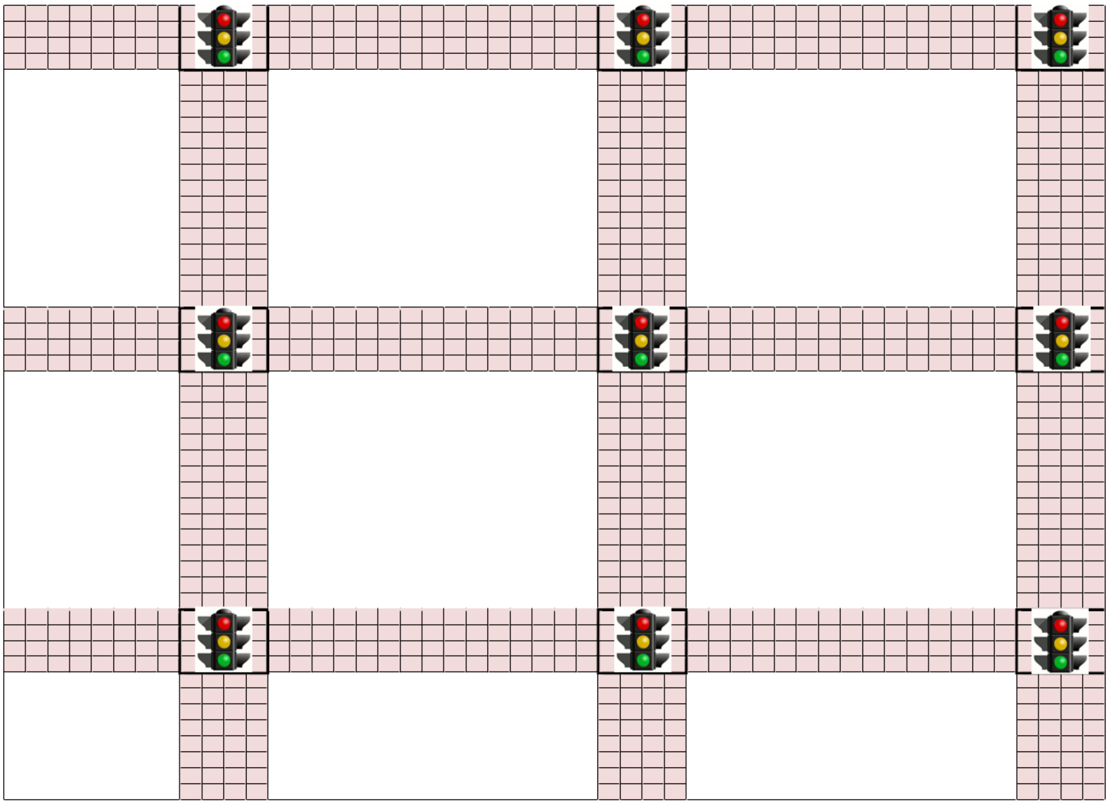


Figure 6.2 An example of the MLCT model

In the MLCT model, the value range of speed v is still $v = 0,1,\dots,v_{max}$. $gap_n = x_{n+1} - x_n$ is the gap in between the *n*th vehicle and the vehicle in front of it on the same lane. gap_{n+} and gap_{n-} are the forward gap and backward gap of the *n*th vehicle on the neighbour lane. p_{change} is the probability of a vehicle to move to the neighbour lane if all the conditions for lane changing are satisfied. gap_{n_sig} is the distance between the current vehicle and the next signal light that it is facing. *Sig* is used to present the colour of the signal light, which has two possible values, *green* and *red*. In the MLCT model, each movement of a vehicle can be split into two sub-steps, moving sideways and moving forward. At each time step $t \rightarrow t+1$, the arrangement of the N vehicles in the four-direction-multi-lane model with periodic boundary is updated in parallel according to the following rules:

- Step 1: Lane changing. A vehicle *n* changes to the neighbour lane with the same traffic direction if all of the following conditions are fulfilled:

$$\begin{cases} gap_n < v_n + 1 \\ gap_{n+} > v_n + 1 \\ gap_{n-} > v_{max} \\ gap_{n\_sig} \geq l_{sig} \\ rand\_uniform(0,1) < p_{change} \end{cases} \tag{6.5}$$

- Step 2: Acceleration. If $v_n < v_{max}$, the speed of the nth vehicle is increased by one, but v_n remains unaltered if $v_n = v_{max}$.

$$v_n \rightarrow min(v_n + 1, v_{max}) \tag{6.6}$$

- Step 3: Deceleration. Suppose τ is the number of the remaining steps before the next signal light turns red. If τ is smaller than a threshold value (usually 5 seconds), the speed of the nth vehicle is determined by:

$$v_n \rightarrow min(v_n, gap_{n\_sig}, gap_n - 1) \tag{6.7}$$

If τ is larger than the threshold value, then v_n is determined by:

$$v_n \rightarrow min(v_n, gap_n - 1) \tag{6.8}$$

- Step 4: Randomization. If $v_n > 0$, the speed of the *n*th vehicle is decreased randomly by unity with probability *p*.

$$v_n \rightarrow max(v_n - 1{,}0) \text{ with probability } p \tag{6.9}$$

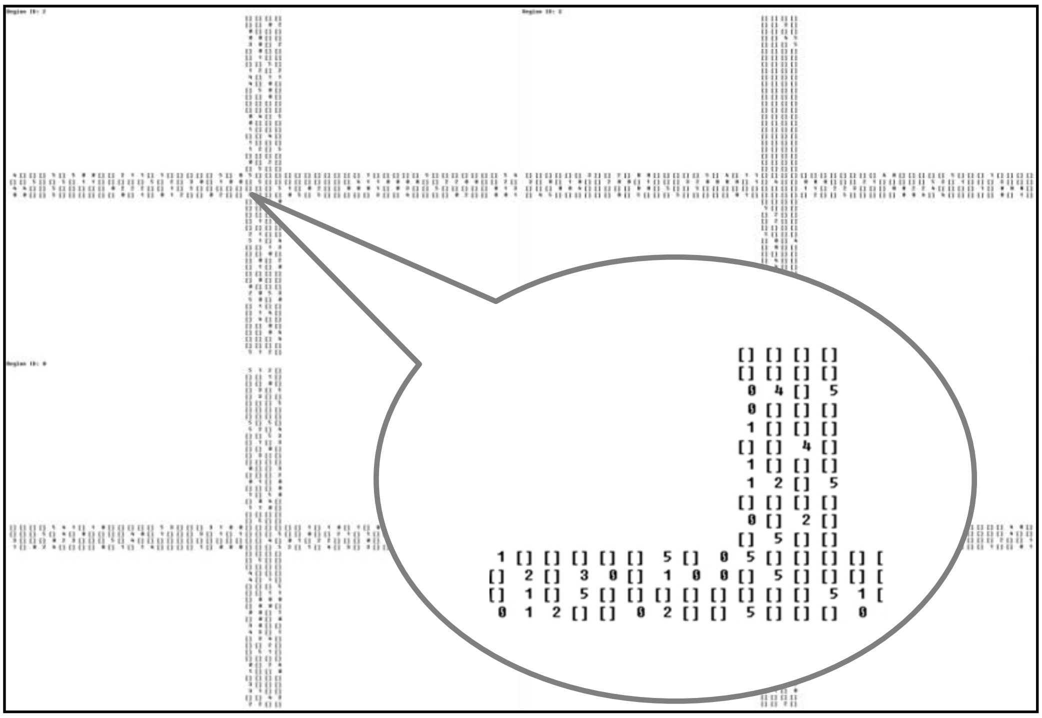
- Step 5: Vehicle movement. Each vehicle is moved forward according to its new velocity determined in Steps 1-4.

$$x_n \rightarrow x_n + v_n \tag{6.10}$$

With the above rules, it is possible for the MLCT model to simulate urban vehicular traffic in a realistic manner. Notice that the position of a vehicle in the MLCT model is denoted by the ID of the site in which the vehicle is staying, but not the Cartesian coordinates. As the movement trajectory of each vehicle can only be presented as a sequence of site IDs, the Order-*k* Markov learning model is applied to learning traffic patterns and make predictions. Learning grids are deployed in the simulation area to capture the $k+1$ length trajectory segments for the Order-*k* Markov models.

6.3 Simulation results

In the simulation, the simulation area was built with 4 regions. All the regions had the same intersection topology, and the number of sites between two intersections was set to 100. The switching period of the signal light in each intersection was set to 20 steps. 700 vehicles were uniformly distributed in the simulation area at the beginning of the simulation. In order to perform the pattern learning and prediction locally, 676 (26×26) learning grids were deployed in each region in the simulation area. In order to evaluate the accuracy of the predicted distribution, 169 (13×13) SUDGs were deployed to cover each region in the simulation area. Figure 6.3 illustrates an example of the simulation area after initialization. Notice that the square brackets represent empty sites; the integers represent occupied sites and the initial speed of the vehicle is an integer value.



```
                           [] [] [] []
                           [] [] [] []
                            0  4 []  5
                            0 [] [] []
                            1 [] [] []
                           [] []  4 []
                            1 [] [] []
                            1  2 []  5
                           [] [] [] []
                            0 []  2 []
                           []  5 [] []
 1 [] [] [] [] []  5 []  0  5 [] [] [] [] [
[]  2 []  3  0 []  1  0  0 []  5 [] [] [] [
[]  1 []  5 [] [] [] [] [] [] [] []  5  1 [
 0  1  2 [] []  0  2 [] []  5 [] [] []  0
```

Figure 6.3 The MLCT model after initialization

In the simulation, the on-line learning mode was applied. The first 3000 snapshots generated by the MLCT model were used to train the deployed Order-6 Markov models without performing predictions. From then on, the historical trajectory library of each learning model was updated, and up to 6-step-ahead predictions were performed for the following 200 snapshots. No traffic event was triggered in the simulation. Figure 6.4 (a) to (f) show the cumulative error rates of the multi-step-ahead predictions in each region. Figure 6.5 shows the overall error rates of the multi-step-ahead predictions.

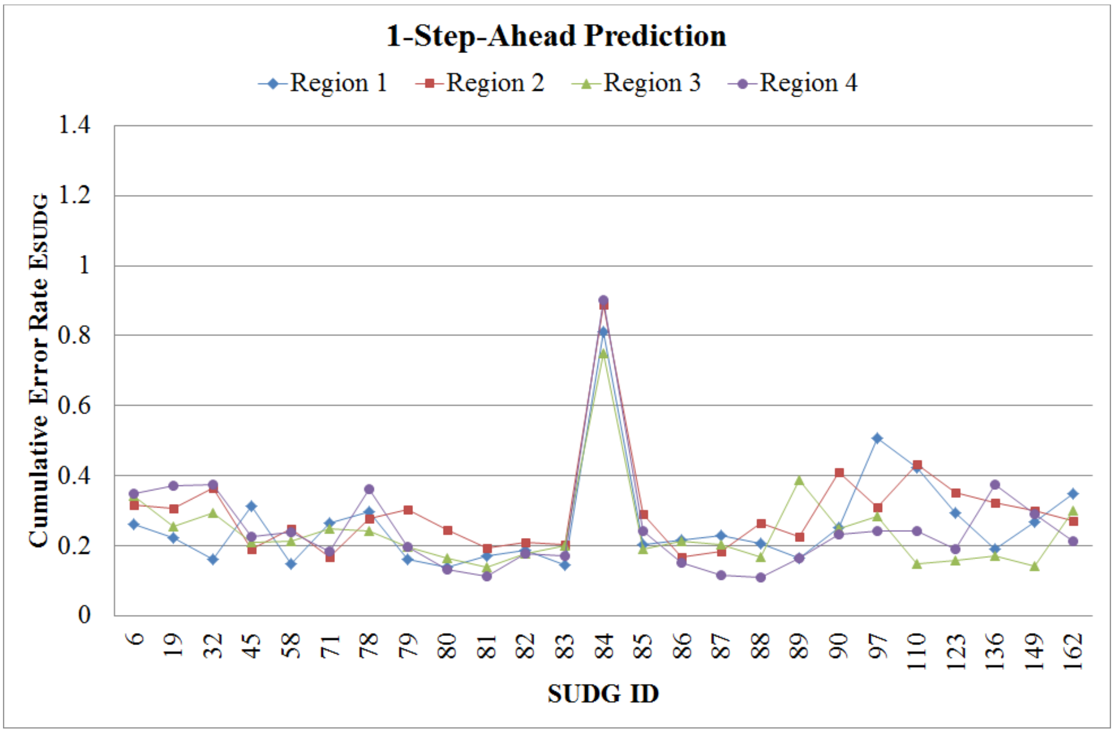


Figure 6.4 (a) Cumulative error rate of one-step-ahead prediction

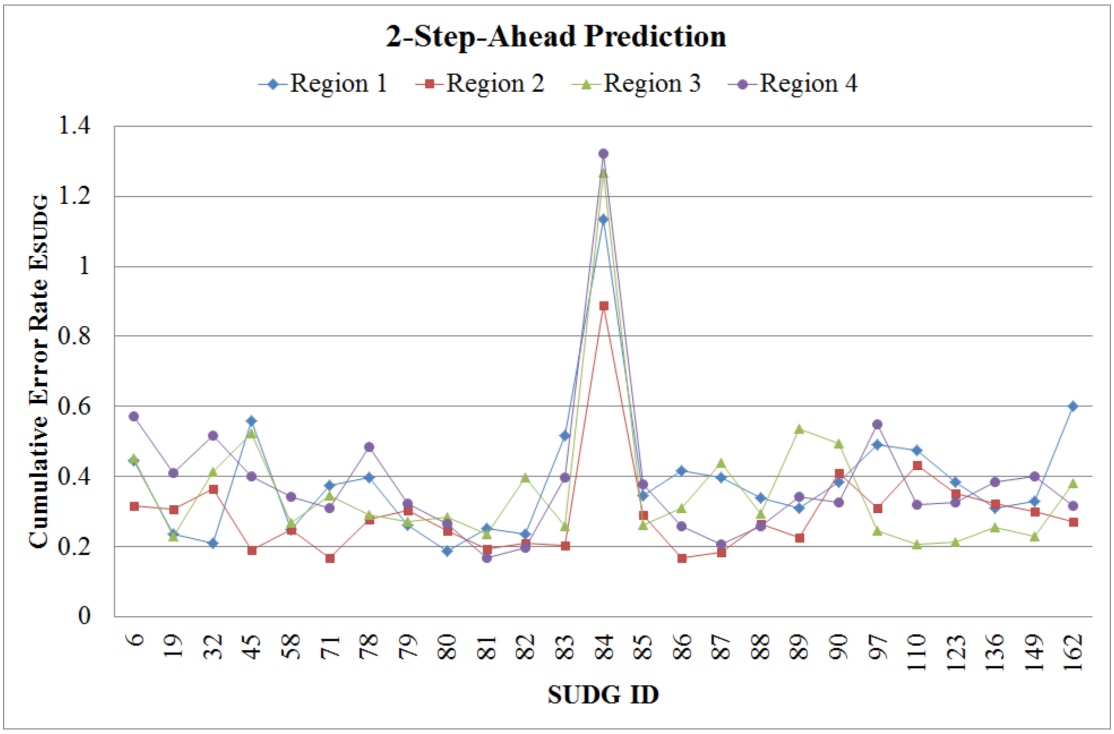


Figure 6.4 (b) Cumulative error rate of two-step-ahead prediction

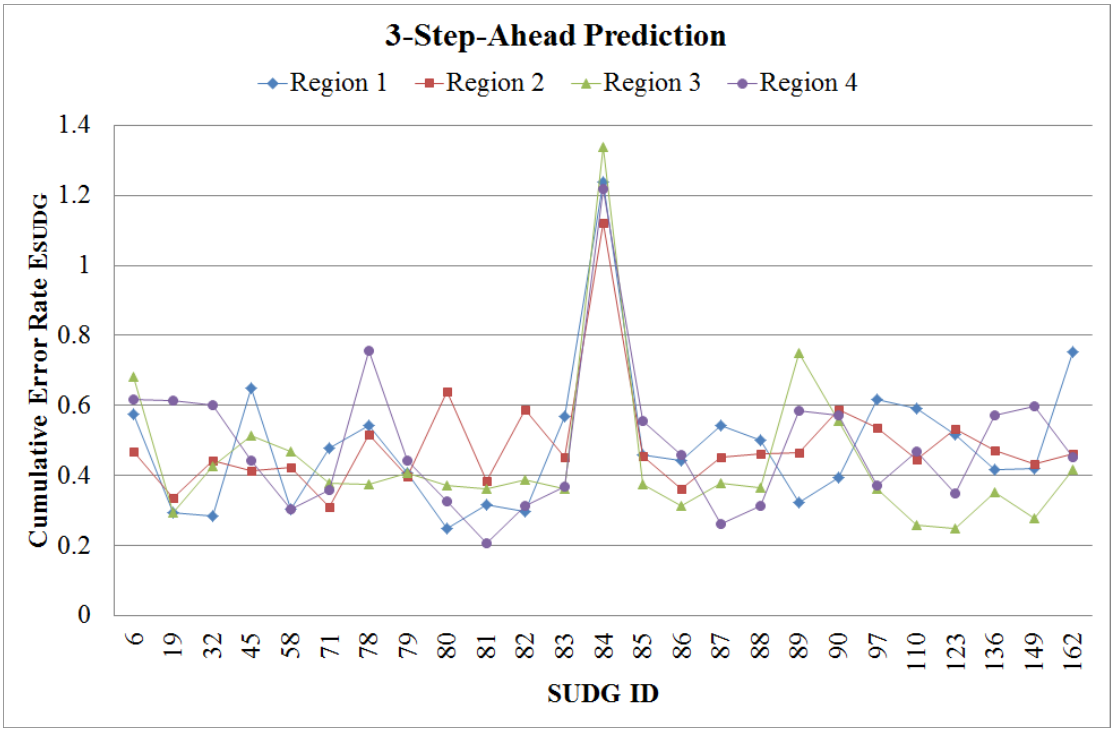


Figure 6.4 (c) Cumulative error rate of three-step-ahead prediction

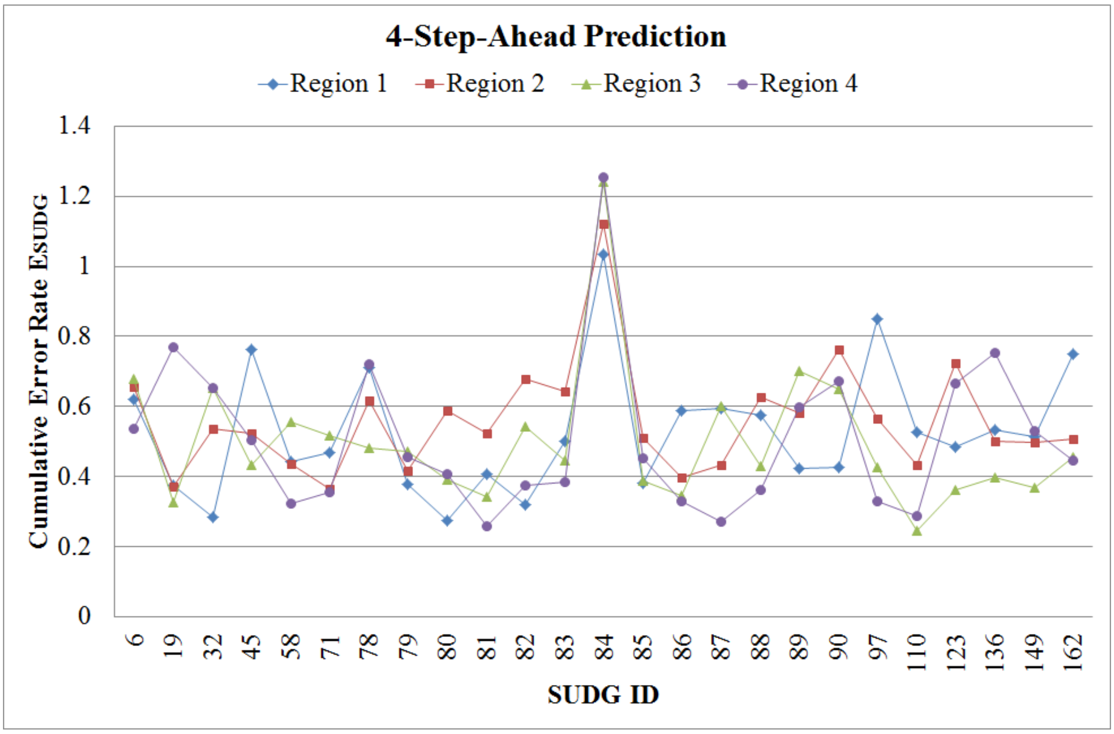


Figure 6.4 (d) Cumulative error rate of four-step-ahead prediction

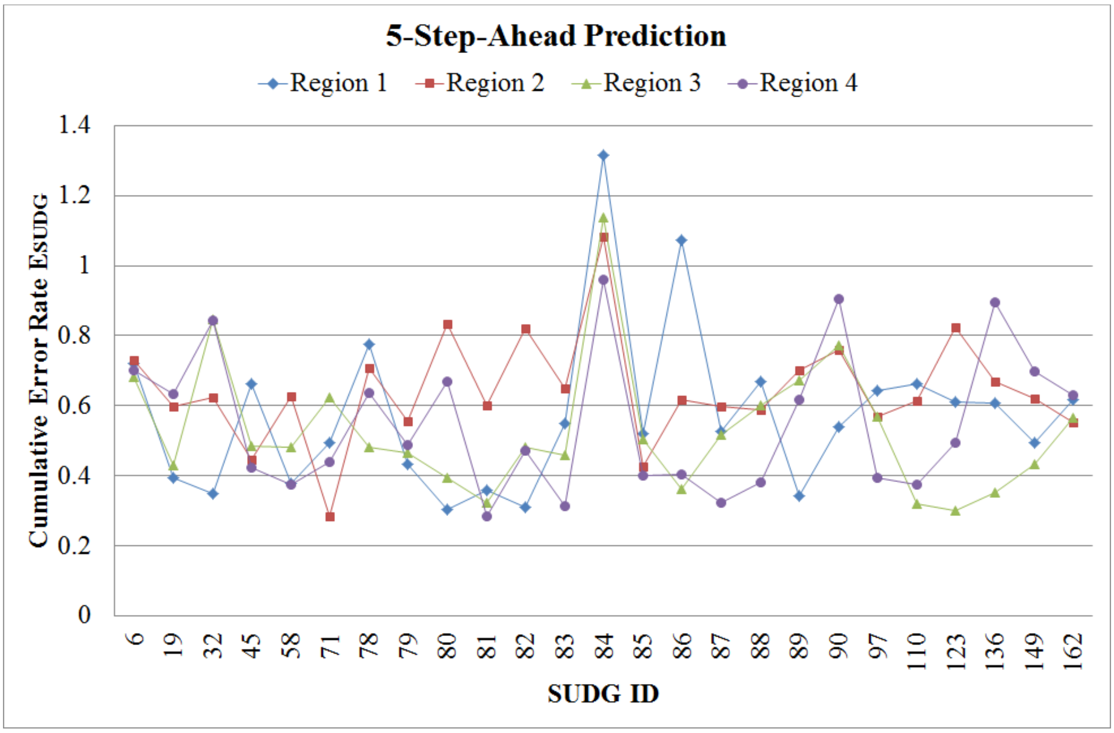


Figure 6.4 (e) Cumulative error rate of five-step-ahead prediction

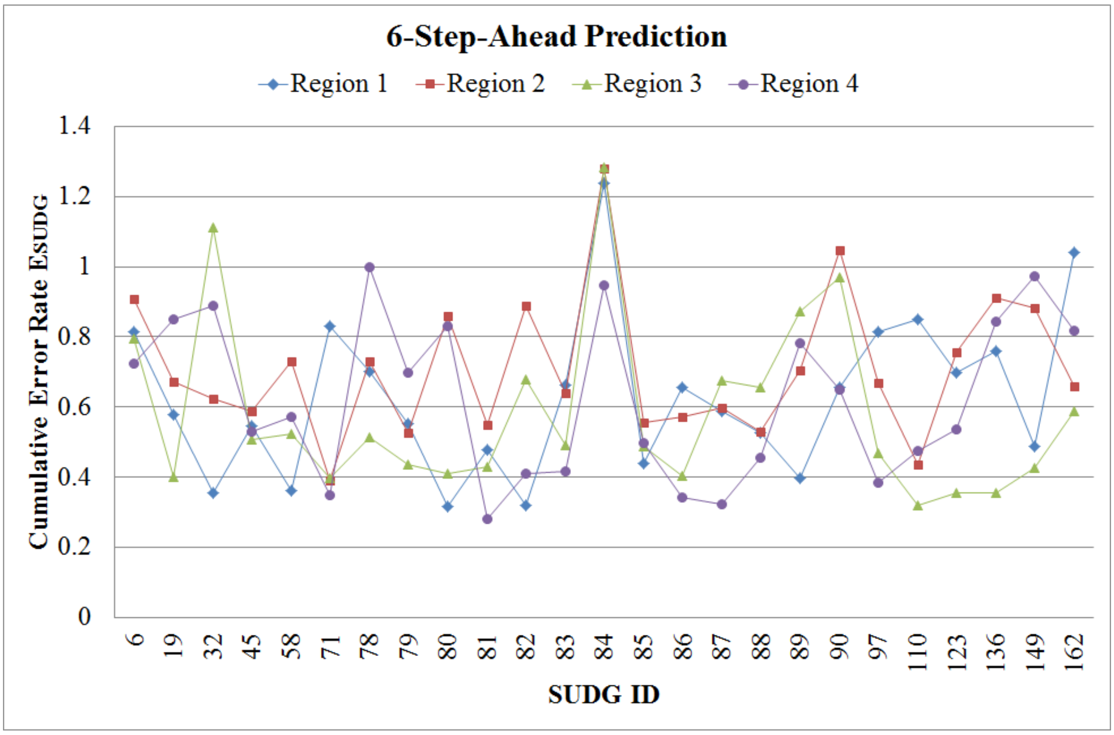


Figure 6.4 (f) Cumulative error rate of six-step-ahead prediction

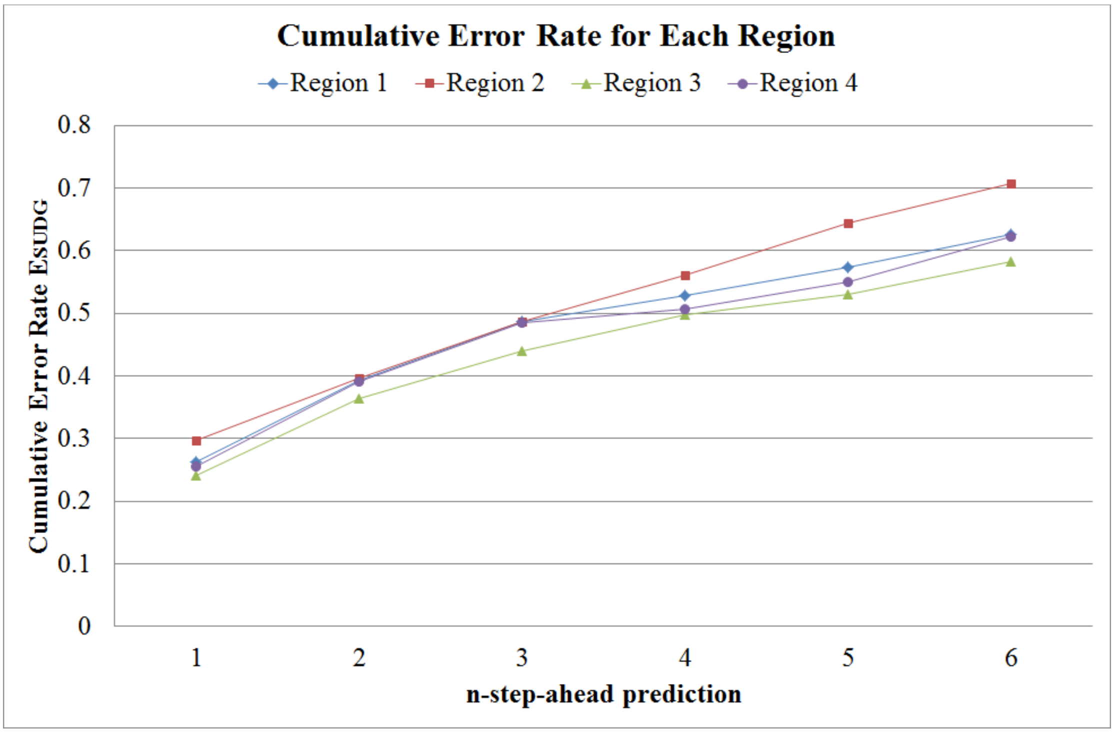


Figure 6.5 Cumulative error rates of each region

In Figure 6.4, the *x*-axis represents the order number of the valid SUDGs; the *y*-axis represents the value of the cumulative error rate. One SUDG is considered as valid if this SUDG covers part of the traffic lane. In Figure 6.5, the *x*-axis represents the number of steps that are predicted ahead; the *y*-axis represents the value of the overall cumulative error rate. Four different colours are used to distinguish the evaluation results of each region.

If only looking over Figure 6.5, it can be seen that the error rate grows from the 1-step-ahead prediction to the 6-step-ahead prediction. The results of the first three predictions can be accepted reluctantly. From the 4-step-ahead prediction, all of the rates are over 0.5. However, a different conclusion can be made after analysing the more detailed results in Figure 6.4. Figure 6.4 shows the cumulative error rates of each valid SUDG in each region. In all sub-figures, a considerably large error rate can be found at SUDG 84 in each region. In some cases, the error rates of UDG 84 are over 1.0. The reason of this phenomenon is that UDG 84 covers the conjunction area of the four-direction traffic. The traffic in this area mainly depends on the status of the signal light, of which the switching period is 20 steps. As the maximal order number in the learning phase was set to 6, information of this switch period of the traffic signal light could not be contained in the learning patterns.

Due to the simple structure of the Order-*k* Markov model, the upcoming vehicular traffic patterns in the MLCT model cannot be predicted accurately. Thus, a more sophisticated predictor should be designed and applied to learn the traffic patterns in the MLCT model.

7 CONCLUSION

In this chapter, the conclusion of the research presented in this thesis is given. Then, relevant work that could extend this research is discussed.

7.1 Conclusion

This thesis has proposed an intra-cell traffic pattern prediction scheme for forecasting the distribution of mobile users and mobile clusters from microscopic perspectives, rather than the existing macroscopic mobility predictions that focus on the inter-cell transition status of mobile terminals. With accurate predictions of the upcoming geographic distribution of mobile users in each cell, semi-smart antenna techniques can be applied to guide the beamforming pattern adjustment so as to alleviate congestion in cellular networks.

In order to implement this microscopic traffic pattern prediction, two mobility models are proposed originally in this research for generating intra-cell traffic patterns. Sophisticated walking rules are applied to let the traffic patterns be more realistic than patterns generated by the existing mobility models. The cellular automaton model is introduced to build the mobility model with inter-user interactions, which is mainly for reproducing the traffic patterns of mobile users in vehicles.

The concept that learning the local geographic features and local movement patterns of mobiles users with cell transition trajectories so as to make predictions on intra-cell distribution is proposed in this research. By learning local features, the intra-cell traffic patterns can be revealed and future traffic patterns can be estimated. Three pattern learning models are proposed to mine the cell transition trajectories in a generalized way. With the STPL model, the local movement characteristics can be learnt from many perspectives; the future geographic traffic distribution can be well estimated; and the location of mobile clusters can be accurately identified; the convergence time for irregular traffic patterns is short. With the MTPL model, no strict requirements are need for the observed traffic; and local traffic characteristics can be learnt from a mix of different traffic pattern observations. According to the simulation results, future distributions of mobile users and mobile clusters can be predicted with high accuracy.

7.2 Future Work

The reason of performing the intra-cell traffic pattern prediction is to benefit the beamforming pattern adjustment of semi-smart antennas with accurately predicted future distribution of mobile terminals. However, cooperation between the intra-cell traffic pattern prediction and load balancing of the semi-smart antennas is not included in this research. It is worth to extend the result of this research by letting the semi-smart antenna operating cooperatively with the local pattern learning models.

More sophisticated pattern learning models are needed to increase the accuracy of the predictions made for the traffic patterns generated in the cellular automaton model. A set of Markov models with different delays is recommended for extending this part of the research.

REFERENCES

[AHMES09] A. Aymen, K. Houda, A. Mohamed, Ez-Zahraouy and T. Sami, “Probabilistic model for mobility in cellular network subscriber,” the 1st International Conference on Wireless Communication, Vehicular Technology, Information Theory and Aerospace & Electronic Systems Technology (Wireless VITAE 2009), pp. 351 – 355, 2009

[AKAET09] A. Ayari, H. Khedher, M. Ayadi, H. Ez-Zahraouy and S. Tabbane, “Modeling of Cellular Network Subscriber Mobility,” the Fifth Advanced International Conference on Telecommunications (AICT09), pp. 447 – 452, 2009

[AS07a] S. Akoush and A. Sameh, "Movement Prediction Using Bayesian Learning for Neural Networks," the 2nd International Conference on Systems and Networks Communications (ICSNC 2007), 2007.

[AS07b] S. Akoush and A. Sameh, "Bayesian Learning of Neural Networks for Mobile User Position Prediction," in Proceedings of 16th International Conference on Computer Communications and Networks (ICCCN 2007), pp.1234 - 1239, 2007.

[AW04] I. F. Akyildiz and W. Wang, “The predictive user mobility profile framework for wireless multimedia networks,” IEEE/ACM Transactions on Networking (TON), Vol. 12, pp. 1021-1035, December 2004.

[AWS06] J. Ariyakhajorn, P. Wannawilai, and C. Sathitwiriyawong, "A Comparative Study of Random Waypoint and Gauss-Markov Mobility Models in the Performance Evaluation of MANET," International Symposium on Communications and Information Technologies (ISCIT06), pp. 894 - 899, 2006.

[AZ01] A. Aljadhai and T. F. Znati, “Predictive Mobility Support for QoS Provisioning in Mobile Wireless Enviornments,” IEEE Journal on

Selected Areas in Communications (JSAC), Vol. 19, No. 10, pp. 1915-1931, October 2001.

[BA98] V. Blue and J. Adler, “Emergent Fundamental Pedestrian Flows from Cellular Automata Microsimulation,” Transportation Research Board, pp. 29-36, 1998.

[BBK11] D. Barth, S. Bellahsene and L. Kloul, "Mobility Prediction Using Mobile User Profiles," IEEE 19th International Symposium on Modeling, Analysis & Simulation of Computer and Telecommunication Systems (MASCOTS), pp.286 - 294, 2011.

[BD99] A. Bhattacharya and S. K. Das, “LeZi-Update: An Information-Theoretic Approach to Track Mobile Users in PCS Networks,” ACM/IEEE MobiCom’99, August 1999.

[BD02] A. Bhattacharya and S. K. Das, “LeZi-Update: An Information-Theoretic framework for personal mobility tracking in PCS Networks,” Wireless Networks, Vol. 8, pp. 121-135, 2002

[BEJ97] J. Biesterfeld, E. Ennigrou, K. Jobmann, “Neural networks for location prediction in mobile networks,” in Proceedings of International Workshops on Application of Neural Networks to Telecommunications 97, pp. S207-S214, Melbourne, June 1997.

[BJH96] S. Benjamin, N. Johnson, and P. Hui, “Cellular Automata models of traffic flow along a highway containing junction,” Journal of Physics A: Mathematical and General 29, pp.3119-3127, 1996.

[BK10] S. Bellahsene and L. Kloul, "A New Markov-Based Mobility Prediction Algorithm for Mobile Networks," the 7th European Performance Engineering Workshop (EPEW), pp.37-50, Italy, 2010.

[BKM94] A. Bar-Noy, I. Kessler, and M. Sidi, “Mobile users: To update or not to update?,” in Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pp. 570-576, 1994.

[BSS98] R. Barlovic, A. Schadschneider and M. Schreckenberg, "Metastable States in Cellular Automata for Traffic Flow," European Physical Journal, pp 793-800, 1998.

[CB04] J. Capka and R. Boutaba, "Mobility Prediction in Wireless Networks using Neural Networks," in Proceedings of the IFIP/IEEE International Conference on the Management of Multimedia Networks and Services (MMNS04), LNCS 3271, pp. 320-334, 2004

[CBCS02] G. Chakraborty, B.B. Bista, D. Chakrabort and N. Shiratori, "Location manage- ment in PCN by movement prediction of the mobile host," in Proc. IEEE International Symposium on Industrial Electronics, Vol. 1, pp. 78-83, July 2002.

[CBD02] T. Camp, J. Boleng, and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," Wireless Communication & Mobile Computing, Vol. 2, no. 5, pp. 483-502, 2002.

[CG10] J. Chen, and X. Guo, "Influences of overtaking on two lane traffic with signals," Physica A: Statistical mechanics and its applications, pp.141-148, 2010.

[Chak02] G. Chakrabarty, "Efficient Location Management by Movement Prediction of Mobile Host," Proc. Int. Workshop on Distributed Computing IWDC 2002, Lecture Nules in CS, LNCS 2571, pp. 142-153, 2002.

[Chry00] M. Chryssomallis, "Smart antennas," IEEE Antennas and Propagation Magazine, vol. 42, no. 3, pp. 129-136, June 2000.

[CJS04] R. Chellappa, A. Jennigs, and N. Shenoy, "A review on current work in mobility prediction for wireless networks," in Proceedings of the 3rd Asian International Mobile Computing Conference, 2004.

[Codd68] E. F. Codd, Cellular Automata, Academic Press, New York, 1968.

[Comp88] R. T. Compton, "Adaptive Antennas: Concepts and Performance," Englewood Cliffs, New Jersey, Prentice-Hall, 1988.

[CRTBB01] L. G. Cuthbert, D. Ryan, L. Tokarchuk, J. Bigham and E. Bodanese, "Using Intelligent Agents to Manage Resource in 3G Networks", Journal of IBTE, vol. 2 part 4, 2001.

[CSS00] D. Chowdhury, L. Santen, and A. Schadschneider, "Statistical physics of vehicular traffic and some related systems," Phys. Rep. pp. 199-329, 2000.

[CST96] P. Camarda, G. Schiraldi and F. Talucci, "Mobility modeling in cellular communication networks," in Proceedings of the 21st IEEE Conference on Local Computer Networks, pp. 518 – 525, 1996.

[CZDTZ04] R. Covaja, A. Zanella, M. Dossi, A. Tontoli, and P. Zennaro, "Experimental Performance of the Handover Procedure in a WiFi Network", in Proceedings of WPMC'04, September 2004.

[DBC03] L. Du, J. Bigham and L. Cuthbert, "Utility-based distributed geographic load balancing in mobile cellular networks," in Proceedings of the Fourth International Conference on Mobile Communication Technologies (IEE 3G2003), pp. 58-62, June 2003.

[DBC04] L. Du, J. Bigham and L. Cuthbert, "A bubble oscillation algorithm for distributed geographic load balancing in mobile networks," Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE INFOCOM2004), vol. 1, pp. 330-338, March 2004.

[DBCNP03] L. Du, J. Bigham, L. Cuthbert, P. Nahi and C. Parini, "Intelligent cellular network load balancing using a cooperative negotiation approach," Wireless Communications and Networking (WCNC03), vol. 3, pp. 1675-1679, 2003.

[DH10] Y. Dan and Z. He, "A dynamic model for urban population density estimation using mobile phone location data," the 5th IEEE Conference on Industrial Electronics and Applications (ICIEA), pp. 1429 – 1433, 2010.

[Du04] L. Du, Intelligent Geographic Load Balancing for Mobile Cellular Networks, PhD thesis, Queen Mary, University of London, London, UK, July 2004.

[EB08] G. Eibl and N. Brandle, “Evaluation of clustering methods for finding dominant optical flow fields in crowded scenes,” the 19th International Conference on Pattern Recognition (ICPR 2008), pp. 1-4, 2008.

[ES97] J. Esser and M. Schreckenberg, “Microscopic Simulation of Urban Traffic: Based on Cellular Automata,” International Journal of Modern Physics, pp 1025-1036, 1997.

[FL07] J. Francois and G. Leduc, “AP and MN-centric Mobility Prediction: A Comparative Study based on Wireless Traces”, in Proceedings of Networking 2007, May 2007.

[FNNAK10] T. Fukabori, H. Nakayama, H. Nishiyama, N. Ansari and N. Kato, “An Efficient Data Aggregation Scheme Using Degree of Dependence on Clusters in WSNs,” IEEE International Conference on Communications (ICC), pp. 1-5, 2010.

[GL04] V. K. Garg and S. R. Laxpati, “Use of smart antenna system in universal mobile communication systems (UMTS),” IEEE Antennas and Wireless Propagation Letters, vol. 3, no. 1, pp. 66-70, 2004.

[GM99] J.J. Garicia-Luna-Aceves and E.L. Madrga, “A multicast routing protocol for ad-hot networks,” in Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), pp. 784-792, 1999.

[Gurn03] K. Gurney, An Introduction to Neural Networks, pp. 1, CRC Press, 2003.

[GV06] A. Gellert and L. Vintan, “Person Movement Prediction Using Hidden Markov Models,” Studies in Informatics and Control, Vol.15, pp.17-30, 2006.

[HKA04] T. H, D. Kotz and I. Abyzov, “The changing usage of a mature campus-wide wireless network,” in Proceedings of the 10th Annual International Conference on Mobile Computing and Networking, pp. 187-201, NY, USA, 2004.

[HMA06] A.S. Hassani, A.R. Momen and P. Azmi, “Mobility Model of Vehicular Terminals in Cellular Networks,” the 2nd Information and Communication Technologies (ICTTA '06), Vol.2, pp. 2434 – 2437, 2006.

[HT04] H. Holma and A. Toskala, WCDMA for UMTS: Radio Access for Third Generation Mobile Communications Third Edition, Wiley & Sons, Ltd., 2004.

[ITV09] S. Intarasothonchun, S. Thipchaksurat and R. Varakulsiripunth, "Effect of Shadow Cluster on Predictive User Mobility Behavior Scheme in Cellular Networks," the 4th International Conference on Computer Sciences and Convergence Information Technology (ICCIT09), pp.835 - 840, 2009.

[JHSRP10] T. Jeong, S. Han, Y. Song, S. Rhee and G. Park, "Mobility Prediction Modeling and Analysis for People in Mobile Wireless Network," in Proceedings of the 5th International Conference on Ubiquitous Information Technologies and Applications (CUTE10), pp.1-5, 2010.

[JWH05] R. Jiang, Q. Wu, and M. Hu, “Honk effect in the two-lane cellular automaton model for traffic flow,” Physica A: Statistical mechanics and its applications, pp. 544-552, 2005.

[KIS09] A. Kanda, K. Ishii, M. Sato, “Environment recognition system based on multiple classification analyses for mobile robots,” IEEE International Conference on Systems, Man and Cybernetics (SMC09), pp. 2782 – 2787, 2009

[KMWS00] H. Klüpfel, T. Meyer-König, and M. Schreckenberg, “Microscopic modeling of pedestrian motion – comparison of an evacuation exercise in a primary school to simulation results," the Fourth International Conference on Celular Automata for Research and Industry, 2000.

[KSSS02] W. Knospe, L. Santen, A. Schadschneider, and M. Schrekenberg, "A realistic two-lane traffic model for highway traffic," Journal of Physics A: Mathematical and General 35, pp. 3369-3388, 2002.

[Laas05] K . Laasonen, Clustering and Prediction of Mobile User Routes from Cellular Data, Proc. PKDD, 2005.

[LBC98] T. Liu, P. Bahl, and I. Chlamtac, "Mobility Modelling, Location Tracking, and Trajectory Prediction in Wireless ATM Networks," IEEE J. Selected Areas in Comm., vol. 16, pp. 922-936, Aug. 1998.

[LH99] B. Liang and Z. Haas, "Predictive distance-based mobility management for PCS networks," in Proceedings of the Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), March 1999.

[LJGJ06] X. Li, B. Jia, Z. Gao, and R. Jiang, "A realistic two lane cellular automaton traffic model considering aggressive lane changing behavior of fast vehicles,". Physics A: Statistical mechanics and its applications, pp. 479-486, 2006.

[LK97] A.M. Law and W.D. Kelton, Simulation Modelling and Analysis, 2nd Edition, McGraw-Hill Higher Education, 1997.

[NS92] K. Nagel and M. Schreckenberg, "A cellular automaton model for freeway traffic," J. Phys I France 2, pp. 2221, 1992.

[OP98] T. Ojanpera and R. Prasad, Wideband CDMA for Third Generation Mobile Communications, Artech House, 1998.

[PCAB06] C. Parini, X. Chen, Y. Alfadhl, J. Bigham, I. Llewellyn, L. G. Samuel, L. Ho and B. Collins, Final Report on Semi-Smart Antenna Technology Project, July 2006.

[PSJ04] P.N. Pathirana, A.V. Savkin and S. Jha, "Location estimation and trajectory prediction for cellular networks with mobile base stations," IEEE Transactions on Vehicular Technology, Vol.53, pp.1903 - 1913, 2004.

[Quin05] A. Quintero, "A user pattern learning strategy for managing users' mobility in UMTS networks," IEEE Transactions on Mobile Computing, Vol.4, pp.552 - 566, 2005.

[QWZW10] C. Qiu, C. Wang, X. Zuo and J. Wu, “A Bayesian regularized neural network approach to short-term traffic speed prediction”, the 13th International IEEE Annual Conference on Intelligent Transportation Systems, pp. 1529 – 1534, September, 2010.

[RC97] I. Rubin and C. Choi, “Impact of the location area structure on the performance of signaling channels in wireless cellular networks”, IEEE Communications Magazine, pp. 108-115, 1997.

[RNSL96] M. Rickert, K. Nagel, M. Schreckenberg, and A. Latour “Two lane traffic simulations using Cellular Automata,”.Physica A: Statistical and Theoretical, pp. 534-550, 1996

[SB07] M. Sun and D. Blough, "Mobility prediction using future knowledge," in Proceedings of the 10th ACM symposium on modelling, analysis, and simulation of wireless and mobile systems, Greece, 2007.

[SC01] C. Smith and D. Collins, 3G Wireless Networks, McGraw-Hill, Inc., 2001.

[SK04] W. Soh and H. Kim, "Dynamic bandwidth reservation in cellular networks using road topology based mobility predictions," the 23rd AnnualJoint Conference of the IEEE Computer and Communications Societies, Vol.4, pp.2766 - 2777, 2004.

[SK99] J. Scourias and T. Kunz, “Activity-based mobility modeling: realistic evaluation of location management schemes for cellular networks” IEEE Wireless Communications and Networking Conference, Vol. 1, pp. 296 – 300, 1999.

[SKJH04] L. Song, D. Kotz, R. Jain and X. He, “Evaluating location predictors with extensive Wi-Fi mobility data,” in Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM), vol. 2, pp. 1414-1424, March 2004.

[SMBS10] S. Sadhukhan, S. Mandal, P. Bhaumik, and D. Saha, "A novel direction-based diurnal mobility model for handoff estimation in cellular networks," Annual IEEE India Conference (INDICON), pp.1-5, 2010.

[SVA06] M. Sricharan, V. Vaidehi and P. Arun, "An activity based mobility prediction strategy for next generation wireless networks," 2006 IFIP International Conference on Wireless and Optical Communications Networks, pp.1-5, 2006.

[SWYS10] H. Si, Y. Wang, J. Yuan and X. Shan, "Mobility Prediction in Cellular Network Using Hidden Markov Model," the 7th IEEE Consumer Communications and Networking Conference (CCNC10), pp.1-5, 2010.

[TAP00] G. V. Tsoulos, G. E. Athanasiadou and R. J. Piechocki, “Low-complexity smart antenna methods for third-generation W-CDMA systems,” IEEE Transactions on Vehicular Technology, vol. 49, no. 6, pp. 2382-2396, November 2000.

[WY02] W. Wang and I.F. A. Yildiz, “On the estimation of user mobility pattern for location tracking in wireless networks”, IEEE Global Telecommunications Conference (GLOBECOM '02), vol. 1, pp. 610 – 614, 2002.

[Yao07] N. Yao, A CBR Approach for Radiation Pattern Control in WCDMA Networks, PhD thesis, Queen Mary, University of London, London, UK, January 2007.

[YC05] N. Yao and L. Cuthbert, “Resource management in 3G networks using case-based reasoning,” in Proceedings of the 8th ACM/IEEE

International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile System (MSWiM2005), pp. 307-312, October 2005.

[YC06] N. Yao and L. Cuthbert, “Prediction of Antenna Patterns over Hotspot cluster in WCDMA Networks,” in Proceedings of the sixth IASTED International Multi-conference on Wireless and Optical Communications (WOC 2006), Banff, Canada, July 2006.

[YC07] N. Yao and L. Cuthbert, “Reducing congestion over hotspot clusters in WCDMA networks,” in Proceedings of the IEEE Wireless Communications and Networking Conference 2007 (WCNC2007), pp. 3734-3738, March 2007.

[YL02] F. Yu and V. Leung, “Mobility-based predictive call admission control and band- width reservation in wireless cellular networks,” Computer Networks, Elsevier Science, Vol. 38, pp. 577-589, April 2002.

[YWZCX11] X. Yang, Y. Wang, T. Zhang, L. Cuthbert and L. Xiao, “Combining CoMP with semi-smart antennas to improve performance,” Electronic Letter 47(13): 775-U70 23, 2011.

[ZD97] M. Zonoozi and P. Dassanayake, “User mobility modeling and characterization of mobility pattern,” IEEE Journal on Selected Areas in Communications, pp. 1239-1252, 1997.

[ZM05] Z. Zaidi and B. Mark, "Real-time mobility tracking algorithms for cellular networks based on Kalman filtering," IEEE Transactions on Mobile Computing, Vol.4, pp.195 - 208, 2005.

APPENDIX

In this section, the test procedures that undertaken to provide evidence that the simulation tools functioned correctly are presented first, followed by the introduction of the Random Number Generator (RNG) used in the simulation.

A. Simulation Validation

All the simulation tools used in this research are written from scratch in C++. In order to test the correctness of the codes, a set of validation checks are planted into the simulation tools to output intermediate results. When carrying out the validation, those intermediate results are checked manually.

In the pattern generating phase, the validation checks are as follows:

- In each step, the position of an RWP should not be outside the boundary of the simulation area;
- The movement direction should not be changed if the time of direction change is larger than a threshold;
- Intermediate results when invoking Equation 3.4, 3.5 and 3.6 are correct.

In the pattern learning phase, the validation checks are as follows:

- The movement contexts generated for each pattern learning model are correct and complete;
- The number of substrings obtained with Equation 3.9 and 3.14 are the same as the manually counted number;
- When using the STML model, intermediate results with Equation 3.10 and 3.11 are correct;
- When using the DTML model, intermediate results when extracting neighbouring movement trajectories are correct;

In the pattern prediction phase, the validation checks are as follows:

- The k-length most recent movement trajectories generated for making predictions are correct;
- When applying the second decision rule in the Markov model and the STML model, intermediate results are correct and the value of the target variable is chosen following the relevant probability distribution;
- When using the DTML model, intermediate results when invoking Equation 5.1, 5.2, 5.3, 5.4, 5.5 and 5.6 are correct.

B. The Mersenne Twister Random Number Generator

The Mersenne Twister RNG was first proposed by Makoto Matsumoto and Takuji Nishimura in 1997 [1] [2]. This generator is based on a matrix linear recurrence over a finite binary field. Many variants are widely used, such as the Mersenne Twister MT19937, the Mersenne Twister MT19937-64, and the SIMD-Oriented Fast Mersenne Twister. The RNG used in this research was developed based on Mersenne Twister MT19937.

The main features of the Mersenne Twister RNG are [1] [3]:

- The period of the Mersenne Twister RNG is very long, $2^{19937}-1$;
- It is k-distributed to 32-bit accuracy for every $1 \leq k \leq 623$;
- Mersenne Twister RNG is fast. It avoids multiplication and division; and it benefits from caches and pipelines.

The Mersenne Twister RNG used in this research was written based on the codes published on [4]. Amendments are made to make functions from [4] suitable to be operated under VC2005 and to be invoked by other original functions.

Reference:

[1] M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator", ACM Transactions on Modeling and Computer Simulation, Vol. 8, pp. 3-30, 1998.

[2] website: http://www.math.sci.hiroshima-u.ac.jp/~m-mat/MT/emt.html

[3] website: http://en.wikipedia.org/wiki/Mersenne_twister

[4] website: http://www-personal.umich.edu/~wagnerr/MersenneTwister.h