Big data-driven optimization in transportation and communication networks

Longbiao Chen

Thèse de Doctorat en Cotutelle de
Sorbonne Université et Zhejiang Université

Spécialité : **INFORMATIQUE**

École Doctorale : Informatique, Télécommunication et Électronique de Paris

Présentée par

Mr Longbiao CHEN

Sujet de la thèse

Big Data-Driven Optimization in Transportation and Communication Networks

Soutenue le **5 Juillet 2018**

Jury :

Anelise Munaretto FONSECA	**Rapporteur**	**Professeur, Universidade Technológica Federal do Paraná – Brazil**
Marco FIORE	**Rapporteur**	**Chargé de recherche, HDR, CNR-IEIIT – Italy**
Guy PUJOLLE	**Examinateur**	**Professeur, Sorbonne Université – France**
Igor Monteiro MORAES	**Examinateur**	**Professeur, Universidade Federal Fluminense – Brazil**
Shijian LI	**Examinateur**	**Professeur, Zhejiang Université – China**
Thi-Mai-Trang NGUYEN	**Directrice de Thèse**	**Maître de conférences, HDR, Sorbonne Université – France**
Gang PAN	**Co-Directeur de Thèse**	**Professeur, Zhejiang Université – China**
Jérémie JAKUBOWICZ	**Co-Encadrant de Thèse**	**Maître de conférences, Télécom SudParis – France**

Declaration

I, Longbiao CHEN, hereby declare that this dissertation presents the results of my original research. I have not copied from any others' work or from any other sources except where due reference or acknowledgment is made explicitly in the text, nor has any part been written for me by another person.

Longbiao CHEN

July, 2018

Abstract

The evolution of metropolitan structures and the development of urban systems have created various kinds of *urban networks*, among which two types of networks are of great importance for our daily life, the *transportation networks* corresponding to human mobility in the *physical space*, and the *communication networks* supporting human interactions in the *digital space*. The rapid expansion in the scope and scale of these two networks raises a series of fundamental research questions on how to optimize these networks for their users. Some of the major objectives include demand responsiveness, anomaly awareness, cost effectiveness, energy efficiency, and service quality.

Despite the distinct design intentions and implementation technologies, both the transportation and communication networks share common fundamental *structures*, and exhibit similar spatio-temporal *dynamics*. Correspondingly, there exists an array of key challenges that are common in the optimization in both networks, including network profiling, mobility prediction, traffic clustering, and resource allocation. To achieve the optimization objectives and address the research challenges, various analytical models, optimization algorithms, and simulation systems have been proposed and extensively studied across multiple disciplines. Generally, these simulation-based models are not evaluated in real-world networks, which may lead to sub-optimal results in deployment.

With the emergence of ubiquitous sensing, communication and computing diagrams, a massive number of urban network data can be collected. Recent advances in big data analytics techniques have provided researchers great potentials to understand these data. Motivated by this trend, we aim to explore a new *big data-driven network optimization paradigm*, in which we address the above-mentioned research challenges by applying state-of-the-art data analytics methods to achieve network optimization goals. Following this research direction, in this dissertation, we propose two data-driven algorithms for *network traffic clustering* and *user mobility prediction*, and apply these algorithms to real-world optimization tasks in the transportation and communication networks.

First, by analyzing large-scale traffic datasets from both networks, we propose a *graph-based traffic clustering algorithm* to better understand the traffic similarities and variations across different area and time. Upon this basis, we apply the traffic clustering algorithm to the following two network optimization applications.

- **Dynamic traffic clustering for demand-responsive bikeshare networks.** In this application, we dynamically cluster bike stations with similar usage patterns to obtain stable and predictable cluster-wise bike traffic demands, so as to foresee over-demand stations in the

network and enable demand-responsive bike scheduling. Evaluation results using real-world data from New York City and Washington, D.C. show that our framework accurately foresees over-demand clusters (e.g. with 0.882 precision and 0.938 recall in NYC), and outperforms other baseline methods significantly.

- **Complementary traffic clustering for cost-effective C-RAN.** In this application, we cluster RRHs with complementary traffic patterns (e.g., an RRH in residential area and an RRH in business district) to reuse the total capacity of the BBUs, so as to reduce the overall deployment cost. We evaluate our framework with real-world network data collected from the city of Milan, Italy and the province of Trentino, Italy. Results show that our method effectively reduces the overall deployment cost to 48.4% and 51.7% of the traditional RAN architecture in the two datasets, respectively, and consistently outperforms other baseline methods.

Second, by analyzing large-scale user mobility datasets from both networks, we propose a *spatio-temporal mobility prediction algorithm* to better model the mobility patterns and fluctuations across different area and time. Based upon this, we apply the mobility prediction algorithm to the following two network optimization applications.

- **Spatio-temporal mobility prediction for anomaly-aware road networks.** In this application, we model the spatial correlations and temporal dependencies of of vehicle GPS trajectories in a unified spatio-temporal mobility model, and predict abnormal mobility events which may correspond to abnormal road conditions. Experiments with real-world data collected from Xiamen City show that our approach accurately predicts and identifies the road obstacles during the 2016 typhoon season with precision and recall both above 90%, and outperforms other baselines.
- **Deep mobility prediction for energy-efficient and quality-aware C-RAN.** In this application, we propose a deep-learning model to capture the spatio-temporal dynamics of user mobility in C-RAN, and accurately predict their movement patterns in next hour, so as to enable RRH cooperation to improve handover performance and increase BBU utilization. Real-world evaluations are conducted on two large-scale mobile network datasets collected from Ivory Coast and Senegal. Results show that our framework effectively increases the BBU utilization rate to more than 75.0% and achieve an RRH internal handover rate above 76.7%, which consistently outperforms the traditional RANs and other baseline methods.

Finally, we summarize the insights learned from the big data-driven network optimization paradigms, and discusses the future research directions from various perspectives, including potential optimization goals, other challenges and issues, new data analytics methods, and real-world deployment.

Key Words

transportation networks, communication networks, network optimization, big data analytics, graph-based traffic clustering, spatio-temporal mobility prediction, deep-learning, bikeshare networks, road networks, C-RAN, RRH-BBU mapping, urban computing, ubiquitous computing, cyber-physical systems

Résumé

Optimisation à base de l'analyse de données dans les réseaux de transport et des réseaux de communications

L'évolution des structures métropolitaines ont créé divers types de *réseaux urbains*. Parmi lesquels deux types de réseaux sont d'une grande importance pour notre vie quotidienne : les *réseaux de transport* correspondant à la mobilité humaine dans *l'espace physique* et les *réseaux de communications* soutenant les interactions humaines dans *l'espace numérique*. L'expansion rapide dans la portée et l'échelle de ces deux réseaux soulève des questions de recherche fondamentales sur la manière d'optimiser ces réseaux. Certains des objectifs principaux comprennent le provisioning de ressources à la demande, la détection des anomalies, l'efficacité énergétique et la qualité de service.

Malgré les différences dans la conception et les technologies de mise en œuvre, les réseaux de transport et les réseaux de communications partagent des *structures* fondamentales communes, et présentent des caractéristiques spatio-temporelles *dynamiques* similaires. En conséquence, ils existent les défis communs dans l'optimisation de ces deux réseaux : le profil du trafic, la prédiction de la mobilité, l'agrégation de trafic, le clustering des nœuds et l'allocation de ressources. Pour atteindre les objectifs d'optimisation et relever les défis de la recherche, différents modèles analytiques, algorithmes d'optimisation et systèmes de simulation ont été proposés et largement étudiés à travers plusieurs disciplines. Ces modèles analytiques sont souvent validés par la simulation et pourraient conduire à des résultats sous-optimaux dans le déploiement.

Avec l'émergence de l'Internet, un volume massif de données de réseau urbain peuvent être collecté. Les progrès récents dans les techniques d'analyse de données Big Data ont fourni aux chercheurs de grands potentiels pour comprendre ces données. Motivé par cette tendance, l'objectif de cette thèse est d'explorer un nouveau *paradigme d'optimisation des réseaux basé sur les données*. Nous abordons les défis scientifiques mentionnés ci-dessus en appliquant des méthodes d'analyse de données pour l'optimisation des réseaux. Nous proposons deux algorithmes data-driven pour *le clustering de trafic réseau* et *la prédiction de la mobilité d'utilisateur*, et appliquer ces algorithmes à l'optimisation dans les réseaux de transport et de communications.

Premièrement, en analysant les jeux de données de trafic à grande échelle des deux réseaux, nous proposons un *algorithme de clustering à base de graphe* pour mieux comprendre les similitudes de la circulation et les variations de trafic entre différents zones et heures. Sur cette base, nous

appliquons l'algorithme d'agrégation (clustering) de trafic aux deux applications d'optimisation de réseau suivants :

- **Un clustering de trafic dynamique pour la planification à la demande des réseaux de vélos partagés.** Dans cette application, nous regroupons dynamiquement les stations de vélos avec des motifs de trafic similaires pour obtenir des demandes de trafic groupées (en cluster) plus stables et plus prédictible, de manière à pouvoir prévoir les stations surchargés dans le réseau et à permettre une planification dynamique de réseau en fonction de la demande. Les résultats d'évaluation en utilisant les données réelles de New York City et Washington, D.C. montrent que notre solution prévoit précisément des clusters surchargés (avec une précision de 0,882 et un recall de 0,938 à New York) et elles sont bien meilleures que les méthodes de référence.
- **Un clustering de trafic complémentaire pour les réseaux C-RAN (Cloud Radio Access Network).** Dans cette application, nous regroupons les RRHs (Remote Radio Head) avec des modèles de trafic complémentaires (par exemple, des RRHs dans un quartier résidentiel et des RRHs dans un quartier business) pour réutiliser la capacité totale des BBUs (Base Band Unit), afin de réduire le coût de déploiement global. Nous évaluons notre solution avec des données de réseau réelles collectées dans la ville de Milan, en Italie et dans la province du Trentinno, en Italie. Les résultats montrent que notre méthode est efficace en réduisant le coût de déploiement global entre 48,4% et 51,7% pour les deux jeux de données respectivement, par rapport à l'architecture RAN (Radio Access Network) traditionnelle, et fonctionne mieux que les autres méthodes de base.

Deuxièmement, en analysant les jeux de données sur la mobilité des utilisateurs à grande échelle des deux types de réseaux, nous proposons un *algorithme de prédiction de mobilité spatio-temporelle* pour mieux modéliser la mobilité et les fluctuations dans de différentes zones et temps. Sur cette base, nous appliquons l'algorithme de prédiction de mobilité à deux applications d'optimisation de réseau suivantes :

- **Prédiction de mobilité spatio-temporelle pour les réseaux routiers en prenant en compte des anomalies.** Dans cette application, nous modélisons les corrélations spatiales et les dépendances temporelles des trajectoires GPS des véhicules dans un modèle de mobilité spatio-temporelle unifiée, et prédire des événements de mobilité anormaux ce qui peut correspondre à des conditions de route anormales. Expériences avec des données réelles collectées de la ville de Xiamen montrent que notre approche prédit et identifie avec précision les obstacles de la route pendant la saison des tempêtes 2016, avec une précision et un recall plus de 90%, qui sont meilleurs par rapport aux méthodes de base.
- **Prédiction profonde de mobilité pour l'économie d'énergie et la qualité de service dans les réseaux C-RAN.** Dans cette application, nous proposons un modèle d'apprentissage en profondeur pour capturer la dynamique spatio-temporelle de la mobilité d'utilisateur dans les réseaux cellulaires, et de prédire avec précision leurs mobilité dans l'heure qui suit, afin de permettre la coopération entre les RRHs et d'améliorer les performances de handover ainsi que d'augmenter le taux d'utilisation des BBUs. Les évaluations sont réalisées sur deux jeux de données de réseaux mobiles à grande échelle de la Côte d'Ivoire et du Sénégal. Les résultats montrent que notre solution augmente efficacement le taux d'utilisation des BBU à

plus de 75,0% et atteint un taux de handover entre les RRHs du même BBU (intra-BBU) plus de 76,7%, ce qui sont meilleurs que les réseaux RAN traditionnels et les autres méthodes de base.

Enfin, nous résumons les grandes lignes dans l'optimisation de réseau à base de l'analyse de données, et discutons les futures directions de recherche de divers points de vue, y compris les objectifs d'optimisation potentiels, les défis et questions ouvertes, les nouvelles méthodes d'analyse des données et le déploiement dans le monde réel.

Mots Clés

réseaux de transport, réseaux de communications, optimisation des réseau, analyse de données Big Data, clustering à base de graphe, prédiction de mobilité spatio-temporelle, l'apprentissage en profondeur, réseaux de vélos partagés, réseaux routiers, C-RAN, RRH-BBU mapping, informatique urbaine, informatique ubiquitaire, systèmes cyber-physiques

Acknowledgment

First and foremost I would like to express my deepest gratitude to my supervisor Prof. Thi-Mai-Trang Nguyen, who has given me the opportunity to conduct my PhD dissertation in Paris during the past years. She encouraged and inspired my research, and shared with me valuable experiences in academia. Without her guidance, I would not be able to overcome the difficulties during my PhD study and to complete this dissertation. The talks and discussions with her always bring me newer and deeper understanding of academic research.

I also would like to express my deep gratitude to Prof. Gang Pan, my co-supervisor at Zhejiang University, China. Without his encouragement, I would not choose to study in Paris. He gave me comprehensive academic training and guidance during my study in Zhejiang University, which laid a solid foundation for me to continue the research work in Paris.

A very special thanks goes out to my advisor Prof. Daqing Zhang, for his excellent guidance, assistance, patience and motivation in leading me towards academic research. Without his help, I would not be able to achieve what I have done today. A very special thanks also goes to my advisor Prof. Jérémie Jakubowicz, for his great guidance and assistance to my PhD research. He not only encouraged me to pursue and experience the research life in Paris, but also took steps forward in France-China academic exchange and cooperation.

I also would like to thank all the jury members for reading my thesis, for offering their valuable time and for their constructive feedbacks.

I must acknowledge as well my friends and colleagues in the PHARE team, who have been so supportive and helpful along the way of my PhD study. A particular thanks to Xin Wang, Shuai Yu, and Yifu Chen for their great help, discussion and collaboration. I also would like to thank all the team members for all the great times that we have shared.

Longbiao CHEN

July, 2018

Paris, France

Publications

Under Review

1. **Longbiao Chen**, Dingqi Yang, Daqing Zhang, Jérémie Jakubowicz, Thi-Mai-Trang Nguyen*, Gang Pan. *Cloud-RAN Optimization Exploiting the Spatial-Temporal Dynamics of Mobile Users*. Submitted to IEEE Transactions on Mobile Computing (**T-MC**).
2. **Longbiao Chen**, Dingqi Yang, Daqing Zhang, Cheng Wang, Jonathan Li, Mai-Trang Nguyen*. *Deep Mobile Traffic Forecast and Complementary Base Station Clustering for Cloud-RAN Optimization*. Submitted to Journal of Network and Computer Applications (**JNCA**).

Conference Papers

1. **Longbiao Chen**, Xiaoliang Fan, Leye Wang, Zhiyong Yu, Johnthan Li, Daqing Zhang, Thi-Mai-Trang Nguyen, Gang Pan, Cheng Wang. *RADAR: Road Obstacle Identification for Disaster Response Leveraging Multi-Source Urban Data*. The ACM International Joint Conference on Pervasive and Ubiquitous Computing (**UbiComp'18**), Singapore, 2018. [Accepted]
2. **Longbiao Chen**, Linjin Liu, Xiaoliang Fan, Johnthan Li, Cheng Wang, Gang Pan, Jérémie Jakubowicz, Thi-Mai-Trang Nguyen. *Complementary Base Station Clustering for Cost-Effective and Energy-Efficient Cloud-RAN*. The 14th IEEE International Conference on Ubiquitous Intelligence and Computing (**UIC'17**), San Francisco, USA, 2017.
3. **Longbiao Chen**, Daqing Zhang, Leye Wang, Dingqi Yang, Xiaojuan Ma, Shijian Li, Zhaohui Wu, Gang Pan, Thi-Mai-Trang Nguyen, Jérémie Jakubowicz. *Dynamic Cluster-Based Over-Demand Prediction in Bike Sharing Systems*. The ACM International Joint Conference on Pervasive and Ubiquitous Computing (**UbiComp'16**), Heidelberg, Germany, 2016, pp. 841-852. *[Honorable Mention Award]*
4. **Longbiao Chen**, Leye Wang, Daqing Zhang, Shijian Li, Gang Pan. *EnUp: Energy-Efficient Data Uploading for Mobile Crowd Sensing Applications*. The 13th IEEE International Conference on Ubiquitous Intelligence and Computing (**UIC'16**), Toulouse, France, 2016, pp. 1074-1078.
5. **Longbiao Chen**, Daqing Zhang, Gang Pan, Xiaojuan Ma, Dingqi Yang, Wangsheng Zhang, Kostadin Kushlev, Shijian Li. *Bike Sharing Station Placement Leveraging Heterogeneous*

Urban Open Data. The ACM International Joint Conference on Pervasive and Ubiquitous Computing (**UbiComp'15**), Osaka, Japan, 2015, pp. 571-575. *[Honorable Mention Award]*

6. **Longbiao Chen**, Dingqi Yang, Jérémie Jakubowicz, Gang Pan, Daqing Zhang, Shijian Li. *Sensing the Pulse of Urban Activity Centers Leveraging Bike Sharing Open Data*. The 12th IEEE International Conference on Ubiquitous Intelligence and Computing (**UIC'15**), Beijing, China, 2015, pp. 135-142.
7. **Longbiao Chen**, Jérémie Jakubowicz. *Inferring Bike Trip Patterns from Bike Sharing System Open Data*. 2015 IEEE International Conference on Big Data (**BigData'15 Poster**), Santa Clara, USA, 2015.

Journal Articles

1. **Longbiao Chen**, Jérémie Jakubowicz, Dingqi Yang, Daqing Zhang, Gang Pan*. *Fine-Grained Urban Event Detection and Characterization Based on Tensor Cofactorization*. IEEE Transactions on Human-Machine Systems (**T-HMS**), 47(3): 380–391, 2017.
2. **Longbiao Chen**, Xiaojuan Ma, Thi-Mai-Trang Nguyen, Gang Pan, Jérémie Jakubowicz*. *Understanding Bike Trip Patterns Leveraging Bike Sharing System Open Data*. Frontiers of Computer Science (**FCS**), 11(1): 38–48, 2017.
3. **Longbiao Chen**, Daqing Zhang*, Xiaojuan Ma, Leye Wang, Shijian Li, Zhaohui Wu, Gang Pan. *Container Port Performance Measurement and Comparison Leveraging Ship GPS Traces and Maritime Open Data*. IEEE Transactions on Intelligent Transportation Systems (**T-ITS**), 17(5): 1227–1242, 2016.
4. Dingqi Yang, Daqing Zhang, **Longbiao Chen**, Bingqing Qu. *NationTelescope: Monitoring and Visualizing Large-Scale Collective Behavior in LBSNs*. Journal of Network and Computer Applications (**JNCA**), 55: 170–180, 2015.

Table of contents

Chapter 1

Introduction

Contents

1.1 Background

The rapid progress of urbanization has modernized many people's lives. Today, 54% of the world's population lives in urban areas [1]. The evolution of metropolitan structures and the development of urban systems have created various kinds of *urban networks* [2], such as transportation networks, communication networks, electric networks, financial and economic networks. The rapid expansion in the scope and scale of these urban networks raises a series of fundamental research questions, e.g., how to reduce network deployment cost, how to decrease network energy consumption, and how to improve network quality of services. Research on *urban network optimization* now extends across many disciplines and over many scales, ranging from geography, sociology, engineering to physics [2]. However, it is still challenging to develop unified and effective mechanisms to answer the above-mentioned questions.

In this dissertation, we focus our research on two of the most important urban networks corresponding to human mobility in the *physical space* and human communication in the *digital space*, i.e., the urban transportation networks and the urban communication networks. We start by elaborating on the basic concepts and optimization goals of these two categories of networks.

Urban Transportation Networks: in the physical urban space, transportation networks serve as the key infrastructure in modern cities, facilitating the movement of people in urban space and allowing for social interactions and activities [3]. In urban planning and management, a typical transportation network usually consists of two components, the *transit networks* and the *road networks* [3]. Transit systems provide various means to meet human mobility demands, including metro, bus, cars, bikeshare [4], and rideshare [5]. Road systems forms the most basic level of urban transportation infrastructure by connecting different geographical areas and social communities to accommodate vehicles and pedestrian traffic [6].

With the rapid growth of urban scale and population, urban transportation networks face significant challenges today, especially in *demand responsiveness* and *anomaly awareness*. On one hand, due to transportation resource limitation, urban transit networks often suffer from overcrowding at peak times, leading to serious traffic congestion and air pollution [7]. According to INRIX [8], it is estimated that traffic congestion costs $124 billion a year in the United States. The incapability of fulfilling the travel demands in a responsive manner greatly hinders urban transportation efficiency and satisfactory [9]. On the other hand, the availability of road networks is not consistent but rather depends on various social and environmental factors. For example, large-scale social events may cause abnormal traffic congestions around the arenas [10], and hurricane-caused water logging may paralyze large numbers of road segments [11]. Failing to monitor the real-time road anomalies have caused serious problems in urban management, such as stampede and chain accidents [12]. Therefore, it is essential to build *demand-responsive* urban transit networks and to monitor the *anomaly-aware* of road networks for sustainable urban development.

The evolution of information and communication technologies (ICT) in recent decades have provided new potentials to realize the above-mentioned vision [13]. With the wide deployment of physical sensors, wireless communication, Internet-of-Things, and cloud computing technologies, researchers have proposed various systems, methodologies, and theories to develop and improve urban transportation systems, which is now recognized as Intelligent Transportation Systems (ITS) [14]. Some of the typical ITS technologies and applications include camera-based traffic surveillance [15], traffic light coordination [15], GPS-equipped vehicle tracking [16], smart card-based bikeshare [4], and Internet-based rideshare [5]. The benefits of intelligent transportation systems can be elaborated from the following two aspects.

- *Demand Responsiveness*: traditional public transit networks provide fixed routes on predefined schedules with limited adaption to user mobility dynamics. With bikeshare and rideshare networks, the dynamics of user travel demands can be accommodated from both spatial and temporal dimensions. Bikeshare networks are densely deployed in the urban areas, making it easy to pick up and return bikes in almost any time and any bike stations for short and long trips [4]. Rideshare networks enable drivers and riders to pair their travel demands with each other in an online manner, making it possible to dynamically schedule transportation resources to accommodate demands with flexible travel time and destination [5].

- *Anomaly Awareness*: the traffic situation and road connectivity can be monitored efficiently via the widely-available sensors in road infrastructures and GPS equipped vehicles. Road infrastructures, including traffic surveillance cameras, ground inductive coils, speed sensor radar, are capable of monitoring road traffic flows and generate congestion and accident reports in real-time. GPS equipped vehicles, on the other hand, can be regarded as mobility sensors in the road networks to provide fine-grained mobility patterns and road anomalies. By collecting and analyzing information from these ubiquitous sensors, we can be aware of road networks conditions, detect anomalies, and provide information services for users in a real-time and predictive manner.

Urban Communication Networks: in the digital urban space, communication networks are the fundamental layer for social and community interactions [17]. Among various kinds of communication networks, cellular networks are becoming increasingly important. In recent decades, the number of cellular network subscriptions is continuously raising at almost 6 percent year-on-year, reaching 7.8 billion at the end of 2017 [18]. Correspondingly, the cellular traffic volume has grown 18-fold over the past five years and will continue to increase exponentially in the upcoming 5G era, as smartphones and Internet-of-Things (IoT) devices become increasingly popular [19].

As cellular network scale and capacity grow, the *deployment cost* and *energy consumption* are becoming increasingly high [20, 21], while the *quality-of-service* of the entire network is becoming increasingly difficult to ensure [22]. First, network operators need to cover the expenses for network construction, operation, maintenance and upgrade [23]. According to Juniper Research Report [21], the deployment cost of the network infrastructures may exceed operator's revenue as network scale grows over the next few years. Second, the energy consumption of cellular network infrastructures are substantially increasing, taking up more than 3% of the worldwide electric energy consumption nowadays [24]. Third, as various sizes of base stations (e.g., pico-cells, micro-cells, and macro-cells) and different generations of technologies (e.g., LTE, UMTS, and GSM) co-exist in real-world cellular networks [25], it is becoming extremely difficult for network operators to provide unified and consistent experiences for users in a *heterogeneous* network environment, such as enabling seamless handover between base stations. Therefore, designing *cost-effective, energy efficient, and high-quality* cellular network architectures is now a great necessity in the field of mobile network operation and research [21].

Fortunately, the emergence of *Cloud Radio Access Network (C-RAN)* [26] has provided new opportunities to address the above mentioned challenges. In C-RAN, a traditional base station is split into two components: the *Remote Radio Head (RRH)* for radio communication with mobile devices, and the *Baseband Unit (BBU)* for signal and data processing [23]. The BBUs are further detached from the RRHs and hosted in centralized *BBU pools* [27]. The RRHs and BBU pools are usually connected via high speed optical fiber [28] to reduce the delay of data transmission. In this way, the baseband resources are pooled, so that they can be shared among base stations, making it possible to achieve *statistical multiplexing gain* for the whole network [23]. We details the benefits of the C-RAN over the traditional RAN from the following three perspectives:

- *Cost Effectiveness*: in traditional RAN architectures, the capacity of the BBU needs to cover the traffic volume in each base station. As the traffic demand is usually *non-uniform* and *highly dynamic*, the deployed BBU capacity are wasted during off-peak hours. With the C-RAN architecture, the RRHs with different traffic patterns can complement each other to generate stable aggregate traffic. Consequently, the utilization rate of BBUs are increased, and fewer BBUs are needed in C-RAN architecture compared with traditional RAN [23]. Moreover, since BBUs are hosted in centralized data centers, the cost to operate, repair, and upgrade these devices are much lower than the distributed on-site solution.
- *Energy Efficiency*: in C-RAN architecture, multiple RRHs can be connected to one BBU and share the processing capacity. Therefore, the number of needed BBUs can be dynamically determined, and the unused BBUs can then be dynamically shutdown to save energy. Consequently, the energy consumption is reduced compared with the traditional RAN. Moreover, by adopting advanced data centers power management technologies, the power consumption in BBU pools can also be significant reduced [29].
- *Service Quality*: when two RRHs are connected to one BBU, the handover and roaming events between them can be handled directly inside the BBU, which greatly reduces the handover delay and roaming overhead [23, 30]. Such a seamless handover and roaming experience is of key importance in 5G networks to support direct video streaming and real-time IoT applications. Consequently, the quality-of-service of the network can be improved.

1.2 Research Challenges

In order to achieve the above-mentioned optimization goals in transportation and communication networks, a number of challenges need to be addressed to optimize the architecture and configuration of these networks [31]. Despite the distinct design intentions and implementation technologies, both the transportation and communication networks share common fundamental *structures*, and exhibit similar spatio-temporal *dynamics* [32, 33]. Specifically, the spatial correlations among network nodes and the temporal dependencies of network traffic and user mobility make the networks high dynamic and complicated to model and optimize. We elaborate on some of the key challenges that are common to both networks, which have been studied in the literature of network optimization, including network profiling, mobility prediction, traffic clustering, and resource allocation.

- *Network Profiling*. Although the overall users and traffic volumes in the transportation and communication networks are growing, the demand patterns in different areas and during different periods of time are not evenly distributed. For example, during weekday working hours, business districts and transit hubs usually observe high traffic volume and massive user movement, while residential areas usually observe fewer users and lower traffic volumes. Such a *spatial-temporal non-uniform property* of network demands poses great challenges for the design and deployment of both networks. Optimally, the cognitive networks [34] shall be able to perceive the non-uniformed traffic demand and to react accordingly. In order to realize this vision, algorithms and models are needed to profile the variations in traffic demand in an automatic manner.
- *Mobility Prediction*. In both networks, users move freely and spontaneously among the road networks and base stations, generating mobility patterns that are highly dynamic in both spatial and temporal dimensions. Being able to predict users' mobility in next few hours and even in next few days are of great importance for a wide range of network optimization applications, including vehicle scheduling and balancing in bikeshare networks [35], dynamic navigation and routing in rideshare networks [36], distributed content caching in C-RAN [37], and seamless handover handling via base station cooperation in C-RAN [23]. How to develop an accurate user mobility modeling and prediction method for network optimization applications still remains a challenge.
- *Traffic Clustering*. Clustering geographic areas or base stations into larger, semantic regions and groups has been proposed in the literature for many purposes, such as reducing traffic and mobility fluctuations [38], understanding region functions [39], enabling base station cooperations, and improving handover handling [23]. For example, in bikeshare networks, clustering bike stations with similar usage patterns can help understand regional functions and predict future usages; in C-RAN, by clustering RRHs with complementary traffic patterns, the utilization rate of BBUs can be increased. Considering the dynamic nature of traffic and mobility, how to design effective clustering mechanisms with the above-mentioned objectives is a key challenge for network optimization.
- *Resource Allocation*. In the design and implementation of the above-mentioned network optimization goals, network resource constraints shall be taken into consideration to align with the real-world situations. For example, in bikeshare networks, the constraints generally include the number of available bikes and docks, and the geographic distances of bike stations. In C-RAN, the constraints may include the RRH coverage area, the available BBUs in the pools, the network bandwidth, etc. How to optimally allocate these resources so as to maximize resource utilization or minimize total budgets is still an open challenge in network optimization research.

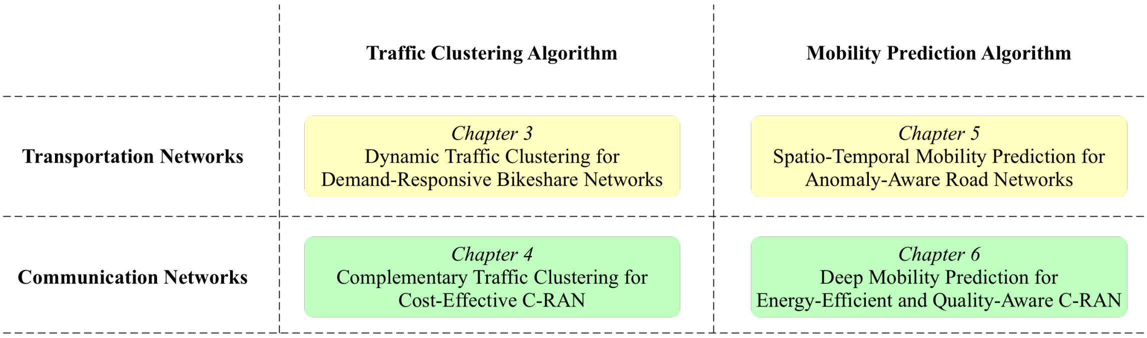
To address the above-mentioned challenges, various analytical models, optimization algorithms, and simulation systems have been proposed and extensively studied in network optimization research. However, most of these works are built upon *simulation environments* without real-world evaluations. For example, network traffic demands are usually generated via Poisson variables [40], while user mobility patterns are generally simulated as random walk processes [28]. Network nodes (e.g., RRHs) are ideally assumed to be distributed in rectangle or hexagonal cells [41], while the available resources (e.g. bikes or bandwidth) are ideally assumed values determined by simulation models [35]. Such a simulation-based mechanism introduces a large number of parameters in the models and algorithms, which need to be carefully selected and fine-tuned when deploying these system to real-world networks. Moreover, since the real-world network traffic and user mobility patterns are highly dynamic, the assumed generation models might not be able to characterize these dynamically accurately. Consequently, the simulation-based models and algorithms may lead to sub-optimal results in real-world deployment.

1.3 Thesis Contributions

With the emergence of ubiquitous sensing, communication and computing diagrams [42, 43], a massive number of urban data have been collected from various kinds of urban networks, providing researchers with great opportunities to understand the dynamics of urban networks [44, 45]. The knowledge discovered from these *urban big data* can be used to design effective mechanisms for network optimization. In this dissertation, aiming at achieving the above-mentioned optimization goals in transportation and communication networks, we explore a *big data-driven network optimization* paradigm to address the research challenges mentioned in the previous section. Specifically, we propose two data-driven algorithms for network traffic clustering and user mobility prediction, and apply these algorithms to real-world optimization tasks transportation and communication networks. Figure 1.1 illustrates the scheme of the dissertation contributions. The detailed contributions are described as follows.

First, by analyzing large-scale traffic datasets from both networks, we propose a *graph-based traffic clustering algorithm* to better understand the traffic similarities and variations across different area and time. Upon this basis, we apply the traffic clustering algorithm to the following two network optimization applications.

- **Dynamic traffic clustering for demand-responsive bikeshare networks.** In this application, we dynamically cluster bike stations with similar usage patterns to obtain stable and predictable bike traffic demands, so as to foresee over-demand stations in the network and enable demand-responsive bike scheduling. Since the bike usage pattern of a station is highly dynamic and context dependent, directly predicting individual over-demand stations to carry



	Traffic Clustering Algorithm	Mobility Prediction Algorithm
Transportation Networks	*Chapter 3* Dynamic Traffic Clustering for Demand-Responsive Bikeshare Networks	*Chapter 5* Spatio-Temporal Mobility Prediction for Anomaly-Aware Road Networks
Communication Networks	*Chapter 4* Complementary Traffic Clustering for Cost-Effective C-RAN	*Chapter 6* Deep Mobility Prediction for Energy-Efficient and Quality-Aware C-RAN

Figure 1.1: The scheme of dissertation contributions.

out preventive measures is difficult. In addition, bike usage pattern is affected not only by common contextual factors (e.g., time and weather) but also by opportunistic contextual factors (e.g., social and traffic events). We propose a dynamic cluster-based framework for over-demand prediction. Depending on the context, we construct a weighted correlation network to model the relationship among bike stations, and dynamically group neighboring stations with similar bike usage patterns into clusters. We then adopt Monte Carlo simulation to predict the over-demand probability of each cluster. Evaluation results using real-world data from New York City and Washington, D.C. show that our framework accurately foresees over-demand clusters (e.g. with 0.882 precision and 0.938 recall in NYC), and outperforms other baseline methods significantly.

- **Complementary traffic clustering for cost-effective C-RAN.** In this application, we cluster RRHs with complementary traffic patterns (e.g., an RRH in residential area and an RRH in business district) to reuse the total capacity of the BBUs, so as to reduce the overall deployment cost. Specifically, we first employ the deep learning-based MuLSTM algorithm to predict the hourly traffic profile for the next day, and then exploit the RCLP algorithm to cluster RRHs with different peak hours (complementary) and within a distance to generate stable aggregated traffic patterns. Finally, we allocate BBUs to satisfy the traffic demands of the clusters under the pool resource constraints. We evaluate our framework with real-world network data collected from the city of Milan, Italy and the province of Trentino, Italy. Results show that our method effectively reduces the overall deployment cost to 48.4% and 51.7% of the traditional RAN architecture in the two datasets, respectively, which consistently outperforms other baseline methods.

Second, by analyzing large-scale user mobility datasets from both networks, we propose a *spatio-temporal mobility prediction algorithm* to better model the mobility patterns and fluctuations across different area and time. Upon this basis, we apply the mobility prediction algorithm to the following two network optimization applications.

- **Spatio-temporal mobility prediction for anomaly-aware road networks.** In this application, we model the spatial correlations and temporal dependencies of of vehicle GPS trajectories in a unified spatio-temporal mobility model, and predict abnormal mobility events which may correspond to abnormal road conditions. We apply this prediction techniques to post-disaster road obstacle identification. One of the first priority in disaster response is to identify and clear road obstacles, such as fallen trees and ponding water. We propose a low-cost and real-time approach to identify road obstacles leveraging large-scale vehicle mobility data and heterogeneous road environment sensing data. First, based on the observation that road obstacles may cause abnormal slow mobility behaviors of vehicles in the surrounding road segments, we detect road obstacles by identifying the collective anomalies of slow mobility behaviors from vehicle trajectory data. Then, we classify the detected road obstacles leveraging the correlated spatial and temporal features extracted from various road environment data, including satellite images and meteorological records. Experiments with real-world data collected from Xiamen City show that our approach accurately detects and classifies the road obstacles during the 2016 typhoon season with precision and recall both above 90%, and outperforms other baselines.
- **Deep mobility prediction for energy-efficient and quality-aware C-RAN.** In this application, we propose a deep-learning model to capture the spatio-temporal dynamics of user mobility in C-RAN, and accurately predict their movement patterns in next hour, so as to enable RRH cooperation to improve handover performance and increase BBU utilization. To achieve these goals, we first employ the deep learning-based algorithm to model the spatial-temporal dynamics of the traffic pattern and user mobility, and accurately predict the traffic volumes and handover events in the network for the next hour. Then, we exploit a resource-constrained clustering algorithm to find a set of cooperative RRH clusters, so that the traffic volume and handover events are properly handled within the corresponding BBUs. Finally, we make the optimal RRH-BBU mapping schemes under the BBU pool resource constraints. Real-world evaluations are conducted on two large-scale mobile network datasets collected from Ivory Coast and Senegal. Results show that our framework effectively increases the BBU utilization rate to more than 75.0% and achieve an RRH internal handover rate above 76.7%, which consistently outperforms the traditional RANs and other baseline methods.

1.4 Thesis Outline

This dissertation is organized as follows. In Chapter 2, we first review the existing related work in network big data analytics and network optimization. The following chapters describe our main contributions. In Chapter 3, we present the application of dynamic traffic clustering for demand-responsive bikeshare networks. In Chapter 4, we describe the application of complementary traffic clustering for energy-efficient C-RAN. In Chapter 5 and Chapter 6, we present two applications of the mobility prediction algorithms in transportation networks and communication networks, i.e., spatio-temporal mobility prediction for anomaly-aware road networks, and deep mobility prediction for cost-effective and quality-aware C-RAN. Finally, Chapter 7 summarizes the insights learned from the big data-driven network optimization paradigms, and discusses the future research directions from various perspectives, including potential optimization goals, other challenges and issues, new data analytics methods, and real-world deployment.

Chapter 2

Literature Review

Contents

In this chapter, we review the existing research works in mobile network optimization. First, we present a brief survey on the optimization of urban transportation and communication networks, covering various optimization objectives and challenges. Next, we discuss the relevant works in big data analytics and state-of-the-art methodologies.

2.1 Urban Transportation Network Optimization

Urban transportation systems provide various means to meet human mobility demands, including metro, bus, cars, bikeshare [4], and rideshare [5]. Based on the applications of our work, we present a brief survey on existing works in bikeshare networks and road networks.

2.1.1 On Bikeshare Networks

Recently, bike sharing systems have been intensively studied from different perspectives, including bike sharing history [4], infrastructure [46], worldwide deployment [47,47], and bike usage patterns [48–51]. The research interests mainly focus on the following problems: (1) *system planning*, such as determining the number, capacity and locations of stations [52, 53]. (2) *system balancing*, such as strategies to transport bikes among stations [35], and mechanisms to encourage users to rent bikes from (or return bikes to) specific stations through incentives [54, 55]. (3) *system prediction*, such as predicting station status and bike usage number using different models.

The earlier work mainly focuses on predicting the number of available bikes and docks (i.e. station status) in the station level. For example, Froehlich et al. [56] adopted a Bayesian network to predict station status based on the current time and current bike/dock number. Kaltenbrunner et al. [51] proposed to model and predict the station status as a time series using an ARIMA model. However, due to the impact between neighboring stations [57] and the complicated contextual factors impacting bike usage (e.g., weather, temperature, social events) [52, 58–60], these station-level prediction methods do not consistently achieve accurate results.

To address this issue, researchers have proposed to cluster similar stations into clusters, and then predict bike usage on a cluster-level. For example, Li et al. [61] first proposed a method to cluster stations based on their geographical locations and transition patterns, then predicted the bike usage of the whole system, and finally allocated the overall bike rental and return number to each cluster based on a proportion learned from historical data. However, the cluster scheme is static across different contexts. Etienne et al. [50] introduced a model-based method to group stations with similar bike usage patterns, such as stations near restaurants and train stations, and predicted their bike usage pattern in different temporal settings. These cluster-level prediction methods could improve the prediction accuracy, however the clusters used in these methods are static regardless of context at the time. Since the bike usage patterns of stations might be affected by various contextual factors such as weather condition and social events [58–60], the prediction results of static clusters may not yield consistent accuracy across different contexts.

2.1.2 On Road Networks

The other important component of urban transportation systems are road networks. Research on road traffic monitoring and transportation anomaly detection of have been conducted in the literature, especially in the disaster response perspective. Restoring the transportation network is usually considered the first step [62, 63], since road network disruptions impede timely rescue, evacuation, and supply [62]. In order to evaluate the post-disaster transportation system performance, Chang et al. [64] develop various transportation quality and accessibility indicators in a quantitative approach. To improve the post-disaster transportation accessibility, Aksu et al. [62] propose a dynamic-path-based mathematical model to clear critical road obstacles in an optimal order with limited resources. However, few works in the transportation research area have addressed the problem of identifying road obstacles in disaster response scenarios.

The road obstacle detection problems have been studied by the computer vision community using several vision-based approaches [65–67]. For example, LeCun et al. [66] proposed a convolution neural network-based approach to detect road obstacles from videos captured by vehicle's on-board camera, and Lefaix et al. [67] used a vehicle-mounted camera to detect and track road obstacles by analyzing the image motion patterns. However, deploying these vision-based solutions to a large crowd of vehicles requires extremely high cost and thus are infeasible for large-scale road obstacle detection, especially in disaster response scenarios. Another approach for road obstacle detection can be inspired by the existing traffic accident reporting systems, where citizens voluntarily report to the traffic police and media outlets about observed road obstacles using phone calls, social networks, etc. However, the processing of these reports include cross-validating the sources, labeling the locations and scopes on the map, and broadcasting the message back to drivers. Such a process can still be labor intensive and time consuming.

2.2 Urban Communication Network Optimization

The fast evolution of cellular networks have shown its great importance in modern urban communication systems [68, 69]. Cellular network operators and researchers are continuously seeking for solutions to optimization to provide stable telecommunication, high speed data rate, and high quality of services to their users [26, 70]. However, the cost to build, operate and upgrade the network infrastructures is becoming increasingly expensive for mobile operators [71]. As the deployment and commercial operation of 4G systems are reaching maturity, researchers and network operators worldwide have begun searching for next generation (5G) mobile network solutions [71].

Cloud radio access network (C-RAN) is targeted by worldwide mobile network operators as a typical realization of green and soft RAN architecture in the 5G mobile networks [23]. In 2010, IBM proposed wireless network cloud (WNC) [72]. The WNC system exploits emerging cloud-computing technology and various wireless infrastructure technologies, such as remote radio head and software radio, to enable RAN resource processing operating in a cloud mode [72]. In 2011, China Mobile Research Institute envisioned a cloud-based RAN architecture to provide mobile broadband Internet access to wireless customers with low bit-cost, high spectral and energy efficiency [26]. Texas Instruments also proposed an enhanced version of its KeyStone multicore architecture to be used to create cloud base stations. For a comprehensive technology survey on C-RAN, the reader is referred to [23].

2.2.1 On C-RAN Traffic Demands

One of the key vision in C-RAN is to provide flexible and configurable data processing capacity according to the mobile traffic demands [26, 73]. In [74], such a vision is coined as a *cognitive networking diagram* in the context of mobile network. Specifically, C-RAN algorithms and solutions need to adapt to the variations of mobile demand. This raises a fundamental problem of understanding the mobile traffic demand patterns. Traditionally, researchers have proposed various kinds of models to simulate mobile traffic demands of RRHs. For example, Poisson processes are usually employed to simulate the number of phone calls in RRHs [28, 75]. However, These models usually require specific parameters for each RRH, and may not be able to accurate characterize the patterns in real-world networks. Furno et al. took a first step in the direction of RRH traffic profiling leveraging a data-driven approach [74]. By leveraging mobile traffic analytics algorithms for large-scale real-world mobile network datasets, they were able to understand the network demand patterns and the resource utilization in an automated manner. In their later work, Furno et al. further presented an approach to infer the spatial-temporal structures hidden in the traffic demand [76].

Cooperation between RRHs is necessary to cope with the vibrations in traffic demands, and to achieve statistical multiplexing gains in C-RAN [23]. An optimal RRH cooperation scheme should facilitate the BBU pool capacity utilization and reduce the BBU deployment cost [23, 40]. To this end, a cooperative cluster usually consists of RRHs from different areas, such as office, residential, and commercial districts [23]. Other factors need to be considered in the RRH clustering algorithm include cluster size, geographic constraints, etc. In the literature, several clustering schemes and algorithms have been proposed with various goals. For example, Bhaumik et al. [77] proposed CloudIQ, a framework for partitioning a set of RRHs into groups and process the signals in a shared data center. By exploiting the variations in the processing load across RRHs, the framework was able to save up to 19% of the computing resources for a probability of failure of one in 100 million. In [78], Namba et al. proposed an C-RAN architecture that can dynamically change the clustering schemes of RRHs in response to traffic demand. They further presented and evaluated adaptive BBU-RRH mapping schemes for C-RAN in [79], where it was proved that the number of BBUs can be reduced by 47% for the proposed adaptive schemes compared with the static assignment.

2.2.2 On C-RAN Mobility Dynamics

Another key issue in the design and implement of C-RAN is the factor of user mobility [80, 81]. One of the important objectives in next generation mobile networks (5G) is improving the quality of cellular service, with handover events nearly invisible to the mobile users [80]. To this end, the RRHs in a network need to be able to cooperate with each other to seamlessly transfer user contexts, forward network resources, and assign cellular channels [30]. This raises an important problem of foreseeing the mobile user mobility dynamics in next few hours. Traditionally, user mobility is usually ideally modeled with specific assumptions, e.g., random walk variables with specific moving speed and diameters [30]. These assumptions ignore the spatio-temporal variations and dynamics of mobile user mobility, which might be inaccurate to foresee the user movement in a future period of time.

There are still few work on data-driven user mobility modeling and prediction for C-RAN optimization. One of the most relevant work on investigating mobile user dynamics is presented by Naboulsi et al. [82]. The authors introduced a methodology to manage user mobility in C-RAN by dynamically adapt the topology of the C-RAN to minimize handover in the network. The proposed method is evaluated with a real-world dataset. In their later work [83], they proposed an algorithm to optimize the mapping between BBUs and RRHs on a time-varying graph representation of the C-RAN. Evaluation results in two large-scale real-world datasets confirmed that the total number of handovers is reduced by more than 20% compared with the current architectures.

2.3 Big Data Analytics Methodologies

With the emergence of ubiquitous sensing and computing diagrams [42], a massive number of mobile datasets have been available for academic research and industrial analytics. They can be collected either from operators' infrastructures [84–86], or by leveraging mobile crowdsensing paradigms [87, 88] with user participation. For example, Telecom Italia [86] have released a large-scale call detail record dataset containing two-months of calls, SMSs and network traffic data from the city of Milan and the province of Trentino, Italy. Orange [84] have also granted access to researchers participating in their Data for Development (D4D) challenges the access to a large-scale anonymized call detail record dataset, which consists of phone calls and SMS exchanges between five million of Orange's customers in Ivory Coast in half a year.

These heterogeneous mobile big data have been extensively analyzed in the literature to retrieve interesting and informative knowledge. For example, Furno et al. [74] proposed a data analytics framework to builds profiles of the city-wide traffic demand, and identifies unusual situations in network usages, aiming at facilitating the design and implementation of cognitive networking [34]. In their later work, Furno et al. [89] further investigated the heterogeneous patterns emerging in the mobile communication activity recorded within metropolitan regions. They applied the proposed method to extensive real-world data collected by mobile operators in ten cities. Results indicates the diversity of baseline communication activities across cities in different countries, as well as the common mobile traffic signatures in all studied areas. Cici et al. [90] studied the decomposition of cell phone activity series, and connect the decomposed series to socio-economic activities such as regular working patterns and opportunistic social events.

Among these big data analytics works, two of the most relevant topics to this dissertation are (1) spatial-temporal traffic clustering, and (2) spatial-temporal mobility prediction. We present a literature review of related works as follows.

2.3.1 Spatio-Temporal Traffic Clustering

In the literature of mobile big data analytics, clustering is a very important and useful technique for *discovering patterns* from a wide range of spatial regions [74], and for *reducing fluctuations* in individual spatial areas [91]. In these situations, the formed clusters also need to meet some explicit and implicit constraints, including the geographic distance of the cluster, the global constraints on the resource blocks available, etc. We survey these works on spatio-temporal data clustering and allocation as follows.

Clustering for discovering patterns. The spatio-temporal dynamics of various stations in a network may exhibit similarities and correlations, which can provide useful knowledge about the hidden urban functions and demographic characteristics in these regions [92]. In [93], Naboulsi et al. proposed a framework to identify a set of clusters of mobile call profiles, and classify the network usages accordingly. By analyzing the cluster traffic patterns, they were able to identify normal and outlying call behaviors. The framework was evaluated on a large-scale call detail record dataset. Similarly, Cici et al. [90] proposed a spectral method to cluster area units with similar activity patterns and validate the results with external municipal and social data sources. Furno et al. [74] proposed to cluster the mobile traffic demand in the temporal dimension, by adopting a hierarchical clustering method on the city-wide traffic snapshots. These methods laid the foundation for our work to cluster the spatio-temporal traffic data in a unified framework.

Clustering for reducing fluctuations. In many real-world network systems, due to insufficient users, over-sparse placement, or limited capacities, the stations (e.g., RRHs in mobile networks) usually observe fluctuating traffic and user mobility patterns. For example, in bike sharing systems, the number of bike usage exhibits highly dynamic and fluctuating patterns due to that fact that users move freely and pick up and return bikes arbitrarily. To address this issue, researchers have proposed to group these stations into clusters, and perform data analytics on a cluster-level. For example, Li et al. [61] first proposed a method to cluster bike stations based on their geographical locations and transition patterns, then predicted the bike usage of the whole system, and finally allocated the overall bike rental and return number to each cluster based on a proportion learned from historical data. However, the cluster scheme is static across different contexts. Etienne et al. [50] introduced a model-based method to group stations with similar bike usage patterns, such as stations near restaurants and train stations, and predicted their bike usage pattern in different temporal settings. These cluster-level prediction methods could improve the prediction accuracy, however the clusters used in these methods are static regardless of context at the time. Since the dynamics of these stations might be affected by various contextual factors, the prediction results of static clusters may not yield consistent accuracy across different contexts.

2.3.2 Spatio-Temporal Mobility Prediction

During the past decades, spatial-temporal mobility data modeling and prediction have been extensively studied in the literature [94, 95]. We survey two of the popular approaches in the literature as follows.

Autoregressive Integrated Moving Average (ARIMA) models: In time series analysis, ARIMA models are commonly used to fit a time series data and to forecast future variations in the series [96, 97]. ARIMA models explicitly extract from a time series three intuitive features, i.e., *auto-regression, moving average, and integration*. The auto-regression (AR) part indicates that the evolving variable of a time series is regressed on its own lagged values. The moving average (MA) part indicates that the regression error can be represented as a linear combination of error terms dependent on the values in the past. The integration (I) part is applied to the regression model to represent non-stationary time series (i.e., the variable in the time series shows a trend of increasing or decreasing). ARIMA models are capable of rapidly adjusting for sudden changes in trend, and it has been proved successful in many short-term forecasting problems [97, 98].

However, for long-term forecasting problems which involve predicting multiple future steps, the error of ARIMA models *accumulate* significantly and the forecasting confidence *decrease* rapidly as the forecasting step grows [96]. In this dissertation, we need to accurately forecast the RRH traffic for several hours to foresee the traffic patterns in the future for RRH clustering, which poses great challenge for ARIMA models. Moreover, ARIMA models are usually used to capture the trends and dynamics of *single* variables, and it is not trivial to directly applying ARIMA models to multi-variables simultaneously [99]. Therefore, it leads to another challenge when dealing with spatio-temporal data, where multiple time series denoting traffic of various correlated regions should be modeled at the same time.

Artificial Neural Network (ANN) models: Recently, ANN models are widely employed to understand time series and forecast the future trend of various kinds of spatio-temporal data [95, 100]. The basic procedure of applying ANNs to time-series modeling is by leveraging a sliding-window-based technique, which can be named *windowed-ANN*, or WANN [101]. More specifically, this technique first slices a time series into several equal-length windows, and then feeds these windows into an ANN model as *features*. The *output* of the model is the forecast of the future values of the time series, which can either be short-term or long-term results, depending on the application scenario. The WANN models have been applied in various domains, such as financial market [102] and operation research [100].

However, one of the biggest problem of the WANN model is its incapability to model the *temporal dependence* between the elements in each time series window. In fact, the elements in a window is treated equally as input features and thus the *sequential order* of the elements is ignored. As a result, the WANN model can make fluctuating and inconsistent forecasts which are not desired in our problem.

In this dissertation, we propose a deep-learning-based [103] architecture to model the temporal dependency and the spatial correlations among RRHs in a unified framework. Such kind of spatial-temporal deep-learning framework has been widely used in IP and transportation network traffic prediction [75,104], electronic health records understanding [105], and human motion and behavior recognition [106]. For a survey about deep-learning-based prediction model for spatio-temporal data, the reader is referred to [75].

2.4 Summary

In this chapter, we have reviewed the literature related to big data-driven network optimization. We start from a brief introduction of the optimization of urban transportation and communication networks. Afterwards, we present the relevant works in data analytics methodologies from both traffic and mobility perspectives. In the next chapters, we are going to introduce the applications of data-driven analytics in network optimization.

Chapter 3

Dynamic Traffic Clustering for Demand-Responsive Bikeshare Networks

Contents

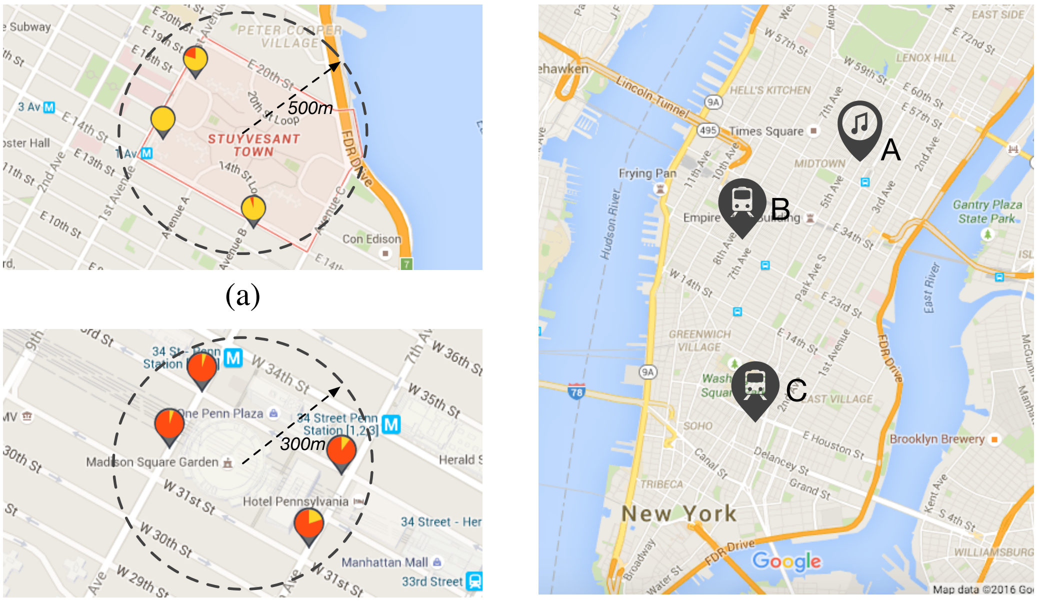
3.1 Introduction

In response to the growing concerns over urban sustainability, practices of green transportation such as bikeshare [107] have emerged. Today, more than 700 cities worldwide have launched bikeshare systems [54]. These systems allow people to pick up and drop off public bikes at self-service stations scattered around a city to make short trips. Given the large investment in infrastructure necessary to support a bikeshare system, such as setting up bike stations and renovating bike lanes, it is important for city authorities to ensure that the system is fully functional [108]. One of the key requirements is to prevent stations from *over-demand*, i.e., being completely empty or full over an extended period of time [49, 54]. Users' experiences may be greatly impaired if they run into an over-demand station, as they need to find another available station to rent or return the bike, which may ultimately hinder user participation in the bikeshare system [54, 56]. Therefore, city authorities often urge bikeshare system operators to resolve and prevent the over-demand problem, for example, by issuing fines when it occurs [109].

Operators have implemented different strategies to address the over-demand issue [59, 109], such as sending trucks to redistribute bikes before rush hours [35], or setting up temporary bike corrals for large social events to provide extra docks [59]. The ability to accurately foresee over-demand stations in the system is critical to the success of these strategies. However, predicting over-demand of individual stations is difficult as users usually choose a station near their origins or destinations on an ad hoc basis [54]. As a result, existing station-level bike demand prediction methods [51, 57] usually have relatively low accuracy.

Based on our observation, while the bike usage of a single station might exhibit high variability, the bike usage of the stations in a certain area over a certain time window (e.g., one hour) can have similar trends. For example, stations near a residential area in morning rush hours usually have more bikes rented than returned (Figure 3.1(a)), and stations near a stadium usually have a surge in dock demand before concerts (Figure 3.1(b)). Such bike usage patterns are highly *context dependent* [110, 111]: time of the day, day of the week, weather condition, social events, and traffic conditions can all lead to different bike usage patterns [49, 58, 112, 113]. Hence, we propose to *cluster* neighboring stations with similar bike usage patterns according to context, and predict over-demand at the cluster level. We define an over-demand cluster as a cluster containing at least one over-demand station in a given time window. Despite some existing work on bike demand prediction [56, 61] also consider station clustering to boost performance, they usually cluster stations *statically* regardless of context, which do not obtain consistent prediction accuracy when the context varies.

However, clustering stations and consequently predicting over-demand occurrence according to the varied and highly dynamic context is not trivial. In fact, bike usage patterns are mainly im-



(a) (b) (c)

Figure 3.1: Examples of bike usage patterns in different contexts. (a) Almost empty stations near a residential area in morning rush hours (7:00–8:00, 06/17/2015). (b) Almost full stations near a stadium before a concert (19:00–20:00, 05/13/2014). (c) An opportunistic context with a concert and two subway delays (12:00–13:00, 11/17/2015).

pacted by two types of contextual factors: (1) the *common contextual factors* that occur frequently and affect all the stations, such as time and weather, and (2) the *opportunistic contextual factors* that happen irregularly and only affect a subset of stations, such as social and traffic events. An intuitive method to cluster stations according to context is to build a statistical clustering template using historical records (e.g., a cluster template for sunny weekday rush hours). Then, given a specific context in a future time window, we can simply apply its corresponding template to cluster the stations and make cluster-level over-demand prediction. Although this template-based method can cope with the common contextual factors, it does not work well when incorporating the opportunistic contextual factors (events) that have rather few instances in history. In other words, these opportunistic events are sparse in time, making it difficult to find enough historical records containing the same events to generate a template. For example, Figure 3.1(c) shows a sunny weekday afternoon (12:00–13:00, 11/17/2015) with a concert in a stadium (Event A) and two subway delay events (Event B and C); no historical records having the same context can be found during the period from 01/01/2014 to 12/31/2015. Therefore, we need to design an effective method to model the impact of both common and opportunistic contextual factors simultaneously, which allows us to cluster station and predict over-demand accordingly.

In this work, we propose a *dynamic cluster-based* framework to predict over-demand occurrence in bikeshare systems according to context. First, we extract the common and opportunistic contextual factors from various urban data [42, 114, 115]. Then, depending on the current context, we construct a weighted correlation graph [116] to model the relationship among bike stations. Specifically, we take each station as a node and connect neighboring stations with links. We use the link weight of two stations to model the relationship between them with consideration of both common and opportunistic contextual factors. The link weight of two stations associated with the common contextual factors is calculated based on the correlation between their historical bike usage patterns, such that two stations with similar bike usage patterns have high link weight. The link weight of two stations with respect to the opportunistic contextual factors is calculated based on the number and types of events taking place near the stations, such that two stations impacted by the same array of events have high link weight. We then build the complete graph by merging the two sets of link weights, and group highly connected stations into clusters, so that each cluster consists of neighboring stations with similar bike usage patterns. Finally, we estimate the number of bikes rented and returned in each cluster, and predict the cluster over-demand probability accordingly. The contributions of this work include:

- To the best of our knowledge, this is the first work on dynamic cluster-based over-demand prediction according to context. Such a dynamic clustering approach leads to high and consistent over-demand prediction accuracy in bikeshare systems.
- We propose a two-phase framework to predict over-demand clusters by considering both common and opportunistic contextual factors. In the *dynamic station clustering* phase, depending on the context, we build a weighted correlation graph to model the relationship among bike stations, and propose a geographically-constrained clustering method to dynamically cluster stations over the graph. In the *over-demand cluster prediction* phase, we first estimate the number of bikes rented and returned in each cluster, and then adopt Monte Carlo simulation to predict the cluster over-demand probability.
- We evaluate the performance of our framework using two years of real-world bikeshare data and urban data in New York City and Washington, D.C. Results show that our framework accurately predicts over-demand clusters across different contexts in both cities (e.g. with 0.882 precision and 0.938 recall in NYC), and outperforms the start-of-the-art methods.

3.2 Preliminary and Framework

Definition 1. Station Status*: the status of station i at time t is defined as a tuple* $\langle B_i(t), D_i(t)\rangle$*, where* $B_i(t)$ *and* $D_i(t)$ *are the number of available bikes and docks in station i at time t, respectively.*

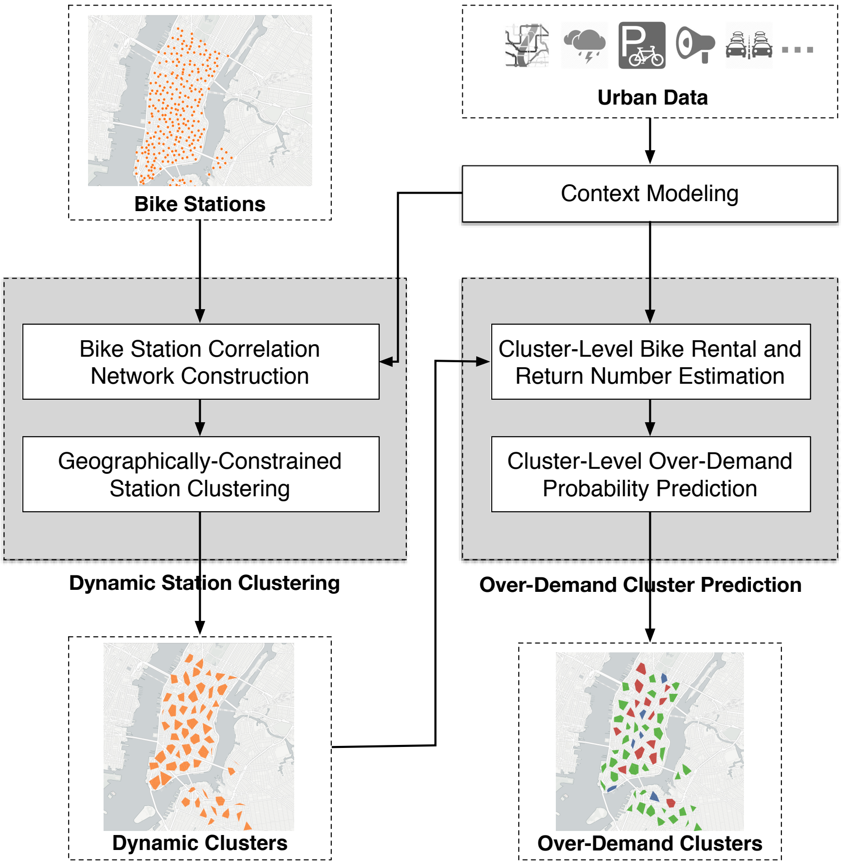

Bike Stations
Urban Data
Context Modeling
Bike Station Correlation Network Construction
Geographically-Constrained Station Clustering
Dynamic Station Clustering
Cluster-Level Bike Rental and Return Number Estimation
Cluster-Level Over-Demand Probability Prediction
Over-Demand Cluster Prediction
Dynamic Clusters
Over-Demand Clusters

Figure 3.2: Overview of the framework.

Definition 2. Bike Usage*: the bike usage of station i in a given time window* $[t, t+\Delta t]$ *is defined as a tuple* $\langle U_i^-(t), U_i^+(t) \rangle$*, where* $U_i^-(t)$ *and* $U_i^+(t)$ *are the number of bikes rented from and returned to station i during* $[t, t+\Delta t]$*, respectively. We further define* $U_i^-(t)$ *and* $U_i^+(t)$ *as the* bike rental number *and* bike return number*, respectively, and the sum of absolute values of the bike rental and return number as the* bike usage number.

Definition 3. Context*: we denote the context of a bikeshare system in a time window* $[t, t+\Delta t]$ *as* $\Psi(t) = \langle \Psi_c(t), \Psi_o(t) \rangle$*, where* $\Psi_c(t)$ *denotes the common contextual factors including time and weather, and* $\Psi_o(t)$ *denotes the opportunistic contextual factors including social and traffic events.*

Definition 4. Over-Demand Station*: we define a station i as an over-demand station if the station is full or empty for a period of time longer than a threshold. In this work, we empirically set the threshold as 10 minutes.*

Definition 5. Cluster*: we define a set of neighboring stations with similar bike usage patterns in a given time window as a cluster C. We define the bike usage number of a cluster as the sum of the* *bike usage number of its member stations.*

Definition 6. Over-Demand Cluster*: we define an over-demand cluster as a cluster containing at least one over-demand station in a given time window*[1].

We propose a two-phase dynamic cluster-based framework to predict over-demand occurrence in a bikeshare system according to context. As shown in Figure 6.2, we extract discriminative features from urban data to model the contextual factors relevant to bike usage, such as weather condition and social events. In the dynamic station clustering phase, we first construct a weighted correlation graph to model the relationship among bike stations according to the current context, and then propose a geographically-constrained clustering method to cluster stations over the graph. In the over-demand cluster prediction phase, we first estimate the bike rental and return number in each cluster, and then predict the cluster over-demand probability.

3.3 Context Modeling Leveraging Urban Data

The bike usage pattern of a bikeshare system may be affected by various contextual factors, such as weather condition and social events [58, 112]. Traditionally, collecting city-wide context information usually requires substantial time and labor [42]. With the ubiquity of urban sensing infrastructures and paradigms, these contextual factors can now be captured at low cost via assorted urban data [114]. However, given the considerable volume and variety of urban data, we need to identify factors relevant to bike usage patterns for modeling contexts. To this end, we conduct a series of empirical studies to analyze the relationship between bike usage and various contextual factors.

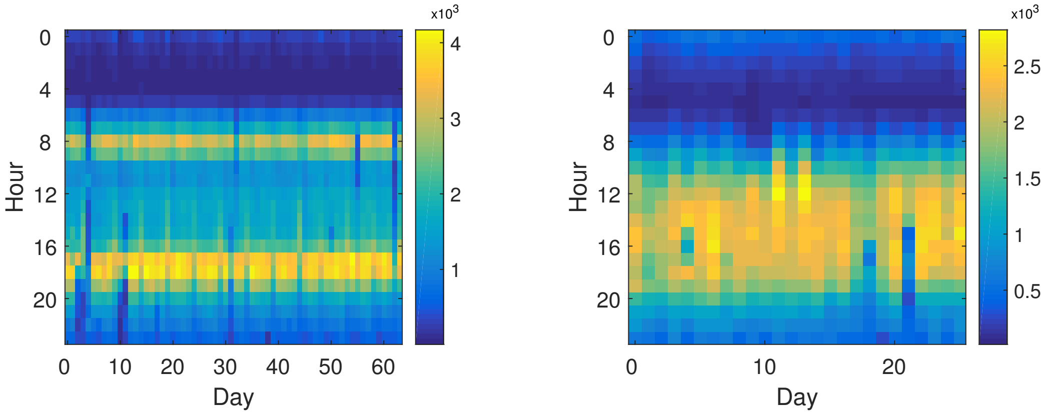
3.3.1 Common Contextual Factors

Based on previous studies and surveys [59, 61, 109], the common contextual factors relevant to bike usage patterns usually include *date and time*, *weather condition*, and *air temperature*. By exploiting the bikeshare data from the NYC Citi Bike system [117] and the meteorological data from the Weather Underground API [118], we study the impact of the common contextual factors as follows.

Date and Time

Intuitively, the bike usage pattern of a station might be different according to time of the day, and day of the week. However, there may be correlations and similarities among different temporal

[1] Our solution in this work can directly adapt to the definition of 'at least K over-demand stations' if necessary. For clarity, we focus on the definition of $K = 1$ now and discuss it later.



(a) Weekdays

(b) Weekends and holidays

Figure 3.3: The bike usage number of all stations in two months (06/01/2014–07/31/2014).

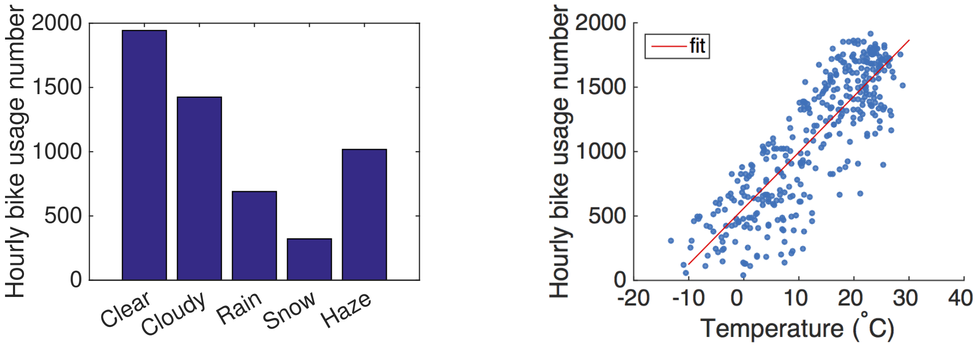
Table 3.1: Groups for modeling temporal context

Day type	**Group name**	**Time span**
Weekdays	morning rush hours	07:00–11:00
	day hours	11:00–16:00
	evening rush hours	16:00–20:00
	night hours	20:00–24:00
Weekends/Holidays	day hours	09:00–19:00
	night hours	19:00–01:00

groups. Figure 3.3 shows a sample of the bike usage number of all Citi Bike stations in two months from 06/01/2014 to 07/31/2014. We observe different bike usage patterns between weekdays and weekends/holidays, as well as between different hours of a day. Based on such observations, we derive six different *temporal groups*, as shown in Table 3.1. Note that we only consider the *active hours* with intensive bike usage, and discard temporal groups of 0:00–7:00 in weekdays and 1:00–9:00 in weekends/holidays.

Weather Condition

As presented in previous studies [59,119], bike usage patterns may vary significantly under different weather condition, such as rain or snow. We quantitatively study the relationship between the bike usage number and weather condition leveraging the hourly weather forecast data during the year of 2014. Specifically, we define the following five weather condition categories: *clear, cloudy, rain, snow, and haze*. Figure 3.4(a) shows the average hourly bike usage number of all stations under different weather condition. We observe that in rainy and snowy days, the bike usage number drops



(a) Weather condition

(b) Air Temperature

Figure 3.4: The hourly bike usage number of all stations across different meteorological contexts.

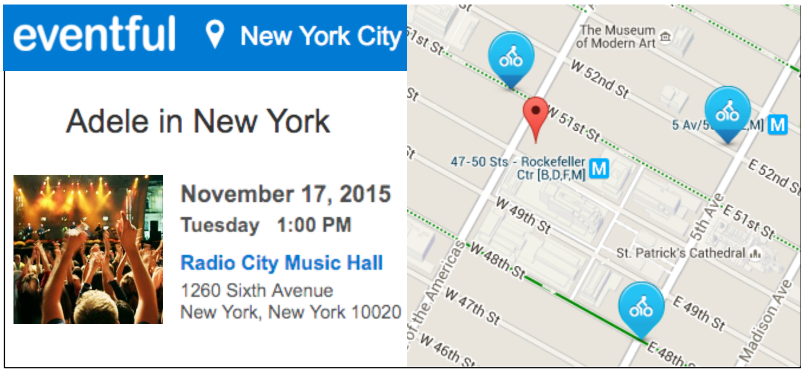
significantly, suggesting that weather condition should be considered as an important contextual factor impacting the bike usage patterns.

Air Temperature

Similarly, air temperature is also considered as an important factor impacting the bike usage patterns [58,119]. By exploiting the same weather forecast data, we study the relationship between the hourly bike usage number and the air temperature over the year of 2014. As shown in Figure 3.4(b), we observe strong correlation between the two variables. We empirically split the air temperature range into four groups according to the seasonal temperature variations, i.e. *below zero* ($< 0°C$), *cold* ($[0°C, 10°C)$), *comfortable* ($[10°C, 22°C)$), and *warm* ($\geq 22°C$).

3.3.2 Opportunistic Contextual Factors

The opportunistic contextual factors, including *social events* and *traffic events*, may cause unusual bike usage in a subset of stations near the event locations [112, 119, 120]. For social events, the impact on bike usage may be observed before, during and after the events. As the information about the event time and location is usually posted by organizers in advance, we can model the impact of these social events in the corresponding time windows. For traffic events (e.g., subway delays), the impact on bike usage is usually observed after the occurrence of the events with a delay. As such traffic events are published by urban authorities in real time, we can model the after-event impact for these traffic events.



eventful New York City

Adele in New York

November 17, 2015
Tuesday 1:00 PM
Radio City Music Hall
1260 Sixth Avenue
New York, New York 10020

Figure 3.5: An example event bulletin containing event name, type, time, and location. Bike icon denotes the nearby bikeshare stations.

Table 3.2: Top 5 most impactive social and traffic event types

Social event	IF	Traffic event	IF
City festival	4.73	Subway delay	1.47
Sports game	3.30	Traffic accident	1.23
Concert	3.24	Road restriction	1.20
Street fair	2.67	Traffic congestion	1.14
Parade/Marathon	2.33	Transit incident	1.11

Social Event

Riding public bikes to attend social events is a convenient transportation mode, especially when there are vehicle restrictions or traffic congestion in the event locations. In order to quantitatively study the impact of social events on bike usage, we collect the event bulletin data from the Eventful API [121]. Figure 3.5 shows an example event bulletin for a concert with detailed event name, type, time, and location. For each event, we select the stations located within a walking distance τ of the event location (we empirically set $\tau = 620m$ based on experiment results as discussed later), and then compare the bike usage number of these stations from one hour before the event start time to one hour after the event end time with the value in the same time window without event. We define the *impacting factor (IF)* of each event as the ratio of the event-time bike usage number to the normal value, and derive the *IF* of each event type. Table 3.2 shows the top 5 most impactive social event types on bike usage with regard to the *IF*.

Traffic Event

Previous surveys [59, 107] have shown that people might resort to public bikes as an alternative means to avoid transportation problems, such as subway delays and traffic accidents. We quantitatively study the impact of these traffic events by exploiting the NYC 511 traffic data feed [122] and the subway delay alerts from the NYCT Subway Twitter account [123]. We employ a similar method as mentioned in the social event analysis to calculate the impacting factor for each type of traffic event on its nearby stations in the next hour after the traffic event occurs. The top 5 most impactive traffic event types are also presented in Table 3.2.

3.4 Dynamic Station Clustering

In this phase, our objective is to dynamically group neighboring stations into clusters according to context, so that the stations in the same cluster have similar bike usage patterns. To this end, we first model the relationship among bike stations using a *weighted correlation graph* [116], which has been widely used in bioinformatics applications such as gene co-expression graph analysis [124, 125]. Specifically, we regard bike stations as nodes, and connect two stations with a link if they are geographically close to each other. We calculate the weight of each link according to the associated common and opportunistic contextual factors, and merge them together to construct the weighted graph model.

We then group neighboring stations with similar bike usage patterns into clusters. These clusters can be considered as communities that are densely connected internally and loosely connected between each other [126]. In the literature, various algorithms have been proposed to find community structures in a graph, such as the Label Propagation algorithm [127] and the Girvan-Newman algorithm [128]. However, directly applying these algorithms to detect communities may not be adequate in our scenario, since we also need to constrain the geographic span of the formed clusters within a reasonable bound for practical purposes. For example, a single cluster spanning several kilometers is not useful for operators to schedule bike redistribution routes or set up temporary bike corrals. Therefore, we proposed a *Geographically-Constrained Label Propagation (GCLP)* method to solve this problem.

3.4.1 Station Correlation Network Construction

We model the relationship among bike stations as an undirected, weighted graph $G = (V, E)$, where $V = \{s_1, \ldots, s_N\}$ denotes the set of N stations, and E denotes the set of links between two stations. We then define the adjacency matrix A of graph G, which is an $N \times N$ symmetric matrix with entries $a_{i,j} = 1$ when there is a link between station s_i and station s_j, and $a_{i,j} = 0$ otherwise $(i, j = 1, \ldots, N)$. We further determine the weight of each link $w(s_i, s_j)$ based on the common and opportunistic contextual factors.

Adjacency Matrix

By definition, only neighboring stations could be grouped into the same cluster. Therefore, we use the geographic distance of two stations to determine whether they are adjacent or not. More specifically, for station s_i and station s_j, we define:

$$a_{i,j} = \begin{cases} 1, & \text{if } dist(s_i, s_j) \leq \tau \\ 0, & \text{otherwise} \end{cases} \tag{3.4.1}$$

where $dist(s_i, s_j)$ is the geographic distance between the two stations[2], and τ is a *neighborhood threshold* controlling the geographic distance of neighboring stations.

Link Weight

We determine the link weight by considering both common and opportunistic contextual factors as follows:

$$w(s_i, s_j) = a_{i,j} \times (\mu\ w_c(s_i, s_j) + (1 - \mu)\ w_o(s_i, s_j)) \tag{3.4.2}$$

where $w_c(s_i, s_j)$ and $w_o(s_i, s_j)$ correspond to the link weight associated with the common and opportunistic contextual factors, respectively, as detailed later. $\mu \in (0, 1)$ controls the influence degree of each type of contextual factor. We consider the case of normalized symmetric positive weights ($w(s_i, s_j) \in [0, 1]$) with no loops ($w(s_i, s_i) = 0$). We note that $w(s_i, s_j) = 0$ when there is no link between s_i and station s_j ($a_{i,j} = 0$).

In order to calculate the link weight associated with the common contextual factors $w_c(s_i, s_j)$, we characterize the two stations by the historical bike usage records having the same common contexts. More specifically, for the two stations s_i and s_j composing the link, we construct a corresponding feature vector

$$\mathbf{f}_c(s_i) = [U_i^+(t_1), U_i^-(t_1), \ldots, U_i^+(t_K), \ldots, U_i^-(t_K)] \tag{3.4.3}$$

and

$$\mathbf{f}_c(s_j) = [U_j^+(t_1), U_j^-(t_1), \ldots, U_j^+(t_K), \ldots, U_j^-(t_K)] \tag{3.4.4}$$

respectively, using the bike rental and return number of historical records having the same common contexts Ψ_c. We remove records with over-demand stations, since in these situations the observed bike rental or return number may be relatively small and not rewarding the potential demand on the station, as users are not able to rent or return bikes in the station. We then calculate the Pearson

[2] Here we use the city-block distance to approximate the real-world walking or riding distance between stations.

correlation coefficient [129] of $\mathbf{f}_c(s_i)$ and $\mathbf{f}_c(s_j)$, denoted as $corr_c(s_i,s_j)$, and normalize it to $[0,1]$ to obtain the link weight associated with the common contextual factors, i.e.,

$$w_c(s_i,s_j) = \frac{1+corr_c(s_i,s_j)}{2} \tag{3.4.5}$$

In order to calculate the link weight associated with the opportunistic contextual factors $w_o(s_i,s_j)$, we characterize the two stations by the number and type of events taking place near the stations. More specifically, for the two stations s_i and s_j composing the link, we search for the events taking place within the neighborhood threshold τ of each station, and count the number of events by type as defined in Table 3.2. We construct a feature vector $\mathbf{f}_o(s_i) = [V_i(1),\ldots,V_i(10)]$ and $\mathbf{f}_o(s_j) = [V_j(1),\ldots,V_j(10)]$, where each $V_i(m)$ and $V_j(m)$ ($1 \leq m \leq 10$ since we consider 5 social event types and 5 traffic event types) corresponds to the number of events of type m taking place near station s_i and s_j, respectively. Similarly, we then calculate the Pearson correlation coefficient of $\mathbf{f}_o(s_i)$ and $\mathbf{f}_o(s_j)$, denoted as $corr_o(s_i,s_j)$, and normalize it to $[0,1]$ to obtain the link weight associated with the opportunistic contextual factors, i.e.,

$$w_o(s_i,s_j) = \frac{1+corr_o(s_i,s_j)}{2} \tag{3.4.6}$$

3.4.2 Geographically-Constrained Station Clustering

Problem Formulation

In this step, we need to group stations into clusters, so that each cluster consists of neighboring stations with similar bike usage patterns. In the constructed station correlation graph, as the link weight encodes the similarity between the two nodes, we need to cluster nodes with high link weights together, which can be identified as a community detection problem [128]. Specifically, given the weighted correlation graph $G = (V,E)$, we first define a set of clusters $\mathbb{P} = \{C_1,\ldots,C_K\}$, where

$$\cup_{\forall C_k \in \mathbb{P}} = V \quad \text{and} \quad \cap_{\forall C_k \in \mathbb{P}} = \emptyset \tag{3.4.7}$$

Then, given a node v, we define the *connectivity* of v to a cluster C as the sum of link weights between v and the nodes in the cluster C:

$$con(v,C) = \textstyle\sum_{v' \in C} w_{v,v'} \tag{3.4.8}$$

Finally, we define the *adjacent clusters* $\mathbb{C}(v)$ of node v as

$$\mathbb{C}(v) = \{C|con(v,C) > 0, C \in \mathbb{P}\} \tag{3.4.9}$$

With the above definition, our objective is to find an optimal set of clusters $\mathbb{P}$, such that the internal connectivity within a cluster is higher than the inter-cluster connectivity, i.e.,

$$\forall v \in C_k,\ con(v, C_k) \geq max\{con(v, C_l), C_l \in \mathbb{P}\} \tag{3.4.10}$$

We also need to bound the geographic span of a cluster within the neighborhood threshold, i.e.,

$$\forall v, v' \in C_k,\ dist(v, v') \leq \tau \tag{3.4.11}$$

Clustering Method

To obtain clusters with high internal connectivity (Equation 4.4.11) while meeting the geographic constraint (Equation 4.4.12) at the same time, we propose the Geographically-Constrained Label Propagation (GCLP) algorithm, which is built on the popular community detection algorithm Label Propagation [127]. The basic idea of GCLP is iteratively assigning nodes to the adjacent clusters, where the gain of assigning node v to cluster C is evaluated by a *value function*. Based on previous discussion, the value function should be designed to reward the connectivity $con(v, C)$ and penalize the geographic span $dist(v, v'), \forall v' \in C$. Therefore, we define the value function as

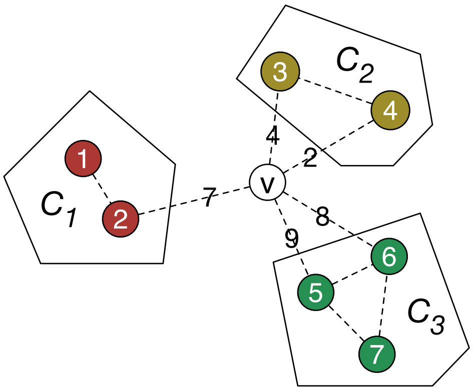
$$value(v, C) = con(v, C) \times log\Big(\frac{\tau}{max\{dist(v, v')\}}\Big) \tag{3.4.12}$$

The GCLP method greedily assigns the node to the adjacent cluster with highest value[3] until none of the nodes are moved among clusters [127]. As the convergence of such a greedy approach is hard to prove [130], we set a maximum iteration number *max_iter* to ensure the algorithm will stop.

Example We use an example to illustrate the node assignment process. As shown in Figure 3.6, node v has three adjacent clusters C_1, C_2, C_3, and the connectivity between v and each adjacent cluster is $7, 4+2, 9+8$, respectively. The maximum distance between v and each cluster is $dist(v, v_1) = 900m, dist(v, v_4) = 500m, dist(v, v_7) = 950m$, respectively. Suppose the neighboring threshold $\tau = 620m$, then the value function of each cluster will yield $-1.13, 0.65, -1.30$, respectively. Hence, we assign node v to cluster C_2 with the highest value.

Algorithm The GCLP algorithm is initialized by assigning each node in the graph to a unique cluster label. In each iteration, we randomly populate a list of nodes L, and traverse the list to update the cluster label of each node. The label update process is as follows. First, we remove the node from its current cluster, and find the set of adjacent clusters to the current node. Then, we compute the value function for all the adjacent clusters, and assign the node to the cluster with the highest value. We mark the the node as *moved* among clusters if its new cluster label is different from the old one. After we finish iterating over the node list, we decide whether to perform another

[3] If two clusters yield the same value, we randomly choose one.



dist(v,v$_1$) = 900m	
dist(v,v$_2$) = 600m	
dist(v,v$_3$) = 450m	
dist(v,v$_4$) = 500m	
dist(v,v$_5$) = 500m	
dist(v,v$_6$) = 550m	
dist(v,v$_7$) = 950m	

Figure 3.6: An illustrative example of the node assignment process of the GCLP algorithm. The number on each link denotes the weight, and *dist* is the geographic distance between two nodes.

iteration or finish the algorithm based on the following stop criteria: (1) the user specified maximum iteration number *max_iter* is reached, or (2) none of the nodes are moved among clusters.

Time Complexity For each iteration of the GCLP algorithm, it first takes $O(|V|)$ steps for node permutation, and then processes all the links when computing the value function for each node, taking $O(|V|*|E|)$ steps in the worst case. Since we limit the number of iterations by *max_iter*, the final time complexity of the algorithm is $O(|V|*|E|)$.

3.5 Over-Demand Cluster Prediction

After grouping stations into clusters, our objective in this phase is to predict the occurrence of over-demand clusters. An intuitive method is to directly model the cluster over-demand probability with regard to the contextual factors. However, since the opportunistic contextual factors are sparse in time, it is difficult to find enough samples for a specific context to train the model. Moreover, the ad hoc bike usage behaviors within a cluster also introduce uncertainty in over-demand prediction. To address these issues, we first estimate the bike rental and return number of each cluster, and then adopt Monte Carlo simulation to predict the cluster over-demand probability.

We separately exploit the common and opportunistic contextual factors to estimate the bike rental and return number of a cluster. Specifically, we first estimate the *base* bike rental and return number of the cluster leveraging historical records having the same common contextual factors. We then infer an *inflation rate* [131] to quantitatively measure the impact of the nearby social and traffic events on the cluster. Finally, we multiply the base bike rental and return number by the inflation rate to obtain the final estimation value the cluster.

With the estimated bike rental and return number and the current station status of a cluster, we adopt Monte Carlo simulation [132] to predict the over-demand probability for each cluster. Specifically, we first model the bike rental and return events in the prediction time window as a Poisson process [133] parameterized by the predicted bike rental and return number. We then generate two stochastic sequences [134] of bike rental and return events based on the corresponding distributions. We simulate the bike rental and return processes by randomly dispatching these events to available stations in the cluster in chronological order, until a station over-demand occurs (i.e., the station stays full or empty for more than 10 minutes), or both sequences are traversed. We repeat the simulation for a number of times, and use a threshold to detect over-demand clusters.

3.5.1 Bike Rental and Return Number Estimation

First, we estimate the base bike rental and return number of a cluster using the cluster's average value in historical records having the same common contextual factors. Note that we deliberately remove records with social or traffic events in the cluster, since in these records, the bike rental and return number caused by opportunistic events are mixed with the ones related to the common contextual factors.

Then, we model the inflation rate at the event type level. We assume that under the same common context, the same type of events have similar inflation rates on the nearby clusters. Here we define an event as being near a cluster if the geographic distance of the event and the cluster center is within the neighborhood threshold τ. Specifically, under a common context $\Psi_c(t)$, we denote the inflation rate of event type i as θ_i ($i = 1, \ldots, 10$ corresponding to the types in Table 3.2). For cluster C, the overall inflation rate is then $\sum_{i=1}^{I} n_i \, \theta_i$, where n_i is the number of events of type i observed near the cluster. In order to infer each θ_i, we select historical records of cluster C with events under the same common contexts $\Psi_c(t)$, and calculate the overall inflation rate in each record by dividing the bike rental and return number by the base number of the cluster (which is calculated in the previous step). We collect the corresponding event number and the overall inflation rate from all clusters, and train a linear regression [135] model to infer each θ_i. With the learned θ_i, we calculate the overall inflation rate for cluster C.

Finally, we multiply the base bike rental (return) number by the overall inflation rate to obtain the final prediction of the bike rental (return) number for each cluster.

3.5.2 Over-Demand Probability Prediction

Given the predicted bike rental and return number in a cluster, we adopt a Monte Carlo method to simulate the bike rental and return process in the cluster. According to [50], the number of bikes rented or returned in the predicted time window follows a Poisson distribution. We take the bike return number as an example to elaborate on the details. Given a cluster C with the predicted bike return number N_C in the time window $[t, t+\Delta_t]$ (e.g., one hour), we divide Δ_t into T small consecutive time intervals $\delta_t = \Delta_t / T$ (e.g., one minute). The number of bikes returned to this cluster k in δ_t follows a Poisson distribution with mean parameter $\lambda = N_c / T$:

$$p(k|\lambda) = \frac{e^{-\lambda}\lambda^k}{k!}, \quad k = 0, 1, 2, \ldots \tag{3.5.13}$$

We then generate a stochastic sequence $Q_i^+ = [k_1, \ldots, k_T]$ from the distribution to simulate the bike return events in the cluster. Similarly, we generate a stochastic sequence Q_i^- based on the bike rental distribution for the bike rental events.

Afterward, we randomly dispatch the bike return and rental events from both sequences to any available stations in chronological order[4]. If a station is observed to be full or empty for more than 10 minutes, we mark the cluster as an over-demand cluster and stop the process. Otherwise we traverse the sequences and output the cluster as a normal cluster in the given time window. We note that if we define the over-demand cluster as 'containing at least K over-demand member stations', our method can directly adapt to the new definition by observing K over-demand stations in the cluster before marking the cluster as being an over-demand cluster.

We repeat the simulation for Γ times (e.g., 10,000 times) to count the over-demand occurrences γ, and estimate the over-demand probability of the cluster as the rate $p = \gamma / \Gamma$. We use a *discrimination threshold* ε to classify a cluster as an over-demand cluster if $p \geq \varepsilon$.

3.6 Evaluation

3.6.1 Experiment Settings

Datasets

We evaluate our framework in New York City and Washington, D.C., respectively. We collect bikeshare data and context data for two years (01/01/2014–12/31/2015), as presented in Table 6.1. The data processing details are as follows.

- *Bike sharing data*: we collect two years' bike trip historical records from the data portals of NYC Citi Bike [117] and DC Capital Bikeshare [60], respectively. The data format of each trip record is: *(rental station, rental time, return station, return time)*. Based on the records, we count the bike rental number and bike return number in each hour for each

[4] In reality, users might have preferences on specific stations, while such preferences are not always significant and consistent within a small cluster based on our observations on the dataset. We plan to model user preferences in our future work.

Table 3.3: Summary of Datasets

Data type	Item	New York City	Washington, D.C
Bike sharing	# Stations	327	203
	# Bike trips	18,019,196	6,138,428
	# Station status	hourly	hourly
	# Over-demand	626,856	318,576
Contextual factors	# Weather forecast	hourly	hourly
	# Social events	435	329
	# Traffic events	958	745
Data collection period		01/01/2014–12/31/2015	

station, respectively. We also collect the hourly *station status* data from the Citi Bike station feed [117] and the Capital Bikeshare station feed [60], respectively, to obtain the number of available bikes and docks in each station at the beginning of each hour.

- *Meteorological data*: we retrieve the hourly weather forecast data for both cities from the Weather Underground API [118], and parse the weather condition and air temperature value for each hour based on the data.
- *Social event data*: we compile a list of social events from the Eventful API [121] in the two years for both cities. We select events based on the types defined in Table 3.2. Each social event record contains the following fields: *(name, type, time, location)*.
- *Traffic event data*: we retrieve the traffic events of NYC from the NYC 511 traffic feed and the NYCT Subway Twitter account, and the traffic events of DC from the DC Police Traffic Twitter account [123]. We process these data records and filter relevant traffic events based on Table 3.2.

We collect the ground truth of over-demand clusters as follows: at the beginning of the hour, we obtain the current numbers of available bikes and docks in each station of a cluster from the station feeds, and then update the status of each station based on the bike rental and return data during the hour. As soon as we observe a station staying full or empty for more than 10 minutes, we mark the enclosing cluster as an over-demand cluster. Otherwise, we mark the cluster as normal in the hour. In this way, we obtain 626,856 and 318,576 over-demand events in NYC and DC during the two years, respectively. These over-demand events usually occur in stations near transportation hubs during rush hours, and stations near parks during weekend daytime.

Table 3.4: The contingency table with an example

Truth / Prediction	Over-demand clusters	Normal clusters
Over-demand clusters	True Positive (TP) 11	False Positive (FP) 2
Normal clusters	False Negative (FN) 1	True Negative (TN) 54

Evaluation Plan

We use the data of 2014 as the training set to learn the relationship between bike usage patterns and contextual factors, and use the data of 2015 for evaluation. We perform a prediction every hour during the active hours of a day. For each prediction, we first obtain the context of the corresponding time window, including the temporal group, the weather and temperature forecast, the social events starting/happening/ending in the next hour, and the traffic events occurred in the previous hour. We then dynamically cluster stations according to the context, and predict the over-demand clusters for the corresponding time window.

Evaluation Metrics

We compare the over-demand prediction of each cluster to the ground truth, and organize the results according to Table 3.4. For example, Table 3.4 shows a clustering scheme with 68 clusters, among which 12 clusters are over-demand, and the proposed method successfully predicts 11 of them. We define the following metrics to evaluate the prediction accuracy [136]:

$$precision = \frac{|TP|}{|TP| + |FP|}, \quad recall = \frac{|TP|}{|TP| + |FN|} \tag{3.6.14}$$

$$F1\text{-}Score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{3.6.15}$$

To further evaluate the prediction performance, we draw the ROC Curve [137] by plotting the true positive rate ($\frac{|TP|}{|TP|+|FN|}$) against the false positive rate ($\frac{|FP|}{|FP|+|TN|}$) under various discrimination threshold settings. We compute the AUC (Area Under ROC Curve) [137] values as another metric to evaluate the prediction performance.

Baseline Methods

We name our method WCN-MC (Weighted Correlation Network and Monte Carlo simulation), and compare our method with two sets of baselines, i.e., the *station-level* and the *cluster-cluster* prediction methods. In particular, we design three station-level baselines:

- **ARIMA**: this baseline method models the number of available bikes (docks) in a station as a time series, and uses an auto-regressive integrated moving average (ARIMA) model [51] to predict the station status in the future. It then detects the occurrence of over-demand stations based on the predicted station status.
- **B-MC**: this baseline method uses a Bayesian network to model and predict the bike rental and return number of each station leveraging station status and the context features [56]. It then directly applies the Monte Carlo simulation method on each single station for over-demand prediction.
- **ANN-S**: this baseline method directly models the over-demand probability with regard to the current station status and the context features by leveraging an Artificial Neural Network (ANN) model.

To make a fair comparison with our method, for each of these station-level baselines, we further infer its cluster-level prediction by clustering the stations in the same way as our method. We also design three cluster-level baselines as follows:

- **SC-MC**: the Static Clustering (SC) baseline method uses the clustering approach proposed by [61] to group stations into static clusters based on the geographic distance and the bike usage patterns of stations in all contexts . It then uses the same Monte Carlo method as in WCN-MC to predict over-demand clusters.
- **CCF-MC**: the Common Contextual Factor-based Clustering (CCF) method does not consider the opportunistic contextual factors and use a template-based method in station clustering. It then applies the same Monte Carlo method as in WCN-MC to predict over-demand clusters.
- **ANN-C**: this baseline method uses the same clustering results from our method, and then directly predicts cluster over-demand probability based on the status of stations in the cluster and the context features using an ANN model. We design this method to verify the effectiveness of our Monte Carlo-based method.

3.6.2 Evaluation Results

We first present the overall prediction results in both cities, and then study the impact of two parameters (neighborhood threshold τ and discrimination threshold ε) on the NYC results, while the results of DC are similar.

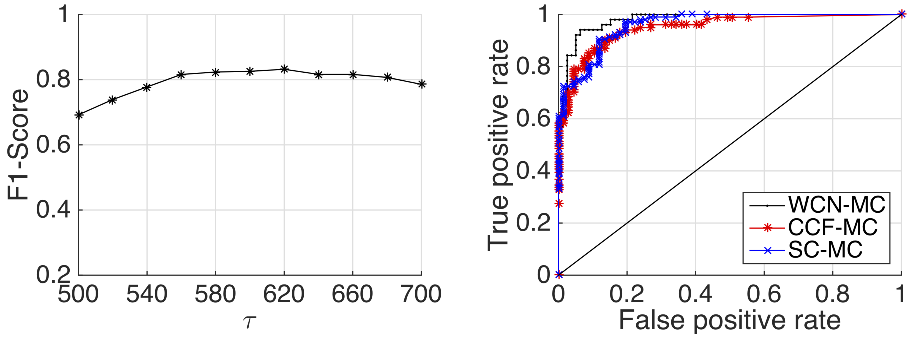
Overall Prediction Results

We compare the over-demand prediction results of different methods in Table 3.5. Our WCN-MC method achieves 0.882 precision and 0.938 recall in NYC, and 0.857 precision and 0.923 recall in DC, outperforming all the baseline methods. In general, the cluster-level methods achieve higher accuracy than the station-level methods. In particular, among the station-level methods, the context-aware method B-MC achieves significantly better results than the time series-based method ARIMA, which justifies the necessity of incorporating context information in over-demand prediction. Among the cluster-level methods, CCF-MC outperforms SC-MC by involving the common contextual factors in the clustering phase. Our WCN-MC method further improves the performance upon CCF-MC by considering not only the common contextual factors but also the opportunistic contextual factors. We also note that the ANN-S and ANN-C methods do not achieve best results in the corresponding station-level and cluster-level baseline groups, indicating that directly exploiting context features to model the over-demand probability does not achieve consistent improvement in prediction accuracy. In contract, our method separately models the impact of the common and opportunistic contextual factors and consistently achieves high over-demand prediction accuracy.

Parameter Impact Study

We examine the impact of the neighborhood threshold τ on the prediction performance. Based on bikeshare system operation reports [59, 108], we vary the threshold τ from $500m$ to $700m$, corresponding to the common walking distance range of users. Figure 3.7(a) shows the F1-Score under different τ values. We can see that setting a small neighborhood threshold leads to relatively lower accuracy, probably because the resulting clusters might be too small to exhibit consistent bike usage pattern. On the other hand, a large cluster might not be practically useful for operators. Therefore, we set $\tau = 620m$ in our experiments, and obtain an average of 67.08 clusters out of 327 stations. Each cluster contains an average of 4.74 stations with an average geographic span of $613.40m$. Based on this setting, we then determine the optimal influence degree $\mu = 0.53$ which maximizes the F1-Score.

We also study the prediction performance under different discrimination thresholds by varying the values of ε from 0 to 1. Figure 3.7(b) shows the ROC curve of our WCN-MC method as well as the two cluster-level baselines CCF-MC and SC-MC. Our method achieves an AUC of 0.97, which is higher than the two baselines (0.93 for CCF-MC and 0.89 for SC-MC, respectively). Based on the ROC plot, we select $\varepsilon = 0.71$ as the optimal discrimination threshold in our experiments.



(a) Neighborhood threshold (b) ROC curves

Figure 3.7: Parameter impact analysis.

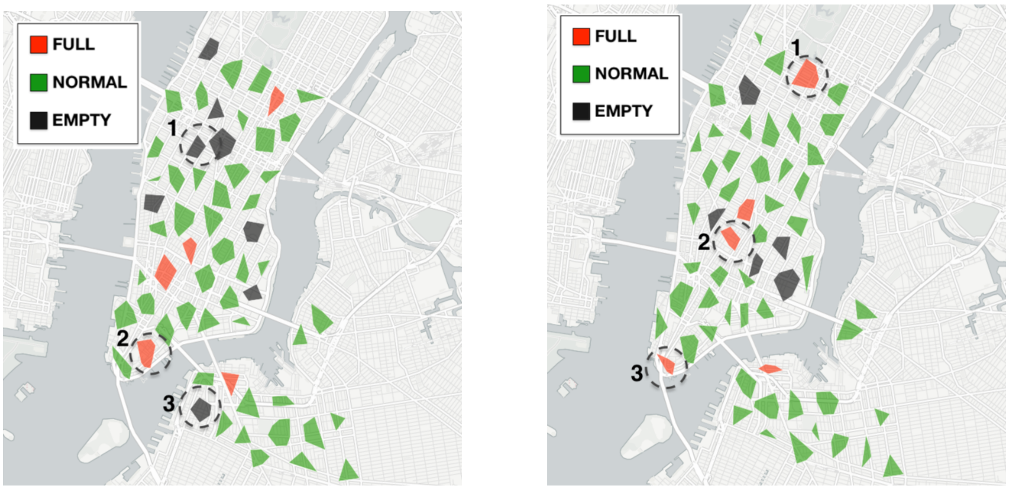
Table 3.5: Over-demand prediction results of different methods

Methods	Precision	Recall	F1	Precision	Recall	F1
	NYC			DC		
ARIMA	0.548	0.506	0.526	0.520	0.541	0.530
B-MC	0.753	0.656	0.692	0.636	0.539	0.583
ANN-S	0.776	0.571	0.658	0.667	0.428	0.521
SC-MC	0.790	0.647	0.711	0.793	0.821	0.807
CCF-MC	0.833	0.832	0.828	0.815	0.880	0.846
ANN-C	0.673	0.852	0.752	0.857	0.600	0.706
WCN-MC	**0.882**	**0.938**	**0.909**	**0.857**	**0.923**	**0.889**

3.6.3 Case Studies

Weekday Rush Hours

Figure 3.8(a) shows the dynamic clustering and over-demand prediction results during the morning rush hours of a typical weekday (8:00–9:00, 06/07/2015), where the red/green/black colors encode full/normal/empty cluster status, respectively. We observe several clusters near major transportation hubs and business/residential districts, such as the Penn Station area (Circle 1), the Wall Street area (Circle 2), and the Brooklyn Heights area (Circle 3). During rush hours, these clusters are usually full or empty, revealing the underlying dynamics and directions of the commuting flow. With such knowledge, bikeshare system operators could take preventive actions to ensure the station availability, such as sending trucks to redistribute bikes among these areas before rush hours.



(a) Weekday morning rush hours

(b) Sunny spring weekend

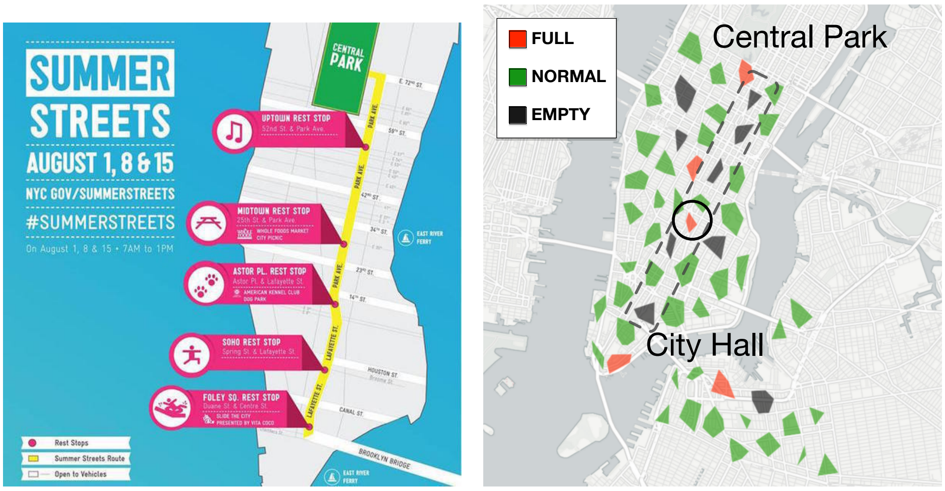
Figure 3.8: Dynamic clustering and prediction results in different temporal and weather contexts.

Weather Condition and Air Temperature

We present the result of a sunny spring weekend afternoon (14:00–15:00, 05/24/2015) in Figure 3.8(b). We observe several full clusters near the major parks of NYC, such as Central Park (Circle 1), Union Square Park (Circle 2), and Battery Park (Circle 3). A possible explanation is that people like to ride bikes to parks to enjoy outdoor activities in the springtime [138]. With such knowledge, bikeshare system operators can provide more pleasant weekend riding experience by, for example, setting up temporary bike corrals around these parks to ensure sufficient docks.

Social Events

We study the case of the city festival Summer Streets [139] in 2015. Summer Streets is a celebration of NYC's streets on three Saturdays in August (we present the results of 12:00–13:00, 08/08/2015), featuring bike tours, block parties, and street arts along Park Avenue from Central Park to New York City Hall (Figure 3.9(a)). Taking the event information into account, our dynamic clustering and prediction method successfully identifies several empty clusters along Park Avenue near Central Park and City Hall, as highlighted in Figure 3.9(b). Interestingly, we notice a full cluster near Union Square (the circle in Figure 3.9(b)). We examine the events and find the *Union Square Greenmarket* [140] is being held in the park. The Greenmarket features foods and cooking demonstrations, which might attract crowds of riders to stop for a rest. With the prediction, operators can adjust bike redistributing plans in Park Avenue before the festival, and set up temporary bike corrals near Union Square.



(a) Summer Streets poster (b) Over-demand clusters

Figure 3.9: Clustering and prediction results of NYC Summer Streets.

Table 3.6: Running time analysis

Procedures	Time (ms)	
	NYC	DC
GCLP clustering	863	532
Bike usage number prediction	701	428
Monte Carlo simulation	8,523	5,349
Total	10,087	6,309

3.6.4 Running Time Analysis

We evaluate the runtime efficiency of our approach on a 64-bit server with an quad-core 3.20GHz CPU and 32GB RAM. We find that the prediction accuracy regarding F1-Score does not increase significantly when the Monte Carlo simulation times Γ exceeds 8,000. Therefore, we set $\Gamma = 8,000$ in each prediction cycle, and present the detailed processing time in Table 3.6. The average time for running a prediction is about 10 seconds for NYC Citi Bike system and about 6 seconds for DC Capital Bikeshare system, respectively.

3.7 Conclusion

In this work, we propose a dynamic cluster-based framework to predict over-demand occurrence in bikeshare systems according to the varied and highly dynamic contexts. To effectively model the relationship among bike stations, we consider two sets of contextual factors, i.e., the common contextual factors including time, weather, and temperature, and the opportunistic contextual factors including social and traffic events. We model the relationship using a weighted correlation graph, and propose a geographically-constrained clustering method to group stations into clusters. Evaluations on NYC and DC show that our framework consistently achieves high over-demand prediction accuracy in both cities across different contexts, and outperforms the start-of-the-art methods.

In the future, we intend to improve this work from the following aspects. First, we plan to better characterize the contexts with richer urban data, such as incorporating the social network check-ins. Second, we plan to explore the impacts of newly established stations and cluster size on the prediction accuracy. Third, we plan to evaluate our method on bikeshare systems in other cities with different cultural settings.

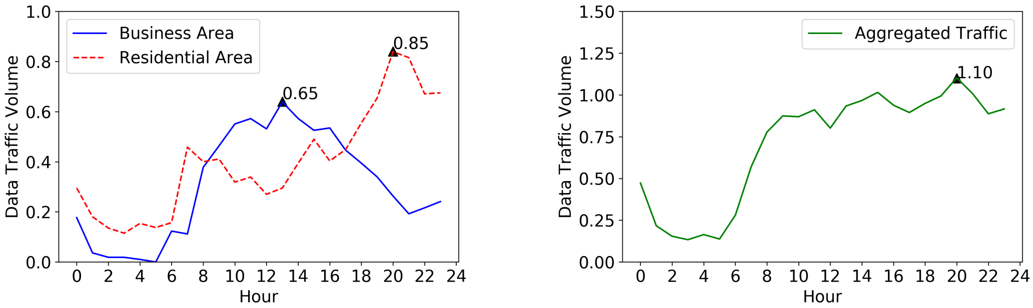
Chapter 4

Complementary Traffic Clustering for Cost-Effective C-RAN

Contents

4.1 Introduction

Today, mobile network data traffic is growing explosively as Internet-enabled smartphones and tablets become increasingly popular [141]. According to Cisco [19], global mobile network data traffic has grown 18-fold over the past five years, and the next-generation cellular systems (e.g., 5G) are expected to experience tremendous data traffic growth [142]. In order to accommodate



(a) (b)

Figure 4.1: (a) Data traffic patterns in different areas of Milan during a typical weekday. The blue solid line denotes the data traffic in a business district (Centro Direzionale), while the red dashed line corresponds to the data traffic in a residential area (Quintosolo District). (b) The aggregated data traffic pattern of the two areas. Triangles indicate the peak traffic hour and volume.

the fast growing data traffic demand, mobile network operators need to increase their *data processing capacity*, such as deploying more base stations, and adding more data processing units to base stations. Consequently, the *capital expenditures* of deploying these network infrastructures are becoming increasingly high, and may harm operator's revenue as network scale grows [21]. Moreover, the *operating expenses* of mobile network infrastructures, such as energy consumption and maintenance spending, are substantially increasing [24]. Therefore, optimizing the capital expenditures and operating expenses has become a necessity for mobile network operators [23, 143].

Even though the overall data traffic demand of the mobile network is growing, the demand in different areas and during different periods of time is not evenly distributed [144]. For example, as shown in Figure 4.1(a), the traffic in a business district (denoted as a blue solid line) observes peaks during working hours, while the traffic in a residential area (denoted as a red dashed line) is relatively higher during evening hours than in working hours. Such a *spatial-temporal non-uniform property* of traffic demand poses great challenges for operators to optimize the capital expenditures and operating expenses of their network infrastructures. On one hand, the data processing capacity of each base station needs to cover its peak traffic volume, leading to high *deployment cost*. On the other hand, the capacity in individual base station is wasted during off-peak hours, resulting in low *capacity utility*.

Fortunately, with the rapid evolution of mobile network architectures, the emergence of *Cloud Radio Access Network (Cloud-RAN)* [26] has presented new opportunities to address the above challenges. In Cloud-RAN, a traditional base station is split into two components: a *Remote Radio Head (RRH)* for radio communication, and a *Baseband Unit (BBU)* for mobile data processing. The BBUs are further detached from the RRHs and hosted in centralized *BBU pools* [23]. The RRHs and BBU pools are usually connected via high speed optical fiber [23]. By clustering RRHs with *complementary traffic patterns* to a BBU, the data processing capacity in the BBU can be shared among RRHs in different time periods, and thus increasing the capacity utility of the BBU [77]. Furthermore, the required capacity of the BBU is expected to be smaller than the sum of capacities of single base stations, leading to a decrease in deployment cost. For example, in Figure 4.1, if we cluster the RRHs in the business district (blue) and in the residential area (red) to a BBU, the aggregated traffic pattern will become relatively stable and the BBU will have a higher capacity utility (Figure 4.1(b)). Meanwhile, the capacity required for the BBU can be reduced from the sum of the two peaks ($1.50 = 0.65 + 0.85$) to a lower aggregated value (1.10). In summary, by pooling BBUs from multiple base stations into a centralized BBU pool, the *statistical multiplexing gain* [23] can be achieved in the cloud-RAN architecture [26].

In order to unlock the power of the Cloud-RAN architecture, it is of great importance to characterize the traffic patterns of RRHs, and to cluster complementary RRHs to a set of BBUs [77,91], so as to maximize the capacity utility and minimize the deployment cost. However, since the data traffic generated in the RRHs are *highly dynamic* over different time and locations, accurately foreseeing and characterizing the RRH traffic patterns in advance is quite challenging, hindering the optimization of RRH clustering and BBU mapping. More specifically, given a set of RRHs in a city, we need to accurately foresee their data traffic patterns in a future period of time (e.g., one day), and find optimal schemes to cluster RRHs with complementary traffic patterns, and map them to a set of BBUs. In order to achieve these goals, we need to address the following issues:

1. **How to foresee the RRH traffic for a future period of time?** The data traffic in each RRH can vary significantly, depending on the impacts of temporal contexts (e.g., weekdays or weekends), human mobility, and social events, etc. Moreover, the data traffic of RRHs located in similar functional areas may demonstrate potential correlations. For example, during weekdays, the RRHs located in business districts usually observe data traffic peaks during working hours, and low data traffic volumes at nights. Capturing the *hidden temporal dependency and spatial correlation* among RRH traffic patterns is not trivial using state-of-the-art time series models, such as ARIMA [94] or neural networks [97]. Therefore, we need to foster more effective techniques for accurate RRH traffic pattern forecasting.

2. **How to measure the complementarity among RRHs?** In order effectively to share and reuse the capacity of a BBU mapped to a cluster of RRHs, the traffic peaks of the RRHs in the cluster should be scattered temporally (i.e., occur at different hours). Meanwhile, to make full use of the BBU mapped to a cluster and avoid BBU overloading, the aggregated cluster traffic should be close to the BBU capacity to a maximal extent, while not exceed the BBU capacity too much. Therefore, we need to take into account both aspects, i.e., *the peak distribution and the capacity utility*, to design an effective metric to measure the

complementarity of RRHs.

3. **How to optimally cluster complementary RRHs into BBUs?** Given the traffic forecast and the complementarity measurements of RRHs, there are potentially enormous numbers of schemes to cluster these RRHs and map them to BBUs in a pool. The optimal scheme not only needs to maximize the average BBU capacity utility, but also needs to minimize the overall deployment cost. Moreover, in order to support fast handover and content offloading between neighboring RRHs [23, 37], the *distances* among a cluster of RRHs should be constrained within a reasonable range. Therefore, we need to design an effective algorithm to find the optimal RRH clustering scheme under the *distance constraint*.

With the above-mentioned objectives and issues, the **main contributions** of this work are:

- We propose a *deep-learning*-based approach to accurately foresee RRH traffic patterns for a future period of time. The proposed approach are capable of modeling the temporal dependency and spatial correlation among the RRH data traffic, and accurately forecasting the future traffic pattern based on historical observations.
- We propose a two-phase framework to dynamically find optimal RRH clustering and BBU mapping schemes under different contexts. In the first phase, we forecast the traffic patterns of RRHs leveraging the proposed MuLSTM model, and propose an *entropy-based metric* to characterize the *complementarity* of RRHs, taking into account both the peak distribution and capacity utility. In the second phase, we build a *weighted graph* to model the complementarity of RRHs, and propose a *distance-constrained clustering algorithm* to find optimal RRH clustering schemes with the objectives of both capacity utility and deployment cost.
- We evaluate the performance of our method using datasets in two months from real-world mobile networks in Milan and Trentino, Italy. Results show that our method effectively increases the average capacity utility to 83.4% and 76.7%, and reduces the overall deployment cost to 48.4% and 51.7% of the traditional RAN architecture in the two datasets, respectively, which consistently outperforms the state-of-the-art baseline methods.

4.2 Preliminaries and Framework

4.2.1 Preliminaries

In mobile network architectures, a set of base stations are deployed over geographical areas called cells [145]. Each base station provides the cell with the network coverage which can be used for transmission of voice and data. With the recent emergence of smartphones and tablets, the data traffic generated from users connected to the RRHs is increasing rapidly [19, 21].

In order to benchmark the data processing capacity of base stations, many operators have collected large scales of RRH traffic statistics data and make them publicly available [141]. In this work, we exploit the dataset released by Telecom Italia for the Big Data Challenge initiative [86]. We extract two months of network traffic data from 11/01/2013 to 12/31/2013 in the city of Milan, Italy and the province of Trentino, Italy. We also collect the locations of active base stations in Milan and Trentino during the two months from CellMapper.net[1], and derive the traffic volume of each base station during the two months on an hourly basis. The traffic data pre-processing steps will be detailed in the evaluation section.

In this work, we consider a Cloud-RAN architecture with one BBU pool for the city-wide mobile network. The benefits of adopting such a centralized pool are two-fold. First, the deployment cost and energy consumption can be greatly reduced by employing data center virtualization technologies [146]. Second, the handover handing and contents offloading among RRHs can be processed internally in the pool, which significantly reduces delays and increases throughput [23]. BBUs in the pool are implemented as virtual machines with specific predefined capacities. In this work, for fair of comparison and simplicity, we assume the BBU capacity to be *fixed and equal to* the on-site BBUs in the traditional architecture.

4.2.2 Framework Overview

We propose a two-phase framework to dynamically cluster complementary RRHs to a set of BBUs, so that the BBU capacity utility and the deployment cost of the entire network can be optimized. As shown in Figure 6.2, in the dynamic RRH profiling phase, given a set of RRHs at a time point, we first propose a deep-learning-based approach to forecast the traffic patterns of RRHs in a future period of time based on their historical traffic data, and then calculate the complementarity of RRHs using a proposed entropy-based metric. In the dynamic RRH clustering phase, we first build a graph model to represent the complementarity among RRHs, and then propose a distance-constrained clustering algorithm to cluster RRHs with complementary traffic patterns. We elaborate on the details of this framework in the following sections.

4.3 Dynamic RRH Profiling

In order to cluster RRHs with complementary traffic patterns to a BBU, we need to be able to forecast the traffic pattern of each RRH in a future period of time. Since the traffic of RRHs vary significantly and exhibit potential spatial correlations, we propose a deep-learning-based approach to model the spatial-temporal dynamics and to forecast the future traffic pattern accurately. Based

[1] https://www.cellmapper.net/map

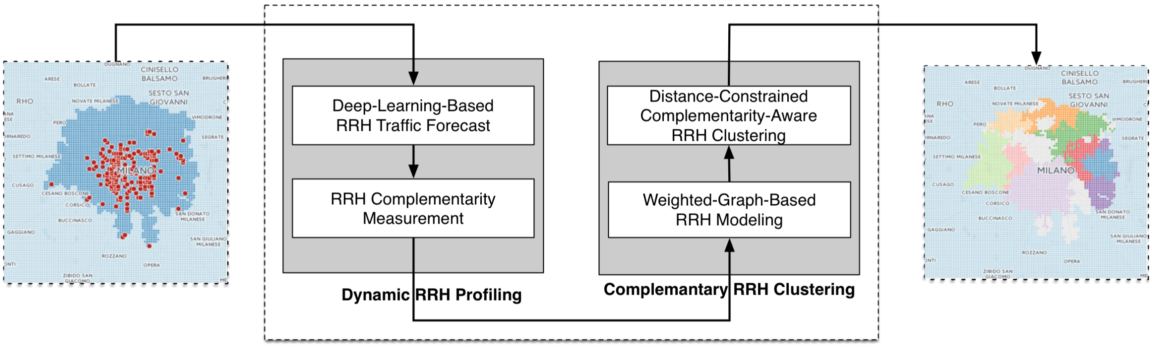


Figure 4.2: Framework overview.

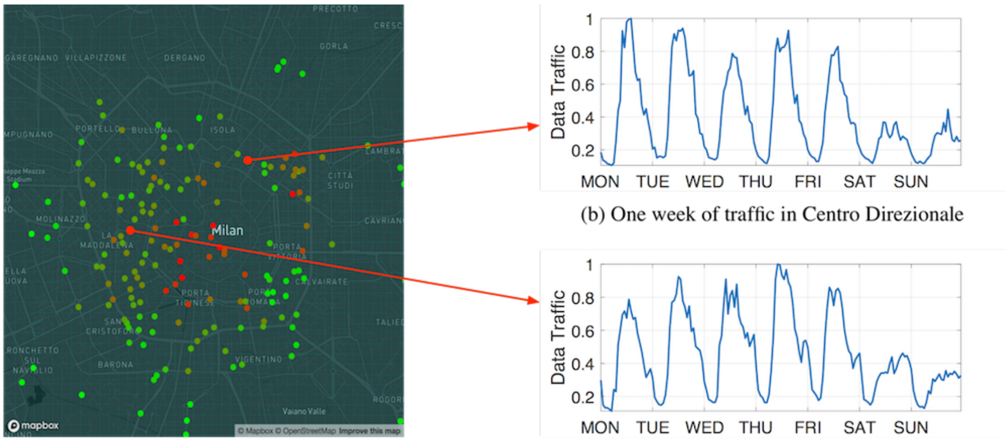
on the traffic forecast, we dynamically characterize the complementarity of RRHs, focusing on the peak distribution and capacity utility of a cluster of RRHs, and design an entropy-based metric to characterize their complementarity.

4.3.1 RRH Traffic Forecasting

Based on the historical traffic data, we observe that the traffic patterns of RRHs are highly dynamic under different temporal contexts. For example, Figure 6.5(b) shows the traffic patterns of two RRHs located in two business districts in Milan during one week, respectively. We observe significant traffic peaks during the working hours of weekdays, and low capacity utility during off-work hours. Moreover, we observe that the traffic patterns of RRHs located in similar functional areas usually demonstrate similar trends. For example, in Figure 6.5(b), the traffic patterns in the two business districts of Milan show similar weekday-weekend patterns.

Basic Idea

In order to accurately forecast the traffic patterns of the RRHs in a future period of time, we need to be able to effectively capture their temporal dependency and spatial correlation. However, this is not trivial using the state-of-the-art techniques. In this work, we propose a *deep-learning*-based approach for our problem. More specifically, we exploit the Recurrent Neural Network (RNN) to automatically capture the intrinsic temporal dependency in our traffic data. An RNN is a special type of neural network designed for sequential pattern mining problems [147]. Built upon the windowed-ANN architecture, an RNN features additional loops to the neurons in the layers of the neural network. Each neuron may pass its signal laterally in addition to forward to the next layer, and consequently, the output of the network for a window may feedback as an input to the network for the next window. Such *recurrent connections* add state or memory to the windowed-ANN



(b) One week of traffic in Centro Direzionale

(a) Base stations in Milan (11/01/2013 - 12/31/2013)

(c) One week of traffic in Piazza Wagner

Figure 4.3: The locations of base stations in Milan and two of the illustrative examples of traffic patterns observed in two business districts from 11/25/2013 to 12/01/2013. Red color denotes high average traffic volume and green color corresponds to low average traffic volume.

architecture and allow it to learn and harness the intrinsic temporal dependency in the time series.

Unfortunately, training an RNN effectively is technically challenging due to the *vanishing or exploding gradient problem*, i.e., the weights in the training procedure quickly became so small as to have no effect (vanishing gradients) or so large as to result in very large changes (exploding gradients). To overcome this problem, researchers proposed the Long Short-Term Memory Network (LSTM) model [148], which introduces the concepts of memory cells and forget gates to generate consistent data flow between the layers of the network and keep the weights stable [149]. In this work, we exploit the LSTM model to effectively learn the temporal dependency of our traffic data.

The other challenge is to model the *spatial correlation* between RRHs in the network. The above-mentioned approaches typically model the traffic of each RRH as a separate time series, making it difficult to capture the correlation between RRHs. In this work, we propose a multivariate-Long Short-Term Memory Network (MuLSTM) approach to model the RRH traffic in a city in a unified model, putting each RRH traffic as a sequence for training and forecasting, and consequently learn the spatial correlation between RRHs.

The MuLSTM Model

Before introducing the MuLSTM model, we define several important terminologies as follows:

Definition 7. ***Remote Radio Head (RRH)****: The RRHs in a city-wide mobile network can be de-*

scribed as a set of points denoted by the following 3-tuple:

$$\{r|r = (rid, lat, lng)\}$$

where rid, lat, lng are the unique ID, latitude, and longitude of the RRH.

Definition 8. ***RRH Traffic**: The mobile data traffic collected from each RRH can be denoted by a set of fixed-length sequences:*

$$\{f|f_i = [u_i(1), \dots, u_i(t), \dots, u_i(N_t)]\}$$

where $u_i(t)$ is the traffic volume of RRH i in time span $t(1 \leq t \leq N_t)$. In this work, we use an one hour time span.

With the collected traffic data, we first organize the collected RRH traffic into a matrix $F^{R_{N_t} \times R_{N_r}}$, where N_t is the number of time spans, and N_r denotes the number of RRHs in the network. We denote the traffic of RRHs we have observed until time t as $F([0,t],:)$, and the traffic of RRHs we would like to forecast in a future period of time Δt as $F([t, t+\Delta t],:)$. In this work, to simplify the implementation, we use one hour time span, and $\Delta t = 24$ hours with $t \mod 24 = 0$, i.e., we forecast the hourly traffic of RRHs for the next day at the end of each day, and dynamically update the RRH clustering scheme based on the forecast. Based upon this, we generate a set of *traffic snapshots* from the traffic matrix, which is defined as follows.

Definition 9. ***RRH Traffic Snapshot**: A traffic snapshot is defined as a matrix F_i, which corresponds to the traffic of all the RRHs during a given period of time Δt, i.e.,*

$$\mathscr{F} = \{F_i|F_i = F([(i-1)*\Delta t, i*\Delta t],:),\ i = 1,2,\dots\}$$

In order to make traffic forecast, we train a *sequence to sequence model* [147] leveraging a unified multivariate LSTM model. During each forecasting, the model accepts F_i as input and outputs F_{i+1}. Note that such a model is called a *many-to-many sequential* model because both the input and output contain Δt time spans, and the order of the time spans play an important role in shaping the model's inner structure. Moreover, the traffic of RRHs are input to the model as *multivariate features* simultaneously, which enables the model to learn the spatial correlation between RRHs.

Finally, we elaborate on the design of the MuLSTM network structure. In general, the MuLSTM model follows the encoder-decoder structure by stacking two LSTM layers L_1 and L_2. The encoder L_1 accepts a snapshot of size $[\Delta t, N_r]$, learns the temporal and spatial structures in the snapshot, and passes the encoded sequences to the decoder. The decoder then makes forecast for a future snapshot of size $[\Delta t, N_r]$ based on the learned structures. The model is trained using the popular Backpropagation Through Time (BPTT) algorithm for multiple iterations. We elaborate the details of the model parameters in the evaluation section.

4.3.2 RRH Complementarity Measurement

Once we have the traffic snapshot forecast for the next day, we are able to evaluate the complementarity of RRHs in that context, and cluster complementary RRHs to a BBU. We consider the following two aspects to design an effective complementarity metric of RRHs.

Peak Distribution

The peak traffic volume of a set of RRHs clustered to the same BBU should be scattered in different temporal contexts, so that the capacity of the BBU can be shared among these RRHs. To this end, we design an entropy-based metric to measure the peak distribution of a set of RRH. Specifically, given a set of clustered RRHs $C = \{r_1, \ldots, r_n\}$, we first find the peak hours in their traffic profiles, respectively, i.e.,

$$T(r_i) = \{t_{i_1}, t_{i_2}, \ldots, t_{i_m}\}, \quad 1 \leq i_m \leq 24 \tag{4.3.1}$$

where t_{i_m} denotes the m_{th} peak time of r_i. Then, we calculate the *Shannon entropy* [150] of the peak hours of the set of clustered RRHs $T(C) = \cup T(r_i)$ as follows:

$$H(C) = -\sum_{k=1}^{K} p_k \log p_k \tag{4.3.2}$$

where $K = |T(C)|$ corresponds to the total quantity of peaks in C, and p_k is the probability of observing the corresponding peak hour in the set $T(C)$. A larger entropy value of a RRH cluster indicates that the RRHs are more complementary in the cluster w.r.t. traffic patterns.

Capacity Utility

To make full use of the BBU mapped to a cluster C, the aggregated cluster traffic should be close to the BBU capacity in different hours of the day. Meanwhile, to prevent the BBU from overload, the aggregated cluster traffic should not exceed the BBU capacity too much. To this end, we design the following metric to quantitatively measure the capacity utility of a BBU B mapped to a cluster C:

$$U(C) = (\frac{\text{mean } \mathbf{f}(C)}{|B|})^{-ln \frac{\text{mean } \mathbf{f}(C)}{|B|}} \tag{4.3.3}$$

where $\mathbf{f}(C) = \sum_{i=1}^{n} \mathbf{f}(r_i)$ denotes the aggregated traffic profile of the RRH cluster, and $|B|$ is the fixed BBU capacity measured in traffic volume. Figure 4.4 shows the illustrative curve of the capacity utility function, which achieves its maximal when the mean aggregated traffic volume is equal to the BBU capacity.

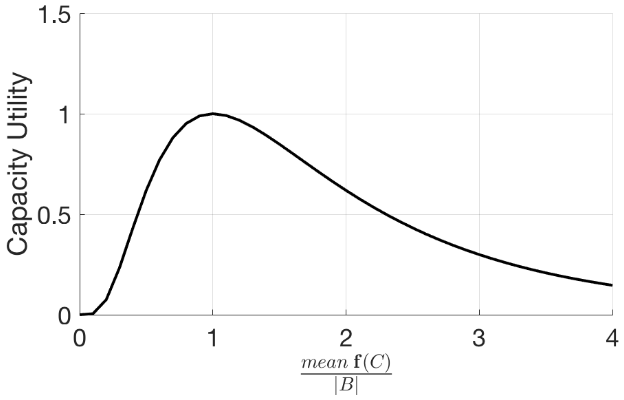


Figure 4.4: The curve of the designed capacity utility function, which reaches its maximal when the cluster traffic volume equals the BBU capacity.

Finally, we calculate the complementarity of the RRH cluster C as follows:

$$M(C) = U(C) * H(C) \tag{4.3.4}$$

$$= -(\frac{\text{mean}\,\mathbf{f}(C)}{|B|})^{-ln\frac{\text{mean}\,\mathbf{f}(C)}{|B|}} \sum_{k=1}^{K} p_k \log p_k \tag{4.3.5}$$

4.4 Complementary RRH Clustering

In this phase, our objective is to cluster RRHs with complementary traffic patterns to a set of BBUs in a pool. One intuitive method is to exhaustively search for RRHs with complementary traffic patterns and iteratively cluster them. However, since there are a tremendous number of clustering schemes, such a method can be computationally intractable as the network scale increases. Moreover, the distance between RRHs and BBU pool should also be constrained within a range, since the propagation delay between RRHs and BBU pool may exceed quality-of-service requirements as distance increases, and we also need to enable machine to machine communications between RRHs such as handover [81] in the mobile network.

Therefore, we propose a graph-model-based algorithm to effectively cluster neighboring RRHs to the same BBU under distance-constraints. First, we construct a weighted graph model to represent the relationship of RRHs, exploiting graph links to express the RRH distance constraints, and link weights to characterize the RRH complementarity measurement. Then, we propose a community-detection-based algorithm to iteratively cluster RRHs into clusters, so that the complementarity of RRHs is maximized within each cluster and minimized across different clusters.

4.4.1 Weighted-Graph-Based RRH Modeling

We model the complementarity among RRHs as an undirected, weighted graph $G = (V,E)$, where $V = \{r_1,\dots,r_N\}$ denotes the set of N RRHs, and E denotes the set of links between two RRHs.

We then define the adjacency matrix A of graph G, which is an $N \times N$ symmetric matrix with entries $a_{i,j} = 1$ when there is a link between RRH r_i and RRH r_j, and $a_{i,j} = 0$ otherwise ($i, j = 1,\dots,N$). We use the geographic distance of two RRHs to determine whether they are adjacent or not. More specifically, for RRH r_i and RRH r_j, we define:

$$a_{i,j} = \begin{cases} 1, & \text{if } dist(r_i,r_j) \leq \tau \\ 0, & \text{otherwise} \end{cases} \tag{4.4.6}$$

where $dist(r_i,r_j)$ is the geographic distance between the two RRHs, and τ is a *neighborhood threshold* controlling the geographic distance of neighboring RRHs.

Given two neighboring RRHs, we use their complementarity measurement to determine their link weight, i.e.,

$$w(r_i,r_j) = M(\{r_i,r_j\}) * a_{i,j} \tag{4.4.7}$$

We consider the case of normalized symmetric positive weights ($w(r_i,r_j) \in [0,1]$) with no loops ($w(r_i,r_i) = 0$). We note that $w(r_i,r_j) = 0$ when there is no link between r_i and r_j ($a_{i,j} = 0$).

4.4.2 Distance-Constrained RRH Clustering

In this step, we need to cluster RRHs to a BBU, so that each cluster consists of neighboring RRHs with complementary traffic patterns. As the link weight of graph G encodes the complementary of RRHs, we need to cluster RRHs with high link weights together, which can be identified as a community detection problem [128].

Problem: Given graph $G = (V,E)$, we first define a set of clusters $\mathbb{P} = \{C_1,\dots,C_K\}$, where

$$\cup_{\forall C_k \in \mathbb{P}} = V \quad \text{and} \quad \cap_{\forall C_k \in \mathbb{P}} = \emptyset \tag{4.4.8}$$

Then, given a RRH v, we define the *connectivity* of v to a cluster C as the sum of link weights between v and the RRHs in the cluster C:

$$con(v,C) = \textstyle\sum_{v' \in C} w_{v,v'} \tag{4.4.9}$$

Finally, we define the *adjacent clusters* $\mathbb{C}(v)$ of v as

$$\mathbb{C}(v) = \{C | con(v,C) > 0, C \in \mathbb{P}\} \tag{4.4.10}$$

With the above definition, our objective is to find an optimal set of clusters $\mathbb{P}$, so that the internal connectivity within a cluster is higher than the inter-cluster connectivity, i.e.,

$$\forall v \in C_k,\ con(v,C_k) \geq max\{con(v,C_l), C_l \in \mathbb{P}\} \tag{4.4.11}$$

We also need to bound the distance span of a cluster within the neighborhood threshold, i.e.,

$$\forall v, v' \in C_k,\ dist(v,v') \leq \tau \tag{4.4.12}$$

Solution: Based on the label propagation concept [91,127], we propose a *Distance-Constrained Complementarity-Aware (DCCA)* algorithm to cluster RRHs. The basic idea of DCCA is iteratively assigning RRHs to the adjacent clusters, where the *gain* of assigning RRH v to cluster C is iteratively evaluated by a *value function* as follows:

$$value(v,C) = con(v,C) \times log\Big(\frac{\tau}{max\{dist(v,v')\}}\Big) \tag{4.4.13}$$

The DCCA algorithm greedily assigns the RRHs to the adjacent cluster with highest value[2] until none of the RRHs are moved among clusters [127]. As the convergence of such a greedy approach is difficult to prove, we set a maximum iteration number *max_iter* to ensure the algorithm will stop.

Algorithm: The DCCA algorithm is initialized by assigning each RRH in the graph to a unique cluster label. In each iteration, we randomly populate a list of RRH $\mathscr{L}$, and traverse the list to update the cluster label of each RRH. The label update process is as follows. First, we remove the RRH from its current cluster, and find the set of adjacent clusters to the current RRH. Then, we compute the value function for all the adjacent clusters, and assign the RRH to the cluster with the highest value. We mark the the RRH as *moved* among clusters if its new cluster label is different from the old one. After we finish iterating over the RRH list, we decide whether to perform another iteration or finish the algorithm based on the following stop criteria: (1) the specified maximum iteration number *max_iter* is reached, or (2) none of the RRH are moved among clusters.

4.5 Evaluation

In this section, based on a real-world mobile network traffic dataset, we evaluate the performance of our framework by assessing its ability to reduce deployment cost and energy consumption. We first describe the experiment settings, and then present the evaluation results and case studies.

[2] If two clusters yield the same value, we randomly choose one.

Table 4.1: Dataset Description

Item	**Milan**	**Trentino**
# Grids	10,000	11,466
Grid size	55,225 m^2	1000,000 m^2
# RRH	182	522
# Covered grids	2,918	2,035
Average coverage	885,420 m^2	3,932,950 m^2
Average traffic volume	0.19	0.13
Data collection period	11/01/2013–12/31/2013	

4.5.1 Experiment Settings

Datasets

The Telecom Italia Big Data Challenge dataset [86] contains two months of network traffic data from 11/01/2013 to 12/31/2013 in Milan and Trentino, Italy, respectively. The city of Milan is partitioned into 100×100 grids with grid size of about 235×235 square meters, while the province of Trentino is partitioned into 117×98 grids with grid size of about $1,000 \times 1,000$ square meters. In each grid, the traffic volume is recorded on an hourly basis. We compile a base station dataset from CellMapper.net, which consists of the locations and coverage areas of active base stations observed in the two months. Based on the location and coverage of each base station, we find the corresponding covered grids and calculate their traffic volume. Finally, we normalize the traffic volumes of each base station to the $[0,1]$ range for the convenience of analytics. The details of these two datasets are listed in Table 4.1.

BBU Capacity

We determine the BBU capacity based on the normalized traffic volume. For the traditional architecture, we assume that each RRH is equipped with a on-site BBU with a capacity of one normalized traffic volume. In this way, the traffic in each RRH can be covered by the BBU. We define the capacity of the on-site as a *capacity unit*. For the cloud-RAN architecture, we assume that the BBUs in the pool (pool BBU) are of the same size, and the capacity is of Q ($Q = 1,2,\dots$) capacity unit, so that the traffic of a cluster of RRHs traffic can be handled in a BBU without causing significant overload. In this work, based on a series of empirical experiments, we choose $Q = 8$ for the city of Milan, and $Q = 10$ for the province of $Trentino$, respectively.

Evaluation Plan

Based on the collected datasets, we map the grids to the coverage areas of RRHs, and aggregate the traffic data to the corresponding RRHs on an hourly basis. We then generate a set of 61 daily traffic snapshots $\mathscr{F}$, each containing the 24 hours' traffic for all the 182 RRHs. We use the snapshots of the first 70% as the training set $\mathscr{F}_{train}$, and the snapshots of the remaining 30% as the test set $\mathscr{F}_{test}$. For the test set, we calculate the complementarity of RRHs based on the traffic forecast, and construct a graph of 182 nodes with the corresponding link structure based on the complementarity metrics. Finally, we perform the DCCA algorithm to cluster the complementary RRHs to a set of BBUs in a centralized pool.

Model Specification

We construct a MuLSTM model with two stacked LSTM layers. The encoder layer L_1 contains $N_{encoder}$ memory units, which accepts a traffic snapshot of shape $[24, 182]$ as input, and outputs an encoded sequence for the decoder. The decoder contains $N_{decoder}$ memory units, which accepts the encoded sequence as input and outputs the forecast of the traffic snapshot. We train the network with the training set $\mathscr{F}_{train}$ for N_{iter} iterations to ensure that the network learns the potential temporal and spatial structures.

Model Training

We use the popular Tensorflow [151] library for constructing our deep-learning model. Based on a series of empirical experiments, we choose the optimal $N_{encoder} = N_{decoder} = 32$, and $N_{iter} = 10,000$. The model is trained on a 64-bit server with an NIVIDA GeForce GTX 1080 graphic card and 16GB of RAM. Each training iteration takes about 3 seconds and the whole process takes 8.3 hours.

Evaluation Metrics

We design the following evaluation metrics to evaluate the RRH traffic forecasting phase and the RRH clustering phase respectively.

(1) For the RRH traffic forecasting phase, we compare the traffic snapshot forecast $\hat{F_i}$ with the ground truth data F_i in the test set, and calculate the Mean Absolute Error (MAE) for each snapshot:

$$MAE(F_i, \hat{F_i}) = \frac{\sum_{t=1}^{N_t} \sum_{r=1}^{N_r} |F_i(t,r) - \hat{F_i}(t,r)|}{N_r \times N_t}$$

(2) For the RRH clustering phase, we quantitatively measure the statistical multiplexing gain from two aspects, i.e., the increase of *average capacity utility* and the decrease of *overall deploy-*

ment cost, compared with the on-site BBUs in traditional architecture. In order to measure the capacity utility of a clustering scheme $\mathbb{P} = \{C_1, \dots, C_K\}$, we derive the following metric based on Equation 4.3.3, i.e.,

$$Utility(\mathbb{P}) = \text{mean}_{C_k} U(C_k) \tag{4.5.14}$$

based upon this, we calculate the average capacity utility of the test set. In order to to measure the overall deployment cost, we sum up the total BBU capacity units required in the pool for a clustering scheme $\mathbb{P}$, i.e.,

$$Cost(\mathbb{P}) = \sum_{k=1}^{K} |\{C_k\}| \tag{4.5.15}$$

based upon this, we use the maximal quantity of capacity units measured in the test set as the overall deployment cost required in the pool.

Baseline Methods

We design the following baseline methods for comparison.

- **Traditional**: In the traditional architecture, one RRH is equipped with one on-site BBU with one capacity unit. The traffic forecast and RRH clustering is not necessary and thus not performed.
- **ARIMA-DCCA**: This baseline method uses the traditional ARIMA model for RRH traffic forecasting, one RRH at a time, and then use the proposed GCLP algorithm for RRH clustering.
- **WANN-DCCA**: This baseline method uses a windowed-ANN model for RRH traffic forecasting, which inputs a traffic snapshot for a day and outputs a traffic snapshot for the next day. The RRH clustering algorithm is the same as the proposed method.
- **MuLSTM-DC**: This baseline method uses the proposed MuLSTM model for RRH traffic forecasting, and then employs a *distance-constrained (DC)* clustering algorithm that clusters neighboring RRHs without considering their traffic complementarity. The clustering steps are similar to the propose DCCA method.

4.5.2 Evaluation Results

Overall Results

Table 6.2 shows the overall evaluation results of the proposed method as well as the baseline methods. For the RRH traffic forecast accuracy, we can see that the proposed Mu-LSTM model achieves the lowest mean absolute error score (0.074 in Milan and 0.083 in Trentino) compared with the two

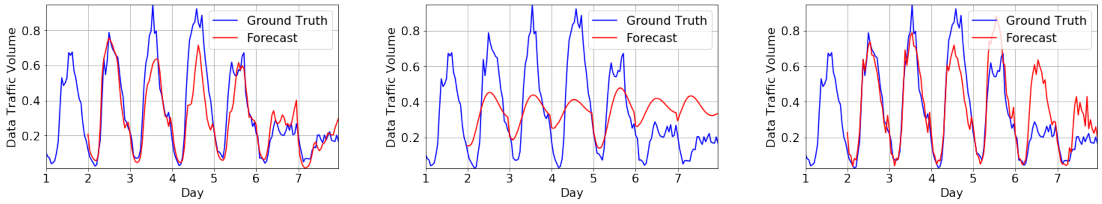
Table 4.2: Evaluation Results

Methods	**Traffic Forecast Error (MAE)**		**Average Capacity Utility**		**Overall Deployment Cost**	
	Milan	Trentino	Milan	Trentino	Milan	Trentino
Traditional	-	-	38.8%	29.4%	182	522
ARIMA-DCCA	0.202	0.237	65.3%	45.2%	112	160
WANN-DCCA	0.175	0.198	73.4%	58.8%	96	120
MuLSTM-DC	0.074	0.083	58.7%	39.2%	120	180
MuLSTM-DCCA	**0.074**	**0.083**	**83.4%**	**76.7%**	**88**	**270**

baselines (ARIMA and WANN), validating its capability of modeling the temporal dependency and spatial correlation of RRH traffic and make accurate forecast. In contrast, the ARIMA method does not capture the spatial correlation among RRHs, while the WANN method is not capable of modeling the intrinsic temporal dependency of RRH traffic patterns. Consequently, the two baselines have higher forecast error rate in both datasets.

For the RRH clustering results, the proposed method consistently achieves the highest average capacity utility (83.4% in Milan and 76.7% in Trentino), as well as the lowest overall deployment cost (88 capacity units in Milan and 270 capacity units in Trentino). Compared with the traditional architecture with on-site BBUs, the clustering schemes increase the average capacity utility rate from 38.8% to 83.4%, and reduce the overall deployment cost from 182 capacity units to 88 capacity units (48.4% of the original cost) in Milan, validating the possibility of achieving significant statistical multiplexing gain though Cloud-RAN optimization. In comparison, the distance-constrained (MuLSTM-DC) clustering baseline does not consider RRH traffic complementarity in the optimization process, and thus are not able to increase capacity utility and decrease deployment cost as effective as the proposed method. Due to inaccurate traffic forecast results, the ARIMA-DCCA and WANN-DCCA baseline methods tend to produce suboptimal clustering schemes and thus achieving lower statistical multiple gain.

We also note that our method performs better in the city of Milan than in the province of Trentino, which can be explained by the geographic characteristic of Trentino. Specifically, Trentino is a mountainous region where cities and villages scatter among valleys. The RRHs are scattered distantly, making it difficult to form complementary RRH clusters in their neighborhoods. In contrast, the metropolitan areas of Milan are larger, more concentrated and more populated, making it easier to form complementary clusters for cloud-RAN optimization.



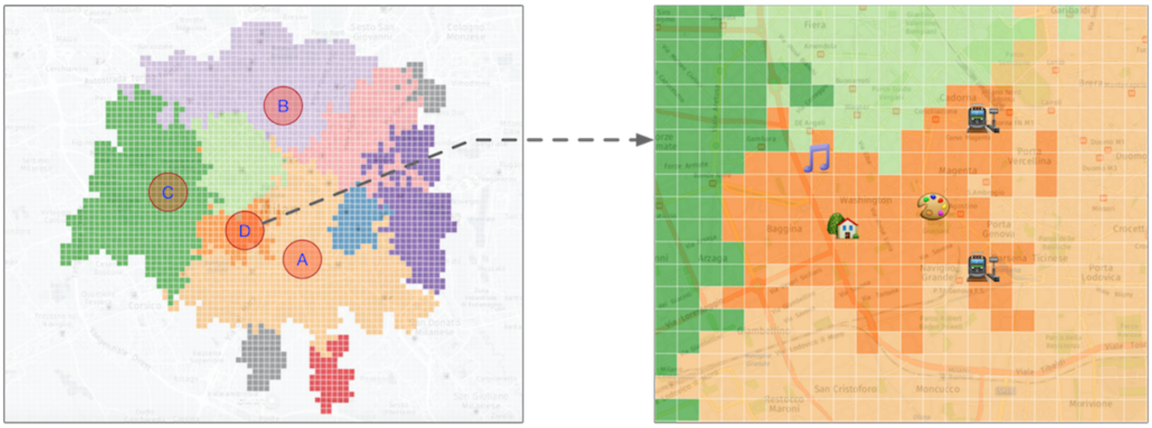
(a) MuLSTM forecast (b) ARIMA forecast (c) WANN forecast

Figure 4.5: RRH traffic forecast results for the base station located in a business district (Centro Direzionale) from 12/25/2013 to 12/01/2013 (one week). The first day traffic is used for input and thus there is not prediction.

Case Studies

We conduct some case studies in Milan to showcase the effectiveness of our method. For RRH traffic forecasting, Figure 4.5 shows an illustrative example of the forecasting results using the proposed MuLSTM method as well as the ARIMA and WANN baseline methods. We can see that our method accurately forecasts the weekday and weekend traffic patterns based on the temporal dependency and spatial correlation it learns from the training set. Instead, the ARIMA method fails to learn the hybrid temporal dependency patterns and outputs the averaged traffic forecast. The WANN method is able to learn some hidden temporal dependency from the single RRH data but is not stable (e.g., on Friday and Saturday).

Figure 4.6 shows the RRH clustering scheme with the proposed method on 2013/11/25 (Monday) in Milan. In general, we obtain 12 RRH clusters, each connected to a BBU in the centralized pool. In Figure 4.6(a), we can see that many clusters (e.g., Cluster A, B, and C) are composed of an urban part and a suburban part, indicating that the traffic patterns in these areas are potentially complementary during a typical weekday. We also note that cluster D is concentrated in a relatively small area, indicating the diverse traffic patterns within this area (Figure 4.6(b)). The reason is probably due to the hybrid functions of this area, which consists of a large residential district (the *Washington neighborhood*), several national museums and theaters (e.g., *Museo Nazionale Scienza e Tecnologia Leonardo da Vinci* and *Teatro Nazionale CheBanca*), and a transportation hub consisting of several train and metro stations (*e.g., Milano Porta Genova* and *Milano Cadorna*). The algorithm is able to identify the RRHs with complementary traffic patterns during the day and effectively cluster them into a BBU to achieve statistical multiplexing gain.



(a) (b)

Figure 4.6: (a) An illustrative example of RRH clustering scheme on 2013/11/25 (Monday) in Milan using the proposed method. Each colored area denotes a cluster and its corresponding coverage area. (b) Cluster D in details: a hybrid area with diverse traffic patterns. Icons denote the featuring city functions in this area.

4.6 Conclusion

In this work, we focus two of the most important objectives in Cloud-RAN optimization to achieve statistical multiplexing gain, i.e., increasing capacity utility and reducing deployment cost. Accordingly, we proposed a deep-learning-based framework to achieve these goals in Cloud-RAN optimization. Specifically, we forecast the traffic patterns of RRHs using a multivariate LSTM model, and then cluster complementary base stations to BBUs based on the traffic patterns. The proposed MuLSTM model is capable of modeling the temporal dependency and spatial correlation between RRHs in the network, and the proposed DCCA clustering algorithm is effective in finding optimal clustering schemes under certain distance constraints, with the objectives of both maximizing the capacity utility and minimizing the deployment cost. Real-world evaluation results in Milan and Trentino show that our framework effectively increases the average capacity utility to 83.4% and 76.7%, and reduces the overall deployment cost to 48.4% and 51.7% of the traditional RAN architecture in the two datasets, respectively, which consistently outperforms the state-of-the-art baseline methods.

In the future, we plan to improve this work in the following directions. Firstly, we plan to explore the variations in the BBU pool, such as considering different sizes of BBU capacity. Secondly, we plan to evaluate our framework in more datasets, and to study the performance of the deep-learning based method under different traffic patterns.

Chapter 5

Spatio-Temporal Mobility Prediction for Anomaly-Aware Road Networks

Contents

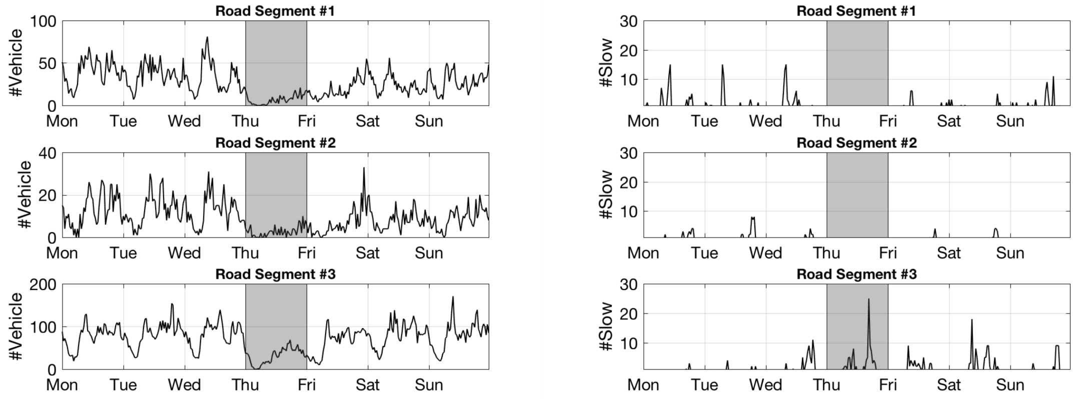
5.1 Introduction

Natural disasters, such as typhoons, hurricanes, and earthquakes, often bring extensive damage to city infrastructures and cause great loss of lives every year. With the rapid population growth and economic development, the cost of natural disasters have been constantly increasing in urban areas [152]. For example, on September 15, 2016, Typhoon Meranti made landfall in Xiamen City, China, leaving more than US$2.6 billion in economic losses[1]. In order to reduce human injuries and prevent further damage after natural disaster strikes, it is important for urban authorities to make efficient disaster response plans and take quick disaster response actions [153]. One of the first steps is to *restore road transportation*, such as cleaning fallen trees, draining ponding waters, and removing crashed vehicles on the road [62, 63]. These *road obstacles* may impede timely search and rescue, evacuation to shelters, and restoration of food and electric supply. Therefore, it is essential for urban authorities to identify and clear road obstacles in a timely manner.

Different strategies have been implemented to identify road obstacles for disaster response, such as sending out investigators to conduct road condition surveys, or reviewing traffic surveillance cameras to detect road obstacles from videos. The ability to accurately report *when, where, and what* types of obstacles are occurring on the road network is critical to the timely restoration of transportation. However, traditional strategies usually consume a great amount of human labor, which is especially expensive in disaster response scenarios, and thus hindering the timely report of road obstacles. Besides, sending out road investigators immediately after the disaster strikes may induce potential human injuries, and traffic surveillance cameras may be destroyed during the disaster, resulting in incomplete road obstacle reporting. Therefore, a *real-time, low-cost, and comprehensive* road obstacle identification method is in great need for disaster response.

Fortunately, with the advance of ubiquitous sensing technologies and paradigms, large amounts of urban sensing data are generated and collected in an unprecedented level [42, 43]. These *cross-domain heterogeneous urban sensing data* provide us with new opportunities to understand road conditions and identify potential road obstacles. In particular, two categories of urban sensing data are highly correlated with road conditions. The first category is *vehicle trajectory data*, which are generated by GPS-equipped vehicles (e.g., taxicabs) running on road surfaces [154]. These vehicles can be viewed as ubiquitous mobile sensors (i.e., *Vehicle-as-a-Sensor*, VaaS) constantly probing road conditions [43]. By analyzing the GPS trajectories of these vehicles, we can identify traffic anomalies potentially caused by road obstacles. For example, when a road segment is blocked by fallen trees, vehicles will not be able to go through it and their trajectories may vary from the normal patterns. The second category is *road environment data*, which describe the spatial and temporal environmental conditions of road segments, such as the road elevation, the roadside trees, and the

[1] https://en.wikipedia.org/wiki/Typhoon_Meranti



(a) Number of vehicles (#Vehicle)

(b) Number of slow mobility behaviors (#Slow)

Figure 5.1: Examples of traffic flow volumes and slow mobility behaviors observed in three different road segments in Xiamen City from September 12, 2016 (Monday) to September 18, 2016 (Saturday). During the landfall of Typhoon Meranti on September 15, 2016 (Thursday), significant decreases of traffic flows are observed in all the three road segments, while only road segment #3 observe a significant increase of slow mobility behavior.

weather conditions [155]. These road environment data can help infer the types of road obstacles after disaster strikes. For example, road segments with flourishing roadside trees may have higher probability of being blocked by fallen trees after a typhoon strike.

Therefore, we propose to leverage the above-mentioned cross-domain urban sensing data for automatic road obstacle detection and classification for disaster response. In the first step, we attempt to detect potential traffic anomalies caused by road obstacles. One intuitive approach is to extract a set of statistical traffic flow parameters (e.g., vehicle number) for each road segment from historical data. Then, one can build anomaly detection models to find significant and unusual decreases in traffic flow, and correspond these anomalies to the potential obstacles in the road segment. However, such a traffic-flow-based approach does not work well in disaster response scenarios. In fact, after a disaster strikes, the number of vehicles running on road usually decreases significantly due to safety concerns, generating abnormal traffic flows in almost every road segments. Consequently, the traffic anomalies caused by potential road obstacles are overwhelmed and thus difficult to detect. For example, in Figure 5.1(a), we observe significant decreases of traffic flow in all of the three road segments in Xiamen City after Typhoon Meranti's landfall, although only the third one is reported with road obstacles (the details are presented in the Xianyue Road case study).

In order to detect road obstacle from vehicle trajectories, we propose a novel approach by exploiting the *slow mobility behaviors* in vehicle trajectories. Based on our observations, when a driver encounters road obstacles, they usually slow down the vehicle, observe the road conditions, and then make a decision to either change direction or slowly bypass the obstacle. Such slow mobility behaviors can be extracted from vehicle GPS trajectories, and be exploited to detect potential road obstacles. For example, in Figure 5.1(b), we observe a significant increase of slow mobility behaviors in road segment #3 after Typhoon Meranti's landfall on Xiamen City, which is induced by fallen trees on the road surface.

Nevertheless, it is still possible that some of the slow mobility behaviors are not induced by road obstacles. For example, traffic lights at intersections may cause vehicles to slow down in a periodical pattern, and road segments with u-turn signs may also observe large number of slow-moving vehicles constantly. These kinds of slow mobility behaviors, however, usually occur *regularly* on specific road segments, demonstrating certain spatio-temporal *patterns*. In contrast, the slow mobility behaviors induced by road obstacles are usually *abnormal* in the given road segment and time span. Moreover, such an anomaly is usually observed in a *collective* way, i.e., impacting a collection of neighboring road segments and lasting for a consecutive period of time. For example, fallen trees in an intersection may cause unexpected slow mobility behaviors in the surrounding roads until they are removed. Therefore, we need to design an effective algorithm to separate the *collective anomalies of slow mobility behaviors* for road obstacle detection.

After detecting the road obstacles, it is essential for urban authorities to identify their types, such as fallen trees, ponding water, and congested vehicles on road. Since different types of road obstacles may induce similar slow mobility behaviors, it is difficult to classify these obstacles merely based on the vehicle trajectory data. Therefore, we propose to involve the road environment data to model the context of the obstacles, and thus inferring their corresponding types. For example, obstacles observed on flourishing road segments after a typhoon strike may probably be classified as fallen trees. To this end, the following challenges need to be addressed:

- *Heterogeneous features*. Due to the considerable variety and volume of road environment data, it is not straightforward to select a set of representative features to model the context of road obstacles. Moreover, how to effectively incorporate these heterogeneous features into a data analytics model is also challenging.
- *Insufficient labels*. In order to train a model for road obstacle classification, we need to collect a set of road obstacle instances as ground truth. However, validating road obstacles is labor-intensive and time-consuming, making it difficult to collect a large enough training set. Therefore, we need to propose an effective road obstacle classification model to learn from these sparse labels.

In this work, we propose a two-phase framework for road obstacle detection and classification. In the first phase, we extract the slow mobility behaviors from vehicle trajectories in each road segment, and build a spatio-temporal matrix to model these slow mobility behaviors in a city-wide level. We then propose a *robust matrix factorization*-based method to separate the collective anomalies from the slow mobility behavior matrix. To ensure that each separated anomalies are collective in neighboring road segments for a consecutive period of time, we incorporate a clustering-based outlier-remover method in the factorization algorithm, and detect road obstacles based on the corresponding collective anomalies. In the second phase, we identify two categories of contextual factors related to road obstacles from various road environment data, i.e., the *spatial contextual features* (e.g., roadside trees, road elevation, and road properties) and the *temporal contextual factors* (e.g., wind, rainfall, and visibility). In order to accurately classify the road obstacle with these heterogeneous features and sparse labels, we propose a semi-supervised learning approach combining co-training [156] with active-learning [157]. More specifically, we first train a spatial classifier and a temporal classifier, respectively, using the corresponding feature categories and a sparse training set. We then iteratively improve the model accuracy by adding the *confident* and *salient* instances in the unlabeled set to the training set, and retrain the model. A *confident* instance is identified if it receives the same label from both classifiers with high classification confidence [158], and a *salient* instance is identified if it is difficult to classify for both classifiers, i.e., receiving different labels with low confidence. We add the confident instances to the training set (co-training), and actively collect the labels of the salient instances from a crowdsensing platform (active learning), and add them back to the training set. Briefly, the contributions of this work include:

- To the best of our knowledge, this is the first work on road obstacle detection and classification for disaster response leveraging cross-domain urban sensing data. By fusing the large-scale vehicle trajectory data with the heterogeneous road environment data, we are able to accurately identify road obstacles for disaster response in a low-cost and automatic manner.

- We propose a two-phase framework to identify road obstacles by leveraging the slow mobility behaviors and road environment contexts. In the detection phase, we exploit a sliding-window based method to extract slow mobility behaviors from vehicle trajectories, and propose a cluster direct robust matrix factorization (CDRMF) approach to detect the collective anomalies induced by road obstacles from the spatio-temporal slow mobility behavior matrix. In the classification phase, we extract two categories of road obstacle contextual features (i.e., spatial-features and temporal-features) from various road environment datasets, and propose a co-training and active learning (CORAL)-based approach to learn an effective classification model using the spatio-temporal features and a sparse training set. The CORAL approach iteratively improves the classification accuracy by adding confident instances to the training set, and actively labeling salient instances using a crowdsensing platform.

- We evaluate our framework on Xiamen City with a large-scale taxi trajectory dataset and various environment sensing datasets. Results show that our framework accurately detects and classifies different types of road obstacles during the 2016 typhoon season, achieving an overall precision and recall both above 90%, and outperforms the state-of-the-art baselines.

5.2 Preliminary and Framework Overview

Definition 10. ***GPS Dataset***: *the vehicle GPS dataset we collect can be described by a set of GPS points denoted by 4-tuples:*

$$P = \{p | p = (v,t,lat,lng)\}$$

where v,t,lat,lng are the vehicle ID, time stamp, latitude, and longitude from GPS transmitters.

Definition 11. ***Vehicle Trajectory***: *we define a vehicle trajectory as a sequence of GPS points $p_1 \to p_2 \to \cdots \to p_n$, where $p_i \in P, 1 \le i \le n$.*

Definition 12. ***Road Segment***: *we partition a city into an $I \times J$ grid map based on the longitude and latitude, and define a road segment r as a grid containing roads for vehicles.*

Definition 13. ***Time Span***: *we divide the duration of observation data into equal time spans t, each time span lasts for a period of time, e.g., half an hour.*

We propose RADAR, a two-phase framework to detect and classify road obstacles for disaster response. As shown in Figure 6.2, we first extract slow mobility behaviors from a large-scale vehicle GPS trajectory dataset with a sliding-window-based method. In the road obstacle detection phase, we organize the slow mobility behaviors into a spatio-temporal matrix with a road segment dimension and a time span dimension. We then perform the CDRMF algorithm on the matrix to extract collective anomalies of slow mobility behaviors, and correspond these anomalies to the potential road obstacles. In order to classify these detected road obstacles, we identify several relevant environment sensing data, and extract a set of spatial and temporal contextual features correspondingly. In the road obstacle classification phase, we exploit the co-training diagram to train a spatial classifier and a temporal classifier, respectively, using the corresponding contextual features and a sparse training set. We iteratively add confident and salient unlabeled instances to the training set to improve the model accuracy, leveraging a crowdsensing-based active learning diagram which actively collect labels for the salient instances from a crowdsensing platform. We elaborate the key steps of the framework in the following sections.

5.3 Slow Mobility Behavior Extraction

The slow mobility behaviors of vehicles may indicate potential obstacles on the road. For example, fallen trees that block a road may force the drivers to slow down and change direction. Such slow

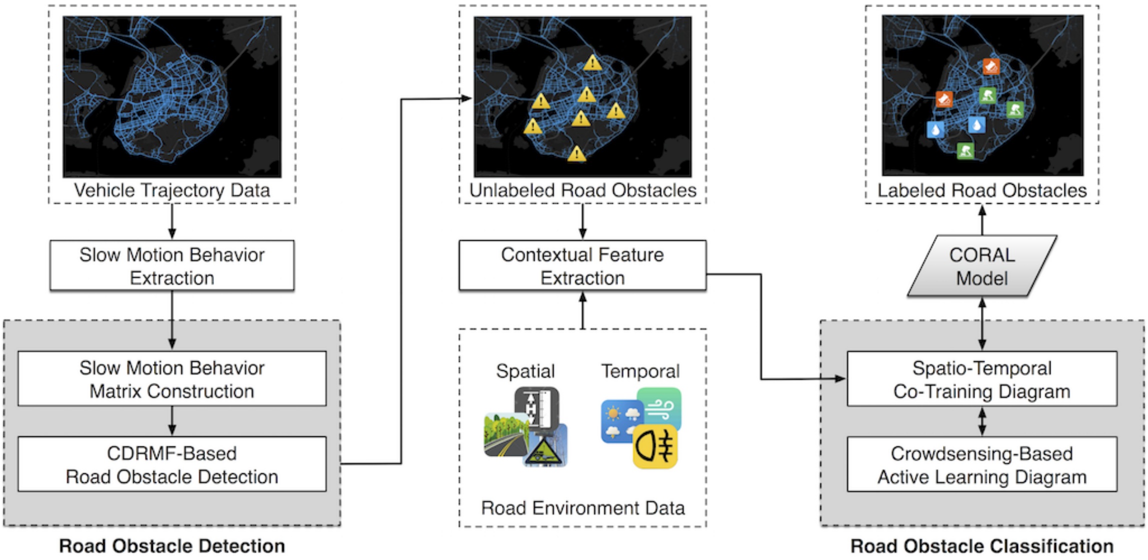

Vehicle Trajectory Data
Slow Motion Behavior Extraction
Slow Motion Behavior Matrix Construction
CDRMF-Based Road Obstacle Detection
Road Obstacle Detection
Unlabeled Road Obstacles
Contextual Feature Extraction
Spatial
Temporal
Road Environment Data
Labeled Road Obstacles
CORAL Model
Spatio-Temporal Co-Training Diagram
Crowdsensing-Based Active Learning Diagram
Road Obstacle Classification

Figure 5.2: Overview of the framework.

mobility behaviors can be captured from vehicles' GPS trajectories if the data points are collected frequently enough. In this work, we use a taxi GPS trajectory dataset collected every one minute. We elaborate on the method to extract slow mobility behaviors from taxi trajectories as follows.

First, we employ an adaptive sliding-window-based method [159] to extract slow mobility sequences from taxi GPS trajectories. More specifically, for a trajectory $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_n$, we extract every slow mobility sequence $p_m \rightarrow p_{m+1} \rightarrow \cdots \rightarrow p_{m+k} (1 \leq m < n, 1 \leq k \leq n-m)$ in which the distance (*dist*) between each pair of adjacent points is less than a threshold δ_p, i.e.,

$$\forall m \leq i < m+k, dist(p_i, p_{i+1}) < \delta_p \tag{5.3.1}$$

We use a sliding-window with adaptive size along the trajectory to find such slow mobility sequences. In particular, we dynamically extend the window size by adding new points until the newly-formed sequence violates requirement 5.3.1. We use an example in Figure 5.3 to elaborate on the process. For the trajectory $p_1 \rightarrow p_2 \rightarrow \cdots \rightarrow p_7$, we start by creating a window consisting of the first two points (p_1, p_2 in this case), and check whether the distance between p_1 and p_2 is less than δ_p. Since $dist(p_1, p_2) > \delta_p$, we discard this window, and slide the window to start over from the end point (p_2), and create a new window (p_2, p_3). We see $dist(p_2, p_3) < \delta_p$ so the window is kept; since $dist(p_3, p_4) < \delta_p$, we extend the window by adding p_4, and repeat this procedure for the next adjacent points until the distance constraint is violated. Finally, we obtain a sequence containing a set of consecutive points $p_2 \rightarrow \cdots \rightarrow p_6$.

We filter out sequences with long-term duration, which may correspond to non-driving be-

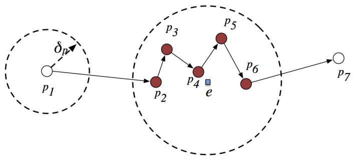


Figure 5.3: Slow mobility sequence extraction from GPS trajectory leveraging an adaptive sliding-window.

haviors such as taxi driver shift or vehicle repair. Finally, we map each slow mobility sequence $p_m \rightarrow p_{m+1} \rightarrow \cdots \rightarrow p_{m+k}$ to a slow mobility behavior, as denoted by a triple:

$$b = (v, r, t)$$

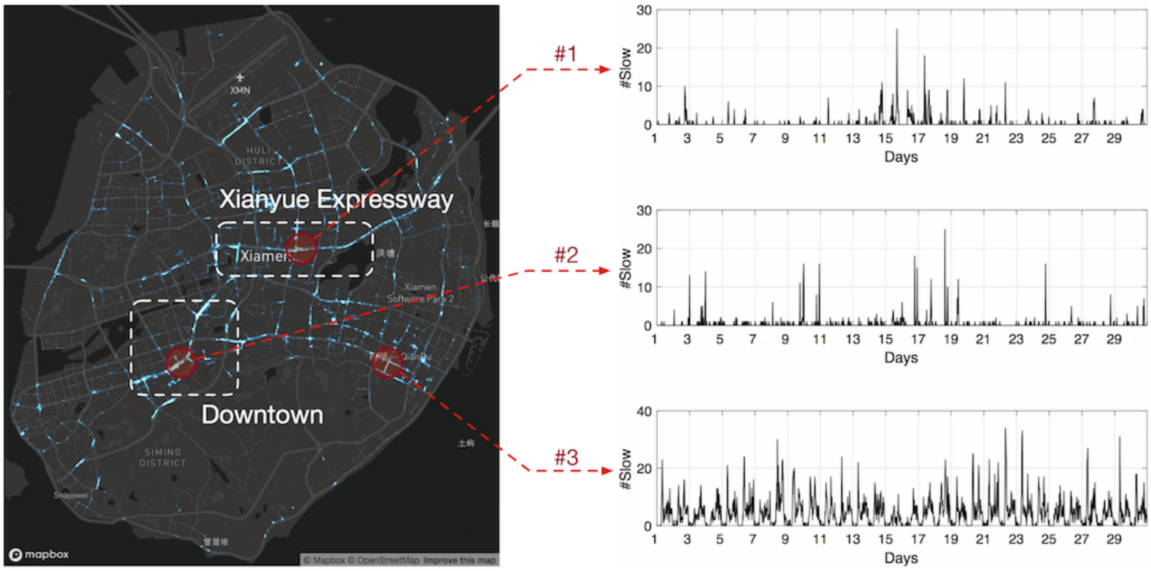
where v is the corresponding vehicle ID. We determine the road segment r by mapping the coordinates of the sequence centroid $[(p_m.lat + p_{m+k}.lat)/2, (p_m.lng + p_{m+k}.lng)/2]$ to the city grid system, and determine the time span t by mapping the middle of the duration $(p_m.t + p_{m+k}.t)/2$ to the time span partition system.

Figure 5.4(a) shows a visualization of the extracted slow mobility behaviors after the landfall of Typhoon Meranti in Xiamen. We observe a cluster of slow mobility behaviors on the major roads of the downtown area, which may be induced by ponding water on the road surface, since the elevation of downtown Xiamen is relatively low[2]. Another cluster of slow mobility behaviors can be found along the Xianyue expressway, which was covered by flourishing trees, and the slow mobility behaviors may be induced by fallen trees during the typhoon landfall.

5.4 Road Obstacle Detection

In this phase, our objective is to detect road obstacles from the extracted slow mobility behaviors. The rationale behind this approach is that when an obstacle is present in a road segment, it may induce a *collective anomaly* [160] of slow mobility behaviors. The meaning of *collectiveness* is two-fold. First, such an anomaly may be observed in a collection of neighboring road segments. Second, such an anomaly may last for a consecutive period of time after the road obstacle is present. For example, when fallen trees block a lane, various slow mobility behaviors, such as turning and bypassing, can be observed in the surrounding road segments and last for a period of time until the road obstacle is removed. Consequently, simply building a time series anomaly detection [161]

[2] http://www.floodmap.net/elevation/ElevationMap/?gi=1790645



(a) Visualization of extracted slow motion behaviors

(b) Temporal variations of three typical road segments

Figure 5.4: An illustrative visualization of the extracted slow mobility behaviors after the landfall of Typhoon Meranti in Xiamen City (2016/09/15 09:00–2016/09/15 17:00), and the temporal variations of slow mobility behaviors in three typical road segments during one month (2016/09/01–2016/09/30). (#1: Xianyue Expressway, #2: Hubin South Road, #3: Qianpu East Road)

model for each road segment to detect road obstacle does not work well in this problem, as the *spatio-temporal collectiveness* is not properly preserved and modeled.

The second challenge is that the slow mobility behaviors we collect are in a *mixed* state due to various causal factors. Based on our observations, slow mobility behaviors can be induced not only by road obstacles, but also by traffic lights and turning signs, rush hour traffic congestions, picking up and dropping off passengers, etc. Directly detecting anomalies from such a mixed state and corresponding them to road obstacles can be very difficult and unreliable [10, 162].

To address these challenges, we propose a robust matrix factorization-based approach to separate the spatio-temporal collective anomalies from the mixed slow mobility behaviors, and correspond them to the road obstacles. First, we build a slow mobility behavior matrix with a road segment dimension and a time span dimension. Each cell of the matrix denotes the number of slow mobility behaviors observed in the specific road segment during the specific time span. Then, we separate the slow mobility behaviors induced by different causal factors based on their spatio-temporal properties. We note that slow mobility behaviors induced by traffic lights and turning signs are usually observed *regularly* in some specific road segments and time spans, while the collective anomalies of slow mobility behaviors induced by road obstacles tend to be *abnormal* events in the spatio-temporal space. Moreover, such collective anomalies are usually observed in geographically *clustered* road segments over a *consecutive* period of time. With these insights, we propose a *Cluster Direct Robust Matrix Factorization (CDRMF)* [163, 164] approach to automatically decompose the mixed slow mobility behavior matrix into a *low-rank* component and a *sparse-and-clustered* component. The low-rank component represents the regular slow mobility behaviors induced by traffic lights and turning signs, etc., while the sparse-and-clustered component corresponds to the unusual and clustered collective anomalies induced by road obstacles. We elaborate on the details of our approach as follows.

5.4.1 Slow Mobility Behavior Matrix Construction

We build a spatio-temporal matrix $M \in \mathbb{R}^{N_r \times N_t}$ with two-dimensions denoting N_r road segments and N_t time spans. Each entry of the matrix $M(r,t)$ denotes the number of slow mobility behaviors observed in road segment r during time span t.

In particular, we analyze the temporal variations of slow mobility behaviors in three typical road segments in Xiamen City during one month (September 2016), as shown in Figure 5.4(b). For road segment #1 (Xianyue Expressway), we observe a significant increase of slow mobility behaviors during Typhoon Meranti's landfall in Xiamen (September 15–17), which corresponds well with the fact that fallen trees induced by strong winds block several lanes in Xianyue Expressway[3]. We also observe several abnormal increases of slow mobility behaviors in road segment #2 (Hubin South Road), which is built in a low elevation area. These anomalies might be correlated with the road surface water ponds induced by heavy rains. As an counter example, we observe regular patterns of slow mobility behaviors on road segment #3 (Qianpu East Road), which is a popular business and activity district in Xiamen City, and thus the regular patterns may correspond to the passenger pick-up and drop-off events instead of road obstacles.

5.4.2 CDRMF-Based Road Obstacle Detection

With the slow mobility behavior matrix M constructed, we then need to separate the regular and anomalous slow mobility behaviors apart. Such a problem can be addressed by matrix decomposition techniques [165] by imposing structural constraints on the decomposed components [166]. In particular, Robust Matrix Factorization (RMF) [163, 164, 166] approaches have been proposed to decompose a mixed matrix into a low-rank part and a sparse part in an automatic manner, and have been widely adopted in robust modeling and anomaly detection [167]. The low rank component can be used to describe the regular patterns of slow mobility behaviors, but the sparse component may have arbitrary structure which does not necessarily correspond to the collective anomalies we

[3] http://weibo.com/1976447603/E8vIamwnv

desire. Therefore, we improve the RMF approach by adding a clustering step to pursue a sparse-and-clustered component that corresponds to the collective anomalies.

Problem: we define our objective as to decompose the mixed matrix M into a low-rank matrix L and a sparse-and-clustered matrix S, i.e.,

$$\begin{aligned} M &= L + S \\ s.t. \quad rank(L) &\leq k, \\ card(S) &\leq c, \\ outlier(S) &\leq \varepsilon \\ L \geq 0, S &\geq 0 \end{aligned} \tag{5.4.2}$$

where $card(S)$ denotes the cardinality of S, i.e., the number of non-zero elements in S. By imposing the constraints on $rank(L)$ and $card(S)$, we pursue a low-rank L and a sparse S, respectively. Besides, we need to make sure that the non-zero elements in S are distributed collectively in neighboring road segments and consecutive time spans. Therefore, we use $outlier(S) \leq \varepsilon$ to prevent outliers that are isolated from their spatio-temporal neighbors, where ε is a relatively small value.

Solution: solving Problem (5.4.2) is not trivial due to its non-convex constraints [168]. Traditionally, such kind of problem is solved using relaxation techniques [167], i.e., by relaxing the matrix rank of L with its nuclear norm [167], and relaxing the cardinality of S using its ℓ_1 norm [169]. The relaxed problem is then solved using alternating minimization techniques [167]. However, the traditional relaxation techniques have several limitations. First, it is difficult to represent and impose the clustered structure of S to the relaxation problem. Second, it is unknown how well the relaxation approximate the original problem in general [168].

In this work, we proposed a Clustered Direct Robust Matrix Factorization (CDRMF) approach for collective anomaly detection, which is built on the recently proposed Direct Robust Matrix Factorization (DRMF) method [168]. Instead of using relaxation techniques, the DRMF approach directly solves the matrix decomposition problem by alternatively optimizing the low-rank component and the sparse component. In order to impose the clustered structure constraint, we improve the DRMF algorithm by iteratively removing the isolated outliers in the sparse component in the optimization process.

Algorithm: the detailed process of CDRMF is described in Algorithm 1. The CDRMF algorithm is initialized with the mixed matrix m, the rank constraint k, and the cardinality constraint c. In each iteration, we perform two steps to update the low-rank component L and the sparse-and-clustered component S. In the first step, we fix and remove the sparse component S from M, and approximate $M - S$ by a low-rank component L. In the second step, we fix and remove the low-rank approximation L from M to obtain the residual $M - L$, and find the optimal sparse-and-clustered S to recover the residual. We repeat the process until the algorithm is converged or the maximum iteration number is reached. Finally, we output the low-rank component L as the regular patterns, and the sparse-and-clustered component S as the collective anomalies.

In order to solve the low-rank approximation problem, we perform Single Value Decomposition (SVD) [170] on $M - S$, and truncate its top-k singular vectors to construct a rank-k approximation $L = M - S$. Since only the first k singular vectors are required, we accelerate the computation using partial SVD algorithms [171]. We propose a two-step approach to solve the sparse-and-clustered optimization problem. First, we find the optimal approximation of the residual $M - L$ under the cardinality constraint $card(S) \leq c$. This can be done directly by copying the top-c largest values in $M - L$ to S and setting the rest entries of S to zeros. The proof of this method can be found in [168] and thus omitted here. Then, we perform a clustering operation for the non-zero elements in S using the DBSCAN algorithm [172], and remove the outliers that are isolated from their spatio-temporal neighbors. In particular, we use the geographic distance between road segments and temporal distance between time spans to determine the search distance in clustering. In this way, we obtain a sparse-and-clustered component S to approximate the residual in the iteration.

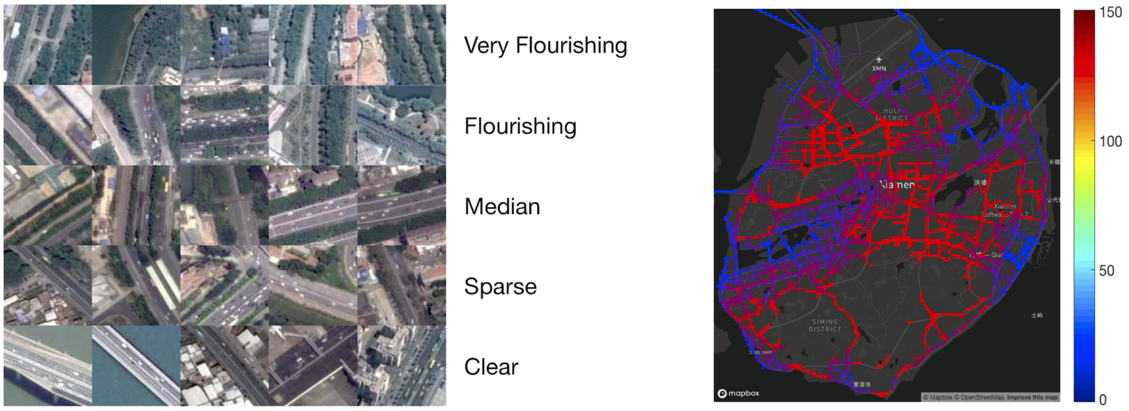
Algorithm 1: The CDRMF algorithm for collective anomaly detection

```
Input:  $M$ the slow mobility behavior matrix
        $k$ the maximum rank
        $c$ the maximum cardinality
        $max\_iter$ the maximum number of iterations
Output: $L$ the low-rank component
        $S$ the sparse-and-clustered component
Initialize: $S \leftarrow \mathbf{0}$
while not converged and iteration < max_iter do
    a) Solve the low-rank approximation problem:
            $L = \arg\min_L ||A - L||_F, \quad A = M - S$
            $s.t. \quad rank(L) \leq k$
    b) Solve the sparse-and-clustered optimization problem:
            $S = \arg\min_S ||B - S||_F, \quad B = M - L$
            $s.t. \quad card(S) \leq c, \quad outlier(S) \leq \varepsilon$
    $error \leftarrow ||M - L - S||_F$
```

5.4.3 Road Obstacle Detection

Finally, we map the detected sparse-and-clustered collective anomalies to the potential road obstacle event. We note that one road obstacle may induce several collective anomalies spanning among neighboring road segments and lasting for a consecutive period of time. In particular, we assign each cluster in S with a label and use the cluster to denote the road obstacle event e , i.e.,

$$e_i = \{(r,t)|S(r,t).label = i\}$$



(a) Roadside tree coverage labeling sample results

(b) Road segment elevation map (in meters)

Figure 5.5: Illustrative results of road environment contextual feature extraction.

5.5 Contextual Feature Extraction

With the road obstacles detected, our next objective is to recognize the types of these obstacles, such as fallen trees and ponding water. However, since different types of obstacles may induce similar slow mobility behaviors, it is rather difficult to distinguish different types of road obstacle merely based on vehicle trajectory data. Therefore, we propose to incorporate the cross-domain environment sensing data to model the context of the road obstacles, and then infer their corresponding types base on the contextual features.

However, due to the considerable variety and volume of these road environment sensing data, it is still not trivial to identify the correlated factors and extract the effective features for road obstacle classification. Therefore, we conduct a series of empirical studies to analyze the correlations between road obstacles and various environment contextual factors, leveraging a set of road obstacle events and road environment datasets collected from Xiamen City. We elaborate the details of analysis as follows.

5.5.1 Spatial Contextual Factors

Based on previous studies and surveys [65, 66], the geographic environment conditions of a road segment can provide strong evidence for inferring the types of potential road obstacles. For example, road segments with flourishing trees may have higher probability of being blocked by fallen trees after strong winds, and low-elevation road segments may be more vulnerable to heavy rains. In particular, we identify the following three key spatial contextual factors and extract a set of features from the corresponding road environment data.

Roadside tree coverage: the conditions of roadside trees can be observed from high-definition satellite images. However, labeling the degree of tree coverage for thousands of road segments requires great human effort. Therefore, we employ a deep learning-based approach [173] to automatically label roadside tree coverage. First, we obtain a set of satellite images I from Google Earth[4], and randomly select a small subset of road segment images I_t as training examples. Then, we manually label the degree of roadside tree coverage for the training examples into five categories, *very flourishing, flourishing, median, sparse, clear*. Some examples of the labeled images can be seen in Figure 5.5(a). Finally, we use a pre-trained deep learning network, AlexNet [174], to extract features from the set of unlabled road segment images $I - I_t$, and predict their corresponding labels using a SVM classifier [175]. Examples of the predicted labels are present in Figure 5.5(a).

Road segment elevation: road segments with low elevation may be vulnerable to heavy rains caused by typhoons and hurricanes. The elevation of road segments can be obtained from various Digital Elevation Model (DEM) [176] data sources with different resolutions. In this work, we extract road segment elevation data from Google Earth, which provides a base resolution of 30 meters [176] in Xiamen City. An extracted elevation map for all the road segments in Xiamen City is shown in Figure 5.5(b).

Road segment properties: road obstacles, such as crashed vehicles caused by traffic accidents and congested vehicles caused by malfunctioning traffic lights, tend to be observed in road segments with complicated conditions, e.g., road intersections, traffic circles, and tunnel entries and exits. We identify these complicated road segments and use this prior knowledge as a feature to infer traffic-induced road obstacles. In particular, we label the road segment properties by the following categories: *intersection, circle, tunnel entry/exit, none*. We retrieve the locations of road interactions and circles from Xiamen Traffic Police, and manually label the locations of tunnel entries and exits.

5.5.2 Temporal Contextual Factors

In a road segment, different types of obstacles may occur under different temporal contexts [177, 178]. For example, fallen trees are usually caused by strong winds, water ponds on road faces are usually formed after heavy rains, and vehicle congestion and accidents are reported more frequently when drivers have limited visibility. By exploiting the meteorological data from the Weather Underground API[5] and Xiamen Meteorological Bureau[6], we identify the following temporal contextual factors and extract a set of corresponding features.

[4] https://www.google.com/earth/

[5] https://www.wunderground.com/weather/api/

[6] http://www.xmqx.gov.cn

Wind speed: since fallen trees are usually observed after strong winds with a delay, we extract the wind speed one hour before for each obstacle as the contextual feature, measured in m/s.

Rain precipitation: similarly, water ponds are usually formed after heavy rains. Therefore we extract the rain precipitation one hour before for each road obstacle as the corresponding contextual feature, measured in centimeters.

Road visibility: visibility may greatly impact driving safety, especially in heavy rains and foggy weather. Limited visibility in a road segment may cause potential risks of traffic accidents and congestions, especially in complicated road segments. We extract fine-grained visibility data in each road segment as the corresponding contextual feature, measured in meters.

5.6 Road Obstacle Classification

In this phase, our objective is to classify the detected road obstacles based on the extracted contextual features. Intuitively, we can train a predictive model (e.g., artificial neural networks) to classify the road obstacles using the various contextual features. However, since the spatial and temporal features are extracted from heterogeneous sources and vary significantly in scales, equally treating these features does not work well in our problem [179]. The other challenge is that obtaining a set of labeled road obstacles large enough for training a predictive model is rather difficult, since road obstacle reporting is labor intensive and time consuming, especially in disaster response scenarios.

In order to address the challenges of heterogeneous features and insufficient labels, we propose a semi-supervised learning approach combining co-training with active-learning (CORAL). Co-training is a multi-view learning technique that leverages two conditionally independent models to predict the target labels, and use the confident prediction results to further improve the model accuracy in an iterative manner [156, 179]. We group the heterogeneous features into two sets, i.e., spatial and temporal feature sets, and input these features into the co-training framework. Active-learning takes another approach to improve prediction accuracy by dynamically selecting a set of uncertain predictions and asking the users to provide labels for these instances, and then it feeds them back to retrain the model iteratively [180, 181]. We propose a crowdsensing-based platform to obtain a relatively small set of training labels, and use the active-learning mechanism to gradually obtain new labels during the training process. The design diagram of the CORAL approach is illustrated in Figure 5.6. We elaborate on the details as follows.

5.6.1 The Co-Training Paradigm

In the co-training paradigm, we feed the two categories of contextual features into a spatial and temporal inference model, respectively, and iteratively add the instances with high prediction con-

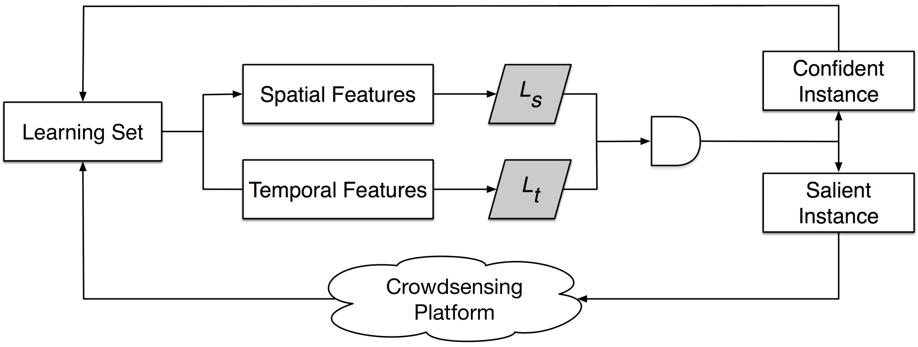


Figure 5.6: The learning diagram of the proposed CORAL model.

fidence to the training set to improve model accuracy.

More specifically, we denote the set of detected road obstacles as $G = \{G_1, G_2\}$, where G_1 is the small initial set of labeled obstacles, and G_2 is the large unlabeled set. We obtain the labels of G_1 by leveraging a crowdsensing-based platform, which is detailed in the evaluation section. Using the set of spatial features F_s and temporal features F_t, we train a spatial learner L_s and a temporal learner L_t respectively, i.e.,

$$L_s \leftarrow L_s.learn(F_s, G_1) \qquad L_t \leftarrow L_t.learn(F_t, G_1) \tag{5.6.3}$$

Afterward, we apply L_s and L_t to each instance of G_2, and select N_c instances that receive the same label from both L_s and L_t with highest confidence, respectively. Finally, we add the selected instances to G_1 as labeled instances. We repeat this process until G_2 is empty or a maximum number of iterations is reached.

5.6.2 The Active Learning Paradigm

Ideally, the co-training paradigm can improve the model accuracy with the confident unlabeled instances. However, this approach does not ensure that the selected confident instances are always valuable for improving the predictive capability of the model [180]. In other words, we can directly select some *salient* instances to the models to improve their accuracy. To this end, we incorporate a active learning process to the co-training diagram.

More specifically, in each iteration, after applying L_s and L_t to each instance of G_2, we further select N_a instances that are considered most difficult to predict for both predictors. In particular, we select instances that receive different labels from L_s and L_t with lowest confidence. We then use the crowdsensing platform to collect the labels for these instances. Finally, we add these salient labeled instances to G_1 and retrain the models using the co-training paradigm.

5.6.3 Online Learning and Classification

The CORAL model works in a online manner so that learning and classification can be achieved simultaneously. More specifically, we maintain a learning set G for training the classification model using the CORAL model. Given a new road obstacle e and the label set $C = \{c_1, c_2, \ldots, c_l\}$, we apply the learned L_s and L_t on e separately, and determine the final label c by the product of the two confidence scores generated by the two learners, defined as follows:

$$c = \arg\max_{c_i \in C} \quad P_s(C = c_i) \times P_t(C = c_i) \tag{5.6.4}$$

where P_s and P_t are the predicted probabilities of the spatial and temporal learners, respectively. After that, we append e to the training set G and retrain the CORAL model. If additional labels are required during the retraining, the corresponding crowdsensing platform tasks will be allocated and completed in a given time constraint. Since online model updating is frequently performed, we adopt a naïve Bayesian network [182] as the ideal multi-class classifier for the learners L_s and L_t, which is highly scalable, easy to train, and outputs the desired classification probabilities for confidence estimation.

5.7 Evaluation

In this section, we first introduce the experiment settings, and then present the evaluation results on road obstacle detection and recognition. We also conduct a series of case studies to demonstrate the effectiveness of our method.

5.7.1 Experiment Settings

Datasets

We evaluated our framework in Xiamen City during the 2016 Pacific typhoon season[7]. Xiamen is a coastal city which suffers from an average of 4–5 times of typhoon each year. We collected a large scale taxi GPS trajectory dataset and various road environment datasets from July 2016 to December 2016, as summarized in Table 6.1. The dataset details and pre-processing steps are elaborated as follows.

Taxi GPS trajectory data: we obtained a large-scale taxi GPS trajectory dataset from Xiamen Traffic Police. The dataset contains GPS trajectories of 5,486 taxis reported every 1 minute. We extracted only trajectories data points in the metropolitan area (i.e., Xiamen Island) during the second half year of 2016.

[7] https://en.wikipedia.org/wiki/2016_Pacific_typhoon_season

Table 5.1: Summary of Datasets

Data type	**Item**	**Value**
Vehicle trajectory data	# Taxis	5,486
	Sampling rate	every minute
Road environment data	# Road segments	3,928
	Satellite image resolution	2.5 meter
	Elevation resolution	30 meter
	Meteorology data	every hour
Road Obstacles	# Fallen trees	71
	# Ponding water	54
	# Congested and crashed vehicles	34
Data collection period	07/01/2016 00:00–12/31/2016 23:59	
Geographic coverage area	Southwest: [24.423250, 118.064743] Northeast: [24.561485, 118.198504]	

Road environment data: we partitioned Xiamen City into $100m \times 100m$ grids, and obtain 154×136 grids. We then extracted grids with vehicle density above average as road segments for vehicles, obtaining 3,928 road segments in total. Upon this basis, we collected a set of satellite image patches and elevation data for each road segment from Google Earth for roadside tree coverage labeling and road elevation sensing, respectively. We also compiled an hourly meteorology dataset for each road segment, containing the wind speed, rain precipitation, and visibility readings, based on the data from the Weather Underground API and Xiamen Meteorological Bureau.

Road obstacle data: we developed a crowdsensing platform to collect road obstacles during typhoon seasons, as shown in Figure 5.7. We recruited 10 participants to finish the crowdsensing tasks, and each task was assigned randomly to three participants for cross-validation to avoid observer bias. Each participant was asked to report the time, location, and type of the road obstacle along with the source materials such as images and videos. An important source is the social media accounts of urban authorities and local news agencies, such as the Weibo[8] Based upon the crowdsensing platform, we collected a total number of 159 road obstacle events from July 2016 to December 2016. We use this dataset as the ground-truth for evaluating the performance of various road obstacle detection and classification methods.

[8] Weibo is a Twitter-like social network popularly used in China. accounts of Xiamen Traffic Police[9] and Xiamen News Network[10]. Besides, the traffic congestion and accident records provided by Xiamen Traffic Police were used as another important source to identify congested and crash vehicles on the road.

Evaluation Plan

We evaluated the performance of our framework in an online manner. We first extracted taxi slow mobility behaviors in all the road segment once every 30 minute. For road obstacle detection, we maintained a slow mobility behavior matrix M with a time window of one month, i.e., $2*24*30 = 1440$ time spans, and updated the matrix when data from a newer time window were collected. We performed the CDRMF algorithm on M to detect a set of road obstacles $\{e_i\}$, and compared the detection results with the ground truth dataset. For road obstacle classification, we maintained a learning set G and incrementally add new instances into the set. Specifically, for each newly detected road obstacle $\{e_i\}$, we used the learned CORAL model to classify it, and then add $\{e_i\}$ to the training set and update the CORAL model. The initial labels and the additional labels required during the model updating were obtained by dynamically allocating crowdsensing tasks to the participants. Finally, we compared the overall classification results with the ground truth dataset to evaluate the model accuracy.

Evaluation Metric

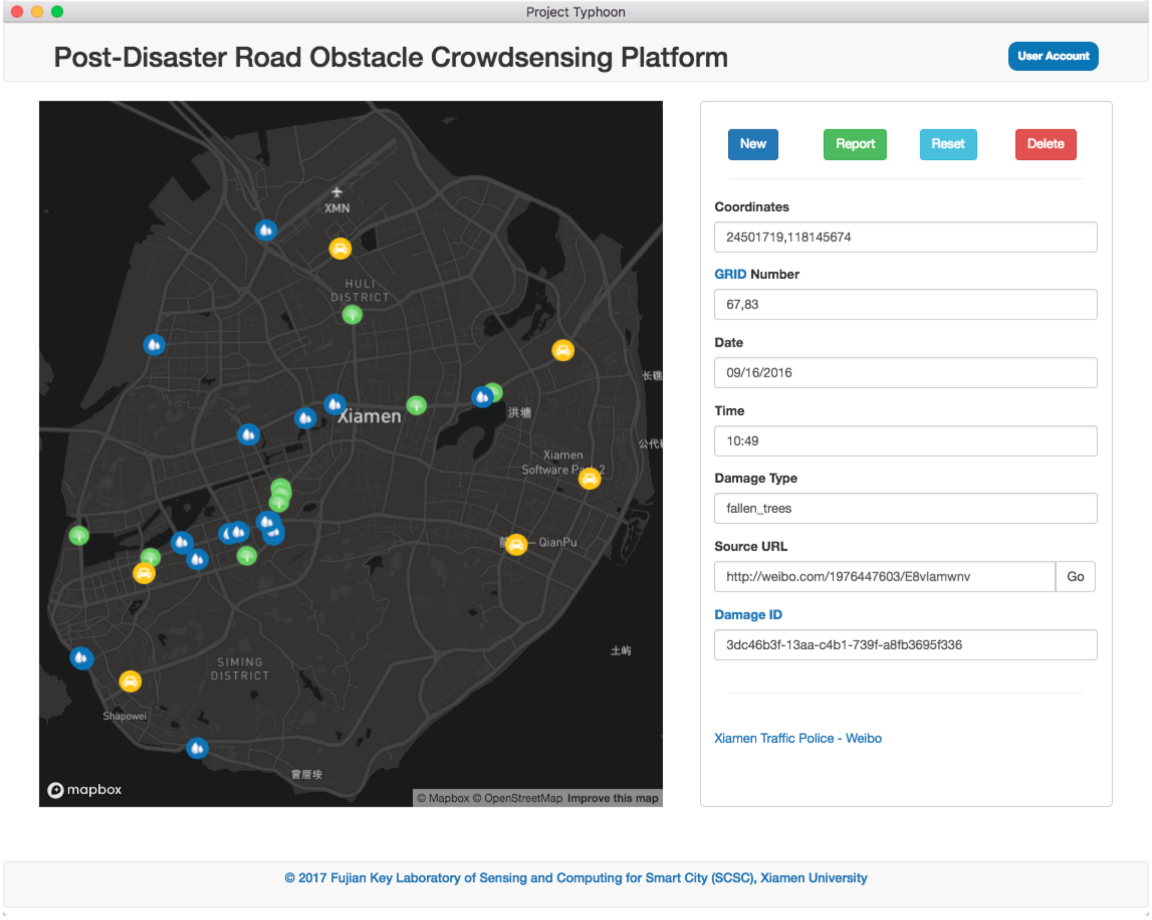
We compared the detected road obstacles with the ground truth dataset to evaluate the accuracy of a detection method. Specifically, if a detected obstacle is found in the ground truth dataset, we call it a *true positive (TP)*, and otherwise a *false positive (FP)*. For a true road obstacle that is not detected using the detection method, we call it a miss, or *false negative (FN)*. With these definitions, we adopted the following metrics to quantitatively evaluate the performance of the detection method:

$$precision = \frac{|TP|}{|TP|+|FP|}, \quad recall = \frac{|TP|}{|TP|+|FN|}, \quad F1\text{-}Score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{5.7.5}$$

We employed the similar metrics to evaluate the performance of the multi-class road obstacle classification method. Specifically, we organized the classification results into a confusion matrix [183] C, where each row of the matrix represents the instances in a predicted class and each column represents the instances in a ground truth class. Each element $C_{i,j}$ counts the number of road obstacles that are predicted as class i while actually are in class j. With these definitions, we derived the following metrics:

$$precision = \frac{C_{i,i}}{\sum_j C_{ij}}, \quad recall = \frac{C_{i,i}}{\sum_j C_{ji}}, \quad F1\text{-}Score = \frac{2 \cdot precision \cdot recall}{precision + recall} \tag{5.7.6}$$

We can see that (5) is a special case of (6).



Post-Disaster Road Obstacle Crowdsensing Platform

Coordinates
24501719,118145674

GRID Number
67,83

Date
09/16/2016

Time
10:49

Damage Type
fallen_trees

Source URL
http://weibo.com/1976447603/E8vlamwnv

Damage ID
3dc46b3f-13aa-c4b1-739f-a8fb3695f336

Xiamen Traffic Police - Weibo

© 2017 Fujian Key Laboratory of Sensing and Computing for Smart City (SCSC), Xiamen University

Figure 5.7: The developed crowdsensing platform for collecting road obstacles.

Baseline Methods

We compared the proposed method with various baselines with regard to road obstacle detection and classification. For road obstacle detection, we design the following baselines to compare with the proposed CDRMF method:

TFBOY: the traffic-flow-based anomaly detection (TFBOY) baseline method directly uses the number of vehicles in each road segment during each time span to construct a traffic flow matrix, and use the same matrix decomposition approach for road obstacle detection.

ARIMA: the single-road-segment time-series-based baseline method models the number of slow mobility behaviors in each road segment as a time series, and employs an auto-regressive integrated moving average (ARIMA) model [184] to detect significant and unusual events as potential road obstacles.

DRMF: this baseline method differs from the proposed CDRMF method in that it does not perform a clustering step to remove isolated anomalies.

For road obstacle classification, we design the following baselines to compare with the proposed CORAL method:

ST-ANN: this baseline method directly concatenate the spatio-temporal contextual features, and uses all the instances in the training set to train an Artificial Neural Network (ANN) model for road obstacle classification.

SCAL: this baseline method exploits only the spatial contextual (SC) features to build a road obstacle classifier using a naive Bayesian network. The active learning (AL) diagram is also exploited to train the model iteratively from a minimal training set.

TCAL: similarly, this baseline method exploits the temporal contextual (TC) features and active learning (AL) to build a road obstacle classifier by iteratively training a naive Bayesian network with a minimal training set.

COTA: this baseline method uses the co-training diagram (COTA) alone without active learning. It trains a naive Bayesian network-based spatio-temporal model and iteratively improves the model accuracy using confident unlabeled data.

In order to achieve fair comparison, we make sure that the size of the initial label set and the number of labels finally used in each active learning-based baselines (including SCAL, TCAL, and CORAL) are the same.

5.7.2 Evaluation Results

We first present the overall results of road obstacle detection and classification using optimal parameters, and then study of the parameter selection strategies in the CDRMF and CORAL models.

Road Obstacle Detection Results

We compare the overall accuracy of different methods in Table 5.2. We can see that our CDRMF method achieves an F1-score of 0.942 (precision=0.953 and recall=0.931), outperforming the other baseline methods. In particular, the TFBOY method fails to detect most of the road obstacles, since the traffic volume decrease induced by road obstacles are overwhelmed by the global decrease of traffic volume during post disaster periods. The ARIMA method achieves a relatively low recall but high precision, meaning that it fails to detect some of the road obstacles (false negatives) but it does not have many wrong detections (false positives), neither. The probable reason is that ARIMA does not model the spatio-temporal collectiveness and thus fails to capture collective anomalies that

Table 5.2: Road obstacle detection results

Methods	Precision	Recall	F1
TFBOY	0.503	0.489	0.496
ARIMA	0.872	0.697	0.775
DRMF	0.730	0.906	0.809
CDRMF	0.953	0.931	0.942

Table 5.3: Road obstacle classification results

Methods	Precision	Recall	F1
ST-ANN	0.921	0.956	0.938
SCAL	0.772	0.738	0.755
TCAL	0.691	0.649	0.669
COTA	0.843	0.870	0.856
CORAL	0.902	0.931	0.916

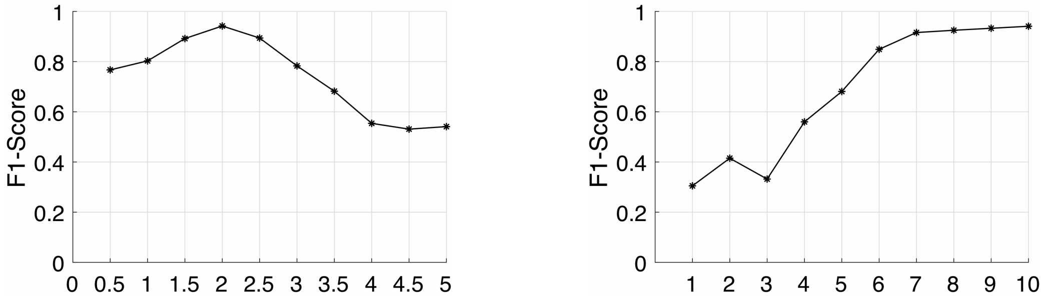
are not obvious from the view of a single road segment. The DRMF method, on the other hand, achieves relatively low precision and high recall, which means that it detects many road obstacles but only some of them are present in the ground truth set (true positives). This is reasonable since DRMF does not impose structural constraints on the anomalies and thus results in many isolated anomalies that should not be considered being induced by road obstacles. In general, our method successfully captures the collective anomalies induced by road obstacles in disaster response scenario and achieves relatively high detection accuracy.

Road Obstacle Classification Results

We present the road obstacle classification results in Table 5.3. The ST-ANN method achieves the highest precision and recall using both spatio-temporal features and a *full* label set. However, in practice, collecting such a training set is labor-intensive and time-consuming, which hinders the online deployment in disaster response scenarios. For the other methods, we start from an initial label set of size 7, and limit the total number of labels to be 20. The SCAL and TCAL baseline methods do not perform well, justifying the assumption that neither set of features are significant enough for building an effective road obstacle classifier. In contrast, the COTA baseline performs better by exploiting both sets of features in a co-training model. The proposed CORAL method further improves the performance by incorporating the active-learning diagram, achieving an F1-score of 0.916 with precision=0.902 and recall=0.931, outperforming the other semi-supervised-learning-based baseline methods, and achieving accuracy comparable with the ST-ANN method while using only 12.5% of the labels.

Parameter Study

Low-rank constraint k: in the CDRMF model, the rank constraint parameter k needs to be carefully selected in order to separate the regular patterns from the mixed slow mobility behaviors. We perform a rank estimation on M using an SVD-shrinkage method [168], and select $k = 6$ in our experiments that preserve most of the significant singular values of M.



(a) Sparsity constraint c in the CDRAM model

(b) Initial label set size $|G_1|$ in the CORAL model

Figure 5.8: Parameter impact analysis and optimal parameter selection.

Sparsity constraint c: the sparsity constraint in the CDRMF model determines the number of collective anomalies that can be detected, and thus influences the precision and recall of the model. Based on [168], we vary c from 0 to 5% of the size of M, and present the F1-score under different c values in Figure 5.8(a). We can see that a sparsity constraint too small or too large may result in suboptimal model accuracy, and thus we select $c = 2\%$ in our experiments.

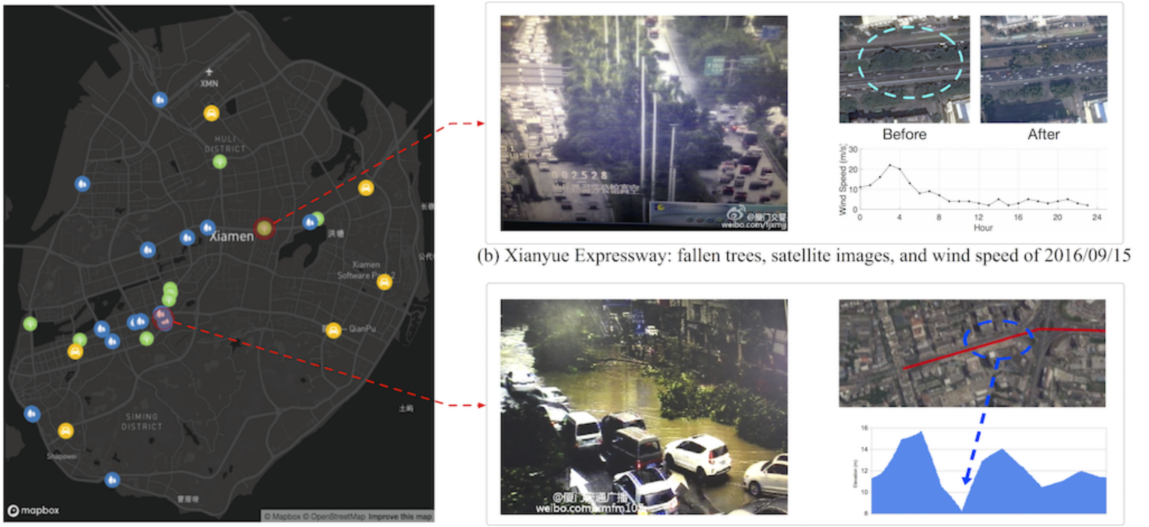
Initial label set size $|G_1|$: in the CORAL model, the set of initially labeled instances is of great importance for iterative model training. A large initial label set can help achieve high classification accuracy, however collecting these labeled instances requires great effort and reduces the feasibility of online deployment. Therefore, we have to make trade-offs between performance and feasibility. As shown in Figure 5.8(b), we study the F1-score of the CORAL method against various $|G_1|$ values. We can see that an initial training set of size 7 is large enough to obtain an F1-score higher than 90%. Therefore, we select $|G_1| = 7$ as the optimal initial label set size in our experiments.

Furthermore, we determine the optimal distance threshold $\delta_p = 1m$ for the slow mobility sequence extraction algorithm, which yields closest results to the observations over a series of repeated experiments. In the CORAL model, we empirically set the number of the selected instances $N_c = Na = 2$ based on repeated experiments.

Runtime Performance

We implement the CDRMF and CORAL algorithms using Matlab, based on the DRMF matlab package provided by [168]. We deploy our framework on a server with Intel core i7-6700K CPU and 16GB RAM, and it takes an average of 6.1 seconds and 3.2 seconds to do one round of road obstacle detection and classification[11], respectively.

[11] We do not count in the extra time of the crowdsensing tasks



(b) Xianyue Expressway: fallen trees, satellite images, and wind speed of 2016/09/15

(a) Identified road obstacles during Typhoon Meranti

(c) Hubin South Road: ponding water, satellite images, and elevation of the road segment

Figure 5.9: Case studies on road obstacle identification after Typhoon Meranti's landfall in Xiamen.

5.7.3 Case Studies

We conduct case studies on road obstacles identification after the landfall of Typhoon Meranti in Xiamen from 09/15/2016 to 09/17/2016. Figure 5.9 shows the overall distribution of the detected and classified road obstacles. In general, we observe several water ponds formed in the lower center part of the island, which corresponds to the downtown area with relatively low elevation. Fallen trees block some of the major roads and cause serious transportation problems. Traffic accidents induced by heavy rains and limited visibility increase significantly in complicated road interactions and tunnel entries. These road obstacles pose great challenges for disaster response, and thus need to be identified and removed as early as possible. In the following analysis, we present two cases of identified road obstacles.

Fallen Trees on Xianyue Expressway

Figure 5.9(b) shows the identified fallen trees in Xianyue Expressway, one of the busiest high-capacity road in Xiamen City. Our framework successfully identifies the road obstacle in 3 neighboring road segments at 2016/09/15 17:00, which is 16 hours ahead of the event report on the Weibo account of Xiamen Traffic Police[12]. As we can see in Figure 5.9(b), the road segments were covered by flourishing trees before the typhoon strike, which posed potential risks of fallen trees during the strong winds brought by Typhoon Meranti. Our framework and analysis can not only

[12] http://weibo.com/1976447603/E8vIamwnv

help urban transportation authorities to clear the obstacles in a timely manner, but also provide decision support for urban environment authorities in roadside tree planning and pruning [178].

Ponding Waters on Hubin South Road

Figure 5.9(c) illustrates the identified ponding water on the surface of Hubin South Road, which is the trunk road in downtown with a relatively low-elevation (Hubin means lakeside). We identified 5 road segments along the road influenced by ponding water at 2016/09/15 08:30, which is 2 hours earlier than the event is reported on the news[13]. From Figure 5.9(c), we can see several low-lying road segments alone Hubin South Road near the Yundang Lake, which are potentially vulnerable to the heavy rains brought by Typhoon Meranti. With the real-time information at hand, the urban authorities could take preventive actions, for example, by sending out crews to drain the ponding water and prevent further security issues on the road.

5.8 Conclusion

In this work, we propose a two-phase framework to detect and classify road obstacles in disaster response scenarios, leveraging large-scale vehicle trajectories and many cross-domain urban sensing datasets. In order to detect road obstacles, we extract slow mobility behaviors from vehicle trajectories, and propose the CDRMF approach to detect collective anomalies from a mixture of slow mobility behaviors, and correspond them to road obstacles. In order to recognize the types of the detected obstacles, we exploit the spatio-temporal contextual features extracted from various environment sensing data to train a classification model. To address the challenges of heterogeneous features and insufficient labels, we propose a semi-supervised approach combining co-training and active learning. We evaluate our framework using real-world datasets collected from Xiamen City. Results show that our framework accurately detects and classifies the road obstacles in the 2016 typhoon season with an overall accuracy both above 90%, and outperforms the baseline methods.

In the future, we intend to improve this work from the following aspects. First, we plan to use trajectories of various kinds of vehicles besides taxis, such as buses and rental cars. Second, we plan to characterize the slow mobility behaviors in a finer granularity, for example, by identifying waiting and turning behaviors. Third, we plan to improve the granularity of road spatial features by incorporating the high-resolution LIDAR (Light Detection and Ranging) data collected by our team in Xiamen University for better classification of road obstacles. Fourth, we plan to evaluate our framework in other cities with different geographic and meteorological settings, and explore the obstacle identification problem under other nature disaster types.

[13] http://weibo.com/1750354532/E8lY2Etxr

Chapter 6

Deep Mobility Prediction for Energy-Efficient and Quality-Aware C-RAN

Contents

6.1 Introduction

In recent decades, the number of mobile subscriptions is growing at almost 6 percent year-on-year, reaching 7.8 billion at the end of 2017 [18]. Correspondingly, the network traffic volume has grown 18-fold over the past five years [19] as Internet-enabled devices become increasingly popular. In order to cope with the fast growing number of subscribers and the surging traffic demand, mobile network operators are deploying more and more base stations to expand their network coverage [18], and adding more powerful processing units to increase their network capacity [141]. However, as network scale and capacity grow, the capital expenditure (CAPEX) and operating expenditure (OPEX) are becoming increasingly high [20, 21]. Consequently, the *surging cost* of infrastructure investment and energy consumption may exceed operator's revenue [24, 185]. Moreover, the *quality-of-service* of the network, such as handover delay between base stations, becomes increasingly difficult to ensure as various sizes of base stations (e.g., pico-cells, micro-cells, and macro-cells) and different generations of technologies (e.g., LTE, UMTS, and GSM) co-exist in the network [25]. Therefore, designing *cost-effective and high-quality* mobile network architectures is now a great necessity in the field of mobile network operation and research [21].

Fortunately, with the rapid evolution of mobile network technologies, the emergence of *Cloud Radio Access Network (cloud-RAN)* [26] has provided new opportunities to address the above mentioned challenges. In cloud-RAN, a traditional base station is split into two components: the *Remote Radio Head (RRH)* for radio communication with mobile devices, and the *Baseband Unit (BBU)* for signal and data processing [23]. The BBUs are further detached from the RRHs and hosted in centralized *BBU pools* [27]. The RRHs and BBU pools are usually connected via high speed optical fiber [28]. For example, Figure 6.1 shows an illustrative network consisting of three RRHs and one BBU pool. By employing the cloud-RAN architecture, the cost-effectiveness and quality-of-service can be improved compared with the traditional radio access network architectures [23]. We exploit the example in Figure 6.1 to elaborate the benefit of cloud-RAN. On one hand, since multiple RRHs can be connected to one BBU and share the processing capacity (RRH #1 and #2 share BBU #1), the utilization rate of the BBUs can be increased, and the unused BBUs (BBU #3) can then be dynamically shutdown to save energy consumption. Consequently, the cost-effectiveness of the network is improved. On the other hand, when two RRHs are connected to one BBU (RRH #1 and #2), the handover events between them can be handled directly inside the BBU (BBU #1), which greatly reduces the handover delay. Consequently, the quality-of-service of the network is improved. With the benefits it brings, cloud-RAN architecture is foreseen as a typical realization of soft and green technologies in fifth generation (5G) mobile network [71].

In order to unlock the power of the cloud-RAN architecture, one of the key problem is to design optimal *mapping schemes* between RRHs and BBU pools, so as to maximize the utilization rate

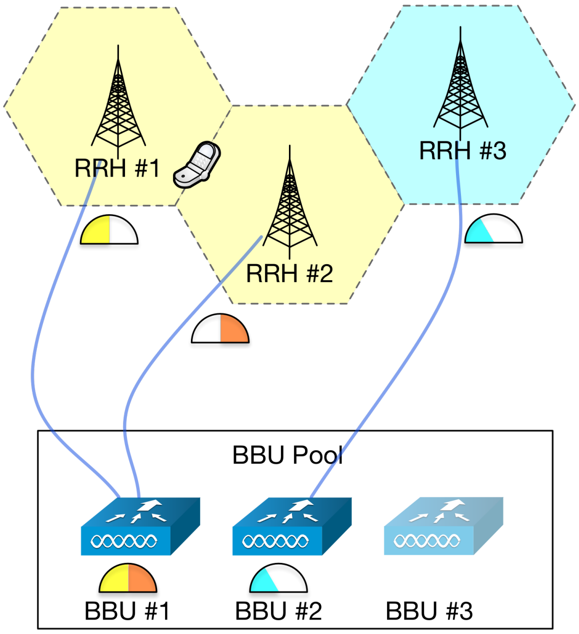


Figure 6.1: An illustrative example of a network consisting of three RRHs and one BBU pool. The BBU pool and the RRHs are connected via high-speed optical fiber. The traffic volume of each RRH is represented in the semicircle, and the mobile device between two RRHs denotes users moving between them (i.e., handover events between RRHs).

(i.e., reduce cost) and minimize the handover overhead (i.e., improve quality) of the entire network [28, 186, 187]. Ideally, a cost-effective and high-quality RRH-BBU mapping scheme partitions a set of RRHs in the network into several *clusters* and allocates BBUs from the pool to the clusters, so that (1) the aggregated traffic volume generated in each RRH cluster is close to the capacity of the BBU allocated to the cluster, and (2) the handover events are handled within each RRH cluster and processed inside the corresponding BBU to the maximal extent. However, finding such optimal RRH-BBU mapping schemes is not trivial, since the traffic demands and handover events among the RRHs are highly *dynamic*, and the possible mapping schemes are *enormous* to verify one-by-one. More specifically, the following challenges need to be addressed:

- **How to accurately foresee RRH traffic volume and handover count?** In order to make optimal RRH-BBU mapping schemes for a future period of time, we need to foresee the traffic volume and handover count in the network beforehand. However, due to the dynamic nature of human activity and mobility, the traffic volume and handover count among the RRHs can vary significantly, depending on the temporal contexts (e.g., weekdays or weekends) and spatial functions (e.g., residential areas or business districts). For example, during weekday working hours, the RRHs located in business districts and transit hubs usually observe high traffic volume and massive handover count. Existing work on RRH-BBU mapping usually employ probability models to simulate traffic volume and handover count [27,28,187], which may not be able to capture the traffic and handover dynamics in real-world, and might result in sub-optimal mapping schemes in real-world deployment. Therefore, we need to propose effective methods to model the dynamics of the network, and accurately predict the traffic and handover among RRHs.

- **How to efficiently find optimal RRH-BBU mapping schemes?** Given the predicted RRH traffic volume and handover count in the network, there are potentially enormous schemes to cluster these RRHs and to allocate BBUs to these clusters. Moreover, the *global resource constraints* of the BBU pools, such as the pool capacity and the BBU size (e.g., CPU and memory specifications), should be taken into consideration during the search of the optimal mapping schemes. Existing work using exhaustive search methods quickly becomes intractable as network scale grows [28], while competitive optimization methods such as coalition games are difficult to impose the global constraints during the mapping procedure [40]. Therefore, we need to design an effective algorithm to find optimal RRH-BBU mapping scheme with cost and quality objectives under the BBU resource constraints.

With the above-mentioned objectives and challenges, the **main contributions** of this work are:

- To the best of our knowledge, this is the first work on data-driven cloud-RAN optimization leveraging dynamic RRH clustering based on network traffic and user mobility prediction. By analyzing real-world RRH traffic and handover data, we are able to capture the dynamics of the network and develop optimal RRH-BBU mapping schemes for real-world deployment.

- We propose a two-phase framework to dynamically find optimal RRH-BBU mapping schemes based on the accurate prediction of RRH traffic and handover in the network. In the first phase, we employ a *data-driven* approach to model the spatio-temporal dynamics of a network with real-world call detail records, and then propose a *deep-learning-based model* leveraging Multivariate Long Short Term Memory (MuLSTM) algorithms to accurately predict the traffic volume and handover count in a future period of time. In the second phase, we model the RRHs in a network as a *weighted graph*, taking into consideration the RRH traffic volume and handover count as graph nodes and link weights, respectively. We then formulate the optimal RRH-BBU mapping with cost and quality objectives as a *set partition problem*, and propose a Resource-Constrained Label-Propagation (RCLP) algorithm to find the optimal mapping schemes under the BBU pool resource constraints.

- We evaluate the performance of our framework using two real-world mobile network datasets collected from Ivory Coast and Senegal. Real-world evaluation results show that our framework effectively increases 24.74% of utilization rate and reduces 12.05% of energy consumption compared with traditional RAN architectures, and outperforms the state-of-the-art baseline methods.

6.2 Preliminaries and Framework

6.2.1 Terminologies

In a traditional mobile network architecture, a set of *base stations* are deployed over geographical areas called cells [145]. Each base station provides the cell with the network coverage which can be used for transmission of voice and data. With the recent emergence of Internet-enabled devices, such as smartphones and tablets, the data traffic generated from the bases stations are increasing explosively [19, 21], demanding for higher data processing capacity in the network. Meanwhile, with the rapid urbanization and economic development, the coverage and scale of mobile networks are expanding tremendously, bringing about networks consisting of various sizes of base stations and different generations of technologies [25]. Such a hybrid architecture poses great challenge for operators to ensure the network quality-of-services (QoS), especially when handling handover delays when users move among base stations [30, 80]. The cloud-RAN has been proposed to address the above-mentioned challenges. We define some of the terminologies used in cloud-RAN as follows:

Definition 14. ***Remote Radio Head (RRH):*** an RRH is the radio transceiver placed in a base station site to facilitate wireless communication between user devices and the network [188]. We define an RRH r as a 3-tuples:

$$r =< label, lat, lng > \tag{6.2.1}$$

where $label$ is the label used to identify the RRH, and lat and lng are the corresponding latitude and longitude coordinates of the RRH.

Definition 15. ***RRH Traffic Volume:*** in this work, we refer to the term *traffic volume* as the quantity of radio resource units [28] consumed in the RRH for communication during a period of time, which can be the total duration of calls, the overall volume of Internet data, etc. Specifically, we denote the traffic volume of RRH r_i during time span t as $f(r_i, t)$.

Definition 16. ***RRH Handover Count:*** in this work, we refer to the term *handover count* as the quantity of users moving between a pair of two RRHs during a period of time. Specifically, we denote the handover count between RRH r_i and RRH r_j during time span t as $h(r_i, r_j, t)$.

Definition 17. ***Baseband Unit (BBU):*** a BBU is a device providing digital signal processing functionalities for RRHs from baseband processing to packet processing [188]. Specifically, we define a BBU as a 3-tuple:

$$b =< label, pool, cluster > \tag{6.2.2}$$

where *label* is the label to identify a BBU instance, *pool* is the BBU pool where the BBU is allocated, and *cluster* is the RRH cluster where the BBU is assigned to.

Definition 18. ***BBU Capacity:*** in the cloud-RAN architecture, BBUs are usually implemented as virtual machine instances with specific sizes of computing resources, including CPU, memory, and storage [23, 189]. Consequently, the resultant BBU capacity can be classified into a set of *discrete* levels, e.g., *LARGE*, *MEDIAN*, and *SMALL*. Specifically, we define the set of BBU capacity level as

$$\mathbb{L} = \{l_1, \ldots, l_{N_l}\} \tag{6.2.3}$$

where N_l is the number of capacity levels. Correspondingly, we denote the capacity level of BBU b_k as $l(b_k) \in \mathbb{L}$.

Definition 19. ***BBU Pool:*** in the cloud-RAN architecture, a BBU pool is a cloud-based data center with low-cost and high-speed interconnect network, a real-time virtualization platform with dynamic shared resource allocation and management, and a general-purpose baseband processing platform with multiple BBUs [23, 26]. For a city-scale network, one or more BBU pools can be implemented and connected to RRHs via high-speed optical fiber. Specifically, we denote a BBU pool as a set of BBUs:

$$\mathbb{B} = \{b_1, b_2, \ldots, b_k\} \tag{6.2.4}$$

In this work, we consider a Cloud-RAN architecture with one BBU pool for the city-wide mobile network. The benefits of adopting such a centralized pool are three-fold. First, the deployment cost and energy consumption can be greatly reduced by employing data center virtualization technologies [146]. Second, the handover handing and contents offloading among RRHs can be processed internally in the pool, which significantly reduces delays and increases throughput [23]. Third, the network upgrades and hardware maintenance are easy to conduct just in one place, without the need of labor-consuming on-site work.

6.2.2 Framework Overview

As presented in Figure 6.2, we propose a two-phase framework to accurately predict RRH traffic volume and handover count based on historical data, and then dynamically find optimal RRH-BBU mapping schemes for cloud-RAN optimization. In the *RRH traffic and handover prediction* phase, we first model the traffic and handover dynamics among RRHs with spatial and temporal dimensions, and then propose a deep-learning-based approach to predict the traffic and handover for a future period of time. In the *dynamic RRH-BBU mapping phase*, we first model the network with

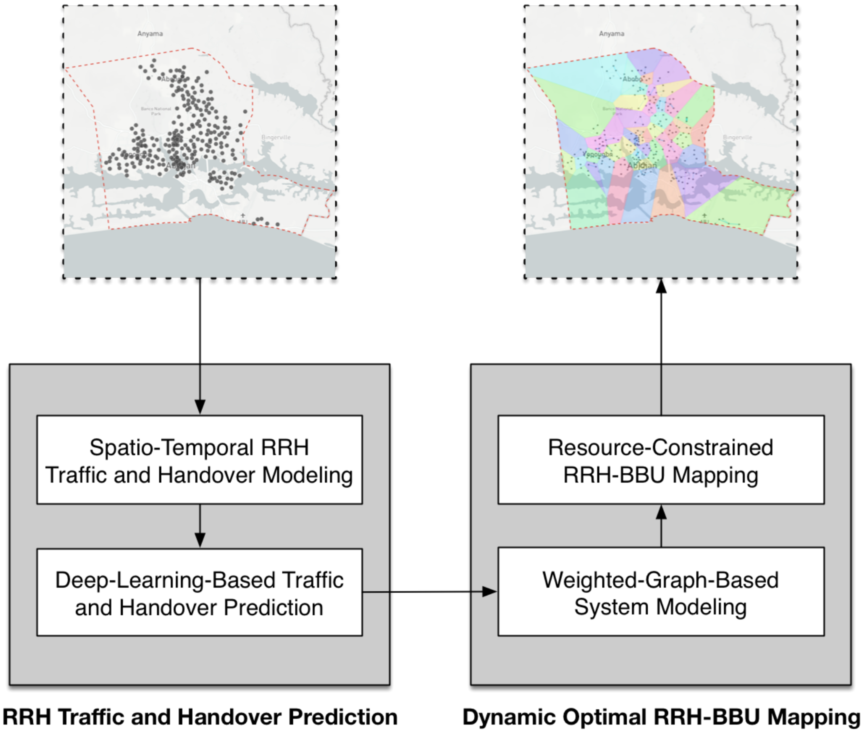
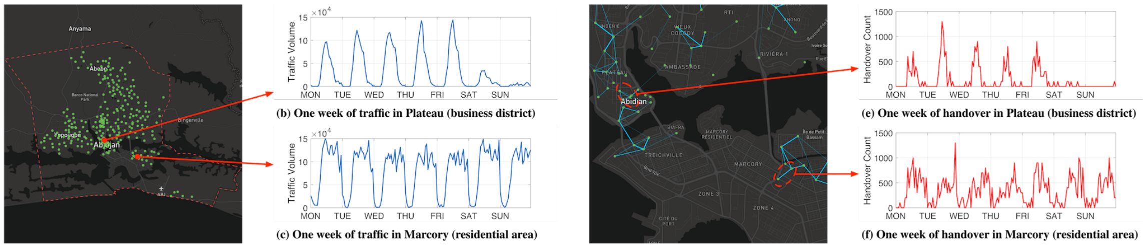

Anyama
Spatio-Temporal RRH Traffic and Handover Modeling
Deep-Learning-Based Traffic and Handover Prediction
RRH Traffic and Handover Prediction
Resource-Constrained RRH-BBU Mapping
Weighted-Graph-Based System Modeling
Dynamic Optimal RRH-BBU Mapping

Figure 6.2: Framework overview.

predicted RRH traffic volume and handover count as a weighted graph, and identify the resource constraints from the BBU pools. We then propose a resource-constrained RRH-BBU mapping algorithm with the objectives of maximizing the network cost and quality. In the following sections, we elaborate on the details of this framework.

6.3 RRH Traffic and Handover Prediction

In this phase, our objective is to accurately predict the RRH traffic volume and handover count in a future period of time, so that we can determine the optimal RRH-BBU mapping scheme in the next phase. However, this is not trivial due to the highly dynamic nature of social activity and user mobility. On one hand, the RRH traffic and user mobility behaviors may vary significantly under different *temporal* contexts. On the other hand, the *spatial* function of an area may have strong impacts on the traffic and mobility patterns of the RRHs located in that area. For example, Figure 6.3 shows an illustrative example of the traffic and mobility dynamics in Abidjan, Ivory Coast during one week. In the business district Plateau, we can observe significant different traffic (Figure 6.3b) and mobility (Figure 6.3e) patterns between weekdays and weekends. In contrast, the



(b) One week of traffic in Plateau (business district)

(c) One week of traffic in Marcory (residential area)

(e) One week of handover in Plateau (business district)

(f) One week of handover in Marcory (residential area)

(a) RRH distributions in Abidjan, Ivory Coast

(d) User mobility pattern in Abidjan, Ivory Coast

Figure 6.3: The RRH traffic and user mobility dynamics in Abidjan, Ivory Coast during a sample week (01/09/2012–01/15/2012). In map (a) and (c), each green dot corresponds to an RRH. In map (c), the blue links denote the number of users moving between pairs of RRHs. Wider link corresponds to higher mobility count.

traffic and mobility patterns in the residential area Marcory are quite different from that of Plateau.

In a word, the real-world traffic and mobility patterns demonstrate *temporal dependency* and *spatial correlation*. However, traditional prediction methods usually model each RRH traffic as single time series [96, 98], and model the user mobility as static graphs [39, 190], which fails to integrate the spatial and temporal dynamics in a unified model, and hinders the accurate prediction of RRH traffic and handover. Therefore, we propose a deep-learning based approach to model the spatial and temporal dynamics as a multivariate Long Short Term Memory (MuLSTM) neural network for accurate prediction. In this section, we first briefly introduce the datasets we use for extracting the RRH traffic and user mobility in a network. Then, we details the models we use to capture the spatial and temporal dynamics of the RRH traffic and handover data. Finally, we propose a deep-learning-based approach for accurate prediction of the RRH traffic volume and handover count.

6.3.1 Call Detail Records

In mobile networks, call detail records are data that document the details of phone calls, text exchanges, or other telecommunication transaction that pass through the network infrastructures [191]. Call detail record data contain rich information about social activity and user mobility, providing opportunities to optimize mobile network infrastructures, such as reducing operation cost and improving service quality. In this work, we exploit two real-world large-scale anonymized call detail record datasets released by Orange group via the Data for Development (D4D) Challenge[1]. The datasets are collected from Orange customers from Ivory Coast for half-a-year, and Senegal in one year, respectively. The datasets consists of the following information:

- **RRH attributes**: including the RRH labels and geographic coordinates.

[1] http://www.d4d.orange.com/

- **RRH communication**: including the number and durations of phone calls and SMS exchanges between RRHs in the network on an hourly basis.
- **User attributes**: including anonymized user labels which are shuffled every two weeks for privacy concerns.
- **User Mobility**: including the user mobility trajectories among RRHs in the network with precise time and RRH information.

Based on the datasets, we aggregate the communication and mobility events by RRH, and perform data cleansing process to extract the RRH traffic volume and RRH handover count on hourly basis, respectively. More details about the datasets are presented in the evaluation section.

6.3.2 Spatio-Temporal RRH Traffic and Handover Modeling

In order to capture the spatial and temporal dynamics of the RRH traffic and handover, we construct two *tensors* [10, 165] to model the traffic volume generated in each RRH and the handover counts observed among each RRH pair, respectively. More specifically, given a network with N_r RRHs and the corresponding call detail record data observed in N_t time spans, the *RRH traffic tensor* and *RRH handover tensor* are defined as follows:

RRH traffic tensor: we build a tensor $\mathcal{F} \in \mathbb{R}^{N_r \times N_t}$ with two dimensions to model the RRH traffic volume, where $\mathcal{F}(r_i, t)$ corresponds to the total incoming and outgoing communication traffic volume of RRH r_i during time span t ($i = 1, \dots, N_r, t = 1, \dots, N_t$). We note that based on different scenarios, the definition of traffic may vary, such as the total duration of calls, the number of messages, and the overall volume of Internet data. For example, Figure 6.3b and 6.3c visualize two typical traffic patterns extracted from two specific RRHs in $\mathcal{F}$.

RRH handover tensor: we build a tensor $\mathcal{H} \in \mathbb{R}^{N_r \times N_r \times N_t}$ with three dimensions to model the RRH handover counts, where $\mathcal{H}(r_i, r_j, t)$ corresponds to the total count to handover events between RRH r_i and RRH r_j during time span t. We consider the case of symmetric modeling where $\mathcal{H}(r_i, r_j, t) = \mathcal{H}(r_j, r_i, t)$. As an example, Figure 6.3e and 6.3f visualize two typical handover patterns extracted from two specific pairs of RRHs in $\mathcal{H}$.

6.3.3 Deep-Learning-Based Traffic and Handover Prediction

In this work, we use the similar deep-learning techniques introduced in Chapter 4 for traffic and handover prediction. Specifically, we model the RRH traffic and handover simultaneously using the MuLSTM model as follows.

The MuLSTM model: in this work, we use LSTM networks to effectively learn the temporal

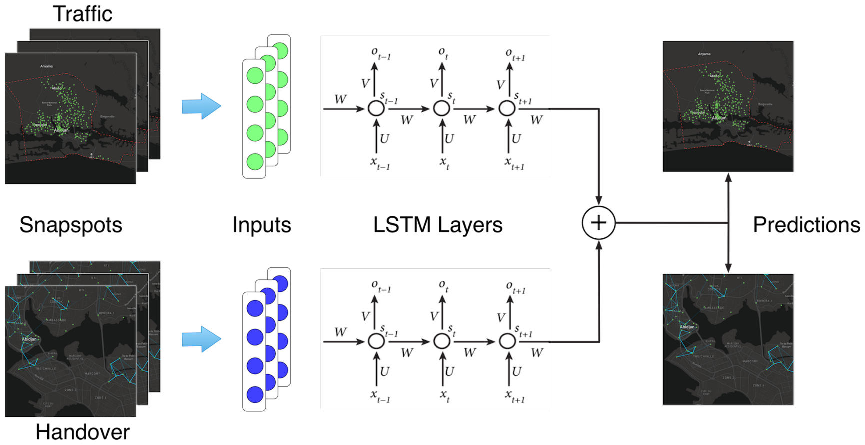


Figure 6.4: The MuLSTM model for RRH traffic and handover prediction.

dependency of RRH traffic patterns and handover patterns from historical data. To further model the *spatial correlation* among RRHs in the network, we propose a multivariate Long Short Term Memory Network (MuLSTM) architecture to simultaneously integrate all the RRHs in a unified model. Specifically, each RRH traffic is regarded as an input variable to a shared LSTM model, while each RRH handover pair is regarded as an input variable to another shared LSTM model. The two LSTM models accept the multivariate inputs and are trained jointly. Figure 6.4 shows the overview of the proposed MuLSTM model. We elaborate on the technical details as follows.

Snapshots: we first generate two sets of consecutive *traffic and handover snapshots* based on the traffic tensor $\mathcal{F}$ and the handover tensor $\mathcal{H}$, respectively. A snapshot is a slice of the tensor along the time axis, which corresponds to the traffic or handover observations among all RRHs during one hour, and can be denoted as $F_t = \mathcal{F}(:,t)$ or $H_t = \mathcal{H}(:,:,t)$, respectively. Consequently, a set of consecutive traffic snapshots can then be represented as $\{F_t, F_{t-1}, \ldots, F_{t-N_s}\}$, and a set of consecutive RRH handover snapshots as $\{H_t, H_{t-1}, \ldots, H_{t-N_s}\}$, where N_s is the number of snapshots.

Inputs: we then extract the appropriate inputs for the LSTM layers based on the snapshots. For traffic, we simply stack N_r RRH traffic observations in each snapshot to form an input vector, and select N_s snapshots as the look-back time steps [192]. For handover, since there are $N_r \times N_r$ handover pairs, directly constructing an input vector with such high dimension will be computational impossible for the LSTMs. In fact, many of the RRH pairs do not observe meaningful handover counts since they are geographically distant from each other. Therefore, we adopt a hypothesis-based method to select RRH pairs with statistical significant handover counts. Based on our observation from the dataset, a significant handover count series exhibits large variations (i.e., over-dispersion) [193], thus we make a hypothesis that the handover count of an RRH pair follows the negative binomial distribution [194]. We test each RRH pair again this hypothesis and remove failure pairs. In this way, we obtain N_h pairs of RRHs with significant handover counts. We stack the pairs in each snapshot to form an input vector, and use N_s snapshots as look-back time steps for the LSTM layers.

LSTM layers: in the LSTM layers, we build two LSTM neural networks for the traffic and handover inputs, respectively. The traffic LSTM accepts an input of N_r traffic variables with N_s time steps, while the handover LSTM accepts an input of N_h handover pairs with N_s time steps, respectively. As illustrated in Figure 6.4, for each time step, the hidden unit s_t in the network computes its current activation o_t as a nonlinear function of both the current input weights U and the weights from the previous state W. In this way, the networks are able to keep a memory of the previous perception and use the knowledge for current decision making.

Predictions: the LSTM layers output the RRH traffic volume and handover count for the next time step as predictions. In order to exploit the correlation between RRH traffic and handover, we aggregate the outputs, and train the the two LSTMs jointly using the popular Backpropagation Through Time (BPTT) algorithm [192] for multiple iterations. The implementation details are elaborated in the evaluation section.

We run our prediction algorithm in an online manner, i.e., at the end of each time step t, we make a new prediction for the traffic and handover of $t+1$. We construct a tensor $\hat{\mathfrak{T}} \in \mathbb{R}^{N_r \times N_t}$ to store the traffic prediction, and a tensor $\hat{\mathcal{H}} \in \mathbb{R}^{N_r \times N_r \times N_t}$ to store the handover prediction, respectively. The prediction results are then used in the next phase for RRH-BBU mapping.

6.4 Dynamic Optimal RRH-BBU Mapping

In this phase, given the predicted RRH traffic and handover in a future period of time, as well as the BBU size and pool capacity constraints, our objective is to find optimal RRH-BBU mapping scheme to maximize BBU utilization rate and minimize RRH handover overhead. Such a problem has been identified as *set partitioning problem* [195, 196], and its complexity is proven to be NP-hard [197]. Therefore, exhaustively searching for every possible mapping scheme is computationally intractable as the network scale increases [28]. Some of the existing work use distributed methods such as coalition games, where the RRHs decide to form or leave clusters based on the gains of the strategies. However, such distributed methods are difficult for imposing global resource constraints in the centralized BBU pool [40] during the search procedure. In order to address these challenges, we propose a resource-constrained RRH-BBU mapping approach based on the weighted-graph model. We first introduce the system model and the problem formulation, and then propose the algorithm to effectively find optimal RRH-BBU mappings under the resource-constraints.

6.4.1 System Model

Based on the above-mentioned definitions, we model a mobile network as an undirected, weighted graph $G = (V, E)$, where $V = \{v_1, \ldots, v_{N_r}\}$ is the set of graph *nodes* denoting the N_r RRHs, and E is the set of graph *links* corresponding to the significant handover pairs among RRHs. We consider our dynamic RRH-BBU mapping problem in an *online* manner, i.e., at the end of time span t, we make decision of the RRH-BBU mapping scheme for the next time span $t+1$. To this end, we need to update the graph dynamically in each time span. We elaborate the details as follows.

Graph weights: we initialize the above-mentioned graph at $t+1$ as $G(t+1)$ with the traffic and handover predictions. Specifically, we define the *weight of node* $|v_i| = \hat{\mathcal{T}}(i, t+1)$, which corresponds to the traffic volume of the RRH i in time span $t+1$. Similarly, we define the *weight of link* $|e_{i,j}| = \hat{\mathcal{H}}(i, j, t+1)$, which is the handover count between RRH i and j in time span $t+1$. We note that if there is no predicted handover count between RRH i and j, then the link weight is set to 0, and we remove the corresponding link in $G(t+1)$. We consider the case of symmetric link weights ($|e_{i,j}| = |e_{j,i}|$) with no loops ($|e_{i,i}| = 0$).

Constraints: we model the resource constraints in the BBU pool according to the available BBU capacity. Since BBUs in the pools are implemented as virtual machine instances with specific sizes of computing resources, their capacities can be classified into a set of discrete levels. For example, we can denote a set of BBU capacity level as $\Phi = \{PICO, SMALL, MEDIAN, LARGE, \ldots\}$, each corresponding to a specific computing resource configuration. The capacity of an allocated BBU b_k shall be in one of the capacity levels, i.e., $\phi(b_k) \in \Phi$. Note that we do not explicitly constrain the number of available BBU with specific capacity level, since large-capacity BBUs (virtual machines) can be allocated by merging two or more small-capacity BBUs, and vice versa. Instead, we constrain the overall *capacity limit* of a BBU pool to be $\mathcal{O}$, since the capacity of a BBU pool is usually fixed once it is deployed. We study the impacts of different capacity level and capacity limit combinations in the evaluation section.

BBU utilization rate: once a BBU b_k is allocated to an RRH or a cluster of RRHs c_k in the time span $t+1$, its utilization rate can be calculated as

$$U(b_k) = \frac{\sum_{v_i \in c_k} |v_i|}{\phi(b_k)} \quad \in [0, 1] \tag{6.4.5}$$

where $\{v_i\}$ are the graph nodes corresponding to the cluster of RRHs mapped to the BBU. BBU utilization rate is one of the key objectives in optimizing our RRH-BBU mapping scheme. Since the BBU capacity $\phi(b_k)$ is discrete and the traffic generated in a cluster is continuous, it is important to ensure that the aggregated traffic volume in the cluster is close to the corresponding BBU capacity. Note that in order to avoid throughput congestion [28] and hurting the network quality of service, we do not allocate BBU for clusters whose aggregated traffic volume is larger than the maximum available BBU capacity in the pool. Thus, we constrain $U(b_k) \in [0,1]$ to avoid cluster traffic overflow.

Inner-BBU handover rate: the other key objective in RRH-BBU mapping optimization is to ensure that the handover events in $t+1$ are processed within a BBU (i.e., the corresponding RRHs are in a cluster) to a maximum extent. Therefore, we calculate the *inner-BBU handover rate* of cluster c_k as

$$W(c_k) = \frac{\sum_{v_i \in c_k} \sum_{v_j \in c_k} |e_{i,j}|}{2|E|} \quad \in [0,1] \tag{6.4.6}$$

where $\{e_{i,j}\}$ are the graph edges between the nodes that corresponding to the cluster of RRHs c_k, and $|E|$ corresponds to the total handover count between all RRH pairs during $t+1$.

6.4.2 Problem Formulation

With the above-mentioned system model, we now present the problem formulation for the BBU-RRH mapping problem with the objectives of maximizing BBU pool utilization and minimizing handover costs. Specially, given the graph representation $G(t+1)$ of a set of RRHs with the corresponding traffic and handover prediction, as well as the BBU pool resource constraints, our objective is to partition the graph into a set of N_k disjoint clusters $\mathbb{C} = \{c_1, \dots, c_{N_k}\}$, and map each cluster c_k to a BBU b_k in the BBU pool $\mathbb{B} = \{b_1, b_2, \dots, b_k\}$, with the following objective function and constraints:

$$\begin{aligned}
(\mathcal{P}_1): \quad & \text{maximize} \quad U(\mathbb{B}) * W(\mathbb{C}) && (6.4.7)\\
= & \text{maximize} \quad \frac{1}{N_k} \sum_{k=1}^{N_k} U(b_k) * \sum_{k=1}^{N_k} W(c_k) && (6.4.8)\\
= & \text{maximize} \quad \frac{1}{N_k} \sum_{k=1}^{N_k} \frac{\sum_{v_i \in c_k} |v_i|}{\phi(b_k)} * \sum_{k=1}^{N_k} \frac{\sum_{v_i \in c_k} \sum_{v_j \in c_k} |e_{i,j}|}{2|E|} && (6.4.9)
\end{aligned}$$

Subject to

$$\begin{aligned}
(\mathcal{C}_1): \quad & \cup_{\forall c_k \in \mathbb{C}} = V \quad \text{and} \quad \cap_{\forall C_k \in \mathbb{P}} = \emptyset && (6.4.10)\\
(\mathcal{C}_2): \quad & U(b_k) \in [0,1] && (6.4.11)\\
(\mathcal{C}_3): \quad & W(c_k) \in [0,1] && (6.4.12)\\
(\mathcal{C}_4): \quad & \phi(b_k) \in \Phi && (6.4.13)\\
(\mathcal{C}_5): \quad & \sum \phi(b_k) \leq \mathscr{O} && (6.4.14)
\end{aligned}$$

In this problem formulation, constraint $\mathcal{C}_1$ ensures that the clusters form a complete disjoint partition of the graph. Constraints $\mathcal{C}_2$–$\mathcal{C}_3$ are posed to avoid large clusters with aggregated traffic volume higher than the maximum available BBU capacity. Constraints $\mathcal{C}_4$–$\mathcal{C}_5$ make sure that the allocated BBU capacity can only be discrete values specified by the pool configuration, and their overall capacity can not exceed the resource limit $\mathscr{O}$.

6.4.3 Proposed Algorithm

The proposed problem $\mathcal{P}_1$ is indeed a graph partitioning problem (GPP) [198], which has been proved to be an NP-hard problem [196, 198]. To tackle this difficult problem, we resort to a fast heuristics approximation algorithm named label propagation (LP) [127, 130]. The basic idea of label propagation is to initialize each node in the graph as a cluster, and iteratively assign a node to its neighboring cluster based on a *gain* function [130]. However, directly applying an label propagation algorithm to our problem may not be adequate, since we also need to impose the resource constraints from the BBU pool, including available BBU capacity levels and pool capacity limit. Therefore, we propose a Resource-Constrained Label Propagation (RCLP) algorithm to solve this problem. We elaborate on the details as follows.

Algorithm: as shown in Algorithm 2, the RCLP algorithm is initialized by assigning each node in the graph to a unique cluster label. In each iteration, we randomly populate a list of node labels L, and traverse the list to update the cluster label of each node. The label update process is as follows. First, we remove the node from its current cluster, and find the set of adjacent clusters to the current node. Then, we compute the gain for adding the current node to the adjacent clusters, and assign it with the label of the cluster with the highest gain[2]. We mark the the node as *moved* among clusters if its new cluster label is different from the old one.

After we finish iterating over the node list, we evaluate whether the allocated resources of the resultant cluster partition are within the capacity limit of the BBU pool. If not, we reset the cluster labels and restart the optimization procedure. As the convergence of such a greedy algorithm is difficult to prove, we set a maximum iteration number *max_iter* to ensure the algorithm will stop. At the end of each iteration, we decide whether to perform another iteration or finish the algorithm based on the following stop criteria: (1) the user specified maximum iteration number *max_iter* is reached, or (2) none of the nodes are moved among clusters.

Gain function: the gain function is used to determine whether a node should be added to a adjacent cluster, and it shall take into consideration the improvement in both BBU utilization and handover performance. To this end, we first design the *utilization gain* of adding node v_i to cluster c_k as

$$gain\_u(v_i, c_k) = \max \Gamma(|v_i \cup c_k|, l), \quad l \in \mathbb{L} \tag{6.4.15}$$

[2] If two clusters yield the same gain, we randomly choose one.

Algorithm 2: The RCLP algorithm

```
Input: Graph $G(t+1) = (V,E)$, pool capacity limit $\mathcal{L}$, maximum iteration number $max\_iter$
Output: Cluster labels $L$ for nodes in the graph
▷ cluster label assignment
Initialize: $L \leftarrow 1,\dots,N$
while $(iter < max\_iter) \wedge (move > 0)$ do
    ▷ random permutation of nodes
    $rand\_perm(V)$;
    $move \leftarrow 0$ ;
    for $i \leftarrow 1$ to $N_r$ do
        ▷ remove current node from its cluster
        $old\_label \leftarrow L(v_i)$;
        $L(v_i) \leftarrow null$;
        ▷ select adjacent clusters
        $\mathbb{C}_{v_i} = get\_adjacent\_clusters(v_i, G, L)$;
        $max\_gain \leftarrow 0$;
        for $c \in \mathbb{C}_{v_i}$ do
            ▷ find cluster with highest gain
            $gain \leftarrow compute\_gain(v_i, c)$;
            if $gain \geq max\_gain$ then
                $new\_label \leftarrow L(c)$;
                $max\_gain \leftarrow gain$;
        ▷ update current node label
        $L(v_i) \leftarrow new\_label$
        if $old\_label \neq new\_label$ then
            $move \leftarrow 1$;
    ▷ reset labels if capacity limit exceeded
    if $allocated\_capacity(L) > \mathcal{L}$ then
        $L \leftarrow 1,\dots,N$
```

where

$$\Gamma(|v_i \cup c_k|, l) = \begin{cases} \dfrac{|v_i \cup c_k|}{l} & \text{if } |v_i \cup c_k| \leq l \quad (6.4.16) \\ -log(\dfrac{|v_i \cup c_k|}{l}), & \text{if } |v_i \cup c_k| > l \quad (6.4.17) \end{cases}$$

The rationale is like this: suppose we add v_i to c_k to form a candidate cluster $v_i \cup c_k$, we try to allocate BBUs with different capacity levels $l \in \mathbb{L}$ to the cluster. If the aggregated traffic of the candidate cluster does not exceed the BBU capacity, we calculate its utilization rate as Equation (6.4.16). Otherwise, we punish the candidate cluster with a *log* function (Equation 6.4.17) to avoid forming a cluster that no BBU can handle. Finally, we assign the maximum possible utilization rate to the candidate cluster with Equation 6.4.15.

Then, we define the *handover gain* of adding node v_i to cluster c_k as

$$gain\_h(v_i, c_k) = \frac{\sum_{v_k \in c_k} |e_{i,k}| + \sum_{v_{k_i} \in c_k} \sum_{v_{k_j} \in c_k} |e_{k_i,k_j}|}{2|E|} \tag{6.4.18}$$

which is a measurement of how strong the nodes in the new cluster $v_i \cup c_k$ are connected to each other. Finally, we define the *gain function* as the combination of both the utilization gain and the handover gain:

$$gain(v_i, c_k) = gain\_u(v_i, c_k) * gain\_h(v_i, c_k) \tag{6.4.19}$$

where $\mu \in (0, 1)$ controls the importance of each gain component.

Time Complexity: for each iteration of the RCLP algorithm, it first takes $O(|V|)$ steps for node permutation, and then processes all the links when computing the value function for each node, taking $O(|V| * |E|)$ steps in the worst case. Since we limit the number of iterations by *max_iter*, the final time complexity of the algorithm is $O(|V| * |E|)$.

6.5 Evaluation

We evaluate the performance of our framework using real-world mobile network traffic datasets. We focus on the accuracy of the traffic and handover prediction method, and the statistical gain of the RRH-BBU mapping schemes. In this section, we first describe the experiment settings, and then present the evaluation results and case studies.

6.5.1 Dataset Description

We exploit two large-scale anonymized datasets released by Orange Group in the D4D challenges [84, 85]. The datasets contain call detail records from Orange customers from Ivory Coast for half-a-year, and Senegal in one year, respectively. Based upon this, we extract two city-scale datasets for *Abidjan* and *Dakar*, the two largest cities in Ivory Coast and Senegal, respectively. We perform data cleansing to remove missing and incomplete data. Specifically, we exclude base stations with no traffic or handover records, and compile two datasets containing the base station positions, call durations, and handover counts. The details of these two datasets are listed in Table 6.1.

We assume the cloud-RAN architecture is deployed in the two cities during the data collection. Specifically, the RRHs are placed in the base station sites, and the centralized BBU pools are deployed and connected to the RRHs via high speed optical fiber. We quantify the RRH traffic based on the aggregated radio resource units [28] allocated to the phone calls, which is proportional to the

Table 6.1: Datasets Description

City	Abidjan	Dakar
Area	422 km^2	83 km^2
Population	4,707,404	1,146,053
Base stations	270	257
Duration	20 weeks 12/05/2011–04/22/2012	50 weeks 01/07/2013–12/22/2013
Average call duration	5.18 minutes	6.82 minutes
Handover events per hour	78,662	113,082

total call durations in each hour[3]. Due to privacy concerns, the mobility data is randomly sampled from a portion of Orange customers (1% for Ivory Coast and 3.33% for Senegal, respectively) [84, 85], therefore we estimate the actual handover count by multiplying the sample rate.

6.5.2 Evaluation on Prediction Accuracy

We first evaluate the accuracy of traffic and handover prediction in the first phase. We detail the evaluation plan, metrics and baseline methods as follows, and present the evaluation results.

Evaluation Plan: we use 70% of the datasets for model training, and the remaining 30% for testing. For each part, we use the first N_s time steps (hours) as input and the next one time step (hour) as the desired output. We then use the training set to train the MuLSTM model. Since the traffic and handover patterns are quite different during weekdays and weekends, we separately train a weekday and a weekend model using the corresponding datasets, respectively. In the MuLSTM model, we implement the LSTM layers using an encoder-decoder architecture. More specifically, the encoder layer L_1 contains $N_{encoder}$ memory units, which accepts as input an array of traffic or handover vectors of N_s time steps, and outputs an encoded sequence for the decoder. The decoder contains $N_{decoder}$ memory units, which accepts as input the encoded sequence and outputs the traffic or handover forecast.

Model Training: we use the popular Tensorflow [151] library for constructing our deep-learning model. We train the MuLSTM model for N_{iter} iterations to ensure that the network learns the potential temporal and spatial structures of the traffic and handover patterns. Based on a series of empirical experiments, we choose the optimal $N_s = 12$ hours, $N_{encoder} = N_{decoder} = 32$, and $N_{iter} = 10,000$. The model is trained on a 64-bit server with an NIVIDA GeForce GTX 1080 graphic card and 16GB of RAM. Each training iteration takes about 1.5 seconds and the whole

[3] We note that if fine-grained network traffic data, such as video stream, are available, our solution can easily adapt to the optimization task with regard to each specific traffic type.

process takes 4.2 hours.

Evaluation Metrics: in the model test period, we use the trained MuLSTM model to predict the city-wide traffic volume and handover count at the beginning of each hour, and compare the results with the ground truth data. We design the following evaluation metrics to evaluate the RRH traffic and handover prediction accuracy.

For RRH traffic prediction, we compare the predicted traffic snapshot $\hat{\mathcal{F}}(:,t)$ with the ground truth data $\mathcal{F}(:,t)$ in the test set of size N_{test}, and calculate the Mean Absolute Percentage Error (MAPE) for each snapshot:

$$MAPE_f = \frac{1}{N_{test}} \sum_{t=1}^{N_{test}} |\frac{\mathcal{F}(:,t) - \hat{\mathcal{F}}(:,t)}{\mathcal{F}(:,t)}| \times 100\% \tag{6.5.20}$$

Similarly, for RRH handover prediction, we compare the predicted handover snapshot $\hat{\mathcal{H}}(:,:,t)$ with the ground truth data $\mathcal{H}(:,:,t)$ in the test set of size N_{test}, and calculate the Mean Absolute Error (MAE) for each snapshot:

$$MAPE_h = \frac{1}{N_{test}} \sum_{t=1}^{N_{test}} |\frac{\mathcal{H}(:,:,t) - \hat{\mathcal{H}}(:,:,t)}{\mathcal{H}(:,:,t)}| \times 100\% \tag{6.5.21}$$

Baseline Methods: we design the following prediction baselines to compare with the proposed deep-learning-based approach.

- **ARIMA**: this baseline method models the traffic of each RRH as a time series, and uses the traditional ARIMA model [199] for traffic prediction. Similarly, it models each significant handover sequence as a time series, and builds individual ARIMA models for handover prediction, respectively.
- **MuANN**: different from the previous two baselines, this baseline method models the RRHs in the network as a whole, and adopts the same architecture as the proposed MuLSTM model except that the predictors are implemented using ANN layers. Since the ANN layers do not have an internal temporal state and thus are not able to model the temporal dependencies among different time steps, the inputs of this method are the traffic and handover snapshots in *only one* time step.

Evaluation Results: Figure 6.5 shows the results of traffic and handover prediction using the baseline and proposed methods, respectively. Each method is evaluated on weekdays, weekends, and all days. In RRH traffic prediction, the proposed MuLSTM method achieves an MAPE of 6.08% for all days, which is much lower than the MAPE of ARIMA (13.23%) and MuANN (9.08%) methods. The possible reason is that the ARIMA method models the temporal dependency of RRH traffic, but fails to capture the correlations among RRHs. Meanwhile, the MuANN method models the RRH correlations in the ANN layers, but is not able to capture the temporal dynamics of RRH traffic. In contrast, the proposed MuLSTM method models the temporal dynamics and spatial correlations simultaneously to achieve lower prediction errors. Furthermore, by separately training different predictive models for weekdays and weekends, the prediction errors can be reduced for the ARIMA and MuLSTM methods, since the temporal patterns can be modeled in a fine-grained manner for these two methods.

We show two illustrative examples of RRH traffic handover predictions using the proposed MuLSTM models (without weekday and weekend separation) in Figure 6.5(b) and Figure 6.5(d), respectively. The example RRH is located in Plateau, the downtown area of Abidjan, Ivory Coast. We can see that the proposed methods successfully learn the dynamic weekday and weekend patterns in both traffic and handover dynamics, and make accurate predictions based on the temporal and spatial factors.

6.5.3 Evaluation on RRH-BBU Mapping

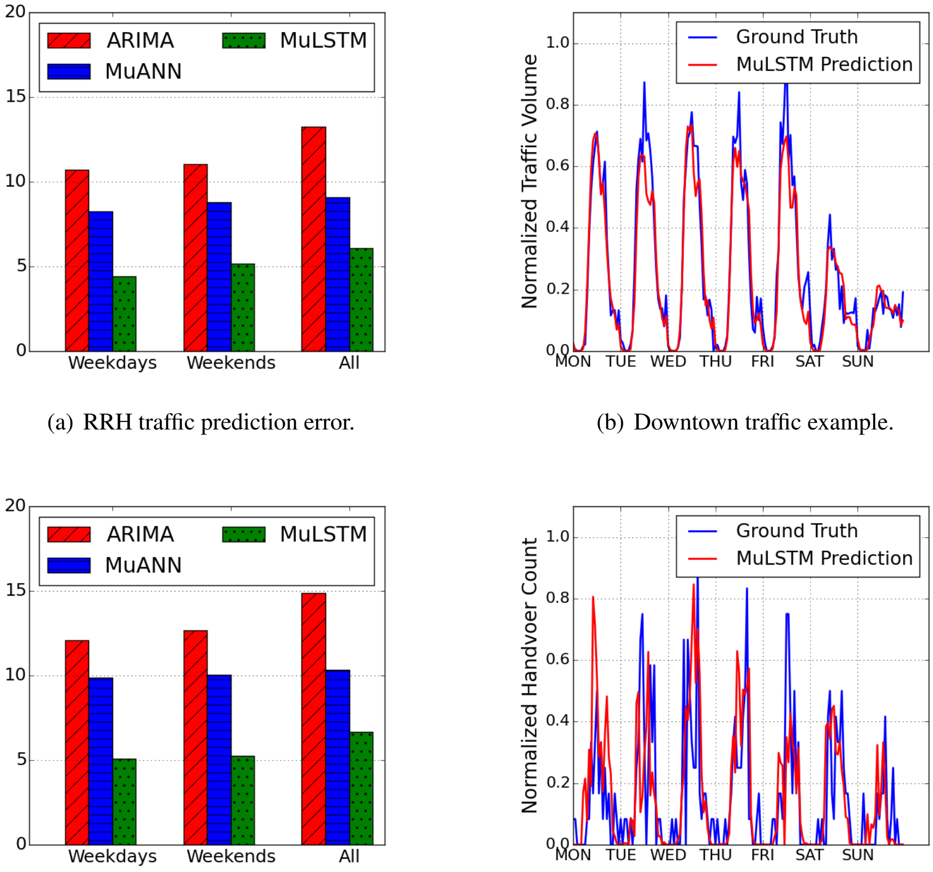
With the predicted RRH traffic and handover in the test set, as well as the BBU size and pool capacity constraints, we evaluate the effectiveness of the proposed method on finding optimal RRH-BBU mapping schemes. Specifically, we run the mapping algorithms for each hour in the test set and calculate the statistical multiplex gains for comparison. We elaborate the parameter selection, evaluation metrics, baseline methods and evaluation results as follows.

Parameter Selection: the most important parameter in the RRH-BBU mapping phase is the *BBU size* in the BBU pool. Since BBUs are implemented as virtual machines, their sizes are usually discrete values corresponding to predefined VM configurations (e.g., $PICO, SMALL, MEDIUM, LARGE$). However, the radio resource units occupied by RRHs are continuous. For example, Figure 6.6 shows the histogram of the radio resource units of all the RRHs in the training set of Abidjan, which ranges from 1×10^5 to 4×10^5 radio resource units. The desired BBU size needs to accommodate the demands of radio resource units occupied by both single RRH and RRH clusters. Based on previous studies [28,40] and empirical experiments, we design the BBU size category as a discrete set as follows:

$$\Phi = \{1RU, 2RU, 4RU, 8RU, \dots\} \tag{6.5.22}$$

where $1RU = 10^5$ radio resource units in this example. In this way, an RRH that occupies 1.5×10^5 radio resource units can allocate a BBU of size $2RU$, while a cluster of RRHs with an aggregated radio resource units of 10.5×10^5 can allocate a BBU of size $16RU$, respectively.

Evaluation Metrics: for an RRH-BBU mapping scheme that partition the RRHs into a set of N_k disjoint clusters $\mathbb{C} = \{c_1, \dots, c_{N_k}\}$ and map each cluster c_k to a BBU b_k in the BBU pool $\mathbb{B} = \{b_1, b_2, \dots, b_k\}$, we evaluate its statistical multiplex gains from the following two aspects.



(a) RRH traffic prediction error.

(b) Downtown traffic example.

(c) RRH handover prediction error.

(d) Downtown handover example.

Figure 6.5: Evaluation results of RRH traffic and handover prediction. Figure (b) and Figure (d) demonstrate illustrative examples of the traffic patterns and prediction results from a downtown RRH in Abidjan in one week (from 01/09/2012 to 01/15/2012).

For evaluating the improvement of BBU utilization, we define the *average utilization rate* of the BBU pool based on Equation 6.4.5, i.e.,

$$U(\mathbb{B}) = \frac{1}{N_k} \Sigma_{k=1}^{N_k} U(b_k) \tag{6.5.23}$$

Similarly, for evaluating the improvement of handover quality, we define the *inner-BBU handover rate* of the RRH clusters based on Equation 6.4.6, i.e.,

$$W(\mathbb{C}) = \Sigma_{k=1}^{N_k} W(c_k) \tag{6.5.24}$$

Baseline Methods: we design the following two sets of baselines to compare with the proposed

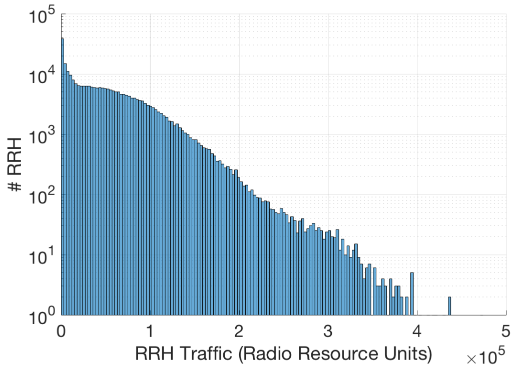


Figure 6.6: The histogram of RRH traffic in the training set of Abidjan, measured in radio resource units.

RRH-BBU mapping method. The first set of baselines adopt straight-forward RRH-BBU mapping schemes without clustering, i.e.,

- **ONE-TO-ONE**: in this baseline, we directly map one RRH to one BBU with maximal utilization rate under the BBU resource constraints. Since no BBUs are shared by RRHs, the inner-BBU handover rate is thus always zero.
- **ALL-TO-ONE**: in this baseline, we map all RRHs to a *virtual* BBU with the capacity of the pool limit $\mathscr{O}$. Since the BBU is shared among all RRHs, the inner-BBU handover rate is thus always 100%.

The second set of baselines adopt the same *RCLP* clustering algorithm as the proposed method, while differs in the design of the optimization objectives, i.e.,

- **UTIL-RCLP**: this baseline method finds RRH-BBU mapping schemes that maximize the BBU utilization rate without considering the inner-BBU handover rate. The constraints and algorithm are the same as the proposed method.
- **HAND-RCLP**: this baseline method finds RRH-BBU mapping schemes that maximize the inner-BBU handover rate without considering the BBU utilization rate. The constraints and algorithm are the same as the proposed method.

Similarly, we name the proposed method as $DUAL-RCLP$, which optimize the BBU utilization rate and the inner-BBU handover rate at the same time.

Evaluation Results: Figure 6.5 shows the results of average utilization rate and inner-BBU handover rate on the test set using the baseline and proposed methods, respectively. We can see that the *ONE-TO-ONE* baseline method achieves lowest utilization rate and inner-BBU handover rate, since each RRH is allocated a separated BBU in the pool, which not only wastes unused

Table 6.2: Evaluation Results of the RRH-BBU Mapping Methods

Methods	**Abidjan**		**Dakar**	
	BBU Utilization	RRH Internal Handover	BBU Utilization	RRH Internal Handover
ONE-TO-ONE	58.7%	0%	49.8%	0%
ALL-TO-ONE	79.9%	100%	51.4%	100%
UTIL-RCLP	99.3%	1.64%	99.2%	0.77%
HAND-RCLP	60.5%	85.8%	62.1%	83.5%
DUAL-RCLP	**75.0%**	**84.6%**	**76.7%**	**81.1%**

BBU capacity, but also hinders handover events to be processed within a BBU. The *ALL-TO-ONE* baseline, on the other hand, achieves the highest BBU utilization rate and inner-BBU handover rate. By aggregating all the RRH traffic in a city into a huge, virtual BBU, the utilization rate can be improved and all the handover events can be processed within that BBU. However, we argue that such a huge, virtual BBU is technically difficult to implement and manage as a virtual machine in the BBU pool, and it may introduce risks to the availability and robustness of the whole network if the BBU is not responding or being damaged.

Results also show that the *UTIL-RCLP* and *HAND-RCLP* baseline methods achieve their optimization goals in both cities by adopting the *RCLP* algorithm. Specifically, the *UTIL-RCLP* method achieves relatively high utilization rate (above 99.2%) of BBUs, but fails to arrange RRHs with frequent handover events into clusters. In contrast, the *HAND-RCLP* method finds clusters with high inner-BBU handover rate (above 83.5%), but these clusters do not necessarily occupy the allocate BBU in an efficient manner (with a utilization rate about 61%).

Finally, the proposed *DUAL-RCLP* method consistently achieves a BBU utilization rate above 75.0% and an inner-BBU handover rate above 81.1% in both cities, validating the effectiveness of our method in finding cost-effective and high-quality RRH-BBU mapping schemes. We note that compared with the *HAND-RCLP* method, the proposed method only show a slight performance decrease in inner-BBU handover rate, while achieving a significant BBU utilization rate increase.

6.5.4 Case Studies

In order to further evaluate the effectiveness of our framework, we conduct a series of case studies in Abidjan and Dakar, respectively. In each case study, we showcase the traffic and handover snapshot in a specific scenario, and present the RRH-BBU mapping results on the map.

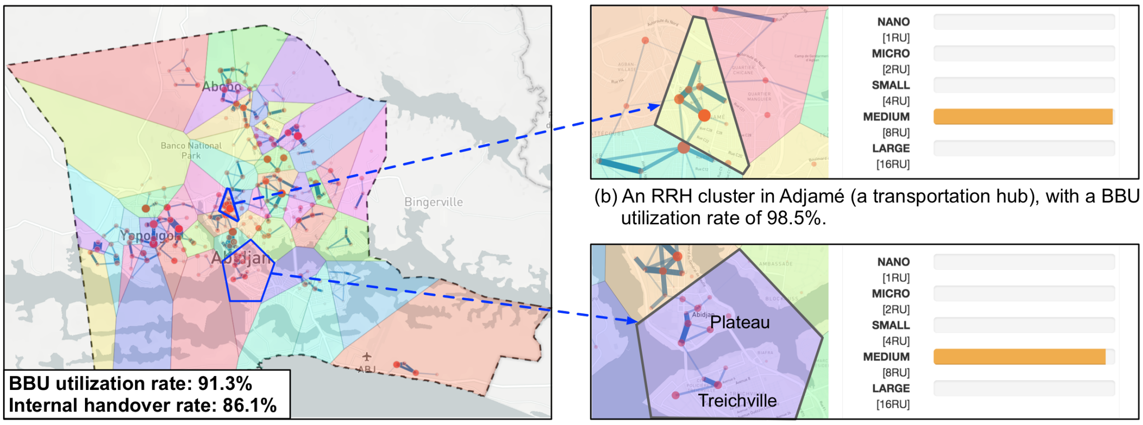
Abidjan Rush Hour: we select a typical weekday morning rush hour (9:00–10:00, 04/10/2012) in Abidjan from the test set for a case study. Figure 6.7(a) shows the RRH traffic and handover patterns during the rush hour, where larger dots denote RRHs with higher traffic volume, and thicker lines correspond to more handover events observed between the two corresponding RRHs. We also visualize the RRH-BBU mapping scheme using a Voronoi diagram [200] in Figure 6.7(a), where each polygon corresponds to a RRH cluster.

In Figure 6.7(a), we can see that during the morning rush hour, the network traffic of the city are mainly generated from the residential areas, the business districts, and the transportation hubs. Correspondingly, the handover events are frequently observed in these areas. Our framework successfully find an optimal RRH-BBU scheme with an average BBU utilization rate of 91.3% and an RRH internal handover rate of 86.1%. In order to further elaborate on the clustering results, we examine two of the typical clusters in this scheme as follows.

- Figure 6.7(b) shows the traffic and handover patterns of a cluster in Adjamé, a transportation hub of Abidjan. Adjamé has several important bus stations from where buses serve the greater Abidjan area as well as all of Ivory Coast. In the morning rush hour, large crowds of commuters and long distance travelers arrive at and depart from this area, generating significant handover events among the RRHs in this area, as well as large traffic volume. Our method successfully identify this RRH cluster and assign a medium-size BBU (8RU) to it, and thus achieving a high BBU utilization rate of 98.8%, as shown in Figure 6.7(b).
- Figure 6.7(c) shows a hybrid cluster formed by RRHs in Plateau and Treichville. Plateau is the central business district of Abidjan, and Treichville is one of the most populated suburban residential areas in Abidjan. In the morning rush hour, significant traffic volumes are observed in the RRHs of Plateau and Treichville, respectively, most probably generated by the residents, commuters, and workers in these areas. By sharing a medium-size BBU with 8RU, the cluster of RRHs in these areas achieves a high BBU utilization rate of 94.9%. More importantly, the large volume of handover events between Plateau and Treichville during the rush hour can be processed within the BBU, which significant improves the handover quality.

Dakar Independence Day: in Dakar, we investigate the RRH-BBU mapping scheme during the morning hours of the 2013 Senegal Independence Day (04/04/2013 10:00–11:00), and compare it with the scheme during the morning hours of a typical weekday (04/11/2013 10:00–11:00, one week later). Figure 6.8 shows the RRH traffic and handover patterns on the two days, as well as the RRH-BBU mapping scheme denoted by a Voronoi diagram.

In Senegal, the Independence Day is celebrated as a public holiday. In Figure 6.8(a), we can see that during the morning hours of that day, most traffic and handover events are generated in the central and northern parts of Dakar, which correspond to the city's residential neighborhoods, restaurants, and parks, etc. Consequently, our framework identifies these communities and allocate high-capacity BBUs for the corresponding RRH clusters. For example, Figure 6.8(a) shows two



(a) An RRH-BBU mapping scheme in Abidjan during the morning rush hour (04/10/2012 9:00–10:00).

(b) An RRH cluster in Adjamé (a transportation hub), with a BBU utilization rate of 98.5%.

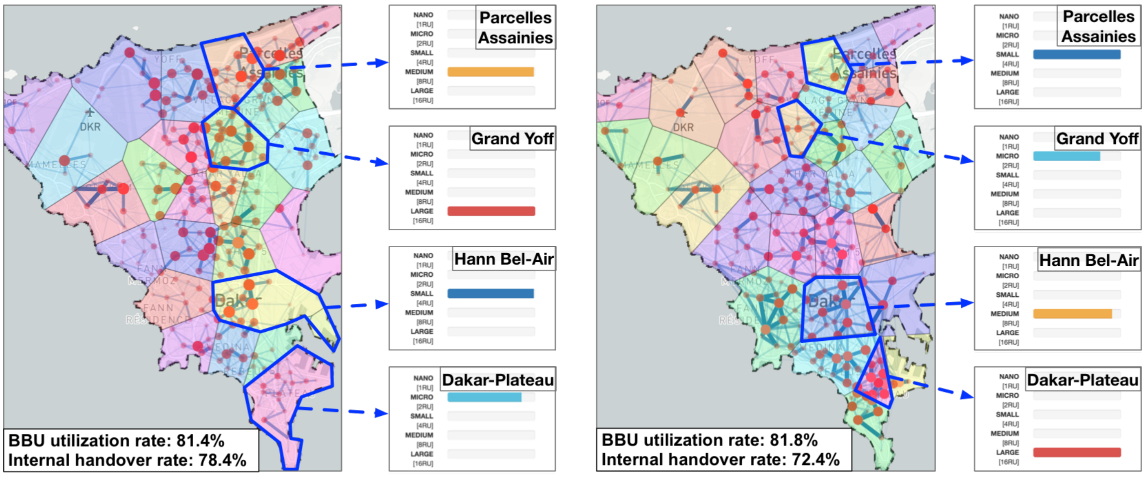
(c) An RRH cluster in Plateau (a business district) and Treichville (a residential area), with a BBU utilization rate of 94.9%.

Figure 6.7: The traffic and handover patterns of a cluster in Adjamé, a transportation hub of Abidjan.

RRH clusters in *Parcelles Assainies* and *Grand Yoff*, two of the largest residential neighborhoods in Dakar, as well as the allocated BBU capacity, respectively. In contrast, the southern parts of the city, including *Hann Bel-Air* and *Dakar-Plateau*, are the central industrial, business and administrative districts of Dakar. On the Independence Day, these areas observe relatively fewer user activities due to public holidays. Consequently, our framework tends to form large clusters consisting of many RRHs to reduce handover cost, while allocating BBUs with relatively small capacities since the aggregated traffic volumes are insignificant. For example, Figure 6.8(a) illustrates two clusters in *Hann Bel-Air* (the port and industrial zone) and *Dakar-Plateau* (the business and administrative center) and the allocated BBUs, respectively. We can see that these two clusters occupy large geographic areas with many RRHs, however the small and micro size BBUs are already adequate to process the traffic. In this way, our framework achieves an average BBU utilization rate of 81.4% and an internal handover rate of 78.4%, respectively.

We also present the RRH-BBU mapping scheme in the morning hours of a typical weekday (one week later) for comparison. From Figure 6.8(b), we can see that during the weekday morning, a large number of RRHs in the southern parts of the city observe significant traffic volume and handover events. Correspondingly, our framework identifies clusters with densely connected RRHs in *Hann Bel-Air* and *Dakar-Plateau*, and allocate BBUs with high capacities for them. In contrast, the clusters formed in the residential areas (e.g., *Parcelles Assainies* and *Grand Yoff*) do not observe significant user activities, and thus the BBUs allocated to them are of lower capacities. Similarly, our framework effectively increase the average BBU utilization rate to 81.8% and achieves an internal handover rate of 72.4%, respectively.

In summary, by adaptively forming different sizes of clusters and allocating BBUs with ade-



(a) An RRH-BBU mapping scheme in Dakar during the morning hours of the Independence Day (04/04/2013 10:00–11:00).

(b) An RRH-BBU mapping scheme in Dakar during the morning hours of a typical weekday (04/11/2013 10:00–11:00).

Figure 6.8: (The RRH-BBU mapping scheme during the morning hours of the 2013 Senegal Independence Day (04/04/2013 10:00–11:00), in comparison with the scheme during the morning hours of a typical weekday (04/11/2013 10:00–11:00, one week later).

quate capacities, our framework effectively improves the BBU utilization rate and handover performance in the Cloud-RAN architecture.

6.6 Conclusion

In this work, we identify two of the most important goals in cloud-RAN optimization, i.e., increasing capacity utilization rate and reducing energy consumption of the entire network. We proposed a deep-learning-based framework to help achieve these goals in cloud-RAN. Specially, we dynamically forecast the traffic of base stations using a multivariate LSTM model, and then cluster the complementary base stations into the same BBU pools based on the traffic forecast. The proposed MuLSTM model is capable of modeling the temporal dependence and spatial correlation between RRHs in the network, and DCCA clustering algorithm is effective in finding optimal clustering schemes for base stations by considering both the distance constraints and the complementarity metric. Real-world evaluation results show that our framework effectively increases 24.74% of utilization rate and reduces 12.05% of energy consumption compared with traditional RAN architectures, and outperforms the state-of-the-art baseline methods.

In the future, we plan to improve this work in the following directions. Firstly, we plan to explore the methods for dynamic capacity adjustment in BBU pools to adapt to the traffic demands. Secondly, we plan to evaluate our framework in broader areas, such as Trentino Italy, and to study the traffic complementarity property under different contexts.

Chapter 7

Conclusion and Future Work

Contents

In this final chapter, we summarize the main contributions of this dissertation, and then discuss the future research opportunities.

7.1 Summary of Contributions

In this dissertation, we explore a *big data-driven network optimization* paradigm to address the research challenges in the optimization of transportation and communication networks. Specifically, we propose two data-driven algorithms for network traffic clustering and user mobility prediction, and apply these algorithms to real-world optimization tasks transportation and communication networks. The detailed contributions are described as follows.

- **Dynamic traffic clustering for demand-responsive bikeshare networks.** In this application, we dynamically cluster bike stations with similar usage patterns to obtain stable and predictable cluster-wise bike traffic demands, so as to foresee over-demand stations in the network and enable demand-responsive bike scheduling. Evaluation results using real-world data from New York City and Washington, D.C. show that our framework accurately foresees over-demand clusters and outperforms other baseline methods significantly.
- **Complementary traffic clustering for cost-effective C-RAN.** In this application, we cluster RRHs with complementary traffic patterns (e.g., an RRH in residential area and an RRH in business district) to reuse the total capacity of the BBUs, so as to reduce the overall deploy-

ment cost. We evaluate our framework with real-world network data collected from the city of Milan, Italy and the province of Trentino, Italy. Results show that our method effectively reduces the overall deployment cost to 48.4% and 51.7% of the traditional RAN architecture in the two datasets, respectively, and consistently outperforms other baseline methods.

- **Spatio-temporal mobility prediction for anomaly-aware road networks.** In this application, we model the spatial correlations and temporal dependencies of of vehicle GPS trajectories in a unified spatio-temporal mobility model, and predict abnormal mobility events which may correspond to abnormal road conditions. Experiments with real-world data collected from Xiamen City show that our approach accurately predicts and identifies the road obstacles during the 2016 typhoon season with precision and recall both above 90%, and outperforms other baselines.
- **Deep mobility prediction for energy-efficient and quality-aware C-RAN.** In this application, we propose a deep-learning model to capture the spatio-temporal dynamics of user mobility in C-RAN, and accurately predict their movement patterns in next hour, so as to enable RRH cooperation to improve handover performance and increase BBU utilization. Real-world evaluations are conducted on two large-scale mobile network datasets collected from Ivory Coast and Senegal. Results show that our framework effectively increases the BBU utilization rate to more than 75.0% and achieve an RRH internal handover rate above 76.7%, which consistently outperforms the traditional RANs and other baseline methods.

7.2 Future Research Opportunities

This dissertation provides a step towards big data-driven optimization for urban network system leveraging large-scale real-world data. While several key issues and challenges have been studied, still a variety of research opportunities exist for further study. We charter the future work direction from potential optimization goals, other challenges and issues, new data analytics methods, and real-world deployment.

Optimization goals. Besides the key concerns on demand responsiveness, anomaly awareness, cost effectiveness, energy efficiency, and service quality, there are potentially large arrays of common optimization objectives that need further investigation. For example, privacy issues and data protection are a great concerns for urban authorities and network operators [201], demanding for novel strategies to prevent personal and individual mobility and communication information from being tracked, inferred, and leaked to third parties.

Challenges and issues. With the proposed data analytics algorithms, we have addressed several key challenges and issues in network optimization, including network profiling, mobility prediction, traffic clustering, and resource allocation. Still, there are many other issues in both networks we have not yet studied, such as opportunistic network resource allocation [202], routing and distributing traffic flows in networks [203], and the orchestration between the transportation and communication networks.

Data analytics algorithms. We have proposed a traffic clustering and a mobility prediction algorithm, and applied them to the network optimization applications. In fact, many other data analytics algorithms may be needed, including data-driven traffic classification [93], anomaly detection and causality inference [36], and privacy preservation [204]. The emerging deep-learning technologies can also be used to improve the performance of these algorithms.

Real-world deployment. Currently, we have only evaluated our methods using real-world datasets, but not yet being able to deploy our applications to real-world networks and systems. To implement and deploy a platform that can incorporate the techniques proposed in this dissertation as well as other state-of-the-art research advances, it will require extensive future endeavors from both academia and industry.

Bibliography

[1] United Nations, *World Urbanization Prospects: The 2014 Revision, Highlights*. 2014.

[2] G. Pflieger and C. Rozenblat, "Introduction. Urban Networks and Network Theory: The City as the Connector of Multiple Networks," *Urban Studies*, vol. 47, no. 13, no. 13, pp. 2723–2735, 2010.

[3] R. Z. Farahani, E. Miandoabchi, W. Y. Szeto, and H. Rashidi, "A review of urban transportation network design problems," *European Journal of Operational Research*, vol. 229, no. 2, no. 2, pp. 281–302, 2013.

[4] P. DeMaio, "Bike-sharing: History, impacts, models of provision, and future," *Journal of Public Transportation*, vol. 12, no. 4, no. 4, pp. 41–56, 2009.

[5] L. Rayle, D. Dai, N. Chan, R. Cervero, and S. Shaheen, "Just a better taxi? A survey-based comparison of taxis, transit, and ridesourcing services in San Francisco," *Transport Policy*, vol. 45, pp. 168–178, 2016.

[6] J.-P. Rodrigue, C. Comtois, and B. Slack, *The Geography of Transport Systems*. Routledge, 2006.

[7] V. R. Vuchic, *Urban Public Transportation; Systems and Technology*. 1981.

[8] B. Pishue, "INRIX US Traffic Hotspot Study 2017," tech. rep., INRIX, Seattle, USA, 2017.

[9] G. Sammer, *Travel Demand Management and Road User Pricing: Success, Failure and Feasibility*. Routledge, 2016.

[10] L. Chen, J. Jakubowicz, D. Yang, D. Zhang, and G. Pan, "Fine-Grained Urban Event Detection and Characterization Based on Tensor Cofactorization," *IEEE Transactions on Human-Machine Systems*, vol. 47, no. 3, no. 3, pp. 380–391, 2017.

[11] L. Chen, X. Fan, L. Wang, D. Zhang, Z. Yu, J. Li, T.-M.-T. Nguyen, G. Pan, and C. Wang, "RADAR: Road Obstacle Identification for Disaster Response Leveraging Cross-Domain Urban Data," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 4, no. 4, pp. 130:1–130:23, 2018.

[12] Z. Fan, X. Song, R. Shibasaki, and R. Adachi, "CityMomentum: An Online Approach for Crowd Behavior Prediction at a Citywide Level," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, pp. 559–569, ACM, 2015.

[13] D. Zhang, Z. Wang, B. Guo, and Z. Yu, "Social and Community Intelligence: Technologies and Trends," *IEEE Software*, vol. 29, no. 4, no. 4, pp. 88–92, 2012.

[14] G. Dimitrakopoulos and P. Demestichas, "Intelligent Transportation Systems," *IEEE Vehicular Technology Magazine*, vol. 5, no. 1, no. 1, pp. 77–84, 2010.

[15] M. Tubaishat, Y. Shang, and H. Shi, "Adaptive traffic light control with wireless sensor networks," in *Consumer Communications and Networking Conference, 2007. CCNC 2007. 4th IEEE*, pp. 187–191, IEEE, 2007.

[16] Y. Zheng, Y. Liu, J. Yuan, and X. Xie, "Urban Computing with Taxicabs," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp'11, pp. 89–98, ACM, 2011.

[17] A. Leon-Garcia and I. Widjaja, *Communication Networks*. New York, NY, USA: McGraw-Hill, Inc., 2 ed., 2004.

[18] C. Patrik, "Ericsson mobility report," tech. rep., Ericsson AB, Stockholm, Sweden, 2017.

[19] Cisco, "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2016–2021 White Paper," tech. rep., Cisco, San Jose, CA, USA, 2016.

[20] Y. Chen, S. Zhang, S. Xu, and G. Y. Li, "Fundamental trade-offs on green wireless networks," *IEEE Communications Magazine*, vol. 49, no. 6, no. 6, pp. 30–37, 2011.

[21] J. Research, "Mobile Operator Business Models: Challenges, Opportunities & Adaptive Strategies 2011-2016," tech. rep., Juniper Research, New York, 2011.

[22] D. Chalmers and M. Sloman, "A survey of quality of service in mobile computing environments," *IEEE Communications Surveys*, vol. 2, no. 2, no. 2, pp. 2–10, 1999.

[23] A. Checko, H. L. Christiansen, Y. Yan, L. Scolari, G. Kardaras, M. S. Berger, and L. Dittmann, "Cloud RAN for Mobile Networks - A Technology Overview," *IEEE Communications Surveys Tutorials*, vol. 17, no. 1, no. 1, pp. 405–426, 2015.

[24] G. Y. Li, Z. Xu, C. Xiong, C. Yang, S. Zhang, Y. Chen, and S. Xu, “Energy-efficient wireless communications: Tutorial, survey, and open issues,” *IEEE Wireless Communications*, vol. 18, no. 6, no. 6, pp. 28–35, 2011.

[25] I. F. Akyildiz, S. Mohanty, and J. Xie, “A ubiquitous mobile communication architecture for next-generation heterogeneous wireless systems,” *IEEE Communications Magazine*, vol. 43, no. 6, no. 6, pp. S29–S36, 2005.

[26] C. M. R. Institute, “C-RAN: The road toward Green RAN,” tech. rep., China Mobile Research Institute, Beijing, China, 2011.

[27] N. Nikaein, “Processing Radio Access Network Functions in the Cloud: Critical Issues and Modeling,” in *Proceedings of the 6th International Workshop on Mobile Cloud Computing and Services*, MCS ’15, (New York, NY, USA), pp. 36–43, ACM, 2015.

[28] H. Taleb, M. E. Helou, K. Khawam, S. Lahoud, and S. Martin, “Centralized and distributed RRH clustering in Cloud Radio Access Networks,” in *2017 IEEE Symposium on Computers and Communications (ISCC)*, pp. 1091–1097, 2017.

[29] A. Checko, A. P. Avramova, M. S. Berger, and H. L. Christiansen, “Evaluating C-RAN fronthaul functional splits in terms of network level energy and cost savings,” *Journal of Communications and Networks*, vol. 18, no. 2, no. 2, pp. 162–172, 2016.

[30] K. Dimou, M. Wang, Y. Yang, M. Kazmi, A. Larmo, J. Pettersson, W. Muller, and Y. Timner, “Handover within 3GPP LTE: Design Principles and Performance,” in *2009 IEEE 70th Vehicular Technology Conference Fall*, pp. 1–5, 2009.

[31] M. J. Neely, “Stochastic Network Optimization with Application to Communication and Queueing Systems,” *Synthesis Lectures on Communication Networks*, vol. 3, no. 1, no. 1, pp. 1–211, 2010.

[32] N. Pelekis, B. Theodoulidis, I. Kopanakis, and Y. Theodoridis, “Literature review of spatio-temporal database models,” *The Knowledge Engineering Review*, vol. 19, no. 3, no. 3, pp. 235–274, 2004.

[33] G. Del Mondo, M. A. Rodríguez, C. Claramunt, L. Bravo, and R. Thibaud, “Modeling consistency of spatio-temporal graphs,” *Data & Knowledge Engineering*, vol. 84, pp. 59–80, 2013.

[34] R. W. Thomas, D. H. Friend, L. A. DaSilva, and A. B. MacKenzie, “Cognitive Networks,” in *Cognitive Radio, Software Defined Radio, and Adaptive Wireless Systems* (H. Arslan, ed.), pp. 17–41, Springer Netherlands, 2007.

[35] C. Contardo, C. Morency, and L.-M. Rousseau, “Balancing A Dynamic Public Bike-Sharing System,” tech. rep., CIRRELT, 2012.

[36] S. Chawla, Y. Zheng, and J. Hu, “Inferring the Root Cause in Road Traffic Anomalies,” in *Proceedings of the IEEE International Conference on Data Mining*, ICDM’12, pp. 141–150, IEEE, 2012.

[37] Z. Zhao, M. Peng, Z. Ding, W. Wang, and H. V. Poor, “Cluster Content Caching: An Energy-Efficient Approach to Improve Quality of Service in Cloud Radio Access Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 34, no. 5, no. 5, pp. 1207–1221, 2016.

[38] A. N. Randriamanamihaga, E. Côme, L. Oukhellou, and G. Govaert, “Clustering the Vélib dynamic Origin/Destination flows using a family of Poisson mixture models,” *Neurocomputing*, vol. 141, pp. 124–138, 2014.

[39] J. Yuan, Y. Zheng, and X. Xie, “Discovering regions of different functions in a city using human mobility and POIs,” in *Proceedings of the 18th SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD’12, pp. 186–194, ACM Press, 2012.

[40] S. C. Zhan and D. Niyato, “A Coalition Formation Game for Remote Radio Head Cooperation in Cloud Radio Access Network,” *IEEE Transactions on Vehicular Technology*, vol. 66, no. 2, no. 2, pp. 1723–1738, 2017.

[41] D. Amzallag, M. Livschitz, J. Naor, and D. Raz, “Cell Planning of 4G Cellular Networks: Algorithmic Techniques and Results,” in *2005 6th IEE International Conference on 3G and Beyond*, pp. 1–5, 2005.

[42] D. Zhang, B. Guo, and Z. Yu, “The emergence of social and community intelligence,” *Computer*, vol. 44, no. 7, no. 7, pp. 21–28, 2011.

[43] Y. Zheng, L. Capra, O. Wolfson, and H. Yang, “Urban Computing: Concepts, Methodologies, and Applications,” *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 3, no. 3, pp. 1–55, 2014.

[44] M. Zhang, H. Fu, Y. Li, and S. Chen, “Understanding Urban Dynamics From Massive Mobile Traffic Data,” *IEEE Transactions on Big Data*, vol. PP, no. 99, no. 99, pp. 1–1, 2017.

[45] F. Calabrese, L. Ferrari, and V. D. Blondel, “Urban Sensing Using Mobile Phone Network Data: A Survey of Research,” *ACM Comput. Surv.*, vol. 47, no. 2, no. 2, pp. 25:1–25:20, 2014.

[46] J. Pucher, J. Dill, and S. Handy, “Infrastructure, programs, and policies to increase bicycling: An international review,” *Preventive Medicine*, vol. 50, pp. 106–125, 2010.

[47] S. Shaheen, H. Zhang, E. Martin, and S. Guzman, "China's Hangzhou public bicycle," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2247, no. 5, no. 5, pp. 33–41, 2011.

[48] P. Vogel, T. Greiser, and D. C. Mattfeld, "Understanding Bike-Sharing Systems using Data Mining: Exploring Activity Patterns," *Procedia - Social and Behavioral Sciences*, vol. 20, pp. 514–523, 2011.

[49] X. Zhou, "Understanding Spatiotemporal Patterns of Biking Behavior by Analyzing Massive Bike Sharing Data in Chicago," *PLoS ONE*, vol. 10, no. 10, no. 10, p. e0137922, 2015.

[50] C. Etienne and O. Latifa, "Model-Based Count Series Clustering for Bike Sharing System Usage Mining: A Case Study with the VÉLib' System of Paris," *Transactions on Intelligent Systems and Technology*, vol. 5, no. 3, no. 3, pp. 39:1–39:21, 2014.

[51] A. Kaltenbrunner, R. Meza, J. Grivolla, J. Codina, and R. Banchs, "Urban cycles and mobility patterns: Exploring and predicting trends in a bicycle-based public transport system," *Pervasive and Mobile Computing*, vol. 6, no. 4, no. 4, pp. 455–466, 2010.

[52] J. C. Garcia-Palomares, J. Gutierrez, and M. Latorre, "Optimizing the location of stations in bike-sharing programs: A GIS approach," *Applied Geography*, vol. 35, no. 1–2, no. 1–2, pp. 235–246, 2012.

[53] J.-R. Lin and T.-H. Yang, "Strategic design of public bicycle sharing systems with service level constraints," *Transportation Research Part E: Logistics and Transportation Review*, vol. 47, no. 2, no. 2, pp. 284–294, 2011.

[54] A. Singla, M. Santoni, G. Bartók, P. Mukerji, M. Meenen, and A. Krause, "Incentivizing Users for Balancing Bike Sharing Systems," in *Proceedings of the 29th AAAI Conference on Artificial Intelligence*, AAAI'15, (Austin, Texas), pp. 723–729, AAAI Press, 2015.

[55] P. Aeschbach, X. Zhang, A. Georghiou, and J. Lygeros, "Balancing Bike Sharing Systems through Customer Cooperation–A Case Study on London's Barclays Cycle Hire," in *Proc. IEEE CDC*, 2015.

[56] J. Froehlich, J. Neumann, and N. Oliver, "Sensing and Predicting the Pulse of the City through Shared Bicycling.," in *Proceedings of the International Joint Conference on Artificial Intelligence*, vol. 9, pp. 1420–1426, 2009.

[57] J. W. Yoon, F. Pinelli, and F. Calabrese, "Cityride: A Predictive Bike Sharing Journey Advisor," in *Proc MDM*, pp. 306–311, 2012.

[58] K. Gebhart and R. B. Noland, “The impact of weather conditions on bikeshare trips in Washington, DC,” *Transportation*, vol. 41, no. 6, no. 6, pp. 1205–1225, 2014.

[59] LDA Consulting, “2014 Capital Bikeshare Member Survey Report,” tech. rep., 2015.

[60] Capital Bikeshare, “Capital Bikeshare System Data.” http://www.capitalbikeshare.com/system-data, 2016.

[61] Y. Li, Y. Zheng, H. Zhang, and L. Chen, “Traffic Prediction in a Bike-sharing System,” in *Proc. ACM SIGSPATIAL*, pp. 33:1–33:10, 2015.

[62] D. Tuzun Aksu and L. Ozdamar, “A mathematical model for post-disaster road restoration: Enabling accessibility and evacuation,” *Transportation Research Part E: Logistics and Transportation Review*, vol. 61, pp. 56–67, 2014.

[63] FEMA, “Public Assistance: Debris Management Guide.,” tech. rep., Federal Emergency Management Agency, Washington, DC., 2007.

[64] S. E. Chang and N. Nojima, “Measuring post-disaster transportation system performance: The 1995 Kobe earthquake in comparative perspective,” *Transportation Research Part A: Policy and Practice*, vol. 35, no. 6, no. 6, pp. 475–494, 2001.

[65] M. Xie, L. Trassoudaine, J. Alizon, and J. Gallice, “Road obstacle detection and tracking by an active and intelligent sensing strategy,” *Machine Vision and Applications*, vol. 7, no. 3, no. 3, pp. 165–177, 1994.

[66] Y. LeCun, U. Muller, J. Ben, E. Cosatto, and B. Flepp, “Off-road Obstacle Avoidance Through End-to-end Learning,” in *Proceedings of the 18th International Conference on Neural Information Processing Systems*, NIPS’05, pp. 739–746, 2005.

[67] G. Lefaix, T. Marchand, and P. Bouthemy, “Motion-based obstacle detection and tracking for car driving assistance,” in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 4, pp. 74–77, IEEE, 2002.

[68] Z. Hasan, H. Boostanimehr, and V. K. Bhargava, “Green Cellular Networks: A Survey, Some Research Issues and Challenges,” *IEEE Communications Surveys Tutorials*, vol. 13, no. 4, no. 4, pp. 524–540, 2011.

[69] K. Lu, B. Rong, S. L. Kota, G. Liu, and X. Wang, “Next generation cognitive cellular networks: Spectrum sharing and trading [Guest editorial],” *IEEE Wireless Communications*, vol. 20, no. 2, no. 2, pp. 10–11, 2013.

[70] A. Munaretto and M. Fonseca, “Routing and quality of service support for mobile ad hoc networks,” *Computer Networks*, vol. 51, no. 11, no. 11, pp. 3142–3156, 2007.

[71] C. L. I, C. Rowell, S. Han, Z. Xu, G. Li, and Z. Pan, “Toward green and soft: A 5G perspective,” *IEEE Communications Magazine*, vol. 52, no. 2, no. 2, pp. 66–73, 2014.

[72] Y. Lin, L. Shao, Z. Zhu, Q. Wang, and R. K. Sabhikhi, “Wireless network cloud: Architecture and system requirements,” *IBM Journal of Research and Development*, vol. 54, no. 1, no. 1, pp. 4:1–4:12, 2010.

[73] J. G. Andrews, S. Buzzi, W. Choi, S. V. Hanly, A. Lozano, A. C. K. Soong, and J. C. Zhang, “What Will 5G Be?,” *IEEE Journal on Selected Areas in Communications*, vol. 32, no. 6, no. 6, pp. 1065–1082, 2014.

[74] A. Furno, D. Naboulsi, R. Stanica, and M. Fiore, “Mobile Demand Profiling for Cellular Cognitive Networking,” *IEEE Transactions on Mobile Computing*, vol. PP, no. 99, no. 99, pp. 1–1, 2016.

[75] J. Zhang, Y. Zheng, and D. Qi, “Deep Spatio-Temporal Residual Networks for Citywide Crowd Flows Prediction,” *Microsoft Research*, 2016.

[76] A. Furno, M. Fiore, and R. Stanica, “Joint Spatial and Temporal Classification of Mobile Traffic Demands,” 2017.

[77] S. Bhaumik, S. P. Chandrabose, M. K. Jataprolu, G. Kumar, A. Muralidhar, P. Polakos, V. Srinivasan, and T. Woo, “CloudIQ: A Framework for Processing Base Stations in a Data Center,” in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, Mobicom ’12, (New York, NY, USA), pp. 125–136, ACM, 2012.

[78] S. Namba, T. Matsunaka, T. Warabino, S. Kaneko, and Y. Kishi, “Colony-RAN architecture for future cellular network,” in *2012 Future Network Mobile Summit (FutureNetw)*, pp. 1–8, 2012.

[79] S. Namba, T. Warabino, and S. Kaneko, “BBU-RRH switching schemes for centralized RAN,” in *7th International Conference on Communications and Networking in China*, pp. 762–766, 2012.

[80] L. Liu, F. Yang, R. Wang, Z. Shi, A. Stidwell, and D. Gu, “Analysis of handover performance improvement in cloud-RAN architecture,” in *7th International Conference on Communications and Networking in China*, pp. 850–855, 2012.

[81] S. Tekinay and B. Jabbari, “Handover and channel assignment in mobile cellular networks,” *IEEE Communications Magazine*, vol. 29, no. 11, no. 11, pp. 42–46, 1991.

[82] D. Naboulsi, R. Stanica, H. Rivano, and M. Fiore, “User Mobility in Dynamic Cloud-Radio Access Networks,” tech. rep., 2017.

[83] D. Naboulsi, A. Mermouri, R. Stanica, H. Rivano, and M. Fiore, “On User Mobility in Dynamic Cloud Radio Access Networks,” pp. 1–9, 2018.

[84] V. D. Blondel, M. Esch, C. Chan, F. Clerot, P. Deville, E. Huens, F. Morlot, Z. Smoreda, and C. Ziemlicki, “Data for Development: The D4D Challenge on Mobile Phone Data,” *arXiv:1210.0137 [physics, stat]*, 2012.

[85] Y.-A. de Montjoye, Z. Smoreda, R. Trinquart, C. Ziemlicki, and V. D. Blondel, “D4D-Senegal: The Second Mobile Phone Data for Development Challenge,” *arXiv:1407.4885 [physics]*, 2014.

[86] G. Barlacchi, M. D. Nadai, R. Larcher, A. Casella, C. Chitic, G. Torrisi, F. Antonelli, A. Vespignani, A. Pentland, and B. Lepri, “A multi-source dataset of urban life in the city of Milan and the Province of Trentino,” *Scientific Data*, vol. 2, p. 150055, 2015.

[87] B. Guo, Z. Wang, Z. Yu, Y. Wang, N. Y. Yen, R. Huang, and X. Zhou, “Mobile Crowd Sensing and Computing: The Review of an Emerging Human-Powered Sensing Paradigm,” *ACM Computer Survey*, vol. 48, no. 1, no. 1, pp. 1–31, 2015.

[88] L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, and A. M’hamed, “Sparse mobile crowdsensing: Challenges and opportunities,” *IEEE Communications Magazine*, vol. 54, no. 7, no. 7, pp. 161–167, 2016.

[89] A. Furno, M. Fiore, R. Stanica, C. Ziemlicki, and Z. Smoreda, “A Tale of Ten Cities: Characterizing Signatures of Mobile Traffic in Urban Areas,” *IEEE Transactions on Mobile Computing*, vol. 16, no. 10, no. 10, pp. 2682–2696, 2017.

[90] B. Cici, M. Gjoka, A. Markopoulou, and C. T. Butts, “On the Decomposition of Cell Phone Activity Patterns and Their Connection with Urban Ecology,” in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc ’15, (New York, NY, USA), pp. 317–326, ACM, 2015.

[91] L. Chen, D. Zhang, L. Wang, D. Yang, X. Ma, S. Li, Z. Wu, G. Pan, T.-M.-T. Nguyen, and J. Jakubowicz, “Dynamic Cluster-based Over-demand Prediction in Bike Sharing Systems,” in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp’16, pp. 841–852, ACM, 2016.

[92] Z. Wang, D. Zhang, X. Zhou, D. Yang, Z. Yu, and Z. Yu, “Discovering and Profiling Overlapping Communities in Location-Based Social Networks,” *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 44, no. 4, no. 4, pp. 499–509, 2014.

[93] D. Naboulsi, R. Stanica, and M. Fiore, “Classifying call profiles in large-scale mobile traffic datasets,” in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 1806–1814, 2014.

[94] J. D. Hamilton, *Time Series Analysis*, vol. 2. Princeton university press Princeton, 1994.

[95] G. Dorffner, “Neural Networks for Time Series Processing,” *Neural Network World*, vol. 6, pp. 447–468, 1996.

[96] G. E. P. Box, G. M. Jenkins, G. C. Reinsel, and G. M. Ljung, *Time Series Analysis: Forecasting and Control*. John Wiley & Sons, 2015.

[97] G. P. Zhang, “Time series forecasting using a hybrid ARIMA and neural network model,” *Neurocomputing*, vol. 50, no. Supplement C, no. Supplement C, pp. 159–175, 2003.

[98] A. Sang and S.-q. Li, “A predictability analysis of network traffic,” *Computer Networks*, vol. 39, no. 4, no. 4, pp. 329–345, 2002.

[99] A. Andreoni and M. N. Postorino, “A multivariate ARIMA model to forecast air transport demand,” 2006.

[100] G. P. Zhang and M. Qi, “Neural network forecasting for seasonal and trend time series,” *European Journal of Operational Research*, vol. 160, no. 2, no. 2, pp. 501–514, 2005.

[101] A. Duncan, A. S. Chen, E. Keedwell, S. Djordjevic, and D. Savic, “Urban flood prediction in real-time from weather radar and rainfall data using artificial neural networks,” International Association of Hydrological Sciences, 2011.

[102] E. M. Azoff, *Neural Network Time Series Forecasting of Financial Markets*. New York, NY, USA: John Wiley & Sons, Inc., 1st ed., 1994.

[103] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *Nature*, vol. 521, no. 7553, no. 7553, pp. 436–444, 2015.

[104] L. Nie, D. Jiang, L. Guo, and S. Yu, “Traffic matrix prediction and estimation based on deep learning in large-scale IP backbone networks,” *Journal of Network and Computer Applications*, vol. 76, pp. 16–22, 2016.

[105] A. Rajkomar, E. Oren, K. Chen, A. M. Dai, N. Hajaj, P. J. Liu, X. Liu, M. Sun, P. Sundberg, H. Yee, K. Zhang, G. E. Duggan, G. Flores, M. Hardt, J. Irvine, Q. Le, K. Litsch, J. Marcus, A. Mossin, J. Tansuwan, D. Wang, J. Wexler, J. Wilson, D. Ludwig, S. L. Volchenboum, K. Chou, M. Pearson, S. Madabushi, N. H. Shah, A. J. Butte, M. Howell, C. Cui, G. Corrado, and J. Dean, “Scalable and accurate deep learning for electronic health records,” *arXiv:1801.07860 [cs]*, 2018.

[106] A. Jain, A. R. Zamir, S. Savarese, and A. Saxena, "Structural-RNN: Deep Learning on Spatio-Temporal Graphs," *ArXiv e-prints*, vol. 1511, p. arXiv:1511.05298, 2015.

[107] S. Shaheen, S. Guzman, and H. Zhang, "Bikesharing in Europe, the Americas, and Asia," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2143, no. 1, no. 1, pp. 159–167, 2010.

[108] A. M. Burden and R. Barth, *Bike-Share Opportunities in New York City*. New York: Department of City Planning, 2009.

[109] S. M. Kaufman, L. Gordon-Koven, N. Levenson, and M. L. Moss, "Citi Bike: The First Two Years," 2015.

[110] A. K. Dey, "Understanding and Using Context," *Personal and ubiquitous computing*, vol. 5, no. 1, no. 1, pp. 4–7, 2001.

[111] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles, "Towards a Better Understanding of Context and Context-Awareness," in *Handheld and Ubiquitous Computing*, no. 1707 in Lecture Notes in Computer Science, pp. 304–307, Springer Berlin Heidelberg, 1999.

[112] L. Chen, D. Yang, J. Jakubowicz, G. Pan, D. Zhang, and S. Li, "Sensing the Pulse of Urban Activity Centers Leveraging Bike Sharing Open Data," in *2015 IEEE 12th Intl Conf on Ubiquitous Intelligence and Computing and 2015 IEEE 12th Intl Conf on Autonomic and Trusted Computing and 2015 IEEE 15th Intl Conf on Scalable Computing and Communications and Its Associated Workshops (UIC-ATC-ScalCom)*, pp. 135–142, 2015.

[113] P. Midgley, "The role of smart bike-sharing systems in urban mobility," *Journeys*, vol. 2, pp. 23–31, 2009.

[114] Y. Zheng, "Methodologies for Cross-Domain Data Fusion: An Overview," *IEEE Transactions on Big Data*, vol. 1, no. 1, no. 1, pp. 16–34, 2015.

[115] D. Zhang, B. Guo, B. Li, and Z. Yu, "Extracting Social and Community Intelligence from Digital Footprints: An Emerging Research Area," in *Proc. UIC'10*, pp. 4–18, 2010.

[116] P. Langfelder and S. Horvath, "WGCNA: An R package for weighted correlation network analysis," *BMC Bioinformatics*, vol. 9, p. 559, 2008.

[117] Citi Bike, "Citi Bike System Data." http://www.citibikenyc.com/system-data, 2016.

[118] Weather Underground, "Weather API." https://www.wunderground.com/weather/api/, 2016.

[119] J. Dill and K. Voros, "Factors Affecting Bicycling Demand: Initial Survey Findings from the Portland, Oregon, Region," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2031, pp. 9–17, 2007.

[120] F. Zhang, D. Wilkie, Y. Zheng, and X. Xie, "Sensing the Pulse of Urban Refueling Behavior," in *Proceedings of the 15th ACM International Conference on Ubiquitous Computing*, (New York, NY, USA), pp. 13–22, ACM Press, 2013.

[121] Eventful, Inc, "Events Feed." https://api.eventful.com/, 2016.

[122] New York State, "511NY Data Feed." https://511ny.org/developers/resources, 2016.

[123] Twitter, "DC Police Traffic." https://twitter.com/DCPoliceTraffic, 2016.

[124] J. M. Stuart, E. Segal, D. Koller, and S. K. Kim, "A Gene-Coexpression Network for Global Discovery of Conserved Genetic Modules," *Science*, vol. 302, no. 5643, no. 5643, pp. 249–255, 2003.

[125] B. Zhang and S. Horvath, "A general framework for weighted gene co-expression network analysis," *Statistical applications in genetics and molecular biology*, vol. 4, no. 1, no. 1, 2005.

[126] M. A. Porter, J.-P. Onnela, and P. J. Mucha, "Communities in networks," *Notices of the AMS*, vol. 56, no. 9, no. 9, pp. 1082–1097, 2009.

[127] U. N. Raghavan, R. Albert, and S. Kumara, "Near linear time algorithm to detect community structures in large-scale networks," *Physical Review E*, vol. 76, no. 3, no. 3, p. 036106, 2007.

[128] M. E. J. Newman and M. Girvan, "Finding and evaluating community structure in networks," *Physical Review E*, vol. 69, no. 2, no. 2, p. 026113, 2004.

[129] J. Cohen, P. Cohen, S. G. West, and L. S. Aiken, *Applied Multiple Regression/Correlation Analysis for the Behavioral Sciences*. Routledge, 2013.

[130] M. Ciglan and K. Nørvåg, "Fast Detection of Size-Constrained Communities in Large Networks," in *Web Information Systems Engineering – WISE 2010*, no. 6488 in Lecture Notes in Computer Science, pp. 91–104, Springer Berlin Heidelberg, 2010.

[131] Y. Zheng, X. Yi, M. Li, R. Li, Z. Shan, E. Chang, and T. Li, "Forecasting Fine-Grained Air Quality Based on Big Data," in *Proceedings of the 21th SIGKDD Conference on Knowledge Discovery and Data Mining*, 2015.

[132] C. Z. Mooney, *Monte Carlo Simulation*. SAGE Publications, 1997.

[133] A. E. Raftery and V. E. Akman, “Bayesian Analysis of a Poisson Process with a Change-Point,” *Biometrika*, vol. 73, no. 1, no. 1, pp. 85–89, 1986.

[134] B. D. Ripley, *Stochastic Simulation*. John Wiley & Sons, 2009.

[135] H. F. Senter, “Applied linear statistical models,” *Journal of the American Statistical Association*, vol. 103, no. 482, no. 482, pp. 880–880, 2008.

[136] D. M. Powers, “Evaluation: From Precision, Recall and F-measure to ROC, Informedness, Markedness and Correlation,” 2011.

[137] J. A. Hayley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (ROC) curve.,” *Radiology*, vol. 143, no. 1, no. 1, pp. 29–36, 1982.

[138] L. Gordon-Koven and N. Levenson, “Citi Bike Takes New York,” *Rudin Center for Transportation Management and Policy*, 2014.

[139] The City of New York, “Summer Streets.” http://nyc.gov/summerstreets, 2016.

[140] GrowNYC, “Union Square Saturday Greenmarket.” http://www.grownyc.org/greenmarket/manhattan-union-square-sa, 2016.

[141] K. Zheng, Z. Yang, K. Zhang, P. Chatzimisios, K. Yang, and W. Xiang, “Big data-driven optimization for mobile networks toward 5G,” *IEEE Network*, vol. 30, no. 1, no. 1, pp. 44–51, 2016.

[142] T. Sigwele, A. S. Alam, P. Pillai, and Y. F. Hu, “Energy-efficient cloud radio access networks by cloud based workload consolidation for 5G,” *Journal of Network and Computer Applications*, vol. 78, pp. 1–8, 2017.

[143] P. Gandotra and R. K. Jha, “A survey on green communication and security challenges in 5G wireless communication networks,” *Journal of Network and Computer Applications*, vol. 96, pp. 39–61, 2017.

[144] L. Chen, T.-M.-T. Nguyen, G. Pan, J. Jakubowicz, L. Liu, X. Fan, J. Li, and C. Wang, “Complementary Base Station Clustering for Cost-Effective and Energy-Efficient Cloud-RAN,” 2017.

[145] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, 2005.

[146] M. Qian, W. Hardjawana, J. Shi, and B. Vucetic, “Baseband Processing Units Virtualization for Cloud Radio Access Networks,” *IEEE Wireless Communications Letters*, vol. 4, no. 2, no. 2, pp. 189–192, 2015.

[147] I. Sutskever, O. Vinyals, and Q. V. Le, “Sequence to Sequence Learning with Neural Networks,” in *Advances in Neural Information Processing Systems 27* (Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, eds.), pp. 3104–3112, Curran Associates, Inc., 2014.

[148] F. A. Gers, D. Eck, and J. Schmidhuber, “Applying LSTM to Time Series Predictable Through Time-Window Approaches,” in *Neural Nets WIRN Vietri-01*, Perspectives in Neural Computing, pp. 193–200, Springer, London, 2002.

[149] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Computation*, vol. 9, no. 8, no. 8, pp. 1735–1780, 1997.

[150] J. Lin, “Divergence measures based on the Shannon entropy,” *IEEE Transactions on Information Theory*, vol. 37, no. 1, no. 1, pp. 145–151, 1991.

[151] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, “TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems,” 2015. Software available from tensorflow.org.

[152] Centre for Research on the Epidemiology of Disasters, “The human cost of natural disasters 2015: A global perspective,” tech. rep., UN Office for Disaster Risk Reduction, Geneva, Switzerland, 2015.

[153] D. A. McEntire, *Disaster Response and Recovery: Strategies and Tactics for Resilience*. John Wiley & Sons, 2014.

[154] Y. Zheng, “Trajectory Data Mining: An Overview,” *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, no. 3, pp. 1–41, 2015.

[155] Y. Yu, J. Li, H. Guan, C. Wang, and C. Wen, “Bag of Contextual-Visual Words for Road Scene Object Detection From Mobile Laser Scanning Data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 12, no. 12, pp. 3391–3406, 2016.

[156] K. Nigam and R. Ghani, “Analyzing the Effectiveness and Applicability of Co-training,” in *Proceedings of the 9th International Conference on Information and Knowledge Management*, CIKM ’00, pp. 86–93, ACM, 2000.

[157] B. Settles, “Active learning literature survey,” tech. rep., University of Wisconsin, 2010.

[158] J. Lei, “Classification with confidence,” *Biometrika*, vol. 101, no. 4, no. 4, pp. 755–769, 2014.

[159] L. Chen, D. Zhang, X. Ma, L. Wang, S. Li, Z. Wu, and G. Pan, “Container Port Performance Measurement and Comparison Leveraging Ship GPS Traces and Maritime Open Data,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, no. 5, pp. 1227–1242, 2016.

[160] Y. Zheng, H. Zhang, and Y. Yu, “Detecting Collective Anomalies from Multiple Spatio-temporal Datasets Across Different Domains,” in *Proceedings of the ACM International Conference on Advances in Geographic Information Systems*, pp. 1–10, ACM, 2015.

[161] P. K. Chan and M. V. Mahoney, “Modeling multiple time series for anomaly detection,” in *Proceedings of the Fifth IEEE International Conference on Data Mining*, ICDM’05, pp. 8–16, ACM, 2005.

[162] Z. Fan, X. Song, and R. Shibasaki, “CitySpectrum: A Non-negative Tensor Factorization Approach,” in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp’14, pp. 213–223, ACM, 2014.

[163] L. Xiong, X. Chen, and J. Schneider, “Direct Robust Matrix Factorizatoin for Anomaly Detection,” in *Proceedings of the 11th IEEE International Conference on Data Mining*, ICDM’11, pp. 844–853, 2011.

[164] E. J. Candès, X. Li, Y. Ma, and J. Wright, “Robust Principal Component Analysis?,” *Journal of the ACM*, vol. 58, no. 3, no. 3, p. 1:37, 2011.

[165] T. Kolda and B. Bader, “Tensor Decompositions and Applications,” *SIAM Review*, vol. 51, no. 3, no. 3, pp. 455–500, 2009.

[166] J. Huang, F. Nie, H. Huang, and C. Ding, “Robust Manifold Nonnegative Matrix Factorization,” *ACM Transactions on Knowledge Discovery from Data*, vol. 8, no. 3, no. 3, pp. 1–21, 2014.

[167] P. Sprechmann, A. M. Bronstein, and G. Sapiro, “Learning Efficient Sparse and Low Rank Models,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, no. 9, pp. 1821–1833, 2015.

[168] H. Xiong, D. Zhang, G. Chen, L. Wang, V. Gauthier, and L. E. Barnes, “iCrowd: Near-Optimal Task Allocation for Piggyback Crowdsensing,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 8, no. 8, pp. 2010–2022, 2016.

[169] N. Srebro and A. Shraibman, "Rank, Trace-norm and Max-norm," in *Proceedings of the 18th Annual Conference on Learning Theory*, COLT'05, pp. 545–560, Springer, 2005.

[170] G. H. Golub and C. Reinsch, "Singular value decomposition and least squares solutions," *Numerische Mathematik*, vol. 14, no. 5, no. 5, pp. 403–420, 1970.

[171] D. Yang, D. Zhang, L. Chen, and B. Qu, "NationTelescope: Monitoring and visualizing large-scale collective behavior in LBSNs," *Journal of Network and Computer Applications*, vol. 55, pp. 170–180, 2015.

[172] J. Sander, M. Ester, H.-P. Kriegel, and X. Xu, "Density-based clustering in spatial databases: The algorithm gdbscan and its applications," *Data Mining and Knowledge Discovery*, vol. 2, no. 2, no. 2, pp. 169–194, 1998.

[173] N. D. Lane, P. Georgiev, and L. Qendro, "DeepEar: Robust Smartphone Audio Sensing in Unconstrained Acoustic Environments Using Deep Learning," in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp'15, pp. 283–294, ACM, 2015.

[174] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell, "Caffe: Convolutional Architecture for Fast Feature Embedding," in *Proceedings of the 22nd ACM International Conference on Multimedia*, MM'14, pp. 675–678, ACM, 2014.

[175] C.-C. Chang and C.-J. Lin, "LIBSVM: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology*, vol. 2, no. 3, no. 3, pp. 1–27, 2011.

[176] N. Rusli, M. R. Majid, and A. H. M. Din, "Google Earth's derived digital elevation model: A comparative assessment with Aster and SRTM data," *IOP Conference Series: Earth and Environmental Science*, vol. 18, no. 1, no. 1, pp. 12–65, 2014.

[177] L. Ewan, A. Al-Kaisy, and D. Veneziano, "Remote Sensing of Weather and Road Surface Conditions," *Transportation Research Record: Journal of the Transportation Research Board*, vol. 2329, pp. 8–16, 2013.

[178] K. L. Wolf, "Urban trees and traffic safety: Considering the U.S. roadside policy and crash data," *Arboriculture and Urban Forestry*, vol. 32, no. 4, no. 4, pp. 170–179, 2006.

[179] Y. Zheng, F. Liu, and H.-P. Hsieh, "U-Air: When Urban Air Quality Inference Meets Big Data," in *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '13, pp. 1436–1444, ACM, 2013.

[180] Y. Zhang, J. Wen, X. Wang, and Z. Jiang, “Semi-supervised learning combining co-training with active learning,” *Expert Systems with Applications*, vol. 41, no. 5, no. 5, pp. 2372–2378, 2014.

[181] L. Wang, D. Zhang, A. Pathak, C. Chen, H. Xiong, D. Yang, and Y. Wang, “CCS-TA: Quality-guaranteed Online Task Allocation in Compressive Crowdsensing,” in *Proceedings of the ACM International Joint Conference on Pervasive and Ubiquitous Computing*, (New York, NY, USA), pp. 683–694, ACM, 2015.

[182] N. Friedman, D. Geiger, and M. Goldszmidt, “Bayesian Network Classifiers,” *Machine Learning*, vol. 29, no. 2-3, no. 2-3, pp. 131–163, 1997.

[183] T. Fawcett, “An introduction to ROC analysis,” *Pattern Recognition Letters*, vol. 27, no. 8, no. 8, pp. 861–874, 2006.

[184] V. Chandola, A. Banerjee, and V. Kumar, “Anomaly Detection: A Survey,” *ACM Computer Survey*, vol. 41, no. 3, no. 3, pp. 1–58, 2009.

[185] A. Fehske, G. Fettweis, J. Malmodin, and G. Biczok, “The global footprint of mobile communications: The ecological and economic perspective,” *IEEE Communications Magazine*, vol. 49, no. 8, no. 8, pp. 55–62, 2011.

[186] M. Khan, R. S. Alhumaima, and H. S. Al-Raweshidy, “Reducing energy consumption by dynamic resource allocation in C-RAN,” in *2015 European Conference on Networks and Communications (EuCNC)*, pp. 169–174, 2015.

[187] K. Boulos, M. E. Helou, and S. Lahoud, “RRH clustering in cloud radio access networks,” in *2015 International Conference on Applied Research in Computer Science and Engineering (ICAR)*, pp. 1–6, 2015.

[188] J. Wu, Z. Zhang, Y. Hong, and Y. Wen, “Cloud radio access network (C-RAN): A primer,” *IEEE Network*, vol. 29, no. 1, no. 1, pp. 35–41, 2015.

[189] N. Yu, Z. Song, H. Du, H. Huang, and X. Jia, “Multi-resource allocation in cloud radio access networks,” in *2017 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2017.

[190] F. Simini, M. C. González, A. Maritan, and A.-L. Barabási, “A universal model for mobility and migration patterns,” *Nature*, vol. 484, no. 7392, no. 7392, pp. 96–100, 2012.

[191] G. M. Weiss, “Data Mining in Telecommunications,” in *Data Mining and Knowledge Discovery Handbook*, pp. 1189–1201, Springer, Boston, MA, 2005.

[192] D. P. Mandic and J. Chambers, *Recurrent Neural Networks for Prediction: Learning Algorithms,Architectures and Stability*. New York, NY, USA: John Wiley & Sons, Inc., 2001.

[193] J. M. Ver Hoef and P. L. Boveng, "Quasi-Poisson Vs. Negative Binomial Regression: How Should We Model Overdispersed Count Data?," *Ecology*, vol. 88, no. 11, no. 11, pp. 2766–2772, 2007.

[194] R. C. Tripathi, "Negative Binomial Distribution," in *Encyclopedia of Statistical Sciences*, John Wiley & Sons, Inc., 2004.

[195] E. Balas and M. Padberg, "Set Partitioning: A survey," *SIAM Review*, vol. 18, no. 4, no. 4, pp. 710–760, 1976.

[196] R. S. Garfinkel and G. L. Nemhauser, "The Set-Partitioning Problem: Set Covering with Equality Constraints," *Operations Research*, vol. 17, no. 5, no. 5, pp. 848–856, 1969.

[197] K. Hoffman and M. Padberg, "Set covering, packing and partitioning problems," in *Encyclopedia of Optimization*, pp. 2348–2352, Springer, Boston, MA, 2001.

[198] K. Andreev and H. Räcke, "Balanced Graph Partitioning," in *Proceedings of the Sixteenth Annual ACM Symposium on Parallelism in Algorithms and Architectures*, SPAA '04, (New York, NY, USA), pp. 120–124, ACM, 2004.

[199] C. Chatfield, *The Analysis of Time Series: An Introduction, Sixth Edition*. CRC Press, 2016.

[200] Okabe Atsuyuki, "Spatial Tessellations," *International Encyclopedia of Geography*, 2016.

[201] F. Giannotti, "Mobility, Data Mining and Privacy Understanding Human Movement Patterns from Trajectory Data," in *2011 IEEE 12th International Conference on Mobile Data Management*, vol. 1, pp. 4–5, 2011.

[202] R. F. Shigueta, M. Fonseca, A. C. Viana, A. Ziviani, and A. Munaretto, "A strategy for opportunistic cognitive channel allocation in wireless Internet of Things," in *2014 IFIP Wireless Days (WD)*, pp. 1–3, 2014.

[203] A. C. K. Vendramin, A. Munaretto, M. R. Delgado, M. Fonseca, and A. C. Viana, "A social-aware routing protocol for opportunistic networks," *Expert Systems with Applications*, vol. 54, pp. 351–363, 2016.

[204] L. Wang, D. Yang, X. Han, T. Wang, D. Zhang, and X. Ma, "Location Privacy-Preserving Task Allocation for Mobile Crowdsensing with Differential Geo-Obfuscation," in *Proceedings of the 26th International Conference on World Wide Web*, WWW '17, (Republic and Canton of Geneva, Switzerland), pp. 627–636, International World Wide Web Conferences Steering Committee, 2017.