UTS UNIVERSITY OF TECHNOLOGY SYDNEY

Mobile Edge Computing: From Task Load Balancing to Real-World Mobile Sensing Applications

by Xiaochen Fan

Thesis submitted in fulfilment of the requirements for the degree of

Doctor of Philosophy

under the supervision of Professor Xiangjian He

University of Technology Sydney
Faculty of Engineering and Information Technology

January 2021

Certificate of Authorship/Originality

I, Xiaochen Fan, declare that this thesis is submitted in fulfilment of the requirements for the award of Doctor of Philosophy, in the School of Electrical and Data Engineering, Faculty of Engineering and Information Technology, at the University of Technology Sydney.

This thesis is wholly my own work unless otherwise referenced or acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

This document has not been submitted for qualifications at any other academic institution.

This research is supported by the Australian Government Research Training Program and UTS-CSC International Research Scholarship (IRS).

Production Note:
Signature removed prior to publication.

Signature _____

Date 25 January 2021

Acknowledgements

The past four years at UTS have been an unforgettable and invaluable experience for me. First and foremost, I would like to express my sincerest gratitude to my supervisor, Professor Xiangjian (Sean) He, for his inspiring guidance, great encouragement, and continuous support. Professor He has always been insightful and perceptive in providing constructive suggestions and great advice to me in both research and philosophy. His passion and patience in academic, teaching and personal interests have motivated me to pursue an academia career. More importantly, Sean is an incredibly kind, considerate and supportive person, and I feel grateful to have him as my advisor, and I am privileged to work with him.

I would like to thank Priyadarsi Nanda and Deepak Puthal, two of my co-supervisors, for their professional views and constructive comments on my research study. Dr. Nanda is an extremely knowledgeable and responsible leader in the Cyber Security Research Group. I enjoy the research seminars and discussions with his team members. Dr. Puthal is a role model who has an incredibly strong commitment to research, and he has always been a mentor to me. Through our research collaborations, I have gained invaluable experience in many ways, particularly in technical writing and logical thinking. I want to thank the Cyber Security Research Group for the meetings and discussions as well.

I am also sincerely thankful to Wenjing Jia, an elegant, gentle and knowledgeable woman, for her generosity and conscientiousness in organising research meetings with Professor He's Computer Vision and Pattern Recognition (CVPR) group. Dr. Jia is highly initiative and detail-oriented in research, and her spirit for teamwork and willingness to help others have bonded all of the research students together as a family. I enjoy learning the advanced deep learning techniques in image processing and pattern recognition with Wenjing and the CVPR research group.

During my Ph.D. study, I have worked as a lab tutor for Gengfa Fang in *Sensing, Actuation and Control*. Dr. Fang has given tremendous support to his students, and he taught me what it takes to be an excellent educator. I have also participated in Gengfa's IoT Group and worked as a research assistant in a joint industry project with Cisco and SAS, *IoT Data Fabric and Analytics for Agile Trading of Distributed Energy Resources*. I believe what I have learned and experienced throughout the collaborative project with Gengfa will always benefit my future career.

I would extend my gratitude and appreciation to the professors, researchers, and colleagues at UTS. Special thanks to Yue Xi, Xudong Song, QingQing Wang, Edward Huang, Saeed Amirgholipour, Muhammad Usman, Qianwen Ye, Mian Jan, Hesam Hesamian, Ming Liu, Zheng Zhou, Manna Dai, Ashish Nanda, Caoyuan Li, Richard Xu, Haodong Chang, Farhan Mohammed, Nazar Waheed, Chengpei Xu, Yuanfang Zhang, who gave me a lot of support through at various times.

I am privileged to have opportunities to collaborate with different researchers, and I learned invaluable experience and knowledge with the following collaborators. Deepak Puthal — with his support and guidance, I finished my first research paper in Ph.D. Deepak has an incredibly progressive mind in academic research, and from his inspiring strivings I learned what perseverance is. Chaocan Xiang — with his guidance and support, I finished my first paper on IEEE Transactions on Mobile Computing. Dr. Xiang is an exceedingly ingenious person with substantial responsibility, and through teamwork with him, I have learned to develop innovative research ideas and focus on high-quality papers. Xudong Song and Qianwen Ye — through joint works with them, I started research on indoor wireless localisation and finished two papers with the proposed CNNLoc and CapsLoc frameworks. Liangyi Gong and Yuben Qu — two senior post-docs from Tsinghua University and Shanghai Jiaotong University, respectively. In our joint-discussions, they can always inspire me with insightful and high-level views about the research fields and provide detail-oriented understandings of the nature of research problems. Panlong Yang — Professor Yang was my supervisor in my Master's study, and he paved the way for me to enter the research world of computer science. I have always been grateful to

him for pushing me beyond what had expected of me. I believe that was an absolute necessity (quotes from the movie “Whiplash”).

Outside the research communities, I have been fortunate to have many great friends, and I cherish the countless happy moments we spent together in Sydney. Chang (Morgan) Xu, my close friend since he came to UTS as a visiting student from France. I learned a lot from Morgan in English communication etiquette, Sichuan cooking techniques (still missing that flavour), travel experience (he has been to major cities like New York, Paris, and London). Xucheng Zhu, my friendly neighbour at UTS housing, who was like a Sydneysider to provide me with useful information when I just settled in Sydney. Yue Xi, my labmate and brother, who gave me tremendous help and support, particularly when I got stuck in overseas life and business. Xudong Song, a brilliant young Ph.D. candidate and a competent programmer, who gave me technical guidance on deep learning and programming skills. Qingqing Wang, an intelligent and enthusiastic dual-Ph.D. student from Shanghai and an expert in deep learning, every discussion with Qingqing inspired me and humbled me to be more determined. Xiaohua Wei, Hanhui Li and Lei Liu, although they were visiting students in our lab, I enjoy the travels and outdoor activities with them to pull me out of stressful daily research study. I would like to extend sincere thanks to UTS alumni Raj Shukla, Xiaolin Zhang, Chris Li, Anita Tong, Huiyang Wang, Xingchi Chen, Blake Regan, Carlos Tirado, Yiliao Song, Xiaopu Zhang, Haiyan (Helen) Lu, Yang Yang, Xiaoshui Huang, Shiping Chen, Yaxin Shi, Zhanzhong Gu, Thomas Tan, Qiang Wu, Min Xu.

I would sincerely thank the China Scholarship Council and the University of Technology Sydney for providing me with scholarships, including the CSC Scholarship and the UTS-CSC International Research Scholarship.

I thank my parents: Jian Fan and Chunlan Shi, for their continuous love and support. Both of my parents work at Xinjiang University, and they made me who I am today with a kind heart. My parents are with me all these years, and I have a very close relationship with them — even I have left home for higher education and overseas study for more than ten years and can only spend 1-2 months staying

home with them every year. I hope that they would be a little proud of me for what I have achieved so far.

Lastly, I would like to thank my wife, Lu Lu, for her love and support. We were sixteen when we became the very best friends at Urumqi No.1 high-school. I still remember when we were walking around the school playground and talking about our dream universities, and that young spirits with engaging grit have been inspiring me all the time. Later on, we both made it to study in Beijing as she entered the Academy of Arts & Design at Tsinghua University, and I became a BITer in the School of Computer Science & Technology at the Beijing Institute of Technology. For over seven years in the past, she is not only my partner, my best friend and my love, but also the person I admire most for her elegance, wisdom, dedication, intelligence, and kindness. I would not have been able to study abroad for a Ph.D. degree without her understanding and inclusiveness. I thank Lu Lu for everything she has done for me.

Xiaochen Fan

Sydney, Australia, 2021

List of Publications

This thesis is based on the following publications [1–4]:

- **Chapter 3**

  Xiaochen Fan, Xiangjian He, Deepak Puthal, Shiping Chen, Chaocan Xiang, Priyadarsi Nanda, Xunpeng Rao, CTOM: Collaborative Task Offloading Mechanism for Mobile Cloudlet Networks, *in* 'Proceedings of IEEE International Conference on Communications (ICC)' (2018): 1-6.

- **Chapter 4**

  Shuang Lai, **Xiaochen Fan***, Qianwen Ye, Zhiyuan Tan, Yuanfang Zhang, Xiangjian He, Priyadarsi Nanda, FairEdge: A Fairness-Oriented Task Offloading Scheme for IoT Applications in Mobile Cloudlet Networks, *in* 'IEEE Access' 8 (2020): 13516-13526 (**Co-first Author and Corresponding author**).

- **Chapter 5**

  Xiaochen Fan, Xiangjian He, Chaocan Xiang, Deepak Puthal, Liangyi Gong, Priyadarsi Nanda, Gengfa Fang, Towards System Implementation and Data Analysis for Crowdsensing Based Outdoor RSS Maps, *in* 'IEEE Access' 6 (2018): 47535-47545.

- **Chapter 6**

  Xiaochen Fan, Chaocan Xiang, Chao Chen, Xudong Song, Panlong Yang, Liangyi Gong, Priyadarsi Nanda, Xiangjiang He, BuildSenSys: Reusing Building Sensing Data for Traffic Prediction with Cross-domain Learning, *in* 'IEEE Transactions on Mobile Computing' (Tier A*) (2020) Early Access, DOI identifier 10.1109/TMC.2020.2976936.

Other publications during the Ph.D. candidature [5–19]:

- Qianwen Ye, **Xiaochen Fan***, Gengfa Fang, Hongxia Bie, Xudong Song, Rajan Shankaran, CapsLoc: A Robust Indoor Localization System with WiFi Fingerprinting Using Capsule Networks, *in* 'Proceedings of IEEE International Conference on Communications (ICC)' (2020) (**Co-first Author**).

- Liangyi Gong, Chaocan Xiang, **Xiaochen Fan**, Tao Wu, Chao Chen, Miao Yu, Wu Yang, Device-free near-field human sensing using WiFi signals, *in* 'Springer Personal and Ubiquitous Computing' (2020): 1-14

- Chaocan Xiang, Zhao Zhang, Yuben Qu, Dongyu Lu, **Xiaochen Fan**, Panlong Yang, Fan Wu, Edge Computing-Empowered Large-scale Traffic Data Recovery Leveraging Low-rank Theory, *in* 'IEEE Transactions on Network Science and Engineering' (2020).

- Jiabin Li, Ming Liu, Zhi Xue, **Xiaochen Fan**, Xiangjian He, RTVD: A Real-Time Volumetric Detection Scheme for DDoS in the Internet of Things, *in* 'IEEE Access' (2020) 8: 36191-36201.

- **Xiaochen Fan**, Chaocan Xiang, Liangyi Gong, Xin He, Yuben Qu, Saeed Amirgholipour, Yue Xi, Priyadarsi Nanda, Xiangjian He, Deep Learning for Intelligent Traffic Sensing and Prediction: Recent Advances and Future Challenges, *in* 'CCF Transactions on Pervasive Computing and Interaction' (2020).

- **Xiaochen Fan**, Chaocan Xiang, Liangyi Gong, Xiangjian He, Chao Chen, Xiang Huang, UrbanEdge: deep learning empowered edge computing for urban IoT time series prediction, *in* 'Proceedings of the ACM Turing Celebration Conference-China' (2019): 1-6.

- Xudong Song, **Xiaochen Fan***, Xiangjian He, Chaocan Xiang, Qianwen Ye, Xiang Huang, Gengfa Fang, Liming Luke Chen, Jing Qin, Zumin Wang, C-NNLoc: Deep-Learning Based Indoor Localization with WiFi Fingerprinting, *in* 'Proceedings of IEEE International Conference on Ubiquitous Intelligence and Computing (UIC)' (2019): 589-595 (**Co-first Author**).

- Xudong Song, **Xiaochen Fan**, Chaocan Xiang, Qianwen Ye, Leyu Liu, Zumin Wang, Xiangjian He, Ning Yang, Gengfa Fang, A Novel Convolutional Neural Network Based Indoor Localization Framework With WiFi Fingerprinting, *in* 'IEEE Access' 7 (2019): 110698-110709.

- Ning Yang, **Xiaochen Fan***, Deepak Puthal, Xiangjian He, Priyadarsi Nanda, Shiping Guo, A Novel Collaborative Task Offloading Scheme for Secure and Sustainable Mobile Cloudlet Networks, *in* 'IEEE Access' (2018) 6: 44175-44189 (**Co-first Author**).

- Xunpeng Rao, Maotian Zhang, Wanru Xu, **Xiaochen Fan**, Hao Zhou, Panlong Yang, You Can Recharge With Detouring: Optimizing Placement for Roadside Wireless Charger, *in* 'IEEE Access' (2017) 6: 47-59.

- Chaocan Xiang, Panlong Yang, **Xiaochen Fan**, Liangyi Gong, Quantifying sensing quality of crowd sensing networks with confidence interval, *in* 'Proceedings of IEEE International Conference on Ubiquitous Intelligence and Computing (UIC)' (2017): 1-6.

- Qingyu Li, Panlong Yang, **Xiaochen Fan**, Shaojie Tang, Chaocan Xiang, Deke Guo, Fan Li, Taming the big to small: efficient selfish task allocation in mobile crowdsourcing systems, *in* 'Concurrency and Computation: Practice and Experience' (2017) 29(14): e4121

- Yue Xi, Wenjing Jia, Jiangbin Zheng, **Xiaochen Fan**, Xiaoshui Huang, Jinchang Ren, Zhiyuan Tan, Xiangjian He, Simultaneous Recovery to Classify: Dual-Stream Representation Learning GAN for Low-Resolution Image Classification, *in* 'IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing': vol.14, pp. 1705-1716 (2021).

- Chaocan Xiang, **Xiaochen Fan**, Chao Chen, Liangyi Gong, Songtao Guo, Fisher information-empowered sensing quality quantification for crowdsensing networks, *in* 'Neural Computing and Applications' (2021).

- Yuan-fang Zhang, Jiangbin Zheng, Long Li, Nian Liu, Wenjing Jia, **Xiaochen Fan**, Chengpei Xu, Xiangjian He, Rethinking feature aggregation for deep RGB-D salient object detection, *in* 'Neurocomputing' (2021) 423: 463-473.

Contents

List of Figures

List of Tables

Abbreviation

IoT - Internet of Things
MEC - Mobile Edge Computing
RAN - Radio Access Network
ISG - Industry Specification Group
ETSI - European Telecommunications Standards Institute
AR - Augmented Reality
C-V2X - Cellular Vehicle-to-Everything
IoV - Internet of Vehicles
VEC - Vehicular Edge Computing
MVC - Mobile Vehicular Cloudlet
CND - Content Delivery Network
AP - Access Point
RMSE - Root Means Square error
QoS - Quality of Service
QoE - Quality of Experience
DRL - Deep Reinforcement Learning
RSS - Radio Signal Strength
MCS - Mobile Crowd Sensing
SVT - Singular Value Thresholding
RNN - Recurrent Neural Network
LSTM - Long Short-Term Memory
MAE - Mean Absolute Error
MAPE - Mean Absolute Percentage Error
Seq2Seq - Sequence to Sequence

Dedication

To my parents Jian Fan and Chunlan Shi

To my wife Lu Lu

ABSTRACT

MOBILE EDGE COMPUTING: FROM TASK LOAD BALANCING TO REAL-WORLD MOBILE SENSING APPLICATIONS

by

Xiaochen Fan

With the rapid development of mobile computing technologies and the Internet of Things, there has been an increasing rise of capable and affordable edge devices that can provide in-proximity computing services for mobile users. Moreover, a massive amount of mobile edge computing (MEC) systems have been developed to enhance various aspects of people's daily life, including big mobile data, healthcare, intelligent transportation, connected vehicles, smart building control, indoor localization, and many others.

Although MEC systems can provide mobile users with swift computing services and conserve devices' energy by processing their tasks, we confront significant research challenges in several perspectives, including resource management, task scheduling, service placement, application development, *etc.* For instance, computation offloading in MEC would significantly benefit mobile users and bring new challenges for service providers. Unbalance and inefficiency are the two challenging issues when making decisions on computation offloading among MEC servers. On the other hand, it is unprecedented to design and implement novel and practical applications for edge-assisted mobile computing and mobile sensing. The power of mobile edge computing has not been fully unleashed yet from theoretical and practical perspectives.

In this thesis, to address the above challenges from both theoretical and practical perspectives, we present four research studies within the scope of MEC, including load balancing of computation task loading, fairness in workload scheduling, edge-

assisted wireless sensing, and cross-domain learning for real-world edge sensing. The thesis consists of two major parts as follows.

In the first part of this thesis, we investigate load balancing issues of computation offloading in MEC. First, we present a novel collaborative computation offloading mechanism for balanced mobile cloudlet networks. Then, a fairness-oriented task offloading scheme for IoT applications of MEC is further devised. The proposed computation offloading mechanisms incorporate algorithmic theories with the random mobility and opportunistic encounters of edge servers, thereby processing computation offloading for load balancing in a distributed manner. Through rigorous theoretical analyses and extensive simulations with real-world trace datasets, the proposed methods have demonstrated desirable results of significantly balanced computation offloading, showing great potential to be applied in practice.

In the second part of this thesis, beyond theoretical perspectives, we further investigate two novel implementations with mobile edge computing, including edge-assisted wireless crowdsensing for outdoor RSS maps, and urban traffic prediction with cross-domain learning. We implement our ideas with the iMap system and the BuildSenSys system, and further demonstrate demos with real-world datasets to show the effectiveness of proposed applications.

We believe that the above algorithms and applications hold great promise for future technological advancement in mobile edge computing.

Dissertation directed by Professor Xiangjian (Sean) He
Dissertation co-directed by Dr.Priyadarsi Nanda and Dr.Deepak Puthal
School of Electrical and Data Engineering
Faculty of Engineering and Information Technology

Chapter 1

Introduction

In recent years, the proliferation of the Internet of Things (IoT) and the ubiquitous wireless communication networks have shifted from centralised mobile cloud computing to mobile edge computing. Unlike the conventional paradigm of cloud computing, mobile edge computing harvests the available computation and storage resources of servers at the edge of networks. In a typical MEC system, edge servers provide computation, service caching, and storage capacity to mobile users within the radio access network (RAN). A variety of computation-intensive and latency-sensitive mobile applications have been benefited from mobile edge computing, including computation offloading, collaborative edge, augmented reality, intelligent video acceleration, smart home, indoor localization, intelligent transportation, and many others. In brief, although mobile edge computing is promising, there are still many challenges that remain unsolved yet, ranging from theoretical and algorithmic issues to real-world application development. This chapter will first present the research background of this thesis and the rationale of mobile edge computing. Then, we address the motivation and challenges in terms of load balancing and mobile sensing for edge computing, respectively. At last, we overview the structure of this thesis and provide a detailed organization.

1.1 Research Background and Rationale

In a typical cloud computing system, with the vision of centralising computing, storage and network management, the centralised data centres are responsible for supporting all elastic delivery of services. Over the last decade, with the proliferation of mobile devices and the rapid development of wireless networking technologies, the computation demand generated by mobile users have been growing unprecedented-

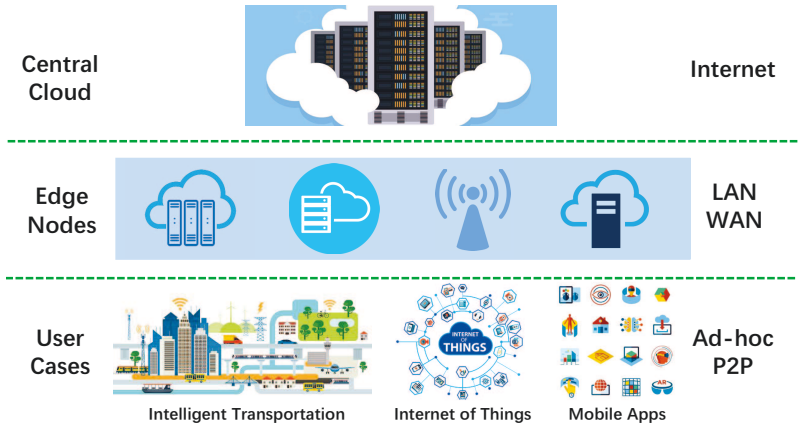


Figure 1.1 : Example of user cases and architecture of the MEC systems.

ly. According to Cisco [22], the total volume of global mobile traffic data will be exponentially increasing to over 70 exabytes (1 exabyte = 10^{18} bytes) per month, and mobile devices will account for 60% of traffic offloading. On this basis, the centralised processing models are significantly challenged by the stringent QoS requirements for computation-intensive and latency-critical applications [23]. To address the above challenges in mobile computing, with emerging 5G mobile communication networks, the concept of mobile edge computing (as shown in Fig. 1.1) has been proposed as a critical technology to provide decentralised computation services in mobile scenarios [24].

1.1.1 History and Research Background

According to [25], the evolution of edge computing involves three stages, including the technical preparation period, rapid growth period, and steady development period. Before 2015, edge computing was still in its early stage with technology preparation. Since 2015, edge mobile and mobile edge computing have entered a rapid growth period in both the academic community and industry counterpart. Fig. 1.2 illustrates the primary development process of edge computing.

Technical preparation. The initial stage of edge computing can be traced back to the early 2010s, when the content delivery network (CND) was developed as an Internet-based caching network to surrogate central platforms for load balancing, distributed scheduling and other functional modules [26]. While CDN can be a root

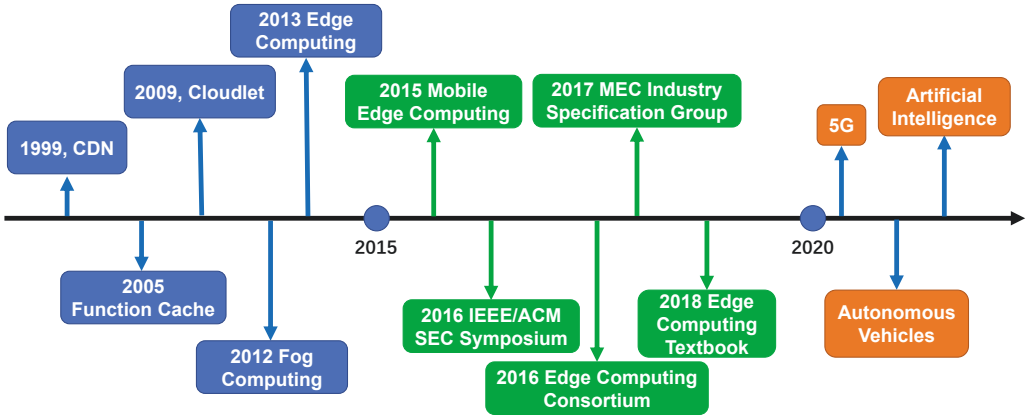


Figure 1.2 : A road map of the development process of edge computing.

concept of edge computing, it emphasizes the backup of data and caching of data. In contrast, the idea of edge computing focuses more on function caching, *i.e.*, accessible computation services. The evolution of edge computing went through various concepts and definitions, including cloudlets, small cell cloud, micro data centres, fog nodes, mobile micro cloud, *etc.* In 2009, Satyanarayanan et al. [27] brought forward the concept of *cloudlet*, a trusted, resource-rich cluster of servers deployed at the edge of networks. Cloudlets can execute offloadable arbitrary code partitions (computation tasks), which are typically computation-intensive and resource-intensive. In 2012, Flavio et al. [28] introduced the term *fog computing* for decentralised cloud computing and extended cloud computing for IoT infrastructures. In 2013, Nokia and Intel released the first edge computing platform, *i.e.*, the Radio Application Cloud Server, an IT architecture inside the cellular base stations to enable service delivery directly from the network's edge [29].

Rapid Growth. Since 2015, edge computing has been rapidly growing in both industry and academia [30]. In the industrial community, In September 2015, the industry specification group (ISG) within the European Telecommunications Standards Institute (ETSI) put forward the standardization of mobile edge computing. Different from conventional cloud computing paradigms, mobile edge computing occurs at the edge of the network while providing near-user computing services within the Radio Access Network (RAN) [24]. The main purpose of the standardization

of MEC is to provide efficient and seamless integration of cloud computing functionalities into the mobile network, thus developing favourable conditions driven by all stakeholders (mobile network operators, service providers, device vendors and mobile users) [26]. In November 2015, the Open Fog Consortium (later emerged to the Industrial Internet Consortium) was established by Cisco, ARM, Dell, Intel, Microsoft, and Princeton University, to jointly accelerate the growth of edge computing [31]. During this period, IT giants in the cloud computing market started to build edge computing platforms and provide relevant services. The pioneering companies include Amazon with the AWS IoT Greengrass [32] and Microsoft with Azure IoT Edge [33]. With edge computing becoming increasingly crucial in the academic community, the IEEE/ACM Symposium on Edge Computing (SEC) was initiated in 2016. Since then, some premier international conferences, such as ICC, INFOCOM, SIGCOMM and MobiCom, have included edge computing tracks or workshops to their main technical sessions.

Steady Development. From 2020, edge computing will step into a new decade with more steady development. In this period, driven by 5G, Artificial intelligence, and Autonomous Driving, edge computing will be realised with academia and industry integration.

1.1.2 Rationale

The preliminary rationale of edge computing is to provide data, computing and storage resources, and applications in close proximity to mobile devices or sensors, thus delivering highly responsive services. In specific, as categorized by Mao et al. [23], the advantages of mobile edge computing over cloud computing can be summarized as follows.

1. Low Latency: With proximity to mobile users, the edge servers can provide highly responsive services with low end-to-end latency, high bandwidth, and low jitter. Such low latency is of great necessity for emerging multimedia applications, such as augmented reality (AR).

2. Mobile Energy Savings: By effectively supporting *computation offloading*, mo-

bile edge servers can execute resource-intensive computing tasks for mobile users, thereby prolonging battery usages for energy-constrained mobile and IoT devices. Many research studies have demonstrated the significant energy saving by computation offloading in mobile edge computing, such as Edge-Box [34] and Heteroedge [35].

3. Context-Awareness: The MEC server can leverage the interactions with in-proximity end users to collect context information, such as population density, user behaviours and urban dynamics. Such context-aware information can be exploited to predict dynamic service demand. Service providers can then arrange more accurate and sufficient context-aware services [36–38] on different edge servers for mobile users.

4. Privacy/Security Enhancement: With the three-tier edge computing model (*i.e.*, IoT/mobile devices, edge servers, and the central cloud), the mobile edge server is the first point of contact for mobile/IoT devices, which provides promising opportunities for enhancing privacy and security policies. Applications with sensitive information exchange requirements would benefit from enterprise deployment of MEC, as the enterprise administrator can manage the authorisation, access control, and service classification to avoid uploading restricted data and material to central clouds.

5. Availability Backup: With more and more data, services and computation resources deployed at the edge network, edge servers can temporarily mask the central cloud's outages in case of network failure or denial-of-service attacks.

1.2 Motivation and Challenges

We address the motivation and challenges in mobile edge computing from two perspectives, *i.e.*, the theoretical and practical perspectives.

First, theoretically, the power of mobile edge computing could not be fully unleashed unless all the heterogeneous edge servers in the MEC systems could serve mobile users and perform their tasks simultaneously. If each edge server's resource

is fully utilized and the tasks can be concurrently processed sustainably by multiple servers, the average task response time can be substantially reduced. However, the distribution of user tasks is usually random. On the one hand, users' mobility cannot be centrally controlled, nor their offloading behaviours (as long as mobile edge servers are accessible). On the other hand, mobile edge servers travel around various metropolitan areas with different population densities. The amount of user task flow to any single edge server is not determined. Therefore, without efficient management of simultaneous offloading by multiple users and task migration, the task response time will be delayed, and the load of all edge servers in both computing resources and communication links will be unbalanced.

In the first part of this thesis, we tackle a significant problem of keeping a balanced load among all mobile edge servers, so that all the edge computing resources can be fully utilized, and the user tasks would be concurrently processed to reduce the average task response time. Nevertheless, two challenges need to be carefully addressed to achieve load balancing for mobile edge servers at the network edge.

- **Collaborative Edge**. The load balancing should be achieved through collaborative task offloading. As cloudlets' mobility cannot be controlled, it is impossible to continually redirect an exact amount of task flow from one cloudlet to another. Fortunately, it is possible for encountering cloudlets to offload tasks to each other by sharing load information collaboratively.

- **Distributed Efficiency**. The balanced task allocation method should be low-cost and light-weight in both communication and computation. It is impractical to query global load information in a dynamic network. Even if it can be achieved, the accumulative cost from the overall network would be extremely high, and the transmission delay may also lead to improper task offloading.

In chapter 3 and chapter 4 of this thesis, we tackle the above challenges of computation offloading in mobile edge networks with two collaborative task offloading mechanisms for mobile edge servers (*i.e.*, mobile cloudlets).

Second, in practice, with ubiquitous mobile devices and unprecedented computation resources at the network edge, there are huge demands for developing and implementing edge-assisted applications and systems. Moreover, the integration of edge computing would further extend the existing computing paradigm. Therefore, in the second part of this thesis, we address the above issue and investigate how to develop novel applications by integrating the concept of edge computing into existing sensing paradigms (*e.g.*, mobile crowdsensing and traffic sensing/prediction). An innovative edge computing empowered sensing system should take both the quality of edge sensing data and the scalability and reliability of sensing services.

- **Quality of Sensing Data**. Robust crowdsensing typically requires high quality of sensing data. Meanwhile, in mobile edge computing, the quality of data (*e.g.*, accuracy) can be seriously influenced by the heterogeneity of edge devices and the diversity of crowdsensing participants. As a consequence, the quality of sensing data will suffer from mismatches, incompleteness, and deficiency.

- **Feasibility and Reliability of Edge Sensing**. Integrating edge-assisted computing with the existing sensing paradigm would require verification of feasibility and reliability in practice. More importantly, it is more challenging to achieve convincing performance with edge sensing to outperform the existing sensing and computing paradigms.

Chapter 5 and chapter 6 of this thesis present two novel edge computing-assisted computing paradigms with crowdsensing and cross-domain sensing. We further address the above challenges with the system implementations based on real-world datasets.

1.3 Thesis Overview

Following the two main themes we have discussed, the technical chapters of this thesis consist of two parts as shown in Fig. 1.3. Part I is Edge Computation Offloading: Mechanism Designs, and Part II is Edge Computing Implementations: Applications.

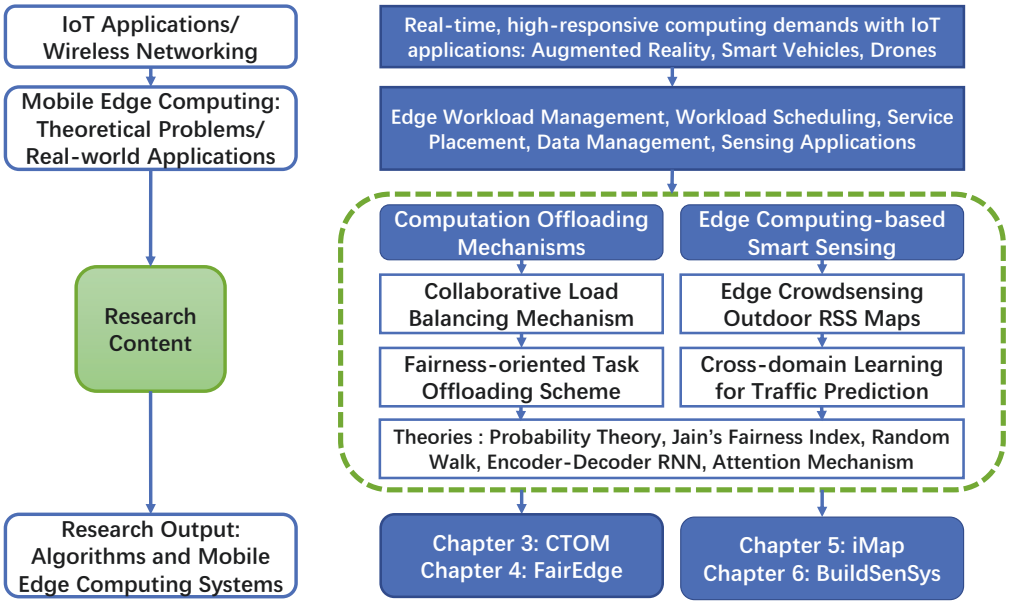


Figure 1.3 : An overview of the thesis structure.

Part I focuses on the theoretical innovations of computation offloading in mobile edge computing, emphasising Collaborative Edge so that mobile cloudlets can offload users' tasks to each other. Part II investigates two novel implementations with mobile edge computing, including an edge computing-assisted crowdsensing system for outdoor RSS maps, and a cross-domain learning-based system for traffic sensing and prediction.

1.4 Thesis Organization

Generally, the thesis is organised as follows.

- *Chapter 2*: We review and discuss the state-of-the-art works of mobile edge computing in computation offloading and mobile sensing, respectively.

- *Chapter 3*: To achieve more efficient and low-cost load balancing among mobile edge servers, we propose 'CTOM', a Collaborative Task Offloading Mechanism for mobile cloudlet networks, where the cloudlets are enhanced with mobility and intermittently connected. To the best of our knowledge, this is the first work focusing on collaborations among mobile cloudlets for task load balancing. To validate the effectiveness of the proposed mechanism, extensive sim-

ulation studies and trace-driven evaluations are conducted. Our simulation results show that the proposed 'CTOM' algorithm can achieve exceedingly balanced results in mobile cloudlet task allocation, thus enhancing the edge resource utilization and QoS of cloudlet networks.

- *Chapter 4*: To further guarantee fairness in task offloading, we propose 'FairEdge', a Fairness-oriented computation offloading scheme to enable balanced task distribution for mobile Edge cloudlet networks. The Jain's fairness index is integrated as a part of the task offloading algorithm. The FairEdge scheme enables a more reasonable computation offloading decision for each edge server by leveraging the task load information and fairness index of two targeted neighbours. The fairness-oriented task offloading further contributes to the overall load balancing and fairness of the mobile cloudlet network.

- *Chapter 5*: To integrate edge computing with mobile crowdsensing, we propose iMap, towards system implementation and data analysis for crowdsensing based outdoor RSS maps. Our system enables sensor-embedded mobile edge devices to collaboratively collect RSS data and build RSS maps in the wild. We develop a mobile application for end-users to collect RSS measurements and use a cloud-based server for RSS data storage and processing. Based on the experimental results, we investigate the error models for heterogeneous smartphones and propose a compressive sensing-based RSS data sampling and recovery algorithm to tame errors in edge sensed data.

- *Chapter 6*: As a preliminary exploration, we design and implement *BuildSenSys*, a first-of-its-kind system for nearby traffic volume prediction by reusing edge sensing data. First, we conduct a comprehensive building-traffic analysis based on multi-source datasets, disclosing *how* and *why* edge sensing data can be correlated with nearby traffic volume. Second, we propose a novel recurrent neural network for traffic volume prediction based on cross-domain learning with two attention mechanisms. Specifically, a cross-domain attention mechanism captures the building-traffic correlations and adaptively extracts the most relevant building sensing data at each predicting step. A temporal at-

tention mechanism is then employed to model the temporal dependencies of data across historical time intervals. This work can open a new gate of reusing edge sensing data for cross-domain sensing and prediction.

- *Chapter 7*: Finally, we conclude this thesis and provides a discussion of further research.

Chapter 2

Review on Computation Offloading and Mobile Sensing in Mobile Edge Computing

In this chapter, we first present an overview of mobile edge computing, from the early theoretical investigation in the early 2010s to the more recent attempts of developing mobile edge computing applications, and further to the latest breakthroughs of deep learning techniques with mobile edge computing. Secondly, we provide our insights into mobile edge computing's key issues and discuss promising solutions based on different methodologies. Then, we review the existing works related to this thesis's main themes in terms of theoretical and practical aspects. Finally, we envision the next stage of mobile edge computing in academia and industry, respectively.

2.1 Computation Offloading in Mobile Edge Networks

Computation offloading is one of the key use cases in mobile edge computing, and edge servers can perform computing offloading, data storage, caching and processing for the user's requests. Therefore, it is vital to design efficient offloading strategies and mechanisms to meet services' requirements, such as reliability, security, and privacy protection. In this respect, we first present the general principles of computation offloading in edge computing, including offloading classification (full, partial offloading). Afterwards, we review the latest research efforts addressing mobile edge computation offloading's critical issues, including edge server placement, mobility management, load balancing, and security issues.

2.1.1 Offloading Classifications

From mobile users' perspective, computation offloading in mobile edge computing is significantly beneficial to their computation-intensive and latency-critical applications at the resource-limited mobile devices. Through computation offloading,

MEC promises a critical reduction in latency and energy consumption. A decision on computation offloading in MEC can be affected by the factors as follows.

1. The offloadability of applications. For applications with executable codes, data partitions and parallelization, they can be considered suitable for computation offloading in MEC. Moreover, if the application is further divisible, the divided parts can be offloaded to edge servers if offloadable or processed locally if not offloadable.

2. The dependency of application divisions. The dependence of application divisions also influences the offloading decisions. If all divided parts are independent, they can be offloaded and executed simultaneously. However, for divisions with mutual dependencies, the output of particular components could be the input of other components.

Based on the above factors, there are generally two types of computation offloading in MEC, *i.e.*, full offloading and partial offloading. **Full offloading** decision in MEC is to offload the whole executable partitions of an application from mobile or IoT devices to the edge servers. In full offloading, most mobile users' applications would have a delay constraint or energy consumption constraint, and the goal of full offloading usually comes with the minimization of execution delay and minimization of energy consumption. **Partial offloading** refers to the computation offloading in MEC that a part of the computation is processed locally, while the rest is offloaded to edge servers for processing.

2.1.2 Computation Offloading in Cloudlet Networks

There have been substantial research efforts on the computation offloading in MEC with various emerging research directions, including edge server deployment, cache-enabled MEC, mobility management for MEC, Green MEC, as well as security and privacy issues in MEC. In this thesis, to specify the scenario of MEC, we use the term of (mobile) *cloudlet* to refer to (mobile) edge servers, and review the state-of-the-art research works in computation offloading within the scope of cloudlet

networks.

Satyanarayanan [39] firstly proposed the concept of 'cloudlet', a trusted, resource-rich computer or a cluster of servers that well-connected to the Internet. With the available accessibility to nearby mobile devices, cloudlets can provide low latency and high bandwidth network connectivity. Moreover, mobility-enhanced cloudlet systems (*e.g.*, cloud-vision [40]) were also proposed with mobile edge computing's emergence to collect computation tasks from mobile users for processing. To determine mobile cloudlets' capacity in providing mobile application services, Li et al. [41] studied the computing performance of a mobile cloudlet with respect to the cloudlet's size, lifetime and reachable time. To tackle the intermittent connections between mobile users and mobile cloudlets, Zhang et al. [42] developed an optimal computation offloading algorithm by considering both the user's local load and the availability of nearby cloudlets. Chen et al. [43] categorized the computation offloading in MEC into three modes: opportunistic mobile cloudlet service, connected cloudlet service and remote cloud service. Based on that, they further proposed a novel mobile cloudlet-assisted service mode to achieve flexible cost-delay tradeoffs between accessing remote cloud and mobile cloudlets [44].

More recently, driven by the concept of Cellular Vehicle-to-Everything (C-V2X) and Internet of Vehicles (IoV), vehicular edge computing (VEC) [45] has emerged as a new paradigm for mobile edge computing. Wang et al. [46] combined the vehicle-assisted cloud computing infrastructure to expend the available computing resources for task requests from smartphones. The serviceability of mobile vehicular cloudlets (MVC) was further assessed [47] based on the taxi mobility trace, and the analysis of spatial and temporal evolutions of serviceability was provided. To facilitate the resource-rich electric vehicles on the road, there are new opportunities for task offloading and data processing onto the vehicular edge network [48]. To extend the scalability of edge computing services and applications in mobile vehicular networks, Wang et al. [46] proposed a novel collaborative vehicular edge computing framework (CVEC) and envisioned the cloudlet collaboration for vehicular networks.

Beyond collaboration mobile vehicular cloudlets, load balancing has become an-

other key issue for reducing task response delay and improving resource usage efficiency [49]. Indeed, load balancing has been considered as an essential factor for QoS in static cloudlets and the related cloudlet placement studies [50]. In mobile edge computing scenarios, load balancing has been investigated with the task allocation problem formulated as an integer linear problem [51]. Given the high dynamics of vehicular edge computing networks, the task offloading was formulated into a constrained optimization problem and further solved with game theory methodologies [52]. To maximize the system utility of vehicular mobile edge computing, a collaborative computation offloading and resource allocation optimization scheme was presented to decouple the optimization problem into two sub-problems, *i.e.*, computation offloading decision-making and resource allocation [53].

To this end, in this thesis, we further investigate the load balancing problem in task offloading of mobile cloudlet networks. In Chapter 3, we propose 'CTOM', a Collaborative Task Offloading Mechanism for mobile cloudlet networks. Our solution is based on the balls-and-bins theory and can balance the task load, only requiring limited information. In Chapter 4, we propose 'FairEdge', a Fairness-oriented computation offloading scheme to enable balanced task distribution for mobile Edge cloudlet networks. By integrating the balls-and-bins theory with the fairness index, our solution promotes effective load balancing with limited information at low computation cost.

2.2 Mobile Sensing Applications with Edge Computing

Mobile crowdsensing [54] provides a cost-efficient solution to accommodate large-scale sensing tasks. For example, Sensorly [55] for monitoring speed and coverage of WiFi, LTE and 3G services. However, there are still some significant challenges for mobile crowdsensing, including lack of data validation, lack of incentive-compatible mechanisms, high traffic load, and high latency, due to the centralised data processing paradigm. The rising edge computing and deep learning motivate researchers to develop more advanced mobile sensing applications by combining state-of-the-art sensing paradigms, edge computing architecture and deep learning algorithms

altogether.

We review some representative works that combine edge computing and deep learning with mobile sensing applications as follows. In [54], Zhou et al. developed a robust mobile crowdsensing (RMCS) framework that integrated deep learning-based data validation and edge computing-based local processing for reliable service delivery. Moreover, Yao et al. [56] proposed DeepSense, a deep learning framework for mobile sensing and computing tasks, by leveraging convolutional and recurrent neural networks to exploit different types of relations in input sensory data. Li et al. [57] designed a novel computation offloading strategy to optimise IoT deep learning applications with edge computing. Notably, the concept of Edge Intelligence has been further defined in [58], which emphasizes the integration of artificial intelligence and edge computing.

In recent years, with the advance in artificial intelligence and the ubiquity of edge devices, there have been more premier mobile sensing frameworks and applications in both academia and industry. For example, Zhang et al. [35] developed HeteroEdge, a novel resource management framework for social mobile sensing-based edge computing. Yao et al. [56] proposed SADeepSense, a deep learning framework with a spatiotemporal self-attention mechanism for sensor inputs of heterogenous devices. SADeepSense first learns the correlations among different sensors over time by employing a self-attention mechanism without supervision. Then, it generates the residual concentrations that are deviated from the equal contributions from multiple sensors.

From mobile applications' perspective, numerous efforts have been made in service placement of edge computing networks. For instance, Konstantinos et al. [59] studied the joint optimization of service placement and request routing in MEC-enabled multi-cell networks with multi-dimensional constraints. Stephen et al. [60] investigated the problem of placing multiple services in a heterogeneous MEC network to maximize the total system reward. Moreover, Gao et al. [61] jointly considered the problem of network selection and service placement in MEC by associating access delay, switching delay and communication delay to improve the QoS for MEC

applications. From the mobile user's perspective, authors in [62] formulated a contextual multi-armed bandit (MAB) problem for the service placement in mobile edge computing, and further developed an online learning algorithm to cope with the dynamics in the edge environment and make accurate service placement decisions. At last, Farhadi et al. [63] separated the time scales of service placement and request scheduling into frames and slots to jointly optimize a set function.

2.3 Summary

In the following four chapters, we will mainly focus on the theoretical aspects of mobile edge computation offloading and the practical aspect of mobile edge-assisted applications, respectively. Chapter 3 is based on a conference paper [1] published in IEEE ICC 2018 and its extended version [11] published in the IEEE Access Journal. Chapter 4 is based on a joint paper [2] with Shuang Lai. Chapter 5 is based on the collaborative work [3] with Chaocan Xiang, which has been published in the IEEE Access Journal. Chapter 6 is based on our recent work published in IEEE Transactions on Mobile Computing [4]. In Chapter 7, we will discuss the future work from both theoretical and practical perspectives and conclude the thesis.

Chapter 3

CTOM: Collaborative Task Offloading Mechanism for Mobile Cloudlet Networks

Mobile cloud computing has emerged as a pervasive paradigm to execute computing tasks for capacity-limited mobile devices. More specifically, at the network edge, a resource-rich and trusted cloudlet system acts as a 'data centre in a box' can support compute-intensive mobile applications. Mobile cloudlets can provide in-proximity services by executing the workloads for nearby devices. Nevertheless, load balancing in mobile cloudlet network is of great importance, as it has a significant impact on task response time. The existing methods for cloudlet load balancing basically rely on strategic placement or user cooperation. However, the above solutions require the global task load information from the whole network, costly in both communication and computation. We propose CTOM, a Collaborative Task Offloading Mechanism for mobile cloudlet networks to achieve more efficient and low-cost load balancing. Our solution is based on the balls-and-bins theory and can balance the task load, only requiring limited information. Extensive simulations and evaluation based on mobility trace demonstrate that our CTOM outperforms the conventional random and proportional allocation schemes by reducing the task gaps among mobile cloudlets by 65% and 55%, respectively. Meanwhile, CTOM's performance is close to that of the greedy algorithm but with much lower computing complexity.

3.1 Introduction

In recent years, with the pervasive proliferation of mobile devices and the advance in networking technologies, mobile users can enjoy various robust and functional applications, such as Augmented Reality, Virtual Reality, *etc.* While these mobile applications are becoming more demanding in computing resources, the capacity of

a single smartphone is still constrained. For example, most mobile users are constantly facing the problems of resource-exhaustion or energy-drain on their devices. To tackle this issue, cloud computing has been proposed and pervasively used for processing resource-intensive tasks. However, due to the long distance between a central server and its mobile users, cloud computing has some inevitable limitations, such as network latency, link noise, and transmission delays [64]. To provide more reliable computing services for mobile users, an alternative cloud computing paradigm has been proposed, called 'cloudlet' [39].

A cloudlet is a trusted, resource-rich cluster of servers integrated with wireless access points (APs), which is accessible and connected to nearby mobile users [65]. By providing seamless access with low-latency and high-bandwidth, cloudlets can execute computation tasks for mobile users in real-time, thereby significantly improving the performance of cloud computing [42, 49]. Recent studies [45, 66–68] have focused on mobile cloudlets that utilise the multitude of near-user vehicular cloudlets to achieve more efficient task offloading and processing.

A key challenge in mobile cloudlet networks is how to keep a balanced load among all of the mobile cloudlets so that cloudlet resources are fully utilised, and tasks can be concurrently processed sustainably by multiple servers, thus reducing the average task response time [49]. As vehicle-based cloudlets travel around various metropolitan areas with different population densities, it is impossible to centrally control the amount of user task flow to any single cloudlet. Besides, the connectivity in mobile cloudlet networks is also intermittent.

Some existing studies address the load balancing problems in static cloudlet systems, either by strategic cloudlet placement [65, 69] or by cloudlet-oriented task redistribution [49]. However, these methods are not applicable in mobile cloudlet networks, where the cloudlets are mobility-enhanced, and the network is intermittently connected. Even worse, for each mobile cloudlet, its neighbours' load information continuously varies, which would make it costly to compute the overall load information. Accordingly, two challenges need to be carefully addressed.

First, load balancing should be achieved through collaborative task offloading.

As cloudlets' mobility cannot be controlled, it is impossible to continually redirect an exact amount of task flow from one cloudlet to another. Fortunately, it is possible for encountering cloudlets to offload tasks to each other by sharing load information collaboratively.

Second, the balanced task allocation method should be low-cost and light-weight in both communication and computation. It is impractical to query global load information in a dynamic network. Even if it can be achieved, the overall network's accumulative cost would be extremely high, and the transmission delay may also lead to improper task offloading.

In this chapter, to deal with the above challenges, we propose 'CTOM', a Collaborative Task Offloading Mechanism for mobile cloudlet networks. Our method leverages the balls-and-bins model to fit the distributed task allocation scenario in mobile cloudlet networks. Based on the 'two-choice' paradigm, by only querying load information from only two random neighbours in each time interval, each cloudlet can process relatively balanced task offloading. Accumulatively, the longest task queue among all mobile cloudlets will be significantly reduced with high probability [70].

We summarise the contributions of this chapter as follows.

1. We propose a collaborative task offloading mechanism for mobile cloudlet networks, where the cloudlets are enhanced with mobility and intermittently connected. This is the first work focusing on mobile cloudlets' collaborations for task load balancing to the best of our knowledge.

2. Inspired by the balls-and-bins probability theory, we propose an innovative solution to the problem of balanced task allocation in distributed mobile cloudlet networks. By comparing only two neighbours' task load, each mobile cloudlet can process valid task offloading at low communication cost in each time interval.

3. To validate and demonstrate the effectiveness of the proposed mechanism, extensive simulation and trace-driven evaluation have been conducted. Our

simulation results show that the proposed 'CTOM' algorithm has achieved exceedingly balanced results in mobile cloudlet task allocation and performed close to the optimal allocation that relies on global task load information.

The rest of this chapter is organised as follows. We first review the related work in Section 3.2. In Section 3.3, we describe the system model of mobile cloudlet networks and formulate the load balancing problem. We present our solution and algorithm design in Section 3.4 and further evaluate the proposed CTOM with extensive simulation and trace-driven evaluation in Section 3.6. Finally, we conclude this work in Section 3.7.

3.2 Related Work

Satyanarayanan [39] proposed 'cloudlet', a ubiquitous facility that acts as a 'data centre in a box' to serve nearby mobile users. As the communication from a local cloudlet to surrounding users is usually within one hop, cloudlets can provide low latency and high bandwidth network connectivity. Moreover, mobility-enhanced cloudlet systems are also proposed with the emergence of mobile edge computing, where cloudlet-integrated vehicles randomly travel in metropolitan areas to collect and process tasks for mobile users [71].

Several existing studies proposed different methods to solve the load balancing problem in cloudlet networks. The first approach is strategic cloudlet placement. Xu et al. [65] proposed a placement strategy for capacitated cloudlets to minimise the cloudlet accessing delay and average task response time for device users. Jia et al. [49] further formulated an optimal task redirection problem and devised a load balancing algorithm to minimize the task response time. However, in our scenario, the cloudlets are enhanced with mobility, so that the connectivity in the network is intermittent. With continuously changing task flows from edge devices to cloudlets, the above solutions become incompetent.

Moreover, Zhang et al. [42] developed an optimal offloading algorithm for mobile users by considering user mobility patterns and cloudlet admission control. In [69],

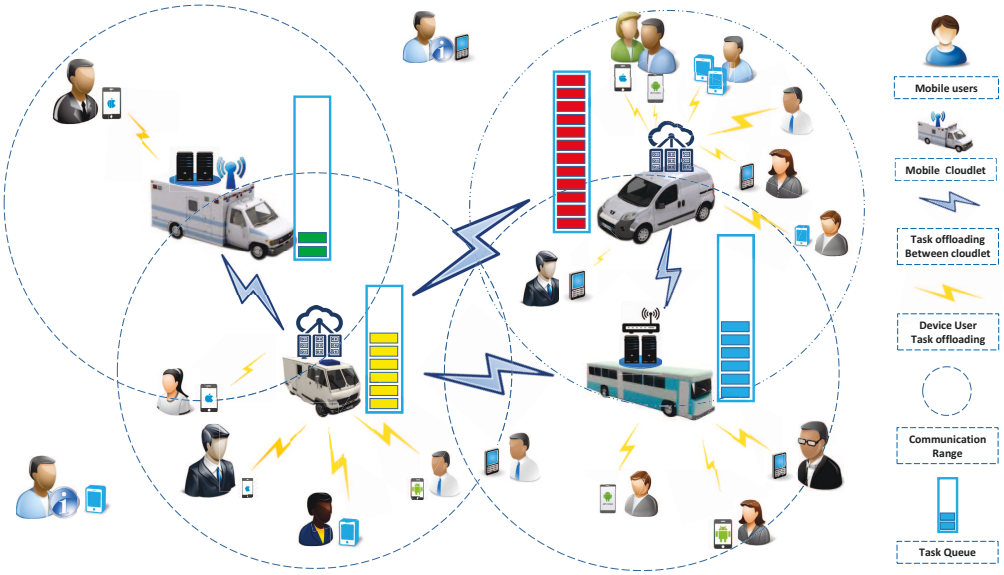


Figure 3.1 : Task offloading in a mobile cloudlet network scenario.

Jia et al. associated the cloudlet placement problem with task assignment at user dense regions. Unlike the above works, in this chapter, we focus on mobile cloudlets' cooperation and aim to explore the possibility of leveraging mobile cloudlets encounters to balance the task load.

3.3 System model and problem formulation

In this section, we first present the preliminaries of our system model. After that, we formulate the load balancing problem in mobile cloudlet networks.

Network Model: We start the network model with a set of mobile cloudlets travelling in a metropolitan area network. We assume that K mobile cloudlets are integrated with vehicular access points (APs), and we denote them by $\mathrm{C} = \{\mathrm{c}_1,\mathrm{c}_2,...,\mathrm{c}_K\}$. The cloudlets can communicate via network connections and offload tasks to its current neighbours in the network, as depicted in Fig. 3.1.

Cloudlet Model: According to [49], for each mobile cloudlet c_i, where $i \in \{1, 2, ..., K\}$, we model it as an $M/M/n$ queue. Each cloudlet i has s_i server(s) with the service rate μ_i. Then, we adopt the random walk model for the mobility of cloudlets. For cloudlet i, as the amount of task offloading from nearby users varies continuously, we also adopt the Poisson process to model the incoming user

tasks [49], where the task arrival rate at cloudlet i is λ_i. Furthermore, to store the arrived tasks pending for execution, each mobile cloudlet i holds a FIFO task queue $Q_i = \{q_1, q_2, .., q_k\}$, where the queueing length is $\|Q_i\|$.

Communication Model: Similar to [49], we assume that the mobile cloudlets in our model are also integrated with wireless access points, which provide for one-hop, low-latency, and high-bandwidth wireless access for task offloading. When the distance d_{ij} between cloudlets i and j is within the inter-contact range R, a communication contact can be established between them [42].

By referring to [72] and [9], the inter-meeting time of cloudlets c_i and c_j follow an exponential distribution with a pairwise rate α_{ij}, i.e., $f(t) = \frac{1}{\alpha_{i,j}} e^{-\frac{1}{\alpha_{ij}} \cdot t}, t \geq 0$. Thus between any two time points t_a and t_b, the encountering probability of cloudlets c_i and c_j is computed by:

$$P_{ij}(t_a, t_b) = e^{-\frac{1}{\alpha_{ij}} \cdot t_a} - e^{-\frac{1}{\alpha_{ij}} \cdot t_b}, t_b > t_a. \tag{3.1}$$

Satyanarayanan et al. [39] conducted several task offloading experiments in cloudlet networks connected by WiFi. The execution time of offloaded tasks is approximately $10^{-4} \sim 10^{-2}$ seconds for applications such as augmented reality and facial recognition. Because the round-trip time (RTT) of wireless transmission only takes tens to hundreds of milliseconds, we consider the time interval set in our model is reasonably long enough for the inter-contact time (including execution time and RTT).

Task Offloading Model: In our model, a 'task' refers to an application phase that involves executable codes and offloadable data that can be processed at any mobile cloudlet [42]. Moreover, the total number of tasks generated by different users fluctuates. We also adopt the percentage imbalance metric η and the statistical skewness φ from [73] to evaluate the network's overall load balancing. The above

metrics are calculated by:

$$\eta = \left(\frac{L_{max}}{\bar{L}} - 1 \right) \times 100\%, \qquad \varphi = \frac{\frac{1}{n} \sum_{i=0}^{n} (L_i - \bar{L})^3}{(\frac{1}{n} \sum_{i=0}^{n} (L_i - \bar{L})^2)^{3/2}}, \tag{3.2}$$

where L_{max} and $\bar{L}$ are the maximum and average loads, respectively. The percent imbalance metric measures the severity of load imbalance, while the skewness provides a detailed load distribution picture [73].

Problem Domain: Given a mobile cloudlet network G with a set of cloudlets C = {c_1,c_2,...,c_K}, where cloudlet i holds a task queue Q_i for $i = 1, 2, \cdots, K$.. Meanwhile, cloudlet i has n_i servers with service rate μ_i, and the task arrival rate at cloudlet i is λ_i. We define the Mobile Cloudlet Load Balancing Problem as follows.

Basic Load Balancing Problem: We investigate how to offload tasks in mobile cloudlet networks collaboratively. Particularly, our goal is to minimise the overall variance of a task queue in achieving balanced task distribution, as computed by

$$\text{minimise} \sum_{i \in C} \|Q_i - E\,[Q]\|, \tag{3.3}$$

subject to $\mu_i \cdot n_i \geq \lambda_i, i \in C$.

Gap minimisation and balance metric evaluation: Minimising the task load gap between the maximum queue and average queue is also worth evaluating. Note that the maximum load $L_{\max}$ and average load $\bar{L}$ count for the imbalance metric and statistical skewness in Equation 3.2. The evaluation of task load gap can be given by

$$\text{minimising} \max_{i \in C} \{\|Q_i\| - E[Q_i]\}, \tag{3.4}$$

subject to $\mu_i \cdot n_i \geq \lambda_i, i \in C$.

3.4 Our solution and algorithm design

In our work, we adopt the balls-and-bins theory and design a collaborative task offloading mechanism, *i.e.*, CTOM, to improve the load balancing of cloudlet networks. Before presenting the details of our CTOM algorithm, we first briefly introduce the balls-and-bins theory.

3.4.1 Leveraging balls-and-bins based probability theory

The balls-and-bins model is a classic probability model for the randomised allocation process [70]. Suppose that n balls are thrown into n bins, with each ball choosing a bin independently and uniformly at random. Then, the *maximum load*, *i.e.*, the largest number of balls in any bin, is approximately $\frac{\log n}{\log \log n}$. Now assume that each ball is placed into the least loaded bin, among $d \geq 2$ bins chosen independently and uniformly. This is called the *d-choice* paradigm. In this case, the maximum load is approximately $\frac{\log \log n}{\log d} + \mathrm{O}(1)$.

The extension of the balls-and-bins model's maximum load problem is further considered, where m balls are sequentially placed into n bins with $m \gg n \log n$. In this case, for random allocation, the number of balls in the fullest bin is approximately $\frac{m}{n} + \sqrt{\frac{m \log n}{n}}$. While for d-choice, if $m \gg n \log n$, then the maximum load is approximately $\frac{m}{n} + \mathrm{O}(\sqrt{\frac{\mathrm{m} \log \mathrm{n}}{\mathrm{n}}})$.

In this work, the d-choice paradigm is adopted to a mobile cloudlet network model, where tasks and mobile cloudlets are considered as balls and bins, respectively. In designing the task offloading algorithm, we apply the *d-choice* paradigm to balance the network load. With the theoretical guarantee from the balls-and-bins theory, the collaborative task offloading could yield a larger reduction on the maximum load in mobile cloudlet networks.

3.4.2 Algorithm Design

We introduce the algorithm design of CTOM as follows.

Overview

In algorithm 1, there are some basic assumptions in the algorithm design. First, we mainly focus on collaborations among mobile cloudlets in task offloading. For each cloudlet c_i, the incoming tasks from users follow a Poisson process with a constant task arrival rate. Second, we assume that the tasks in the network are of the same size, so that the final allocation results can be measured precisely. Third, at each cloudlet c_i, the arrived tasks are stored in the task queue Q_i. Fourth, the time interval is long enough for an inter-contact connection (including execution time and RTT).

Algorithm

Basic inputs and outputs: The basic inputs include a set of cloudlets C, time interval $\mathbf{T}$, the inter-contact range R. For each cloudlet c_i, we denote the user task flowing rate as λ_i, the number of servers as s_i and the service rate as μ_i. The outputs include the set of task load queues Q, the imbalance metric η, and the statistical skewness φ.

Exploring opportunistic encounters: In the initialising step, the algorithm first randomly generates each mobile cloudlet's location. When each time interval begins, all cloudlets will perform random walks, and the algorithm will update their locations. According to Equation 3.1, the algorithm will first check whether there are new mobile cloudlets within its communication range for each cloudlet c_i. Then, with the neighbouring list $l(i)$, the algorithm will calculate the number of neighbours for each cloudlet c_i. If the number of neighbours is greater than d, the algorithm will apply the d-choice paradigm; otherwise, the algorithm will assign $d/2$ to d until the value of d is smaller than the number of current neighbours.

Task offloading paradigm: The proposed CTOM will randomly select d neighbouring mobile cloudlets for cloudlet i and iteratively compare their task loads, after that, cloudlet i will offload one portion of its task to the least loaded neighbour in the current time interval. For the greedy algorithm, the value of d equals to the total number of each cloudlet's neighbours. The proportional algorithm [74] will further

Algorithm 1 The CTOM Algorithm

Require:
Mobile Cloudlet **C**, Time Interval **T**, Contact Range R
User Task Flow λ_i, Number of Servers **S**, Service Rate μ_i
Ensure:
Task Queue **Q**, Imbalance Metric and Statistical Moment

```
Minimise $\sum_{i \in C} \|Q_i - E[Q]\|$ using the $d$-choice method.
Initialise cloudlet's location (X, Y)
for Interval $t = [1 : T]$ do
  Cloudlets perform random walks in a metropolitan area
  Update each cloudlet's location in this time interval
  Update cloudlet's load information with $\lambda_i$, $q_i$, $\mu_i$
  for Cloudlet c_i, $i = [1 : k]$ do
    Add mobile user's task offloading into $q_i$
    Calculate encountering cloudlets according to Eq. 3.1
        Update neighbouring list $l(i)$
        if $\|l(i)\| \geq d$
        Select $d$ neighbours randomly and independently
        else if $\|l(i)\| \leq d$
        do $d \leftarrow d/2$ until $\|l(i)\| \geq d$
        end if
        Compare the load of $d$ selected neighbours
        $s \leftarrow$ the first selected neighbour in d
          for $v = 2$ to $d$ do
            if $Q_s > Q_v$ then $s \leftarrow v$
            end if
          end for
        if $Q_i > Q_s$ then
          $P \leftarrow 1 - Q_s/Q_i$
          $Q_s \leftarrow l_s + W(i) * P$
          $Q_i \leftarrow l_i - W(i) * P$
        end if
  end for
end for
Calculate imbalance metric and statistical moment skewness according to Equation 3.2
return Q, $\eta$, $\varphi$
```

compute the offloading probability based on the task loads of the current cloudlet and the selected cloudlet. When all mobile cloudlets finish the task offloading, the next time interval begins. Finally, the imbalance metric, together with statistical moment skewness, will be calculated according to Equation 3.2.

3.5 Method Validation

In this section, we present the claims made in the proposed load balancing algorithm and provide proof. First, we give out the definitions and notations as follows.

We basically follow [75] and consider task offloading as a *finite* process, where there are m tasks and n mobile cloudlets. Initially, the mobile cloudlets are all idle, and each of the tasks is allowed to be offloaded into one of d ($d \geq 2$) neighbouring cloudlets chosen independently and uniformly at random. The arrived tasks at each cloudlet are stored by FIFO. We denote the above task allocation process as an (m, n, d)-problem. In our proof, to make the exposition clearer, we first prove the case when $m = n$, and then we can shift the proof to $m > n$ case.

Table 3.1 : Notations and definitions

Notations	*Definitions*
$l_j^c(t)$	the load of cloudlet j, *i.e.*, the number of tasks in cloudlet j at time t, resulting from the proposed CTOM algorithm
$N_k^c(t)$	the number cloudlets that with the load of k at time t
$N_{\geq k}^c(t)$	the number of cloudlets that have the load larger than or equal to k at time t, *i.e.*, $N_{\geq k}^c(t) = \sum_{i>k} N_i^c(t)$
H_t^c	the length of the task queue t, which equals to the number of tasks at time t in a cloudlet
$M_k^c(t)$	the number of tasks that have a height of k at time t
$M_{\geq k}^c(t)$	the number of tasks with the height larger than or equal to k at time t, *i.e.*, $M_{\geq k}^c(t) = \sum_{i\geq k} M_i^c(t)$

The proposed algorithm CTOM assigns a task j from its current cloudlet to the cloudlet with the lowest load among its d randomly selected neighbours. Next, we prove the upper bound of tasks in the fullest cloudlet under the CTOM algorithm.

Claim 1. Suppose there are n tasks to be allocated to n cloudlets. For each cloudlet,

it allocates the task to the least loaded neighbour out of d selected neighbours. Then the upper bound, *i.e.*, the total number of tasks in the fullest cloudlet, is at most $\frac{\ln \ln n}{\ln d}$ with a high probability. We list the definitions of variables used in our proof in Table 3.1).

Proof. The basic intuition of the proof is as follows. Let $p_i = M_{\geq i}/n$. For each cloudlet, it offloads the current task independently and $N^c_{\geq k} \leq M^c_{\geq k}$, then we roughly have $p_{i+1} \leq p_i^d$ (d is the number of offloading choices), which shows the decrease in p_i is doubly exponential, as long as $M_{\geq i} < n/2$. Obviously, $M_{\geq i+1}$ is based on the condition that $M_{\geq i}$. □

We consider that the task allocation process is finite and denote a binomial and distributed random variable by $B(n, p)$. According to [75], we have a standard Lemma as follows.

Lemma 1. *Let $X_1, X_2, ..., X_n$ be a sequence of random variables with arbitrary values. Let $Y_1, Y_2, ..., Y_n$ be a sequence of binary random variables, with $Y_i = Y_i(X_1, ..., X_i)$. If*

$$\Pr(Y_i = 1 | X_1, ..., X_{i-1}) \leq p,$$

then, we have

$$\Pr(\sum Y_i \geq k) \leq \Pr(B(n, p) \geq k).$$

Similarly, if

$$\Pr(Y_i = 1 | X_1, ..., X_{i-1}) \geq p,$$

we have

$$\Pr(\sum Y_i \leq k) \leq \Pr(B(n, p) \leq k).$$

□

As the d choices are independent for each task, we have $\Pr(H_t \geq i + 1 | N_{\geq i}(t - 1)) \leq \frac{(N_{\geq i}(t-1))^d}{n^d}$.

We use θ_i to denote the event of $N_{\geq i}(n) \leq \alpha_i$ (α_i will be illustrated in the following steps), which implies that $N_{\geq i}(t) \leq \alpha_i$ for $t = 1, 2, ..., n$).

For $i \geq 1$, we consider Y_t $(t = 2, ..., n)$ as the serial binary variables, where $Y_t = 1 \iff h_t \geq i + 1$ and $\nu_{\geq i}(t-1) \leq \beta_i$.

That is to say, $Y_t = 1$ if the height of the task t is greater than $i + 1$, even the number of cloudlets that have more than i tasks is less than α_i.

We use γ_j to denote the number of choices available for the j_{th} ball. Then, we have

$$\Pr(Y_t = 1 | \gamma_1, ..., \gamma_{t-1}) \leq \frac{\alpha_i^d}{n^d} \triangleq p_i.$$

Now we apply Lemma 1 to conclude that

$$\Pr(\sum Y_{\mathrm{t}} \geq k) \leq \Pr(B(n, p_i) \geq k).$$

When conditioned on θ_i, we have $M_{\geq i+1} = \sum Y_t$, thereby,

$$\Pr(\sum M_{\geq i+1} \geq k | \theta_i) = \Pr(\sum Y_t \geq k | \theta_i) \leq \frac{Pr(\sum Y_{\mathrm{t}} \geq k)}{\Pr(\theta_i)}.$$

By combining the above two formulas, we can obtain

$$\Pr(\sum N_{\geq i+1} \geq k | \theta_i) \leq \frac{\Pr(B(n, p_i) \geq k)}{\Pr(\theta_i)}.$$

□

According to [76] (see Appendix A), the large deviations in the binomial distribution can be bounded as

$$\Pr(B(n, p_i) \geq e p_i n) \leq e^{-p_i n},$$

Therefore, we can set

$$\alpha_i = \begin{cases} n, & i = 1, 2, ..., 5; \\ \frac{n}{2e}, & i = 6; \\ \frac{e\alpha_{i-1}^d}{n^{d-1}}, & i > 6. \end{cases}$$

As $\theta_{\geq 6} = \{N_6 \leq n/(2e)\}$ still holds, for $i \geq 6$,

$$\Pr(\neg\theta_{i+1}|\theta_i) \leq \frac{1}{n^2 \Pr(\theta_i)},$$

with $p_i n \geq 2 \ln n$.

Since

$$\Pr(\neg\theta_{i+1}) \leq \Pr(\neg\theta_{i+1}|\theta_i) \Pr(\theta_i) + \Pr(\neg\theta_i),$$

we have

$$\Pr(\neg\theta_{i+1}) \leq \frac{1}{n^2} + \Pr(\neg\theta_i).$$

Let i^* be the smallest i, then, $\alpha_{i^*}^d / n^d \leq 2 \ln n / n$. While

$$\alpha_{i+6} = \frac{n e^{(d^i - 1)/(d-1)}}{(2e)^{d^i}} \leq \frac{n}{2^{d^i}},$$

we have $i^* \leq \ln \ln n / \ln d + \mathrm{O}(1)$

As above,

$$\begin{aligned} \Pr(N_{\geq i^*+1} \geq 6 \ln n | \theta_{i^*}) &\leq \frac{\Pr(B(n, 2 \ln n/n) \geq 6 \ln n)}{\Pr(\theta_{i^*})} \\ &\leq \frac{1}{n^2 \Pr(\theta_{i^*})}. \end{aligned} \tag{3.5}$$

Thus, we have

$$\Pr(N_{\geq i^*+1} \geq 6 \ln n) \leq \frac{1}{n^2} + \Pr(\neg\theta_{i^*})$$

.

Finally,

$$\begin{aligned}\Pr(M_{\geq i^*+2}|N_{\geq i^*+1} \leq 6\ln n) &\leq \frac{\Pr(B(n, 6\ln n/n)^d \geq 1)}{\Pr(N_{\geq i^*+1} \leq 6\ln n)} \\ &\leq \frac{n(6\ln n/n)^d}{\Pr(N_{\geq i^*+1} \leq 6\ln n)}.\end{aligned} \tag{3.6}$$

□

Based on the Markov inequality [77], we can obtain

$$\Pr(M_{\geq i^*+2} \geq 1) \leq \frac{n(6\ln n)^d}{n^{d-1}} + \Pr(N_{\geq i^*+1} \geq 6\ln n).$$

By combining the above three formulas, we have

$$\Pr(N_{\geq i^*+2} \geq 1) \leq \frac{n(6\ln n)^d}{n^{d-1}} + \frac{i^*+1}{n^2} = o(1). \tag{3.7}$$

Note that $i^* \leq \ln\ln n/\ln d + \mathrm{O}(1)$. Then, the above proof shows that the maximum load achieved by the proposed CTOM is no more than i^*+2 with a high probability, where $i^*+2 = \frac{\ln\ln n}{\ln d} + \mathrm{O}(1)$.

For the case $m > n$, *i.e.*, (m, n, d)-problem, if we consider θ_i to be the event that $N_{\geq i}(m) \leq \alpha_i$ and also define $p_i = \alpha_i^d/n^d$. Following the proof from the $m = n$ case, we can derive that

$$\Pr(\sum N_{\geq i+1} \geq k|\theta_i) \leq \frac{\Pr(B(m, p_i) \geq k)}{\Pr(\theta_i)}.$$

We suppose that $\alpha_x = n^2/(2em)$ for special values of x while θ_x also holds, *i.e.*,

$$\Pr(N_x \geq \frac{n^2}{2em}) = o(1).$$

Then, we can have

$$\alpha_{i+x} = \frac{n}{2^{d^i}}(\frac{me}{n})^{(d^i-1)/(d-1)-d^i} \leq \frac{n}{2^{d^i}}.$$

By continuing as the proof of $m = n$ case, we can obtain that

$$\Pr(M \geq x + \ln\ln n/\ln d + 2) = o(1).$$

Above all, we show that for (m, n, d)-problem, the maximum task queue in any cloudlet is no more than

$$(1 + o(1))\ln\ln n/\ln d + \mathrm{O}(m/n). \quad \square \tag{3.8}$$

To this end, we have proved the upper bound of task load under CTOM.

Claim 2. The communication cost of the proposed CTOM (applying the 2-choice paradigm) is no more than twice the random allocation on a ρ-round (infinite) (m, n, d)-problem.

Proof. For an (m, n, d)-problem, we denote the average communication cost of our CTOM, the random allocation and the greedy allocation as $C_C(m, n)$, $C_R(m, n)$ and $C_G(m, n)$ respectively.

Under the random allocation scheme, a mobile cloudlet queries the load information from a randomly selected neighbour within each interval's contact range (round). Thus, we have

$$C_R(m, n) \leq \rho n.$$

For the case of greedy allocation, a mobile cloudlet queries the global load information from its neighbours, which results in a high communication cost as

$$C_G(m, n) \leq \rho(n - 1)^2.$$

In our CTOM, applying the 2-choice paradigm, a cloudlet only queries two randomly selected neighbours. However, there are chances that only one or no cloudlet is within the communication range of the current cloudlet. Therefore, we have

$$C_C(m, n) \leq \rho \cdot 2n.$$

Above all, the communication cost under different task allocation schemes are ranked as

$$C_R(m,n) < C_C(m,n) < C_G(m,n), \quad \square \tag{3.9}$$

where $C_R(m,n) \leq 2C_C(m,n)$.

3.6 Performance Evaluation

The performance evaluation of the proposed scheme is twofold. First, we evaluate the proposed CTOM in a simulated network scenario, where cloudlet encounters are generated from random walk simulations. Second, we apply the proposed algorithm to a real-world trace for further evaluations.

3.6.1 Simulation Study

Basic setups

We run the simulation in a $10km^2$ region, which is of a similar scale to a city's central area. Here, we set the number of mobile cloudlets to be 100, and the communication range to be 20 metres. The total number of time slots is 600. According to [49], for each cloudlet i, we set the service rate μ_i by sampling normal distribution $\mathcal{N}(2,1) > 0$, and we set the number of its servers by sampling the Poisson distribution with a mean of 2. For the tasks arriving at cloudlet i, we set task arrival rate λ_i by sampling the Normal distribution $\mathcal{N}(4,2) > 0$. We consider extreme task distribution that overwhelmed any cloudlet as the potential DDoS attacks.

Under our CTOM scheme, during each time interval, a cloudlet first randomly chooses 2 neighbours in its contact range. After querying and comparing their load states, the cloudlet offloads a task to the neighbours having lower task load, where the computing complexity in each time interval is $O(1)$. Similar to [9], we compare the performance of the proposed scheme with three benchmarks, *i.e.*, random allocation, proportional allocation [78], and greedy allocation.

In the random allocation, a mobile cloudlet offloads tasks by randomly selecting

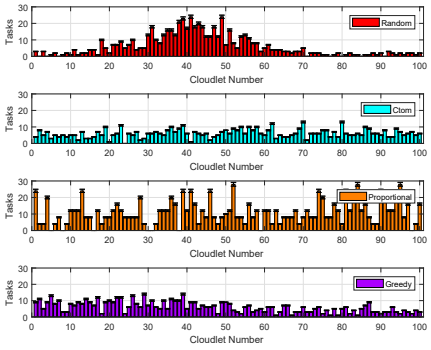


Figure 3.2 : Task allocation result.

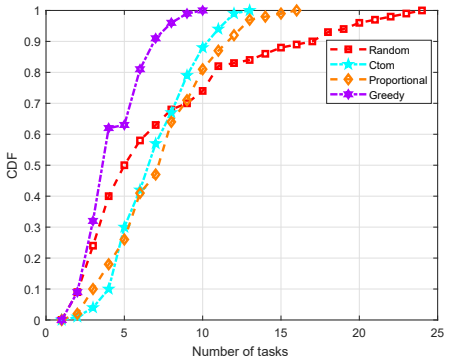


Figure 3.3 : The distribution of task loads obtained with the four schemes.

another mobile cloudlet in its contact range. Conversely, the greedy allocation method first queries all load information from its neighbours and compares their task loads. Then, it allocates tasks to the optimal cloudlet (with a computing complexity of $O(n)$). As for the proportional allocation, the chance for tasks to be offloaded to a randomly selected cloudlet depends on a probability parameter calculated with task load information.

The simulation programs are all written in MATLAB codes. We run the programs in a Dell laptop with the Intel Core i5 processor and 8 GB RAM. In general, each simulation program is executed 100 times, and we take the average results as the final performance.

Overall Performance

Fig. 3.2 plots the overall task allocation results of mobile cloudlets obtained with our CTOM scheme and the three benchmark methods, *i.e.*, random allocation, proportional allocation, and greedy allocation. Since the cloudlet's servers keep processing tasks, the overall allocation shows each mobile cloudlet's remaining tasks. In the random allocation, an adjacent group of cloudlets (ID 18 to 60) is overloaded with potential DDoS tasks, where most of their task loads are more than 10 and up to 24. For example, legitimate tasks can not be processed normally. Meanwhile, the mobile cloudlets at edge area are loaded with much fewer tasks (average less than

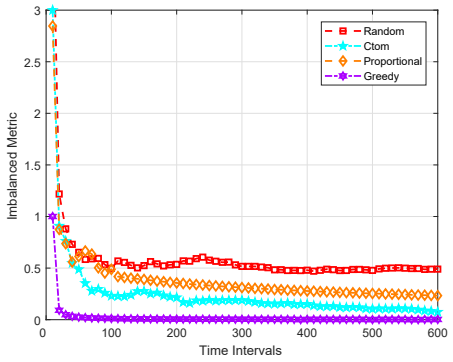


Figure 3.4 : The imbalance metrics obtained with the four schemes.

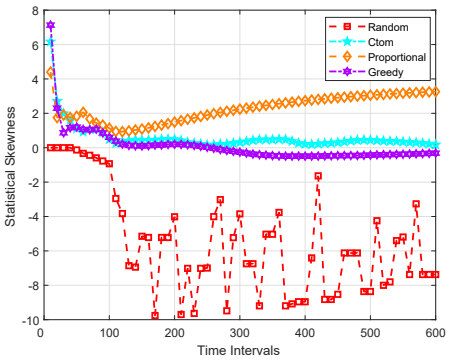
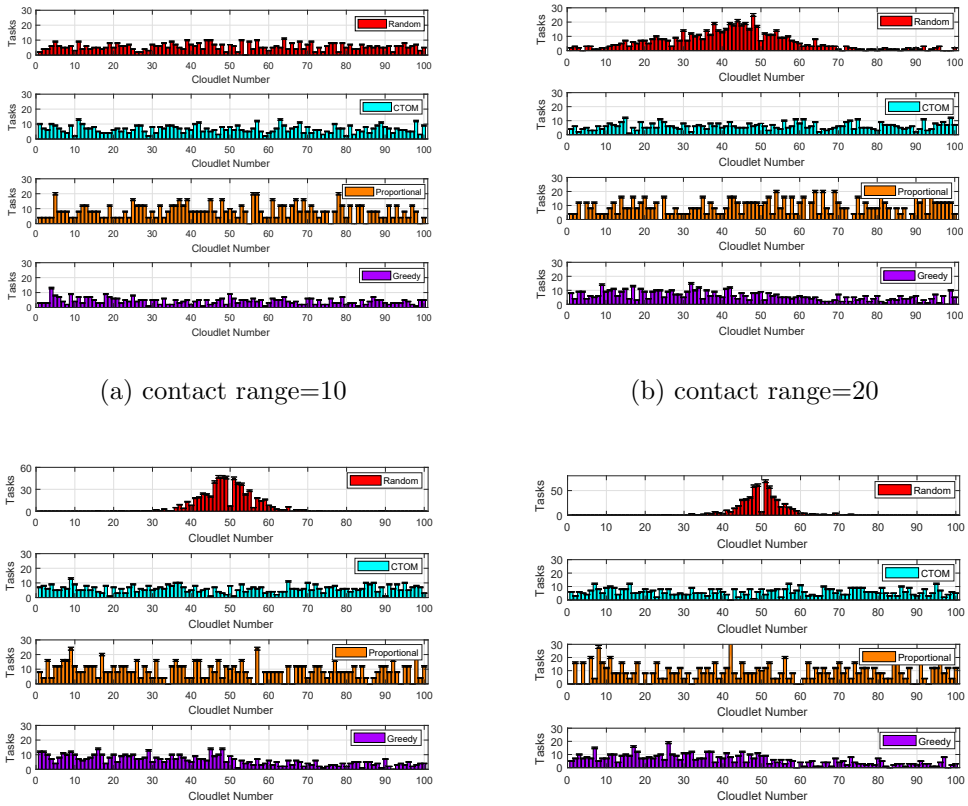


Figure 3.5 : The statistical skewness metrics obtained with the four schemes.

5) even at idle state. Similarly, the task allocation obtained by the proportional allocation is also extremely unbalanced, where the distribution of overwhelmed (with over 25 remaining tasks) cloudlets is more sparse. In contrast, under CTOM and the greedy allocation, mobile cloudlets are equally allocated with tasks (around or below 10). The proposed CTOM outperforms the conventional random and proportional allocation schemes in reducing the long task queues by 65% and 55%, respectively. In this way, the potential DDoS attack tasks will be effectively processed and filtered out from the cloudlet network. Fig. 3.3 demonstrates the task allocation performance of the four methods in cumulative distribution. Under the schemes of random allocation and proportional allocation, about 30% of mobile cloudlets are allocated with more than 10 tasks, which will affect the overall task response time. Meanwhile, our CTOM performs closely to the greedy method in balanced task offloading, where nearly 90% cloudlets are with task load under 10, and 55% cloudlets are offloaded with 5 to 10 tasks.

We further evaluate the task offloading performance using the imbalance metric [73], and the imbalance percentage and statistical skewness are calculated as in 3.2. The lower imbalance metric means the better balance performance in task allocation, *i.e.*, the lower ratio of maximum and average task loads. From Fig. 3.4, it is obvious that the greedy algorithm achieves the best performance in terms of imbalance metric, which converges to almost 0. The imbalance metrics of our pro-



(a) contact range=10

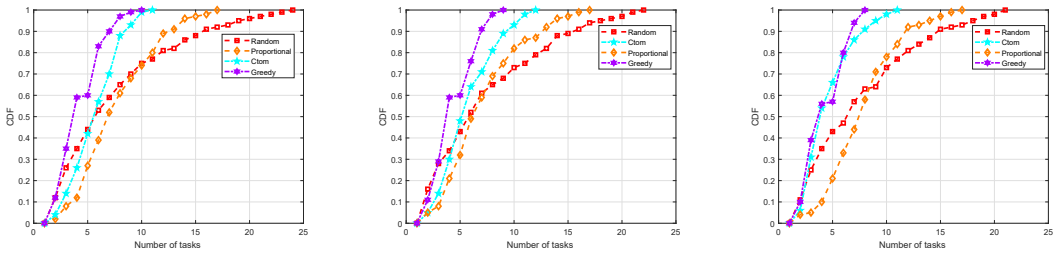
(b) contact range=20

(c) contact range=30

(d) contact range=50

Figure 3.6 : Load analysis with the inter-contact range r ranging from 10-50.

posed CTOM and the proportional allocation scheme converge to 0.1 and 0.25, respectively. The random allocation scheme performs worst with an imbalance metric 0.5. Meanwhile, Fig. 3.5 shows the statistics of the skewness obtained by the four schemes, where a positive or negative skewness indicates that the quantities of the mobile cloudlets having a higher or lower task load than average, respectively. In Fig. 3.5, we can observe that the greedy allocation and our CTOM have achieved the ultimate skewness values at about 0, which means that there are few cloudlets with an unbalanced load. As a contrast, the proportional method has a skewness of 2. The random allocation's skewness fluctuates violently in a negative range (from -10 to 0), which means there are many mobile cloudlets with much lower task load than the average. which means there exist many mobile cloudlets with much lower task load than the average. For proportional allocation, the skewness varies from



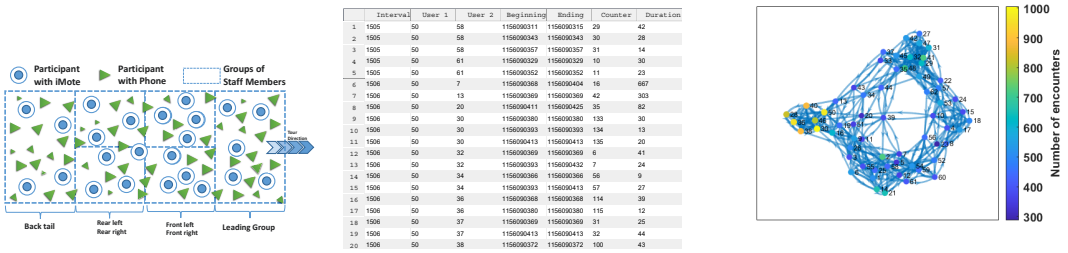
(a) d=4 (b) d=8 (c) d=16

Figure 3.7 : Load analysis with the number of choices d ranging from 4-16.

about 4 to 1, revealing that there are also many overloaded mobile cloudlets.

Analysis of parameters

We further evaluate the influence of the d in d-choice and the value of the inter-contact range on the load allocation performance. Firstly, we show the task offloading results with a contact range from 10 meters to 50 meters in Fig. 3.6. It is quite obvious that when the contact range increases, the tasks in random allocation are more centralised at a few mobile cloudlets, resulting in an unbalanced task distribution. Moreover, the proportional method performs poorly for all contact ranges, where a significant number of cloudlets are overloaded (with up to 30 tasks) or at idle. The performance of CTOM and the greedy method are sustainable, where the overall task allocation is balanced and well distributed (most of the cloudlets are with around 10 tasks). Secondly, we investigate the number of choice d. In Fig. 3.7, we plot the CDF of task allocation results with different values of d. From Figs. 3.7a to 3.7c, the CDF lines of all methods pullback (maximum load decreases) as d increased. With greater values of d, in each time interval, one mobile cloudlet may have more options to offload its tasks, so as to achieve sustainable task allocation. The above simulation results demonstrate that the proposed CTOM can achieve balanced task allocation, and in this way, the overall tasks can be processed concurrently. CTOM improves mobile cloudlets' utilisation efficiency and shortens the task response time, thus handling the potential DDoS attack tasks smoothly.



(a) Diagram of device deployment in roller tour (b) The encounter dataset of iMote (c) The node-relation graph in iMote trace dataset

Figure 3.8 : The illustration of the iMote dataset.

3.6.2 Trace-driven Evaluation

We further explore the balanced task allocation in the trace-driven evaluation. We use a mobility dataset called RollerNet [72]. The RollerNet dataset was collected during a rollerblading tour with 15,000 participants in Paris, France. The entire tour lasted for three hours and travelled 20 miles, covering the major metropolitan area in Paris.

Basic setups

Our real-world evaluation is based on the real-world trace dataset for mobility-enhanced cloudlets, named RollerNet, which includes the traces of opportunist sightings by wireless networking nodes called iMotes. The iMotes were distributed to a group of people to collect any opportunistic sighting of other mobile devices (including the other iMotes) via Bluetooth. We drew a sample diagram of iMote deployment, as depicted in Fig. 3.8a, where a total of 62 skaters were equipped with iMotes, and they were divided into 6 groups at different regions in the roller crowd. In this evaluation, we consider each iMote as a mobile cloudlet that can remotely execute computing tasks for mobile users. For cloudlet i with a service rate of μ_i, we assign the service rate by sampling the normal distribution $N(6, 2) > 0$. The number of servers at cloudlet i is sampled from a Poisson distribution with a mean of 3. The task arrival rate λ_i follows a normal distribution $0 < N(18, 6) < s_i \cdot \mu_i$, where s_i is the number of servers at mobile cloudlet i. We assume that there are potential

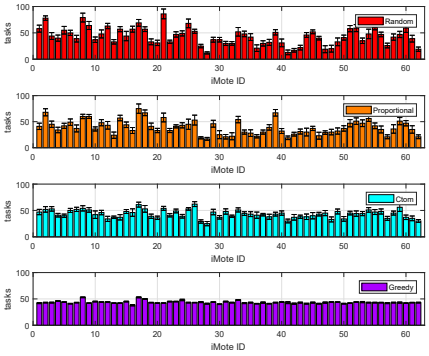


Figure 3.9 : Task load results in the trace-driven evaluation.

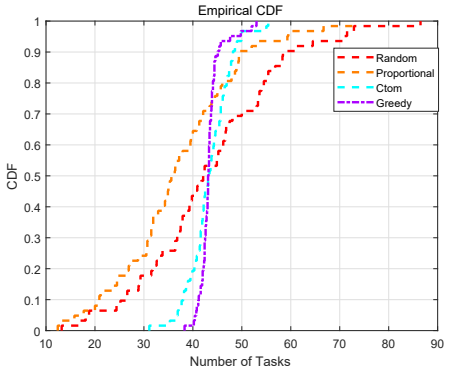


Figure 3.10 : Load distribution in trace-driven evaluation.

DDoS attackers in this rollerblading tour, and the extreme task arrival rate reveals the attack tasks. All of the settings are derived according to [49]. Meanwhile, our evaluation is based on a real-world trace dataset, and the cloudlets are enhanced with mobility.

We conduct a twofold pre-processing on the RollerNet dataset. Firstly, we unify the timing of user encounter records. By setting a common starting time based on the earliest record, we convert the duration of all encounters into serial time slots by minutes. Based on the unified encounter records, we find that the total inter-contact time is $1567-1417 = 150$. We set the total time interval for task offloading to be 150. Secondly, we plot an encounter graph to depict the frequency of communications among all the iMote skaters in Fig. 3.8c. and we find that the iMote carriers can be roughly divided into three groups based on their communication frequencies, *i.e.*, active group (with 800-1000 contacts), common group (with 500-800 contacts) and passive group (with 300-500 contacts). The above division consists of iMote skaters' formation: skater association, staff, and a set of friends.

Evaluation performance

Fig. 3.9 shows the task allocation results in a bar graph obtained on RollerNet. As revealed from this figure, the performance of our CTOM method is comparable to that of the greedy allocation, where most of the mobile cloudlets are offloaded

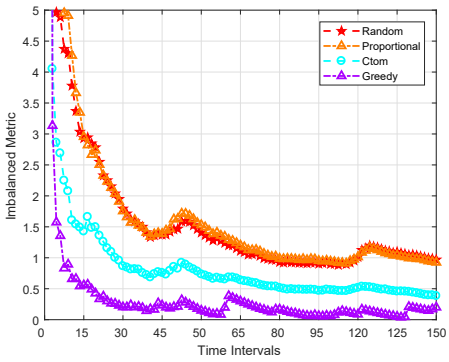


Figure 3.11 : The imbalance metric of different schemes.

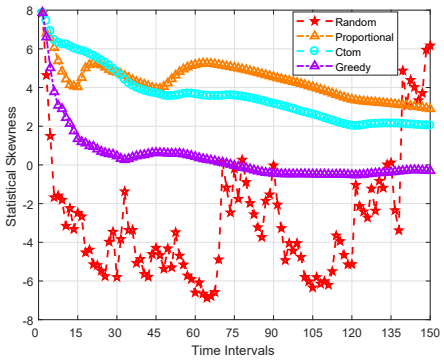
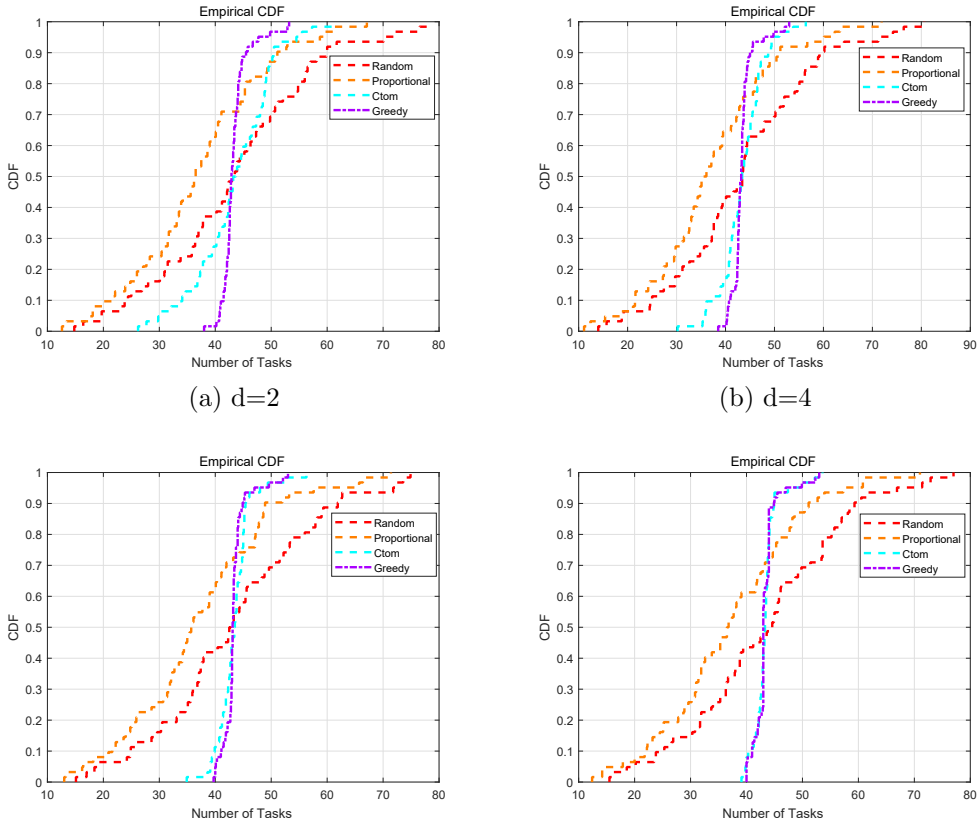


Figure 3.12 : The statistical skewness of different schemes.

with around 50 tasks. Meanwhile, in the random and proportional allocations, the allocation results are unbalanced with task loads fluctuating severely among different cloudlets (up to 80 and down to 10). In this case, the extremely overwhelmed iMotes can be viewed as attacked cloudlets, whose computing resources have been consumed by DDoS attack tasks.

Fig. 3.10 illustrates the cumulative distribution of task allocation. More than 30% of mobile cloudlets have more than 50 tasks in the random allocation scheme, and about 30% of others are with fewer than 30 tasks. Unbalance can result in longer average task response time. Meanwhile, under the CTOM scheme, around 95% of cloudlets are allocated with 30-50 tasks, so the cloudlets are collaboratively processing tasks, and no cloudlets are overwhelmed by DDoS attacks. As the greedy algorithm's CDF line is the most centralised, the task loads at different cloudlets only vary within a small range (around 40 to 50).

We further evaluate the percentage of the imbalance metric and the statistical skewness. In Fig. 3.11, the greedy algorithm achieves the best performance with a 0.2 imbalance value, followed by CTOM with converged results of 0.5. Interestingly, random and proportional allocation perform similarly with imbalance metric around 1, showing that both methods are not applicable in the trace-driven scenario. In Fig. 3.12, the skewness obtained by the random allocation scheme fluctuates violently between positive and negative values, so the task loads are continuous-



(a) d=2 (b) d=4

(c) d=8 (d) d=16

Figure 3.13 : Load analysis in trace-driven with the number of choices d ranging from 2-16.

ly unbalanced throughout the whole process of allocation. The greedy allocation scheme achieves the best performance with a skewness value of 0. The skewness values of CTOM and proportional allocation are 2 and 3, respectively, so there are overloaded mobile cloudlets. It is worth noting that within the first 50 intervals, the proportional method's skewness curve is lower than that of CTOM. After the 50th interval, the skewness curve of CTOM keeps falling while that of proportional method firstly shows a rapid rise then converges to lower values. The main reason for the above trend is that it would take some time for CTOM to show its advantage of two-choice offloading. The offloading strategy of the proportional method is straightforward and has an immediate effect, as revealed by the skewness curve.

We also evaluate the influence of d on trace-driven task allocation. Interestingly,

in Fig. 3.13, with d increasing from 2 to 16, the proposed CTOM performs more and more closely to greedy method's results, with most of the mobile cloudlets offloaded with 40 to 50 tasks.

The above simulation and evaluation results validate proposed CTOM in balancing task loads among mobile cloudlets. Under CTOM, the total number of overloaded cloudlets are significantly reduced, and the gaps between the longest and the shortest task queues are also narrowed. In this way, the DDoS tasks cannot overwhelm any cloudlet to prevent legitimate users from accessing computing resources. In summary, CTOM can efficiently tame the potential DDoS attacks to achieve secure and sustainable task offloading.

3.7 Conclusion

This chapter has investigated the load balancing problem in mobile cloudlet networks. By leveraging balls-and-bins theory, we have devised 'CTOM', a collaborative task offloading scheme for mobile cloudlet networks. By locally querying limited task load information, the proposed solution can reduce the longest task queue in the allocation process effectively. Our simulation and trace-driven evaluation results have demonstrated that our CTOM performs exceedingly close to the optimal solution in load balancing and outperforms the conventional random and proportional allocation schemes by 65% and 55% respectively in task gaps. In contrast, its computing complexity has been largely reduced at each time interval.

Chapter 4

FairEdge: A Fairness-oriented Task Offloading Scheme for IoT Applications in Mobile Cloudlet Networks

Mobile cloud computing has emerged as a promising paradigm, as it efficiently facilitates the computation-intensive and delay-sensitive mobile applications. Computation offloading services at the edge mobile cloud environment are provided by small-scale cloud infrastructures such as cloudlets. While offloading tasks to in-proximity cloudlets can benefit mobile users with lower latency and smaller energy consumption, new issues are arising related to the cloudlets. For instance, unbalanced task distribution and huge load gaps among heterogeneous mobile cloudlets are becoming more challenging, concerning the network dynamics and distributed task offloading. In this chapter, we propose 'FairEdge', a Fairness-oriented computation offloading scheme to enable balanced task distribution for mobile Edge cloudlet networks. By integrating the balls-and-bins theory with fairness index, our solution promotes effective load balancing with limited information at low computation cost. The evaluation results from extensive simulations and experiments with real-world datasets show that FairEdge outperforms conventional task offloading methods, and it can achieve network fairness up to 0.85 and reduce the unbalanced task offload by 50%.

4.1 Introduction

In recent years, with the rapid development of mobile computing technologies and pervasive proliferation of mobile devices, mobile traffic data has been growing at an unprecedented rate. According to a latest white paper released by Cisco [22], the global mobile traffic data will increase seven-fold between 2017 and 2022, reach-

ing 77 exabytes (1 exabyte = 10^{18} bytes) per month. Notably, of all IP traffic in 2022, over 50% will be Wi-Fi and smartphones will account for nearly 60% traffic offloading. While mobile applications are aggressively demanding in computation resources, mobile devices are still constrained by the limited capacities in the batteries, memory, and processers. As a consequence, the enlarging gap between resource-constrained mobile devices and computing-intensive applications has become a great challenge [79]. Nevertheless, cloud computing is the ultimate solution to deal with this challenge.

Generally, cloud computing allows mobile users to offload computation tasks*, *i.e.*, the executable application phases, on to cloud computing infrastructures (*i.e.*, IaaS, PaaS, and SaaS). In the scenario of mobile computing, by migrating computing-intensive tasks to the cloud, mobile devices can benefit from lower energy consumption and enjoy virtually unlimited computing capacity. This is exemplified by a wide range of cloud computing platforms, including Amazon Web Services, Microsoft Azure, and Google Cloud [30]. These cloud computing platforms provide computing services that can be remotely accessed by mobile users. However, existing studies have shown the limitations of solely relying on offloading tasks to remote clouds. Since mobile users access remote clouds via a wide area network (WAN), they may experience long latencies caused by congested transmission over long distances between end devices and clouds [30].

Subsequently, the concept of mobile edge computing (MEC) has been proposed to provide edge users with in-proximity computing resources, such as cloudlets [79]. A mobile cloudlet is a trusted, resource-limited cluster of computing servers, which is integrated with wireless local area networks (WLAN). By offloading tasks to a nearby mobile cloudlet, the demands of fast and interactive response can be sufficiently satisfied with the low-latency, one-hop, and high-bandwidth access. In comparison with remote cloud computing resources, the mobile cloudlets at the edge network can improve the task processing time significantly. As a result, mobile users of

*In the remainder of this chapter, we will use the terms task offloading and computation offloading interchangeably unless otherwise stated.

computation-intensive applications [80] (*e.g.*, virtual reality, image processing, and augmented gaming) can enjoy faster response and better Quality of Services (QoS), and enhanced Quality of Experience (QoE) [81].

However, considering the capacity of each cloudlet is limited, a mobile cloudlet would become overloaded if it travels in an area where too many mobile users offload computation-intensive tasks. In that case, above QoS and QoE for mobile users can be seriously impacted, making the communication cost and delay even higher than offloading tasks to a remote cloud. Therefore, it is greatly important to maintain load balancing among all mobile cloudlets at the edge, so that each cloudlet's computing resource can be fully exploited, and mobile users can also have a quick response on their offloaded tasks.

Unfortunately, most existing solutions for improving the performance of edge networks have overlooked a fundamental issue, *i.e.*, the fairness of task offloading among mobile cloudlets. Indeed, it is difficult to achieve fairness in task offloading among mobile cloudlets, as the mobility of each cloudlet is random, and the network is intermittently connected. Moreover, as computation offloading behaviours of mobile users are uncontrollable, the task load of each cloudlet is highly dynamic, making it costly to probe the overall information in the cloudlet network for load comparison and decision making. Accordingly, two challenges need to be formally addressed.

- First, load balancing should be achieved under the collaboration among mobile cloudlets. As the mobility-enhanced cloudlets opportunistically encounter each other, it is essential for them to collaboratively offload tasks to each other for the benefit of overall load balancing in edge networks.

- Second, it should be low-cost and lightweight to achieve fairness in the mobile cloudlet network. Therefore, a universal fairness metric should be adopted to measure the fairness based on the load information of each cloudlet. The fairness metric should be further taken into consideration when mobile cloudlets are offloading tasks to each other. Moreover, the fairness value of the cloudlet

network should be updated in each time interval, as load information of each cloudlet constantly changes.

In this chapter, to deal with the above challenges, we propose FairEdge, a Fairness-oriented task offloading scheme for collaborative mobile cloudlets at Edge networks. FairEdge integrates the balls-and-bins theory [82] with fairness index [83] to achieve effective load balancing in mobile cloudlet networks. Notably, under the FairEdge scheme, each cloudlet only needs to query load information from two random neighbours in each time interval. By comparing the task load and fairness indexes of these two neighbours, each cloudlet can make a practical decision on task offloading to preserve both load balancing and fairness. Ultimately, the fairness of the mobile cloudlet network will converge, and the fairness-oriented load balancing can be achieved.

The main contributions of this chapter are summarised as follows.

- To the best of our knowledge, this work is the first to investigate the fairness issue in mobile cloudlet networks. By jointly considering the balancing property as well as the fairness index, we propose the FairEdge scheme based on *balls-into-bins* theory and Jain's fairness index. The task load information of each cloudlet is collected and compared in a low-cost manner, which resolves the difficulty in information collection from highly dynamic mobile edge networks.

- The Jain's fairness index is integrated as a part of the task offloading algorithm. By leveraging the task load information and fairness index of two targeted neighbours, the proposed FairEdge scheme enables a more reasonable computation offloading decision for each cloudlet. Fairness-oriented task offloading further contributes to the overall load balancing and fairness of the mobile cloudlet network.

- We evaluate the proposed FairEdge with simulations and experiments based on two real-world trace datasets. The evaluation results show that FairEdge

can successfully achieve load balancing with guaranteed performance, with a near-optimal fairness index value of 0.85 and an improvement of 50% in balancing tasks among mobile cloudlets.

The rest of this chapter is organised as follows. We introduce the related works and preliminaries on investigated issues in Section 4.2 and Section 4.3, respectively. Then, we present the system model and task offloading problem in Section 4.4. We further propose the FairEdge algorithm with detailed descriptions in Section 4.5. In Section 4.6, we present comprehensive simulation studies with real-world datasets. We conclude this chapter in Section 4.7.

4.2 Related Work

The comprehensive reviews on mobile edge computing can be found in [26]. Particularly, Nayyer et al. [79] compared the mobile augmentation approaches for resource optimisation from the perspective of cloudlet-based networks. Generally, existing studies for cloudlet load balancing can be categorised into two groups, *i.e.*, optimal cloudlet placement and computation offloading optimisation.

For computation offloading optimisation, the objects of offloading algorithms include optimising device energy [84], cloud workload [85], and application latency [3]. For instance, Sun et al. [86] proposed a latency-aware workload offloading strategy to balance tasks from mobile users to suitable cloudlets. Huang et al. [87,88] investigated service provisioning problems under the cloudlet-based network and proposed an adaptive update scheme to maximise a weighted profit for network operators. Yang et al. [11] jointly considered security and sustainability issues of cloudlet networks and proposed a novel task offloading scheme to avoid DDoS attacks. Similarly, Fan et al. [1] proposed CTOM, a collaborative task offloading mechanism for mobile cloudlet networks.

For computation offloading in mobile edge computing scenarios, Du et al. [89] investigated the computation offloading problem in a mixed fog/cloud system by jointly optimising the offloading decisions and allocation of computation resource, transmit power, and radio bandwidth. Zhang et al. [83] studied the fair task of-

floading for fog computing networks, where the task delay and corresponding energy consumption were formulated into the performance index with fairness scheduling metrics. Moreover, Zhu et al. [90] formulated the fair resource allocation problem in mobile edge computing as a resource allocation game with Nash bargaining. Meskar et al. [91] designed a multi-resource allocation mechanism by jointly considering dominant resource fairness and external resources fairness. Different from the existing works, in this chapter, we study the fairness in a mobile edge network, where mobility-enhanced cloudlets collaborate to offload computation tasks to each other. We aim to achieve both fairness and load balancing for the overall edge network of all mobile cloudlets.

4.3 Preliminary

In this section, we first introduce the preliminary of computation offloading in mobile edge networks, and then, we explain the theory of balls-into-bins for modelling the task offloading in mobile edge computing scenarios.

4.3.1 Mobile Offloading with Edge Computing

With the proliferation of mobile devices and advances in wireless communication technologies, mobile computing has experienced a significant shift from centralised cloud computing to mobile edge computing [26]. In a typical mobile edge network, edge cloud servers (*e.g.*, cloudlets) are deployed at fixed locations or enhanced with mobility to provide computing services for nearby mobile users with proximate, high-speed and wireless access. Subsequently, mobile users can offload computation-intensive and latency-sensitive tasks to edge cloud servers for processing, thus saving both energy and computation resources on their own devices [79]. However, computation offloading in mobile edge networks also brings new challenges in how to efficiently utilise edge computing resources and enhance the overall performance of cloudlets. In this work, we investigate how to improve resource sharing via cooperation and collaboration among mobile cloudlets based on the assumption that computation tasks can be offloaded from one cloudlet to others for more efficient processing. In particular, we adopt balls-into-bins theory for task distribution

among mobile cloudlets, to optimise the offloading decisions in a distributed manner with low communication and computation cost.

4.3.2 The Two-Choice Paradigm of Balls-into-Bins Process

Balls-into-bins is a classic process to model task distribution among a group of uniform servers [82]. In this study, we adopt the load balancing theory of balls-into-bins process to assist the fairness-oriented task offloading for IoT applications in mobile cloudlet networks. The original goal of balls-into-bins processes is to allocate m balls into n bins, with each ball to be thrown into a uniformly and randomly selected bin at a probability of $1/n$. Based on this allocation process, the key criterion of load balancing in a balls-into-bins process is the maximum load, *i.e.*, the largest number of balls in any bin $\mathbb{M}$. Firstly, when $m = n$ and the task offloading is random, with high probability, the expectation of maximum load $\mathbb{M}$ is [92]:

$$\mathbf{E}(\mathbb{M}) = \mathrm{O}(\frac{\log n}{\log \log n}). \tag{4.1}$$

Meanwhile, if each ball has a chance to query the load information from d random selected bins and then makes allocation decisions based on load comparison of the above d bins, the maximum load can be dramatically decreased. By comparison, if each ball is allocated to the least loaded of among d bins, the maximum load can be reduced to [92]:

$$\mathbf{E}(\mathbb{M}) = \frac{\log \log n}{\log d} + \mathrm{O}(1). \tag{4.2}$$

Similarly, in a more general case, when $m \gg n$ and the task offloading is random, then with high probability, the maximum load is [92]:

$$\mathbf{E}(\mathbb{M}) = \frac{m}{n} + \mathrm{O}(\sqrt{\frac{m \log n}{n}}). \tag{4.3}$$

If the task offloading is based on the load comparison of d random choices, with high probability, the maximum load is reduced to [92]:

$$\mathbf{E}(\mathbb{M}) = \frac{m}{n} + \frac{\log \log n}{\log d} + \mathrm{O}(1). \tag{4.4}$$

In this study, we use the balls-into-bins process to model load balancing with computation tasks (*e.g.*, balls) and mobile cloudlets (*e.g.*, bins), and we explicitly adopt the 'two-choice' paradigm for low-cost communication and computation. Accordingly, supposing that m user tasks are to be distributed into n mobile cloudlets, each task can be offloaded into the least loaded of $d = 2$ cloudlets independently and uniformly. When $m = n$, with high probability the maximum load is [70]:

$$\mathbf{E}(\mathbb{M}) = \mathrm{O}(\log \log n). \tag{4.5}$$

When $m \gg n$, with high probability the maximum load of any cloudlet is [70]:

$$\mathbf{E}(\mathbb{M}) = \frac{m}{n} + \mathrm{O}(\log \log n). \tag{4.6}$$

Consider a mobile edge network where tasks are allocated into n mobile cloudlets by following an arrival rate λ, with the random choice in task offloading, the maximum load under an arbitrary round t is [93]:

$$\mathbf{E}(\mathbb{M}) = \mathrm{O}(\frac{1}{1-\lambda} \cdot \log \frac{n}{1-\lambda}), \tag{4.7}$$

where $\lambda = \lambda(n) < 1$. As we mainly focus on the task redistribution among all mobile cloudlets, we further specify the number of random choices as 2 to enhance the adaptability and scalability of balls-into-bins theory. Accordingly, each mobile cloudlet can randomly and independently choose 2 nearby cloudlets within its inter-contact range as candidates for task offloading. As a result, with a fixed arbitrary round t, the theoretical maximum load of any cloudlet becomes [93]:

$$\mathbf{E}(\mathbb{M}) = \mathrm{O}(\log \frac{n}{1-\lambda}), \tag{4.8}$$

where $\lambda = \lambda(n) \in [1/4, 1)$.

4.4 System Model and Problem Definition

In this section, we consider a mobile edge network for cooperative task offloading. First, we model the mobile cloudlet and user task offloading. Then, we formulate the fairness-oriented load balancing problem for a mobile edge network.

4.4.1 Edge Cloudlet Model

In this study, we consider a mobile edge network in an urban area, which consists of (1) a group of edge mobile cloudlets that are integrated with AP for data transmission and task processing, and (2) a number of mobile users that periodically send computation tasks to nearby cloudlets for task processing. First, we denote k edge mobile cloudlets by $\{1, 2, ..., k\}$, the location for each cloudlet as (x_i, y_i), and each mobile cloudlet is enhanced with random mobility to have opportunistic encounters with other cloudlets and mobile users. Moreover, we model each cloudlet i as an $M/M/n$ queue with reference to [49], *i.e.*, each cloudlet i has n_i servers with the service rate μ_i. Specifically, a cloudlet i stores the offloaded tasks as an FIFO queue, with the length of q_i^t at time t. In the edge cloudlet network, computation offloading by mobile users to each cloudlet i is modelled as a Poisson process with task arrival rate λ_i, as the number of tasks would constantly change at each time interval. During time interval t, the response time of a cloudlet i can be calculated as $\left\lceil \frac{q_i^t + \lambda_i}{\mu_i} \right\rceil$. Moreover, each cloudlet i also stores information on the number of tasks offloaded to another cloudlet j as $s_{j,i}$.

4.4.2 Task Transmission Model

In this chapter, we assume that each cloudlet can contact with other nearby cloudlets to exchange load information and redirect tasks. As some mobile cloudlets may be overloaded with user tasks, the tasks stored in them could experience long processing delays, which would degrade the service experience for the corresponding mobile users. Therefore, the task transmission model is formulated for mobile cloudlets to perform computation offloading for load balancing collaboratively. From the perspective of cloud service provides, it is also important to enhance the perfor-

mance of mobile cloudlets to make the edge network more efficient and sustainable. The task transmission model is formulated to address the above issues with the following two considerations. First, only when the distance between two cloudlets is within an inter-contact range R, they can establish an intermittent connection. Second, based on [9], the connecting probability of cloudlets i and j is computed by:

$$P_{i,j}(t_a, t_b) = e^{-\frac{1}{\alpha_{i,j}} \cdot t}, t \geq 0, \tag{4.9}$$

where $\alpha_{i,j}$ is the pairwise connection rate of an exponential distribution of $f(t) = \frac{1}{\alpha_{i,j}} \cdot e^{-\frac{1}{\alpha_{i,j}} \cdot t}$, t_a and t_b are any two consecutive time points. Then, based on the Jain's fairness index [83], we calculate the value of fairness index for each cloudlet i by:

$$f^t(i) = \frac{(\sum_{j=1}^{k} s_{j,i}^t)^2}{k \sum_{j=1}^{k} s_{j,i}^2}, \tag{4.10}$$

where k is the total number of edge cloudlets. Likewise, we further calculate the value of fairness index for the mobile edge network by:

$$F = \frac{(\sum_{i=1}^{k} q_i^t)^2}{K \sum_{i=1}^{k} (q_i^t)^2}, \tag{4.11}$$

where q_i^t is the number of loaded tasks in cloudlet i at time t. The main difference between the proposed fairness and the load balancing is that fairness emphasizes each cloudlet's equality to get tasks. In contrast, load balancing reveals the offloading results on the overall cloudlet network.

4.4.3 Problem Definition

The fairness-oriented load balancing problem in a mobile cloudlet network is defined as follows. Given a set of k mobile edge cloudlet $\{1, 2,..., \mathrm{k}\}$, where each cloudlet i with service rate μ_i. Each cloudlet performs a random walk to collect random user tasks with arrival rate λ_i, which follows a Normal distribution. Meanwhile, for each mobile cloudlet i, it has a fairness index value $f^t(i)$. When two cloudlets encounter each other, they collaboratively share load information and fairness val-

ues. Then, they would perform fairness-oriented task offloading to enhance the load balancing of the mobile edge network.

Load Balancing Problem: The objective of fairness-oriented task offloading is to minimise the differences among task queues of all cloudlets so that the user tasks can be processed with the maximum utilisation rate of edge cloudlet computing resources. Here, we formulate the optimisation function with each cloudlet's task queue by:

$$\arg\min(\parallel q_i - \bar{q} \parallel), \forall i \in \{1, ..., k\}, \tag{4.12}$$

where $\bar{q}$ is the averaged value of the task queues of all mobile cloudlets.

Fairness Optimisation Problem: Maximising the fairness value for each mobile cloudlet can further enhance the efficiency and sustainability of a mobile edge network. Note that the lowest fairness value in a task offloading process among k mobile cloudlets is $\frac{1}{k}$. On the contrary, the highest fairness index value is 1, corresponding to the most balanced task offloading result that all mobile cloudlets hold the same number of user tasks for processing. The fairness optimisation problem is as follows:

$$\arg\max(f^t(i)), \forall i \in \{1, ..., k\}, \tag{4.13}$$

where $f^t(i)$ is the fairness value of cloudlet i during time interval t, based on Equation 4.10.

4.5 Algorithm Design

4.5.1 Algorithm Overview

To tackle the load balancing problem and fairness optimisation problem in mobile cloudlet networks, we propose a heuristic algorithm called FairEdge. The major issue of achieving fairness-oriented computation offloading for mobile edge networks is the opportunistic encountering of mobile cloudlets. It would be costly in both computation and communication to control and regulate the task offloading process for all mobile cloudlets in a centralised manner. In contrast, a distributed task offloading scheme is more desirable, since each mobile cloudlet can collaboratively

share its load information and fairness value nearby cloudlets. Moreover, with new incoming tasks at each cloudlet, the load information of the edge network constantly changes. To collect the above information and broadcast it to all mobile cloudlets could result in intensive overhead for the edge networks. Last but not least, for each mobile cloudlet, it only needs the load information from nearby and contactable targets when making computation offloading decisions. To address the above concerns, we are inspired by the 'balls-into-bins' theory and further adopt the 'two-choice' paradigm to design the FairEdge algorithm.

In general, we have three basic assumptions over the mobile edge network. First, each cloudlet i receives user tasks that follow a Poisson process of λ_i. Meanwhile, these tasks are executable and offloadable to any other mobile cloudlet for processing. Second, we assume that the mobility trajectory of each cloudlet follows a random walk process within the edge network area. In each time interval, a mobile cloudlet is contactable to any other cloudlet within its communication range. Third, according to [50], the duration of the time interval is long enough for each mobile cloudlet to perform a complete computation offloading.

With the above considerations, according to the models in Section 4.4, we devise an algorithm that enables each mobile cloudlet i to randomly select d target cloudlets within its communication range for computation offloading in each time interval. By probing and comparing load information from d nearby cloudlets, each mobile cloudlet i selects the least loaded one as the target for computation offloading. Then, the fairness index value of the target cloudlet will be computed based on Equation 4.10 and further compared with the fairness index value of the overall mobile cloudlet network. The computation offloading decision will be made based on the above comparison result. In the following, we formally present FairEdge, the fairness-oriented computation offloading algorithm for mobile cloudlet networks in Algorithm 2.

4.5.2 FairEdge Algorithm Design

The FairEdge algorithm is proposed to achieve fairness-oriented task offloading in mobile cloudlet networks. To begin with, we define the input and output of FairEdge according to the edge cloudlet model and task transmission model. Next, the algorithm initialises the task queue q_i and record of task offloading $\mathbf{S}_i$ for each cloudlet i as well as the time interval t. Starting from the first time interval, FairEdge generates a random location for each cloudlet i and calculate each cloudlet's corresponding task load q_i at the current time interval. At this stage, the fairness index value f of the mobile edge network is also calculated using the updated load information of all cloudlets. Next, each cloudlet i will send a probing message and add other cloudlets within communication range into its contact list c_i. To adopt the balls-into-bins process for task offloading, FairEdge uses the d-choice policy to select d potential offloading targets in its contact list c_i randomly and further chooses the least loaded one as the computation offloading target. By comparing the fairness index value f_{choice_i} of the chosen target with f, FairEdge will decide whether to allow cloudlet i to perform task offloading to choice_i or not. If $f_{choice_i} \geq f$, the task offloading will be executed, and the attributes of cloudlet i and the target cloudlet will be updated.

The above process will iterate for each mobile cloudlet i and repeatedly execute for T time intervals. Finally, the FairEdge will output the ultimate task queue q_i and fairness index value f_i for each cloudlet i as well as the ultimate fairness index value of the mobile edge network. Note that, the d-choice here is presented for general computation offloading with balls-into-bins theory. In practice, to reduce the communication cost and computation cost in the task offloading process, we apply the '2-choice' paradigm. Thereby, the FairEdge algorithm will only allow each cloudlet i randomly choose 2 contactable cloudlets for load comparison in each time interval.

At last, we briefly discuss the theoretical performance of FairEdge. First, as have been discussed in Section 4.3.2, the mobile cloudlet network fits the case where user tasks follow an arrival rate λ into k cloudlets. With the random choice in

Algorithm 2 FairEdge Algorithm

Input:

The number of cloudlets k, time slots T, random choices d; servers, task arrival/service rates of cloudlet i: n_i, λ_i, μ_i; boundaries: a and b, contact range: r.

Output:

Each cloudlet's: contact list c_i, task load q_i, fairness index f_i, overall fairness index: f, offload target: choice$_i$, offloading record: $\mathbf{S}_i = s_{1,i}, ..., s_{k,i}$.

Initialise q_i=0, $\mathbf{S}_i$=$\emptyset$, t=0;
while $t \leq T$ **do**
  Generate random location for each cloudlet i as: (x_i, y_i), where $0 < x_i < a$, $0 < y_i < b$;
  Calculate task load q_i for i with m, μ_i and λ_i;
  Calculate fairness index f with Equation 4.11;
  Select offloading targets for each cloudlet i:
  while $j \leq$ k **do**
    if $(x_i - x_j)^2 + (y_i - y_j)^2 < r^2$ **then**
      add j into c_i as $c_i^j = 1$;
    end if
  end while
  if $\|c_i\| \geq d$ **then**
    do: randomly choose d cloudlets from c_i;
    choice$_i$ is the least load in d chosen cloudlets;
  else if $0 < \|c_{ij}\| < d$ **then**
    choice$_i$ is the least load in $\|c_{ij}\|$ cloudlets;
  else if $\|c_{ij}\| = 0$ **then**
    skip task offloading for cloudlet i in this round;
  end if
  if $f_{choice_i} \geq f$ **then**
    cloudlet i performs task offloading to choice$_i$;
  end if
  j = choice$_i$;
  $s_{j,i} = s_{j,i} + 1$, $q_j = q_j + 1$;
  update f_i and f_j;
  $t = t + 1$;
end while
return task load q_i, offloading record $\mathbf{S}_i$.

task offloading, the maximum load under an arbitrary round t would be $\mathbf{E}(\mathbb{M}) = \mathrm{O}(\frac{1}{1-\lambda} \cdot \log \frac{k}{1-\lambda})$, where $\lambda = \lambda(n) < 1$. For the 2-choice process, if $\lambda = \lambda(n) \in [1/4, 1)$, the maximum load of any cloudlet becomes $\mathbf{E}(\mathbb{M}) = \mathrm{O}(\log \frac{k}{1-\lambda})$. Second, by leveraging the '2-choice' paradigm for selecting the target for computation offloading, FairEdge only probes load information from two contactable neighbouring cloudlets for comparison. According to [94] and [9], such a process would significantly reduce the complexity overhead to O(1) compared with greedy offloading's O(n) complexity.

4.6 Experimental Studies

In this section, we evaluate the performance of FairEdge with simulations and trace-driven evaluations. We first introduce the basic setups of simulation experiments and then present the evaluation results.

4.6.1 Simulation study

Simulation Setup

According to the mobile edge network model in Section 4.4 and FairEdge Algorithms in Section 4.5, we develop a simulation environment by referencing [1]. The fairness-oriented task offloading scheme is simulated in a 20 km^2 region, and we set the number of mobile cloudlets to 100, the total number of time slots as 600, and the contact range of mobile cloudlets to 20 meters. For each cloudlet i, we set the number of its server n_i by sampling the Poisson distribution with a mean of 2 as well as its service rate μ_i by sampling from the Normal distribution $\mathcal{N}(2, 1) > 0$. Meanwhile, the mobile user's task arriving rate at cloudlet i is sampled from the Normal distribution $\mathcal{N}(4, 2) > 0$. We adopt three baseline methods for comparison, including random task offloading, proportional task offloading [94] and greedy task offloading [9]. We run the simulation codes on a Dell laptop with Intel Core i5 CPU, 8GB RAM. Each simulation is executed 100 times, and we report the final average results as follows.

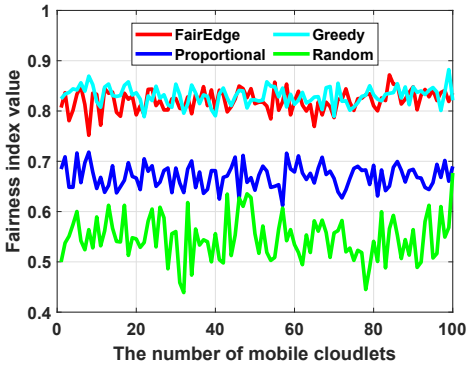
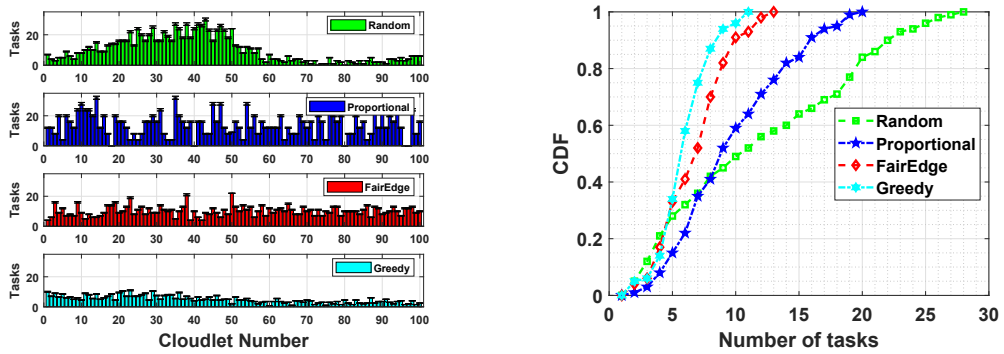


Figure 4.1 : The fairness values of all mobile cloudlets in task offloading collaborations.

Evaluation of Fairness Index

We first evaluate the fairness of task offloading by calculating the fairness index of individual mobile cloudlet using Equation 4.10. The fairness index ranges from 0 to 1, with 0 as the most unfair case and 1 as the purely fair case. As shown in Fig. 4.1, under the random offloading scheme and proportional offloading scheme, most of the fairness index values of mobile cloudlets are below 0.6 and 0.7, respectively. In comparison, the proposed FairEdge and greedy algorithm achieve an average value of fairness index over 0.8, showing that the task distribution is well-balanced across all mobile cloudlets. The greedy algorithm applies node traversal on each cloudlet when finding the least loaded and contactable neighbours for task offloading. Therefore, it ensures both the balance and fairness of task offloading at high communication costs. Meanwhile, the proposed FairEdge adopts balls-into-bins theory with the 'd-choice' scheme in task offloading. It achieves a close-to-greedy performance in fairness values while significantly reduces computation complexity to $o(1)$, as each cloudlet only needs to probe load information from two random neighbours and make a one-time comparison. Moreover, the fairness index values of FairEdge at some cloudlets are higher than those of greedy offloading. In summary, the FairEdge can achieve network fairness up to 0.85 and reduce the unbalanced task offload by up to 50% in comparison with other baseline methods.



(a) Task distribution under different offloading methods

(b) CDF of task offloading results

Figure 4.2 : Comparison of task offloading results in a simulation study.

Evaluation of Task Distribution

Second, we evaluate the task distribution results under different task offloading algorithms. The bar plot of Fig. 4.2a shows the final task distribution after all offloading time intervals. Obviously, under both random task offloading and proportional offloading schemes, there are huge gaps (up to 30) among different mobile cloudlets. In random task offloading, a group of mobile cloudlets (number 15 to 50) are processing many more tasks than other cloudlets (*e.g.*, number 60 to 90). Meanwhile, under proportional offloading, the overloaded mobile cloudlets are distributed more dispersedly. The above unbalance in task distribution would not only harm the fairness of the mobile cloudlet network but also degrade the user experience, as cloudlets need more time to process all tasks. In comparison, the proposed FairEdge successfully enhances the balance in task offloading, as most cloudlets have nearly 10 tasks to process. The greedy method achieves the best performance in balancing task distribution at the cost of high communication and computation overheads, where most cloudlets are offloaded with less than 10 tasks, and no cloudlet is idle.

To make a further comparison, we present the empirical cumulative distribution results of task offloading in Fig. 4.2b. Here, the performance of FairEdge is very close to that of greedy offloading, where over 90% of mobile cloudlets are offloaded with less than 10 tasks. In contrast, the task offloading result by the proportional method

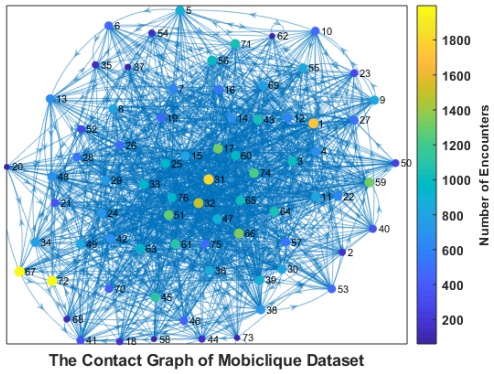


Figure 4.3 : Nodes and encountering records in the MobiClique [20] dataset.

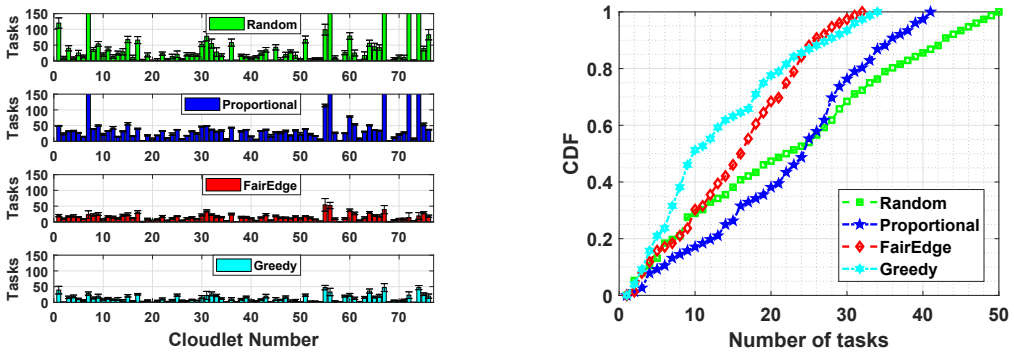
shows that almost 20% of cloudlets are offloaded with more than 15 tasks. Besides, over 20% of mobile cloudlets have more than 20 tasks to process under random offloading. The above evaluation results validate the effectiveness of FairEdge, as it manages to balance the task distribution by using the fairness index and '2-choice' paradigm. In the following, we further evaluate the FairEdge in real-world scenarios by using mobility trace datasets for simulation.

4.6.2 Evaluation of Real-world Trace Datasets

To explore the feasibility of FairEdge in real-world scenarios, we conduct trace-driven studies of mobile computation offloading with two real-world mobility trace datasets. In brief, the two trace datasets contain Bluetooth encounter records of mobile nodes that can be used to emulate the communications among mobile cloudlets at edge networks. The reasons for using two different trace datasets for evaluations are 1) to test the performance of FairEdge in different network scenarios, where cloudlets have different patterns of mobility; 2) to examine the scalability of FairEdge with mobile cloudlet networks in different scales. We present the details of each dataset and corresponding evaluation results of mobile computation offloading as follows.

MobiClique Dataset

Basic Setups. We adopt a real-world mobility dataset called 'MobiClique' [20] to emulate the random mobility of mobile cloudlets for task sharing and computa-



(a) Task distribution in MobiClique dataset (b) CDF of task offloading results in MobiClique dataset

Figure 4.4 : Comparison of task offloading results in trace-driven evaluations with the MobiClique dataset.

tion offloading. This dataset contains encountering records collected by a mobile network software called MobiClique. MobiClique leverages opportunistic contacts (*e.g.*, Bluetooth encounters) between smartphones to form a decentralised ad-hoc network for information sharing. The trace data of MobiClique was collected with 76 participants during the SIGCOMM 2009 conference in Barcelona, Spain. In Fig. 4.3, we consider each user a node and visualise the contact graph of MobiClique dataset. Here, each vertex represents a mobile cloudlet, an edge between two vertices represents a contact, and the colour of a vertex shows its active level in the network. In total, there are 76 mobile cloudlets and 69,186 contacts in the MobiClique dataset.

Moreover, the timestamp of the first contact is 30 seconds, and the timestamp of the last contact is 320,684 seconds. Based on the above, we set the length of a time slot for computation offloading to 200 seconds, so that there are 1,604 time slots in total. For each cloudlet i, we set its number of servers, service rate, and task arriving rate to be the same as those in the previous simulation setup.

Task Offloading Results. We conduct the mobile cloudlet task offloading with MobiClique dataset for 100 times and take the average values of task distributions and standard deviations as the final task offloading results. The baseline methods include random task offloading, proportional task offloading, and greedy task offloading. As shown in Fig. 4.4a, under random offloading and proportional

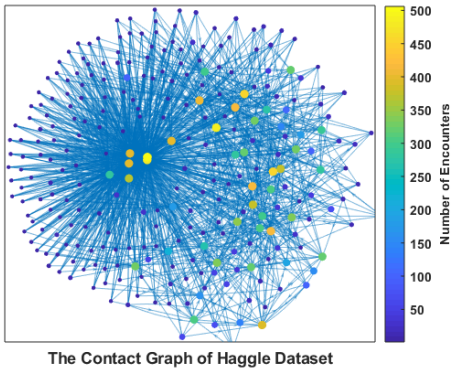


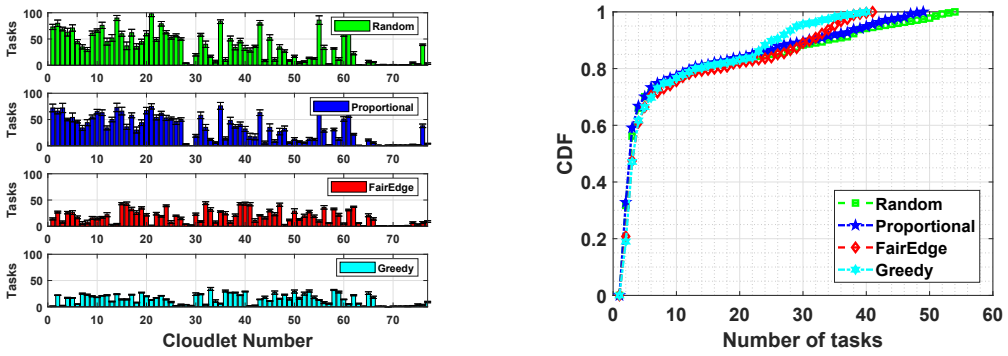
Figure 4.5 : Nodes and encountering records in the Haggle [21] dataset.

offloading, some particularly active mobile cloudlets are extremely overloaded (*e.g.*, offloaded with over 150 and even 200 tasks). While proportional offloading partially reduces the number of overloaded cloudlets, the overall task distribution is still highly imbalanced. In contrast, the proposed FairEdge scheme and greedy offloading scheme show remarkable performance in balancing the task distributions over the entire network, where the task load of each cloudlet is under 50. Also, by combining the '2-choice' paradigm from balls-into-bins theory with Jain's Fairness index, FairEdge further achieves a slightly lower task load on each cloudlet throughout the task offloading process.

The empirical cumulative distribution of task offloading results with the MobiClique dataset is presented in Fig. 4.4b. More than 80% of mobile cloudlets in FairEdge and greedy schemes are offloaded with less than 25 tasks. Meanwhile, more than 20% of mobile cloudlets in random and proportional schemes have more than 30 tasks. The above evaluation results show the effectiveness of the fairness-oriented task offloading scheme in a real-world scenario. FairEdge can effectively achieve balanced task offloading on a real-world mobility trace dataset, where different cloudlets have significant disparities in the active level of mobility.

Haggle dataset

Basic Setups. We further evaluate the performance of FairEdge in a larger mobility trace dataset, *i.e.*, Haggle dataset [21]. The Haggle dataset is under the



(a) Task distribution in Haggle dataset

(b) CDF of task offloading results in Haggle dataset

Figure 4.6 : Comparison of task offloading results in trace-driven evaluations with the Haggle dataset.

project of Koblenz Network Collection (KONECT) [95] for systematic study on diverse networks. In short, the Haggle dataset contains mobility and connectivity traces that were generated from iMote devices. The iMote devices are small portable devices to capture Bluetooth sightings (encounters) of their carriers. We process and visualise the contact graph of the Haggle dataset in Fig. 4.5, where all 274 vertices are with 28,244 edges. Similar to the MobiClique dataset, each vertex in the contact graph represents a mobile cloudlet, an edge between two vertices represents a contact, and the colour of each vertex shows its active level in the network. Different from the MobiClique dataset, the vertices in the Haggle contact graph are more distributed, where a small number of vertices form a 'contact centre' (1-77) that links the rest edge vertices with sparse contacts. In the following evaluations, all basic setups are the same as those for the evaluations with the MobiClique dataset, except for the time interval. In the Haggle dataset, the beginning and ending timestamps are 20,733 seconds and 364,094 seconds, respectively. As the overall duration in the Haggle dataset is much longer than that of MobiClique, we set the length of a time slot to 3,600 seconds (*i.e.*, 1 hour) for computation offloading simulation.

Task Offloading Results. In evaluations with the Haggle dataset, we also conduct task offloading with mobile cloudlets for 100 times. We take the average values of task distributions and standard deviations as the final results. As shown

in Fig. 4.6a, a group of cloudlets with high contact levels take the majority of tasks in the Haggle network. This is due to that the rest of each mobile cloudlet only has several contact opportunities for task offloading, and target cloudlets in these contacts are all in the group of the 'contact centre' (cloudlets 1-77). As the cloudlets of numbers 80-274 have very few encounters with others, most of them are offloaded only a few tasks or even none. To make it clearer for performance comparison, we only provide the offloading results for cloudlets 1-77. The random task offloading shows the worst performance in balancing the task load in the contact centre, as some cloudlets are overloaded with nearly 100 tasks. While proportional task offloading slightly improves the task distribution result, there still exist huge gaps (over 90 tasks) among cloudlets in the contact centre. In contrast, FairEdge and greedy task offloading schemes significantly enhance the balance in task distribution over all cloudlets in the contact centre, where most of the cloudlets are offloaded with less than 50 tasks. Besides, for the cloudlets with low contact levels, FairEdge still preserves their fairness by re-balancing tasks from overloaded cloudlets to others. The empirical cumulative distribution of task offloading results with the Haggle dataset is presented in Fig. 4.6b. This CDF figure reveals that even there are considerable gaps in contact level among different mobile cloudlets, FairEdge can still achieve balanced task offloading with the close performance to the greedy algorithm, showing the effectiveness of preserving fairness in mobile computation offloading.

4.7 Conclusion

In this chapter, we have proposed FairEdge, a Fairness-oriented task offloading scheme to enable balanced task sharing and computation offloading for mobile Edge cloudlet networks. The FairEdge integrates balls-into-bins theory and Jain's fairness index for distributed task offloading among mobile cloudlets. We have developed the system model of computation offloading in edge mobile cloudlet networks and formulated the load balancing problem together with a fairness optimisation problem. By adopting the 'two-choice' paradigm and using calculated fairness index values for cloudlets and the network, we have further proposed the algorithm design of *FairEdge* and conducted extensive evaluation studies with simulations and

experiments on real-world trace datasets. The experimental results have shown that FairEdge successfully achieves load balancing with guaranteed performance, with a near-optimal fairness index of up to 0.85 and an improvement of 50% over conventional baseline methods.

Chapter 5

iMap: Towards System Implementation and Data Analysis for Edge Crowdsensing Based Outdoor RSS Maps

With the explosive usage of mobile edge devices, sustainable access to wireless networks (e.g., WiFi) has become a pervasive demand. Most mobile edge users would expect a seamless network connection at a low cost. Indeed, this can be achieved by using an accurate Received Signal Strength (RSS) map for wireless access points. While existing methods are either costly or unscalable, the emerging mobile edge crowdsensing paradigm is a promising technique for building RSS maps. MEC applications leverage pervasive mobile edge devices to collect data collaboratively. However, the heterogeneity of edge devices and the mobility of users could cause inherent noises and blank spots in the collected dataset. In this chapter, we study (1) how to tame the sensing noises from heterogeneous mobile edge devices, and (2) how to construct accurate and complete RSS maps with random mobility of edge crowdsensing participants. First, we build a mobile edge crowdsensing system called iMap to collect RSS measurements with heterogeneous mobile devices. Second, through observing experimental results, we build statistical models of sensing noises and derive different parameters for each kind of mobile edge device. Third, we present the signal transmission model with a measurement error model, and further propose a novel signal recovery scheme to construct accurate and complete RSS maps. The evaluation results show that the proposed method can achieve 90% and 95% recovery rate in the geographic coordinate system and the polar coordinate system, respectively.

5.1 Introduction

With the proliferation of smart mobile devices, mobile crowdsensing has become a promising paradigm. Mobile edge users can exploit their smartphones to perform large-scale sensing tasks [96] cooperatively. Based on mobile edge crowdsensing, both industry and academia have developed numerous novel applications [97], such as traffic monitoring [98, 99], route planning [100, 101], air quality sensing [102, 103], localisation [104, 105] and digital map construction [106–108].

Nevertheless, the above crowdsensing applications usually require high network bandwidth for data transmission. In terms of cost and efficiency, WiFi networks enable compute-intensive applications to provide more reliable computing services for mobile edge users. For instance, the public WiFi access points (APs) have been pervasively deployed in metropolises, especially in indoor environments (*e.g.*, apartments, shopping centres, airports, *etc.*) [109]. In contrast, the Quality of Service (QoS) of the outdoor WiFi network is difficult to quantify [110].

The major concerns of outdoor access points are signal coverage and transmission capacity [111]. The above information can be obtained from the Received Signal Strength (RSS). However, it is non-trivial to collect complete RSS data in large areas, and many researchers have put their efforts into achieving it. Ayon et al. [112] proposed SpecSense, a platform for large scale spectrum monitoring. Similarly, Wu et al. [113] presented CrowdWiFi, a vehicular crowdsensing system for looking up roadside WiFi networks. In [114], the authors proposed CRAD, a crowdsensing based approach to detect rogue APs. The above works focus more on data collection rather than the accuracy and reliability of the raw crowdsensed RSS data.

However, our experimental results show that the accuracy and reliability of RSS data can be seriously influenced by the mobility pattern of edge users and the heterogeneity of edge devices. Even on the same observing spot, the RSS measurements from different devices can always have mismatches or misalignments. This is mainly due to the differential capabilities of mobile devices in sensing signals. To address this issue, in [115], an Expectation-Maximization based mechanism is

proposed to compute the maximum likelihood estimation of sensor noises. Furthermore, Xiang et al. [116] proposed an iterative algorithm to reduce the error-rate of crowd sensed RSS data. Moreover, Kim et al. [117] presented a mobile crowdsensing framework for a large-scale WiFi fingerprinting system, using physical-layout and signal-strength measurements.

Nevertheless, if we assume that the noises in RSS data can be reduced or even eliminated, it is still difficult to construct a complete RSS map, because the crowd sensed data is usually incomplete, and such data cannot cover every spot on the map. Due to the different trajectories of mobile edge users, there are blank spots without any data in RSS maps. These blank spots are particularly challenging in constructing large-scale RSS maps, as the sensing cost is in proportion to the crowdsensing coverage [118]. Wang et al. [119] discussed the above issues and designed a general framework for sparse mobile crowdsensing applications. In [120], the authors proposed a crowdsensing-based WiFi radio map construction mechanism for mobile users to choose appropriate access points. Wu et al. [121] proposed PRES-M, a privacy-preserving RSS map generation scheme for crowdsensing networks. However, the above works neglect the importance of data quality in building WiFi related digital maps. Therefore, how to construct accurate and complete RSS maps for outdoor APs remains challenging.

Specifically, two challenges need to be formally addressed.

- First, how to build accurate RSS maps with unpredictable noises in the crowd sensed data. In our experiments, the difference of collected RSS data between two smartphones could be up to 40 dBm. Such large noises, either from the sensing errors or malicious measurements, should all be fairly tamed for constructing accurate RSS maps.

- Second, how to construct complete RSS maps with the missing data in blank spots. In practical crowdsensing, it is hard to fully cover the target area considering the random mobility of mobile users and the overall cost. The RSS maps need to be accurately constructed with incomplete sensing data.

In this chapter, we propose iMap, towards system implementation and data analysis for crowdsensing based outdoor RSS maps. Our system enables mobile edge users to use their sensor-embedded smartphones to collaboratively collect RSS data in the wild. By systematically analysing the collected RSS measurements from heterogeneous devices, we have found the following facts.

- First, although the noise is inevitable, for each type of smartphone, the crowd sensed data could fit into a statistical model fairly well. Moreover, the variances between two different types of smartphones could roughly fit into a specific linear model. In addition, we recruit a group of volunteers to collect RSS data with three different types of smartphones. The experimental results validate the above claims. Hence, we can leverage the features of RSS data to estimate data quality.
- Second, even if the crowd sensed RSS data is incomplete, we can still form a sufficiently sparse matrix on it. By leveraging compressive sensing methods, we can sample the sparse matrix to recover the data on unsensed spots adaptively.

Based on the above observations, in this work, we apply a model-based mechanism to reduce errors and noises in RSS data. With more reliable data, we can further apply an adaptive sparse sampling algorithm to recover RSS data and build complete RSS maps. The major contributions of this chapter are listed as follows.

- To the best of our knowledge, iMap is the first edge crowdsensing system for constructing accurate and complete RSS maps in the wild. We develop an application for mobile edge users to collect RSS measurements. Meanwhile, we use a cloud-based central server for RSS data storage and processing.
- We conduct real-world experiments and analyse the RSS data from diverse aspects. Based on the experimental results, we investigate the error models for heterogeneous smartphones.
- We propose a compressive sensing-based RSS data sampling and recovery algorithm. The experimental results show that the proposed algorithm can achieve

90% and 95% recovery rates in the geographic coordinate system and the polar coordinate system, respectively.

The rest of this chapter is organised as follows. We introduce the design and implementation of the iMap system in Section 5.2. Then, in Section 5.3, we present and explore the crowdsensing experiments to collect RSS data. In Section 5.4, we propose the signal propagation model and the measurement error model. We further devise a sparse sampling-based algorithm to recover the blank spots and show the experimental results of signal recovery. Finally, we conclude this work in Section 5.5.

5.2 System Design

The proposed iMap system is designed for building accurate and reliable RSS maps. The main functions include RSS data collection, data processing, and RSS map visualisation. By running the iMap application on smartphones, mobile users can participate in measuring RSS of surrounding wireless access points. In the meantime, iMap will automatically attach location information to collected RSS data. The crowdsensed data is then uploaded to an edge-based server that is responsible for calibrating the noises and generating the visualisation data. With the iMap system, we can leverage the edge crowdsensing paradigm to measure the signal strength of access points in large-scale urban areas. Accordingly, we conduct real-world experiments with iMap in an urban square in Wuxi City, China. The details of the experimental results will be presented in Section 5.3.

5.2.1 Design overview

We build the iMap system on two ends, *i.e.*, the edge user end and the server end. In the user end, we develop a mobile application for users to measure RSS values of surrounding wireless access points. In the server end, we build a laptop-based edge server, where the user's data is organised by its location information. We further store the RSS data into an online database for RSS map visualisation.

The architecture of the *iMap* system is shown in Fig. 5.1, where the user-end mobile application consists of four modules. The content provider is one of the

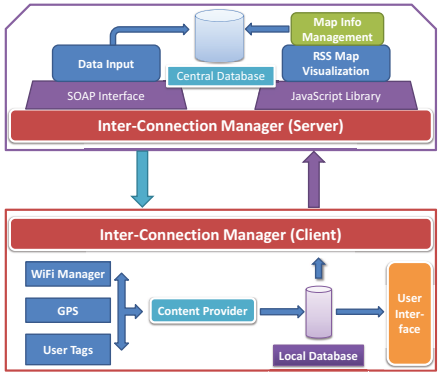


Figure 5.1 : The architecture of the iMap system.

main function modules. It encapsulates the original crowd sensed data, acting as the interface for transmitting data to the local database. To display the collected RSS data observed at the current location, the locally stored data is sent to the user interface module periodically. Similarly, the local RSS data is sent to the connection manager module. The connection manager module handles the communication and data exchange between smartphones and the edge server. We build an online edge server based on LeanCloud [122] and implement the Simple Object Access Protocol (SOAP) in the data input module. The requests from mobile users of accessing the database are processed through the data input module. Furthermore, we emulate a JavaScript-based interface to extract RSS data divided by geographic information. After that, the visualisation module will iteratively attach RSS data to the map module. The map information management module is set up to transfer and store visualisation data into the central database. In the edge server, the RSS data is organised by its geographic coordinate. Besides, the online server provides the participating users with the latest RSS dataset for visualisation on the iMap application. We describe the details of system architectures in the following subsections.

5.2.2 Smartphone as a client: the real-time measurement

In the iMap application (as shown in Fig. 5.2), we build a real-time RSS data processing module. As the Android operating system has provided specific classes in signal sensing, we mainly use four important classes (*i.e.*, WifiManager, ScanResult,

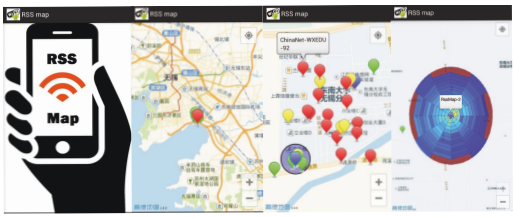


Figure 5.2 : The overview of the iMap mobile application.

WifiConfiguration, and WifiInfo) in the RSS data processing module. The WifiManager class provides a variety of APIs for WiFi management, such as WiFi scanning, establishing network connection and configuration options. We instantiate the WifiManager class by simply invoking Context.getSystemService(Context.WIFI_SERVICE). We further call its public method *getScanResults* to return a table list of access points in the last scan. From this table list, we can acquire complete information on surrounding access points, including SSID, MAC address, RSS values, capabilities, and frequency. Considering the generality, we run the scanning module for 5 times at each sensing spot and take the average value of crowdsensed RSS data.

5.2.3 Communication to the edge server: geographic data processing

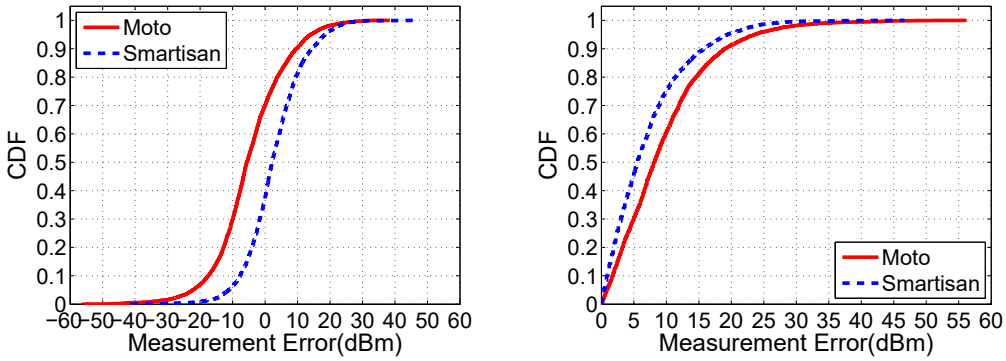
As it is non-trial to build RSS Maps, the iMap application will use the LocationManager class to access the system's location services during the signal sensing. This allows iMap to obtain real-time updates of each device's geographic location. We leverage the location information provided by either GPS (Global Positioning System) or cellular network to localise each mobile user and tag coordinate information to the RSS data. The iMap application will periodically upload the latest RSS data to the edge server through the inter-connection manager. We use JSON as the data transmission format in iMap and separate uploaded data in a central database by geographic coordinates. More importantly, the coordinate is also the key unit for RSS visualisation. Therefore, mobile users can send RSS data from the application to the edge-based server. Once a user opens the application, the iMap will send update requests to the edge server and download the latest RSS dataset at the current location.

5.2.4 Edge server: RSS map visualisation

The edge server is built on a laptop using Lean Cloud, and it is responsible for data aggregation and RSS map visualisation. Considering efficiency and accuracy, we leverage a commercial map platform called 'Amap' to visualise RSS maps on mobile edge devices. With Amap's location SDK (software development kit) and API (application programming interface), we apply the *getLongitude* method and *getLatitude* method to acquire the geographic location of each smartphone. When users are moving, the *requestLocationData* method is invoked to capture the real-time longitude and latitude data. In the practical setting, we program the iMap application to request for updated geographic coordinates when the change of location exceeds 5 meters. Meanwhile, iMap will re-scan wireless access points once the geographic coordinates are updated. We visualise WiFi access points on Amap by two steps. First, we use the *marker* class in the Amap SDK to mark the individual access point on the map. The different colours represent the different levels of RSS (high, medium, or low). Second, we apply the InfoWindow method to add information windows on the access points. When a user clicks the marker, the information window will pop up and show detailed information about the corresponding access point.

5.2.5 Incentive Mechanism: data access control

To motivate mobile edge users to participate in the RSS map crowdsensing, we further design an incentive mechanism with data access control. By the first-time use of the iMap application, a mobile user can only access the RSS data within the district he is localised. When a user uploads a new piece of data from a different district, the corresponding RSS data of that district will be released to the user. Once the RSS data is unlocked, the iMap application will send requests to the edge server and download the new data. The above data access control flow is automatically executed in the iMap application.



(a) Deviation (b) Absolute deviation

Figure 5.3 : Measurement deviations of different smartphones.

5.3 Experimental Study and Observation

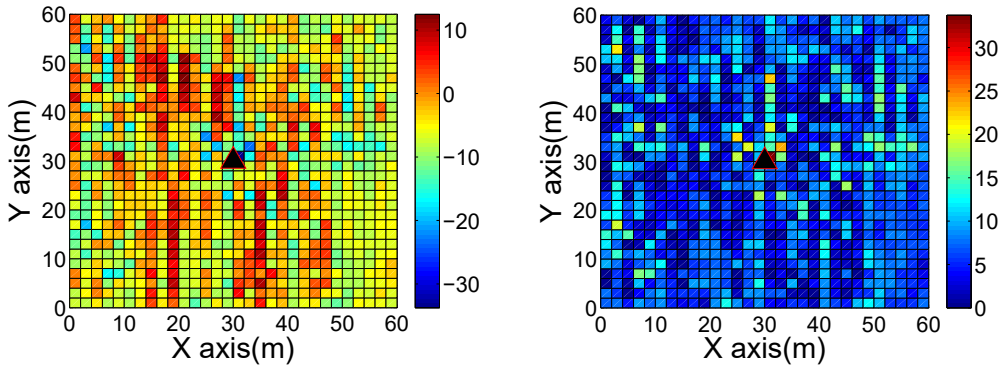
By leveraging the iMap system, we first conduct an experimental study with 18 volunteers from the university and the parameter settings are as follows. The volunteers are divided into 3 groups using three different types of smartphones, *e.g.*, Samsung, Motorola, and Smartisan. Each participant takes a random walk in a 5,000 square-meter urban area for 2 sessions to collect RSS data. In the first session, participants in each group together take random walks for 30 minutes. In the second session, each individual participant walks randomly for another 30 minutes. In both sessions, the iMap application is running on each volunteer's smartphone. At last, each group will upload the crowdsensed data to the edge server. We make the following observations and analysis of the collected data.

5.3.1 Diversity of RSS measurements

First, we explore the diversity of RSS measurements with deviation and spatial deviation.

Deviations of RSS measurements

Specifically, we compare the deviation of measurements collected by different smartphones in Fig. 5.3. Here, we use the measurements of Samsung smartphones as the benchmark. In Fig. 5.3a, the measurement deviations of both Moto and



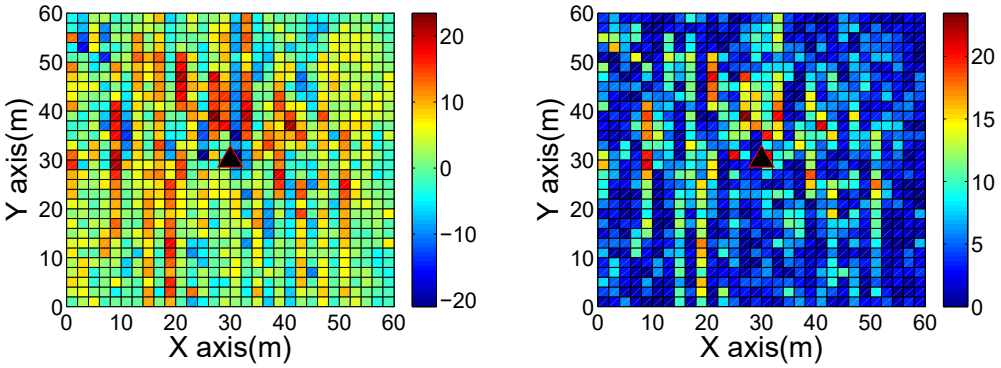
(a) Deviation (b) Absolute deviation

Figure 5.4 : Spatial distribution for the measurement deviations between Samsung smartphone and Moto smartphone.

Smartisan have positive values and negative values, which indicates that the noises caused by heterogeneity are fluctuant. As illustrated in Fig. 5.3b, about 90% of the absolute deviations between Smartisan and Samsung are less than 15 dBm. Meanwhile, from Smartisan to Samsung, the absolute deviations are within 15 dBm for 80% measurements. If we narrow down the deviation range, still about 60% of absolute deviations are less than 10 dBm for Moto's measurements, and 50% of absolute deviations are less than 5 dBm for Smartisan's measurements. Note that, the maximum deviations can be up to 50 dBm for both Moto and Smartisan. The above results demonstrate that the deviations between the measurements of different devices are significant and can cause inaccuracy of the RSS map. Hence, sensing noises among heterogeneous devices need to be carefully addressed.

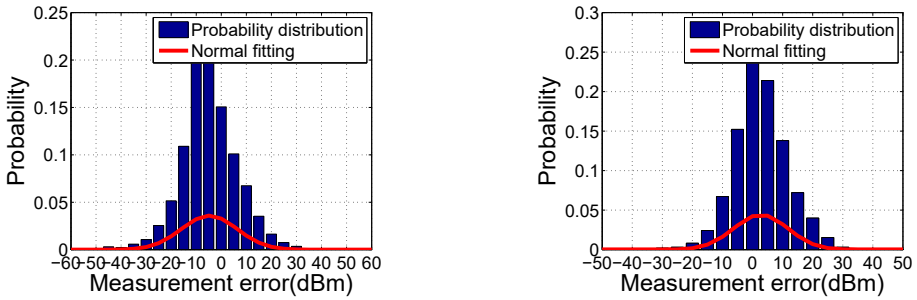
Spatial deviations of RSS measurements

We further explore the spatial distribution of measurement deviations. In this experiment, we still use the measurements of Samsung as the benchmark. Fig. 5.4 shows the measurement deviations and absolute measurement deviations between Moto and Samsung in spatial distribution. The deviations are randomly distributed, where the large deviations fall into the area between 15 to 40 meters on X-axis and 10 to 50 meters on Y-axis. Most of the absolute deviations are smaller than 10 dBm, and only a few exceed 20. Similarly, Fig. 5.5 shows the spatial distribution



(a) Deviation (b) Absolute deviation

Figure 5.5 : Spatial distribution for the measurement deviations between Samsung smartphone and Smartisan smartphone.



(a) Samsung vs. Moto (b) Samsung vs. Smartisan

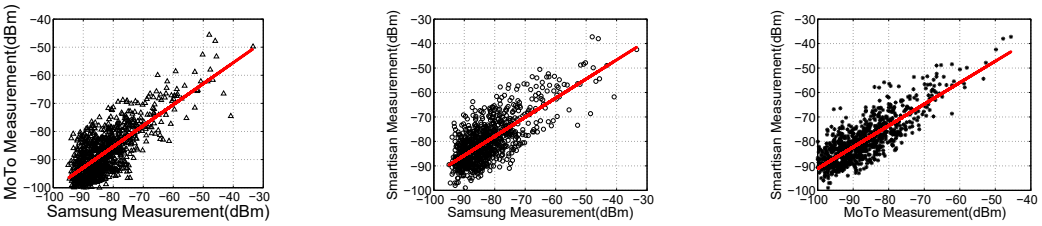
Figure 5.6 : The probability distribution of the measurement deviations of different devices.

of deviations between Smartisan and Samsung. With more negative deviations, the large deviations fall into the area between 20 to 50 meters on both X- and Y-axes. In Fig. 5.5b, the absolute deviations are sparse on the map, showing that there exist noises in RSS data caused by the heterogeneity of devices.

5.3.2 Exploring models of measurement deviations

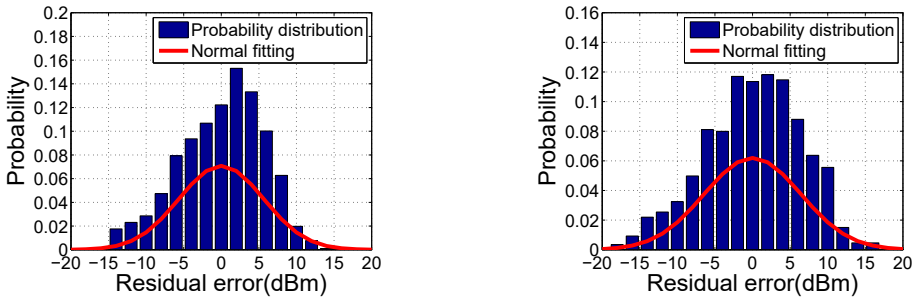
Next, we explore the error models of measurement deviations from RSS measurements among Samsung, Moto, and Smartisan.

Firstly, we explore whether the measurement deviations of different devices satisfy the normal model. As shown in Fig. 5.6, we compare the probability distribution



(a) Samsung vs. Moto (b) Samsung vs. Smartisan (c) Moto vs. Smartisan

Figure 5.7 : The linear relationship between the measurements of different devices.



(a) Samsung vs. Moto (b) Samsung vs. Smartisan

Figure 5.8 : The probability distribution of the residual errors after linear fitting.

of measurement errors with normal fittings. By using the Lilliefors test, we find that the statistical significance is only 1%, showing the rejection of normal model assumptions.

Secondly, we explore whether the measurement deviations satisfy the linear model. As shown in Fig. 5.7, the measurements of different devices follow linear models. Specifically, we calculate the linear fittings as follows. For Samsung vs. Moto, the fitting model is $0.74737x \pm 25.6226$, with a standard deviation of 5.63 dBm. For Samsung vs. Smartisan, the fitting model is $0.78078x \pm 15.4472$, with a standard deviation of 6.43 dBm. For Moto vs. Smartisan, the fitting model is $0.87764x \pm 3.331$ with a standard deviation of 5.37 dBm.

Based on the above observations, we further explore whether the residual errors of linear fitting follow the norm model. As illustrated in Fig. 5.8, we use the Lilliefors method to test this assumption. However, the statistical significance is still 1%. As a result, the residual errors do not follow the normal model.

5.3.3 Challenges

Based on the experimental observations, we find that with both the signal propagation model and measurement error model, it is still non-trivial to achieve accurate RSS map construction. Adding to the blank spots without any RSS data, two challenges need to be formally addressed as follows.

The first challenge is to model the signal propagation and measurement error. The measurement error model is essential for calibrating the noises in RSS data of different devices. However, the model parameters are not known as a prior, and the values of parameters usually depend on the type of mobile devices.

The second challenge is to recover signal strength data with incomplete measurements. Although compressed sensing can be used for data recovery, how to design a measurement matrix and a recovery algorithm remains challenging. Our goal is to extract salient information from the k-sparse or compressible signals, without compromising signal accuracy by the dimensionality reduction.

5.4 Sparse Signal Recovery Design

In this section, we present how to construct accurate and complete RSS maps with sparse sampling and signal recovery. In practice, it is not feasible for mobile edge users to collect fully complete RSS sensing data that covers every spot in a large district, due to the high cost and low efficiency. In addition, it is also a daunting task to directly construct a complete RSS map at high accuracy only with partial RSS data. Meanwhile, compressive sensing techniques [123,124] prove to be capable of recovering sparse signals with limited information. Hence, in this section, we leverage the compressive sensing technique to recover the RSS maps with partially sampled data.

5.4.1 Preliminaries in Compressive Sensing

Compressive sensing is an innovative signal sampling paradigm that related to several topics in signal processing [125], including sparse sampling, heavy hitters, and *etc.* Compressive sensing theory asserts that a relatively small number linear

combination of a compressible and sparse signal can contain most of its salient information [124].

Compressibility of Signals

By following [124], we consider a WiFi signal as an one-dimensional, real-valued and finite-length signal $\mathbf{x}$, which can be viewed as an $N \times 1$ column vector in $\mathbb{R}^N$. Any signal in $\mathbb{R}^N$ can be represented in the form of an orthogonal basis of $N \times 1$ vectors $\{\psi_i\}_{i=1}^N$. Using an $N \times N$ basis matrix $\Psi = [\psi_1|\psi_2...|\psi_N]$, where the vectors $\{\psi_i\}$ are as columns. The signal $\mathbf{x}$ can be expressed as:

$$\mathbf{x} = \Psi\mathbf{s} = \sum_{i=1}^{N} s_i\psi_i, \tag{5.1}$$

where $\mathbf{s}$ is the $N \times 1$ column vector of weighting coefficients s_i. As shown above, the s_i can be calculated by $s_i = \langle \mathbf{x}, \psi_i \rangle = \psi_i^T\mathbf{x}$, where $\cdot^T$ is the transposition operation. Note that $\mathbf{x}$ and $\mathbf{s}$ are the equivalent representations of signal, with $\mathbf{x}$ in the temporal or the spatial domain while $\mathbf{s}$ in the Ψ's domain.

We identify the signal $\mathbf{x}$ as K-sparse, if it is a linear combination of only K basis vectors. That is to say, only K components (i.e., K s_i) of $\mathbf{s}$ in Equation 5.1 are nonzero and the remaining $(N - K)$ components are zero. The signal $\mathbf{x}$ has *compressibility* if the representation in Equation 5.1 has a few large coefficients and many small coefficients [124].

The Problem of Recovering RSS Signals

Different from traditional methods, compressive sensing [126] directly acquires a compressed signal representation without the middle stage of acquiring N samples. Considering RSS sensing as a general linear measuring process, which computes M inner products between $\mathbf{x}$ and a set of vectors $\{\phi_j\}_{j=1}^M$ as in $y_j = \langle \mathbf{x}, \phi_j \rangle$, for $M < N$. We set the measurement y_j in an $M \times 1$ vector $\mathbf{y}$ and the measurement vector ϕ_j^T as rows in an $M \times N$ matrix Φ. Then, by substituting Φ from Equation 5.1, $\mathbf{y}$ can be calculated by

$$\mathbf{y} = \Phi\mathbf{x} = \Phi\Psi\mathbf{s} = \Theta\mathbf{s}, \tag{5.2}$$

where $\Theta = \Phi\Psi$ is an $M \times N$ matrix.

Using compressive sensing to recover RSS signals induces two subproblems: (i) the design of a stable measurement matrix Φ so that the salient information from the compressible K-sparse signal is not damaged by the dimensionality reduction from $\mathbf{x} \in \mathbb{R}^N$ to $\mathbf{y} \in \mathbb{R}^M$. and (ii) the design of an efficient signal reconstruction algorithm to recover $\mathbf{x}$ from $\mathbf{y}$.

Design of Measurement Matrix

The measurement matrix Φ must allow the reconstruction of the length-N signal $\mathbf{x}$ from the length-M signal $\mathbf{y}$, for $M < N$. If $\mathbf{x}$ is K-sparse and the K locations of the nonzero coefficients in $\mathbf{s}$ are known, the problem can be solved provided that $M > K$. The necessary and sufficient condition for the above simplified problem is that *for any vector* $\mathbf{v}$ *sharing the same* K *nonzero entries as* $\mathbf{s}$ *and for some* $\varepsilon > 0$ [124]:

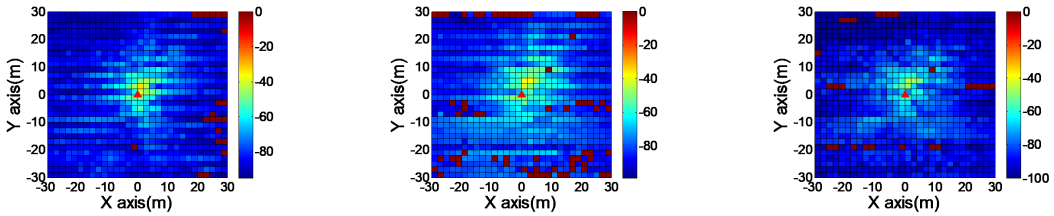
$$1 - \varepsilon \leq \frac{\|\Theta\mathbf{v}\|_2}{\|\mathbf{v}\|_2} \leq 1 + \varepsilon. \tag{5.3}$$

In such a case, the matrix Θ preserves the lengths of the particular K-sparse vectors.

However, the locations of the K nonzero entries in $\mathbf{s}$ are generally unknown as a prior. Note that a sufficient condition for a stable solution to both K-sparse and compressible signals is that the Θ satisfies Equation (5.3) for an arbitrary $3K$-sparse vector $\mathbf{v}$, where $3K < N$. This loose condition is usually referred to as the *restricted isometry property* (RIP) [127]. Existing studies [126,128] have shown that the RIP condition can be achieved by selecting Φ as a random Gaussian measurement matrix [124].

Design of the Signal Recovery Algorithm

Generally, a signal recovery algorithm should take M measurements from vector $\mathbf{y}$ and reconstruct the length-N signal $\mathbf{x}$ (or the sparse coefficient vector $\mathbf{s}$). In particular, if Φ satisfies the condition of RIP, $\mathbf{x}$ and $\mathbf{s}$ can be successfully recovered using only $M \geq cK\log(N/K)$ Gaussian measurements [124], where c is a small constant.



(a) Samsung (b) Smartisan (c) Moto

Figure 5.9 : Geographic RSS map constructed with sensing data from different types of smartphones.

Since our main objective is to leverage compressive sensing for application-oriented RSS data recovery, the details are omitted and can be referred to [124].

According to [124], the signal reconstruction algorithm based on the ℓ_1 norm can exactly recover K-sparse signals and closely approximate signals of compressibility by

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \|\mathbf{s}\|_1, \text{s.t.}, \Phi\mathbf{s} = \mathbf{y}. \tag{5.4}$$

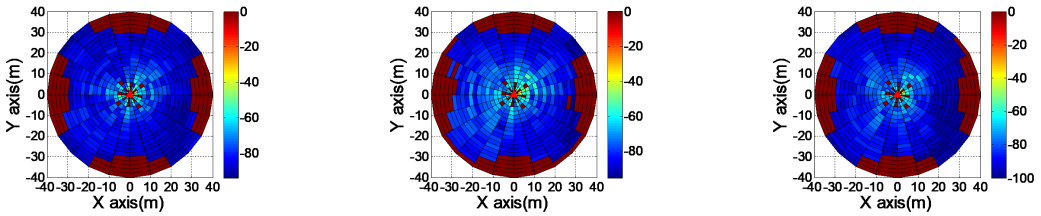
In this study, when the measurement vector $\mathbf{y}$ contains noise, the signal can still be reconstructed via ℓ_1 and ℓ_2 norms by

$$\hat{\mathbf{s}} = \underset{\mathbf{s}}{\operatorname{argmin}} \|\mathbf{s}\|_1, \text{s.t.}, \|\Phi\mathbf{s} - \mathbf{y}\|_2^2 \leq \varepsilon, \tag{5.5}$$

where ε is the bound of the noise.

5.4.2 RSS map construction with partial RSS data

In this work, we construct the RSS map with partial data collected from different types of smartphones, including Samsung, Smartisan, and Moto. In Fig. 5.9, we construct RSS maps in the geographic coordinate system. The red spots on the maps are unsensed, *i.e.*, with no available data. We find that the constructed maps with data from Samsung smartphones are with the least unsensed spots, which shows the best performance in map construction. Meanwhile, the RSS maps constructed with data from Smartisan are with the most unsensed spots. In terms of coverage, Samsung and Moto achieve similar performance on coverage of RSS data in the



(a) Samsung (b) Smartisan (c) Moto

Figure 5.10 : Polar RSS maps constructed with sensing data from different types of smartphones.

maps, while Smartisans collect sparser RSS data. The above observations show that there are significant data variances caused by heterogeneous mobile devices.

We further construct RSS maps in the polar coordinate system, as shown in Fig. 5.10. In polar RSS maps, the unsensed spots are represented by red quadrilaterals. Clearly, there are more unsensed spots at the edge of polar RSS maps. Similar to the construction results in geographic maps, the Smartisan's measurements are sparser and more inaccurate in comparison with Samsung's and Moto's. In the following, to recover data at the unsensed spots on RSS maps, we devise an adaptive algorithm for RSS data sampling and recovery.

5.4.3 Adaptive Data Sampling and Recovery Algorithm for RSS Map Reconstruction

To reconstruct the outdoor RSS map, we set a WiFi access point to cover an outdoor area, which is further divided into a set of block-based subregions. With a total number of $N \times N$ blocks, we denote the RSS sensing matrix as $T_{N\times N}$ and use each element T_{xy} to represent the averaged RSS data measurement at block (x, y). We further adapt the Singular Value Thresholding (SVT) from [129] to recover the RSS matrix with sequential and adaptive sampling. The key challenge here is to select the most effective samples for data recovery as well as to avoid redundant samples. In the following, we address this challenge step by step.

During the data recovery process, we denote $S(t)$ and $S(t+1)$ to be the corresponding sets of samples taken at time interval t and $t+1$, respectively. The

recovered RSS data matrices are further denoted by $\widehat{T}(t)$ and $\widehat{T}(t+1)$, respectively. In addition, the recovered data entries of (x, y)-th element in $\widehat{T}$ are denoted by $\widehat{T}_{xy}(t)$ at t and $\widehat{T}_{xy}(t+1)$ at $t+1$, respectively.

If $\widehat{T}_{xy}(t)$ and $\widehat{T}_{xy}(t+1)$ are close to each other and $T_{xy} \notin S(t+1)$, then, the element T_{xy} is considered to have been recovered by the algorithm in time $t+1$. Otherwise, with $\widehat{T}_{xy}(t)$ and $\widehat{T}_{xy}(t+1)$ being far from each other, the element T_{xy} in the RSS sensing matrix is considered not have been recovered accurately, and in this case, we could gain more information by taking an extra sample at (x, y)-th entry of T for further recovery.

Information-based metric. When taking new samples into the sample set, we need to quantify and evaluate the informativeness of each element in the RSS data matrix. Therefore, we define the *information-based metric* for element T_{xy} as $I_{(x,y)}$ [130] and calculate it as follows:

$$I_{(x,y)} = \frac{\left|\widehat{T}_{xy}(t+1) - \widehat{T}_{xy}(t)\right|}{\frac{1}{2}\left|\widehat{T}_{xy}(t+1) + \widehat{T}_{xy}(t)\right|}. \tag{5.6}$$

If an element T_{xy} has a large value of $I_{(x,y)}$, it is considered to be much informative and thereby should be taken into the sample set in the next time interval. To this end, the key to our adaptive sampling strategy is to iteratively take samples from the most-informative entries in future steps for data recovery. Next, we introduce the stopping condition for the sampling iteration.

Sampling Stopping Condition. To make a trade-off between the completeness of the RSS matrix and the computation cost, we define the *Sampling Stopping Condition* as follows.

Definition 1. ***Sampling Stopping Condition.*** *Given two recovered matrices* $\widehat{T}_{N\times N}(t)$ *and* $\widehat{T}_{N\times N}(t+1)$*, we define* $\widehat{T}(t) \triangleq \widehat{T}(t+1)$ *as the sampling stopping*

condition, if the above two matrices satisfy

$$\frac{\sqrt{\sum \left(\widehat{T}_{xy}(t+1) - \widehat{T}_{xy}(t)\right)^2}}{\sqrt{\sum \left(\frac{1}{2}((\widehat{T}_{xy}(t+1) + (\widehat{T}_{xy}(t))\right)^2}} \leq \varepsilon, \tag{5.7}$$

where ε is a small constant.

Algorithm Design. We present the data sampling and recovery algorithm for the RSS matrix in Algorithm 3. In line 2, the algorithm conducts uniform sampling to obtain an initial measurement matrix. The current sample set is denoted by Ω and the size of the current sample set is $|\Omega| = \eta \times N \times N$. The value of $N \times N$ equals the total number of RSS sensing blocks and $\eta \in [0, 1]$ is a parameter that determines the fraction of initial uniform samples to the raw RSS sensing matrix. Note that if η is too small, the algorithm will require excessive samples in further steps, and if η it too large, the initial sample set will be redundant and thereby wasting the computation resources.

With the initialised uniform sampling set, the algorithm first recovers the RSS sensing matrix $\widehat{T}(t)$ at the current time t (line 5). Then, from lines 6-7, an extra sample set Ω' will be taken and integrated to Ω as $\Omega = \Omega \cup \Omega'$. Here, the Ω at time t and $\Omega \cup \Omega'$ at time $t + 1$ are corresponding to the previously mentioned sample sets $S(t)$ and $S(t + 1)$, respectively.

By applying the SVT method [129] again, the algorithm obtains a newly recovered matrix $\widehat{T}(t + 1)$ at time $t + 1$ (line 8). To this end, if the *Sampling Stopping Condition* (*i.e.*, $\widehat{T}(t+1) \triangleq \widehat{T}(t)$) is satisfied (lines 10-12), the data recovery process will stop. Otherwise, the algorithm will identify the most informative elements and select them as the new samples in the next time interval for iterative RSS data matrix recovery.

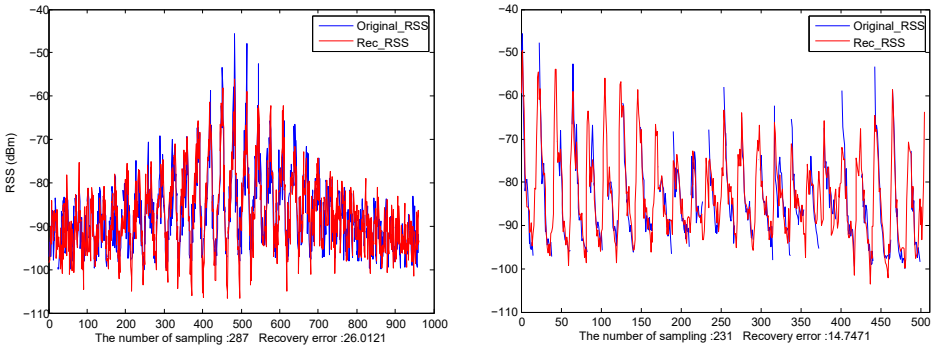
Particularly, in selecting the new extra sampling set, the algorithm will first sort the top $\alpha N \log N$ elements as suggested by [130], where $\alpha = \frac{\sum_{xy} \theta_{xy}}{N \times N}$ and θ_{xy} can be derived from Equation 5.8. Moreover, the parameter $\alpha \in [0, 1]$ and it restricts the number of elements that will be taken as new samples in the next time step. If the

Algorithm 3 Data Sampling and Recovery Algorithm for RSS Data Matrix

Initialise t=0.
Initialise the measurement matrix $T(t)$ with uniform sampling.
Determine the initial RSS sample set Ω with its size as $|\Omega| = \eta \times N \times N$, where $\eta \in [0, 1]$ and $N \times N$ is the size of the RSS sensing matrix.
for $t = 1, ..., n$ **do**
  Apply matrix completion and obtain partially recovered matrix $\widehat{T}(t)$;
  Initialise an extra uniform sample set Ω', by setting the size of Ω' as $|\Omega'| = 0.5N \log N$;
  Integrate Ω with Ω', *i.e.*, $\Omega = \Omega \cup \Omega'$;
  Acquire the new measurement matrix $T(t+1)$ for time interval $t+1$;
  Apply SVT [129] for RSS sensing matrix completion and obtain $\widehat{T}(t+1)$;
    if $\widehat{T}(t+1) \stackrel{\Delta}{=} \widehat{T}(t)$ **then**
      Sampling Stopping Condition is satisfied, stop sampling.
      Return $\widehat{T}(t+1)$ as the recovery RSS sensing matrix.
    else
      for $T_{xy} \notin \Omega$, **do**
        calculate $I_{(x,y)}$;
      end for
    end if
  Sort $I_{(x,y)}$ with the descending order, and then select the first $\alpha N \log N$ elements into Ω', where α can be calculated as $\alpha = \frac{\sum_{xy} \theta_{xy}}{N \times N}$ and θ_{xy} is a binary variable (*e.g.*, 0-1 variable) that can be determined by:

$$\theta_{xy} = \begin{cases} 1, & \frac{\left|\widehat{T}_{xy}(t+1) - \widehat{T}_{xy}(t)\right|}{\frac{1}{2}\left|\widehat{T}_{xy}(t+1) + \widehat{T}_{xy}(t)\right|} > \mu, (i,j) \notin \Omega \\ 0, & \text{otherwise} \end{cases} \tag{5.8}$$

  where μ is a small constant.
  t=t+1;
end for



(a) Sparse sampling and recovery in the geographic coordinate system

(b) Sparse sampling and recovery in the polar coordinate system

Figure 5.11 : One-dimensional sparse sampling and signal recovery.

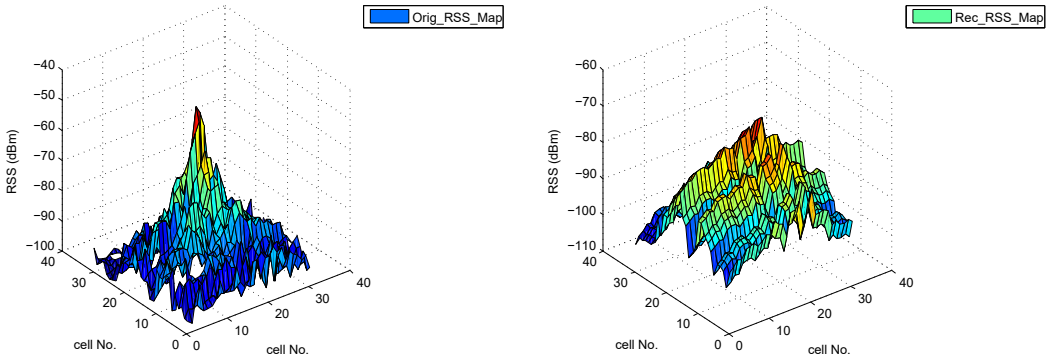
RSS sensing matrix is far from being accurately recovered, α should be larger so that more RSS samples can be integrated into Ω. Otherwise, α should be smaller to make the algorithm to converge.

5.4.4 RSS data sampling and recovery results

We apply Algorithm 3 on crowdsensed RSS data for reconstructing RSS maps in both the geographic coordinate system and the polar coordinate system. In the recovery simulation, according to [130], we set η of $|\Omega|$ to be 12.5%, ε in Equation 5.7 to be 0.03, and μ in Equation 5.8 to be 0.05. The measurement matrix $T_{N\times N}$ contains the average values of RSS sensing data from different types of smartphones over all block subregions. In this experiment, the total number of measurement samples is more than 18,000, and the experimental field is divided into 900 sensing blocks.

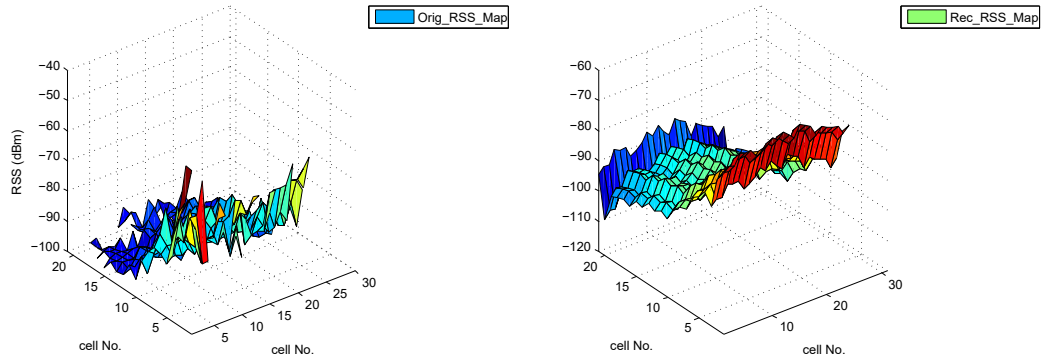
One-dimensional recovery results. We first present the result of one-dimensional sparse sampling and signal recovery on RSS sensing data in Fig. 5.11. The original RSS data is represented by the blue curve, and the recovered RSS data is red. From Fig. 5.11a to Fig. 5.11b, the numbers of sampled data elements are 287 and 231, respectively. Meanwhile, the average recovery error in the geographic coordinate system is 26.0121dBm, which is relatively higher than that of the polar coordinate system of 14.7471dBm. The one-dimensional recovery results show that under the polar coordinate system, the RSS map could be reconstructed with higher accuracy.

Two-dimensional recovery results. We further visualize the recovered RSS



(a) Original signal (b) Signal Recovery after sparse sampling

Figure 5.12 : Two-dimensional sparse sampling and signal recovery in the geographic coordinate system.



(a) Original signal (b) Signal Recovery after sparse sampling

Figure 5.13 : Two-dimensional sparse sampling and signal recovery in the polar coordinate system.

sensing matrix in two-dimensional space. In this experiment, the proposed algorithm will take up to 240 samples. Fig. 5.12 shows the original RSS signals and the recovered RSS signals in the geographic coordinate system, where the original signal is incomplete with missing data at different blocks. After recovery by the proposed algorithm, in Fig. 5.12b, the signal diffusion is smoother, and all the subregions have accurate RSS values. Fig. 5.13 presents the original RSS signals and the recovery results in the polar coordinate system. Similarly, the total number of available measurements is 240. With original RSS data being sparser and sharp in the polar coordinate system, the proposed algorithm successfully recovers a complete RSS signal with high accuracy of 95%.

The above experimental results show that the proposed algorithm can accurately recover RSS signals with sparsely crowdsensed data samples. The recovered results can be further used for constructing accurate edge sensing-based outdoor RSS maps.

5.5 Conclusion

In this work, we have explored to construct RSS maps with edge sensing data collected by heterogeneous crowdsensing devices. We have developed an innovative iMap system that consists of the mobile application for crowdsensing devices and the edge server for RSS data aggregation. Moreover, we have tested the system with different types of smartphones and conduct comprehensive model-based observations. To construct accurate and complete RSS maps, we have further devised a compressive sensing-based recovery algorithm to recover RSS maps with adaptive sampling. The experimental results have demonstrated that the proposed algorithm can achieve accurate and reliable recovery for crowdsensed RSS data. The recovery rates are 90% and 95% in the geographic coordinate system and the polar coordinate system, respectively.

Chapter 6

BuildSenSys: Reusing Building Sensing Data for Traffic Prediction with Cross-domain Learning

With the rapid development of smart cities, smart buildings are generating a massive amount of building sensing data by the equipped sensors. Indeed, building sensing data provides a promising way to enrich a series of data-demanding and cost-expensive urban mobile applications. In this chapter, as a preliminary exploration, we study how to reuse building sensing data to predict traffic volume on nearby roads. In comparison with existing studies, reusing building sensing data has considerable merits of cost-efficiency and high reliability. Nevertheless, it is non-trivial to achieve accurate prediction with such cross-domain data with two major challenges. First, relationships between building sensing data and traffic data are not known as prior, and the spatio-temporal complexities impose more difficulties to uncover the underlying reasons behind the above relationships. Second, it is even more daunting to accurately predict traffic volume with dynamic building-traffic correlations that are cross-domain, non-linear, and time-varying. To address the above challenges, we design and implement *BuildSenSys*, a first-of-its-kind system for nearby traffic volume prediction by reusing building sensing data. Our work consists of two parts, *i.e.*, *Correlation Analysis* and *Cross-domain Learning*. First, we conduct a comprehensive building-traffic analysis based on multi-source datasets, disclosing *how* and *why* building sensing data is correlated with nearby traffic volume. Second, we propose a novel recurrent neural network for traffic volume prediction based on cross-domain learning with two attention mechanisms. Specifically, a cross-domain attention mechanism captures the building-traffic correlations and adaptively extracts the most relevant building sensing data at each predicting step. Then, a temporal attention mechanism is employed to model the temporal dependencies of data across historical time intervals. The extensive experimental studies

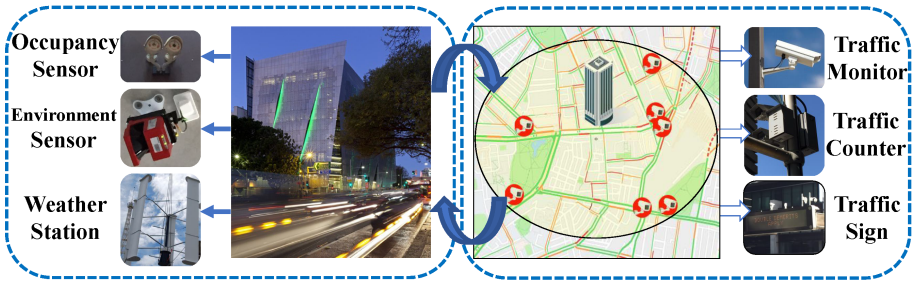


Figure 6.1 : Illustrations for reusing building sensing data to predict nearby traffic volume with cross-domain learning.

demonstrate that *BuildSenSys* outperforms all baseline methods with up to 65.3% accuracy improvement (*e.g.*, 2.2% MAPE) in predicting nearby traffic volume. We believe that this work can open a new gate of reusing building sensing data for urban traffic sensing, thus establishing connections between smart buildings and intelligent transportation.

6.1 Introduction

Smart buildings that equipped with an increasing number of IoT sensors are rising rapidly, producing a large amount of building sensing data (also called *building data** [131–133]). According to Statista's reports in [134], the explosive volume of sensing data collected by global smart buildings was nearly 7.8 ZB (about 7.8×2^{40}G) in 2015, and it is expected to be growing up to nearly 37.2 ZB by 2020. For instance, as shown in Fig. 6.1a, a CBD building in Sydney is installed with more than 2,000 sensors, generating over 100 million sensor readings in monitoring the status of the building, including building occupancy, indoor/outdoor environment, *etc.*

Reusing existing building sensing data is with great significance for ubiquitous sensing of smart cities, as building data is cost-efficiency and sustainable [135]. Assuming building sensing data is available for the long-term, it has recently enabled a series of new applications in urban sensing, such as developing urban 3D mobility

*In the remainder of this chapter, we will use the terms *building sensing data* and *building data* interchangeably unless otherwise stated.

models [136] and identifying abnormal appliances [137, 138]. In this chapter, as a preliminary exploration (illustrated in Fig. 6.1), we are motivated to reuse building sensing data for nearby traffic sensing for the following reasons. Intuitively, most urban buildings are connected by roads, while residents move among different buildings, mainly via commuting on these roads [139]. According to the report in [140], the American citizens averagely spend over 93% of their daily time in enclosed buildings and vehicles (accounting for 87% and 6%, respectively). Thus, there could exist some underlying relationships between sensing data of buildings and traffic data of buildings' surrounding roads, named as *building-traffic correlations* [138]. Such relationships reveal a promising direction of reusing building sensing data, *i.e.*, for predicting traffic of nearby roads. As a proof-of-concept, this work mainly focuses on predicting *traffic volume, which is defined as the number of vehicles traversing on a road segment per hour* [141]. Accurate predictions on traffic volume are fundamentally crucial for Intelligent Transportation Systems (ITSs) applications, such as traffic light control, road navigation, and estimation of vehicle emission [138, 142, 143].

In comparison with the existing methods [141, 144–148] of predicting nearby traffic volume, reusing building data has considerable merits of both *low cost* and *high reliability*, which are crucial to traffic sensing [143]. Most of the conventional traffic prediction methods heavily rely on the fixed road-based traffic sensing systems, such as loop detectors [144] and traffic surveillance cameras [145]. Although these methods can provide accurate traffic volume information, they incur extremely high costs on installation and maintenance, impeding their extensions to large-scale cities [138]. On the contrary, re-using off-the-shelf building sensing data will significantly reduce the cost, as with no extra deployment and maintenance [138, 141]. In terms of opportunistic sensing data of floating vehicles, GPS trajectories [141] and cellular records of passengers [148] have been recently exploited to infer traffic volume to address the high-cost issue. However, opportunistic sensing suffers from the uncontrollable property of users and the data deficiency of particular roads, resulting in unreliable performance in traffic volume prediction [148]. Buildings and

their surrounding roads, as two basic constructions in urban infrastructure, are both stationary with long-term spatial relations. Therefore, in comparison to the opportunistic sensing [141], buildings can generate more reliable and sustainable sensing data, providing better opportunities for accurately predicting nearby traffic volume for long term [147, 149].

Nevertheless, it is necessary to formally address two principal challenges in achieving reusing building sensing data for predicting nearby traffic volume.

- **Challenge 1**: *Investigating the unknown building-traffic relationships.* While a few works [136, 138] have shown certain evidence of building-traffic correlations, they have failed to investigate what exactly these relationships are. To precisely reveal building-traffic correlations is difficult with great complexity. Furthermore, it is extremely challenging to uncover the underlying reasons behind these correlations.

- **Challenge 2**: *Accurate prediction with cross-domain building-traffic correlations.* The building-traffic relationships are cross-domain and non-linear, so that accurate prediction of traffic volume is non-trivial. Even more challenging, these correlations would vary dynamically along with time, imposing more difficulties on accurate traffic prediction [150].

To address these two challenges, we design and implement ***BuildSenSys*** [4], a first-of-its-kind ***Build***ing ***Sen***sing data-based ***Sys***tem for nearby traffic volume prediction. First, for Challenge 1, we conduct extensive experiments based on real-world multi-source datasets (in Section 6.4). We delve into the relationships between building data and nearby traffic volume, and the experimental results indicate that building-traffic correlations are *non-linear*, *time-varying*, and *cross-domain*. Second, for Challenge 2, we propose a novel recurrent neural network based on cross-domain learning with two attention mechanisms for traffic volume prediction (in Section 6.5). In general, an RNN-based encoder encodes the input sequence data into a feature representation and another RNN-based decoder decodes the encoded input information for the prediction results. Specifically, a cross-domain attention

mechanism captures building-traffic correlations and adaptively extracts the most relevant building data at each time interval. Then, a temporal attention mechanism is employed to model the temporal dependencies across historical time intervals. Finally, we implement a prototype system of *BuildSenSys* and conduct comprehensive evaluations with one-year real-world datasets.

In summary, we make three key contributions in this chapter as follows.

- To the best of our knowledge, we are the first to conduct comprehensive building-traffic analysis with multi-source real-world datasets. By applying multi-source cross-verification, this work not only discloses *how* but also sheds light on *why* the building data is correlated with nearby traffic volume. Essentially, the changes in building occupancy induced by the commuting activities of occupants are highly related to the dynamics of traffic volume on nearby roads. In addition, the higher probability building occupants pass through a road, the stronger building-traffic correlation there exists.

- We propose a novel recurrent neural network for traffic volume prediction based on cross-domain learning. It leverages a cross-domain attention-based encoder and a temporal attention-based decoder to extract the *non-linear*, *time-varying*, *cross-domain* building-traffic correlations accurately and further achieves accurate traffic volume prediction.

- The extensive evaluation studies demonstrate that *BuildSenSys* outperforms all baseline methods with up to 65.3% accuracy improvement (*e.g.*, 2.2% mean absolute percentage error) in predicting nearby traffic volume. We believe that this work can open a new gate of reusing building sensing data for traffic sensing, and further establish connections between smart buildings and intelligent transportation.

The rest of this chapter is organised as follows. Section 6.2 reviews related literature, and Section 6.3 introduces an overview of the *BuildSenSys* system. Then, Section 6.4 presents the correlation analysis of building data and nearby traffic data. Section 6.5 formulates the prediction problem and presents the cross-domain

learning-based recurrent neural network for traffic prediction. Section 6.6 evaluates *BuildSenSys* through extensive experimental studies with real-world datasets. Finally, Section 6.7 discusses some critical issues of reusing building data, and Section 6.8 concludes this chapter.

6.2 Related Work

Reusing building data: Building sensing data is originally dedicated to management and control purposes [151]; therefore, reusing building sensing data for traffic prediction will not entail any extra cost. Beyond building management, reusing building sensing data has attracted considerable research interests from smart city applications, including urban transportation [138], crowd flow patterns [136], and data integration [152]. According to [153], a single set of traffic monitoring systems with camera detectors could easily cost $2500 USD, and over 100 million dollars of such devices can only cover a quarter of roads in a typical metropolitan city [138]. In comparison with conventional methods [141,145–147] that fully rely on sensing data from traffic monitoring systems, it is considerably low-cost while highly reliable to make a second use of building sensing data to predict nearby traffic. For example, Zheng et al. [136] studied the impacts of buildings on human movements and further developed a new urban mobility model for urban planning. Hu et al. [154] proposed a communication sharing architecture for smart buildings to organise in-building IoT devices with heterogeneous data communication.

Moreover, Zheng et al. [138] proposed to use indoor CO_2 data to estimate building occupancy and further developed an occupancy-traffic model for traffic speed prediction. Inspired by the above existing works, it is of great significance to unify traffic monitoring systems with external sensing infrastructures to enhance traffic prediction accuracy and reduce marginal cost. In this study, we propose innovative reuse of multi-dimension building sensing data for traffic prediction. As a preliminary exploration, we extensively investigate the building-traffic correlations and apply cross-domain learning to achieve accurate traffic prediction.

Traffic volume prediction: Most existing works of predicting traffic volume

use data from fixed road-based traffic sensors, such as loop detectors, microwave radars, and video cameras [143]. The main advantage of road-based sensors is that they can provide reliable data by capturing all vehicles passing by the corresponding roads [155]. Recently, opportunistic sensing and crowdsensing techniques have been utilised to collect GPS data [141, 156, 157], and cellular record data [158–160] from floating vehicles and mobile passengers. These trajectories provide detailed mobility traces for network-wide traffic sensing and prediction. Many existing works have integrated both traffic sensor data and opportunistic sensing data for traffic prediction. For example, Meng et al. [161] further combined loop detector data and taxi trajectories with a spatio-temporal semi-supervised learning model to infer traffic volume. However, with the inherent biases and random data deficiency, most trajectory data cannot cover the entire traffic dynamics. For instance, GPS data of a 6,000-taxi network can only cover 28% of the overall road segments in a large city with distinctive operating time [148]. With such data sparsity, trajectory sensing data cannot guarantee the sustainability and reliability in traffic prediction, especially on explicitly targeted road segments. In this work, the source data for traffic prediction is building sensing data, which is quite different from source data in most existing works. Both the buildings and their surrounding roads are stationary, and inherently they have permanent spatial relations. Most importantly, as building-traffic correlations are sustainable, reusing building data is cost-efficiency and highly reliable for long-term traffic prediction.

Traffic prediction models: Conventional short-term traffic prediction approaches mainly apply parameter-based prediction models, for example, Autoregressive Integrated Moving Average (ARIMA) [162], Vector Autoregression (VAR) [163] and Locally Weighted Linear Regression (LWR) [138]. With the rising of deep learning, deep neural networks [164] have been adopted for traffic prediction as they can capture complex temporal-spatial dependencies through feature learning [165]. Moreover, Recurrent Neural Networks (RNN) have also been adopted by [166] to perform sequence learning on historical traffic data. However, RNNs are not capable of preserving long-term dependencies on historical traffic data, as their performance

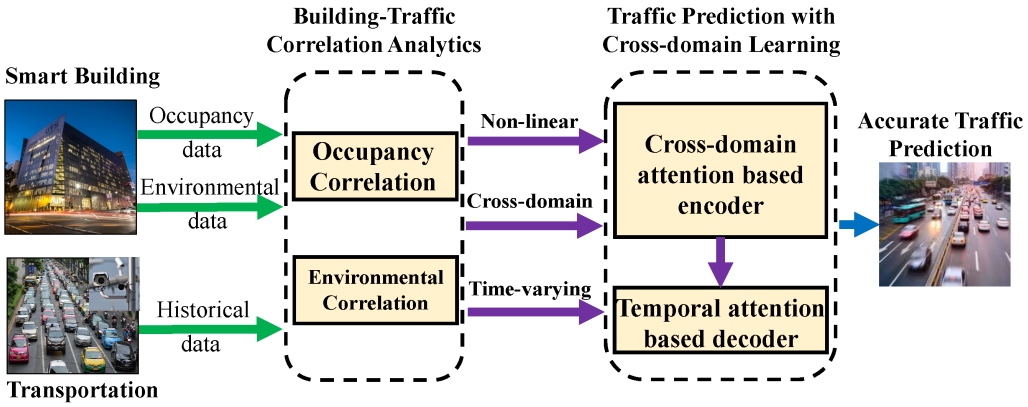


Figure 6.2 : An overview of the BuildSenSys system for reusing building data for nearby traffic prediction.

would deteriorate with longer input. As a result, Long Short-Term Memory (L-STM) networks are further adopted by many research studies [167, 168] to perform long-term prediction tasks. Inspired by the human's ability to capture a focus in certain visions, attention mechanisms have been integrated into the neural network-s for sequence-to-sequence learning [169]. For instance, [170] proposed a spatial-temporal dynamic network with a periodically shifted attention mechanism to capture periodic temporal similarity in traffic predictions. In this chapter, we devise a cross-domain learning-based recurrent neural network with a cross-domain attention mechanism and a temporal attention mechanism. Our model can effectively extract cross-domain, non-linear, and time-varying building-traffic correlations for accurate traffic prediction.

6.3 System Overview

In this section, we briefly present the system overview of reusing building data for traffic volume prediction. As shown in Fig. 6.2, the data sources of *BuildSenSys* include a smart building (generating occupancy data and environmental data) and nearby traffic monitors (providing traffic volume data). More importantly, *Build-SenSys* consists of two main components as follows.

1) **Building-traffic correlation analysis with multi-source real-world datasets** (in Section 6.4). Extensive experimental analysis is conducted based on

real-world smart building data, traffic data of fixed-road sensors, and Google traffic data. We investigate not only whether building data is correlated with the traffic data but also what the underlying reasons are. The results show that the correlation is positive, and there are two types of cross-domain correlations, including i) **occupancy correlation** (in Section 6.4.1) between the building occupancy and traffic data, and ii) **environmental correlation** (in Section 6.4.2) between the building/outdoor environmental data and traffic data. Besides, the building-traffic correlations are dynamic and time-varying, generating the temporal correlations along with the time.

2) **Cross-Domain Learning for Accurate Traffic Prediction** (in Section 6.5). In correspondence to cross-domain correlations and temporal correlations, we propose cross-domain learning to predict traffic volume accurately. Specifically, we propose a cross-domain attention mechanism for the encoder to learn features from non-linear, cross-domain correlations between building data and traffic data (in Section 6.5.2). By correlating these features with traffic volume data, the most relevant cross-domain correlations are extracted through the training process of the *BuildSenSys* model. Furthermore, we present a temporal attention mechanism for the decoder, which aims to learn the temporal dependencies for the predicted traffic data (in Section 6.5.2). After that, the output context factors are leveraged to predict traffic volume.

6.4 Building-traffic Correlation Analysis with Multi-source Datasets

In this section, we conduct comprehensive experiments to explore spatial and temporal building-traffic correlations with multi-source real-world datasets as follows. To begin with, we briefly introduce the basics of building sensing data and traffic volume data as follows.

First, we collect building sensing data from the office building of Faculty of Engineering and Information Technology at the University of Technology Sydney, New South Wales, Australia. The FEIT building is a 16-level campus building with

a total usable floor area of 23,500 m^2, and it has the capacity to accommodate a maximum population of over 5,000. As illustrated in Fig. 6.1a, the FEIT building is described as a 'living laboratory', with around 2,500 internal environment sensors installed across all floors and public spaces. These sensors constantly monitor the internal and external environments of the building, including indoor environment (by environmental sensors), outdoor environment (by a roof-top weather station), and building occupancy (by smart cameras). We access all building sensing data through an online database server via MySQL workbench, which contains 33 types of building sensing data with a total volume of over 10 GB [171]. In this study, we have leveraged 10 most relevant building sensing data from FEIT building to achieve cross-domain traffic sensing and prediction.

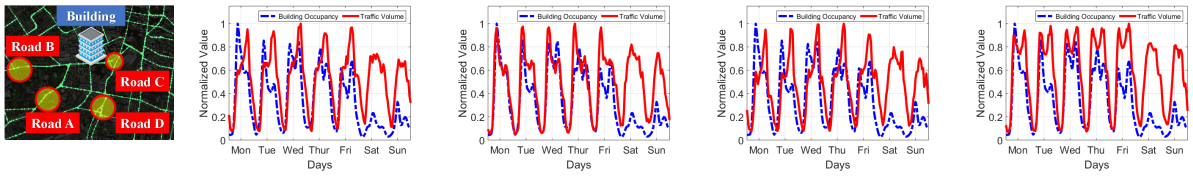
Second, as shown in Fig. 6.1b, the traffic data are the traffic volumes collected by permanently deployed traffic counters on several road segments that are proximate to the FEIT building. Each traffic counter is integrated to a traffic monitoring system for calculating hourly traffic volume data. We access historical traffic volume data from the official website of the Department of Roads and Maritime Services, New South Wales State [172]. In the following, we present a detailed cross-domain correlation analysis with traffic volume data and different building sensing data.

6.4.1 Correlation analysis with building occupancy data

In this section, we investigate cross-domain correlations between building occupancy data and traffic volume. The building occupancy data is collected by a number of cameras installed at building entrances, stairways, and walkways to monitor the people's movement inside the building. Based on the sensing data of these cameras, we use the PLCount algorithm [173] to compute the total building occupancy accurately. As the calculation of building occupancy is not our focus, we omit its details that can be referred from [173].

Correlation quantification with metrics

With the preprocessed building occupancy, we closely study the correlations between building occupancy and nearby traffic volume. First, we compare building

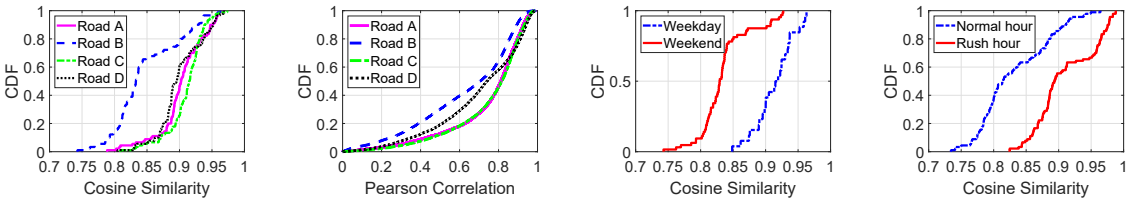


(a) Locations (b) Road A (c) Road B (d) Road C (e) Road D

Figure 6.3 : Comparisons of normalised building occupancy and normalised traffic volume on different roads.

occupancy with the traffic volume of nearby roads in one week. Fig. 6.3 shows the comparison of normalised building occupancy (dashed blue line) and traffic volume (solid red line) of roads A-D. Note that the above four roads are different types of roads, where road A is a minor highway, road B is a major highway, road C is the main street, and road D is a primary street. We observe that the building occupancy and traffic volume exhibit similar hourly patterns on weekdays and different patterns on weekends, respectively. This is because the high volume of building occupancy is driven by the working days, and there are just a few overtime workers inside the building on weekends. In particular, the building occupancy closely follows the rise of traffic volume in both morning rush hours and evening rush hours.

However, the comparison between normalised building occupancy and traffic volume only shows the general dynamics and correlations. To further quantify the correlations between building occupancy and traffic volume, we employ two correlation metrics, *i.e.*, Cosine Similarity and Pearson Correlation Coefficient, to evaluate the above correlations. The *Cosine Similarity* quantifies the similarity between two non-zero vectors on an inner product space. We use s_i to represent the cosine similarity between building occupancy and traffic volume on the day i. It is computed as $s_i = \cos(\mathbf{b}_i, \mathbf{t}_i)$, where $\mathbf{b_i} = [b_{i,1}, b_{i,2}, ..., b_{i,24}]$ and $\mathbf{t_i} = [t_{i,1}, t_{i,2}, ..., t_{i,24}]$ denote the vectors of building occupancy and traffic volume over 24 hours on the day i, respectively. In this way, the overall cosine similarity between building occupancy and traffic volume for n days is denoted by $\mathbf{s} = [s_1, ...s_i, ...s_n]$. Moreover, the *Pearson Correlation* is computed as $p = \sum_{j=1}^{k} (b_{i,j} - \bar{b}_i)(t_{i,j} - \bar{t}_i) / \sqrt{\sum_{j=1}^{k} (b_{i,j} - \bar{b}_i)^2 \sum_{j=1}^{k} (t_{i,j} - \bar{t}_i)^2}$,



(a) Different roads (Cosine) (b) Different roads (Pearson) (c) Weekday vs. Weekend (d) Normal hour vs. Rush hour

Figure 6.4 : Quantification of correlations between building occupancy data and traffic volume data with two metrics.

where k denotes the time interval for each Pearson Coefficient, $\bar{b}_i$ and $\bar{t}_i$ are the average building occupancy and traffic volume at the day i, respectively. Note that the absolute value of Pearson Correlation indicates the strength of the correlation, ranging from 0 (weak correlation) to 1 (strong correlation).

We quantify correlations between building occupancy and traffic volume on four different road segments, using both Cosine Similarity and Pearson Correlation. As illustrated in Fig. 6.4a, the cosine similarities on all four roads are greater than 0.8, showing that building occupancy is strongly correlated with traffic volume. Meanwhile, Fig. 6.4b shows that over 70% of traffic volume data has strong and positive ($\geq$0.5) Pearson Correlation with building occupancy data. In addition, we evaluate the cosine similarity of building occupancy and traffic volume by dividing the data into different groups, *i.e.*, weekdays/weekends and rush hours/normal hours, respectively. From Fig. 6.4c and Fig. 6.4d, it is clear that building occupancy and traffic volume have higher correlations on the weekdays ($\geq$0.85) and rush hours ($\geq$0.82) than the weekends ($\geq$0.75) and normal hours ($\geq$0.75), respectively.

Correlation verification with Google Maps

The results in Section 6.4.1 indicate that the building occupancy data is highly correlated with the traffic volume on nearby road segments. Through an in-depth analysis, the main reason is that most of the building occupants entering or leaving a building would pass the surrounding roads via transportation. Intuitively, the building-traffic correlations are dependent on the probability of building occupants

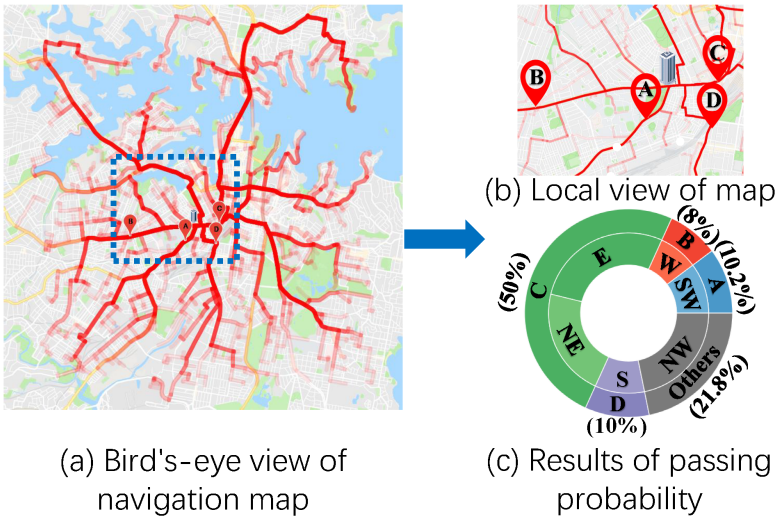


Figure 6.5 : Cross-verification for building-traffic correlations via Google Maps Navigation.

passing by each road segment. That is, the building-traffic correlation would be stronger when a road is with higher passing probability by the building's occupants.

To verify this assumption, we further investigate the relationship between building occupancy and traffic volume by conducting cross-verification experiments with Google Maps. First, as shown in Fig. 6.5a, we set this building's location as the starting points or the endpoints, while randomly selecting 500 locations within 10 km distance of this building as the endpoints or starting points. Each randomly selected point, together with the building's location, is packaged as a navigation request and then sent to Google Maps for navigation. Then, the best-fit road geometries are returned with a series of GPS locations. We visualise 500 navigation results in Fig. 6.5, where the colour of each route indicates its passing probability, and a route with the deeper colour has a higher passing probability.

As illustrated in Fig. 6.5a, some long-distance routes would pass through the main roads, for example, major highways and primary streets. Moreover, Fig. 6.5b shows a zoom-in view of roads A-D. Specifically, the passing probabilities of roads A-D are shown in Fig. 6.5c, where the outer circle represents the passing probability of four road segments, and the inner circle represents the main directions of navigation routes. It can be observed that the passing probabilities of roads A, B,

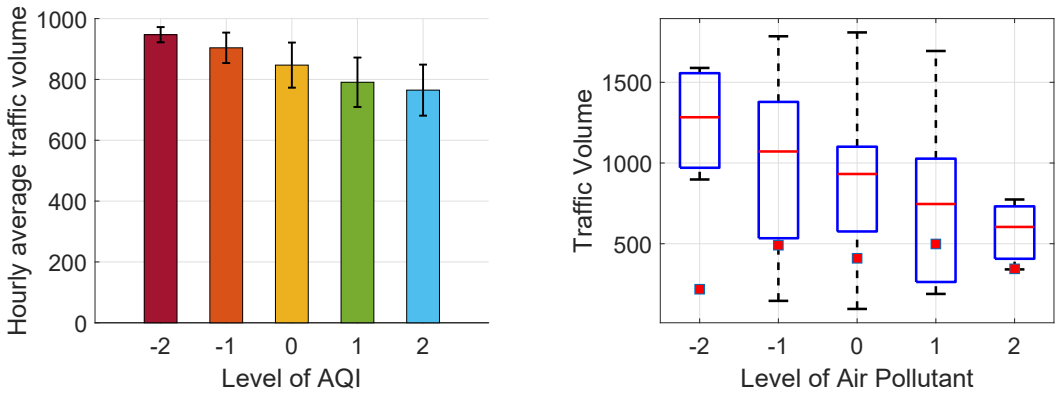
Road Name	Passing Ratio	Cosine ranking	Pearson Ranking	Distance
Road C	50%	1st	1st	0.3 km
Road A	10.2%	2nd	2nd	1.6 km
Road D	10%	3rd	3rd	1.9 km
Road B	8%	4th	4th	3.0 km

Table 6.1 : Cross-verification: comparison between roads A, B, C, and D in navigation passing probability, Cosine similarity, Pearson correlation, and distance to the building.

C, D are 10.2%, 8%, 50%, and 10%, respectively, while the others only have about 20% passing ratio. By using the cross-verification method, we compare these results based on Google Maps with building-traffic correlations in Section 6.4.1. As indicated in Table 6.1, the results from multi-source datasets are consistent with each other. That is, a road segment with higher passing probability and shorter distance to the building would have stronger building-traffic correlations with this building. As a result, the above results of cross-verification further verify the reason behind building-traffic correlation as follows. *The building occupants entering or leaving a building would pass the surrounding roads via transportation. Hence, the change of building occupancy data (induced by the commuting activity of the building's occupants) is highly related to the traffic volume of nearby road segments.*

6.4.2 Correlation analysis with environmental data

Besides building occupancy, environmental data have been proved to affect traffic on the nearby roads evidently, and they can also contribute to the improvement of traffic prediction accuracy [146,174]. In this study, for cross-domain traffic prediction, we typically select 5 types of indoor environmental data (CO_2 concentration, building humidity, O_2 concentration, building temperature and building air pollution), and 4 types of outdoor environmental data (including the outdoor temperature, rainfall, wind speed, and air quality index) from the FEIT building. Through conducting a series of correlation studies with environmental data and traffic volume data, we find that different types of environmental data have different levels of correlations with traffic volume data. While some of the environmental data have



(a) Outdoor AQI (b) Indoor Air Pollutant

Figure 6.6 : Correlation analysis between the building environmental data and traffic data (-2 stands for the worst level, 0 for the moderate level, and 2 for the best level).

stronger correlations with outdoor traffic volume, others show weak correlations. To this end, we report two representative cases for correlations between environmental data and traffic data as follows.

Fig. 6.6a presents correlations between the outdoor Air Quality Index (AQI) and traffic volume by averaging values of traffic data under categorised air quality conditions. We categorise air quality values into five levels from the worst to the best, represented by -2 to 2. It can be observed that AQI stays at a lower level when the traffic volume is in higher values. The main reason is that when the daily traffic volume is at a high level, more emissions from vehicles can directly pollute the air quality, thereby affecting outdoor AQI.

Moreover, we evaluate the correlations between traffic volume and indoor air pollutant levels. As shown in Fig. 6.6b, the box plot illustrates averaged traffic volume with categorised air pollutant levels. It can be observed that the indoor air pollutant becomes worse with higher traffic volume. Although indoor environmental factors are not directly related to outdoor traffic volume, the indoor environmental data can be used as the indirect sensing data for building occupants [138]. An essential objective of using indoor environmental data is to assist traffic prediction when the value of building occupancy is at abnormal values. For example, in case of an emergency evacuation, there will be a rapid removal of people inside the building,

which will result in higher but false occupancy data generated by smart cameras. If we combine indoor environmental data in the prediction model, the prediction errors in the above cases can be eliminated effectively.

To conclude, while the environmental data is correlated with the traffic volume, their correlations are not very strong. Nevertheless, the environmental data and the occupancy data can be complementary to each other in cross-domain traffic prediction with building data.

6.5 Accurate Traffic Prediction with Cross-Domain Learning of Building Data

In this section, based on the correlation analysis in Section 6.4, we propose a cross-domain learning method for traffic volume prediction by reusing building sensing data. First, we formulate the traffic volume prediction problem of using cross-domain building sensing data. Second, to address this problem, we propose a cross-domain learning-based recurrent neural network to learn the non-linear, time-varying, and cross-domain building-traffic correlations for accurate traffic volume prediction.

6.5.1 Problem Formulation

In this work, our goal is to predict traffic volume by leveraging building sensing data and historical traffic volume data. Assume that there are N types of building sensing data for T time intervals, we further introduce the following notations.

- **Building data types**: building's IoT sensors generate N types of sensing data that are used for traffic prediction, including N_o types of occupancy sensing data (*e.g.*, covering different public zones) and N_e types of environmental sensing data (*e.g.*, indoor environmental and outdoor environmental data). Intuitively, $N = N_o + N_e$.

- **Building sensing data**: let $x_t(i)$ denote i-th building sensing data at time t and $\mathbf{x}_t = \{x_t(i)|1 \leq i \leq N\}$ denote a vector of all building sensing data at the

time t. Accordingly, $\mathbf{X}_{[1:T]} = [\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_T]_{T \times n}$ denotes a measurement matrix of all building data across T time intervals.

- **Traffic volume data**: let y_t represent traffic volume of a target road segment at time t, $1 \leq t \leq T-1$. Accordingly, $\mathbf{y}_{[1:T-1]}$ denotes a vector of historical traffic volume of the target road segment across $T-1$ time intervals, where $\mathbf{y} = \{y(j) | 1 \leq j \leq T-1\}$.

- **Future Traffic volume**: the predicted traffic volume of a target road segment is denoted as $\widehat{\mathbf{Y}}_{[T:T+\tau]} = \{y(j) | T \leq j \leq T+\tau\}$ while using $\mathbf{Y}_{[T:T+\tau]}$ to represent its ground truth, where τ is the time intervals for prediction.

Problem Definition: formally, given n types of building sensing data over T time intervals and historical data of a target road segment over $T-1$ time intervals, the traffic volume prediction problem for τ future time intervals is to optimise the prediction errors and defined as:

$$\textbf{Minimize} \quad ||\widehat{\mathbf{Y}}_{[T:T+\tau]} - \mathbf{Y}_{[T:T+\tau]}||_F^2, \tag{6.1}$$

$$\text{where} \quad \widehat{\mathbf{Y}}_{[T:T+\tau]} = F(\mathbf{y}_{[1:T-1]}, \mathbf{X}_{[1:T]}), \tag{6.2}$$

where $F(\cdot)$ denotes the nonlinear mapping function from building sensing data to traffic data that a predicting model needs to learn.

6.5.2 Attention mechanisms-based encoder-decoder Recurrent Neural Network

To solve the cross-domain learning-based traffic volume prediction problem, we propose *BuildSenSys*, an LSTM based encoder-decoder architecture with dual-attention mechanisms. First, we employ a cross-domain attention on input data to capture building-traffic correlations and adaptively select the most relevant building sensing data at each step of prediction. Second, we apply a temporal attention to capture temporal features from historical dependencies. Then, *BuildSenSys* adaptively select the most relevant encoder hidden states across all time intervals.

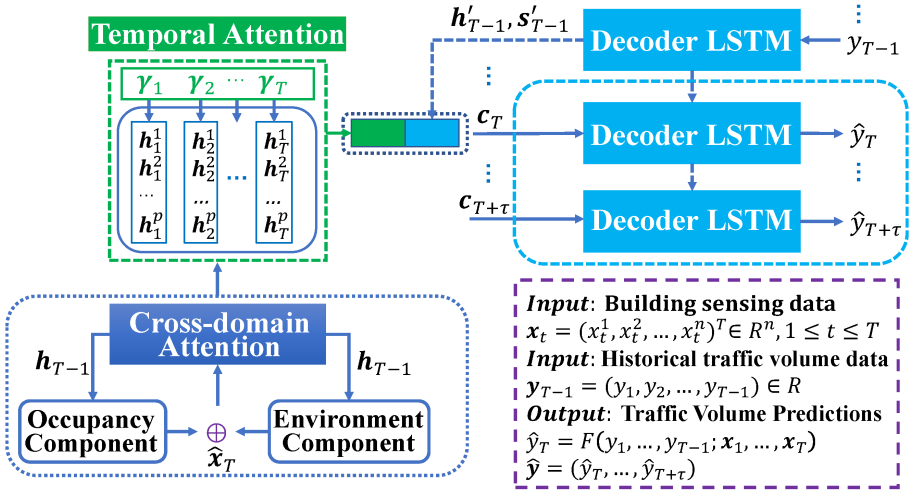


Figure 6.7 : The graphical architecture of cross-domain attention-based recurrent neural networks for cross-domain traffic prediction.

By integrating the above attention mechanisms with LSTM-based recurrent neural networks, we can further jointly train the *BuildSenSys* model with standard back-propagation. As a result, *BuildSenSys* is capable of selecting the most relevant building sensing data for traffic prediction and capturing long-term temporal features of traffic volume data. The overall framework of *BuildSenSys* is presented in Fig. 6.7.

Encoder with cross-domain attention

The encoder in the *BuildSenSys* framework is an LSTM-based recurrent neural network, and it encodes the input sequence into a feature vector. For cross-domain traffic volume prediction, given N types of building sensing data, we denote the input sequence as $\mathbf{X} = (\mathbf{x}_1, ..., \mathbf{x}_t, ..., \mathbf{x}_T)$, where $\mathbf{x}_t \in \mathbb{R}^N$. The hidden state of the encoder at time interval t is computed by

$$\mathbf{h}_t = f_e(\mathbf{h}_{t-1}, \mathbf{x}_t), \tag{6.3}$$

where $\mathbf{h}_{t-1} \in \mathbb{R}^p$ is the previous hidden state of the encoder at time interval $t-1$, p is the size of the hidden state in the encoder, and f_e is an LSTM based recurrent neural network. Since LSTMs are capable of learning long-term dependencies, we

employ the classic LSTM unit that is with one memory cell and three sigmoid gates. At time interval t, the cell state of memory is $\mathbf{s}_t$, the forget gate is $\mathbf{f}_t$, and the input gate is $\mathbf{i}_t$. The encoder LSTM updates its hidden state by

$$\mathbf{f}_t = \sigma(\mathbf{W}_f[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_f), \tag{6.4}$$

$$\mathbf{i}_t = \sigma(\mathbf{W}_i[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_i), \tag{6.5}$$

$$\mathbf{o}_t = \sigma(\mathbf{W}_o[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_o), \tag{6.6}$$

$$\mathbf{s}_t = \mathbf{f}_t \odot \mathbf{s}_{t-1} + \mathbf{i}_t \odot \tanh(\mathbf{W}_s[\mathbf{h}_{t-1}; \mathbf{x}_t] + \mathbf{b}_s), \tag{6.7}$$

$$\mathbf{h}_t = \mathbf{o}_t \odot \tanh(\mathbf{s}_t), \tag{6.8}$$

where $[\cdot;\cdot]$ is a concatenation operation, σ is a logistic sigmoid function, $\odot$ is a pointwise multiplication, and $\mathbf{W}_f$, $\mathbf{W}_i$, $\mathbf{W}_o$, $\mathbf{W}_s$, $\mathbf{b}_f$, $\mathbf{b}_i$, $\mathbf{b}_o$ and $\mathbf{b}_s$ are the learnable parameters.

Inspired by that visual attention allows human to focus on a certain region of images or sentences for creating the perception of information, attention mechanism has become an integral part of the compelling sequence modelling and transduction models [169]. In general, attention mechanisms are proposed to make a soft selection over historical data by calculating and assigning different weights to them. In this work, we aim to predict the traffic volume with different types of building sensing data and historical traffic volume data. To achieve this goal, we propose a cross-domain attention to capture complex building-traffic correlations and enhance the feature representation of all input data. In specific, based on the results of cross-domain correlation analysis, we propose to learn different correlations with the occupancy component and environmental component, respectively.

Occupancy Component: we envision a multi-zone scenario for traffic volume prediction, with j_{th} zone's occupancy as $\mathbf{x}^j = (x_1^j, x_2^j, ..., x_t^j)^\top \in \mathbb{R}^T$ and $1 \leq j \leq N_o$. The cross-domain attention on occupancy components is calculated by referring to

the previous hidden state $\mathbf{h}_{t-1}$ and previous cell state $\mathbf{s}_{t-1}$ of encoder LSTM by

$$o_t^j = \mathbf{v}_o^\top \tanh(\mathbf{W}_o[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}] + \mathbf{U}_o \mathbf{x}^j + \mathbf{b}_o), \tag{6.9}$$

$$\beta_t^j = \frac{\exp(o_t^j)}{\sum_{i=1}^{N_o} \exp(o_t^i)}, \tag{6.10}$$

where $\mathbf{v}_o, \mathbf{b}_o \in \mathbb{R}^T$, $\mathbf{W}_o \in \mathbb{R}^{T \times 2p}$, and $\mathbf{U}_o \in \mathbb{R}^{T \times T}$ are learning parameters. By using another softmax function, we ensure that all attention weights with the occupancy component are normalised, and the total weight for all occupancy input data is 1. With cross-domain attention weights for all public zones, we acquire the output vector from the occupancy component at time interval t as:

$$\hat{\mathbf{x}}_t^{occ} = (\beta_t^1 x_t^1, ...\beta_t^j x_t^j, ...\beta_t^{N_o} x_t^{N_o})^\top. \tag{6.11}$$

Environmental Component: given k_{th} type of environmental input data as $\mathbf{x}^k$, where $\mathbf{x}^k = (x_1^k, x_2^k, ..., x_t^k)^\top \in \mathbb{R}^T$ and $1 \leq k \leq N_e$. We employ the cross-domain attention for the environmental component to adaptively capture the dynamic correlation between traffic volume and k_{th} type of environmental data by

$$e_t^k = \mathbf{v}_e^\top \tanh(\mathbf{W}_e[\mathbf{h}_{t-1}; \mathbf{s}_{t-1}] + \mathbf{U}_e \mathbf{x}^k + \mathbf{b}_e), \tag{6.12}$$

$$\alpha_t^k = \frac{\exp(e_t^k)}{\sum_{i=1}^{N_e} \exp(e_t^i)}, \tag{6.13}$$

where $[\cdot;\cdot]$ is a concatenation operation. Here, $\mathbf{v}_e, \mathbf{b}_e \in \mathbb{R}^T$, $\mathbf{W}_e \in \mathbb{R}^{T \times 2p}$ and $\mathbf{U}_e \in \mathbb{R}^{T \times T}$ are learnable parameters. By applying a softmax function to e_t^k, we obtain the normalised attention weight α_t^k for k_{th} environmental data at the time interval t. With cross-domain attention weights for all types of environmental data input, the output vector of the environmental component at time interval t can be adaptively acquired by

$$\hat{\mathbf{x}}_t^{env} = (\alpha_t^1 x_t^1, ...\alpha_t^k x_t^k, ...\alpha_t^{N_e} x_t^{N_e})^\top. \tag{6.14}$$

Finally, for encoder LSTM, we adaptively concatenate above output vectors from different components and extract the final output vector of cross-domain attention mechanism as:

$$\hat{\mathbf{x}}_t = [\hat{\mathbf{x}}_t^{env}; \hat{\mathbf{x}}_t^{occ}], \tag{6.15}$$

where $\hat{x}_t \in \mathbb{R}^N$. We feed the final output vector $\hat{\mathbf{x}}_t$ into the encoder LSTM as its new input at time interval t. Consequently, the hidden state of encoder LSTM in Eq. 6.3 is updated by

$$\mathbf{h}_t = f_e(\mathbf{h}_{t-1}, \hat{\mathbf{x}}_t), \tag{6.16}$$

where f_e is the encoder LSTM network described in Eq. 6.4 to Eq. 6.8.

Decoder with temporal attention

The basic idea of attention mechanism is to distinguish task-related importance of input data in historical time intervals. Intuitively, the decoder with a temporal attention mechanism can be trained to capture temporal dependencies between traffic volume and building data by calculating attention weights for all encoder hidden states. With the temporal attention weights assigned to all encoder hidden states, decoder LSTM can focus on the most relevant input data during decoding [169]. To this end, we employ a temporal attention mechanism that enables the decoder to select relevant encoder hidden states across all time intervals adaptively. In specific, when computing the attention vector for encoder hidden state at time interval t, the temporal attention mechanism refers to the previous hidden state $\mathbf{h'}_{t-1}$ and previous cell state $\mathbf{s'}_{t-1}$ of decoder LSTM by

$$\mu_t^i = \mathbf{v}_d^\top \tanh(\mathbf{W}_d[\mathbf{h'}_{t-1}; \mathbf{s'}_{t-1}] + \mathbf{U}_d\mathbf{h}_i + \mathbf{b}_d), 1 \leq i \leq T, \tag{6.17}$$

$$\gamma_t^i = \frac{\exp(\mu_t^i)}{\sum_{l=1}^{T} \exp(\mu_t^l)}, \tag{6.18}$$

where $[\cdot]$ is a concatenation operation, and $\mathbf{v}_d, \mathbf{b}_d \in \mathbb{R}^p$, $\mathbf{W}_d \in \mathbb{R}^{p \times 2q}$ and $\mathbf{U}_d \in \mathbb{R}^{p \times p}$ are learning parameters. Through a softmax layer, the temporal attention weight γ_t^i is calculated for the i_{th} encoder hidden state at time interval t. As the component of the input sequence is temporally mapped to each encoder, we calculate the context

vector $\mathbf{c}_t$ as a weighted sum of all encoder hidden states by

$$\mathbf{c}_t = \sum_{i=1}^{T} \gamma_t^i \mathbf{h}_i. \tag{6.19}$$

To capture the dynamic temporal correlation in traffic volume data, we further combine the context vector with $\mathbf{y} = (y_1, ..., y_t, ..., y_{T-1})$ as

$$\tilde{y}_{t-1} = \tilde{w}^\top [y_{t-1}; \mathbf{c}_{t-1}] + \tilde{b}, \tag{6.20}$$

$$\mathbf{h'}_t = f_d(\mathbf{h'}_{t-1}, \tilde{y}_{t-1}), \tag{6.21}$$

where f_d is an LSTM-based recurrent neural network as decoder, and $\tilde{w} \in \mathbb{R}^{p+1}$ and $\tilde{b} \in \mathbb{R}$ are parameters to map the concatenation result to the size of the decoder input. As the structure of the LSTM unit in the decoder is exactly the same as the encoder (referred to Eq. 6.4 to 6.8), we omit the update process of f_d. Finally, the attention mechanism based recurrent neural network concatenates the context vector $\mathbf{c}_T$ with decoder hidden state $\mathbf{h'}_T$, predicting the traffic volume at time interval T as

$$\hat{y}_T = \mathbf{v}_y^\top (\mathbf{W}_y [\mathbf{c}_T; \mathbf{h'}_T] + \mathbf{b}_y) + b, \tag{6.22}$$

where $[\mathbf{c}_T; \mathbf{h'}_T] \in \mathbb{R}^{p+q}$ is a concatenation operation, and parameters $\mathbf{W}_y \in \mathbb{R}^{q \times (p+q)}$ and $\mathbf{b}_y \in \mathbb{R}^q$ together map the concatenation to the size of decoder hidden states. The final output is generated by a mapping function with weights $\mathbf{v}_y \in \mathbb{R}^q$ and $\mathbf{b}_y \in \mathbb{R}$.

6.6 Performance Evaluation

To evaluate the performance of *BuildSenSys*, we develop a prototype system, as shown in Fig. 6.8. We first train the cross-domain learning-based RNN model with a training set of building sensing data and traffic data. With the pre-trained model, *BuildSenSys* can output the predicted traffic volume on nearby roads only with building sensing data as the input. In the following, we first introduce the experimental settings, including dataset, baseline methods, evaluation metrics, and

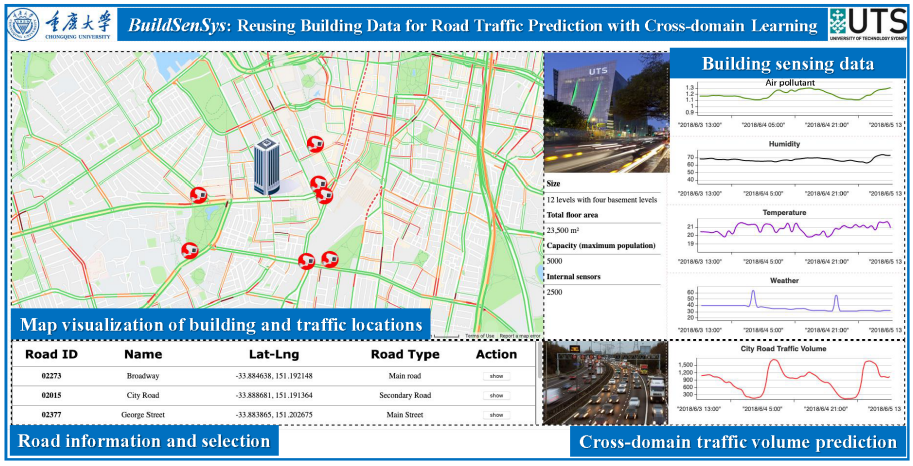


Figure 6.8 : The prototype system of *BuildSenSys* for data visualisation and traffic prediction.

model parameters. Then, we conduct extensive experimental studies on *BuildSenSys* and evaluate its performance in terms of baseline comparison, parameter study, ablation study, attention weight, and other extensive comparisons.

6.6.1 Experimental Methodology and Settings

Dataset Description

Based on the results of the observation study, we select traffic data from nearby roads and relevant building sensing data as the training input data for the *BuildSenSys* model. For traffic data, we collect traffic volume counting data from the official website of the Department of Roads and Maritime Services, New South Wales State [172]. The traffic volume count data is generated by permanent and temporary roadside collection devices, monitoring the number of passing vehicles on each road with the one-hour interval. We collect 12-month traffic volume data (from 1/1/2018 to 31/12/2018) on four nearby roads of the building. For building sensing data, we collect data from three categories, *i.e.*, building occupancy, building environmental data, outdoor environmental data. First, building occupancy data is generated by camera sensors distributed at Point-of-Interest Zones inside the building. We process and aggregate these data with the PLCount algorithm [173] for overall building occupancy. Second, building environmental data include CO_2 concentration, build-

ing humidity, O_2 concentration, building temperature, and building air pollution. Third, the outdoor environmental data is collected from the rooftop weather station of our building, including the outdoor temperature, rainfall, and wind speed. Besides, as vehicle emissions have a direct influence on outdoor air quality, we further adopt the hourly AQI data from the Bureau of Meteorology's official website and integrate the AQI data into outdoor environmental data for traffic prediction. At last, we synchronise all building sensing data and traffic volume data with the one-hour interval for training and testing purposes.

Baseline Methods

To comprehensively evaluate the performance of *BuildSenSys*, we compare it with seven baseline methods as follows.

- **HA** [175]: The historical average (HA) model, which predicts the traffic volume by averaging the historical value of all corresponding time intervals.
- **ARIMA** [176]: The autoregressive integrated moving average (ARIMA) model, which is a classic model to predict future time series.
- **VAR** [177]: The Vector Auto-regressive (VAR) model, which is an extension of the univariate autoregressive model and has been widely used for multivariate time series forecasting.
- **LWR** [138]: The Locally Weighted Linear Regression (LWR) model, which is a non-parametric model and performs regressions around points of interest.
- **LSTM** [167]: The Long Short-Term Memory (LSTM) network, which is a variation of recurrent neural networks designed for avoiding the vanishing gradient problem.
- **Seq2Seq** [178]: The Sequence to Sequence model based on an encoder-decoder architecture and recurrent neural networks, consisting of three parts, *i.e.*, encoder, context vector, and decoder.

- **Seq2Seqw/attn** [178]: The Sequence to Sequence model with a temporal attention mechanism.

Note that all baseline methods take the time of a day as an essential feature of input data, and they use this feature for traffic prediction from different perspectives. For example, the HA leverages time of the day with statistical regression, while ARIMA and VAR process this feature with autoregression. Moreover, the neural networks of the LSTM and Seq2Seq models learn the time of the day as a feature with their hidden states. In the proposed *BuildSenSys* model, we further address the time of the day by employing a temporal attention mechanism to learn temporal features of traffic data.

Evaluation Metrics and Parameter Settings

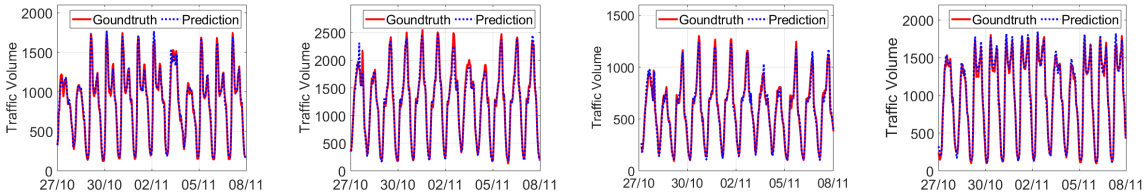
To evaluate the prediction accuracy, we employ three widely used evaluation metrics, *i.e.*, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and Mean Absolute Percentage Error (MAPE) [179]. Both MAE and RMSE are scale-dependent metrics, and MAPE is a scale-independent metric. Specifically, MAE measures the average magnitude of errors in prediction results as Eq. 6.23. RMSE measures the square root of the average squared differences between prediction results and ground truth as Eq. 6.24. MAPE measures the size of errors in percentage terms to quantify the prediction accuracy as Eq. 6.25.

$$\mathbf{MAE} = \frac{1}{\tau} \sum\nolimits_{[T:T+\tau] \in \mathbb{R}^T} \left| \mathbf{Y}_{[T:T+\tau]} - \hat{\mathbf{Y}}_{[T:T+\tau]} \right|, \tag{6.23}$$

$$\mathbf{RMSE} = \sqrt{\frac{1}{\tau} \sum\nolimits_{[T:T+\tau] \in \mathbb{R}^T} \left(\mathbf{Y}_{[T:T+\tau]} - \hat{\mathbf{Y}}_{[T:T+\tau]} \right)^2}, \tag{6.24}$$

$$\mathbf{MAPE} = \frac{1}{\tau} \sum\nolimits_{[T:T+\tau] \in \mathbb{R}^T} \left| \frac{\mathbf{Y}_{[T:T+\tau]} - \hat{\mathbf{Y}}_{[T:T+\tau]}}{\mathbf{Y}_{[T:T+\tau]}} \right|, \tag{6.25}$$

where $\hat{\mathbf{Y}}_{[T:T+\tau]}$ and $\mathbf{Y}_{[T:T+\tau]}$ are the prediction results of traffic volume from time interval T to $T+\tau$, respectively.



(a) Road A (b) Road B (c) Road C (d) Road D

Figure 6.9 : Comparison between the predicted traffic volume of *BuildSenSys* and the ground truth on four different roads.

In this work, the proposed *BuildSenSys* model is implemented with the Tensorflow framework and trained together with other baseline models on two NVIDIA Quadro P5000 GPUs with 16 GB memory. For model training, the tunable hyperparameters for *BuildSenSys* include the time window L (*i.e.*, the length of input data in hours), the length of predicting window τ (the number of days for future traffic prediction) and the size of hidden states in encoder/decoder (denoted by h_a and h_b, respectively). For LSTM, Seq2Seq, Seq2Seqw/attn and *BuildSenSys* model, h_a and h_b are tuned from 32, 64, 126, 256 to 512, and L is tuned from 4, 6, 12, 18, 24 to 48 (hours), respectively. In the training session, the batch size, learning rate, and dropout rate are set to 256, 0.001, and 0.2, respectively, and Adam is the optimiser for the *BuildSenSys* model. For all building sensing data and traffic volume data, we split the dataset into the training set (70%), validation set (10%) and test set (20%) in chronological order.

6.6.2 Experimental Evaluations

Evaluations on overall prediction results

First, we evaluate the overall prediction accuracy of *BuildSenSys* by comparing prediction results with the ground truth on four different types of roads, *i.e.*, roads A (minor highway), B (major highway), C (main street), and D (primary street). Fig. 6.9 illustrates the predicted traffic volume and the ground truth over 12 days. As shown in Fig. 6.9a to Fig. 6.9d, the prediction results are highly close to the ground truth. Thus, *BuildSenSys* can successfully capture the cross-domain correlations

and temporal dependencies in reusing building data for traffic volume prediction. Moreover, the above results indicate that the distance between a road segment and the building can significantly affect the prediction accuracy. For example, as illustrated in Fig. 6.9b, prediction errors on the road B are much larger than those of the roads A, C and D. As shown in Table 6.1, road B is the furthest (3.0 km) to the building among all four roads.

Comparisons with Baselines

We further quantify the performance of *BuildSenSys* by comparing it with seven baseline methods by using three evaluation metrics. The evaluation experiments are conducted on four road segments. The basic parameter settings of baseline methods are adopted from [180]. For *BuildSenSys*, we set the length of input time window $L = 24$ and the size of hidden states in both encoder LSTM and decoder LSTM as 128 or 512, respectively. Table 6.2 presents the performance comparison between *BuildSenSys* and all baseline methods.

Overall, *BuildSenSys* achieves the best prediction accuracy with the lowest RMSE at 30.49, lowest MAE at 20.29, and lowest MAPE at 2.05%, respectively. It outperforms the best prediction of Seq2Seqw/attn by up to 46.2% in RMSE on the road D, 45.6% in MAE on the road C and 65.3% in MAPE on the road D. In contrast, the LWR model, which also uses building data to predict traffic volume [138], shows the worst performance with RMSE at 239.48, MAE at 190.79 and MAPE at 22.13% on the road D. The key reason here is that the LWR model [138] is based on the assumption of linear building-traffic correlation. Meanwhile, the actual building-traffic correlations are proved to be non-linear, time-varying, and far more complex than linear relations. For other baseline methods that are based on historical traffic data, we present the detailed comparison results as follows.

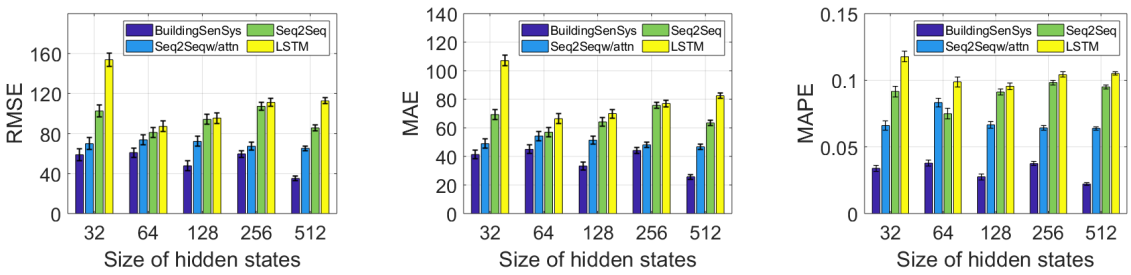
First, ARIMA and VAR methods show the worst prediction performance, as they are not capable of making accurate predictions for the long term, especially on the 'turning points' (*e.g.*, rush hours). Interestingly, the Historical Average (HA) method, as a most naive scheme, performs better than ARIMA and VAR for the

Table 6.2 : Performance comparison with baseline methods on different roads.

Models	Road A			Road B			Road C			Road D		
	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE	RMSE	MAE	MAPE
LWR	244.51	155.69	22.78%	257.86	176.94	25.91%	206.52	176.35	22.61%	239.48	190.79	22.13%
ARIMA	152.74	141.22	14.51%	192.73	142.66	15.58%	149.11	138.09	14.70	173.32	124.31	14.22%
VAR	117.35	110.83	12.99%	120.37	116.9	13.06%	98.65	94.15	11.05	124.61	110.68	12.46%
HA	108.01	89.72	11.37%	125.88	93.57	12.78%	90.71	65.20	10.26%	117.03	89.45	11.12%
LSTM	74.37	58.30	9.83%	91.71	72.47	11.04%	70.71	58.9	9.31%	90.18	67.37	10.39%
Seq2Seq	72.86	51.06	7.40%	89.25	64.78	8.29%	66.19	53.32	7.12%	87.22	63.20	8.14%
Seq2Seqw/attn (128)	59.58	45.18	6.89%	73.5	59.12	7.75%	54.07	46.98	6.73%	81.15	56.95	7.49%
Seq2Seqw/attn (512)	50.18	39.74	5.71%	68.71	50.71	6.97%	45.51	37.33	5.49%	65.3	46.72	6.4%
BuildSenSys (128)	46.89	33.35	4.85%	60.18	41.89	4.36%	36.42	27.14	5.25%	50.45	38.56	5.2%
***BuildSenSys* (512)**	**33.08**	**22.78**	**2.05%**	**58.33**	**39.71**	**3.64%**	**30.49**	**20.29**	**2.51%**	**35.10**	**25.75**	**2.22%**

following reasons. Traffic volumes have daily patterns and weekly patterns that are relatively stable in our dataset. Nevertheless, the HA method can produce unsatisfied prediction results with large errors in holidays, extreme weather conditions, and social events, because it cannot predict traffic volume that does not follow regular patterns.

Second, the Recurrent Neural Network-based methods, *i.e.*, LSTM, Seq2Seq, and Seq2Seqw/attn, show superior performance with MAEs lower than 70 and RMSEs lower than 90. In specific, the performance of LSTM and Seq2Seq are competitive, while Seq2Seq outperforms the LSTM by 2.35% and 2.43% in MAPE metric on the Road D and Road A, respectively. As for the Seq2Seq model based on the encoder-decoder architecture, it encodes the input traffic volume data into a feature vector, from which the decoder generates the fixed-length prediction iteratively. Nevertheless, the performance of encoder-decoder networks will deteriorate rapidly when the input sequence data becomes longer. Both *BuildSenSys* and Seq2Seqw/attn employ temporal attention mechanisms. Hence, the decoders of the above models can select the most relevant encoder hidden states adaptively to improve prediction accuracy. The effectiveness of attention mechanism is validated by the more accurate prediction results by Seq2Seqw/attn and *BuildSenSys*. With 512 hidden states, both Seq2Seqw/attn and *BuildSenSys* greatly improve the prediction accuracy in comparison with models that have 128 hidden states. More importantly, the prediction accuracy of *BuildSenSys* is further enhanced by the cross-domain attention that learns building-traffic correlations. In comparison with Seq2Seqw/attn, *BuildSenSys* shows 32.3% and 37.8% improvements in MAE and RMSE, respectively. *In summary, compared with RNN-based baseline methods, BuildSenSys jointly leverages cross-domain attention and temporal attention to learn cross-domain, time-varying, and non-linear building-traffic correlations adaptively. As a result, BuildSenSys outperforms all baseline methods with up to 65.3% accuracy improvement (e.g., 2.2% MAPE) in predicting nearby traffic volume.*



(a) RMSE (b) MAE (c) MAPE

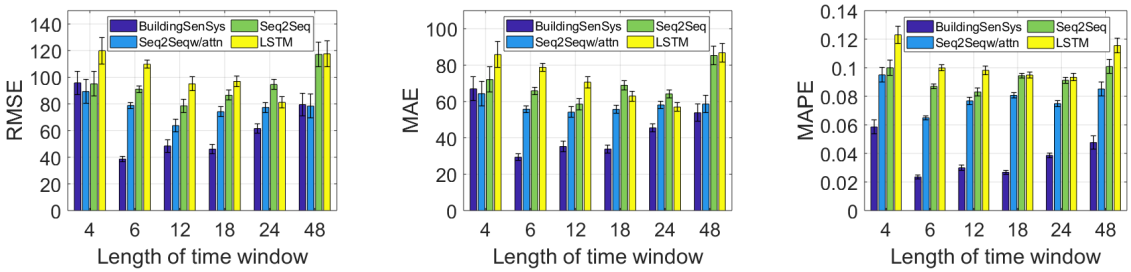
Figure 6.10 : Impact of the hidden states on the performance of *BuildSenSys* and three RNN-based baseline methods.

Evaluation of Parameters

We further evaluate the impact of parameters in RNN networks on *BuildSenSys*'s performance, since *BuildSenSys* is based on RNN. The parameters include the size of the hidden states in the encoder (h_a) and decoder (h_b), and the length of the input time window (L) and the prediction window (τ). Moreover, three RNN-based baseline methods, *i.e.*, LSTM, Seq2Seq, and Seq2Seqw/attn, are used as the comprised benchmarks.

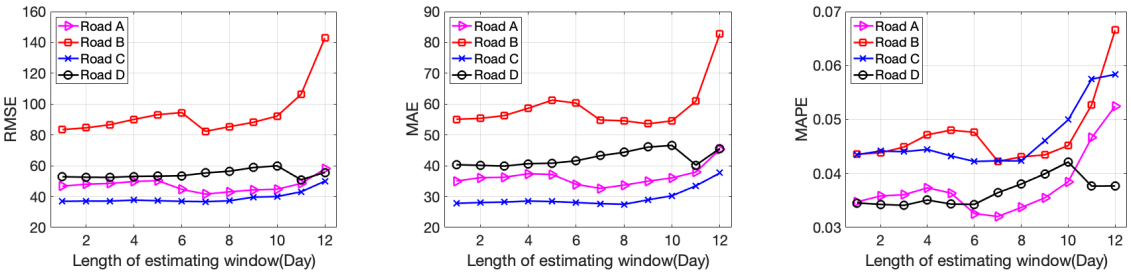
1) **Size of hidden states**: Following [181], we set the size of hidden states in both encoder and decoder to be the same, *i.e.*, $h_a = h_b$. We change the value of h_a (h_b) from 32 to 64, 128, 256, and 512 with a grid search. As shown in Figs. 6.10a, 6.10b, and 6.10c, *BuildSenSys* outperforms LSTM, Seq2Seq, and Seq2Seqw/attn in all settings of hidden states in terms of RMSE, MAE, and MAPE. Moreover, the experimental results show that the prediction accuracy of *BuildSenSys* roughly improves with the larger size of hidden states. Overall, *BuildSenSys* achieves the best prediction results (*i.e.*, RMSE $\leq$ 33, MAE $\leq$ 25 and MAPE $\leq$ 0.22) when $h_a = h_b = 512$.

2) **Length of input time window**: The time window L denotes how many hours of historical data are fed into the traffic prediction models (1 hour as the basic unit). In order to evaluate the impact of L, we change its value from 4 to 6, 12, 18, 24, and 48 hours. Thereby, the historical data of the last 4, 6, 12,



(a) RMSE (b) MAE (c) MAPE

Figure 6.11 : Impact of the input time window on the performance of *BuildSenSys* and three RNN-based baseline methods.



(a) Root Mean Squared Error (b) Mean Average Error (c) Mean Absolute Percentage Error

Figure 6.12 : Prediction accuracy of *BuildSenSys* on four roads by varying different lengths of predicting window.

18, 24, and 48 hours will be used to predict traffic volume in the next hour. As illustrated in Figs. 6.11a, 6.11b, and 6.11c, the prediction accuracies of all RNN-based methods increase with the greater length of the time window. Meanwhile, the performance of RNN-based methods also decreases when the time window is too large, *e.g.*, $L = 48$. Accordingly, *BuildSenSys* achieves the best performance, *i.e.*, RMSE= 31.7, MAE= 24, and MAPE= 0.019 when $L = 6$.

3) **Length of prediction window:** We evaluate the prediction accuracy of *BuildSenSys* in terms of the length of predicting window τ. We change τ from 1 day to 12 days. From Figs. 6.12a, 6.12b, and 6.12c, we observe that the RMSE, MAE, and MAPE results increase with the greater length of predicting windows, *i.e.*, the prediction accuracy decreases with the longer predicting windows. However, except for Road B, the decreasing speed of prediction accuracy is very slow. Therefore,

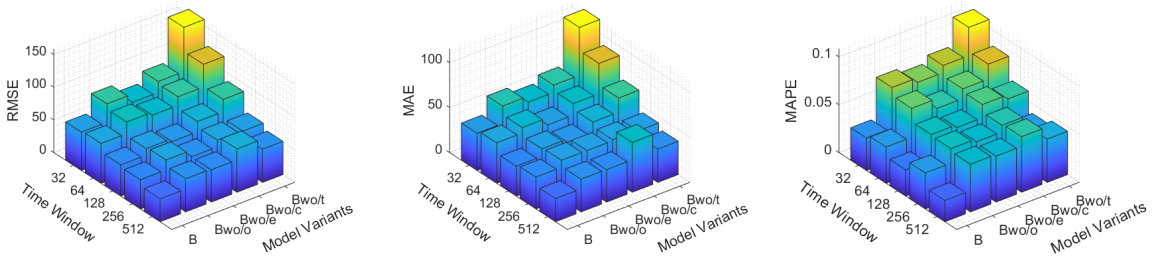
BuildSenSys can achieve a stable prediction with high accuracy by reusing building data, *e.g.*, RMSE and MAE increase from 36 to 52 and 28 to 30, respectively. Meanwhile, as Road B is the furthest road from the building, its building-traffic correlation is not as strong as other roads. Therefore, the prediction errors on Road B increase significantly with the increasing value of τ.

Evaluation of Cross-domain Learning

BuildSenSys incorporates a cross-domain attention mechanism and a temporal attention mechanism to jointly extract spatio-temporal features from building-traffic correlations. To evaluate the impact of each component on the overall performance, we conduct an ablation study on *BuildSenSys* with its variants as follows.

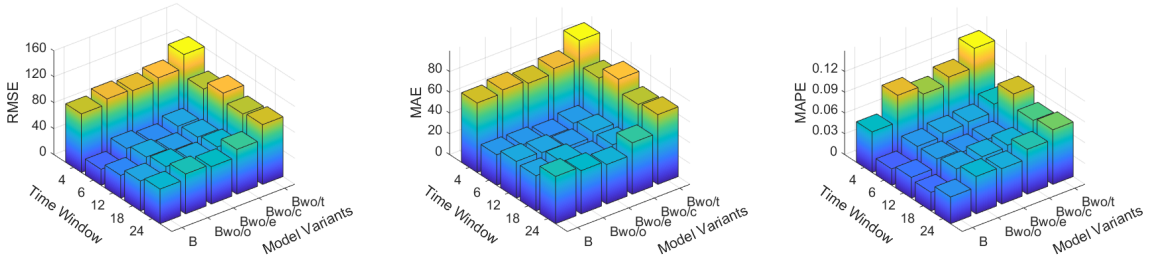
- **B**, *i.e.*, *BuildSenSys*, leverages cross-domain attention and temporal attention to jointly learn building-traffic correlations and temporal correlations with both occupancy data and environmental data.
- **Bwo/o**, a variant of **B** without **o**ccupancy component.
- **Bwo/e**, a variant of **B** without **e**nvironmental component.
- **Bwo/c**, a variant of **B** without a **c**ross-domain attention, which only employs a temporal attention to learn temporal dependencies of building sensing data and traffic volume data.
- **Bwo/t**, a variant of **B** without a **t**emporal attention, which only employs a cross-domain attention to learn cross-domain building-traffic correlations.

First, we evaluate the performance of different components of *BuildSenSys* by changing the size of hidden states. As shown in Fig. 6.13, *BuildSenSys* demonstrates the best performance, while Bwo/t shows the worst performance as it lacks temporal correlations for traffic prediction. Meanwhile, the prediction accuracy of both Bwo/o and Bwo/e is better than Bwo/c. It indicates that cross-domain learning with building data can effectively improve overall performance. Moreover, as illustrated



(a) Root Mean Squared Error (b) Mean Average Error (c) Mean Absolute Percentage Error

Figure 6.13 : Performance comparison among different variants of *BuildSenSys* with varying the size of hidden states.



(a) Root Mean Squared Error (b) Mean Average Error (c) Mean Absolute Percentage Error

Figure 6.14 : Performance comparison among different variants of *BuildSenSys* with varying the length of time window (in hours).

in Figs. 6.13a, 6.13b. and 6.13c, the prediction accuracy of Bwo/e is higher than that of Bwo/o. This result shows that the building occupancy data has more contribution to cross-domain traffic prediction than building environmental data.

Second, we evaluate the performance of different components of *BuildSenSys* by changing the length of the input time window. As shown in Fig. 6.14, the performance of Bwo/t deteriorates significantly as it has no temporal attention to capture the temporal correlations in building data and traffic data. As a result, the prediction accuracy of Bwo/t is the worst, *e.g.*, the RMSE over 115, MAE over 80, and MAPE over 10%, respectively. Meanwhile, *BuildSenSys* shows the best prediction accuracy when the input time window is 6. However, when the length of the input time window increases to 12, 18, and 24, the prediction accuracy of

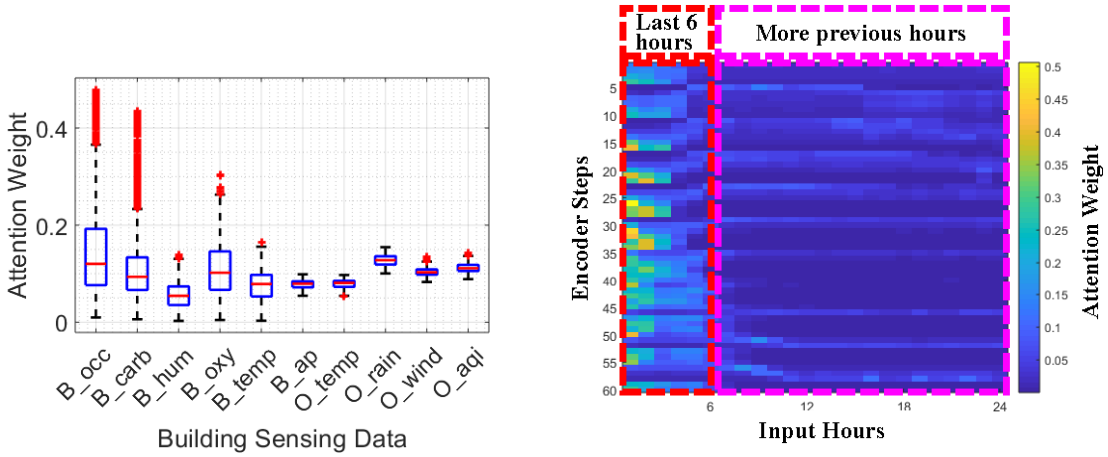
all variants decreases continuously. This can be explained as the longer inputs could pose greater difficulty in capturing building-traffic correlations and temporal correlations.

To sum up, the above results indicate that each component in cross-domain learning has its contribution to enhancing the prediction accuracy of *BuildSenSys*. Specifically, the occupancy component and the environmental component are complementary to each other in reusing building data to predict nearby traffic volume. Moreover, it is highly essential to jointly leverage cross-domain attention and temporal attention in cross-domain learning of building data. Each attention mechanism has its unique contribution to the improvement of prediction accuracy, *e.g.*, up to 45.5% improvement by temporal attention and 30.9% improvement by cross-domain attention.

Evaluation of Attention Weight

1) **Cross-domain attention**: We analyse the attention weights of the proposed traffic prediction model in two folds, *i.e.*, cross-domain attention weight and temporal attention weight. First, as the correlations between traffic volume data and different building sensing data are non-linear, *BuildSenSys* employs a cross-domain attention mechanism to achieve the non-linear mapping from building sensing data to traffic data in predicting traffic volume. To demonstrate the different attention weights from different types of building sensing data, we conduct extensive experiments to analyse the attention weight of each type of building sensing data. As shown in Fig. 6.15a, the 10 groups of attention weights correspond to 10 types of sensing data, including building occupancy (B_occ), CO_2 concentration (B_carb), building humidity (B_hum), O_2 concentration (B_oxy), building temperature (B_temp), building air pollution (B_ap), as well as outdoor temperature (O_temp), rainfall (O_rain), wind (O_wind) and air quality index (O_aqi). The box plot in Fig. 6.15a compares the distribution of attention weight by each type of building sensing data.

As demonstrated in Fig. 6.15a, each type of building sensing data has its own



(a) Cross-domain Attention Weight (b) Temporal Attention Weight

Figure 6.15 : Visualisation of Cross-domain Attention and Temporal Attention.

contribution to traffic prediction, while building occupancy data has the highest attention weights across all predicting steps. Specifically, the whiskers (*i.e.*, lines extending above and below each box) of B_occ have the most extensive range (from 0.01 to 0.36). B_occ covers the furthest adjacent values of attention weight, showing the significant importance of building occupancy data in cross-domain traffic prediction. Moreover, the B_occ and B_carb have the largest number of outliers with the highest values (*i.e.*, up to 0.48 and 0.39, respectively). It indicates that they have stronger correlations with traffic data and higher attention weights in traffic prediction. Finally, the median values (*i.e.*, the central marks) of attention weights boxes are roughly close, illustrating that all types of building data make non-negligible contributions in traffic predictions.

In summary, the above experimental results indicate that each building data has its own contribution to enhancing the prediction accuracy of *BuildSenSys*. Meanwhile, the building occupancy data and the environmental data are complementary to each other in reusing building data for traffic sensing and prediction.

2) **Temporal attention**: We further investigate how the temporal attention mechanism captures the correlations between predicted traffic values and historical traffic data. More specifically, we conduct the exploring experiment to visualise how

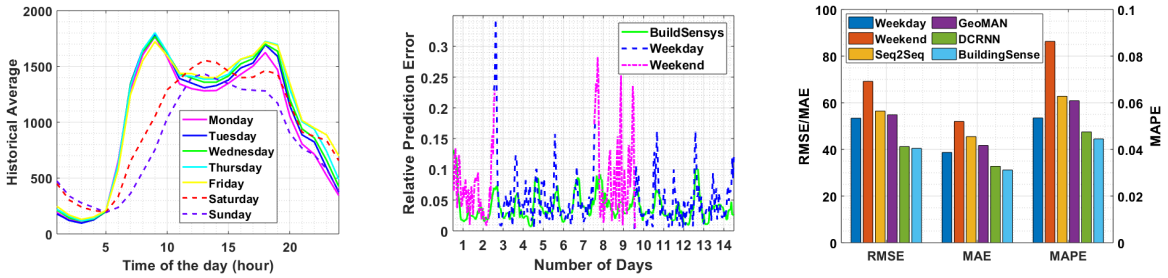
BuildSenSys assigns different attention weights to the historical traffic data in the time window 'L'. As explained in the experimental setups, the input time window 'L' represents the length of input data measured in hours, and we set 'L' as 24 (hours) in the following experiment.

As a result, the heat map in Fig. 6.15b visualises the temporal attention weights over 24 hours and 60 predicting steps. We divide the input time window into two categories, *i.e.*, the latest 6 hours and more previous (the last 6-24) hours. The above two categories have great disparities in assigned attention weights. For the input data of the latest 6 hours, *BuildSenSys* assigns more attention weight to them, as each hour's data has up to 0.5 attention weight.

In contrast, most input data in the last 6-24 hours have particularly small values of attention weights (under 0.05). The above result shows that the temporal attention mechanism in *BuildSenSys* would focus more on the latest 6 hours of the input data. Meanwhile, as all of the temporal attention weights sum to 1, with the longer input time windows (*e.g.*, 12, 18, 24 and 48 in our ablation studies), the latest 6 hours will share lower attention weights. In such cases, *BuildSenSys* will not be able to specifically focus on the most recent data but to process both short-term and long-term temporal correlations for traffic volume prediction. Consequently, the prediction accuracy of *BuildSenSys* will be influenced, as demonstrated and verified by the evaluation results in previous experimental studies.

Extensive evaluations of comparison

Baseline methods with the weekday's data and the weekend's data: To further explore the impact of data patterns on the traffic prediction results, we conduct an extensive experiment by applying two different models to predict traffic volumes on weekdays and weekends, respectively. First, we visualise a distinct daily average of traffic data on a target road in Fig.6.16a, where different colours show the average values of one-year traffic data on each day of the week. We can observe that the traffic volumes on weekdays and weekends follow two different patterns, respectively. For weekdays, the peak hours range from 8 a.m. to 10 a.m. and 4



(a) Distinct average daily traffic. (b) Relative prediction error by different models. (c) Comparison with state-of-the-art models.

Figure 6.16 : Performance comparison with the Weekday model and the Weekend model.

p.m. to 7 p.m. Meanwhile, for weekends, there is a longer range for peak hours, i.e., 12 a.m. to 6 p.m. Our goal is to study whether using two different models of *BuildSenSys* to predict traffic volume on weekdays and weekends respectively would impact the results of the overall results in traffic volume prediction.

In this experimental study, the basic settings are the same as previous experiments. The only difference is that we divide one-year traffic data into two groups of data (weekdays and weekends), and correspondingly we train two models (weekday model and weekend model) of BuildSenSys for each group. At the same time, we train a BuildSenSys model based on the whole dataset, and the total epochs for all models are set as 2,500. The prediction results of one-model prediction and two-model prediction are shown in Fig. 6.16b and Fig. 6.16c, where the relative prediction error is the ratio of the prediction error to the ground truth. Weekday represents a special model of *BuildSenSys* that is trained only with weekday data to predict the traffic volume on weekdays. Similarly, Weekend stands for another special model of *BuildSenSys* that is trained only with weekend data to predict the traffic volume on weekends. As illustrated in Fig. 6.16b and Fig. 6.16c, we have compared the prediction accuracy of the Weekday model and the Weekend model, Seq2Seq model, and *BuildSenSys* model. The results show that in comparison with the Seq2Seq model, both the Weekday model and the Weekend model can achieve some improvements in prediction accuracy. However, the combined prediction results are still not

satisfactory as with even lower accuracy than the original model. Meanwhile, by using a temporal attention mechanism, the proposed *BuildSenSys* can dynamically learn the impact of historical data on the predicting target, thereby it outperforms both the Weekday model and the Weekend model. The experimental results also validate that breaking the continuity of the dataset in the temporal dimension will compromise the performance of temporal attention mechanisms.

Comparison with state-of-the-art methods based on different data sources: To address how different data measurements would impact the traffic prediction results, we further employ three baseline methods for comparison. Among these baseline methods, the Sequence to Sequence model (Seq2Seq) [178] uses traffic volume data from the traffic sensing system on one road for prediction; the Multi-level Attention Networks for Geospatial Sensors (GeoMAN) [181] exploits the data of one road along with the weather data; and the Diffusion Convolutional Recurrent Neural Network (DCRNN) [182] leverages the sensing data from multiple roads with the road graph. In this work, BuildSenSys leverages building sensing data to predict the traffic volume on nearby roads. The experimental results in Fig. 6.16c show that BuildSenSys achieves the best performance, even compared with DCRNN, which uses traffic data from 10 road segments to predict traffic volume on the target road. GeoMAN outperforms Seq2Seq, as it considers both spatial and temporal attention with weather data and historical road data for traffic prediction. Meanwhile, DCRNN further improves the prediction accuracy by using both diffusion convolution and sequence to sequence learning framework on traffic volume data of multiple roads with sensor topology information. The above results also assess the practicability and usefulness of cross-domain traffic prediction by re-using building sensing data.

6.7 Discussion

In this work, we explore the possibility of reusing building sensing data to predict the traffic volume of nearby road segments. Although we have made some advances toward this new direction, there are still several issues that need to be further

investigated as follows.

Applicable conditions of *BuildSenSys*: It is of great importance to discuss: 1) which kind of buildings are suitable for accurate traffic sensing and prediction, and 2) what kind of real-world applications could benefit from cross-domain traffic prediction by reusing building sensing data. Based on our initial studies, we roughly summarise three critical factors of a building for cross-domain traffic prediction, including location, capacity, and building types. First, human movement patterns (including traffic) among urban sites are affected by buildings, which can 'temporarily hold' human mobility [136]. Therefore, buildings located close to road networks are more feasible for traffic sensing and prediction, as they would have more chances to affect human mobility on the road [136]. Second, the capacity of a building, *i.e.*, the volume of occupants it can hold, is essential for accurate cross-domain traffic sensing. For instance, Zheng et al. [138] showed a commercial building with a capacity of 10000 occupants can effectually impact the traffic on nearby roads with substantial evidence. Third, the types of buildings can affect the applicable conditions of *BuildSenSys*. Different types of buildings have different patterns of occupancy dynamics, which will directly affect building-traffic correlations. To this end, we recommend commercial building (*e.g.*, office buildings and retail buildings) are the first choice when performing cross-domain traffic sensing with *BuildSenSys*. In practice, *BuildSenSys* can be used in a variety of real-world applications. For example, it can be exploited to provide real-time traffic volume data to support controls of intelligent traffic light [183, 184]. Besides, *BuildSenSys* can also be applied in traffic management systems, which requires accurate traffic predictions, especially in rush hours [185].

Extension to large-scale scenarios: According to the investigation of sensing coverage, the *BuildSenSys* system can predict traffic volume for road segments within 5 km of the building. To extend this coverage of traffic predictions to a larger scale, it requires to incorporate more sensing data that are correlated with traffic volume. Intuitively, traffic data of road segments in the same district would have spatial and temporal correlations [153]. For example, Liu et al. [186] probed traffic conditions

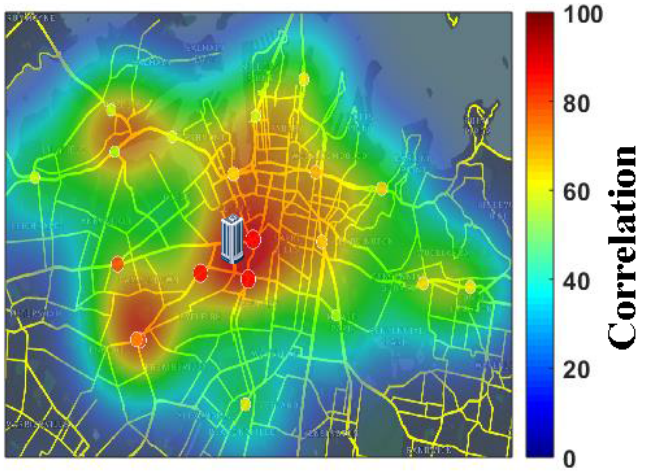


Figure 6.17 : Illustrations of prediction coverage by *BuildSenSys*.

on different road segments with GPS data from bus riders. In our future work, we will further exploit the above multi-road and multi-region correlations to extend the coverage of *BuildSenSys* in traffic sensing and prediction. Besides, the pervasive street cameras are the cost-efficient data source for aggregating urban crowd flow, which may have direct and indirect correlations with traffic volume on nearby roads. Existing works in computer vision have proposed highly effective approaches to generate density maps [187] and achieve accurate crowd counting [188], which have the potential to contribute to cross-domain traffic sensing and prediction.

Sensing coverage of *BuildSenSys*: In this chapter, *BuildSenSys* is a proof-of-concept in reusing building data for traffic sensing. As shown in the experimental evaluations of Section 6.6, the prediction accuracy of *BuildSenSys* is in accordance with the building-traffic correlation of a road. Thus, we further conduct experiments to investigate the coverage of cross-domain traffic prediction by reusing building sensing data. In specific, as illustrated in Fig. 6.17, we calculate the correlations between building occupancy and traffic data on nearby roads using the Pearson Correlation Coefficient, and further use the building-traffic correlations as the index of sensing coverage of *BuildSenSys*. Fig. 6.17 demonstrates that BuildSenSys generally has a prediction coverage within 5 km to the building.

Privacy and security of building data: With the massive amount of IoT

data generated by smart buildings, it is essential to reuse building data in a privacy-preserving manner. In our work, the building datasets, including occupancy data and environmental data, are anonymised by the data provider. As a consequence, they cannot be used to trace back to any personal information. Notably, the people counters (cameras) do not have facial recognition capability, and there would be no personal privacy leakage from the building occupancy data. Moreover, if needed, we can adapt existing privacy protection methods [189, 190] to prevent privacy leakage when utilising building sensing data.

6.8 Conclusion

In this chapter, we have proposed *BuildSenSys*, a first-of-its-kind building sensing data-based traffic volume prediction system with cross-domain learning. Firstly, we have conducted extensive experimental analysis on building-traffic correlations based on multi-source real-world datasets. It has been disclosed that building data has strong correlations with traffic data. Then, we have proposed a cross-domain learning-based RNN with cross-domain and temporal attention mechanisms to jointly extract building-traffic correlations for accurate traffic prediction. Moreover, we have implemented a prototype system of *BuildSenSys* and conducted extensive experiments. The experimental results have demonstrated that *BuildSenSys* outperforms seven baseline methods with up to 65.3% accuracy improvement in predicting nearby traffic volume. We believe this work can open a new gate to reuse building sensing data for traffic sensing and prediction, hence indicating an interconnection between smart buildings and intelligent transportation systems.

Chapter 7

Conclusion and Future Work

In this thesis, we have presented four works contributing to the development of mobile computing, including mechanism design for edge computation offloading and two application implementations for mobile edge sensing. In Chapter 1, we have presented an overview of mobile edge computing, covering the early theoretical investigation in the early 2010s, the more recent attempts to develop mobile edge computing applications, and the latest breakthroughs of deep learning techniques with mobile edge computing. Secondly, we have provided our insights into some key issues in mobile edge computing and discussed the promising solutions based on different methodologies. In Chapter 2, we have reviewed the existing works of mobile edge computing from both theoretical aspects and the practical perspectives. After that, we have presented four of our representative works in mobile edge computing from Chapter 3 to Chapter 6, respectively. In this Chapter, we conclude the technical contributions of this thesis as follows.

7.1 Computation Offloading Mechanisms for Mobile Cloudlet Networks

In Chapter 3, we have investigated the load balancing problem in mobile cloudlet networks. By leveraging balls-and-bins theory, we have devised 'CTOM', a collaborative task offloading scheme for mobile cloudlet networks. By locally querying limited task load information, the proposed solution can reduce the longest task queue in the allocation process effectively. The simulation and trace-driven evaluation results have demonstrated that CTOM performs exceedingly close to the optimal solution in load balancing and outperforms the conventional random and proportional allocation schemes by 65% and 55% in task gaps, respectively. In

contrast, its computing complexity has been largely reduced at each time interval.

In Chapter 4, we have proposed FairEdge, a Fairness-oriented task offloading scheme to enable balanced task sharing and computation offloading for mobile Edge cloudlet networks. The FairEdge integrates balls-into-bins theory and Jain's fairness index for distributed task offloading among mobile cloudlets. We have developed the system model of computation offloading and formulated the load balancing problem with an objective of fairness optimisation. By adopting the 'two-choice' paradigm and using calculated fairness index values of cloudlets and the network, we have further presented algorithm design of FairEdge and conducted extensive evaluation studies by simulations and experiments on different real-world trace datasets. The experimental results have shown that FairEdge successfully achieves load balancing with guaranteed performance of a near-optimal fairness index of up to 0.85 and an improvement of 50% over conventional baseline methods.

7.2 Edge Computing Implementations: Applications and Systems

In Chapter 5, we have investigated to build accurate and complete RSS maps with raw crowdsensing data collected by heterogeneous mobile edge devices. We have developed an innovative system iMap for mobile users to crowdsense RSS signals of outdoor wireless access points. We have further tested the system with different types of smartphones and observed the collected RSS measurements with model-based analysis. To construct accurate and complete RSS maps, we have devised a compressive sensing-based algorithm to recover RSS data with the adaptive sampling. The experimental results have shown that the proposed method can achieve accurate and complete RSS recovery with partial RSS data. The recovery rates are up to 90% and 95% in the geographic coordinate system and the polar coordinate system, respectively.

In Chapter 6, we have proposed *BuildSenSys*, a first-of-its-kind building sensing data-based traffic volume prediction system with cross-domain learning. Firstly, we have conducted extensive experimental analysis on building-traffic correlations

based on multi-source real-world datasets. It discloses that building data has strong correlations with traffic data. Then, we have proposed a cross-domain learning-based RNN with cross-domain and temporal attention mechanisms to jointly extract building-traffic correlations for accurate traffic prediction. Moreover, we have implemented a prototype system of *BuildSenSys* and conducted extensive experiments. The experimental results demonstrated that *BuildSenSys* outperforms seven baseline methods with up to 65.3% accuracy improvement in predicting nearby traffic volume. We believe this work can open a new gate to reuse building sensing data for traffic sensing and prediction, hence indicating an interconnection between smart buildings and intelligent transportation.

7.3 Future Work

We envision that the emerging MEC-based applications will prosper continuously and become tremendously pervasive in people's daily life. Driven by the progressive development of revolutionary cutting-edge technologies (*e.g.*, 5G, V2X), mobile edge computing will further bring unprecedented opportunities to the research community. We will continue our research to explore theoretical and practical knowledge in data-driven mobile edge computing. Particularly, we will keep focusing on the following research directions to promote our ongoing research works:

First, data-driven service placement and request scheduling for computation-intensive applications mobile edge clouds. To serve computation-intensive applications from the edge, Farhadi et al. [63] have recently presented a comprehensive research work on joint service placement and request scheduling in edge computing. In particular, MEC allows flexible and configurable placement of services (*i.e.*, service provisioning) on edge servers, thereby relieving mobile users from experiencing large transmission delays for accessing remote central clouds via backbone networks. Thereby, a fundamental question of *service provisioning* has been raised, concerning which kind of services (*e.g.*, application codes and data) should be provisioned among heterogeneous resource-constrained MEC servers.

With the previous experience in computation offloading mechanism design for

edge cloudlets, we aim to further explore spatial and temporal service placement and request scheduling for edge clouds, based on data-driven predictions. The basic idea is that, if we could predict the popularity of application services that mobile users tend to request, we can place these service placement in advance and make the best of request scheduling for the entire edge cloud network. In this way, we can significantly improve the efficiency and sufficiency of mobile edge computing and further enhance user experience in mobile edge networks.

Second, deep learning frameworks for heterogeneous geospatial sensors in urban IoT applications. Deep neural networks have gained increasing attention and interest in mobile sensing and computing [56,191]. To further advance such development, we aim to propose a deep learning framework for urban IoT sensors. The basic idea is to capture the cross-domain, spatiotemporal, and dynamic correlations among multiple types of urban sensory data, and further to use such dependencies for representation learning and model training. Furthermore, we will implement the deep learning framework into a prototype system and conduct the real-world experimental study with urban IoT applications.

Additionally, we propose to summarise the state-of-the-art works in deep learning and edge computing-empowered traffic prediction into a review paper. Our goal is to bring the latest development traffic prediction methodologies together and provide insights into mobile edge computing and deep learning in future Intelligent Transportation Systems.

We hope to collaborate with more researchers to conduct high-quality research on theories and real-world applications for mobile edge computing, and further use the cutting-edge deep learning models to solve emerging technical challenges in mobile edge computing. We believe that mobile edge computing will lead us towards a whole new world of IoT, V2X, and Smart City applications, making the world better, smarter, and a little more fun.

Bibliography

[1] X. Fan, X. He, D. Puthal, S. Chen, C. Xiang, P. Nanda, and X. Rao, "C-TOM: Collaborative task offloading mechanism for mobile cloudlet networks," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[2] S. Lai, X. Fan, Q. Ye, Z. Tan, Y. Zhang, X. He, and P. Nanda, "Fairedge: A fairness-oriented task offloading scheme for iot applications in mobile cloudlet networks," *IEEE Access*, vol. 8, pp. 13 516–13 526, 2020.

[3] X. Fan, X. He, C. Xiang, D. Puthal, L. Gong, P. Nanda, and G. Fang, "Towards system implementation and data analysis for crowdsensing based outdoor rss maps," *IEEE Access*, vol. 6, pp. 47 535–47 545, 2018.

[4] X. Fan, C. Xiang, C. Chen, P. Yang, L. Gong, X. Song, P. Nanda, and X. He, "Buildsensys: Reusing building sensing data for traffic prediction with cross-domain learning," *IEEE Transactions on Mobile Computing*, 2020.

[5] Y. Zhang, J. Zheng, L. Li, N. Liu, W. Jia, X. Fan, C. Xu, and X. He, "Rethinking feature aggregation for deep rgb-d salient object detection," *Neurocomputing*, vol. 423, pp. 463–473, 2021.

[6] C. Xiang, X. Fan, C. Chen, L. Gong, and S. Guo, "Fisher information-empowered sensing quality quantification for crowdsensing networks," *Neural Computing and Applications*, 2021.

[7] Y. Xi, W. Jia, J. Zheng, X. Fan, Y. Xie, J. Ren, and X. He, "Drl-gan: Dual-stream representation learning gan for low-resolution image classification in uav applications," *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, vol. 14, pp. 1705–1716, 2021.

[8] C. Xiang, P. Yang, X. Fan, and L. Gong, "Quantifying sensing quality of crowd sensing networks with confidence interval," in *2017 IEEE Ubiquitous Intelligence Computing*, 2017, pp. 1–6.

[9] Q. Li, P. Yang, X. Fan, S. Tang, C. Xiang, D. Guo, and F. Li, "Taming the big to small: efficient selfish task allocation in mobile crowdsourcing systems," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 14, p. e4121, 2017.

[10] X. Rao, Y. Yan, M. Zhang, W. Xu, X. Fan, H. Zhou, and P. Yang, "You can recharge with detouring: Optimizing placement for roadside wireless charger," *IEEE Access*, vol. 6, pp. 47–59, 2018.

[11] N. Yang, X. Fan, D. Puthal, X. He, P. Nanda, and S. Guo, "A novel collaborative task offloading scheme for secure and sustainable mobile cloudlet networks," *IEEE Access*, vol. 6, pp. 44 175–44 189, 2018.

[12] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, and G. Fang, "A novel convolutional neural network based indoor localization framework with wifi fingerprinting," *IEEE Access*, vol. 7, pp. 110 698–110 709, 2019.

[13] X. Song, X. Fan, X. He, C. Xiang, Q. Ye, X. Huang, G. Fang, L. L. Chen, J. Qin, and Z. Wang, "Cnnloc: Deep-learning based indoor localization with wifi fingerprinting," in *2019 IEEE Ubiquitous Intelligence and Computing(UIC)*, 2020.

[14] X. Fan, C. Xiang, L. Gong, X. He, C. Chen, and X. Huang, "Urbanedge: Deep learning empowered edge computing for urban iot time series prediction," in *Proceedings of the ACM Turing Celebration Conference - China*, ser. ACM TURC '19, 2019.

[15] X. Fan, C. Xiang, L. Gong, X. He, and X. He, "Deep learning for intelligent traffic sensing and prediction: recent advances and future challenges," *CCF Transactions on Pervasive Computing and Interaction*, pp. 1–21, 2020.

[16] J. Li, M. Liu, Z. Xue, X. Fan, and X. He, "Rtvd: A real-time volumetric detection scheme for ddos in the internet of things," *IEEE Access*, vol. 8, pp. 36 191–36 201, 2020.

[17] C. Xiang, Z. Zhang, Y. Qu, D. Lu, X. Fan, P. Yang, and F. Wu, "Edge computing-empowered large-scale traffic data recovery leveraging low-rank theory," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 4, pp. 2205–2218, 2020.

[18] L. Gong, C. Xiang, X. Fan, T. Wu, and W. Yang, "Device-free near-field human sensing using wifi signals," *Personal and Ubiquitous Computing*, no. 10, pp. 1–14, 2020.

[19] Q. Ye, X. Fan, G. Fang, H. Bie, X. Song, and R. Shankaran, "Capsloc: A robust indoor localization system with wifi fingerprinting using capsule networks," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*. IEEE, 2020, pp. 1–6.

[20] A.-K. Pietilainen and C. Diot, "Social pocket switched networks," in *IEEE INFOCOM Workshops 2009*. IEEE, 2009, pp. 1–2.

[21] J. Kunegis, "Haggle network dataset konect," http://konect.uni-koblenz.de/networks/contact, 2017, accessed: December, 1, 2019.

[22] C. V. N. Index, "Cisco visual networking index: global mobile data traffic forecast update, 2017–2022," *Cisco, San Jose, CA, USA, Tech. Rep*, 2019.

[23] Y. Mao, C. You, J. Zhang, K. Huang, and K. B. Letaief, "A survey on mobile edge computing: The communication perspective," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2322–2358, 2017.

[24] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computinga key technology towards 5g," *ETSI White Paper*, vol. 11, no. 11, pp. 1—16, 2015.

[25] W. Shi, G. Pallis, and Z. Xu, "Edge computing [scanning the issue]," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1474–1481, 2019.

[26] P. Mach and Z. Becvar, "Mobile edge computing: A survey on architecture and computation offloading," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 3, pp. 1628–1656, 2017.

[27] M. Satyanarayanan, V. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, 2009.

[28] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*, 2012, pp. 13–16.

[29] I. Nokia, "Increasing mobile operators value proposition with edge computing," *Technical Brief*, 2013.

[30] A. J. Ferrer, J. M. Marquès, and J. Jorba, "Towards the decentralised cloud: Survey on approaches and challenges for mobile, ad hoc, and edge computing," *ACM Computing Surveys (CSUR)*, vol. 51, no. 6, p. 111, 2019.

[31] G. I. Klas, "Fog computing and mobile edge cloud gain momentum open fog consortium, etsi mec and cloudlets," *Google Scholar*, 2015.

[32] A. W. Services, "Amazon web services greengrass," Retrieved from https://aws.amazon.com/greengrass/, 2019.

[33] M. Azure, "Azure iot edge," Retrieved from https://azure.microsoft.com/en-us/campaigns/iot-edge/, 2019.

[34] B. Luo, S. Tan, Z. Yu, and W. Shi, "Edgebox: Live edge video analytics for near real-time event detection," in *2018 IEEE/ACM Symposium on Edge Computing (SEC)*. IEEE, 2018, pp. 347–348.

[35] D. Zhang, T. Rashid, X. Li, N. Vance, and D. Wang, "Heteroedge: taming the heterogeneity of edge computing system in social sensing," in *Proceedings*

of the International Conference on Internet of Things Design and Implementation, 2019, pp. 37–48.

[36] S. Nunna, A. Kousaridas, M. Ibrahim, M. Dillinger, C. Thuemmler, H. Feussner, and A. Schneider, "Enabling real-time context-aware collaboration through 5g and mobile edge computing," in *2015 12th International Conference on Information Technology-New Generations*. IEEE, 2015, pp. 601–605.

[37] J. Xu, K. Ota, and M. Dong, "Saving energy on the edge: In-memory caching for multi-tier heterogeneous networks," *IEEE Communications Magazine*, vol. 56, no. 5, pp. 102–107, 2018.

[38] M. Zeng, T.-H. Lin, M. Chen, H. Yan, J. Huang, J. Wu, and Y. Li, "Temporal-spatial mobile application usage understanding and popularity prediction for edge caching," *IEEE Wireless Communications*, vol. 25, no. 3, pp. 36–42, 2018.

[39] M. Satyanarayanan, "The emergence of edge computing," *Computer*, vol. 50, no. 1, pp. 30–39, 2017.

[40] T. Soyata, R. Muraleedharan, C. Funai, M. Kwon, and W. Heinzelman, "Cloud-vision: Real-time face recognition using a mobile-cloudlet-cloud acceleration architecture," in *2012 IEEE symposium on computers and communications (ISCC)*. IEEE, 2012, pp. 000 059–000 066.

[41] Y. Li and W. Wang, "Can mobile cloudlets support mobile applications?" in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*. IEEE, 2014, pp. 1060–1068.

[42] Y. Zhang, D. Niyato, and P. Wang, "Offloading in mobile cloudlet systems with intermittent connectivity," *IEEE Transactions on Mobile Computing*, vol. 14, no. 12, pp. 2516–2529, 2015.

[43] M. Chen, Y. Hao, Y. Li, C.-F. Lai, and D. Wu, "On the computation offloading at ad hoc cloudlet: architecture and service modes," *IEEE Communications Magazine*, vol. 53, no. 6, pp. 18–24, 2015.

[44] M. Chen, Y. Hao, C.-F. Lai, D. Wu, Y. Li, and K. Hwang, "Opportunistic task scheduling over co-located clouds in mobile environment," *IEEE Transactions on Services Computing*, vol. 11, no. 3, pp. 549–561, 2016.

[45] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, and S. Chen, "Vehicular fog computing: A viewpoint of vehicles as the infrastructures," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 6, pp. 3860–3873, 2016.

[46] K. Wang, H. Yin, W. Quan, and G. Min, "Enabling collaborative edge computing for software defined vehicular networks," *IEEE Network*, vol. 32, no. 5, pp. 112–117, 2018.

[47] C. Wang, Y. Li, D. Jin, and S. Chen, "On the serviceability of mobile vehicular cloudlets in a large-scale urban environment," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 10, pp. 2960–2970, 2016.

[48] G. Qiao, S. Leng, K. Zhang, and Y. He, "Collaborative task offloading in vehicular edge multi-access networks," *IEEE Communications Magazine*, vol. 56, no. 8, pp. 48–54, 2018.

[49] M. Jia, W. Liang, Z. Xu, and M. Huang, "Cloudlet load balancing in wireless metropolitan area networks," in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*, vol. 27, no. 10. IEEE, 2016, pp. 1–9.

[50] M. Jia, W. Liang, Z. Xu, M. Huang, and Y. Ma, "Qos-aware cloudlet load balancing in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, vol. Early Access, pp. 1–1, 2018.

[51] D. Yao, L. Gui, F. Hou, F. Sun, D. Mo, and H. Shan, "Load balancing oriented computation offloading in mobile cloudlet," in *2017 IEEE 86th Vehicular Technology Conference (VTC-Fall)*. IEEE, 2017, pp. 1–6.

[52] J. Zhang, H. Guo, J. Liu, and Y. Zhang, "Task offloading in vehicular edge computing networks: A load-balancing solution," *IEEE Transactions on Vehicular Technology*, 2019.

[53] J. Zhao, Q. Li, Y. Gong, and K. Zhang, "Computation offloading and resource allocation for cloud assisted mobile edge computing in vehicular networks," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 8, pp. 7944–7956, 2019.

[54] Z. Zhou, H. Liao, B. Gu, K. M. S. Huq, S. Mumtaz, and J. Rodriguez, "Robust mobile crowd sensing: When deep learning meets edge computing," *IEEE Network*, vol. 32, no. 4, pp. 54–60, 2018.

[55] Sensorly, "Unbiased wireless network information," Retrieved from https://http://www.sensorly.com/, 2017.

[56] S. Yao, S. Hu, Y. Zhao, A. Zhang, and T. Abdelzaher, "Deepsense: A unified deep learning framework for time-series mobile sensing data processing," in *Proceedings of the 26th International Conference on World Wide Web*. International World Wide Web Conferences Steering Committee, 2017, pp. 351–360.

[57] H. Li, K. Ota, and M. Dong, "Learning iot in edge: Deep learning for the internet of things with edge computing," *IEEE network*, vol. 32, no. 1, pp. 96–101, 2018.

[58] Z. Zhou, X. Chen, E. Li, L. Zeng, K. Luo, and J. Zhang, "Edge intelligence: Paving the last mile of artificial intelligence with edge computing," *Proceedings of the IEEE*, vol. 107, no. 8, pp. 1738–1762, 2019.

[59] K. Poularakis, J. Llorca, A. M. Tulino, I. Taylor, and L. Tassiulas, "Joint service placement and request routing in multi-cell mobile edge computing networks," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 10–18.

[60] S. Pasteris, S. Wang, M. Herbster, and T. He, "Service placement with provable guarantees in heterogeneous edge computing systems," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 514–522.

[61] B. Gao, Z. Zhou, F. Liu, and F. Xu, "Winning at the starting line: Joint network selection and service placement for mobile edge computing," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1459–1467.

[62] T. Ouyang, R. Li, X. Chen, Z. Zhou, and X. Tang, "Adaptive user-managed service placement for mobile edge computing: An online learning approach," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1468–1476.

[63] V. Farhadi, F. Mehmeti, T. He, T. La Porta, H. Khamfroush, S. Wang, and K. S. Chan, "Service placement and request scheduling for data-intensive applications in edge clouds," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1279–1287.

[64] X. Chen, L. Jiao, W. Li, and X. Fu, "Efficient multi-user computation offloading for mobile-edge cloud computing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 5, pp. 2795–2808, 2016.

[65] Z. Xu, W. Liang, W. Xu, M. Jia, and S. Guo, "Efficient algorithms for capacitated cloudlet placements," *IEEE Transactions on Parallel and Distributed Systems*, pp. 2866–2880, 2016.

[66] Y. Liu, M. J. Lee, and Y. Zheng, "Adaptive multi-resource allocation for cloudlet-based mobile cloud computing system," *IEEE Transactions on Mobile Computing*, vol. 15, no. 10, pp. 2398–2410, 2016.

[67] H. Cao and J. Cai, "Distributed multi-user computation offloading for cloudlet based mobile cloud computing: A game-theoretic machine learning approach," *IEEE Transactions on Vehicular Technology*, 2017.

[68] S. Jeong, O. Simeone, and J. Kang, "Mobile edge computing via a uav-mounted cloudlet: Optimization of bit allocation and path planning," *IEEE Transactions on Vehicular Technology*, 2017.

[69] M. Jia, J. Cao, and W. Liang, "Optimal cloudlet placement and user to cloudlet allocation in wireless metropolitan area networks," *IEEE Transactions on Cloud Computing*, vol. 5, no. 4, pp. 725–737, 2015.

[70] M. Mitzenmacher, "The power of two choices in randomized load balancing," *IEEE Transactions on Parallel and Distributed Systems*, vol. 12, no. 10, pp. 1094–1104, 2001.

[71] X. Sun and N. Ansari, "Green cloudlet network: A distributed green mobile cloud network," *IEEE Network*, vol. 31, no. 1, pp. 64–70, 2017.

[72] P.-U. Tournoux, J. Leguay, F. Benbadis, V. Conan, M. D. De Amorim, and J. Whitbeck, "The accordion phenomenon: Analysis, characterization, and impact on dtn routing," in *INFOCOM*. IEEE, 2009, pp. 1116–1124.

[73] O. Pearce, T. Gamblin, B. R. De Supinski, M. Schulz, and N. M. Amato, "Quantifying the effectiveness of load balance algorithms," in *Proceedings of the 26th ACM international conference on Supercomputing*. ACM, 2012, pp. 185–194.

[74] P. Berenbrink, T. Friedetzky, L. A. Goldberg, P. W. Goldberg, Z. Hu, and R. Martin, "Distributed selfish load balancing," *SIAM Journal on Computing*, vol. 37, no. 4, pp. 1163–1181, 2007.

[75] Y. Azar, A. Z. Broder, A. R. Karlin, and E. Upfal, "Balanced allocations," *SIAM journal on computing*, vol. 29, no. 1, pp. 180–200, 1999.

[76] N. Alon and J. H. Spencer, *The probabilistic method*. John Wiley & Sons, 2004.

[77] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of markov chains," *The annals of mathematical statistics*, vol. 41, no. 1, pp. 164–171, 1970.

[78] B. Vöcking, "How asymmetry helps load balancing," *Journal of the ACM (JACM)*, vol. 50, no. 4, pp. 568–589, 2003.

[79] M. Z. Nayyer, I. Raza, and S. A. Hussain, "A survey of cloudlet-based mobile augmentation approaches for resource optimization," *ACM Computing Surveys (CSUR)*, vol. 51, no. 5, p. 107, 2018.

[80] H. Gao, Y. Duan, L. Shao, and X. Sun, "Transformation-based processing of typed resources for multimedia sources in the IoT environment," *Wireless Networks*, vol. 25, no. 1, pp. 1–17, 2019.

[81] H. Gao, Y. Xu, Y. Yin, W. Zhang, R. Li, and X. Wang, "Context-aware QoS prediction with neural collaborative filtering for internet-of-things services," *IEEE Internet of Things Journal*, vol. Early Access, pp. 1–1, 2019.

[82] M. Mitzenmacher and E. Upfal, *Probability and computing: randomization and probabilistic techniques in algorithms and data analysis*. Cambridge university press, 2017.

[83] G. Zhang, F. Shen, Y. Yang, H. Qian, and W. Yao, "Fair task offloading among fog nodes in fog computing networks," in *2018 IEEE International Conference on Communications (ICC)*. IEEE, 2018, pp. 1–6.

[84] L. Gong, Y. Zhao, C. Xiang, Z. Li, C. Qian, and P. Yang, "Robust light-weight magnetic-based door event detection with smartphones," *IEEE Transactions on Mobile Computing*, vol. 18, no. 11, pp. 2631–2646, 2019.

[85] Y. Qu, S. Tang, C. Dong, P. Li, S. Guo, H. Dai, and F. Wu, "Posted pricing for chance constrained robust crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 19, no. 1, pp. 188–199, 2020.

[86] X. Sun and N. Ansari, "Latency aware workload offloading in the cloudlet network," *IEEE Communications Letters*, vol. 21, no. 7, pp. 1481–1484, 2017.

[87] H. Huang and S. Guo, "Service provisioning update scheme for mobile application users in a cloudlet network," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[88] H. Huang and G. Song, "Adaptive service provisioning for mobile edge cloud," *ZTE Communications*, vol. 15, no. 2, pp. 1–9, 2017.

[89] J. Du, L. Zhao, J. Feng, and X. Chu, "Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee," *IEEE Transactions on Communications*, vol. 66, no. 4, pp. 1594–1608, 2018.

[90] Z. Zhu, J. Peng, X. Gu, H. Li, K. Liu, Z. Zhou, and W. Liu, "Fair resource allocation for system throughput maximization in mobile edge computing," *IEEE Access*, vol. 6, pp. 5332–5340, 2018.

[91] E. Meskar and B. Liang, "Fair multi-resource allocation with external resource for mobile edge computing," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*. IEEE, 2018, pp. 184–189.

[92] P. Berenbrink, K. Khodamoradi, T. Sauerwald, and A. Stauffer, "Balls-into-bins with nearly optimal load distribution," in *Proceedings of the twenty-fifth annual ACM symposium on Parallelism in algorithms and architectures*. ACM, 2013, pp. 326–335.

[93] P. Berenbrink, T. Friedetzky, P. Kling, F. Mallmann-Trenn, L. Nagel, and C. Wastell, "Self-stabilizing balls & bins in batches: The power of leaky bins," in *Proceedings of the 2016 ACM Symposium on Principles of Distributed Computing*. ACM, 2016, pp. 83–92.

[94] Q. Li, P. Yang, S. Tang, M. Zhang, and X. Fan, "Equilibrium is priceless: selfish task allocation for mobile crowdsourcing network," *EURASIP Journal on Wireless Communications and Networking*, vol. 2016, no. 1, p. 166, 2016.

[95] K. Jerome, "Konect: the koblenz network collection," in *Proceedings of the 22nd International Conference on World Wide Web*. ACM, 2013, pp. 1343–1350.

[96] W. Gong, B. Zhang, and C. Li, "Task assignment in mobile crowdsensing: Present and future directions," *IEEE Network*, 2018.

[97] H. Li, K. Ota, M. Dong, and M. Guo, "Mobile crowdsensing in software defined opportunistic networks," *IEEE Communications Magazine*, vol. 55, no. 6, pp. 140–145, 2017.

[98] X. Wang, J. Zhang, X. Tian, X. Gan, Y. Guan, and X. Wang, "Crowdsensing-based consensus incident report for road traffic acquisition," *IEEE Transactions on Intelligent Transportation Systems*, 2017.

[99] L. Shao, C. Wang, L. Liu, and C. Jiang, "Rts: road topology-based scheme for traffic condition estimation via vehicular crowdsensing," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 3, 2017.

[100] C. Chen, D. Zhang, X. Ma, B. Guo, L. Wang, Y. Wang, and E. Sha, "Crowddeliver: planning city-wide package delivery paths leveraging the crowd of taxis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 6, pp. 1478–1496, 2017.

[101] C. Chen, S. Jiao, S. Zhang, W. Liu, L. Feng, and Y. Wang, "Tripimputor: Real-time imputing taxi trip purpose leveraging multi-sourced urban data," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 10, pp. 3292–3304, 2018.

[102] Z. Pan, H. Yu, C. Miao, and C. Leung, "Crowdsensing air quality with camera-enabled mobile devices." in *AAAI*, 2017, pp. 4728–4733.

[103] S. Yang, F. Wu, S. Tang, X. Gao, B. Yang, and G. Chen, "On designing data quality-aware truth estimation and surplus sharing method for mobile crowdsensing," *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 4, pp. 832–847, 2017.

[104] J. Wang, N. Tan, J. Luo, and S. J. Pan, "Woloc: Wifi-only outdoor localization using crowdsensed hotspot labels," in *Proc. IEEE INFOCOM*, 2017.

[105] X. Tian, W. Zhang, J. Wang, W. Li, S. Li, X. Wu, and Y. Yang, "Online pricing crowdsensed fingerprints for accurate indoor localization," in *Vehicular Technology Conference (VTC-Fall), 2017 IEEE 86th.* IEEE, 2017, pp. 1–5.

[106] Z. Peng, S. Gao, B. Xiao, S. Guo, and Y. Yang, "Crowdgis: Updating digital maps via mobile crowdsensing," *IEEE Transactions on Automation Science and Engineering*, vol. 15, no. 1, pp. 369–380, 2018.

[107] C. Cao, Z. Liu, M. Li, W. Wang, and Z. Qin, "Walkway discovery from large scale crowdsensing," in *Proceedings of the 17th ACM/IEEE International Conference on Information Processing in Sensor Networks.* IEEE Press, 2018, pp. 13–24.

[108] X. Fan, P. Yang, C. Xiang, and L. Shi, "imap: A crowdsensing based system for outdoor radio signal strength map," in *Trustcom/BigDataSE/ISPA, 2016 IEEE.* IEEE, 2016, pp. 1442–1447.

[109] L. Zhang, L. Zhao, Z. Wang, and J. Liu, "Wifi networks in metropolises: From access point and user perspectives," *IEEE Communications Magazine*, vol. 55, no. 5, pp. 42–48, 2017.

[110] M. S. Afaqui, E. Garcia-Villegas, and E. Lopez-Aguilera, "Ieee 802.11 ax: Challenges and requirements for future high efficiency wifi," *IEEE wireless communications*, vol. 24, no. 3, pp. 130–137, 2017.

[111] J. Ling, S. Kanugovi, S. Vasudevan, and A. K. Pramod, "Enhanced capacity and coverage by wi-fi lte integration," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 165–171, 2015.

[112] A. Chakraborty, M. S. Rahman, H. Gupta, and S. R. Das, "Specsense: Crowdsensing for efficient querying of spectrum occupancy," in *IEEE INFOCOM*, 2017.

[113] D. Wu, Q. Liu, Y. Li, J. A. McCann, A. C. Regan, and N. Venkatasubramanian, "Adaptive lookup of open wifi using crowdsensing," *IEEE/ACM Transactions on Networking*, vol. 24, no. 6, pp. 3634–3647, 2016.

[114] T. Zhou, Z. Cai, B. Xiao, Y. Chen, and M. Xu, "Detecting rogue ap with the crowd wisdom," in *Distributed Computing Systems (ICDCS), 2017 IEEE 37th International Conference on.* IEEE, 2017, pp. 2327–2332.

[115] C. Xiang, P. Yang, C. Tian, C. Li, Q. Li, and X. Li, "Accurate quantification of sensor noise in participatory sensing network," *Adhoc & Sensor Wireless Networks*, vol. 30, 2016.

[116] C. Xiang, P. Yang, C. Tian, L. Zhang, H. Lin, F. Xiao, M. Zhang, and Y. Liu, "Carm: crowd-sensing accurate outdoor rss maps with error-prone smartphone measurements," *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2669–2681, 2016.

[117] Y. Kim, Y. Chon, and H. Cha, "Mobile crowdsensing framework for a large-scale wi-fi fingerprinting system," *IEEE Pervasive Computing*, vol. 15, no. 3, pp. 58–67, 2016.

[118] Z. Zheng, F. Wu, X. Gao, H. Zhu, S. Tang, and G. Chen, "A budget feasible incentive mechanism for weighted coverage maximization in mobile crowdsensing," *IEEE Transactions on Mobile Computing*, vol. 16, no. 9, pp. 2392–2407, 2017.

[119] 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, pp. 161–167, 2016.

[120] T. Amano, S. Kajita, H. Yamaguchi, T. Higashino, and M. Takai, "A crowdsourcing and simulation based approach for fast and accurate wi-fi radio map construction in urban environment," in *IFIP Networking Conference (IFIP Networking) and Workshops, 2017.* IEEE, 2017, pp. 1–9.

[121] X. Wu, P. Yang, S. Tang, X. Zheng, and Y. Xiong, "Privacy preserving RSS map generation for a crowdsensing network," *IEEE Wireless Communications*, vol. 22, no. 4, pp. 42–48, 2015.

[122] "Lean cloud," https://leancloud.cn/.

[123] Z. Han, H. Li, and W. Yin, *Compressive sensing for wireless networks.* Cambridge University Press, 2013.

[124] R. G. Baraniuk, "Compressive sensing [lecture notes]," *IEEE signal processing magazine*, vol. 24, no. 4, pp. 118–121, 2007.

[125] A. Massa, P. Rocca, and G. Oliveri, "Compressive sensing in electromagnetics-a review," *IEEE Antennas and Propagation Magazine*, vol. 57, no. 1, pp. 224–238, 2015.

[126] D. L. Donoho, "Compressed sensing," *IEEE Transactions on information theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

[127] C. Dick, F. Harris, and M. Rice, "Synchronization in software radios. carrier and timing recovery using fpgas," in *Proceedings 2000 IEEE Symposium on Field-Programmable Custom Computing Machines (Cat. No. PR00871).* IEEE, 2000, pp. 195–204.

[128] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Transactions on information theory*, vol. 52, no. 2, pp. 489–509, 2006.

[129] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[130] K. Xie, L. Wang, X. Wang, G. Xie, G. Zhang, D. Xie, and J. Wen, "Sequential and adaptive sampling for matrix completion in network monitoring systems," in *Computer Communications (INFOCOM), 2015 IEEE Conference on.* IEEE, 2015, pp. 2443–2451.

[131] A. P. Plageras, K. E. Psannis, C. Stergiou, H. Wang, and B. B. Gupta, "Efficient iot-based sensor big data collection–processing and analysis in smart buildings," *Future Generation Computer Systems*, vol. 82, pp. 349–357, 2018.

[132] O. Bates and A. Friday, "Beyond data in the smart city: repurposing existing campus iot," *IEEE Pervasive Computing*, vol. 16, no. 2, pp. 54–60, 2017.

[133] N. Nesa and I. Banerjee, "IoT-based sensor data fusion for occupancy sensing using dempster–shafer evidence theory for smart buildings," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1563–1570, 2017.

[134] T. C. Team, "A to Z of smart buildings," https://www.comfyapp.com/blog/d-data/, accessed August, 2019.

[135] D. Snoonian, "Smart buildings," *IEEE Spectrum*, vol. 40, pp. 18–23, 2003.

[136] Z. Zheng, F. Wang, D. Wang, and L. Zhang, "Buildings affect mobile patterns: developing a new urban mobility model," in *Proceedings of the 5th Conference on Systems for Built Environments*. ACM, 2018, pp. 83–92.

[137] H. Rashid, N. Batra, and P. Singh, "Rimor: towards identifying anomalous appliances in buildings," in *Proceedings of the 5th Conference on Systems for Built Environments*. ACM, 2018, pp. 33–42.

[138] Z. Zheng, D. Wang, J. Pei, Y. Yuan, C. Fan, and F. Xiao, "Urban traffic prediction through the second use of inexpensive big data from buildings," in *Proceedings of the 25th ACM International on Conference on Information and Knowledge Management*. ACM, 2016, pp. 1363–1372.

[139] D. Jo, B. Yu, H. Jeon, and K. Sohn, "Image-to-image learning to predict traffic speeds by considering area-wide spatio-temporal dependencies," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 2, pp. 1188–1197, 2018.

[140] N. E. Klepeis, W. C. Nelson, W. R. Ott, J. P. Robinson, A. M. Tsang, P. Switzer, J. V. Behar, S. C. Hern, and W. H. Engelmann, "The national human activity pattern survey (nhaps): a resource for assessing exposure to environmental pollutants," *Journal of Exposure Science and Environmental Epidemiology*, vol. 11, no. 3, p. 231, 2001.

[141] X. Zhan, Y. Zheng, X. Yi, and S. V. Ukkusuri, "Citywide traffic volume estimation using trajectory data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 2, pp. 272–285, 2016.

[142] D. Deng, C. Shahabi, U. Demiryurek, L. Zhu, R. Yu, and Y. Liu, "Latent Space Model for Road Networks to Predict Time-Varying Traffic," in *ACM KDD*. New York, New York, USA: ACM Press, 2016, pp. 1525–1534.

[143] A. M. Nagy and V. Simon, "Survey on traffic prediction in smart cities," *Pervasive and Mobile Computing*, 2018.

[144] J. Tang, G. Zhang, Y. Wang, H. Wang, and F. Liu, "A hybrid approach to integrate fuzzy c-means based imputation method with genetic algorithm for missing traffic volume data estimation," *Transportation Research Part C: Emerging Technologies*, vol. 51, pp. 29–40, 2015.

[145] T. Idé, T. Katsuki, T. Morimura, and R. Morris, "City-wide traffic flow estimation from a limited number of low-quality cameras," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 950–959, 2016.

[146] L. Lin, J. Li, F. Chen, J. Ye, and J. Huai, "Road traffic speed prediction: a probabilistic model fusing multi-source data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 7, pp. 1310–1323, 2018.

[147] J. Wan, J. Liu, Z. Shao, A. V. Vasilakos, M. Imran, and K. Zhou, "Mobile crowd sensing for traffic prediction in internet of vehicles," *Sensors*, vol. 16, no. 1, p. 88, 2016.

[148] Z. Qin, Z. Fang, Y. Liu, C. Tan, W. Chang, and D. Zhang, "Eximius: A measurement framework for explicit and implicit urban traffic sensing," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*. ACM, 2018, pp. 1–14.

[149] B. Balaji, A. Bhattacharya, G. Fierro, J. Gao, J. Gluck, D. Hong, A. Johansen, J. Koh, J. Ploennigs, Y. Agarwal *et al.*, "Brick: Towards a unified metadata

schema for buildings," in *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*. ACM, 2016, pp. 41–50.

[150] Z. Pan, Y. Liang, W. Wang, Y. Yu, Y. Zheng, and J. Zhang, "Urban traffic prediction from spatio-temporal data using deep meta learning," in *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2019, pp. 1720–1730.

[151] S. P. Mohanty, U. Choppali, and E. Kougianos, "Everything you wanted to know about smart cities: The internet of things is the backbone," *IEEE Consumer Electronics Magazine*, vol. 5, no. 3, pp. 60–70, 2016.

[152] D. Minoli, K. Sohraby, and B. Occhiogrosso, "Iot considerations, requirements, and architectures for smart buildingsenergy optimization and next-generation building management systems," *IEEE Internet of Things Journal*, vol. 4, no. 1, pp. 269–283, 2017.

[153] Z. Liu, Z. Li, M. Li, W. Xing, and D. Lu, "Mining road network correlation for traffic estimation via compressive sensing," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 7, pp. 1880–1893, 2016.

[154] C. Hu, W. Bao, D. Wang, Y. Qian, M. Zheng, and S. Wang, "stube+: an iot communication sharing architecture for smart after-sales maintenance in buildings," *ACM Transactions on Sensor Networks (TOSN)*, vol. 14, no. 3-4, p. 29, 2018.

[155] K. Nellore and G. P. Hancke, "A survey on urban traffic management system using wireless sensor networks," *Sensors*, vol. 16, no. 2, p. 157, 2016.

[156] X. Kong, X. Song, F. Xia, H. Guo, J. Wang, and A. Tolba, "Lotad: long-term traffic anomaly detection based on crowdsourced bus trajectory data," *World Wide Web*, vol. 21, no. 3, pp. 825–847, 2018.

[157] Z. Liu, P. Zhou, Z. Li, and M. Li, "Think like a graph: Real-time traffic estimation at city-scale," *IEEE Transactions on Mobile Computing*, 2018.

[158] Y. Cui, B. Jin, F. Zhang, B. Han, and D. Zhang, "Mining spatial-temporal correlation of sensory data for estimating traffic volumes on highways," in *Proceedings of the 14th EAI International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*. ACM, 2017, pp. 343–352.

[159] A. Janecek, D. Valerio, K. A. Hummel, F. Ricciato, and H. Hlavacs, "The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring," *IEEE transactions on intelligent transportation systems*, vol. 16, no. 5, pp. 2551–2572, 2015.

[160] C. Costa, G. Chatzimilioudis, D. Zeinalipour-Yazti, and M. F. Mokbel, "Towards real-time road traffic analytics using telco big data," in *Proceedings of the International Workshop on Real-Time Business Intelligence and Analytics*. ACM, 2017, p. 5.

[161] C. Meng, X. Yi, L. Su, J. Gao, and Y. Zheng, "City-wide traffic volume inference with loop detector data and taxi trajectories," in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2017, p. 1.

[162] A. Sarker, H. Shen, and J. A. Stankovic, "Morp: data-driven multi-objective route planning and optimization for electric vehicles," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 4, p. 162, 2018.

[163] D. Pavlyuk, "Short-term traffic forecasting using multivariate autoregressive models," *Procedia Engineering*, vol. 178, pp. 57–66, 2017.

[164] Y. Lv, Y. Duan, W. Kang, Z. Li, F.-Y. Wang *et al.*, "Traffic flow prediction with big data: A deep learning approach." *IEEE Trans. Intelligent Transportation Systems*, vol. 16, no. 2, pp. 865–873, 2015.

[165] Z. Liu, Z. Li, K. Wu, and M. Li, "Urban traffic prediction from mobility data using deep learning," *IEEE Network*, vol. 32, no. 4, pp. 40–46, 2018.

[166] J. Xu, R. Rahmatizadeh, L. Bölöni, and D. Turgut, "Real-time prediction of taxi demand using recurrent neural networks," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2572–2581, 2018.

[167] Z. Zhao, W. Chen, X. Wu, P. C. Chen, and J. Liu, "Lstm network: a deep learning approach for short-term traffic forecast," *IET Intelligent Transport Systems*, vol. 11, no. 2, pp. 68–75, 2017.

[168] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "Lstm-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, 2018.

[169] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, 2017, pp. 5998–6008.

[170] H. Yao, X. Tang, H. Wei, G. Zheng, and Z. Li, "Revisiting spatial-temporal similarity: A deep learning framework for traffic prediction," in *AAAI Conference on Artificial Intelligence*, 2019.

[171] "EIF research data interface," https://eif-research.feit.uts.edu.au/, accessed August, 2019.

[172] Roads and N. Maritime Services, "Roads and maritime services collects traffic volume information from roadside traffic collection devices across the nsw road network," https://www.rms.nsw.gov.au/about/corporate-publications/statistics/traffic-volumes/index.html, accessed August, 2019.

[173] F. C. Sangoboye and M. B. Kjærgaard, "Plcount: A probabilistic fusion algorithm for accurately estimating occupancy from 3d camera counts," in *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*. ACM, 2016, pp. 147–156.

[174] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: a deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9508–9517, 2016.

[175] J. Zhang, Y. Zheng, and D. Qi, "Deep spatio-temporal residual networks for citywide crowd flows prediction." in *AAAI*, 2017, pp. 1655–1661.

[176] W. Min and L. Wynter, "Real-time road traffic prediction with spatio-temporal correlations," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 4, pp. 606–616, 2011.

[177] S. R. Chandra and H. Al-Deek, "Predictions of freeway traffic speeds and volumes using vector autoregressive models," *Journal of Intelligent Transportation Systems*, vol. 13, no. 2, pp. 53–72, 2009.

[178] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Advances in neural information processing systems*, 2014, pp. 3104–3112.

[179] Y. Qin, D. Song, H. Chen, W. Cheng, G. Jiang, and G. Cottrell, "A dual-stage attention-based recurrent neural network for time series prediction," *arXiv preprint arXiv:1704.02971*, 2017.

[180] G. Lai, W.-C. Chang, Y. Yang, and H. Liu, "Modeling long-and short-term temporal patterns with deep neural networks," in *The 41st International ACM SIGIR Conference on Research & Development in Information Retrieval*. ACM, 2018, pp. 95–104.

[181] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction." in *IJCAI*, 2018, pp. 3428–3434.

[182] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," *https://openreview.net/forum?id=SJiHXGWAZ*, 2018.

[183] H. Zhang, S. Feng, C. Liu, Y. Ding, Y. Zhu, Z. Zhou, W. Zhang, Y. Yu, H. Jin, and Z. Li, "Cityflow: A multi-agent reinforcement learning environment for large scale city traffic scenario," in *The World Wide Web Conference*. ACM, 2019, pp. 3620–3624.

[184] H. Wei, G. Zheng, H. Yao, and Z. Li, "Intellilight: A reinforcement learning approach for intelligent traffic light control," in *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. ACM, 2018, pp. 2496–2505.

[185] L. Zhu, F. R. Yu, Y. Wang, B. Ning, and T. Tang, "Big data analytics in intelligent transportation systems: a survey," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–16, 2018.

[186] Z. Liu, S. Jiang, P. Zhou, and M. Li, "A participatory urban traffic monitoring system: the power of bus riders," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 10, pp. 2851–2864, 2017.

[187] C. Xu, K. Qiu, J. Fu, S. Bai, Y. Xu, and X. Bai, "Learn to scale: Generating multipolar normalized density maps for crowd counting," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 8382–8390.

[188] X. Cao, Z. Wang, Y. Zhao, and F. Su, "Scale aggregation network for accurate and efficient crowd counting," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 734–750.

[189] R. Jia, F. C. Sangogboye, T. Hong, C. Spanos, and M. B. Kjærgaard, "Privacy-preserving building-related data publication using pad," in *Proceedings of the 4th ACM International Conference on Systems for Energy-Efficient Built Environments*. ACM, 2017, p. 32.

[190] K. Zhang, J. Ni, K. Yang, X. Liang, J. Ren, and X. S. Shen, "Security and privacy in smart city applications: Challenges and solutions," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 122–129, 2017.

[191] S. Yao, Y. Zhao, H. Shao, D. Liu, S. Liu, Y. Hao, A. Piao, S. Hu, S. Lu, and T. F. Abdelzaher, "Sadeepsense: Self-attention deep learning framework for heterogeneous on-device sensors in internet of things applications," in *IEEE INFOCOM 2019-IEEE Conference on Computer Communications*. IEEE, 2019, pp. 1243–1251.