DISSERTATION

DATA MINING AND SPATIOTEMPORAL ANALYSIS OF MODERN MOBILE DATA

Submitted by

Luoyang Fang

Department of Electrical and Computer Engineering

In partial fulfillment of the requirements

For the Degree of Doctor of Philosophy

Colorado State University

Fort Collins, Colorado

Summer 2019

Doctoral Committee:

Advisor: Liuqing Yang

Anura P Jayasumana
Jie Luo
Haonan Wang

ABSTRACT

DATA MINING AND SPATIOTEMPORAL ANALYSIS OF MODERN MOBILE DATA

Modern mobile network technologies and smartphones have successfully penetrated nearly every aspect of human life due to the increasing number of mobile applications and services. Massive mobile data generated by mobile networks with timestamp and location information have been frequently collected. Mobile data analytics has gained remarkable attention from various research communities and industries, since it can broadly reveal the human spatiotemporal mobility patterns from the individual level to an aggregated one. In this dissertation, two types of spatiotemporal modeling with respect to human mobility behaviors are considered, namely the individual modeling and aggregated modeling.

As for individual spatiotemporal modeling, location privacy is studied in terms of user identifiability between two mobile datasets, merely based on their spatiotemporal traces from the perspective of a privacy adversary. The success of user identification then hinges upon the effective distance measures via user spatiotemporal behavior profiling. However, user identification methods depending on a single semantic distance measure almost always lead to a large portion of false matches. To improve user identification performance, we propose a scalable multi-feature ensemble matching framework that integrates multiple explored spatiotemporal models.

On the other hand, the aggregated spatiotemporal modeling is investigated for network and traffic management in cellular networks. Traffic demand forecasting problem across the entire mobile network is first studied, which is considered as the aggregated behavior of

network users. The success of demand forecasting relies on effective modeling of both the spatial and temporal dependencies of the per-cell demand time series. However, the main challenge of the spatial relevancy modeling in the per-cell demand forecasting is the uneven spatial distribution of cells in a network. In this work, a dependency graph is proposed to model the spatial relevancy without compromising the spatial granularity. Accordingly, the spatial and temporal models, graph convolutional and recurrent neural networks, are adopted to forecast the per-cell traffic demands.

In addition to demand forecasting, a per-cell idle time window (ITW) prediction application is further studied for predictive network management based on subscribers' aggregated spatiotemporal behaviors. First, the ITW prediction is formulated into a regression problem with an ITW presence confidence index that facilitates direct ITW detection and estimation. To predict the ITW, a deep-learning-based ITW prediction model is proposed, consisting of a representation learning network and an output network. The representation learning network is aimed to learn patterns from the recent history of demand and mobility, while the output network is designed to generate the ITW predicts with the learned representation and exogenous periodic as inputs. Upon this paradigm, a temporal graph convolutional network (TGCN) implementing the representation learning network is also proposed to capture the graph-based spatiotemporal input features effectively.

ACKNOWLEDGMENTS

It has been almost eight years when I started to pursuit my Ph.D. degree, and it now comes to the final stage. I have wandered for years and finally encounter this exciting topic at the beginning of the sixth year. Before that, I have entirely changed my Ph.D. thesis topic twice. During the entire process, I would like first to thank God Father, Lord Jesus Christ, and Holy Spirit, who gives me courages, wisdoms, and strengths, especially when I was considering quitting the Ph.D. program by the end of the fourth year.

I want to sincerely express my gratitude to my advisor Dr. Liuqing Yang, who has spent her efforts and time for the advisory. Her encouragements and supports are tremendous and critical. I would also like to appreciate Dr. Haonan Wang's guidance on this dissertation. I also would like to thank my Ph.D. committee members, Dr. Jie Luo and Dr. Anura Jayasumana, who provided valuable comments and great discussions in my preliminary and final exams.

I am also very grateful for my friends and colleagues: Dr. Rongqing Zhang, Dr. Dongliang Duan, Dr. Xiang Cheng, Dr. Dexin Wang, Dr. Xilin Cheng, Dr. Wenshu Zhang, Dr. Bo Yu, Pan Deng, Dr. Ning Wang, Robert Griffin (R.I.P.), Dr. Jian Dang, Dr. Xiaotian Zhou, Dr. Yang Cao, Shijian Gao, Xinhu Zheng, Qiang Cui, Dr. Rui Hou, and Yupeng Li.

Finally, I am grateful to my parents, Chunzhen Fang and Jianhua Fang, for their continuous supports. I would like to thank my wife, Yunshi Hu, who supports me with her hearts and minds in the past eight years. Without her, I cannot be here. Also, I am thankful for God's priceless gift, my son Isaac, who gives me endless joys and consolations.

DEDICATION

To my Lord, God Father, Jesus Christ, Holy Spirit

and my wife, Yunshi

and my kids, Isaac and Hosanna

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

INDEX OF NOTATION

$\boldsymbol{A}$	Matrix
$\boldsymbol{a}$	Vector
$\mathcal{A}$	Set
$\boldsymbol{A}_{ij}$	(i, j)-th entry of matrix $\boldsymbol{A}$
$\boldsymbol{A}^{-1}$	Inverse of matrix $\boldsymbol{A}$
$\text{diag}(\boldsymbol{h})$	a diagonal matrix with elements of vector $\boldsymbol{h}$ on its diagonal
$\mathbf{0}$	All-zero matrix
$\mathbf{1}$	All-one column vector
$\mathbb{1}(\cdot)$	Indicator function
$\lvert \cdot \rvert$	Absolute value or set cardinality
$(\cdot)^\top$	Transpose
$\lVert \cdot \rVert_2$	Frobenious norm
$\arg\max_x f(x)$	Argument of the maximum
$\arg\min_x f(x)$	Argument of the minimum

CHAPTER 1

INTRODUCTION

Since the appearance of the first commercial cellular network launched by Nippon Telegraph and Telephone (NTT) in 1979, mobile network technology has become a necessity of human society during the past four decades of its amazingly rapid development. In 2009, the long-term evolution (LTE) network (the most popular fourth generation standard) was first deployed in Oslo, Norway, and Stockholm, Sweden. Since then, mobile phones (smartphones) have successfully penetrated nearly every aspect of human life, due to the flourished mobile applications and services. At the same time, the massive data generated by mobile devices, during mobile network operations, and at backend servers, termed as mobile big data, has attracted significant attention from various research communities and industries. However, large-scale collection and analysis on mobile big data only become possible in the past decade, resulting from the advance of the computing and transmission capability in dealing with such a large volume of mobile data. In this dissertation, we focus on data mining on mobile big data collected by cellular network operators.

1.1 Spatiotemporal Data

One of the most distinct characteristics of mobile big data is its spatiotemporal feature. Almost every entry in mobile big data is tagged with a time stamp and certain geolocation information, which enables a large number of new applications. Almost every smartphone is equipped with a GPS receiver, which provides accurate outdoor location information

with raw data containing the latitude and longitude. Even when the location service of a smartphone based on GPS is not enabled or not reliable, different granularities of location information can be inferred by other data entries, e.g., service set identifier (SSID) of WiFi access points, cell ID in call detail records (CDR) [3], WiFi signal strength [4–6], and even IP addresses [7–9].

As a result, the mobility of human being can be studied based on the highly informative mobile big data in the literature. Behavior patterns revealed by the mobile big data can facilitate many novel data-driven applications spanning subjects from personalized location-based recommendation and pervasive health computing to aggregated public services, including urban planning and network management. Spatiotemporal modeling is critical in various mobile data mining and data-driven applications. Data mining on mobile data can be categorized into two types, namely individual and aggregated spatiotemporal modeling. On the one hand, the individual spatiotemporal modeling is aimed to analyze the behavior pattern of a specific subscriber so that further applications, such as destination prediction, personal advertisement, etc., can be facilitated. On the other hand, the aggregated spatiotemporal modeling can provide a big picture on aggregated or crowd mobility behaviors, whose potential applications include urban analysis, transportation planning and management, network management, etc.

1.2 Individual Spatiotemporal Modeling for Privacy Evaluation

In this dissertation, we study individual spatiotemporal modeling in terms of the most fundamental concern on mobile big data, namely privacy evaluation. User re-identification attacks consists of one critical privacy concerns on mobile big data [10]. In this work, we

study mobile privacy concerning user re-identifiability merely based on their spatiotemporal traces from the perspective of privacy adversaries. The success of user re-identification hinges on effective distance measures via successful spatiotemporal behavior profiling to distinguish two users. However, user re-identification with one single semantic distance measure may lead to a large portion of false matches, especially when only a few users coexist in two datasets. In this work, we study and propose a scalable multi-feature ensemble matching framework to improve the user re-identification performance.

With multiple distance measures, a scalable ensemble matching mechanism is proposed to integrate these multiple matching results by the majority voting rule. At the same time, multiple spatiotemporal features are explored to characterize users in various semantic aspects. Besides, a user clustering algorithm based on user mobility behaviors is studied to reduce the overall computational complexity of the proposed ensemble matching framework.

1.3 Aggregated Spatiotemporal Modeling for Traffic Management

The aggregated spatiotemporal modeling is generally employed to study the behavior pattern of crowds. In this dissertation, we aim to study the aggregated network demands of subscribers in terms of base stations across cellular networks for predictive network management. Two predictive network applications are studied in this dissertation, namely demand forecasting and idle time window (ITW) prediction.

The demand forecasting plays a crucial role in the predictive physical and virtualized network/traffic management in cellular networks, which can effectively reduce both the capital and operational expenditures by fully exploiting the network infrastructure. In this work, we study the per-cell demand forecasting in cellular networks. The success of de-

mand forecasting relies on the effective modeling of both the spatial and temporal aspects of the per-cell demand time series. However, the main challenge of the spatial relevancy modeling in the per-cell demand forecasting is the irregular spatial distribution of cells in a network. Consequently, applying grid-based models (e.g., convolutional neural networks) would lead to degradation of spatial granularity. In this work, we propose to model the spatial relevancy among cells by a dependency graph based on spatial distances among cells without losing spatial granularity. Hence, the graph convolutional networks (GCNs) and long short-term memory (LSTM) from deep learning are employed to model the spatial and temporal aspects, respectively. Besides, the deep graph-sequence model, graph convolutional LSTM (GCLSTM), is further employed to simultaneously characterize both the spatial and temporal aspects of mobile demand forecasting.

Idle time windows (ITW) consist of one critical trigger for various functions in green intelligent network management and traffic scheduling in mobile networks. In this work, we study the ITW prediction in mobile networks based on network subscribers' demand and mobility behaviors observed by network operators. We first formulate the ITW prediction into a regression problem with an ITW presence confidence index that facilitates direct ITW detection & estimation. Feature extraction on the demand and mobility history is then proposed to capture the current trends of subscribers' demand and mobility as well as to account for the periodicity underlying subscribers' demand and mobility patterns as exogenous inputs. In light of feature engineering, a deep-learning-based ITW prediction model is proposed, which consists of two components, namely a representation learning network and an output network. The representation learning network is aimed to learn useful patterns, while the output network is designed to generate the desired ITW presence confidence index

and ITW estimates by integrating the learned representation and exogenous as inputs. In this work, a novel temporal graph convolutional network (TGCN) for representation learning network is proposed to capture the graph-based spatiotemporal input features effectively.

1.4 Dissertation Organization

The rest of the dissertation is organized as follows. In Chapter 2, data collection at different network component of cellular networks is summarized with the studied signaling dataset introduced. In Chapter 3, the individual spatiotemporal modeling for privacy evaluation is investigated in terms of user re-identification across two datasets. Two applications based on aggregated spatiotemporal modeling, namely demand forecasting and ITW prediction, are studied in Chapter 4 and Chapter 5, respectively. In Chapter 6, concluding remarks are provided for this dissertation.

CHAPTER 2

DATA SOURCE, COLLECTION AND DESCRIPTION

2.1 Overview of Data Sources

[1]Mobile data can be collected from various sources in the mobile network. These data are usually divided into two categories [12]. One category consists of the *app-level* data directly collected by mobile App vendors from mobile phone sensors. As sensor technologies are ubiquitously equipped in smartphones (e.g., GPS, accelerometer, magnetic field sensor, gyroscope, etc.), the phone usually acts as a sensor hub with enriched connectivity for data collection and transmission. The other data category is the *network-level* one traditionally collected by content service providers and mobile operators, which is a vast amount of different mobile service contents, as well as spatiotemporal mobile broadband data about their systems and customers. This type of data records the system status, the service requests, as well as user information (e.g., user ID, location, device type, time stamps, type of service, etc.).

In terms of the sources of data collection, the app-level data mainly come from the mobile terminals, whereas the network-level data are usually from the over the top (OTT) servers and the network operators. The raw data collected from these sources is summarized in Fig. 2.1. Embedded in these raw data is a large amount of valuable information about the users, including user characteristics, habits, preferences, and even motivations and purposes. Harvesting from these raw data, one can construct more useful information such as context,

[1]This work has been published in [11].

behavior, relationship, etc. Based on these, additional and more implicit information can be further extracted via data mining. Examples include essential user characteristics (age, gender, race), occupation, group, habit, interest, political opinion, etc. These could then be used in followup data analytics to restore the original context of the related mobile terminal utilization.

Data collection is the process in which data containing user characteristics, preferences, or activities is obtained. How the data collection is implemented can be classified into implicit and explicit approaches. In the explicit approach, users are prompted to manually provide various information [13–16]. While being simple and straightforward, this requires each user to be not only clear about what relevant information he/she is disclosed, but also willing to take time and effort to participate. However, this is usually hard to achieve, as users could be discouraged by such inquiries. On the other hand, the implicit approach does not require manual user intervention and is accomplished without interfering with normal user activities. The implicit approach also facilitates more frequent information updates since explicit user responses are not required in such updates. For these reasons, the implicit approach is more prevalent. Nevertheless, implicitly collected data usually contains quite a lot of redundancy and irrelevant information, which could complicate the follow-up processing of the data. In the following subsections, we will present the data in terms of app level and network level.

2.1.1 The App-Level Data

Data collected from mobile devices may be from either the software side or the hardware side. The hardware-side data includes the device usage information, sensor information, etc. The software-side data includes the application information, the user profile associated with

	Data	Parameter
app-level data	Device	Device Type, Device Usage, etc.
	Profile	MSISDN, IMEI, IMSI, User preference, Calendar, Appointment, etc.
	Sensor	Sensing Data, e.g., GPS, Gyroscope, Accelerometer, etc.
	App	Terminal Application Type, Application Usage, etc.
	Service	Service Information, e.g., Bundle Type, Service Charge, etc.
	Log	Terminal Device Log, Server System Log, etc.
network-level data	Time	Connection Starting Time, Session Starting Time, etc.
	Location	Terminal Location, BS Location, Router Location, Cell Location, etc.
	Address	Client IP, Server IP, Client Tunnel IP, Server Tunnel IP, etc.
	URL	Uniform Resource Location, Link Information, Link Content, etc.
	Flow	Uplink traffic, Downlink Traffic, Packet Number, etc.
	Record	Conversation Log, e.g., Conversation Duration, Conversation Time, Conversation Frequency, etc.

Figure 2.1: Basic data and parameters.

the devices, and the system logs [17]. There have been quite a few projects focusing on the collection of data from the mobile terminals. Reality mining carried out by the MIT Human Dynamics Lab over 9 months in 2004 was among the earliest efforts, where 75 faculty and students with the MIT Media Lab and 25 students at the MIT Sloan business school, participated using 100 Nokia 6600 smartphones [18]. In this experiment, call logs, Bluetooth devices in proximity, cell tower IDs, phone status (charging or idle), and popular application usage data have been collected. In the more recent Mobile Data Challenge (MDC) by Nokia, 200 volunteers participated using Nokia N95 in the Lake Geneva region from October 2009 to March 2011 [19]. Data collected include calls, short messages, photos, videos, application events, calendar entries, location points, historically connected cell towers, accelerometer

Project	Time	Organization	Data Collected
Reality Mining http://realitycommons.media.mit.edu/realitymining.html	2004	MIT Human Dynamic Lab	call logs, Bluetooth devices in proximity, cell tow IDs, phone status, popular application usage data
Mobile Data Challenge (MDC) https://www.idiap.ch/dataset/mdc	2009-2011	Nokia	calls, SMS, photos, videos, application events, calendar entries, location points, unique cell towers, accelerometer samples, etc.
Device Analyzer Experiment https://deviceanalyzer.cl.cam.ac.uk/	2011 - ~	Computer Laboratory at the University of Cambridge	covered countries, phone types, OS versions, device settings, installed applications, system properties, Bluetooth devices, WiFi networks, disk storage, energy and charging, telephony, data usage, CPU and memory, alarms, media and contacts

Figure 2.2: Summary of Mobile Data Collection Projects

samples, Bluetooth observations, historically connected Bluetooth devices, WLAN observations, historically connected WLAN access points and audio samples. Since March 2011, the Device Analyzer experiment at a much larger scale involving 12,500 Android devices was carried out by the Computer Laboratory at the University of Cambridge [20,21]. The records of covered countries, phone types, OS versions, device settings, installed applications, system properties, Bluetooth devices, WiFi networks, disk storage status, energy and charging status, telephony, data usage, CPU and memory status, alarms, media and contacts, as well as sensors have been collected and analyzed. These campaigns have been summarized in Fig. 2.2

2.1.2 The Network-Level Data

These data are typically collected either at the OTT servers or the network operator servers. The raw information at the OTT servers consists of a vast amount of texts, user profiles, system logs, audio and visual contents, etc. Most of OTT service providers directly interact with end users, rendering network operators pure "pipes," and thus keeping them away from the invaluable data flow.

On the other hand, the radio access network data mainly come from the interactions between mobile terminals and base stations, which involve cell search, synchronization, link establishment, uplink and downlink data transfer, handover, and system information broadcast. These lead to the exchange of a variety of data involving multiple network layers, such as network and device identity, power/carrier/antenna indices, payload and transmission mode, timing information, and location. Details of data collection by network operators will be discussed in next section.

Compared with the data from the content service providers and mobile terminal devices, the server data items unique to network operators include location, address, time, record, flow, URL, etc. Among these, "location" contains the locations of the base stations (location area code, LAC), the cells (service area code, SAC) and the routers (routing area code, RAC), from which each user's physical position could be uniquely determined, without the assistance of the mobile terminal GPS. "Address" contains the IP addresses of the clients, the servers, and the tunnels, etc. "Time" contains the starting time stamps of user's connections and sessions. Also uniquely accessible by the network operators are the user mobile number (MSISDN) and user device identity (IMEI), from which each user's specific device

can be determined. These data, being privacy sensitive, are not typically accessible by other sources of data collection, unless voluntarily provided by the users. The latter case, however, could potentially compromise the reliability of collected data depending on the user's real willingness to disclose such data.

2.2 Data Collection in Mobile Networks

In this section, the architecture of mobile networks and critical network components, as well as the mobility management mechanism, are first reviewed, based on which the revealed user network behaviors could be better understood. Then, the data collection and data categorization based on the different data collection points in cellular networks are described and discussed in detail.

2.2.1 Network Architecture Overview

The mobile (cellular) network emerged in the '90s of the last century and has become one of the most successful technologies. The first cellular network is aimed to provide voice service wirelessly by distributing multiple base stations within a covered area, each of which is covering a small region exclusively (abstracted as a hexagon in Fig. 2.3). The data traffic capability was added to cellular networks from the second generation of cellular networks and flourished in the fourth generation, the long-term evolution (LTE). Although cellular networks have significantly evolved since its first generation, its two main components remain the same, namely the radio access networks (RAN) and the core networks (CN). In a cellular network, the RAN is responsible for processing wireless signals (baseband and passband) from a user equipment (UE), while the CN is aimed to reliably direct the outgoing and incoming

traffic flow to their respective destinations.

In general, two trends of the cellular network architecture evolution can be observed, namely the packetization and the user-control plane separation. Such cellular network evolution trends significantly improve network delay performance and capacity, which could also facilitate accessible network-level data collection. In 3G networks, data traffic service in the core network is accomplished by the packet switched core (PS Core), while the legacy circuit-switched core (CS Core) inherited from the 2G cellular networks fulfills the traditional voice and texting services. However, all services in the LTE cellular networks are fulfilled via the evolved packet core (EPC), which could simplify the system architecture and enhance system efficiency. Data collection in mobile networks can also benefit from packetization, as the packetized networks could provide more bandwidth for big data collection and transmission.

The other trend of cellular network evolution is the user-control plane separation. In general, the user plane in a network refers to the network that carries data traffic, while the control plane is the network for controlling signal transmissions. In LTE networks, the user-control plane on the interfaces between E-UTRAN and EPC is first separated (interfaces S1-C and S1-U in Fig. 2.3), and then the interface between the serving gateway (SGW) and the packet data network gateway (PGW) (interface S5 (internal) / S8 (roaming) in Fig. 2.3) in 3GPP LTE Standard Release 14. The user-control plane separation could generally reduce the network delay via a centralized control function and support the increase of data traffic by adding user plane nodes without changing the network controlling components. At the same time, the user-control plane separation can also facilitate collection of user data related to the distinct network behaviors.

As LTE consists of the mainstream of mobile networks nowadays, the mobile network

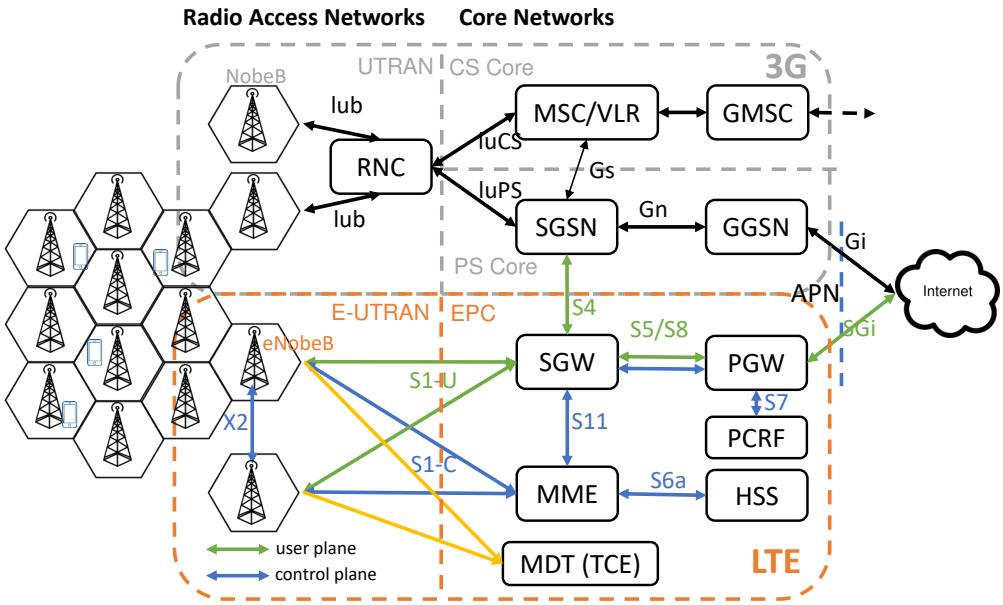


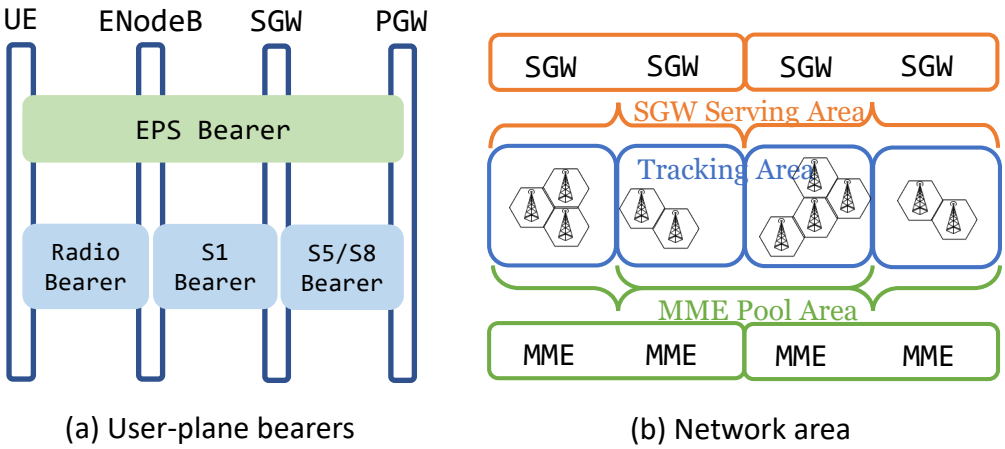
Figure 2.3: Cellular network architecture overview (3G and LTE).

architecture will be illustrated from the perspective of LTE, and the counterparts of the network functionalities in 3G networks will be briefly introduced. In Fig. 2.3, the network architectures of both 3G and LTE (4G) cellular networks are plotted. The double-arrow lines in the figure refer to the logical network connection, beneath which physical transport networks, typically IP networks, are employed to fulfill the network logical connections. Besides, it is worth noting that a logical connection may not necessarily imply a direct physical connection. For example, the interface among nearby eNodeBs, X2, is not necessarily implemented as direct physical connections but can be achieved by routing through the core network.

2.2.2 Key Network Components

The architecture of LTE is outlined in Fig. 2.3, and the main components therein are introduced as follows:

Evolved Node B: The evolved node B (eNodeB or eNB) represents base stations covering



(a) User-plane bearers (b) Network area

Figure 2.4: Bearer and various networks area definition in the LTE.

user equipment (UEs) in a certain area, via which a UE can only communicate with and reach the remote destination. The eNobeB has two main functions. The first function is to process the uplink and downlink radio signals via analog and digital signal processing, while the other one is to fulfill low-level controls via signaling messages (e.g., handover). In fact, the low-level control functions of eNodeB in LTE are inherited from the radio network controller (RNC) in 3G networks as shown in Fig. 2.3, which could reduce the delay due to the reduction of control message exchanges between RNC and base stations. Each eNodeB is connected to EPC via interface S1 and to nearby eNodeBs via interface X2.

Tracking Area (TA): To facilitate mobility management, one partitions the entire covered area into multiple tracking areas (TA), each of which is exclusively comprised of several base stations (eNodeBs) spatially adjacent to each other. In fact, the TA serves as a basic geographic unit for the service coverage area of network components as shown in Fig. 2.4(b). Also, the TA is the basic location unit for user mobility management in LTE networks when users are in the idle state.

Mobility Management Entity (MME): Mobility management entity (MME) is the critical

controlling component in LTE networks, which is the main signaling node in the EPC control plane. Some control functionalities of the MME are inherited from the RNC in 3G networks. In the initial UE attaching phase (UE switch on), the MME will first authenticate and authorize the UE by cooperating with the home subscriber server (HSS) and then assign a proper serving gateway (SGW) to serve the UE. The MME also balances the load of SGWs by directing UE from a heavy-loaded SGW to the light-loaded one. Also, the MME keeps tracking the location of each assigned UE at the granularity of TAs in their idle state (details provided in the next subsection). Based on the location information of UEs, the MME is also responsible for waking up idle UEs, termed as paging in the context of mobile networks, when an incoming flow for the UE arrives at the associated MME. The MME is the component in the LTE network that could monitor user spatiotemporal behaviors, regardless of the UE status (active or idle), which could potentially provide tremendous value to the data collected here.

Serving Gateway (SGW): The serving gateway acts as a high-level router, forwarding the data (user) traffic between eNodeBs and packet data network gateways (PGWs). A network typically contains many serving gateways, each of which handling UEs in a geographical area in terms of TAs. The latter is termed as the SGW serving area, which is not necessarily the same as MME pool area (as shown in Fig.2.4(b)). The SGW is also responsible for inter-eNodeB handovers in the user plane to seamlessly direct data traffic from the outdated eNodeB to the updated one. The downlink traffic for an idle UE is also buffered at the SGW before the idle UE is woken up via the paging procedure scheduled by the MME.

Packet Data Network Gateway (PGW): The packet data network gateway (PGW) is the

point of connection between the PC and external IP networks via interface SGi. Each packet data network (PDN) can be pinpointed by an identifier termed as the access point name (APN). Each UE will be assigned a default PGW in its switch-on initialization. The latter could be attached to other PDNs for private accesses. Typically, the HSS holds a PDN list to which a UE can connect. PGWs are also responsible for packet filtering, charging support, QoS rule, and policy enforcement, which is fulfilled by the policy control enforcement function (PCEF). Generally, the PCEF resides in the PGW and is connected to the policy and charging rule function (PCRF) via interface S7, which is responsible for policy control decision-making and the flow-based charging functionality. PCRF could be viewed as a data aggregation combining device, network, location, and billing information of subscribers. PCRF is a typical data collection point in cellular networks.

Bearers: In LTE, the logical connection between two nodes in the EPC is termed as the bearer (session). It could be viewed as a bidirectional tunnel. The bearer is designed to address the unique issues in LTE networks, namely mobility and quality of service control. Two types of bearers are defined in LTE networks, namely control-plane (signaling) bearers, and user-plane (data traffic) bearers. In Fig. 2.4(a), the user-plane bearer from UE to PGW is illustrated. A default evolved packet system (EPS) bearer will be assigned to UEs in their switch-on initialization, which provides a tunnel for UEs to communicate with external networks. The EPS bearer is comprised of three low-level bearers, each of which corresponding to a specific interface. The resultant bearers include the radio bearer, the S1 bearer, and the S5/S8 bearer. The activation of these bearers relies on the network behaviors of UEs, which will be discussed in the following subsection.

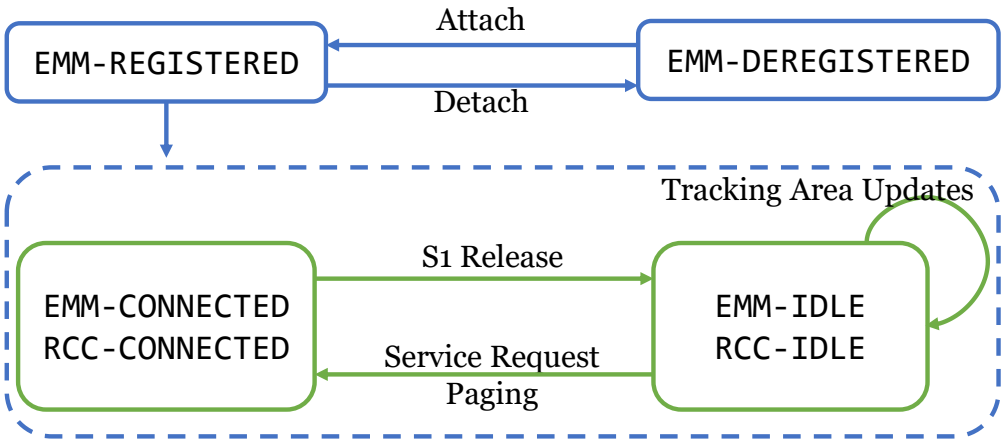


Figure 2.5: User network behaviors.

2.2.3 Mobility Management and User Behaviors

The network behavior of users does not remain unaltered when it registers and is attached to an LTE network. The state of network users is defined by the network behavior diagram shown in Fig. 2.5. Such user management mechanism is aimed to address the issues of limited UE battery life and signaling traffic overload in LTE networks. Once a UE is attached to an LTE network in its switch-on initialization, the UE enters the EPS mobility management (EMM) REGISTERED state from the DEREGISTERED state. At the same time, the UE will be assigned a serving MME, a serving SGW and a default EPS bearer, based on the UE location and the load status of the available MMEs and SGWs. In this phase, the UE enters the EMM/RCC CONNECTED state, indicating that the UE has the full connectivity to the external world. The radio resource control (RCC) state is the one viewed from the perspective of RANs, while the EMM one is viewed from the EPC. Generally, these two states are equivalent. In the EMM/RCC CONNECTED state, the MME has the UE's location information at the granularity of eNobeB. That is, the MME knows the exact eNodeB the UE is attached to as long as the UE is in the EMM/RCC CONNECTED state. It is also

worth noting that UEs in the EMM/RCC CONNECTED state will trigger a handover (HO) event when it arrives a new cell so that the ongoing service could be seamlessly transferred from the outdated eNodeB to the new one.

When the UE is registered but does not consume any radio resources for any services, the S1 release procedure will be scheduled to shift the UE into the EMM/RCC IDLE state. The S1 release procedure is initialized by the UE-attached eNodeB to release the assigned radio bearer and S1 bearer resources. However, the S5/S8 bearer will be retained to accept the UE's downlink data traffic from the external networks. In the EMM/RCC IDLE state, the UE could freely move around with limited signaling message exchanges with eNodeBs and EPC. Also, the MME only has the location knowledge of the UE at the granularity of tracking areas. Tracking area updates will be triggered by two events to maintain the MME's knowledge of the registered UEs' status and location, to facilitate mobility management in LTE. The first event is that the UE enters a new tracking area that is not in the UE's recent tracking area list. The second event is the expiration of a periodic tracking area update timer, whose time duration is typically 54 minutes but can be customized by network operators. During a tracking area update procedure, the UE will temporally enter the EMM/RCC CONNECTED state and then finally return to the EMM/RCC IDLE state.

Two events trigger the transition from the EMM/RCC IDLE state to the EMM/RCC CONNECTED state of UEs. First, the incoming flow to the UE arrives at the serving SGW via interface S5/S8. The paging procedure is triggered by the SGW and scheduled by the MME to search and wake the UE up within the latest tracking area updated by the UE. During the paging procedure, the radio and S1 bearers will be re-assigned to the UE so that the connection between the UE and the external networks could be established. Thus, the

UE's state changes from IDLE to CONNECTED. Secondly, the UE will initialize a service request procedure when it has a communication demand. The service request procedure will sequentially re-establish the radio bearer and S1 bearer at the eNodeB and the serving SGW, respectively. As a result, the UE's state is changed to CONNECTED so that the UE could communicate with external networks.

2.2.4 Data Collection and Categorization

Based on the previous description of network architecture and user network behaviors, the characteristics of data collected at different spots of mobile networks will be discussed here. Generally, mobile data collected in mobile networks could be categorized into four types, namely the call detail records (CDRs) data, the user-plane traffic (UPT) data, the control-plane traffic (CPT) data, and the radio measurement reports (RMR) data.

Call Detail Records (CDR): The CDR data is the most popular dataset studied in the literature [22, 23]. Originally collected for service charging purposes by network operators, the CDR data typically record users' voice and texting activities. Its data fields include the user identifier, when (timestamp) and where (at the granularity of base stations) the event occurs, the duration that the event lasts for voice service. The CDR data may also include the data traffic volume consumed by each UE. The reason behind the high popularity of CRD data is the high accessibility of such data, as the CDR data typically resides at a single server and is well structured. However, the CDR data can only provide the user information for users in the CONNECTED state. Users in the IDLE state do not generate any input to the CDR data. Also, users' data traffic behaviors may be difficult to be thoroughly monitored

	CDR	UPT	CPT	RMR
Collection Point	Charging Station	PGW (LTE) / GGSN (3G)	MME (LTE) / MSC (3G)	Base Stations / MDT (LTE)
Data Fields	User ID Service Type Time Start Location Duration (voice) Volume (traffic)	User ID Session Start Time Session End Time Session Start Location Uplink Volume Downlink Volume Upper Layer Protocols	User ID Event Type Time Location	User ID Time Location WCQI Serving RSRP Serving RSRQ RB Load
Location Granularity	Cells	Cells	Cells Tracking Area	GPS Coordinates
Sampling Rate	Per Service	Per Data Traffic Service	Per Service Per Periodic TAU (default: 54 min)	Determined by Network Operator
Participated Users	All	All	All	Limited (Per User's Permission)

Figure 2.6: Summary of data collections in cellular networks.

based on the CDR data alone.

User-Plane Traffic (UPT) Data: The UPT data refers to the IP session data collected at the PGW (LTE) or GGSN(3G) of cellular networks. The UPT data is generated by inspecting messages tunneled in GRPS tunneling protocol (GTP-U). It encapsulates the IP traffic between UEs and the external networks. The UPT data fields generally include the IP session start and end time, device/user pseudo identifier, type of service, and uplink/downlink traffic volume. Occasionally, the UPT might also include the location of UE at the session start time. However, the user location information in UPT data is sparse and less accurate, as the duration of the session will allow users to move to new cells without any records updated in the UPT data.

Control-Plane Traffic (CPT) Data: The CPT data refers to the data collected at controlling components in mobile networks. Signaling data is a typical data type collected at the mobile

switching center (MSC) in the CS core of 3G networks or the MME in the EPC of LTE networks. In LTE, the data collected at the MME could have higher observability on UE mobility behaviors, compared with the location information of CDR and UPT data. Based on the network mobility management mechanism in LTE networks discussed previously, the MME has the knowledge of the UE location at the granularity of cells when the UE is in the CONNECTED state. Even when UEs are in the IDLE state, the MME still knows their location at the granularity of tracking area via the tracking area updating mechanism of mobility management. Tracking area updates provide the location information in terms of cells at which UEs report their locations. Furthermore, the periodic tracking area update frequency could be significantly increased from a 54-minute update interval to a 14-minute one [24], providing more detailed and more accurate observations on UE mobility behaviors. The data collected at the MSC of 3G networks also have records of UEs' voice and texting service activities. The data fields of CPT data typically include the user identifier, event type, cell ID, and time stamp, etc.

Radio Measurement Reports (RMR): The RMR refers to the data based on radio measurement reports generated at UEs. It is originally aimed to facilitate radio network operation and radio network performance assessments. The RMR is generally challenging to collect, due to the distributed nature of base stations and UEs. Also, the limited storage and computation capabilities of base stations limit the availability of the RMR data. A typical example of RMR data is the measurement reports collected from the minimization of drive tests (MDT) server. The MDT functionality [25] is initially designed in LTE standards to collect radio measurement reports directly from UEs to minimize the drive testing of network

operators for radio network performance assessments. The data fields of MDT data typically include the user ID, wideband channel quality indication (WCQI), serving reference signal received power (RSRP) and quality (RSRQ), as well as resource block (RB) load [26]. Occasionally, user throughput is also included in the MDT data. Their GPS receivers provide the location information of UEs at the granularity of meters, which results in much more precise location observations at the intra-cell level, in comparison with other data. However, such data collection requires the permission of UEs and the investment of infrastructure for data collection and transmission, both of which will limit the availability of RMR data.

2.3 Studied Signaling Dataset

The studied signaling data is a typical example of control-plane data (CPT) collected from mobile networks, which is collected at the mobility management entity of LTE networks. The signaling dataset records every communication/location update event of all active subscribers in a mobile network. Data fields of the signaling data include 1) *subscriber's anonymized identifier*, 2) *time stamp* (e.g., 20160101184312), 3) *location coordinates* (i.e., the longitude and latitude of the base station), 4) *event type*, and 5) *cell type* (i.e., small cell or macro cell). The longitude and latitude coordinates where the base station of each cell is located are accurate to 6 decimal places, and timestamps are accurate to seconds. Besides, the signaling data logs event type as well as the direction of the event (e.g., initiating a call or being called). Compared with the commonly used CDR data, the signaling data does not record the duration information of voice services. However, the signaling data further logs two types of location update events in addition to the regular event types (calls or texts), namely the regular location update and the periodic location update. In cellular networks, location

updating is a fundamental technique of idle mobile device mobility management. The regular location update is triggered by crossing tracking areas, while the periodic location update is prompted by a timeout event when no event occurs for a subscriber within a predefined time. In the studied dataset, the timeout interval is about 1 hour, which can guarantee that any active subscriber in the mobile network has at least one observation per hour in the dataset. In this datasets, around three millions of subscribers are recorded.

CHAPTER 3

INDIVIDUAL SPATIOTEMPORAL MODELING: USER IDENTIFICATION FOR LOCATION PRIVACY EVALUATION

3.1 Background

[1]The privacy of mobile big data is primarily concerned, as human mobility is highly regularized and highly predictable via mobile data spatiotemporal analysis [28, 29]. Mobile big data with spatiotemporal information may need to be released to third parties or even to the public, to facilitate various mobile data-driven applications and services. However, direct data publishing may lead to subscriber privacy leakage risks [30], immediately resulting in data availability issues. For subscribers' privacy protection, the common practice is to anonymize the dataset by replacing subscribers' identifiers (e.g., name, social security number, etc.) with pseudo identifiers. Moreover, the anonymized identifiers are replaced frequently (e.g., every other month) as a data management practice for further privacy protection. However, these practices may not be able to effectively protect subscriber's privacy, due to the uniqueness of human spatiotemporal mobility trajectory [2, 31–36].

Such uniqueness of subscriber mobility behaviors can also lead to another significant concern on subscriber's privacy risk, user re-identification [2, 37]. In this work, we study the mobile privacy in terms of user identity linkage across two datasets as a privacy attacker, based on their spatiotemporal behaviors. The primary purpose of this work is to evaluate subscriber's privacy leakage risk in terms of user identifiability across two datasets.

[1]Part of this work has been published in [27].

A *scalable multi-feature ensemble matching framework* is proposed in this work to study the user identifiability across two datasets, which is aimed to effectively integrate multiple semantic spatiotemporal features and their associated distance measures. User re-identification was studied in [2] based on linear assignment problem (LAP) formulation with the prior knowledge that at most one trace can be exclusively generated by one user in a dataset (termed as exclusiveness). It has been proved [1] that the exclusiveness prior knowledge can effectively improve user re-identification performance. The success of the LAP-based user re-identification relies on effective quantitative distance measures between two spatiotemporal traces.

However, user re-identification with a single distance measure may lead to a large portion of false matches, especially when the coexisting users in two datasets are few. In this work, we argue that the privacy adversary not only concerns about how many user pairs in total can be identified (evaluated by the performance metric *recall*), but also on the *precision* performance metric that suggests the reliability of declared results by a user re-identification algorithm. Most of the works in the literature only focus on the user identification evaluation in terms of the performance metric "recall." In the literature, to discover as many as possible correct pairs (i.e., improve the recall) has been the primary objective without false matches considered. However, though correctly matched pairs are mostly identified in declared matching results, user's privacy could be still maintained to some extent, if many false matched pairs also largely exist (i.e., precision is low). In other words, correctly matched pairs are hidden under false matches, especially when the coexisting number of users across two datasets are small. This is the reason why this work intends to reduce false matches and improve the precision from the perspective of privacy attackers.

As a result, our proposed multi-feature ensemble matching framework is aimed to improve the precision performance of user identification by addressing the following questions:

1. how to effectively integrate identification results based on multiple diverse spatiotemporal features to reduce false matches and ensure a higher user identification precision, especially when the number of coexisting users in two datasets is unknown;

2. how to model and extract various features that could distinctly represent a user and be employed to measure the distance between two spatiotemporal attributes from diverse perspectives.

Firstly, a *LAP-based ensemble matching mechanism* is proposed to integrate diverse distance measures by the philosophy of majority voting. The intuition underlying the proposed ensemble matching mechanism is to crossly validate matched candidates via different semantic spatiotemporal user modeling and their associative distance measures. As a result, a match candidate with minority votes will be considered as a false match so that the precision of the proposed multi-feature ensemble matching framework can be significantly enhanced. Our proposed ensemble matching approach acts as an information/result fusion inspired by the "stacking" approach [38]. The ensemble matching framework is divided into two phases, namely the *vote generation* phase and the *final matching* phase. The first ensemble matching algorithm, *matching-filtered ensemble (MF-Ensemble) matching* , is proposed directly based on LAP formulation: 1) filtering out user-pair candidates via solving LAP for distance matrices in the vote generation phase; 2) obtaining the final result by again solving LAP on the aggregated vote matrix in the final matching phase.

However, the computational complexity of both the vote generation and final matching

phases in the proposed MF-Ensemble matching algorithm is $\mathcal{O}(N^3)$, which significantly limits the scalability. In this work, we also tackle the scalability issue in both the vote generation and ensemble phases. On the one hand, a dual-selection strategy is proposed in the vote generation phase, by relaxing the exclusiveness constraints in LAP, which can significantly reduce the complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N^2)$. On the other hand, a *partitioning and matching (P&M)* algorithm is further proposed in the final matching phase, by taking advantage of the sparsity of vote matrix. The P&M algorithm is to first partition the bipartite graph to subgraphs with the desired size, and then the matching could be employed on resulted subgraph so that the computational complexity can be significantly reduced from $\mathcal{O}(N^3)$ to $\mathcal{O}(N \log N)$. As a results, two additional ensemble matching algorithms, *dual-selection ensemble (DS-Ensemble) matching* and *dual-selection ensemble partitioning & matching (DS-Ensemble P&M)*, are proposed for user re-identification.

Multiple distance measures originated from diverse semantic spatiotemporal modeling are required to facilitate the proposed ensemble matching mechanism. Addition to the *visiting frequency only (VFO)* in [2], we propose to model the *visiting frequency and duration (VFD) jointly*, in which we utilize the temporal information provided by user spatiotemporal traces. Furthermore, we propose to explore the semantic *daily habitat regions (DHR)* for each user, characterizing the maximum area that a user covers in a calendar day. Also, a mobility-pattern-based user grouping or clustering algorithm is further investigated based on non-negative matrix factorization (NMF) [39, 40] for large-scale user re-identification.

Experiment results validate the effectiveness of the proposed multi-feature ensemble matching framework and also suggest that the proposed framework can significantly reduce false matchings with the maximal recall performance slightly compromised. Via large-

scale user re-identification analysis, the mobile privacy of users revealed by mobile data is at high risk even only with two-day data collection periods, where the result of two-day re-identification analysis shows that around 30% users can be re-identified with 70% confidence. The contributions of this work are summarized as follows:

- The importance of user identification precision is emphasized in terms of privacy risk evaluation, based on which the tradeoff between the precision and the recall performance is identified from the perspective of privacy attackers.
- A multi-feature ensemble matching framework is proposed to integrate the results of multiple semantic spatiotemporal modeling so that the false matching can be significantly reduced and the precision of user identification is significantly enhanced.
- The proposed low-complexity DS-Ensemble P&M algorithm with user grouping can facilitate a large-scale user re-identification analysis, whose analytic complexity can be less than $\mathcal{O}(N^2)$.

3.2 Related Work

Generally, privacy protection is highly concerned in any personal-data-related services and applications. *k-anonymity* is a common metric to evaluate the effectiveness of privacy preservation [41], which requires any record in a database to be indistinguishable to at least $k-1$ other records in the database. The most common anonymization technique is to replace critical identifiers (e.g., phone number, IMEI, etc.) with random pseudo identifiers. However, such identifier anonymization fails for the mobile data with which the subscribers' spatiotemporal behavior is recorded, due to the uniqueness of human mobile trajectories [32].

In [42], Zang and Bolot studied a large-scale nationwide dataset with more than 30 billion call records corresponding to 25 million users with different spatial granularities (i.e., cell sector, cell, zip code, city, state). The spatiotemporal footprint of each user is represented by the N most visited places within a predefined time (e.g., day, week, month, ect.), based on which the privacy leakage risk could be evaluated. The authors concluded that the spatiotemporal data sharing or publishing that is only anonymized by pseudo identifiers leads to a severe privacy leakage risk. The potential privacy-preserving solution is to coarsen the temporal resolution, which restricts the accuracy of extracting N most visited locations from the dataset. However, the privacy protection mechanism, including detail-reduction [43] and obfuscation [44], may broadly reduce the utility of the data. It is concluded in [32] that spatiotemporal resolution curtailments may not be useful as expected, based on a human mobility study with 15-month mobile data and 1.5 million people in a country. That is, the uniqueness reduction is magnitudes of order slower than the resolution coarsening.

Therefore, a generalized scheme on the spatiotemporal privacy preserving based on k-anonymity was proposed in [35]. Based on such uniqueness of user mobility behavior, a data-driven spatiotemporal routing generator is developed in [45] to simulate mobility trajectories of users. Also, it is demonstrated in [46] that the aggregated mobility dataset (e.g., the number of subscribers covered by a cell at a specific time) may also lead to a privacy breach of individual mobility trajectory. In [47], a visualization method is developed to infer one's living address based on twitter check-in data.

The user identification (or user reconciliation) [2,31,37,48–56] is another critical problem in privacy protection, which is to link the spatiotemporal records generated by the same user in two datasets. The user identification is closely related to "de-anonymization" attacks. A

typical example is the Netflix prize task that is aimed to de-anonymize user identities by public user reviews [48]. Two types of user identification can be roughly categorized, namely matching users from different domains but in the same time span [37, 49, 50] and matching the users from the same domains in different time spans [2]. In addition, two types of location information, actual GPS coordinates [49, 53, 55] and base station location [2, 31, 54], are mainly studied in the literature.

In [31], De Mulder *et al.* studied the user identification based on the location update dataset from GSM networks, which records the phone's network location with geographical information periodically. The mobility Markovian model of each user is constructed based on their spatiotemporal history. However, such a Markovian model requires the dataset with subscribers' transitions among cells to be recorded, which is not widely adopted or collected by mobile network operators. In [2, 37], user identification is formulated as the minimum (maximum) cost bipartite matching with two sets of vertices representing users in two datasets, respectively, where the edge weight is obtained by a distance (similarity) measure between any pair of nodes in the bipartite graph. In [2], Naini *et al.* suppress the temporal information of users' spatiotemporal trajectories and represent the user fingerprint as the histogram of visited location for a given time length. The distance between the two histograms is calculated by the Jensen-Shannon divergence. Instead of temporal information suppression, Riederer *et al.* in [37] models the number of spatiotemporal appearances of a given spatial and temporal bins by a Poisson process for each dataset, based on which the similarity scores could be generated. However, the task of [37] could is to identify the user of two datasets from different domains during the same time. In [53], the user matching based on vehicle trajectories is investigated based on the improved term frequency and inverse

document frequency (ITF-IDF) mobility feature. The modified Hausdorff distance between two GPS traces has been studied in [49] to distinguish users from different domains. In [51], a privacy risk assessment is studied by assuming that the privacy adversarial has a small portion of the information on users' trajectories, based on which the assessment is aimed to match the prior knowledge with the full record. The privacy leakage assessment is evaluated based on the re-identification rate—the reciprocal of the total number of users that matched the prior information. In [56], a partition-and-group framework is proposed to prevent user re-identification attacks from the adversarial with random prior knowledge.

Although a similar bipartite matching (LAP) formulation for user re-identification is adopted in this work, the unique contributions of this work stand out from previous works in the following aspects. The ensemble concept or cross-validation via different spatiotemporal features is studied to enhance the robustness of user re-identification. Accordingly, a scalable ensemble matching algorithm with user grouping and bipartite graph partitioning is proposed, which can reliably re-identify users. Although we study the user re-identification in the same data domain with different time spans in this work, the proposed ensemble matching framework can be easily extended to the user re-identification in heterogeneous domains, as it provides an effective and scalable approach to integrate the matching results by diverse distance measures.

3.3 Problem Statement

Assume that a spatiotemporal dataset $\mathcal{X}$ is collected by a mobile network operator during a specific time period, in which the i-th subscriber with his/her corresponding mobility trace X_i is represented as a tuple, i.e, $(i, X_i) \in \mathcal{X}$. The mobility trace X_i is a sequence of

timestamped location points (time t_h and location x_h). That is,

$$X_i = [(t_1, x_1), \cdots, (t_h, x_h), \cdots], \ x_l \in \mathcal{A}\ , \tag{3.1}$$

where $\mathcal{A}$ denotes the discrete location point set (i.e., base stations) covered by the mobile network. In other words, $x_l \in \mathcal{A}$ is the identifier (ID) of the base station l or its GPS coordinate, i.e., $x_l = (\text{lng}_l, \text{lat}_l)$. A typical example of such data is the commonly studied call detail records (CDR) [30], which are voice or text event logs collected by network operators for service charging.

Assume that the privacy attacker can access two of such datasets, $\mathcal{X}$ and $\mathcal{Y}$, collected in two time periods. Without loss of generality, the true user identity information of dataset $\mathcal{Y}$ is assumed to be known to the privacy attacker, and then attacker attempts to connect the spatiotemporal information generated by the same user in datasets $\mathcal{X}$ and $\mathcal{Y}$ based on their attributes, despite that these mobility traces are associated with different anonymized IDs within these two datasets. By the assumption that each user can have at most one record in a dataset, the user identification problem can be formulated into a *k-cardinality linear assignment problem* (kLAP) as in [2], that is,

$$\begin{aligned} \underset{c_{ij}}{\text{minimize}} \quad & \sum_{i}^{N}\sum_{j}^{M} c_{ij} w_{ij} \\ \text{subject to} \quad & \sum_{j}^{M} c_{ij} \leq 1,\ \sum_{i}^{N} c_{ij} \leq 1,\ c_{ij} \in \{0,1\}\ , \\ & \sum_{i}^{N}\sum_{j}^{M} c_{ij} = k,\ \forall j \in [M],\ \forall i \in [N] \end{aligned} \tag{3.2}$$

where $N = |\mathcal{X}|$ and $M = |\mathcal{Y}|$ denote the cardinality of $\mathcal{X}$ and $\mathcal{Y}$, respectively[2], and k denotes the number of users coexisting in both the datasets with different anonymized IDs, i.e., $k = |\mathcal{X} \cap \mathcal{Y}|$. The classic solution to the LAP problem is the *Kuhn-Munkres (Hungarian)* algorithm [57] and the *Jonker-Volgenant (JV)* algorithm [58], both of which the complexity is $\mathcal{O}(N^3)$.

It is worth noting that such a kLAP formulation would mainly take advantage of the prior knowledge that one user can generate at most one record in one dataset, termed as *exclusiveness* in this work. The weight w_{ij} in (3.2) denotes the distance between user i from $\mathcal{X}$ and user j from $\mathcal{Y}$, i.e.,

$$w_{ij} = \Delta(X_i, Y_j), \tag{3.3}$$

where $\Delta(X_i, Y_j)$ is a distance measure based on some specific feature modeling and user representation on mobility traces, X_i and Y_j, which will be discussed thoroughly in Section 3.5.

With respect to user re-identification performance assessment, the criterion—how many correct pairs out of the ground truth across two datasets are identified (i.e., *recall*)—is mostly concerned. However, we argue that the privacy adversarial not only concerns about the recall performance but would also care about the *precision* performance for a robust attack, where the precision is defined as the ratio of the number of correctly identified pairs over the total number of declared pairs. The matching with a single distance measure will lead to an inferior precision performance, especially when the coexisting user number k is unknown.

As a common practice, one would assume a maximum coexisting user number k (i.e., $k = \min(N, M)$) in (3.2) in order to extract as many pairs as possible, regardless of inevitable

[2]Without loss of generality, we assume $N \leq M$ in the rest of this chapter.

false matching. Instead, we propose a *scalable multi-feature ensemble matching framework* to effectively and efficiently integrate multiple distance measures based on diverse feature and user representation modeling. The intuition underlying the proposed multi-feature ensemble matching framework is to cross validate the identified pairs by diverse semantic features and eventually determine the final result via majority voting strategy. Accordingly, falsely identified pairs in some distance measures could be eliminated.

3.4 Ensemble Matching

Ensemble learning is a category of algorithms to integrate multiple *weak learners* [38], in order to obtain a much more powerful learner. Ensemble learning is designed initially for classification problems, where *weak learners* should satisfy the following two criteria: 1) weak learners should be accurate to some degree (at least better than random guess) without contaminating final results; 2) weak learners should also be diversified to capture different aspects. In this work, we propose an *ensemble matching* framework to integrate diverse distance measures via different spatiotemporal feature extractions. None of the existing frameworks could be directly applied, as the user identification problem is studied in an unsupervised manner in this work. However, the strategy of majority voting in the classic ensemble learning is adopted, while the information fusion philosophy behind the "stacking" method inspires our proposed ensemble matching. Besides, the exclusiveness property in our studied user identification problem—each user generates at most one record in a dataset—should also be enforced to taking advantage of such prior knowledge to achieve better performance.

Let $\mathcal{W}$ denote the set of G user distance matrices,

$$\mathcal{W} = \left\{ \boldsymbol{W}^1, \boldsymbol{W}^2, \cdots, \boldsymbol{W}^G \right\} , \tag{3.4}$$

where the element of each distance matrix is the pair-wise distance w_{ij} generated by a specific distance measure between user i from $\mathcal{X}$ and user j from $\mathcal{Y}$. Accordingly, the proposed ensemble learning mechanism consists of two phases:

- **Vote generation:** the vote generation phase is to identify the matched candidates corresponding to each distance matrix;
- **Final matching:** the final matching phase is to generate the final matching result with majority voting and exclusiveness property ensured, by regarding the initial identification as votes on specific candidates.

Based on different vote generation strategies, we first propose two ensemble matching algorithms in this work, namely *matching-filtered ensemble (MF-Ensemble) matching* and *dual-selection ensemble (DS-Ensemble) matching*. For the final matching phase, we further propose a low-complexity *Partitioning and Matching (P&M)* algorithm by taking advantage of the sparsity of vote matrix.

3.4.1 Matching-Filtered Ensemble Matching

Each distance matrix can produce k matched pairs by (3.2) from the perspective of its underlying spatiotemporal modeling and user representation. Such matching based on kLAP (3.2) can be regarded as a filter to select k matched candidates out of massive $\binom{N}{k}\binom{M}{k}k!$

possibilities. Therefore, the first phase of the proposed ensemble matching mechanism—vote generation—is fulfilled based on kLAP matching, which termed as matching-filtered ensemble (MF-Ensemble) matching.

With distance matrix set $\mathcal{W}$, let matrix $\boldsymbol{C}^{(g,k)} \in \{0,1\}^{N\times M}$ denotes the matching result by (3.2) based on the g-th distance measure with the assumption of $\hat{k}$ coexisting user number,

$$\boldsymbol{C}^{(g,\hat{k})} = \text{kLAP}(\boldsymbol{W}^g, \hat{k}) \ .$$

Let vote matrix $\boldsymbol{V}_{\text{MF}}^{\hat{k}} \in \mathcal{Z}^{N\times M}$ collect the matching results by total G distance measures on each possible matching pair, that is,

$$\boldsymbol{V}_{\text{MF}}^{(\hat{k})} = \sum_{g=1}^{G} \boldsymbol{C}^{(g,\hat{k})}. \tag{3.5}$$

Therefore, by the strategy of majority votes, the proposed MF ensemble matching algorithm is aimed to maximize the sum vote by solving following combinatoric optimization problem,

$$\begin{aligned} \underset{z_{ij}^{\hat{k}}}{\text{maximize}} \quad & \sum_{i}^{N}\sum_{j}^{M} z_{ij}^{\hat{k}} v_{ij,\text{MF}}^{(\hat{k})} \\ \text{subject to} \quad & \sum_{j}^{M} z_{ij}^{\hat{k}} \le 1,\ \sum_{i}^{N} z_{ij}^{\hat{k}} \le 1,\ z_{ij}^{\hat{k}} \in \{0,1\}\cdot \\ & z_{ij}^{\hat{k}}(v_{ij,\text{MF}}^{(\hat{k})} - \tau) \ge 0, \forall i \in [N], j \in [M] \end{aligned} \tag{3.6}$$

where $\boldsymbol{Z}^{\hat{k}} \in \{0,1\}^{N\times M}$ denotes the final result generated by the proposed MF-Ensemble algorithm. The first two conditions in (3.6) are the same as kLAP (3.2), which guarantee the exclusiveness property. The τ denotes the *vote threshold* that ensures that the final

result is voted by majority, whose typical value is $\tau = \lceil G/2 \rceil$. Thus, the third condition, $z_{ij}^{\hat{k}}(v_{ij,\mathrm{MF}}^{(\hat{k})} - \tau) \geq 0$, is designed to ensure the solution voted by majority. The objective function in (3.6) is aimed to maximize total votes generated by multiple distance measures without any specific restriction on the cardinality of final results, as the cardinality restriction condition has already been enforced in (3.2) before ensemble matching.

In fact, the intuition behind sum vote maximization in (3.6) is to choose the one with more votes when the selection of certain two candidate pairs violates the exclusiveness property, e.g.,

$$\max(v_{ij}, v_{il}),\ v_{ij} \geq \tau,\ , v_{il} \geq \tau.$$

Moreover, we reformulate (3.6) into the classical linear assignment problem (LAP) as follows,

$$\begin{aligned} \underset{z_{ij}}{\text{minimize}} \quad & \sum_{i}^{n}\sum_{j}^{m} z_{ij}\left(G - v_{ij,\mathrm{MF}}^{\hat{k},\tau}\right) \\ \text{subject to} \quad & \sum_{j}^{M} z_{ij} = 1,\ \sum_{i}^{N} z_{ij} \leq 1 \\ & \forall i \in [N],\ \forall j \in [M],\ z_{ij} \in \{0,1\} \end{aligned} \quad . \tag{3.7}$$

where $v_{ij,\mathrm{MF}}^{\hat{k},\tau} = \xi(v_{ij,\mathrm{MF}}^{(\hat{k})}, \tau)$ denotes the votes after thresholding as follows,

$$\xi(v,\tau) = \begin{cases} v, & v \geq \tau \\ 0, & \text{otherwise} \end{cases} . \tag{3.8}$$

Via the Hungarian or JV algorithm of the classic LAP, N pairs are generated (Line 8, Algorithm 1), from which final results are determined by removing the matched pairs whose

Algorithm 1 Matching-Filtered Ensemble Matching

Input: $\mathcal{W} = \{\boldsymbol{W}^1, \boldsymbol{W}^2, \cdots, \boldsymbol{W}^G\}$, k, τ
Output: $\boldsymbol{Z}$
Initiating vote collection matrix $\boldsymbol{V} = \boldsymbol{0}$
for $g \in \{1, 2, \cdots, G\}$ **do** ▷ for each distance measure
    $\boldsymbol{C}^{(g,k)} \leftarrow \text{kLAP}(\boldsymbol{W}^g, k)$ ▷ solve (3.2) on $\boldsymbol{W}^g$
    $\boldsymbol{V} \leftarrow \boldsymbol{V} + \boldsymbol{C}^{(g,k)}$
end for
$\boldsymbol{Z} \leftarrow \text{LAP}(\boldsymbol{V}, G, \tau)$ ▷ solve (3.7)
for $i, j \in [N] \times [M]$ **do**
    if $z_{ij} <> 0$ and $v_{ij}^{\hat{k}} == 0$ **then**
        $z_{ij} \leftarrow 0$ ▷ remove non-major-voted
    end if
end for

votes do not satisfy $v_{ij,\text{MF}}^{\hat{k},\tau} > 0$. Details of the proposed ensemble matching framework are demonstrated in Algorithm 1.

3.4.2 Dual-Selection Ensemble Matching

The proposed MF-Ensemble matching needs to solve kLAP G times in the vote generation phase and solve the LAP in the final matching phase, both of which the computational complexity is $\mathcal{O}(N^3)$. Such high computational complexity may make the proposed MF-Ensemble matching infeasible when user size is large. In fact, the MF-Ensemble algorithm enforces the exclusiveness property in both the vote generation phase and the final matching phase, which may not be necessary. Therefore, we propose a dual selection strategy in the vote generation phase by relaxing the exclusiveness constraint in the vote generation phase, termed as *dual-selection ensemble (DS-Ensemble) matching.*

For each distance matrix $\boldsymbol{W}^g$, matched candidates can be generated based on the minimum distance in terms of each user in both the datasets. For example, each user in dataset

$\mathcal{X}$ would select the most similar user from dataset $\mathcal{Y}$ in terms of distance measure g, i.e.,

$$\mathcal{C}^{(g,\mathcal{X})} = \left\{ (i,j) \,\middle|\, j = \arg\min_{j \in [M]} w_{ij}^{g}, i \in [N] \right\} \tag{3.9}$$

Similarly, each user in dataset $\mathcal{Y}$ can again identify their candidates, i.e.,

$$\mathcal{C}^{(g,\mathcal{Y})} = \left\{ (i,j) \,\middle|\, i = \arg\min_{i \in [N]} w_{ij}^{g}, j \in [M] \right\} \tag{3.10}$$

Therefore, such procedure is termed as *dual selection* (Line 7&10, Algorithm 2). By regarding each pair via the dual selection procedure as one vote, the candidate matrix takes the form as follows,

$$\boldsymbol{C}_{ij}^{(g,\{\cdot\})} = \begin{cases} 1 & (i,j) \in \mathcal{C}^{(g,\{\cdot\})} \\ 0 & \text{otherwise} \end{cases},$$

where $\{\cdot\}$ denotes dataset $\mathcal{X}$ or $\mathcal{Y}$. Hence, the final candidate matrix can be obtained by superimposing these two candidate matrix (Line 12 in Algorithm 2), i.e.,

$$\boldsymbol{C}^{g} = \boldsymbol{C}^{(g,\mathcal{X})} + \boldsymbol{C}^{(g,\mathcal{Y})}. \tag{3.11}$$

It is worth noting that each true pair can get two votes for one distance matrix in the ideal case. Also, an incorrect selection in one dataset does not impact the selection of the other, as the selection in two datasets are independent of each other. The computational complexity of dual selection is $\mathcal{O}(NM)$, an order of magnitude less than matching filtering based vote generation.

The vote collection can be achieved according to (3.5), i.e., $\boldsymbol{V}_{\mathrm{DS}} = \sum_g \boldsymbol{C}^g$. In the DS-Ensemble matching algorithm, the final matching phase needs to ensure k-cardinality condition and exclusiveness property, in order to determine the final matching. Similar to (3.7), the DS-Ensemble matching algorithm is to solve the assignment problem with the constraint of majority voting (Line 14, Algorithm 2) as follows,

$$\begin{aligned} \underset{z_{ij}}{\text{minimize}} \quad & \sum_{i}^{n}\sum_{j}^{m} z_{ij}[2G - v^{\tau}_{ij,\mathrm{DS}}] \\ \text{subject to} \quad & \sum_{j}^{M} z_{ij} < 1, \sum_{i}^{N} z_{ij} \leq 1, z_{ij} \in \{0,1\}\ , \\ & \sum_{i}^{N}\sum_{j}^{M} z_{ij} = \hat{k}, \forall i \in [N], \forall j \in [M] \end{aligned} \tag{3.12}$$

where $v^{\tau}_{ij,\mathrm{DS}} = \xi(v_{ij,\mathrm{DS}}, \tau)$. It is worth noting that the maximum votes for one pair (i,j) through dual selection procedure is $2G$. Thus, the majority voting threshold is $\tau = G$. Details of the proposed DS-Ensemble matching can be found in Algorithm 2.

3.4.3 Dual-Selection Ensemble Partitioning & Matching

The DS-ensemble matching can reduce the computational complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(NM)$ in the vote generation phase, compared with the MF-ensemble matching algorithm. Nonetheless, the scalability issue of our proposed ensemble matching framework still exists due to the high-complexity LAP-based approach in the final matching phase (i.e., (3.7) and (3.12)). In this subsection, we aim to tackle the scalability issue in the final matching phase by reducing the average time complexity from $\mathcal{O}(N^3)$ to $\mathcal{O}(N \log N)$.

It is worth noting that the vote matrix $\boldsymbol{V}$ is an extremely sparse matrix, as each candidate

Algorithm 2 Dual-Selection Ensemble Matching

Input: $\mathcal{W} = \left\{\boldsymbol{W}^1, \boldsymbol{W}^2, \cdots, \boldsymbol{W}^G\right\}$, k, τ
Output: $\boldsymbol{Z}$
Initiating vote collection matrix $\boldsymbol{V} = \boldsymbol{0}$
for $g \in \{1, 2, \cdots, G\}$ **do** ▷ candidate dual selection
    $\boldsymbol{C}^{(g,\mathcal{X})} \leftarrow 0$, $\boldsymbol{C}^{(g,\mathcal{Y})} \leftarrow 0$
    for $i \in [N]$ **do**
        $j \leftarrow \arg\min_j \boldsymbol{W}^g_{ij}$, $\boldsymbol{C}^{(g,\mathcal{X})}_{ij} \leftarrow 1$ ▷ solve (3.9)
    end for
    for $j \in [M]$ **do**
        $i \leftarrow \arg\min_i \boldsymbol{W}^g_{ij}$, $\boldsymbol{C}^{(g,\mathcal{Y})}_{ij} \leftarrow 1$ ▷ solve (3.10)
    end for
    $\boldsymbol{V} \leftarrow \boldsymbol{V} + \boldsymbol{C}^{(g,\mathcal{X})} + \boldsymbol{C}^{(g,\mathcal{Y})}$
end for
$\boldsymbol{Z} \leftarrow \mathrm{kLAP}(\boldsymbol{V}, G, \tau)$ ▷ solve (3.12)
for $i, j \in [N] \times [M]$ **do**
    if $z_{ij} <> 0$ and $v_{ij} < \tau$ **then**
        $z_{ij} \leftarrow 0$ ▷ remove non-major-voted
    end if
end for

matrix by a distance measure (i.e., $\boldsymbol{C}^g$) has at most $2N$ non-zero elements. Besides, the superimposition of a total G candidate matrices will further reduce the number of non-zero elements in the vote matrix $\boldsymbol{V}$. In the worst case, the nonzero element number of vote matrix $\boldsymbol{V}$ will be at the level of $\mathcal{O}(GN)$, where $G \ll N$. Also, the majority voting strategy can further reduce the nonzero element number of vote matrix $\boldsymbol{V}$, as each element of $\boldsymbol{V}$ less than vote threshold τ will be set to zero. One can regard the vote matrix as the *adjacency matrix* of a bipartite graph, $\mathcal{G}(\mathcal{X}, \mathcal{Y}, \boldsymbol{V})$, whose nonzero elements can be regarded as weighted edges between two vertex sets in the bipartite graph. The intuition to resolve the scalability issue in the final matching phase is to first partition the bipartite graph into subgraphs and then conduct matching on the subgraphs to generate final matching results, by taking advantage of the high sparsity of $\boldsymbol{V}$.

3.4.3.1 Bipartite Graph Partitioning without Loss

The sparsity of vote matrix indicates that the entire bipartite graph may be partitioned without loss of votes, as most of the matrix elements are already zero. In other words, vote matrix $\boldsymbol{V}$ may be rearranged into the block diagonal form by shuffling rows and columns as follows,

$$\boldsymbol{V} = \text{diag}\{\boldsymbol{V}_1, \boldsymbol{V}_2, \cdots, \boldsymbol{V}_r\}, \tag{3.13}$$

where $\boldsymbol{V}_i$ denotes a submatrix of $\boldsymbol{V}$ that cannot be further diagonalized without loss of nonzero elements. As a result, one could perform bipartite matching (i.e., (3.12)) on each submatrix $\boldsymbol{V}_i$ to generate final matching results. Rearranging the vote matrix $\boldsymbol{V}$ into a block diagonal form is equivalent to searching the connected components of the bipartite graph $\boldsymbol{V}$. Hence, an efficient tree-based data structure in the literature, *union find* or *disjoint set* [59, Chapter 1], can be easily employed to find the connected components with the time complexity $\mathcal{O}(GN)$.

3.4.3.2 Bipartite Graph Partitioning with Loss

Although the sparsity of the vote matrix may reduce the size for bipartite matching without loss of nonzero elements, the size of each submatrix cannot either be controllable nor be guaranteed to be small enough. In the worse case, the bipartite graph $\boldsymbol{V}$ cannot be partitioned at all, especially when users have very similar mobility behaviors, while vote matrix $\boldsymbol{V}$ still remains extremely sparse as previously discussed. As a result, we propose to partition the bipartite graph with the minimum nonzero loss, where the size of submatrices could be controllable to a certain degree.

Starting from binary partitioning (i.e., each vertex set of a bipartite graph is partitioned into two subsets), the minimization of *normalized cut* is commonly employed as an objective function for bipartite graph partitioning [60,61]. Let vote matrix $\boldsymbol{V}$ be expressed in a block format as follows,

$$\boldsymbol{V} = \begin{bmatrix} \boldsymbol{V}_{11} & \boldsymbol{V}_{12} \\ \boldsymbol{V}_{21} & \boldsymbol{V}_{22} \end{bmatrix}.$$

where $\boldsymbol{V}_{ij}$ corresponds to vertex subsets X_i and Y_j, $i, j \in \{1, 2\}$. Thus, the normalized cut is defined as follows,

$$\text{NCut} = \frac{\text{Cut}}{2\mathbf{1}^T\boldsymbol{V}_{11}\mathbf{1} + \text{Cut}} + \frac{\text{Cut}}{2\mathbf{1}^T\boldsymbol{V}_{22}\mathbf{1} + \text{Cut}}, \tag{3.14}$$

where $\text{Cut} = (\mathbf{1}^T\boldsymbol{V}_{12}\mathbf{1} + \mathbf{1}^T\boldsymbol{V}_{21}\mathbf{1})$ denotes the loss of elements due to bipartite graph partitioning. It is worth noting that the normalized cut minimization is not only aimed to minimize the loss of elements due to graph partitioning, but also designed to balance the partitioning (i.e., the cardinality difference between two vertex subsets should approach to zero).

It has been shown in [61] that the normalized cut minimization based bipartite graph partitioning can boil down to finding the *second largest singular vectors ($\tilde{\boldsymbol{u}}$ and $\tilde{\boldsymbol{v}}$)* of $\tilde{\boldsymbol{V}} = \boldsymbol{D}_X^{-1/2}\boldsymbol{V}\boldsymbol{D}_Y^{-1/2}$

$$\tilde{\boldsymbol{V}}\tilde{\boldsymbol{u}} = \sigma_2\tilde{\boldsymbol{v}},$$

where $\boldsymbol{D}_X = \text{diag}\{\boldsymbol{V}\mathbf{1}\}$ and $\boldsymbol{D}_Y = \text{diag}\{\boldsymbol{V}^T\mathbf{1}\}$ denote the degree of each vertex in X and

Algorithm 3 DS-Ensemble Partitioning&Matching

Input: $\boldsymbol{V}_{\text{DS}}$, X, Y, $\hat{k}$, G, τ, t
Output: $\boldsymbol{Z}$
$\boldsymbol{V}_{\text{DS}}^{\tau} = \xi(\boldsymbol{V}_{\text{DS}}, \tau)$ ▷ vote thresholding
$\boldsymbol{V}_1, \cdots, \boldsymbol{V}_R \leftarrow \text{UnionFind}(\boldsymbol{V}_{\text{DS}}^{\tau})$ ▷ partitioning w/o loss
Initialize $\mathcal{Z} \leftarrow \emptyset$
for $r \in [R]$ **do** ▷ partitioning w/ loss
    $\boldsymbol{Z}_r \leftarrow$PartitionAndMatch($\boldsymbol{V}_r$, X_r, Y_r)
    $\mathcal{Z} \cap \{\boldsymbol{Z}_r\}$
end for
$\boldsymbol{Z} \leftarrow \text{diag}(\mathcal{Z})$
Find top $\hat{k}$ pair based on $\boldsymbol{Z}$ and $\boldsymbol{V}_{\text{DS}}^{\tau}$
function PartitionAndMatch($\boldsymbol{V}$, X, Y)
    if $|X| > t$ or $|Y| > t$ **then** ▷ partitioning
        $\hat{\boldsymbol{u}}$, $\hat{\boldsymbol{v}} \leftarrow \text{Lanczos}(\boldsymbol{D}_X^{-1/2} \boldsymbol{V} \boldsymbol{D}_Y^{-1/2})$
        $X_1 \leftarrow \{i | (\boldsymbol{D}_X \hat{\boldsymbol{u}})_i \geq 0\}$ and $X_2 \leftarrow X - X_1$
        $Y_1 \leftarrow \{j | (\boldsymbol{D}_Y \hat{\boldsymbol{v}})_j \geq 0\}$ and $Y_2 \leftarrow Y - Y_1$
        $\boldsymbol{Z}_{11} \leftarrow$PartitionAndMatch($\boldsymbol{V}_{11}$, X_1, Y_1)
        $\boldsymbol{Z}_{22} \leftarrow$PartitionAndMatch($\boldsymbol{V}_{22}$, X_2, Y_2)
        $\boldsymbol{Z} \leftarrow \text{diag}\{\boldsymbol{Z}_{11}, \boldsymbol{Z}_{22}\}$
    else ▷ matching
        $\boldsymbol{Z} \leftarrow \text{LAP}(\boldsymbol{V}, 2G)$
    end if
    for $i, j \in [N] \times [M]$ **do** ▷ clean via majority voting
        if $z_{ij} <> 0$ and $v_{ij} < \tau$ **then**
            $v_{ij} \leftarrow 0$
        end if
    end for
    return $\boldsymbol{Z}$
end function

Y, respectively. As a result, both user set X and Y can be segmented as follows,

$$\mathcal{X}_1 = \{i | \boldsymbol{u}_i \geq 0\} \text{ and } \mathcal{Y}_1 = \{j | \boldsymbol{v}_j \geq 0\}, \tag{3.15}$$

where $\boldsymbol{u} = \boldsymbol{D}_X \tilde{\boldsymbol{u}}$ and $\boldsymbol{v} = \boldsymbol{D}_Y \tilde{\boldsymbol{v}}$. Furthermore, $\mathcal{X}_2$ and $\mathcal{Y}_2$ can be obtained by finding the complement of $\mathcal{X}_1$ and $\mathcal{Y}_1$, respectively.

3.4.3.3 DS-Ensemble Partitioning and Matching

Based on the previous discussions on vote matrix sparsity and bipartite graph partitioning, we propose a recursive *DS-Ensemble partitioning and matching (P&M)* algorithm with a much lower computational complexity, compared with the DS-Ensemble matching. First, the bipartite graph will be partitioned without loss based on the *union find* (Algorithm 3 Line 4). For each subgraph, a recursive partitioning and matching algorithm (Algorithm 3 Line 12-29) is employed to further segment the graph into multiple subgraphs, whose size is not greater than the size threshold (t in Algorithm 3). In each final subgraph, the Hungarian or JV algorithm (Algorithm 3 Line 21) will be employed to obtain the final matching result (3.7) with the majority voting ensured (Algorithm 3 Line 23-27). After collecting all the matching pairs from all the subgraphs, one can output the top-$\hat{k}$ matched pairs. It is worth noting that the DS-Ensemble P&M algorithm is a suboptimal algorithm, compared with the DS-Ensemble matching. Details of the algorithm refer to Algorithm 3.

In the DS-Ensemble P&M algorithm, the heaviest computational load of bipartite graph partitioning is the second largest singular vector calculation, where the full singular value decomposition is computationally intensive (i.e., $\mathcal{O}(N^3)$). However, thanks to the high sparsity of $\tilde{\boldsymbol{V}}$, the computational complexity of the second largest singular vector calculation can be reduced to $\mathcal{O}(\text{nnz}(\tilde{\boldsymbol{V}}))$ based on Lanczos method in [61] and [62, Chapter 8], where $\text{nnz}(\tilde{\boldsymbol{V}})$ denotes the number of nonzeros in matrix $\tilde{\boldsymbol{V}}$. As a result, the time complexity of binary bipartite graph partitioning is $\mathcal{O}(GN)$. The recursive number of bipartite graph partitioning depends on the size threshold t. By roughly assuming that each bipartite graph partitioning can exactly divide the graph into two equal-size subgraphs, the recursive num-

ber is $\mathcal{O}(\log(N/t))$, and each recursive layer is on the complexity of $\mathcal{O}(GN)$. Thus, the computational complexity of the proposed P&M algorithm is $\mathcal{O}(Nt^2 + \log(N/t)GN)$, where $\mathcal{O}(Nt^2)$ originates from N/t matchings on subgraphs with the size less than t. As t and G are fixed and predefined, the average computational complexity of the DS-Ensemble P&M algorithm could be simplified to $\mathcal{O}(NM + N\log N)$, where $\mathcal{O}(NM)$ originates from the dual selection procedure in the vote generation phase.

3.5 Features, Distances and User Grouping

As stated previously, distance measures w_{ij} play a critical role in user identification, which could largely determine the performance of a user identification algorithm by solving the kLAP in (3.2). However, the raw spatiotemporal attributes could not be directly employed to assess the distance between two spatiotemporal attributes without proper data modeling and feature extraction. Each feature extracted from the raw spatiotemporal attributes could be regarded as a fingerprint of the user. Two spatiotemporal features are proposed to profile users' spatiotemporal behaviors in different aspects, which can contribute significantly to the user identification task via the proposed ensemble matching framework. Each feature and its corresponding distance measures will be discussed in terms of *data modeling*, *representing feature*, and the corresponding *distance measures*.

3.5.1 Visiting Frequency Only (VFO) Modeling [1, 2]

The location visiting frequency only (VFO) is utilized as a representing feature in [1, 2] to distinctly characterize a user.

Data Modeling: With the location point set (base station set) being abstracted as an alphabet

set $\mathcal{A} = \{a_1, \cdots, a_L\}$, the raw spatiotemporal attribute (3.1) could be first modeled as a string with length T by discarding the time information,

$$X_i = x_{i1}, x_{i2}, \cdots, x_{iT}. \tag{3.16}$$

Every element $x_{it} \in \mathcal{A}$ in the string is assumed to be i.i.d. from the alphabet set $\mathcal{A}$ based on an unknown location visiting probability mass function Π_i.

Representing Feature: Based on the i.i.d assumption of the string generation, the location visiting probability Π_i could be estimated by the empirical probability distribution or histogram $\widehat{\Pi}_i$, i.e.,

$$\widehat{\Pi}_{i,l} = \frac{N_i(a_l)}{T}, a_l \in \mathcal{A}\ , \tag{3.17}$$

where $N_i(a_l) = \sum_{x_{it}=a_l} 1$ denotes the number of appearance of letter a_l in the string X_i, counting the number of visits of user i at location a_l. Thus, the spatiotemporal behaviors of a user could be represented by the *histogram*, characterizing his/her *visiting frequency* over location point set $\mathcal{A}$. However, such feature extraction further discards the *temporal information* for modeling simplicity.

Distance Measures: To evaluate whether two spatiotemporal attributes are associated with the same user in terms of the VFO is to find a good measure to assess the distance between the two histograms. Here, the intuitive yet heuristic L_1 distance function could be employed to assess the distance between two histograms as follows,

$$\Delta_{\text{VFO-L1}}(X_i, Y_j) = \frac{1}{2} \sum_{a_l \in \mathcal{A}} \left| \widehat{\Pi}_{i,l} - \widehat{\Pi}_{j,l} \right| \ . \tag{3.18}$$

Based on the multi-hypothesis test framework discussed in [1], to determine the optimum hypothesis using the log likelihood test is equivalent to solving the kLAP with distance generated by the Jensen-Shannon divergence (JSD). Thus, the JSD could serve as a distance measure on the histograms as follows [2],

$$\Delta_{\text{VFO-JSD}}(X_i, Y_j) = \text{JSD}(\widehat{\Pi}_i, \widehat{\Pi}_j) \ . \tag{3.19}$$

where

$$\text{JSD}(p, q) = \text{KL}\left(p \,\|(p+q)/2\right) + \text{KL}\left(q \,\|(p+q)/2\right) \ . \tag{3.20}$$

3.5.2 Visiting Frequency and Duration (VFD) Modeling

The VFO only captures one spatial aspect of the available spatiotemporal attributes, while neglecting the potential temporal information valuable for user identification. Though the collected dataset may be an event log with users' spatiotemporal trajectory sporadically sampled, the temporal information could still be employed to characterize users. In this subsection, we propose a visiting frequency and duration (VFD) feature to jointly capture the distance in both the spatial and temporal aspects.

Data Modeling: Atop the previous string model (3.16), the raw spatiotemporal attribute (3.1) could be modeled as a tuple string with size P_i as follows:

$$X_i = (x_{i1}, t_{i1}), (x_{i2}, t_{i2}), \cdots, (x_{iP_i}, t_{iP_i}) \ , \tag{3.21}$$

where $x_{ip} \in \mathcal{A}$ denotes the p-th recorded location of user i, and t_{ip} denotes the corresponding

duration between the current event and the next one.

Based on the spatiotemporal tuple string modeling, we also assume that each tuple is i.i.d. generated by an unknown probability distribution, where the duration of a user at a given location $a_l \in \mathcal{A}$ is modeled as an exponential (EXP) distribution conditioned upon the location a_l,

$$f(t|a_l;\lambda_{i,l}) = \lambda_{i,l}\exp(-\lambda_{i,l}t),\ t>0, \tag{3.22}$$

where $\lambda_{i,l}$ denotes the reciprocal of the average duration of user i at location point a_l.

Representing Feature: Assume that duration generated at locations are uncorrelated, the likelihood of X_i takes the form

$$\mathcal{L}(X_i) = \prod_l \mathcal{L}(X_i;a_l)\Pi_i(a_l) \tag{3.23}$$

where $\mathcal{L}(X_i;a_l)$ denotes the likelihood of X_i observed at location a_l as in (3.22).

As a result, one can obtain two representations to characterize users in both the spatial and temporal aspects. The spatial representation is the visiting frequency by the empirical probability distribution $\widehat{\Pi}_i$ calculated via (3.17). The temporal representation could be obtained by the location-dependent exponential distribution parameter set $\widehat{\Lambda}_i = \{\widehat{\lambda}_{i,l}\}$, where each element $\lambda_{i,l}$ can be estimated at each $a_l \in \mathcal{A}$,

$$\hat{\lambda}_{i,l} = N_l(a_l)/\sum_{x_{ip}=a_l} t_{ip}. \tag{3.24}$$

Distance Measures: With the similar multi-hypothesis test framework in [1], a distance measure between two users in terms of both $\widehat{\Pi}_i$ and $\widehat{\Lambda}_i$ can be derived. With respect to

the likelihood function (3.23), the derived distance measure can be decomposed into two components, namely visited frequency only (VFO) and visited duration only (VDO), as follows,

$$\Delta_{\text{VFD-WD}}(X_i, Y_j) = \Delta_{\text{VFO-WD}}(X_i, Y_j) + \Delta_{\text{VDO-WD}}(X_i, Y_j) \tag{3.25}$$

Here, the "WD" is short for weighted divergence, which is a generalization of Jensen-Shannon divergence. The $\Delta_{\text{VFO-WD}}$ is originated from (3.20) with weighted divergence employed, while the $\Delta_{\text{VDO-WD}}$ is obtained based on the divergence between two exponential distributions on their corresponding visited durations as follows,

$$\begin{aligned}\Delta_{\text{VDO-WD}}(X_i, Y_j) = \sum_{a_l \in \mathcal{A}} \Big[& q_i \widehat{\Pi}_{i,l} \text{KL}\left(\hat{\lambda}_{i,l} \| \hat{\lambda}_{ij,l}\right) \\ & + q_j \widehat{\Pi}_{j,l} \text{KL}\left(\hat{\lambda}_{j,l} \| \hat{\lambda}_{ij,l}\right)\Big]\end{aligned} . \tag{3.26}$$

where $\text{KL}(\lambda_1 \| \lambda_2)$ denotes the KL divergence on two EXP distributions, i.e., $\text{KL}(\lambda_1 \| \lambda_2) = \log(\lambda_1/\lambda_2) + (\lambda_2/\lambda_1) - 1$. And $\hat{\lambda}_{ij,l}$ is the weighted harmonic average over $\hat{\lambda}_{i,l}$ and $\hat{\lambda}_{j,l}$. Details of derivations on (3.25) and (3.26) can be found in Appendix 3.8.

Assume that the string length of each user and the number of each observation are the same, the JSD could be easily obtained. In addition, the L1 distance could also be applied as follows

$$\Delta_{\text{VFD-L1}}(X_i, Y_j) = \sum_{a_l \in \mathcal{A}} \left| \frac{\widehat{\Pi}_{i,l}}{\hat{\lambda}_{i,l}} - \frac{\widehat{\Pi}_{j,l}}{\hat{\lambda}_{j,l}} \right| . \tag{3.27}$$

Table 3.1: Summary of Data Model, Features and Distance Measures

	Data Modeling	Representation Features	Distance Measures	Complexity
VFO	Location String	Visiting Histogram $\widehat{\Pi}_i$	$\Delta_{\text{VFO-L1}}(X_i, Y_j)$ $\Delta_{\text{VFO-JSD}}(X_i, Y_j)$	$\mathcal{O}$(support)
VFD	Location and Duration Tuple String	Visiting Frequency $\widehat{\Pi}_i$ Location-Dependent Exp. Distribution Set $\widehat{\Lambda}_i$	$\Delta_{\text{VFD-WD}}(X_i, Y_j)$ $\Delta_{\text{VFD-JSD}}(X_i, Y_j)$ $\Delta_{\text{VFD-L1}}(X_i, Y_j)$	$\mathcal{O}$(support)
DHR	Daily Visiting Location Point Set	Convex Hull Set $\mathcal{C}_i$	$\Delta_{\text{DHR-COS}}(X_i, Y_j)$ $\Delta_{\text{DHR-IOU}}(X_i, Y_j)$	$\mathcal{O}$(pnt num.^{2})

3.5.3 Daily Habitat Region (DHR) Modeling

The previously discussed spatiotemporal features abstract discrete location points as independent and unrelated letters in an alphabet set $\mathcal{A}$. Such modeling discards the critical geospatial information. In other words, the relationship between locations via the raw latitude and longitude coordinates is ignored. The geospatial information may help combat the information loss due to the sporadic sampling of users' spatiotemporal trajectories. Thus, a heuristic spatiotemporal feature is employed for user identification [27], *daily habitat regions* (DHR), as well as its corresponding distance measures, based on the geospatial information in this subsection. The daily habitat regions capture the daily spatial coverage of a subscriber, which are expected to be consistent to some degree and may serve as the subscriber's mobility fingerprints.

Data Modeling: The spatiotemporal attribute (3.1) is first formulated into sets of location points:

$$X_i = \{\mathcal{X}_{i1}, \mathcal{X}_{i2}, \cdots, \mathcal{X}_{iQ_X}\}, \tag{3.28}$$

where each set $\mathcal{X}_{iq} \subseteq \mathcal{A}$ denotes a set of location points that the user visits during a calendar date q and Q_X and Q_Y denote the number of days collected in dataset $\mathcal{X}$ and $\mathcal{Y}$, respectively.

Representing feature: Here, we employ a classical computational geometry concept, convex

hull, to approximate the spatial coverage that a user visits daily. By approximating a small region of geo-surface as an Euclidean space, the convex hull of a given point set $\mathcal{X}_{iq}$ in a 2-dimensional surface is defined as the set of the convex combination of the given finite point set as follows,

$$C_{iq} = \left\{ \sum_{l=1}^{|\mathcal{X}_{iq}|} \beta_l a_l \,\middle|\, \forall l, \beta_l > 0, a_l \in \mathcal{X}_{iq} \text{ and } \sum_{l=1}^{|\mathcal{X}_{iq}|} \beta_l = 1 \right\} .$$

Thus, the daily convex hull, C_{iq}, is employed to represent the spatiotemporal behaviors of a user for a given day. Hence, the spatiotemporal attributes of user i is represented as a set of daily convex hulls,

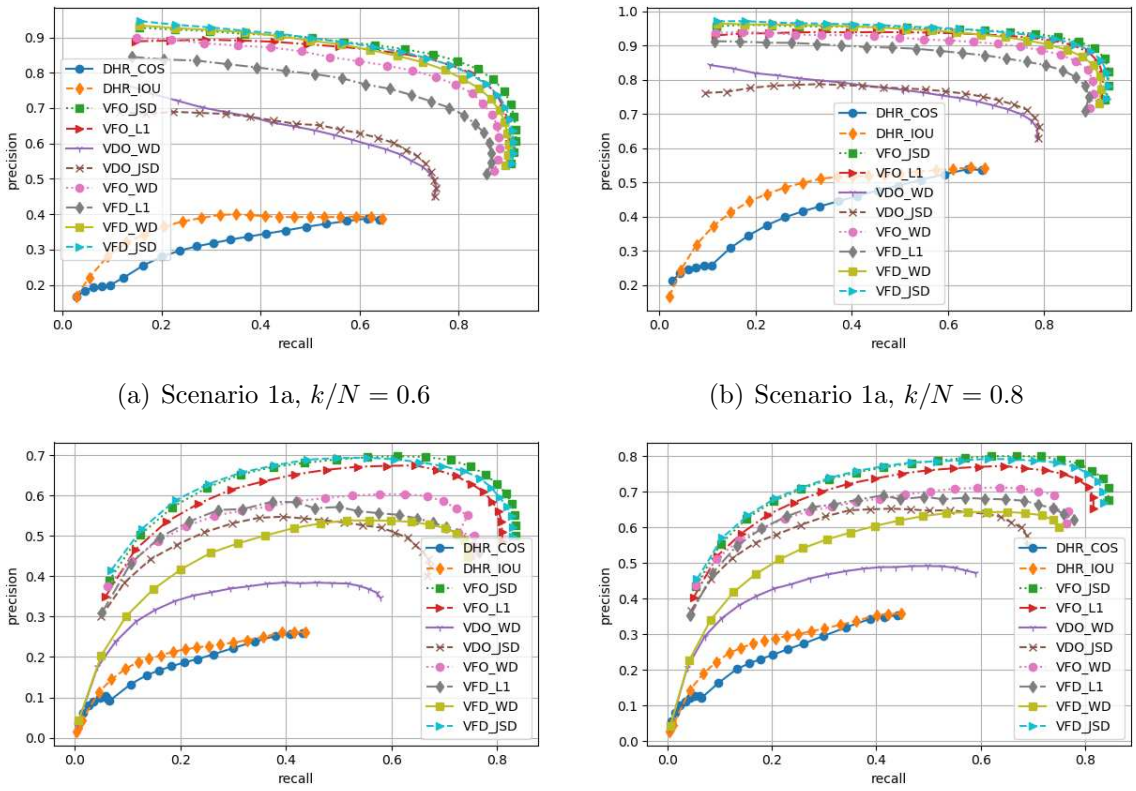
$$\mathcal{C}_i = \{C_{i1}, C_{i2}, \cdots, C_{iQ_i}\}, \tag{3.29}$$

where each convex hull is again assumed to be i.i.d. generated from an unknown probability distribution.

Distance Measures: With the convex hull set representing users' spatiotemporal behaviors, we first define two distance measures between two convex hulls based on the cosine distance and the intersection-over-union (IoU), respectively,

$$\begin{aligned} \delta_{\cos}(C_p, C_q) &= 1 - \frac{\text{area}(C_p \wedge C_q)}{\sqrt{\text{area}(C_p) \times \text{area}(C_q)}} , \\ \delta_{\text{iou}}(C_p, C_q) &= 1 - \frac{\text{area}(C_p \wedge C_q)}{\text{area}(C_p \vee C_q)} , \end{aligned} \tag{3.30}$$

where $C_p \wedge C_q$ and $C_p \vee C_q$ denote the intersection and union of the two convex hulls, respectively, and the operator area$(\cdot)$ is to calculate the area of a polygon. Therefore, a distance measure between two convex hull sets is proposed based on (3.30) to evaluate the



(a) Scenario 1a, $k/N = 0.6$

(b) Scenario 1a, $k/N = 0.8$

(c) Scenario 2, $k/N = 0.6$

(d) Scenario 2, $k/N = 0.8$

Figure 3.1: Performance comparison of different distance measures in terms of single matching, where VFO-JSD and VFO-L1 are originated from [2].

similarity of two subscribers as follows,

$$\begin{aligned}\Delta_{\text{DHR-COS}}(X_i, Y_j) &= \frac{1}{Q_i \times Q_j} \sum_{C_{ip} \in X_i} \sum_{C_{iq} \in Y_j} \delta_{\cos}(C_{ip}, C_{iq}), \\ \Delta_{\text{DHR-IOU}}(X_i, Y_j) &= \frac{1}{Q_i \times Q_j} \sum_{C_{ip} \in X_i} \sum_{C_{iq} \in Y_j} \delta_{\text{iou}}(C_{ip}, C_{iq}).\end{aligned} \tag{3.31}$$

Intuitively, the distance measure between two convex hull sets is to calculate the average distance between any two convex hulls in two respective sets. When the convex hull cannot be obtained because the number of distinct visited location points within a day is less than 3, the daily habitat region would be omitted. If no convex hull could be generated, the user will be labeled as non-identifiable.

3.5.4 User Grouping

The complexity of each distance measure between two users depends on the support of histogram or the number of points in two convex hulls. However, one needs to calculate $N \times M$ distances across two datasets so that a distance matrix $\boldsymbol{W}^g$ can be generated. The complexity of distance matrix generation, $\mathcal{O}(\psi NM)$, may lead to scalability issue when the user size of two datasets is tremendous. Here, ψ denotes the computational complexity of distance measures per pair. Inspired by the graph partitioning concept employed in the DS-Ensemble P&M algorithm, we propose to first cluster users into small groups so that the distance matrix generation and the ensemble matching within each group could be conducted.

The mobility feature of user i on each base station takes the form as follows,

$$\boldsymbol{f}_i = [f_{i1}, f_{i2}, \cdots, f_{iL}]^T \ . \tag{3.32}$$

In fact, f_{il} characterizes the mobility behavior of user i on location l, $f_{il} = \hat{\Pi}_{il} \times \log(N/n_l)$, where n_l is the number of users visiting location l out of total user N. The term $\log(N/n_l)$ is similar to inverse document frequency in the field of document clustering [39], which is designed to depict the importance of location points for clustering. In other words, if most users visit one location, the value of $\log(N/n_l)$ will be small, meaning that such location is less important to distinguish users.

Hence, one can obtain a feature matrix by stacking the feature of all the users from both datasets as follows,

$$\boldsymbol{F} = [\boldsymbol{f}_1, \cdots, \boldsymbol{f}_{(N+M)}] \in \mathcal{R}^{L \times (N+M)}, \tag{3.33}$$

where each column represents the mobility behavior of a user. Two characteristics of feature matrix $\boldsymbol{F}$ could be observed: 1) the number of base stations could be very large, up to $6,500$ in the studied dataset, due to a large geographic area studied; 2) the mobility feature of users $\boldsymbol{f}_i$ could be very sparse, as one user can most likely visit a small portion out of all the base stations. It is worth recalling that the objective of user grouping is to reduce the user set size for matching, whose complexity is $\mathcal{O}(NM)$. In other words, the computational complexity of the clustering algorithm cannot be as high as $\mathcal{O}(NM)$. Otherwise, direct user re-identification on the entire user set would be more meaningful. Besides, the high dimension of user mobility feature can lead to the uselessness of the commonly employed low-complexity k-means clustering algorithm.

As a result, the clustering algorithm based on non-negative matrix factorization (NMF) [39] is employed to cluster users in this work. The NMF is essentially to minimize the Frobenius norm of the difference between the original matrix and the multiplication of two non-negative factorized matrices as follows,

$$\begin{aligned} \underset{\boldsymbol{P},\boldsymbol{Q}}{\text{minimize}} \quad & \|\boldsymbol{F}-\boldsymbol{P}\boldsymbol{Q}\|_{\mathcal{F}} \\ \text{subject to} \quad & \boldsymbol{P}=[\boldsymbol{p}_1,\cdots,\boldsymbol{p}_R]\in\mathcal{R}_+^{L\times R} \\ & \boldsymbol{Q}=[\boldsymbol{q}_1,\cdots,\boldsymbol{q}_{(M+N)}]\in\mathcal{R}_+^{R\times(N+M)} \end{aligned} \quad , \tag{3.34}$$

where R denotes the factorization rank and also the number of user groups. Based on the non-negativity of both matrices, each user can be represented by the non-negative weights

$\boldsymbol{q}_i$ on group representations $\boldsymbol{P}$ as follows,

$$\boldsymbol{f}_i = \sum_{r=1}^{R} q_{ir} \boldsymbol{p}_r \ ,$$

where q_{ir} denotes the weight on group r of user i. As q_{ir} is non-negative, the user group for each user could be determined by finding the maximum user group weight as follows,

$$r_i = \arg\max_r q_{ir} \ . \tag{3.35}$$

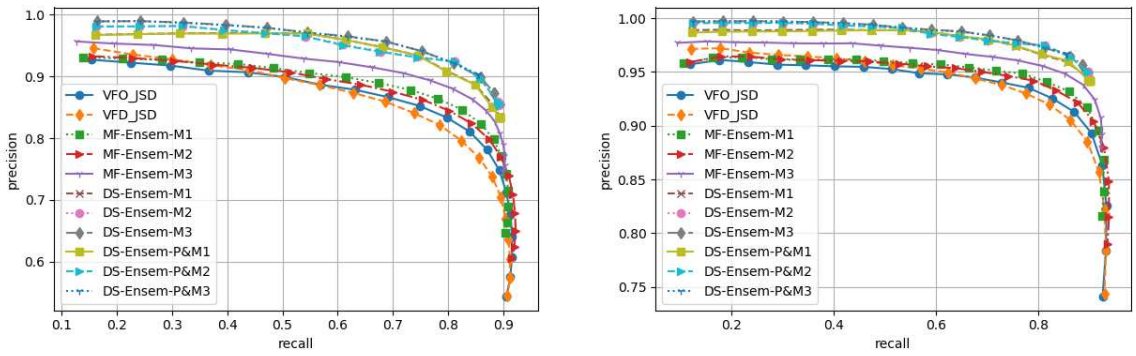
Therefore, one could obtain user grouping results with the complexity of $\mathcal{O}(R(N+M))$, once the feature matrix is factorized. In the literature, the multiplicative update method [39, 40] is commonly employed for NMF with the complexity of $\mathcal{O}(iR(N+M))$, where i denotes the overall iteration. As a result, by combining with the DS-Ensemble P&M algorithm, the complexity of the entire user re-identification procedure could be significantly less than $\mathcal{O}(NM)$.

3.6 Experiments

In this Section, we validate our proposed feature extraction, distance measures, and ensemble matching via experiments on a real-world signaling dataset collected in a mobile network, which is an extension of the commonly studied call detail record (CDR) dataset.

3.6.1 Test Scenarios

Two test scenarios are generated from the dataset to evaluate the proposed multi-feature ensemble matching framework and to compare with the existing methods in the literature.



(a) Scenario 1a, $k/N = 0.6$

(b) Scenario 1a, $k/N = 0.8$

(c) Scenario 2, $k/N = 0.6$

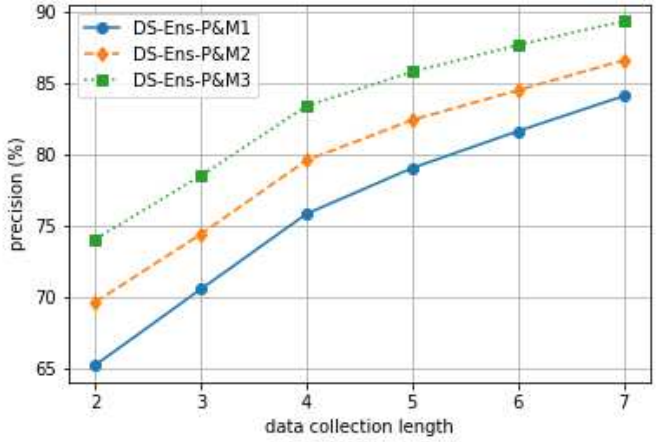
(d) Scenario 2, $k/N = 0.8$

Figure 3.2: Performance comparison of different distance measures in terms of ensemble matching, where VFO-JSD and VFO-L1 are originated from [2].

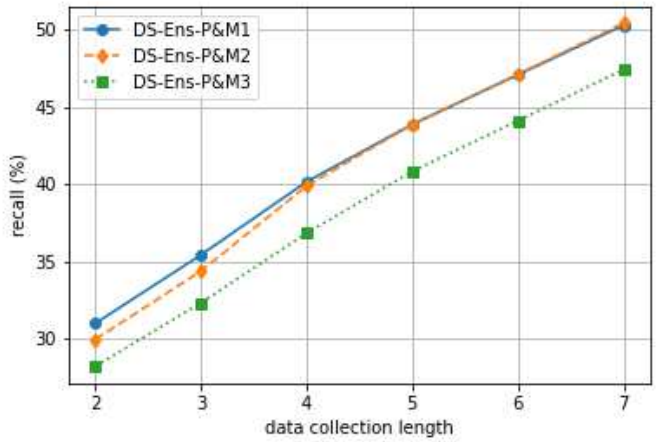
- *Scenario 1a.* 15,000 users are randomly selected out of the three million users for comparisons. The time span of the selected dataset is from July 1st, 2016 to July 14th, 2016. That is, the dataset $\mathcal{X}$ covers July 1st to July 7th, 2016, while the dataset $\mathcal{Y}$ covers July 8th to July 14th, 2016. The number of users in both datasets is 10,000 with different coexisting user number k specified later;

- *Scenario 1b.* 150,000 users are randomly selected out of three million users with different data collection periods for further large-scale user re-identification analysis.

- *Scenario 2.* This scenario is to mimic that the network operator publishes a region-based dataset. That is, a total of 5976 users are extracted based on their mobility



(a) Precision



(b) Recall

Figure 3.3: Large-scale user re-identification analysis on Scenario 1b.

behaviors in specific regions of interest for certain applications. The dataset $\mathcal{X}$ covers January 1st to January 7th, 2016, while the dataset $\mathcal{Y}$ covers January 8th to January 14th, 2016, where $M = N = 2,500$ with different k values specified in each experiment.

3.6.2 User Identification Performance

To evaluate the user identification performance, the classical *precision-recall* is employed. In general, the tradeoff between the precision and recall could be controlled by a preset

Table 3.2: Distance measures for each ensemble.

Ensemble 1	VFO-L1, VFO-JSD, VFD-JSD
Ensemble 2	VFO-L1, VFO-JSD, VFD-JSD, DHR-IOU
Ensemble 3	VFO-L1, VFO-JSD, VDO-JSD, VFD-JSD, DHR-IOU

parameter $\hat{k}$, which can be regarded as an estimate of the true number of coexisting users.

Fig. 3.1 shows the comparisons among different spatiotemporal feature extraction strategies and their corresponding distance measures. In both scenarios, it can be observed that the proposed VFD feature with JSD distance measure has the highest precision at low recall rates, while the VFD-JSD is slightly worse than the one VFO-JSD at high recall rates. The reason behind such a phenomenon is that the duration modeling could contribute to user identification if it is accurate, otherwise it may introduce noise. In both scenarios, the performance of DHRs is the worst among all compared features and their associated distance measures, because the DHR modeling is more sensitive to outliers and noises. However, DHR still can achieve more than 50% and 60% in Scenario 1a in terms of the precision and recall rate, respectively. Furthermore, the worst performance of DHR does not mean it is useless at all. It can provide a unique and distinct aspect to model a user, which can contribute to user identification in the proposed ensemble matching mechanism, as shown in Figs. 3.2.

Figs. 3.2 show the precision-recall comparison between the best single matching (VFO-JSD & VFD-JSD), MF-Ensemble Matching (Algorithm 1), DS-Ensemble Matching (Algorithm 2), and DS-Ensemble P&M (Algorithm 3) algorithms. The distance measures involved in each ensemble can be found in Table 3.6.1. It can be observed in both scenarios that the ensemble can broadly outperform the best individual matching in terms of the precision. The performance gain is more significant as one distance measure is not capable of distinguishing

certain user pairs, due to their similar residency areas. However, the proposed MF-Ensemble algorithm can effectively take advantage of multiple diverse distance measures.

The tradeoff between the maximum recall and precision rates can also be observed in Figs. 3.2. In other words, the proposed ensemble matching framework can achieve much higher precision at the same recall, while it has a smaller maximum recall compared with the individual ones (i.e., the absolute user pairs that the ensemble matching can discover is less than that of individual matchings). However, the maximum recall gap between the ensemble and the individual is shown negligible compared with the precision performance gain. Besides, the more distance measures involved in the ensemble can result in a better precision performance but with maximum recall performance slightly compromised.

It can be observed that the DS-Ensemble algorithm can achieve higher precision and trade off more maximum recall performance. The DS-Ensemble can achieve almost 100% precision at low recall rates and more than 95% at high recall rates. As a result, the DS-Ensemble algorithm can be viewed as the most reliable user identification algorithm. However, the reliability of DS-Ensemble comes at the cost of the maximum recall performance, especially when users are similar to each other, as shown in Scenario 2. The proposed low-complexity DS-Ensemble P&M algorithm has a similar performance as the DS-Ensemble matching, as the size of most subgraphs after partitioning without loss is less than the threshold ($t = 1,000$ in Algorithm 3).

3.6.3 Privacy Evaluation

Figs. 3.3 shows the precision and recall performance of large-scale user re-identification analysis on a $150,000$ user set, based on the proposed DS-Ensemble P&M algorithm. In

Figs. 3.3, the length of user data collection ranges from 2 to 7 days (x axis). For complexity reduction, the entire user set is first partitioned by user grouping, as discussed in Section 3.5.4. The user grouping would lead to the loss of recall performance, since some users may be clustered into different groups. The incorrect clustering rate ranges from 16.36% (7-day data collection) to 19.64% (2-day data collection). It can be observed that both the recall and precision performance can be improved as the data collection length grows, which suggests the reduction of data collection could lead to privacy protection to some degree. Overall, the subscriber privacy is vulnerable in terms of user identifiability across two datasets, if the dataset is released only with ID anonymization. In Figs. 3.3, it shows that the user privacy is still at high risk, as the proposed DS-Ensemble P&M algorithm still can recognize almost half of the user pairs at very high confidence (up to 90%).

3.7 Summary

In this work, we studied the privacy attack in terms of user re-identifiability across two datasets based on the spatiotemporal data collected from mobile networks. With the LAP formulation, a scalable multi-feature ensemble matching framework was proposed. In this work, we proposed to extract two new semantic spatiotemporal features as well as their associated distance measures. With multiple matching results via the diverse features, a scalable ensemble matching framework was proposed to fuse matching results so that the final result is reliable and robust. Experiments demonstrated that our proposed multi-feature ensemble matching achieves superior performance (up to 100% precision), which also suggested the vulnerability of mobile network subscriber's privacy.

3.8 Distance Measure Derivation for VFD

The likelihood that two tuple strings X_i and Y_j by (3.21) takes the form based on the assumption that both are generated by the same user as follows,

$$\mathcal{L}(X_i, Y_j|\Pi_{ij}, \Lambda_{ij}) = \begin{array}{l} \prod_{p=1}^{P_i} f(t_{ip}|x_{ip} = a_l; \lambda_{ij,l})\Pi_{ij}(x_{ip}) \\ + \prod_{p=1}^{P_j} f(t_{jp}|y_{jp} = a_l; \lambda_{ij,l})\Pi_{ij}(y_{jp}) \end{array}, \tag{3.36}$$

where Λ_{ij} denotes the collection of $\lambda_{ij,l}$. By the maximum likelihood estimation, the empirical probability distribution estimate $\widehat{\Pi}_{ij}$ is

$$\widehat{\Pi}_{ij}(a_l) = q_i\Gamma_{X_i} + q_j\Gamma_{Y_j} = \frac{N_{ij}(a_l)}{P_{ij}},\ \forall a_l \in \mathcal{A}, \tag{3.37}$$

where $q_i = P_i/P_{ij}$, $q_j = P_j/P_{ij}$, $P_{ij} = P_i + P_j$, and $N_{ij}(a_l) = N_i(a_l) + N_j(a_l)$. Furthermore, each parameter in set $\widehat{\Lambda}_{ij}$ takes the form as,

$$\widehat{\lambda}_{ij,l} = \frac{1}{k_{i,l}/\widehat{\lambda}_{i,l} + k_{j,l}/\widehat{\lambda}_{j,l}} = \frac{N_{ij}(a_l)}{\tilde{t}_{i,l} + \tilde{t}_{j,l}},\ \forall a_l \in \mathcal{A}, \tag{3.38}$$

where $\tilde{t}_{i,l} = \sum_{p,z_{ip}=a_l} t_{ip}$ and $\tilde{t}_{j,l} = \sum_{p,z_{jp}=a_l} t_{jp}$ denote the sum of durations at a_l of users i and j, respectively. $k_{i,l} = N_i(a_l)/N_{ij}(a_l)$ and $k_{j,l} = N_j(a_l)/N_{ij}(a_l)$ denote the weights. Here, $\hat{\lambda}_{i,l}$ and $\hat{\lambda}_{j,l}$ are maximum likelihood estimates of X_i and Y_j, respectively. Based on the multi-hypothesis test framework in [1], the log likelihood of hypothesis $\mathcal{H}_r$ could be expressed

as follows,

$$\begin{aligned}\mathcal{L}(\mathcal{H}_r) = \sup_{\substack{\Pi_{ij},\Pi_i,\Pi_j,\\ \Lambda_{ij},\Lambda_i,\Lambda_j}} & \sum_{(i,j)\in\Phi_r} \log\left[\mathcal{L}(X_i, Y_j|\Pi_{ij}, \Lambda_{ij})\right] \\ & + \sum_{(i,j)\notin\Phi_r} \log\left[\mathcal{L}(X_i|\Pi_i, \Lambda_i)\right] + \log\left[\mathcal{L}(Y_j|\Pi_j, \Lambda_j)\right]\end{aligned} \tag{3.39}$$

where $\mathcal{L}(X_i, Y_j|\Pi_{ij}, \Lambda_{ij}) = \mathcal{L}(X_i|\Pi_{ij}, \Lambda_{ij})\mathcal{L}(Y_j|\Pi_{ij}, \Lambda_{ij})$ based on the i.i.d. assumption.

The likelihood function (3.36) could be rewritten in terms of two components, namely visiting frequency and location-dependent duration as follows,

$$-\frac{\mathcal{L}(X_i, Y_j|\Pi_{ij}, \Lambda_{ij})}{P_{ij}} = \mathcal{L}_{\text{freq}}(\widehat{\Pi}_i, \widehat{\Pi}_j) + \mathcal{L}_{\text{dura}}(\widehat{\Pi}_i, \widehat{\Pi}_j, \hat{\Lambda}_{ij}). \tag{3.40}$$

The first part, $\mathcal{L}_{\text{freq}}$, can be obtained on the frequency features in terms of unequal string lengths,

$$\mathcal{L}_{\text{freq}}(\widehat{\Pi}_i, \widehat{\Pi}_j) = q_i H(\widehat{\Pi}_i) + q_j H(\widehat{\Pi}_j) + \text{wdiv}_{q_i}(\widehat{\Pi}_i \| \widehat{\Pi}_j)\ . \tag{3.41}$$

The second component $\mathcal{L}_{\text{dura}}$ is related to duration modeling, representing the weighted sum of the cross entropy between the exponential distribution at each location point a_l. With parameter estimates $\widehat{\Pi}_{ij}$ and $\widehat{\Lambda}_{ij}$, $\mathcal{L}_{\text{dura}}(\widehat{\Pi}_{ij}, \widehat{\Lambda}_{ij})$ (See Appendix 3.9) could be easily obtained based on (3.41) and (3.44) as follows,

$$\mathcal{L}_{\text{dura}}(X_i, Y_j|\widehat{\Pi}_{ij}, \widehat{\Lambda}_{ij}) = \sum_{a_l\in\mathcal{A}} \widehat{\Pi}_{ij}(a_l)\left(1 - \log\hat{\lambda}_{ij,l}\right). \tag{3.42}$$

With $\frac{1}{\hat{\lambda}_{ij,l}} = \frac{k_{i,l}}{\hat{\lambda}_{i,l}} + \frac{k_{j,l}}{\hat{\lambda}_{j,l}}$, $\mathcal{L}_{\text{dura}}$ could be further expressed at each a_l as follows,

$$\frac{\mathcal{L}_{\text{dura}}(a_l)}{\widehat{\Pi}_{ij}(a_l)} = k_{i,l}\left[\frac{\hat{\lambda}_{ij,l}}{\hat{\lambda}_{i,l}} - \log\hat{\lambda}_{ij,l}\right] + k_{j,l}\left[\frac{\hat{\lambda}_{ij,l}}{\hat{\lambda}_{j,l}} - \log\hat{\lambda}_{ij,l}\right].$$

The differential entropy of exponential distributions is $H(\lambda) = 1 - \log\lambda$. The KL divergence between two exponential distributions, λ_1 and λ_2, is $\mathrm{KL}(\lambda_1 \| \lambda_2) = \log(\lambda_1/\lambda_2) + \lambda_2/\lambda_1 - 1$. Therefore, $\mathcal{L}_{\text{dura}}$ could be further rewritten in terms of entropies and KL divergences as follows,

$$\mathcal{L}_{\text{dura}} = \sum_{a_l \in \mathcal{A}} \widehat{\Pi}_{ij}(a_l) \Big\{ k_{i,l} \left[H(\hat{\lambda}_{i,l}) + \mathrm{KL}(\hat{\lambda}_{i,l} \| \hat{\lambda}_{ij,l}) \right] + k_{j,l} \left[H(\hat{\lambda}_{j,l}) + \mathrm{KL}(\hat{\lambda}_{j,l} \| \hat{\lambda}_{ij,l}) \right] \Big\} .$$

With $\widehat{\Pi}_{ij}(a_l) k_{i,l} = q_i \widehat{\Pi}_{i,l}$ and $\widehat{\Pi}_{ij}(a_l) k_{j,l} = q_j \widehat{\Pi}_{j,l}$, $\mathcal{L}_{\text{dura}}$ could be further expressed as follows,

$$\begin{aligned} &\mathcal{L}_{\text{dura}}(\widehat{\Pi}_i, \widehat{\Pi}_j, \widehat{\Lambda}_i, \widehat{\Lambda}_j) \\ &\quad = \sum_{a_l \in \mathcal{A}} q_i \widehat{\Pi}_{i,l} \left[H(\hat{\lambda}_{i,l}) + \mathrm{KL}(\hat{\lambda}_{i,l} \| \hat{\lambda}_{ij,l}) \right] . \\ &\qquad + q_j \widehat{\Pi}_{j,l} \left[H(\hat{\lambda}_{j,l}) + \mathrm{KL}(\hat{\lambda}_{j,l} \| \hat{\lambda}_{ij,l}) \right] \end{aligned} \tag{3.43}$$

Similarly as in [1], the entropy part of (3.41) and (3.43) could be eliminated for it is a constant for all the hypotheses. To determine the most likely hypothesis is to perform a k-cardinality minimum cost bipartite matching, where the edge weights are the pair-wise distance measure via both frequency and duration modeling. Thus, (3.25) and (3.26) could be easily obtained by keeping the divergence parts in (3.41) and (3.43).

3.9 MLE Derivation of Concatenated Tuple String

By the assumption that each observation is drawn i.i.d. and users are uncorrelated, the log likelihood of any tuple string pair (3.36) under the assumption that the pair are generated

by the same user could be studied independently as follows,

$$\begin{aligned}
&\mathcal{L}(X_i, Y_j|\Pi_{ij}, \Lambda_{ij}) = \\
&\sum_{a_l \in \mathcal{A}} N_i(a_l) \left[\log(\Pi_{ij}(a_l)) + \log \lambda_{ij,l} - \lambda_{ij,l}\bar{t}_{i,l}\right] \\
&\quad + N_j(a_l) \left[\log(\Pi_{ij}(a_l)) + \log \lambda_{ij,l} - \lambda_{ij,l}\bar{t}_{j,l}\right] \ ,
\end{aligned} \tag{3.44}$$

where $\bar{t}_{i,l} = \frac{\sum_{p, z_{ip}=a_l} t_{ip}}{N_i(a_l)}$ and $\bar{t}_{j,l} = \frac{\sum_{p, z_{jp}=a_l} t_{jp}}{N_j(a_l)}$ denote the average time length of users i and j at location point a_l, respectively. The problem of maximum log likelihood (3.44) is:

$$\begin{aligned}
\underset{\Pi_{ij}, \Lambda_{ij}}{\text{maximize}} \quad & \sum_{a_l \in \mathcal{A}} [N_i(a_l) + N_j(a_l)][\log \Pi_{ij}(a_l) + \log \lambda_{ij,l}] \\
& - \lambda_{ij,l}[N_i(a_l)\bar{t}_{i,l} + N_j(a_l)\bar{t}_{j,l}] \\
\text{subject to} \quad & \sum_{a_l \in \mathcal{A}} \Pi_{ij}(a_l) = 1
\end{aligned} \tag{3.45}$$

It could be observed that the empirical probability distribution Π_{ij} is independent from the exponential distribution of duration for any given location points. Therefore, the estimate of Π_{ij} could be first obtained as (3.37) by optimizing (3.45) in terms of Π_{ij}. Furthermore, $\widehat{\lambda}_{ij,l}$ could be obtained as (3.38) by optimizing (3.45) with respect to $\lambda_{ij,l}$ for each $a_l \in \mathcal{A}$.

CHAPTER 4

AGGREGATED SPATIOTEMPORAL MODELING: DEMAND FORECASTING FOR TRAFFIC MANAGEMENT

4.1 Background

[1]The mobile big data collected by mobile network operators can also benefit the management of mobile networks. Mobile big data could help uncover and understand user' behavior patterns [64] via effective data mining techniques, which could benefit to the resource-constraint network optimization, from network planning, network traffic monitoring to network management. In recent years, self-organizing networks (SON) is widely studied to automatically manage and organize networks without manual intervention [65, 66]. One motivation to employ SONs in cellular networks is the reduction of network operational expenditures (OPEX) and capital expenditures (CAPEX), which requires full exploitation of the capability of network infrastructure. The demand forecasting will play an essential role in providing predictive knowledge [67] in various cellular SON functions, especially for the future cellular networks with the virtualization and cloudization of network functions [68,69].

In this work, we study the mobile demand forecasting, the foundation of predictive mobile network management. In the literature, the mobile traffic/demand forecasting schemes have been studied for traffic apprehension and prediction via the Holt-Winter's exponential smoothing technique [70], information theory [71], and the seasonal ARIMA model [72]. However, all these demand forecasting models only consider the temporal aspect via various

[1]Parts of this work have been published in [10, 63].

time series models without taking into account the spatial relevancy among cells. Models of mobile demand forecasting accounting for the spatial relevancy have been recently studied based on deep learning [73, 74]. In these models, the temporal aspect of demand time series is commonly studied via the recurrent neural networks (RNNs), while the spatial relevancy is captured by various grid-based spatial models.

However, the main challenge of applying grid-based spatial models to per-cell demand forecasting is the uneven spatial distribution of cells in the real-world setting. Generally, the cell towers are distributed in a network covered area according to the population density. That is, the distance between two cell towers is about 500 meters in the urban area, but can reach 2000 meters in the rural area. Hence, grid-based models [73, 74] do not directly apply. To utilize the grid-based models, one first needs to redivide the network covered area into a uniform square grid, and then predict the aggregated demands of multiple cell towers residing in each lattice. Such spatial area re-division and demand aggregation will lead to the loss of the spatial granularity and will significantly limit the applications to future cellular network management that requires variable spatial granularity.

To this end, we propose a flexible *graph-based spatial model* for the per-cell demand forecasting without any spatial resolution degradation and data aggregation. First, we realize that the spatiotemporal analysis of the per-cell demand time series via the semivariogram [75] reveals that the relevancy between the demands of two cells relies on the spatial distance of the two cells. That is, the dependency level of two cells would decrease when their spatial distance increases. Hence, we can build a dependency graph characterizing the relevancy of cells based on their spatial distances. In other words, the per-cell demands generated at each cell tower could be regarded as signals generated at the vertices of a graph. Also, not only

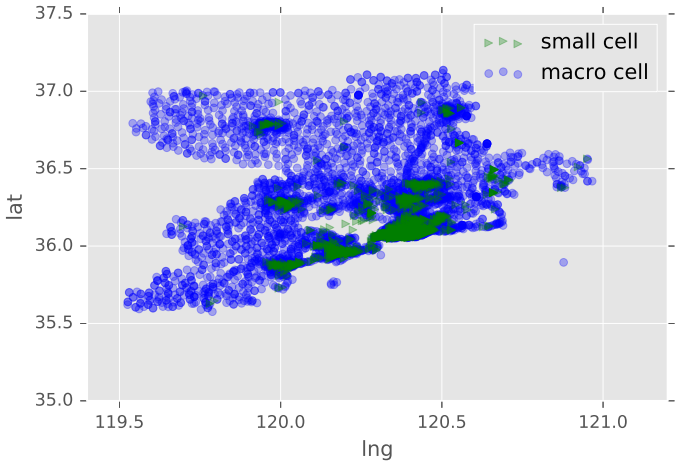


Figure 4.1: Cell distribution heatmap.

the recent demand history is applied to forecast the future demands, but also the periodic history (e.g., day(s) ahead demands) are considered in order to obtain an accurate demand predictor.

With the dependency graph formulation, the recently developed graph convolutional networks (GCNs) [76, 77] and the long short-term memory (LSTM) neural networks [78] are employed to characterize the spatial aspect and the temporal aspect for demand forecasting, respectively. The LSTM is a gated version of recurrent neural networks (RNNs) in deep learning, which is well known for its excellent performance on sequence modeling. In GCNs, the graph convolution operation originated from signal processing theory on graphs [79–81], is employed to replace the matrix multiplication in the feedforward neural networks. The power of graph convolution comes from the parameter sharing and sparse interaction techniques, which have been discussed in the traditional convolutional neural networks (CNN) [82]. The sparse interaction in per-cell demand prediction means that the demand prediction of one cell is only related to itself and its nearest neighbors in the dependency graph. The parameter sharing assumes that the model parameters are shared across all cells of the network.

In this work, we first formulate the demand forecasting problem as a one-step-ahead demand prediction problem. The demand forecasts after one step in the future are dynamically generated by the one-step-ahead predictor. Three models, namely the spatial-only (GCNs), the temporal-only (LSTM), and the spatiotemporal (GCLSTM), are studied. The graph convolutional LSTM (GCLSTM) [83] is the model replacing the matrix multiplication operation with the graph convolution operation in LSTM, inspired by the convolutional LSTM (convLSTM) [84]. Compared with GCLSTM, LSTM without the embedded spatial information will predict the demand of one cell based on all other cells in the network, which would lead to an inferior generalization performance. Experiments show that the temporal-only LSTM could achieve a superior performance for the very-short-term demand forecasting for its much larger model capacity but rapidly deteriorates when the forecast horizon increases. This results from the inferior generalization performance of LSTM and the accumulated errors in the generated predicts. The GCLSTM with the spatial and the temporal aspects modeled will generally have a superior forecast performance except for the very-short-term one. Main contributions of this work are summarized as follows,

- To the best of the authors' knowledge, this is the first work modeling the spatial relevancy among cells by a dependency graph. The graph-based spatial modeling could completely retain the spatial granularity without any data aggregation.

- The periodicity of the per-cell demand time series is explicitly taken into account by adding past periodic observations as input features in our studied models so that the accuracy of demand forecasting could be enhanced without significantly increasing model size.

- The graph convolutional and recurrent neural networks are proposed to simultaneously characterize both the spatial and temporal attributes with parameter sharing, which could lead to a superior generalization performance of demand forecasting.

4.2 Per-Cell Demands

Based on the studied signaling dataset, two categories of service demands could be extracted, namely communication demands and tracking demands. The communication demands include the first 4 events on calls and texts recorded in the signaling dataset, to forecast which is the very task of this work. The tracking demands could be obtained based on the location update events, which is closely related to crowd mobility. The location update frequency is once per hour, which may be too coarse to exactly describe the crowd flow, especially in the urban area (where cells are densely distributed). Hence, we focus on the communication demand forecasting in this work.

With the spatiotemporal information of each event recorded, we define the per-cell demand as the number of communication events occurring within a cell during an event counting time window ΔT. Hence, a per-cell demand time series could be generated as follows,

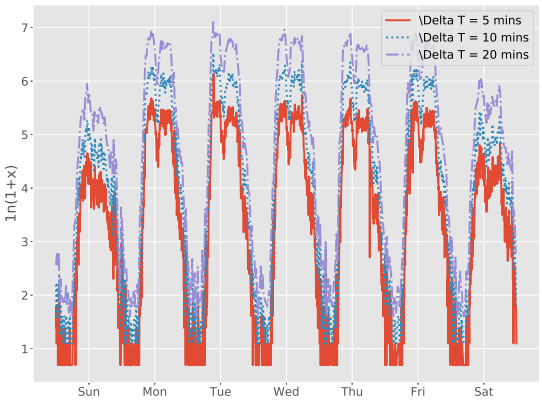
$$\left[x_t^n, x_{t-1}^n, x_{t-2}^n, \cdots, x_{t-l+1}^n, \cdots\right], \tag{4.1}$$

where $x_t^n = \ln(1 + c_t^n)$ denotes the per-cell demand within time window $[t - \Delta T, t)$, where c_t^n is the number of communication events of the n-th cell. Here, we utilize the commonly used logarithm function $\ln(1 + x)$ to convent the integer event number domain to the real number domain of demands. In this work, we mainly study the demand forecasting in terms of the

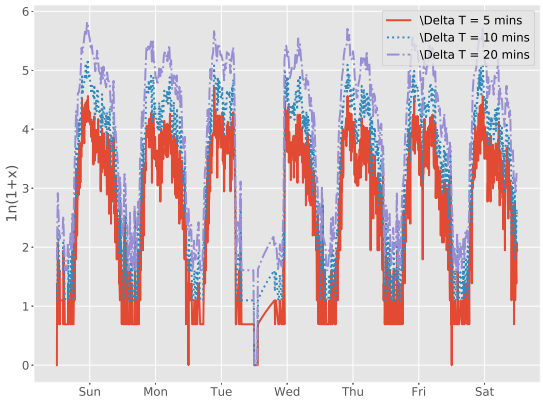
10-minute counting time windows, i.e., $\Delta T = 10$.

It can be observed that small cells are densely deployed in the studied urban area (green areas as shown in Fig. 4.1). In heterogeneous cellular networks, small cells are designed to assist their corresponding macro cell by offloading data traffic, whose coverage is also relatively much smaller than that of macro cells. As a result, the communication demands of small cells is sparse, which is not of interest in this work. Hence, we aggregate the demand of small cells to its corresponding macro cell, which is determined by their spatially closest macro cell based on the location information (i.e., the longitude and latitude of cell towers). In other words, we study the per-cell aggregated demands within a spatial area covered by a macro cell.

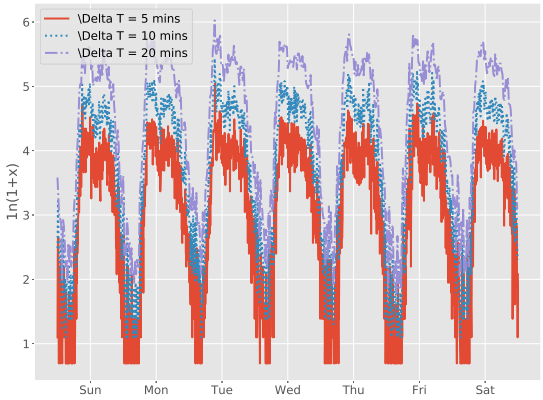
In Fig. 4.2, the per-cell demands with different cell types are illustrated, namely business, entertainment, and residence. In each subfigure, three demand time series with different counting time window are plotted, $\Delta T = 5$ minutes, $\Delta T = 10$ minutes, and $\Delta T = 20$ minutes. One can easily observe that the large counting time window could significantly reduce the noise of the per-cell demand time series, as the larger counting time window acts like a smoothing filter applied on the one generated by the small counting time window. However, such noise reduction is at the cost of lowering the temporal resolution of demand time series. Besides, it can be easily observed that per-cell demands are strongly periodic in terms of calendar days, regardless of cell types. Another periodic effect that the demands during weekends are less than those during weekdays could be observed from the demand time series of the business type (Fig. 4.2(a)). Such effects would inspire the feature engineering for demand forecasting, which will be discussed in detail later.



(a) 7-day demands, business



(b) 7-day demands, entertainment



(c) 7-day demands, residential

Figure 4.2: Demand time series of various cell type, where the 7-day demands are recored from Nov. 27th, 2016 to Dec. 3rd, 2016 and 24-hour demands are recorded on Nov. 27th, 2016. The business cell is located in the central business district (CBD), the entertainment-type cell is located in a public park, and the residence area is located in a large residential area.

4.3 Demand Prediction Problem Formulation

With the definition of per-cell demands, the demand forecasting is aimed to predict the per-cell demands of all cells in a mobile network based on its history. In this work, demand forecasting is studied as the one-step ahead prediction problem as follows,

$$\hat{\boldsymbol{x}}_{t+1} = f\left(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \cdots, \boldsymbol{x}_{t-l+1}, \cdots\right) \tag{4.2}$$

where $\boldsymbol{x}_t = [x_t^1, x_t^2, \cdots, x_t^N]^T$ denotes the per-cell demands of cells across the covered area at time t and N is the total number of macro cells in the network. Hence, the prediction problem essentially amounts to the estimation of a function or predictor f based on the collected history data and the knowledge of cell locations. In this section, we will discuss the one-step-ahead demand prediction with innovative spatiotemporal modeling.

4.3.1 Graph-Based Spatial Formulation

By the spatiotemporal analysis of multiple per-cell demand time series (see Section 4.5.2, Appendix 4.8), it can be concluded that the demand relevancy between two cells declines when their spatial distance increases. Hence, we first propose to model the spatial relevancy between cells in the network by a dependency graph. The adjacency matrix $\boldsymbol{A}$ of the dependency graph can be obtained based on the spatial distance between cells as follows,

$$\boldsymbol{A}_{ij} = \begin{cases} 1, & \text{dist}(s_i, s_j) \leq \zeta \\ 0, & \text{otherwise} \end{cases}, \tag{4.3}$$

where s_i denotes the location of cell i and ζ is the threshold, a hyperparameter that could be tuned. We set $\zeta = 2$ km in this work. The threshold suggests that any two cells whose distance is beyond the threshold will be considered irrelevant. Such graph modeling could successfully make the cell relevancy sparse (from N^2 to $\sum_{i,j} \boldsymbol{A}_{i,j}$), which can lead to a good demand forecasting generalization performance with the graph modeled in the predictor as detailed in Sections 4.4 and 4.5. As a result, each cell could be regarded as a vertex in the spatial dependency graph and the per-cell demand $\boldsymbol{x}_t$ is viewed as the signal observed at

each vertex of the graph at time t.

4.3.2 Periodicity-Based Temporal Features

As shown in Fig. 4.2, it is obvious that the per-cell demand time series is periodic concerning calendar days or weeks. Such periodicity could provide valuable information for one-step-ahead per-cell demand prediction at time t. Accordingly, we could reformulate the per-cell demand time series in terms of calendar days at time t as follow,

$$\begin{bmatrix} x_t^i & x_{t-1}^i & \cdots & x_{t-L+1} & \cdots \\ x_{t-n_d}^i & x_{t-1-n_d}^i & \cdots & x_{t-L+1-n_d} & \cdots \\ x_{t-2n_d}^i & x_{t-1-2n_d}^i & \cdots & x_{t-L+1-2n_d} & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots \\ x_{t-7n_d}^i & x_{t-1-7n_d}^i & \cdots & x_{t-L+1-7n_d} & \cdots \\ \ddots & \ddots & \ddots & \ddots & \ddots \end{bmatrix},$$

where n_d denotes the number of per-cell demand observations in one calendar day. To predict x_{t+1}^i, not only the recent demand history $[x_t^i, x_{t-1}^i, \cdots, x_{t-L+1}^i]$ of cell i is taken into accounts, but also their corresponding days ahead demand observations will be regarded as input features for a predictor. Here, we only take the one-day ahead and 6-day ahead observations as the extra features in order to make the predictor more dependent on the current trend. Hence, the input features of all cells in the network at time t take the form,

$$\boldsymbol{Z}_t = [\boldsymbol{z}_t^1, \boldsymbol{z}_t^2, \cdots, \boldsymbol{z}_t^N]^T, \tag{4.4}$$

where $\boldsymbol{z}_t^i$ denotes the input features of cell i at time t, i.e., $\boldsymbol{z}_t^i = [x_t^i, x_{t-n_d}^i, x_{t-7n_d}^i]$.

4.3.3 Graph-Sequence Demand Prediction Formulation

Based on the spatial and temporal modeling discussed above, the one-step ahead demand prediction problem could be further expressed as

$$\hat{\boldsymbol{x}}_{t+1} = f\left(\boldsymbol{Z}_t, \boldsymbol{Z}_{t-1}, \cdots, \boldsymbol{Z}_{t-L+1}; \boldsymbol{A}\right) \tag{4.5}$$

where L is the length of recent history used for demand prediction. We will discuss the selection of L in Section 4.5. In this work, we employ the commonly used mean absolute predicted error (MAE) as the evaluation criterion and cost function. Hence, the demand prediction problem could be expressed as follows,

$$\min_f \ \frac{\mathbb{1}^T \mathrm{E}\left[|\boldsymbol{x}_{t+1} - \hat{\boldsymbol{x}}_{t+1}|\right]}{N}. \tag{4.6}$$

Next, we will discuss the proposed per-cell demand predictor with effective graph and sequence information embedded based on deep learning.

4.4 Deep Graph-Sequence Spatiotemporal Modeling

In this work, the graph-based (GCN) model and the sequence-based model (LSTM) are first proposed to capture the spatial and temporal aspects, respectively. Also, we study their integrated version (GCLSTM), which embeds the graph information in the sequence model.

4.4.1 Spatial Modeling - Graph Convolutional Networks

The graph convolution is the convolution operation in graph signal processing domain, defined as $g_\theta(\boldsymbol{L}) \star \boldsymbol{x}_t$, where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ denotes the graph Laplacian and $g_\theta(\boldsymbol{L})$ denotes a filter with respect to the graph $\boldsymbol{L}$. The graph convolution would relate the signal of one vertex to others in terms of the graph topology, where the corresponding graph filter coefficients could be trainable based on data. Details of the graph convolution and graph filter description refer to Appendix 4.7.

As only the nearest neighbors are considered in this work, the first-order graph filter based on (4.19) , $g_\theta^{(1)}(\boldsymbol{\Lambda}) = \theta_0 + \theta_1 \boldsymbol{\Lambda}$, is considered. In [76], Kipf and Welling proposed a simple first-order graph filer approximation based on Chebyshev polynomials of first kind [77] by forcing $\theta = \theta_0 = -\theta_1$ as follows,

$$g_\theta^{(1)}(\widetilde{\boldsymbol{L}}) \star \boldsymbol{x}_t = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{x}_t \theta \tag{4.7}$$

where $\widetilde{\boldsymbol{A}} = \boldsymbol{I} + \boldsymbol{A}$ and $\widetilde{\boldsymbol{D}}$ is a diagonal matrix, $\widetilde{\boldsymbol{D}}_{ii} = \sum_j \boldsymbol{A}_{ij}$.

Therefore, a graph convolutional network could be built based on the approximated first-order graph convolution operation to replace the matrix multiplication in the feedforward neural networks, which embeds the prior knowledge of graph topology into the learning model. As a result, each layer of graph convolutional networks is defined as[2]

$$\boldsymbol{H}^{l+1} = \sigma\left(\widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{H}^l \boldsymbol{\Theta}^l\right) \tag{4.8}$$

[2]For simplicity, we ignore the bias terms in the presentation of each studied model.

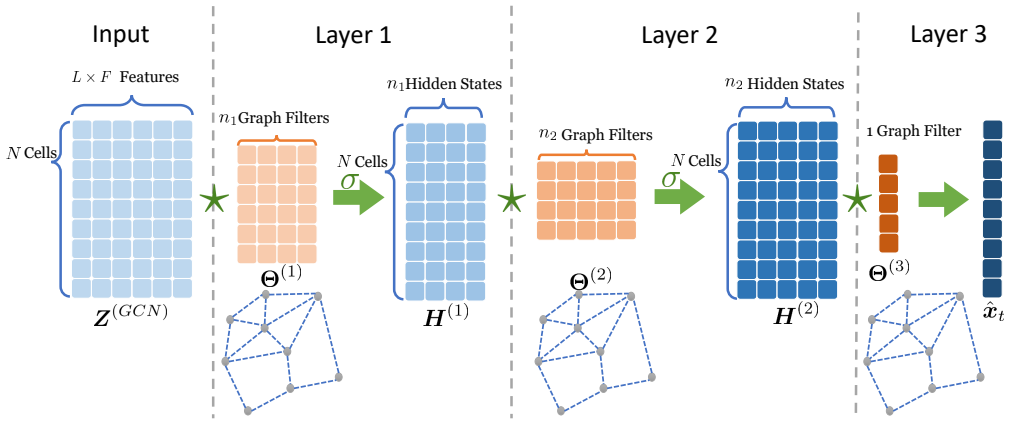


Figure 4.3: Spatial modeling: graph convolutional networks (GCN).

where $\sigma(\cdot)$ denotes the activation function for nonlinearity modeling. $\boldsymbol{H}^l \in \mathcal{R}^{N \times n_l}$ denotes the inputs of the l-th layer and $\boldsymbol{\Theta}^l \in \mathcal{R}^{n_l \times n_{l+1}}$ is the trainable parameters in the model. Again, N denotes the number of vertices of the graph. In each graph convolution operation, the $\boldsymbol{H}^l\boldsymbol{\Theta}^l$ in (4.8) is first to learn the pattern in a cell-wise manner with shared parameters $\boldsymbol{\Theta}^l$. The product of $\boldsymbol{H}^l\boldsymbol{\Theta}^l$ and $\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$ is essentially equivalent to the weighted sum over the cell and its first-order neighbors.

In the context of the per-cell demand prediction problem, we propose a three-layer graph convolutional network as the demand predictor f as detailed in Model 1 and Fig. 4.3.

Model 1 (Graph Convolutional Networks (GCN)) *A per-cell demand predictor is approximated by a three-layer graph convolutional network,* $\hat{\boldsymbol{x}}_{t+1} = \hat{f}(\boldsymbol{Z}_t^{(GCN)}, A)$*, i.e.,*

$$\begin{aligned}
&\textit{Layer 1:} \quad \boldsymbol{H}^{(1)} = \sigma\left(\widehat{\boldsymbol{A}}\boldsymbol{Z}_t^{(GCN)}\boldsymbol{\Theta}^{(1)}\right), \quad \boldsymbol{\Theta}^{(1)} \in \mathcal{R}^{(L\times F)\times n_1} \\
&\textit{Layer 2:} \quad \boldsymbol{H}^{(2)} = \sigma\left(\widehat{\boldsymbol{A}}\boldsymbol{H}^{(1)}\boldsymbol{\Theta}^{(2)}\right), \quad \boldsymbol{\Theta}^{(2)} \in \mathcal{R}^{n_1\times n_2} \\
&\textit{Layer 3:} \quad \hat{\boldsymbol{x}}_{t+1} = \widehat{\boldsymbol{A}}\boldsymbol{H}^{(2)}\boldsymbol{\Theta}^{(3)}, \quad \boldsymbol{\Theta}^{(3)} \in \mathcal{R}^{n_2\times 1}
\end{aligned} \tag{4.9}$$

where $\widehat{\boldsymbol{A}} = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}}\widetilde{\boldsymbol{A}}\widetilde{\boldsymbol{D}}^{-\frac{1}{2}}$ *and* $\boldsymbol{Z}_t^{(GCN)}$ *denotes the input of the GCN with L-length window,*

Here, $\boldsymbol{Z}_t^{(GCN)}$ is the L-length demand history with days ahead features as the input, i.e.,

$$\boldsymbol{Z}_t^{(GCN)} = [\boldsymbol{Z}_t, \cdots, \boldsymbol{Z}_{t-L+1}].$$

In other words, the L-length demand history and extra days ahead features of each cell are regarded as its input features of GCNs without the explicit sequence modeling. As a result, the total number of free trainable parameters in the proposed three-layer GCN is $n_{h_1}(L \times F) + n_{h_1} n_{h_2} + n_{h_2}$.

4.4.2 Temporal Modeling - Long Short-Term Memory (LSTM)

In the literature, the recurrent neural networks (RNNs) is proved to be an effective sequence model [85], which is designed to capture the sequential information inherited in data, e.g., audio, natural language, etc. Essentially, RNNs adds a feedback path in the feedforward neural networks, which could provide the information of the previous inputs so that the current output is not only dependent on the current inputs but also relies on the hidden state learned from previous inputs as follows,

$$\boldsymbol{h}_t = \sigma\left(\boldsymbol{W}\boldsymbol{z}_t + \boldsymbol{V}\boldsymbol{h}_{t-1}\right), \tag{4.10}$$

where $\boldsymbol{h}_{t-1}$ denotes the hidden states updated previously.

The long short-term memory networks (LSTM) is one of special designed RNNs, which has a capability of controlling the updating process by adding three gates, namely input gate

$\boldsymbol{g}_i$, forget gate $\boldsymbol{g}_f$, and output gate $\boldsymbol{g}_o$ in a LSTM cell,

$$\begin{aligned} \boldsymbol{g}_i &= \sigma\left(\boldsymbol{W}_i \boldsymbol{z}_t + \boldsymbol{V}_i \boldsymbol{h}_{t-1}\right) \\ \boldsymbol{g}_f &= \sigma\left(\boldsymbol{W}_f \boldsymbol{z}_t + \boldsymbol{V}_f \boldsymbol{h}_{t-1}\right) \cdot \\ \boldsymbol{g}_o &= \sigma\left(\boldsymbol{W}_o \boldsymbol{z}_t + \boldsymbol{V}_o \boldsymbol{h}_{t-1}\right) \end{aligned} \tag{4.11}$$

where $\sigma(\cdot)$ denotes the sigmoid function. These gates control how much information should be passed through in different places of LSTM cells as follows,

$$\begin{aligned} \boldsymbol{c}_t &= \boldsymbol{g}_f \circ \boldsymbol{c}_{t-1} + \boldsymbol{g}_i \circ \tanh\left(\boldsymbol{W}_c \boldsymbol{x}_t + \boldsymbol{V}_c \boldsymbol{h}_{t-1}\right) \\ \boldsymbol{h}_t &= \boldsymbol{g}_o \circ \tanh(\boldsymbol{c}_t) \end{aligned}, \tag{4.12}$$

where $\boldsymbol{c}_t$ and $\boldsymbol{h}_t$ denote the cell state and the hidden state at time t, respectively. Here, the operator "$\circ$" denotes the element-wise multiplication. In LSTM, the cell state is employed to remember the current state of the cell and the hidden state records the output of the LSTM cell, which could be further inputted to next layer of the network.

In this work, we propose a three-layer LSTM network as a per-cell demand predictor as described in Model 2, which regards the per-cell demand of all cells at each time stamp as inputs.

Model 2 (Long Short-Term Memory (LSTM)) *A per-cell demand predictor is approximated by a three-layer LSTM network with two LSTM layers and one full-connection layer. The LSTM sequence model is demonstrated in Fig. 4.4 and illustrated mathematically as*

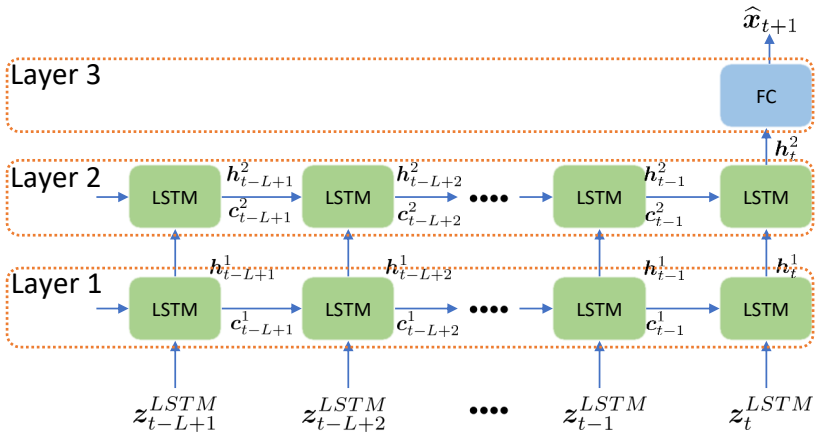


Figure 4.4: Temporal modeling: long short-term memory (LSTM).

follows,

$$\begin{aligned} &\textit{Layer 1:} \quad (\boldsymbol{h}_t^{(1)}, \boldsymbol{c}_t^{(1)}) = \eta_{lstm}^{(1)}\left(\boldsymbol{z}_t^{(LSTM)}, \boldsymbol{h}_{t-1}^{(1)}, \boldsymbol{c}_{t-1}^{(1)}\right) \\ &\textit{Layer 2:} \quad (\boldsymbol{h}_t^{(2)}, \boldsymbol{c}_t^{(2)}) = \eta_{lstm}^{(2)}\left(\boldsymbol{h}_t^{(1)}, \boldsymbol{h}_{t-1}^{(2)}, \boldsymbol{c}_{t-1}^{(2)}\right) \\ &\textit{Layer 3:} \quad \hat{\boldsymbol{x}}_{t+1} = \boldsymbol{W}^{(3)}\boldsymbol{h}_t^{(2)} \end{aligned} \tag{4.13}$$

where $\eta_{lstm}^{(i)}(\cdot,\cdot,\cdot)$ *denotes the updating function of the layer* i *LSTM cell as described in* (4.11) *and* (4.12)*, in which the trainable parameters are listed as follows,*

$$\begin{aligned} &\textit{Layer 1:} \quad \boldsymbol{W}_{i,o,f,c}^{(1)} \in \mathcal{R}^{(N\times F)\times n_{h_1}}, \boldsymbol{V}_{i,o,f,c}^{(1)} \in \mathcal{R}^{n_{h_1}\times n_{h_1}} \\ &\textit{Layer 2:} \quad \boldsymbol{W}_{i,o,f,c}^{(2)} \in \mathcal{R}^{n_{h_1}\times n_{h_2}}, \boldsymbol{V}_{i,o,f,c}^{(2)} \in \mathcal{R}^{n_{h_2}\times n_{h_2}} \\ &\textit{Layer 3:} \quad \boldsymbol{W}^{(3)} \in \mathcal{R}^{n_{h_1}\times n_{h_2}}, \end{aligned}$$

where n_{h_1} *and* n_{h_2} *denote the size of hidden states in layer* 1 *and layer* 2*, respectively.*

Here, the input $\boldsymbol{z}_t^{(LSTM)}$ is a vector that contains features of all cells at time t, whose size is $(N \times F) \times 1$. As a result, the number of trainable parameters in Model 2 is $4n_{h_1}(N \times F + n_{h_1}) + 4n_{h_2}(n_{h_1} + n_{h_2}) + n_{h_2}N$. In LSTM, we only model the temporal aspect of the per-cell demand data, but omit the spatial information. In other words, the local spatial dependence

is not considered in the LSTM model, but the full connection from one cell to all other cells are taken into account, which may lead to overfitting issue in the LSTM model.

4.4.3 Spatiotemporal Modeling - Graph Convolutional LSTM (GCLSTM)

With the spatial and temporal information modeled, the LSTM and GCN can be integrated to utilize both the spatial and temporal information, which is termed as graph convolutional LSTM (GCLSTM). In GCLSTM, the global connection among vertices (matrix multiplication in LSTMs) is replaced by the local graph convolution (4.8) in each gates as follows,

$$\begin{aligned}
\boldsymbol{G}_i =& \sigma\left(\widehat{\boldsymbol{A}}(\boldsymbol{Z}_t\boldsymbol{\Theta}_i + \boldsymbol{H}_{t-1}\boldsymbol{\Psi}_i)\right)\\
\boldsymbol{G}_f =& \sigma\left(\widehat{\boldsymbol{A}}(\boldsymbol{Z}_t\boldsymbol{\Theta}_f + \boldsymbol{H}_{t-1}\boldsymbol{\Psi}_f)\right),\\
\boldsymbol{G}_o =& \sigma\left(\widehat{\boldsymbol{A}}(\boldsymbol{Z}_t\boldsymbol{\Theta}_o + \boldsymbol{H}_{t-1}\boldsymbol{\Psi}_o)\right)
\end{aligned} \tag{4.14}$$

where $\boldsymbol{G}_{i,f,o} \in \mathcal{R}^{N\times n_h}$. Also, the hidden states are also updated locally as follows,

$$\begin{aligned}
\boldsymbol{C}_t &= \boldsymbol{G}_f \circ \boldsymbol{C}_{t-1} + \boldsymbol{G}_i \circ \tanh\left(\widehat{\boldsymbol{A}}(\boldsymbol{Z}_t\boldsymbol{\Theta}_c + \boldsymbol{H}_{t-1}\boldsymbol{\Psi}_c)\right)\\
\boldsymbol{H}_t &= \boldsymbol{G}_o \circ \tanh(\boldsymbol{C}_t)
\end{aligned} \tag{4.15}$$

Accordingly, a per-cell demand predictor based on GCLSTM is proposed to model both the spatial and temporal dimension of the per-cell demand time series as illustrated in Model 3.

Model 3 (Graph Convolutional LSTM (GCLSTM)) *A per-cell demand predictor is approximated by a three-layer GCLSTM with two layers of GCLSTM cells and one graph*

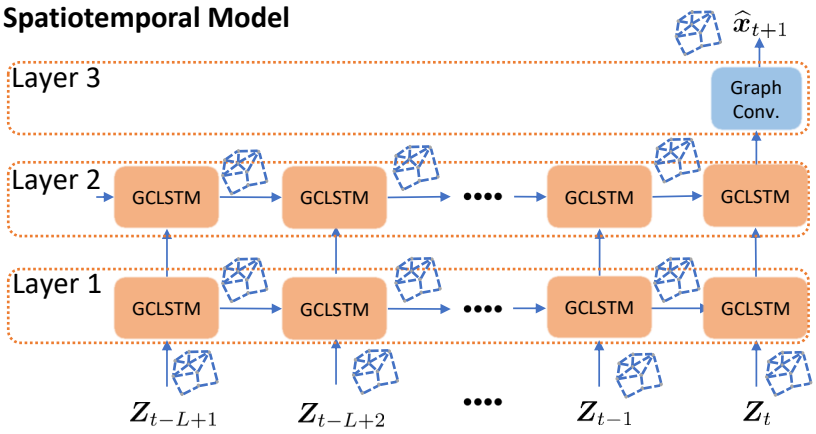


Figure 4.5: Spatiotemporal modeling: graph convolutional LSTM (GCLSTM).

convolutional layer, i.e.,

$$\begin{aligned}
&\textit{Layer 1:}\quad (\boldsymbol{H}_t^{(1)}, \boldsymbol{C}_t^{(1)}) = \eta_{gclstm}^{(1)}\left(\boldsymbol{Z}_t, \boldsymbol{H}_{t-1}^{(1)}, \boldsymbol{C}_{t-1}^{(1)}\right)\\
&\textit{Layer 2:}\quad (\boldsymbol{H}_t^{(2)}, \boldsymbol{C}_t^{(2)}) = \eta_{gclstm}^{(2)}\left(\boldsymbol{H}_t^{(1)}, \boldsymbol{H}_{t-1}^{(2)}, \boldsymbol{C}_{t-1}^{(2)}\right)\\
&\textit{Layer 3:}\quad \hat{\boldsymbol{x}}_{t+1} = \widehat{\boldsymbol{A}}\boldsymbol{H}_t^{(2)}\boldsymbol{\Theta}^{(3)}
\end{aligned} \tag{4.16}$$

where $\eta_{gclstm}^{(i)}(\cdot,\cdot,\cdot)$ *denotes the layer* i *GCLSTM cell based on* (4.14) *and* (4.15)*,where the trainable parameters are illustrated as follows,*

$$\begin{aligned}
&\textit{Layer 1:}\quad \boldsymbol{\Theta}_{i,f,o,c}^1 \in \mathcal{R}^{F\times n_{h_1}}, \boldsymbol{\Psi}_{i,f,o,c}^1 \in \mathcal{R}^{n_{h_1}\times n_{h_1}}\\
&\textit{Layer 2:}\quad \boldsymbol{\Theta}_{i,f,o,c}^2 \in \mathcal{R}^{n_{h_1}\times n_{h_2}}, \boldsymbol{\Psi}_{i,f,o,c}^2 \in \mathcal{R}^{n_{h_2}\times n_{h_2}}.\\
&\textit{Layer 3:}\quad \boldsymbol{\Theta}^3 \in \mathcal{R}^{n_{h_2}\times 1}
\end{aligned}$$

Again, n_{h_1} *and* n_{h_2} *denote the size of hidden states in layer* 1 *and layer* 2*, respectively.*

Here, the input $\boldsymbol{Z}_t$ at time t is a matrix with the shape $N \times F$ defined by (4.4). The number of trainable parameters is $4n_{h_1}(n_{h_1}+F)+4n_{h_2}(n_{h_2}+n_{h_1})+n_{h_2}$. Compared with LSTM, the number of trainable parameters could be largely reduced, since the parameters are shared

Table 4.1: Comparisons of Three Per-Cell Demand Prediction Models

	Model Type	Input Dimension	Feature Size	Trainable Param. Num. (Example*)
GCN	Spatial	2-D Matrix $\boldsymbol{Z} \in \mathcal{R}^{N \times (L \times F)}$	$L \times F$	$n_{h_1}(L \times F) + n_{h_1} n_{h_2} + n_{h_2}$ $(2,208)$
LSTM	Temporal	1-D Vector $\boldsymbol{z}_t \in \mathcal{R}^{(N \times F) \times 1}$	$N \times F$	$4n_{h_1}(N \times F + n_{h_1}) + 4n_{h_2}(n_{h_1} + n_{h_2}) + n_{h_2} N$ $(310,976)$
GCLSTM	Spatiotemporal	2-D Matrix $\boldsymbol{Z}_t \in \mathcal{R}^{N \times F}$	F	$4n_{h_1}(n_{h_1} + F) + 4n_{h_2}(n_{h_2} + n_{h_1}) + n_{h_2}$ $(12,704)$

*$n_{h_1} = n_{h_2} = 32$, $F = 3$, $N = 718$ and $L = 12$

across the graph with local dependence modeled. Such parameter sharing could mitigate the overfitting problem by structurally shrinking the capacity of the model. Details of model comparisons are summarized in Table 4.1.

4.5 Experiments

In this section, we verify three proposed spatial, temporal, and spatiotemporal models based on the extracted per-cell demand data of 718 cell towers in the mobile network. The per-cell demands are first normalized by their mean and standard deviation in a cell-wise manner. The demand predictors proposed in this work are implemented by PyTorch [86], which is a deep learning framework with automatic differentiation and dynamic computational graph. The training dataset is from Aug. 22, 2016 to Nov. 26, 2016 and the test dataset is from Nov. 27, 2016 to Dec. 3, 2016.

4.5.1 Per-cell Demands Autocorrelation Analysis

We first investigate the autocorrelation analysis of the per-cell demands in a cell-wise manner, in order to determine the window length L should be taken into account for one-step ahead prediction. In the literature, the autocorrelation analysis and its partial derivative are commonly adopted to determine the order of autoregression integrated moving average (ARIMA) model. Specifically, the autocorrelation function (ACF) would decide the order of the moving average, while the partial autocorrelation function (PACF) could shed lights

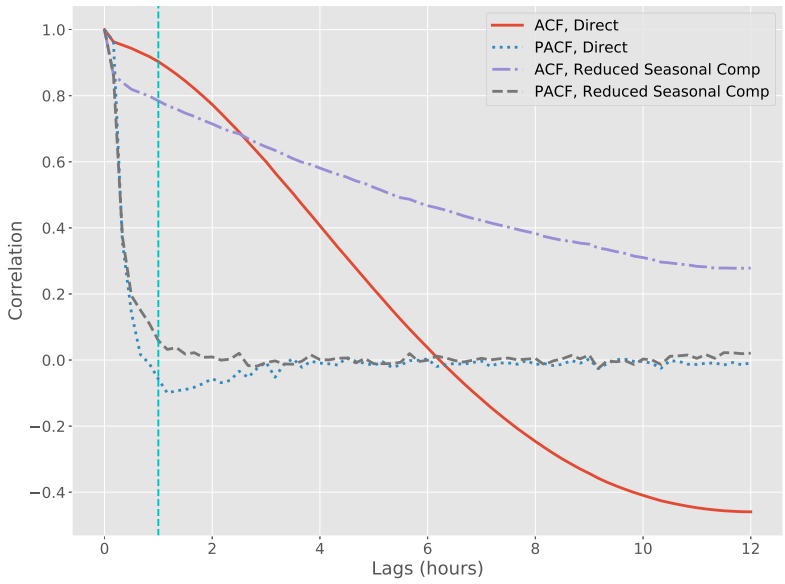


Figure 4.7: Autocorrelation function and partial autocorrelation function of demand time series with event counting time windows $\Delta T = 10$ minutes.

on the order selection for the autoregression. While the proposed time series model is quite different from ARIMA, the autocorrelation analysis could still be employed to suggest the window-length L selection.

Fig. 4.7 shows the correlation analysis on the per-cell demand time series with the counting time window, $\Delta T = 10$. As the per-cell demand is strongly periodic with respect to calendar days as shown in Fig. 4.2, the per-cell demand of cell i can be further decomposed into two parts, mean and its random component,

$$\boldsymbol{x}_d^i = \bar{\boldsymbol{x}}_d^i + \boldsymbol{\epsilon}^i$$

where $\bar{\boldsymbol{x}}_d^i$ is the periodic component. Hence, Fig. 4.7 shows two kinds of curves, namely the direct and the periodic (seasonal) component reduced, which demonstrate the (partial) autocorrelation analysis directly on the per-cell demand time series and the random com-

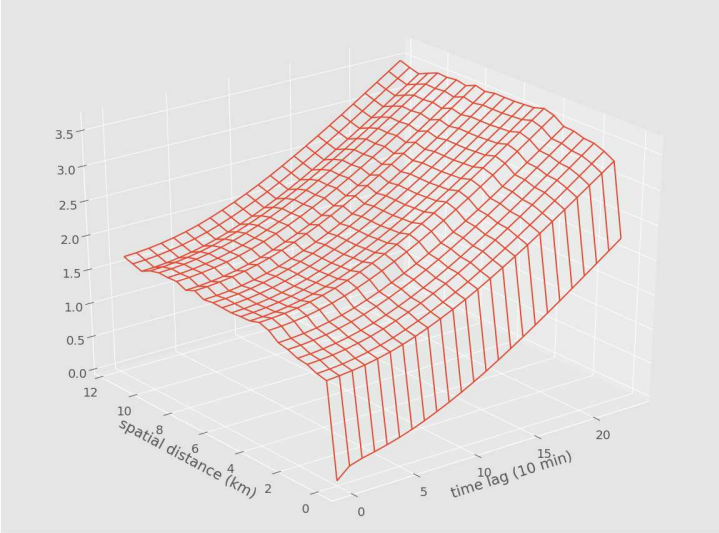


Figure 4.8: Spatiotemporal Semivariogram.

ponent, respectively. It could be observed that the PACF curves rapidly decrease to zero with the time lag increased, while the ACF curves are slowly decreasing, especially the direct one. One can conclude that one-hour history is sufficient for one-step-ahead prediction, but a long history could benefit from capturing the random component in the time series. As a result, we compare the different history lengths (half-hour, 1-hour, 2-hour, and 3-hour) for all three proposed models in various settings.

4.5.2 Spatiotemporal Analysis

The objective of the spatiotemporal analysis on the multiple per-cell demand signals is to evaluate how demand signals vary in space and time. In other words, the correlation between two signals in terms of both the time lags and the spatial distance is of significant interest. Such spatiotemporal analysis would lead to our critical spatial modeling of demands observed by many cells irregularly spatially distributed. In this work, the semivariogram, originated from spatial statistics, is employed to analyze the per-cell demands. Details of

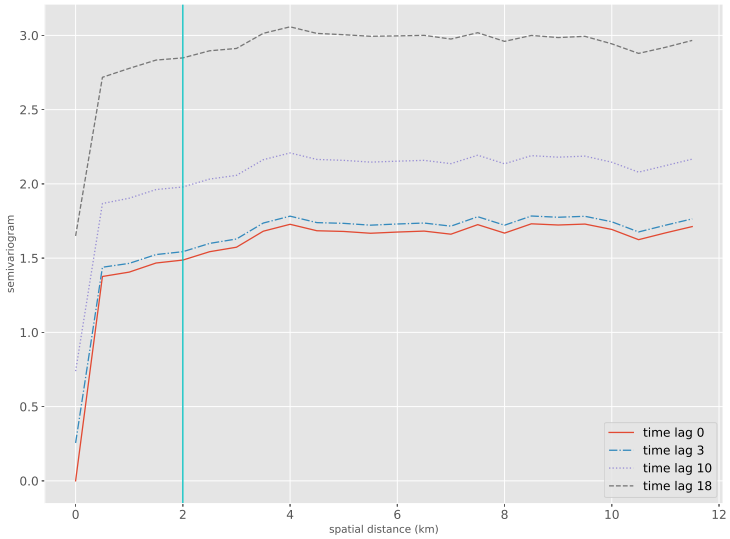


Figure 4.9: Semivariogram in terms of spatial distance.

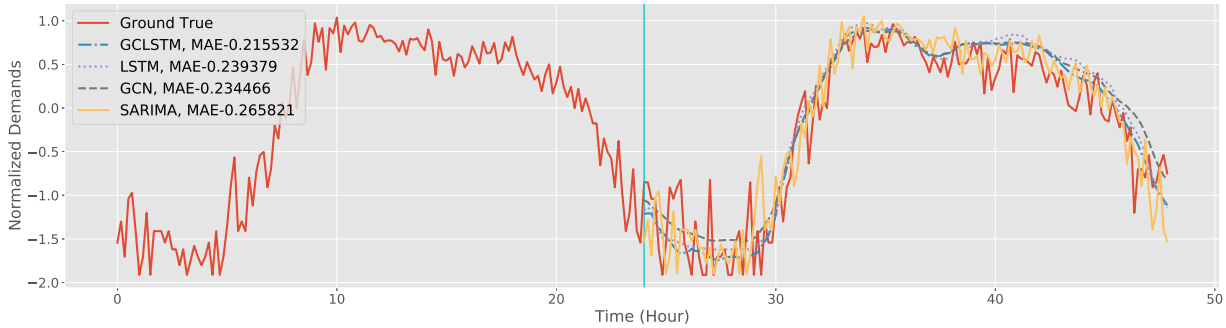


Figure 4.10: A example of dynamic per-cell demand forecasting.

semivariogram refer to Appendix 4.8. Fig. 4.8 shows the semivariogram of per-cell demand time series with different counting time window lengths $\Delta T = 10$. Based on the definition of semivariogram, the small value of semivariogram indicates the high dependence between signals separated at distance h and time lag τ. It could be observed that the semivariogram slowly grows along the time lag axis when $h = 0$, which suggests that the current per-cell demand is highly correlated with its history.

As for the spatial dependence, it can be observed in Fig. 4.9 that the value of semivariogram will stay the same after the spatial distance is 4 km. Such a flat curve suggests

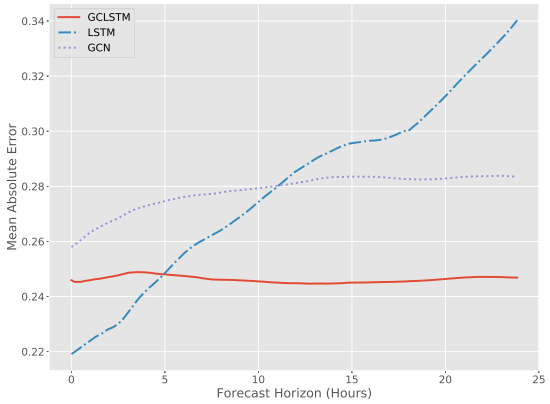


Figure 4.11: MAE performance of dynamic forecasting over all cells.

that any two cells with the distance more significant than 4 km could be considered as irrelevant. In this work, two-layer graph convolution operations are employed in each graph-based model, to mimic the second order graph filter based on the simple first-order graph filter approximation. Accordingly, we set the threshold ζ to be 2 km to capture the neighbors within 4 km after two-layer graph convolution operations.

4.5.3 Prediction Performance

In this work, we employ the mean absolute error (MAE) as the criterion to evaluate the predictors studied in this work. Though the forecast problem is formulated as a one-step-ahead prediction problem (4.2), the per-cell demand predictor should be capable of forecasting the demands of a future time window. In fact, the demand forecasting is fulfilled by the dynamic prediction via the one-step ahead predictor, which would take predicted demands as inputs to further forecast the future demands, e.g., $\hat{\boldsymbol{x}}_{t+2} = f(\hat{\boldsymbol{x}}_{t+1}, \boldsymbol{x}_t, \cdots)$.

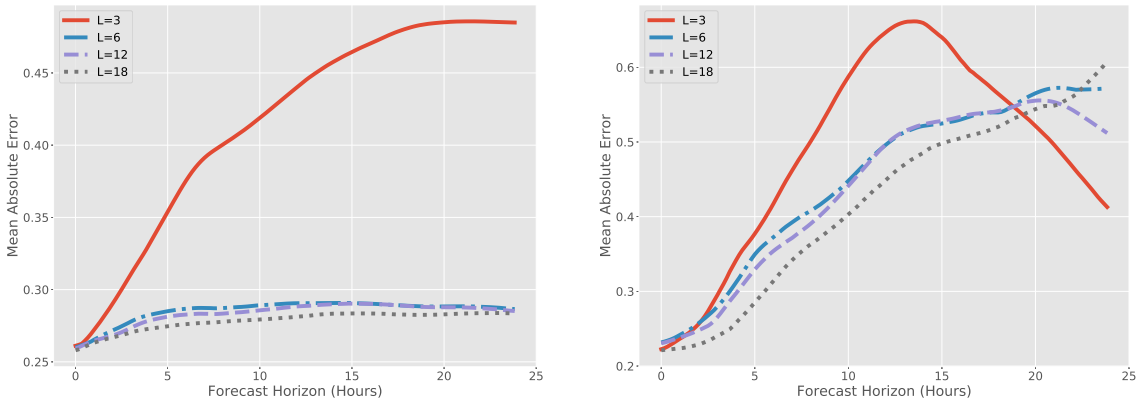
As a result, two parameters, forecast horizon and forecast resolution, are essential for a forecasting problem. The forecast resolution relies on the length of event counting time window, which is a predict per 10 minutes in this work. In this work, we focus on the

studied models with the forecasting horizon of 24 hours. In [72], a seasonal ARIMA model is proposed to predict the per-cell demands of a single cell with seasonal component modeled, SARIMA $(1,0,3) \times (1,1,1)$,

$$(1-ar_1 z^{-1})(1-sar_1 z^{-n_d})(1-z^{-n_d})x_t^i = (1+ma_1 z^{-1}+ma_2 z^{-2}+ma_3 z^{-3})(1+sma_1 z^{-n_d})\epsilon_t,$$

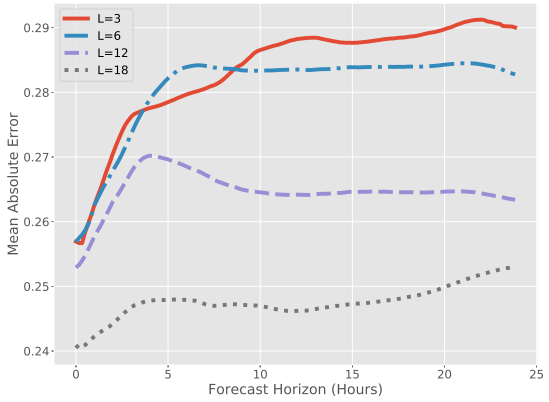
where ϵ_t denotes the noise component and z_{-1} denotes the operation of one time lag. Though SARIMA cannot model the spatial correlation among cells nor simultaneously predict the per-cell demands across the entire network, we could still perform the comparisons in a cell-wise manner. In Fig 4.10, an example of 24-hour demand forecasting of a cell is shown, including the proposed models and the SARIMA. It could be observed that the predicts by the SARIMA is more fluctuate than that of our models, while our proposed models smoothly trace the ground truth curve. In Fig 4.11, the average predicted MAE comparisons among three proposed models over all cells in the network is demonstrated. Overall, the spatiotemporal model (GCLSTM) is the best except for the case that forecast horizons are less than 5 hours. As the capacity of the LSTM model without parameter sharing and locality modeling is much larger than the one of GCLSTM, demonstrated by their number of trainable parameters (see Table 4.1), the LSTM can well capture the insight for one-step-ahead prediction. However, the LSTM also efficiently models the noise into the predictor during training, which could lead to the overfitting issue and worsen the forecasting performance of the model. Fig. 4.13 also demonstrates the our proposed GCLSTM model performs better than the SARIMA.

Fig. 4.12 illustrates the differences in demand history length for per-cell demand predic-



(a) GCN

(b) LSTM



(c) GCLSTM

Figure 4.12: MAE comparison between different window length L, where the event count time window is 10 minutes.

tion. Overall, a long demand history could improve the accuracy for long forecast horizon, especially for the LSTM-based models, which may result from the hidden states of LSTM-based models could remember more information when their hidden states are updated longer. On the other hand, the GCN model is not sensitive to the demand history length when $L \geq 6$ (longer than or equal to 1 hour) due to the lack of explicit temporal modeling, as shown in Fig. 4.12(a).

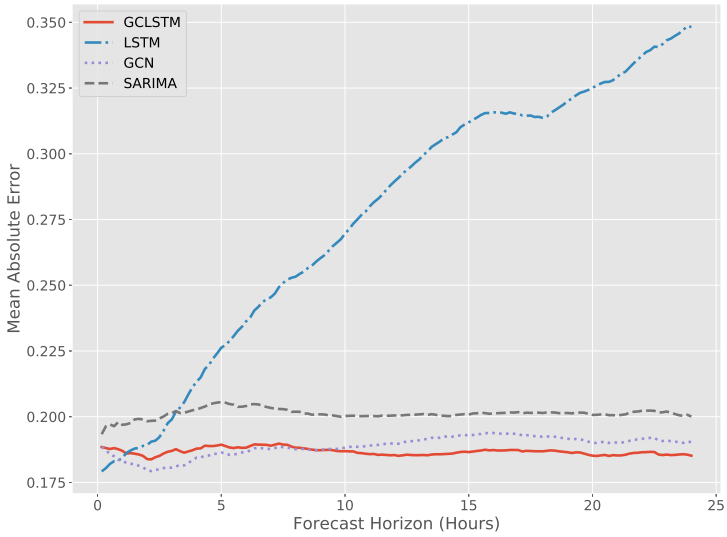


Figure 4.13: Compared with SARIMA $(1, 0, 1) \times (1, 1, 1)$.

4.5.4 Discussion

As demonstrated in the experiment results, the LSTM model could always have the best performance for the very-short-term demand forecasting, namely less than 3 hours. However, due to accumulated errors during dynamic prediction and week generalization of the LSTM model, the GCLSTM model is more capable for the short-term, mid-term, and day ahead demand forecasting. The GCN is also stable for such forecast horizons but is less accurate, while the number of trainable parameters is much smaller as illustrated in Table 4.1. The SARIMA model performs well for the per-cell demands prediction task, but it is modeled in a cell-wise manner. That is, the per-cell demand needs to be predicted cell-by-cell. As a result, the parameters of SARIMA is linearly scaling with the number of cells in the network, while our proposed GCLSTM takes both the spatial and temporal into accounts with fixed number trainable parameters and could have a relatively small trainable parameter for a large mobile network.

4.6 Summary

In this work, we study the per-cell demand forecasting in cellular networks. To deal with the irregular cell spatial distribution for spatial relevancy modeling among cells, we proposed to model the spatial relevancy among cells as a dependency graph based on spatial distances among cells without losing spatial granularity. Accordingly, we studied three models for demand forecasting, the spatial only (graph), the temporal only (sequence), and the spatiotemporal model (graph-sequence) based on deep learning. The spatiotemporal model simultaneously could capture both the spatial and temporal aspects in demand forecasting, which could achieve a superior forecasting performance demonstrated by experiment results.

4.7 Graph Filters and Graph Convolution

The graph signal processing (GSP) [79–81] is recently developed to deal with signals generated from a graph, such as social networks and sensor networks, which is a general extension of the traditional signal processing techniques from regular sampled data (e.g., audio or image) to the irregular data (social network data). The graph signal processing combines both the signal processing and graph spectral theory, to fulfill the standard signal processing operations on the graph, e.g., convolution, filtering, translation, etc.

The main motivation of building a spatial dependence graph in this paper is to predict the demand of one cell not only based on the its own demand history but also taking the demand history of its neighbors into account. In the graph signal processing theory, such

motivation could be captured by the graph Lapacian operation,

$$(\boldsymbol{L} \cdot \boldsymbol{x}_t)_i = \sum_{j \in \mathcal{N}_i} \left[x_t^i - x_t^j \right], \tag{4.17}$$

where $\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A}$ is the graph Laplacian and $\boldsymbol{D}$ is the diagonal matrix, i.e., $\boldsymbol{D}_{ii} = \sum_j A_{ij}$, recording the connectivity of each vertex in the graph. Intuitively, the graph Laplacian operation is essentially to capture the information of one vertex and its nearest neighbors.

Analogous to the filter design in the traditional signal processing, a graph filter could be expressed as polynomials in terms of the graph Laplacian [79],

$$g_\theta(\widetilde{\boldsymbol{L}}) = \theta_0 \boldsymbol{I} + \theta_1 \widetilde{\boldsymbol{L}} + \theta_2 \widetilde{\boldsymbol{L}}^2 + \cdots + \theta_K \widetilde{\boldsymbol{L}}^K, \tag{4.18}$$

where $\widetilde{\boldsymbol{L}}$ is the normalized graph Laplacian, i.e., $\widetilde{\boldsymbol{L}} = \boldsymbol{I} - \boldsymbol{D}^{-\frac{1}{2}} \boldsymbol{A} \boldsymbol{D}^{-\frac{1}{2}}$. And θ_k is the filter coefficient of tap k. The order of graph filters would determine the order of neighbors of vertices in the graph affected by the filter.

By the eigendecomposition on the graph Laplacian, $\widetilde{\boldsymbol{L}} = \boldsymbol{U} \Lambda \boldsymbol{U}^T$, any graph signal could be transformed to the corresponding graph spectral domain, $\boldsymbol{X} = \boldsymbol{U}\boldsymbol{x}$, analogous to the discrete Fourier transform [79–81], where the eigenvectors $\boldsymbol{U}$ are viewed as a basis. As a result, the graph filter could be further expressed in the graph spectral domain,

$$g_\theta(\boldsymbol{\Lambda}) = \theta_0 + \theta_1 \boldsymbol{\Lambda} + \theta_2 \boldsymbol{\Lambda}^2 + \cdots + \theta_K \boldsymbol{\Lambda}^K. \tag{4.19}$$

Hence, the graph convolution operation $g_\theta(\widetilde{\boldsymbol{L}}) * \boldsymbol{x}_t$ can be calculated as multiplication oper-

ations in the graph spectral domain,

$$g_\theta(\widetilde{\boldsymbol{L}}) \star \boldsymbol{x}_t = \boldsymbol{U} g_\theta(\boldsymbol{\Lambda}) \boldsymbol{U}^T \boldsymbol{x}_t. \tag{4.20}$$

4.8 Spatiotemporal Semivariogram

The per-cell demand time series (4.1) could be further expressed in terms of both the spatial and temporal aspects as follows,

$$z(s_n, t) = x_t^n \tag{4.21}$$

where s_n represents the detailed spatial information of the n-th cell (i.e., location coordinates). The semivariogram $\gamma(h)$ is a function to describe the spatial dependence of two stochastic processes generated in two locations s_n and s_m separated at h distance,

$$\gamma(h) = \mathrm{E}\left[(z(s_n) - z(s_m))^2 | \mathrm{dist}(s_n, s_m) = h\right] .$$

With the temporal dependence considered, the time lag τ should be further considered atop the spatial variogram $\gamma(h)$.

$$\gamma(h, \tau) = \mathrm{E}\left[(z(s, t) - z(s + h, t + \tau))^2\right]$$

However, the cell towers are distributed irregularly in the covered area according to the population density. Hence, we analyze the multiple per-cell demand processes in terms of

the empirical spatiotemporal semivariogram [75, 87] as follows,

$$\gamma(h(l),\tau) = \frac{1}{|\mathcal{N}(h(l),\tau)|} \times \sum_{(n,m,t,t')\in\mathcal{N}(h(l),\tau)} \left[z(s_n,t) - z(s_m,t')\right]^2, \tag{4.22}$$

where

$$\mathcal{N}(h(l),\tau) = \{(n,m,t,t') | \text{dist}(s_n,s_m) \in h(l), |t-t'| = \tau\}.$$

The $\mathcal{N}(h(l),\tau)$ is a set to collect any signal pairs spatially separated at distance within the distance tolerance $h(l)$ and temporally separated at τ. The distance tolerance $h(l)$ is employed to discretize the continuous spatial distance. In this paper, we utilize a linear uniform discretization with the spatial resolution 0.5 km. As a result, $h(l) = [(l-1) \times 0.5, l \times 0.5)$.

CHAPTER 5

AGGREGATED SPATIOTEMPORAL MODELING: IDLE TIME WINDOW PREDICTION FOR TRAFFIC MANAGEMENT

5.1 Background

[1]In recent years, self-organizing networks (SON) is widely studied to automatically manage and organize networks with much less manual interventions so that network operational expenditures (OPEX) and capital expenditures (CAPEX) can both be reduced [65, 66]. Mobile data plays a critical role in cellular SONs, providing system observability and predictability for network management. One of the most substantial portions of OPEX is the power consumption of cell towers [89]. To switch cells off when traffic loads across the network are extremely low [89, 90] is a potential approach to lower the power consumption of mobile networks.

However, cell towers may not be able to switch on to meet the traffic demands in real time. As a result, the cell on/off switching requires reliable predictive knowledge of ITW for each cell in a mobile network to reduce power consumption with subscribers' quality of experience ensured. ITWs in the network may vary spatially, which needs to be carefully learned [91]. Also, the ITW prediction is not limited to the application of cell switching on/off but can facilitate flexible traffic scheduling and network management applications. For example, time-dependent pricing [92] is one of the recently proposed solutions to reduce the peak-to-average (PAR) by motivating delay-tolerant traffic consumed in idle time windows so

[1]This work has been published in [88].

that the network congestion could be alleviated during the peak time. The dynamic pricing mechanism highly relies on the predictive knowledge of the ITWs across the network, as traffics and ITWs may vary temporally and spatially [93].

In this chapter, we propose to study the ITW prediction for each cell in mobile networks, based on both the recent history and periodic factors of mobile demands and subscribers' aggregated mobility behaviors observed by mobile operators. Prediction of the ITWs essentially amounts to answer the following questions:

1. Will the ITW start within the prediction horizon in the future?

2. When will the ITW start and how long will it last?

The first question is essentially a detection problem, while the second question leads to a regression or localization problem. The intuitive approach of ITW prediction is to first perform a long-term demand forecasting for each cell in the network, based on which ITWs in the future could be extracted. However, the long-term demand forecasting is usually formulated as one-step-ahead prediction [63, 72–74, 91, 94] and then generates long-term forecasts one-by-one sequentially based on predicted results, leading to mediocre ITW prediction due to error accumulation during forecasting. Furthermore, to predict ITW based on long-term demand forecasting may be expensive, as it needs to generate far more estimates than desired (i.e., start time and duration).

Differing from the approaches as mentioned earlier, we propose to directly predict ITW for each mobile network cell in this work. Specifically, the start time and duration within the prediction horizon in the future will be directly estimated. The ITW prediction is formulated as a regression problem with an *ITW presence confidence index*, which simultaneously

tackles the ITW detection problem (whether an ITW will present or not) and the ITW estimation (where the ITW is located within the prediction horizon). The novel ITW presence confidence index proposed in this work can effectively indicate the presence of ITWs in the forecasting horizon, and also provide the flexibility and capability to control the robustness of the prediction model in different practical scenarios. In terms of feature engineering, we first propose an innovative feature extraction scheme to obtain multiple demands and mobility features from the raw signaling datasets so that reliable ITW prediction can be facilitated. In addition, the day-ahead and week-ahead periodic observations will be regarded as exogenous inputs to account for the strong temporal seasonality. Furthermore, the spatiotemporal semivariogram demonstrates that mobile demand at the cell level has strong temporal relevancy yet relatively weak spatial relevancy, where the spatial relevancy among cells is modeled as a relevancy graph as in our previous work [63].

In light of the proposed feature engineering, exogenous inputs, and desired prediction output, we propose a novel ITW prediction model, consisting of the representation learning network and the output network. The representation learning network is aimed to learn useful patterns from the recent demand and mobility history for ITW prediction via effective graph sequence modeling. The output network is aimed to combine the learned patterns via representation learning networks and exogenous inputs to generate the desired ITW presence confidence index and ITW location. In the literature, two graph-sequence spatiotemporal models, GeoMAN [95] and DCRNN [96] were recently proposed to deal with the sequence-to-sequence environmental pollution prediction and road traffic prediction, respectively. However, these two models were designed for strong spatial and strong temporal relevancy, which may not be ideally suitable for the mobile demand traffic with strong tem-

poral relevancy yet relatively weak spatial relevancy. As a result, we further propose a novel prediction model, termed as temporal graph convolutional networks (TGCN), based on the cutting-edge temporal modeling [97] and graph modeling [76] techniques.

To effectively evaluate the performance of ITW estimation, we propose to employ a metric called intersection-over-union (IoU) borrowed from the object detection task in the field of computer vision [98], to assess how well the predicted time window overlaps with the ground truth. Experiment results demonstrate that our proposed general ITW prediction model can achieve a significant performance improvement compared with baselines, and the proposed TGCN-based model can outperform the temporal convolutional networks (TCN) [97], long short-term memory (LSTM) [78], and graph convolutional LSTM (GCLSTM) [63]. The good prediction performance suggests the validity of the proposed problem formulation and feature engineering as well as the superiority of our proposed TGCN model. The key contributions of this work are summarized as follows:

- Direct ITW prediction is proposed for the first time. It is formulated as a regression problem with an ITW presence confidence index, which can simultaneously tackle both the ITW detection and estimation tasks.

- A feature extraction scheme is proposed to extract the demand and mobility features from the raw signaling dataset as well as the exogenous inputs to account for the inherent characteristics of the demand and mobility behaviors.

- A novel ITW prediction model consisting of a representation learning network and an output network is proposed to account for both the spatiotemporal feature and the exogenous inputs. In addition, a graph-sequence representation network model, TGCN,

is proposed to characterize both the spatial and temporal relevancy in subscribers' demand and mobility behaviors to facilitate a good ITW prediction.

- A cost function combining the cross-entropy loss for ITW prediction and the mean absolute error for ITW estimation is proposed to train the prediction model effectively. In addition, an evaluation metric, intersection-over-union (IoU), is proposed to assess the performance of the ITW estimation.

5.2 Dataset and Problem Formulation

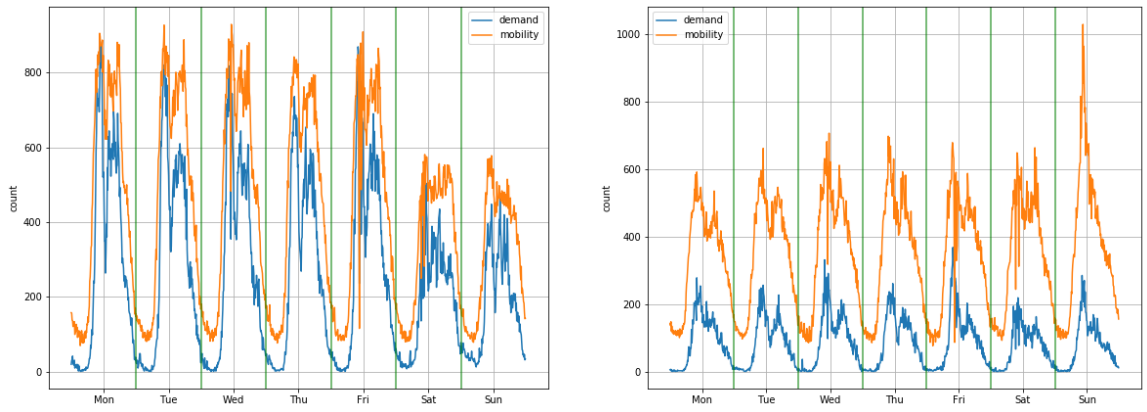
In this section, two semantic time series from the perspective of cells (base stations) based on the studied signaling dataset can be extracted from the raw data, namely the demand time series and the mobility time series. According to these two semantic time series, the problem formulation of ITW prediction will be discussed.

5.2.1 Demand and Mobility Time Series

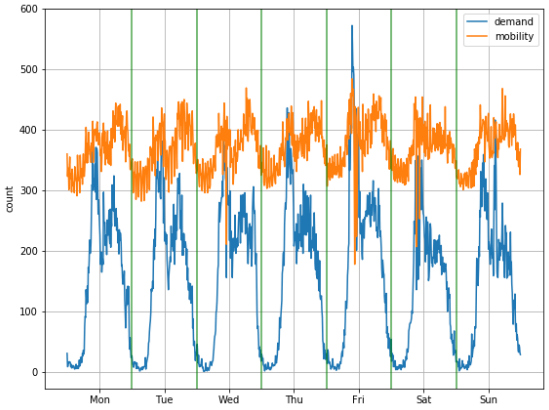
According to the event type, user pseudo-ID, timestamp and location information recorded in the studied signaling dataset, one can extract two semantic time series for each cell in the mobile network to understand the aggregated spatiotemporal behaviors of subscribers, namely

- *Demand Time Series.* The demand time series can be extracted by counting the *number of communication events* occurring in a counting time window for a specific cell; that is,

$$\boldsymbol{d}_i = \{\cdots,\ d_{t-l,i},\ d_{t-l+1,i},\ \cdots,\ d_{t-1,i},\ d_{t,i},\ \cdots\}; \tag{5.1}$$



(a) business (b) entertainment



(c) residence

Figure 5.1: Mobility and demand time series of typical cells, including business, entertainment, and residence.

- *Mobility Time Series.* The mobility time series is obtained by counting the *number of unique subscribers* observed in a counting time window for a specific cell; that is,

$$\boldsymbol{m}_i = \{\cdots,\ m_{t-l,i},\ m_{t-l+1,i},\ \cdots,\ m_{t-1,i},\ m_{t,i},\ \cdots\}; \tag{5.2}$$

where subscripts i and t denote the i-th cell and the t-th counting time window (i.e., $[t\Delta, (t+1)\Delta)$), respectively, and Δ denotes the counting time window length ($\Delta = 20$ minutes in this work). The demand time series, counting voice and text service events of each cell, can directly illustrate the load of mobile networks both spatially and temporally, while

the mobility time series can capture the mobility behavior of subscribers in an aggregated manner, shedding lights on the crowd flow of network subscribers. Both mobility and demand time series extracted from the studied signaling dataset can lead to a better understanding of mobile network subscribers.

Three demand and mobile time series examples of selected cells located at different typical points of interest are demonstrated in Fig. 5.1. The cells of business, entertainment, and residence are located at the central business district (CBD), zoo, and residential area, respectively. One can easily observe that both the mobile time series and demand time series are daily periodic in all three examples, while demands and mobility in the business and the entertainment cells behave differently during weekdays and weekends, which may be regarded as weekly periodicity in both time series. Both the demand and the number of visited subscribers (mobility) of the business-type cell decline during weekends, while the number of visited subscribers in the entertainment-type cell increases, compared with the ones during weekdays. In general, demands tend to increase as the number of visited subscribers rises, but the relationship between the two is nonlinear and depends on the cell type and the time within a day. Two peaks can be observed within a day for both the demand and mobile time series, and the demands of cells can drop to zero after midnight. As a result, one can easily find that the ITWs typically occur in the early morning in terms of the loads of each cell as illustrated by demand time series, regardless of the mobility pattern shown by the mobility time series.

5.2.2 Idle Time Window Prediction Problem Formulation

In this work, we propose to predict the ITW in the near future based on the features extracted from the demand and mobility time series in recent history, where the definition of ITWs is given in **Definition 5.1**.

Definition 5.1 (Idle Time Window) *The idle time window (ITW) of cell i at time t is represented as a tuple, $(S_{t,i}, D_{t,i})$, indicating that a consecutive time period during which demands fall below a predefined threshold ζ_i within the prediction horizon H, i.e.,*

$$\begin{aligned} S_{t,i}, D_{t,i} \; = \underset{S,D}{argmax} \quad & D \\ subject\ to \quad & d^i_{t+l} < \zeta_i, \forall l \in [S, S+D) \\ & 1 \leq S, D \leq H \end{aligned}$$

where S and D denote the start time and duration of the ITW within the time horizon $[t+1, t+H]$ in the future, respectively.

To predict the per-cell ITW, one needs to answer the following questions:

1. Will an ITW present within the prediction horizon, i.e., $[t+1, t+H]$?
2. When will the ITW start (S) and how long (D) will it last?

In fact, the first question is essentially a binary classification or detection problem to determine whether an ITW will present within the future horizon H. The second one can be regarded as a regression problem, which is to estimate the start time S and the duration D for each cell, respectively.

Inspired by the object detection algorithm in the field of computer vision [98], we formulate the ITW prediction as a regression problem with a confidence index to simultaneously account for both the ITW detection and the ITW regression tasks, i.e.,

$$C_{t,i}, S_{t,i}, D_{t,i} = f(\boldsymbol{X}_t, \boldsymbol{E}_t, \boldsymbol{A}) \tag{5.3}$$

where $S_{t,i} = (S_{t,i} - 1)/H$ and $D_{t,i} = D_{t,i}/H^2$ denote the normalized start time and duration with respect to the horizon H, respectively, when the ITW presents within the future horizon H. In addition, $C_{t,i}$ denotes the confidence index to suggest the confidence that an ITW presents within the future horizon. Hence, $C_{t,i}$, $S_{t,i}$, $D_{t,i} \in [0, 1)$. In (5.3) and (5.5), $\boldsymbol{X}_t$, $\boldsymbol{E}_t$, and $\boldsymbol{A}$ represent the input features, the exogenous inputs at time t, and the geospatial foreknowledge, respectively. Moreover, $f(\cdot)$ represents the predictor or the mapping that is trainable in a supervised manner based on extracted input-output pairs from the raw data.

Furthermore, the presence of ITWs can be determined by the confidence index $C_{t,i}$, i.e.,

$$\begin{cases} \mathcal{H}_0, & C_{t,i} < \tau \\ \mathcal{H}_1, & C_{t,i} \geq \tau \end{cases}, \tag{5.4}$$

where $\mathcal{H}_1$ and $\mathcal{H}_0$ represent the presence and absence of ITWs, respectively. Also, τ denotes the threshold for the confidence index to determine the presence of ITWs. In this work, we also aim to predict the ITWs for all cells across the network simultaneously. As a result, the

[2]Without confusion, $S_{t,i}$ and $D_{t,i}$ are employed to denote the start time and duration variable and also their normalized ones.

per-cell ITW prediction (5.3) could be further rewritten as follows,

$$\boldsymbol{C}_t, \boldsymbol{S}_t, \boldsymbol{D}_t = f(\boldsymbol{X}_t, \boldsymbol{E}_t, \boldsymbol{A}) \tag{5.5}$$

where $\boldsymbol{C}_t = [C_{t,1}, \cdots, C_{t,N}]^T$, $\boldsymbol{S}_t = [S_{t,1}, \cdots, S_{t,N}]^T$, and $\boldsymbol{D}_t = [D_{t,1}, \cdots, D_{t,N}]^T$. Based on the problem formulation in (5.5), all components of ITW prediction will be thoroughly discussed in the next sections.

5.2.3 Combined Cost Function for Model Training

Although the value of all three outputs in our problem formulation ranges between 0 and 1, the meanings underlying these three outputs are different. The confidence index $C_{t,i}$ serves as a detection statistics to determine the presence of ITWs, where $S_{t,i}$ and $D_{t,i}$ are to estimate where an ITW is located. As a result, the loss of these outputs in training should be specifically designed. Due to the underlying meaning of confidence index, we employ the cross entropy loss for binary classification (or detection) to train the model with respect to the confidence index output, i.e.,

$$\text{closs}(\widehat{C}_{t,i}, \tilde{C}_{t,i}) = \tilde{C}_{t,i} \log(\widehat{C}_{t,i}) + (1 - \tilde{C}_{t,i}) \log(1 - \widehat{C}_{t,i}) \tag{5.6}$$

where $\widehat{C}_{t,i}$ and $\tilde{C}_{t,i}$ denote estimated confidence index $C_{t,i}$ and its ground truth, respectively. As the true value of $\tilde{C}_{t,i}$ is either 1 or 0, the cross entropy loss function could be reduced to

$$\text{closs}(\widehat{C}_{t,i}, \tilde{C}_{t,i}) = \begin{cases} \log(\widehat{C}_{t,i}) & \tilde{C}_{t,i} = 1 \\ \log(1 - \widehat{C}_{t,i}) & \tilde{C}_{t,i} = 0 \end{cases} \tag{5.7}$$

With respect to the estimation of ITW start time and duration, the absolute error is employed to evaluate the estimates as follows,

$$\text{bloss}(\widehat{S}_{t,i}, \tilde{S}_{t,i}, \widehat{D}_{t,i}, \tilde{D}_{t,i}) = |\widehat{S}_{t,i} - \tilde{S}_{t,i}| + |\widehat{D}_{t,i} - \tilde{D}_{t,i}| \tag{5.8}$$

Accordingly, a cost function combining the above two loss functions is employed for model training as follows,

$$\begin{aligned} \text{cost} = \frac{1}{T \times B} \sum_t \sum_i \Big\{ & \text{closs}\left(\widehat{C}_{t,i}, \tilde{C}_{t,i}\right) + \\ & \lambda \times \mathbb{1}\left[\text{bloss}\left(\widehat{S}_{t,i}, \tilde{S}_{t,i}, \widehat{D}_{t,i}, \tilde{D}_{t,i}\right)\right] \Big\} \end{aligned} \tag{5.9}$$

where $\mathbb{1}[\cdot]$ is an indicator function to let the cost function only consider the start time and duration estimation when an ITW presents. In addition, a weight hyperparameter λ is further employed to help the cost function emphasize the model trained on the start time and duration prediction when an ITW exists.

5.3 Feature and Foreknowledge Engineering

5.3.1 Input Features $\boldsymbol{X}_t$

As ITWs are directly defined based on demand time series as shown in **Definition 5.1**, the trend of demands in each cell captured by the demand time series should be a key feature for the ITW prediction. In addition, the mobility time series (5.2), describing the number of subscribers observed by each cell within a counting time window, contains the information of aggregated crowd mobility behavior trend in the network. As a result, a series of demand and mobility observation of each cell should be regarded as features to predict the ITWs, i.e.,

$$\boldsymbol{X}_{t,i}^{\mathrm{dm}} = \begin{bmatrix} d_{t-L+1,i} & d_{t-L+2,i} & \cdots & d_{t,i} \\ m_{t-L+1,i} & m_{t-L+2,i} & \cdots & m_{t,i} \end{bmatrix}, \tag{5.10}$$

where L denotes the length of recent history considered for ITW prediction. According to the characteristics of the signaling data, the mobility time series can only observe active subscribers with communication demands or location updates. However, the counting time window (20 minutes) is much smaller than the periodic location update interval (60 minutes). Hence, the mobility time series may not be able to capture all subscribers attached to cells within one counting window, as inactive subscribers might stay in the same cell after a location update but unobserved at time t.

In this work, we propose an innovative feature extraction scheme to characterize the aggregated subscriber's mobility in each cell mainly. Let $\mathcal{U}_{t,i}^{1h}$ denote a subscriber set of cell i in a one-hour time window W_t^{1h}, i.e., $W_t^{1h} = [(t-2)\Delta, (t+1)\Delta)$ based on 20-minute counting windows employed in this work. We then propose to extract the following semantic feature

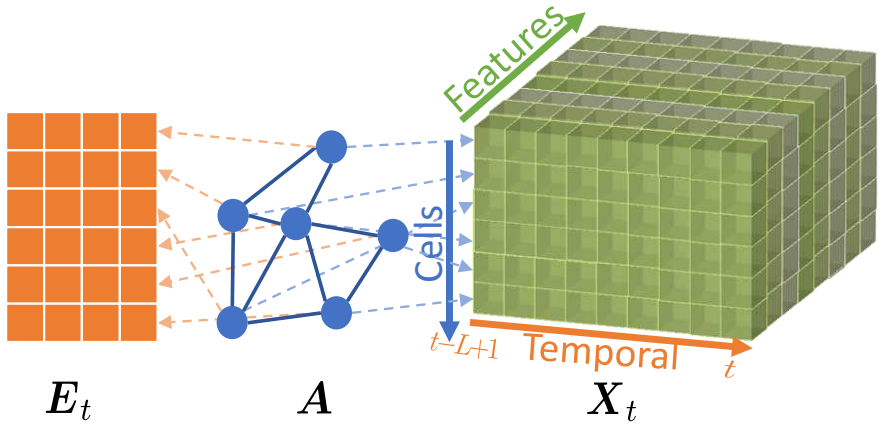


Figure 5.2: Features, exogenous inputs and relevancy graph for ITW prediction at time t.

time series based on the subscriber set $\mathcal{U}_{t,i}^{1h}$ and its one-step past $\mathcal{U}_{t-1,i}^{1h}$ as follows,

- **Arriving** $\delta_{t,i}^{m,+}$: The new arrival subscriber number is defined as the number of subscribers that are only observed in the counting time window t but not present in its one-step past subscriber set $\mathcal{U}_{t-1,i}^{1h}$, that is

$$\delta_{t,i}^{m,+} = \left|\mathcal{U}_{t,i}^{1h} - \mathcal{U}_{t-1,i}^{1h}\right|.$$

  Thus, the demand $\delta_{t,i}^{d,+}$ generated by the newly arrived subscribers $\mathcal{U}_{t,i}^{1h} - \mathcal{U}_{t-1,i}^{1h}$ can also be extracted in the counting time window $[t\Delta, (t+1)\Delta)$.

- **Staying** $\delta_{t,i}^{m,=}$: The subscribers observed in both sets, $\mathcal{U}_{t,i}^{1h}$ and $\mathcal{U}_{t-1,i}^{1h}$, are assumed to be the subscribers staying at cell i in the past one-hour time window, that is

$$\delta_{t,i}^{m,=} = \left|\mathcal{U}_{t,i}^{1h} \cap \mathcal{U}_{t-1,i}^{1h}\right|.$$

- **Departing** $\delta_{t,i}^{m,-}$: The subscribers observed only in one-step ahead set $\mathcal{U}_{t-1,i}^{1h}$, but do

not appear in current subscriber set $\mathcal{U}_{t,i}^{1h}$, that is

$$\delta_{t,i}^{m,-} = \left| \mathcal{U}_{t-1,i}^{1h} - \mathcal{U}_{t,i}^{1h} \right|.$$

Here, the operation $|\cdot|$ denotes the set cardinality. Clearly, $\delta_{t,i}^{m,+}, \delta_{t,i}^{m,=}, \delta_{t,i}^{m,-}, \delta_{t,i}^{d,+} \geq 0$. Accordingly, each cell could provide multiple features for its ITW prediction based on above operations on subscriber sets as follows,

$$\boldsymbol{X}_{t,i}^{\text{diff}} = \begin{bmatrix} \delta_{t-L+1,i}^{m,+} & \delta_{t-L+2,i}^{m,+} & \cdots & \delta_{t,i}^{m,+} \\ \delta_{t-L+1,i}^{m,=} & \delta_{t-L+2,i}^{m,=} & \cdots & \delta_{t,i}^{m,=} \\ \delta_{t-L+1,i}^{m,-} & \delta_{t-L+2,i}^{m,-} & \cdots & \delta_{t,i}^{m,-} \\ \delta_{t-L+1,i}^{d,+} & \delta_{t-L+2,i}^{d,+} & \cdots & \delta_{t,i}^{d,+} \end{bmatrix}. \tag{5.11}$$

Similar to our previous work on mobile demand forecasting [63], we also add the one-day ahead and 7-day ahead demand observations as features in order to capture both the daily periodic and weekly periodic effects as observed in Fig. 5.1, i.e.,

$$\boldsymbol{X}_{t,i}^{\text{period}} = \begin{bmatrix} d_{t-L+1-n_d,i} & d_{t-L+2-n_d,i} & \cdots & d_{t-n_d,i} \\ d_{t-L+1-7n_d,i} & d_{t-L+2-7n_d,i} & \cdots & d_{t-7n_d,i} \end{bmatrix} \tag{5.12}$$

where n_d denotes the number of observations in one day (i.e., $n_d = 72$). In this work, we take periodic demands to predict ITWs, while only recent mobility information is considered for ITW prediction.

In summary, the input features for the ITW prediction of cell i at time t can be expressed

by stacking $\boldsymbol{X}_{t,i}^{\text{dm}}$, $\boldsymbol{X}_{t,i}^{\text{diff}}$, and $\boldsymbol{X}_{t,i}^{\text{period}}$ as follow,

$$\boldsymbol{X}_{t,i} = \left[(X_{t,i}^{\text{dm}})^T, (X_{t,i}^{\text{period}})^T, (X_{t,i}^{\text{diff}})^T \right]^T. \tag{5.13}$$

By stacking $\boldsymbol{X}_{t,i}$ of all cells in the network, one can easily obtain a three-dimensional tensor,

$$\boldsymbol{X}_t \in \mathcal{R}^{N \times L \times 8},$$

each axis of which represents cells, temporal sequence, and features, respectively, as shown in Fig. 5.2. The one-day plot of the input features $\boldsymbol{X}_t$ is shown in Fig. 5.3, in which subscriber movement can be rarely observed at the early morning from 3 am to 6 am by both $\delta_{t,i}^{m,+}$ and $\delta_{t,i}^{m,-}$. Also, the quantity $(\delta_{t,i}^{m,+} - \delta_{t,i}^{m,-})$ is positive from 6 am to 9 am, meaning that subscribers move into this cell in this interval. The quantity $(\delta_{t,i}^{m,+} - \delta_{t,i}^{m,-})$ is negative from 5 pm to 8 pm, indicating that subscribers move out from this cell during this interval. Hence, time series $\delta_{t,i}^{m,+}$ and $\delta_{t,i}^{m,-}$ can effectively capture the movement of subscribers.

5.3.2 Exogenous Inputs $\boldsymbol{E}_t$

As shown in Fig. 5.1, it can be observed that both demands and mobility are daily periodic and weekly periodic. As a result, both the one-day-ahead and the 7-day ahead ITWs of each cell at the corresponding time point could provide valuable information to guide the ITW predictor to obtain more accurate estimates. The information of the one-day-ahead and 7-day ahead ITWs in each cell can serve as a relatively good starting point for ITW estimation, which will be employed as baselines in comparison with our proposed

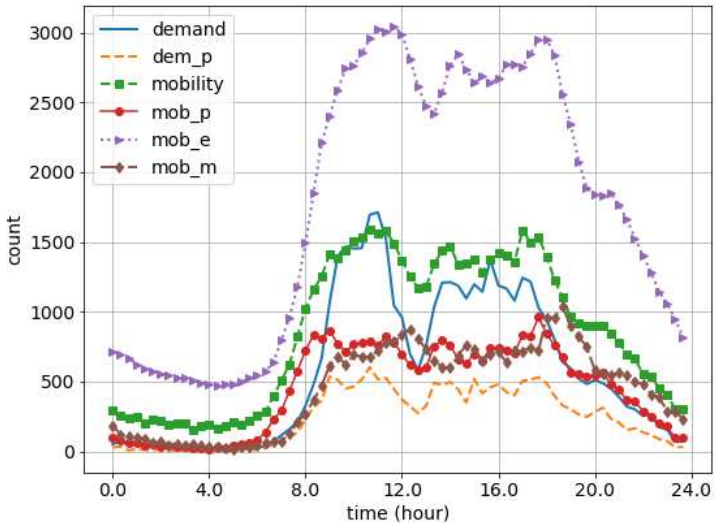


Figure 5.3: One-day plot of $\boldsymbol{X}_{t,i}$ at the cell corresponding to Fig. 5.1(a): demand ($d_{t,i}$), dem_p ($\delta_{t,i}^{d,+}$), mobility ($m_{t,i}$), mob_p ($\delta_{t,i}^{m,+}$), mob_e ($\delta_{t,i}^{m,=}$), mob_m ($\delta_{t,i}^{m,-}$).

predictors. Hence, the information of the one-day and 7-day ahead ITWs in each cell at time t will be regarded as exogenous inputs to our proposed predictors, i.e.,

$$\boldsymbol{E}_t = \begin{bmatrix} E_{t,1} & E_{t,2} & \cdots & E_{t,N} \end{bmatrix}^T \in \mathcal{R}^{N\times 4}, \tag{5.14}$$

where $E_{t,i} = [S_{t-n_d,i}\ D_{t-n_d,i}\ S_{t-7n_d,i}\ D_{t-7n_d,i}]^T$.

5.3.3 Geospatial Modeling via Graph $\boldsymbol{A}$

Based on the spatiotemporal semivariogram analysis of the demand time series across the network as in our previous work [63], it can be concluded as shown in Fig. 5.4 that the demand relevancy between two cells declines when their spatial distance increases. Hence, we employ the relevancy graph as in [63] to capture the spatial relevancy between cells across the network. The adjacency matrix $\boldsymbol{A}$ of the dependency graph can be obtained based on

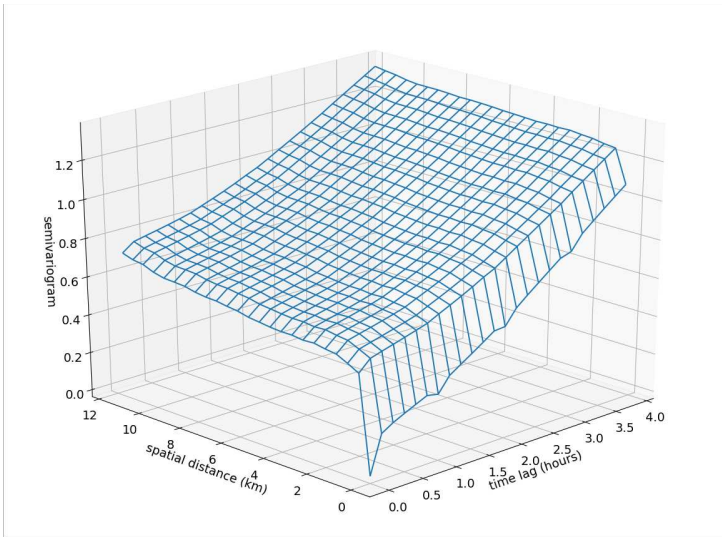


Figure 5.4: Spatiotemporal semivariogram analysis on demand time series.

the spatial distance between cells as follows,

$$\boldsymbol{A}_{ij} = \begin{cases} 1, & \text{dist}(s_i, s_j) \leq \eta \\ 0, & \text{otherwise} \end{cases}, \tag{5.15}$$

where s_i denotes the geolocation of cell i, and η is the distance threshold, that is a hyper-parameter that could be tuned. We set $\eta = 2$ km in this work. The threshold suggests that any two cells whose distance is beyond the threshold will be considered irrelevant. Such graph modeling could successfully make the cell relevancy sparse (from N^2 to $\sum_{i,j} \boldsymbol{A}_{i,j}$). As a result, each cell could be regarded as a vertex in the spatial dependency graph and the input $\boldsymbol{X}_{t,i}$ is viewed as the signal observed at node i of the graph at time t.

5.4 ITW Prediction Model

To predict the ITW for all cells in the network at each time t, a deep-learning-based ITW prediction model is proposed to account for the inherited structure of input features $\boldsymbol{X}_t$ and exogenous features $\boldsymbol{E}_t$ as well as the spatial relevancy foreknowledge encoded in the graph adjacency matrix $\boldsymbol{A}$. As demonstrated previously, the input features take the form of a three-dimensional tensor with both the temporal and spatial structures, as shown in Fig. 5.3. The proposed ITW predition model comprises two main components as shown in Fig. 5.5, namely

- **Representation Learning Network:** The representation learning network is aimed to learn the high-level representations from the spatiotemporal input tensor $\boldsymbol{X}_t$ with the foreknowledge provided by the relevancy graph $\boldsymbol{A}$,
- **Output Network:** The output network is responsible to integrate the learned high-level representations (obtained from the representation learning network) and the exogenous inputs $\boldsymbol{E}_t$ to generate the potential ITWs with confidence index (i.e., $[\boldsymbol{C}_t, \boldsymbol{S}_t, \boldsymbol{D}_t]$).

In this work, we employ the *feedforward neural networks (FNN)* as the output network structure. As for the representation learning network, a *temporal graph convolutional network (TGCN)* is proposed to account for the spatiotemporal structure of the input features $\boldsymbol{X}_t$ and to incorporate the spatial relevancy preknowledge $\boldsymbol{A}$. In the proposed TGCN, we innovatively integrate the temporal convolutional networks (TCN) [97] and graph convolutional network (GCN) [76] into a spatiotemporal model. It is worth noting that we propose to use the same network (the same network architecture and network parameters) at each cell in the network

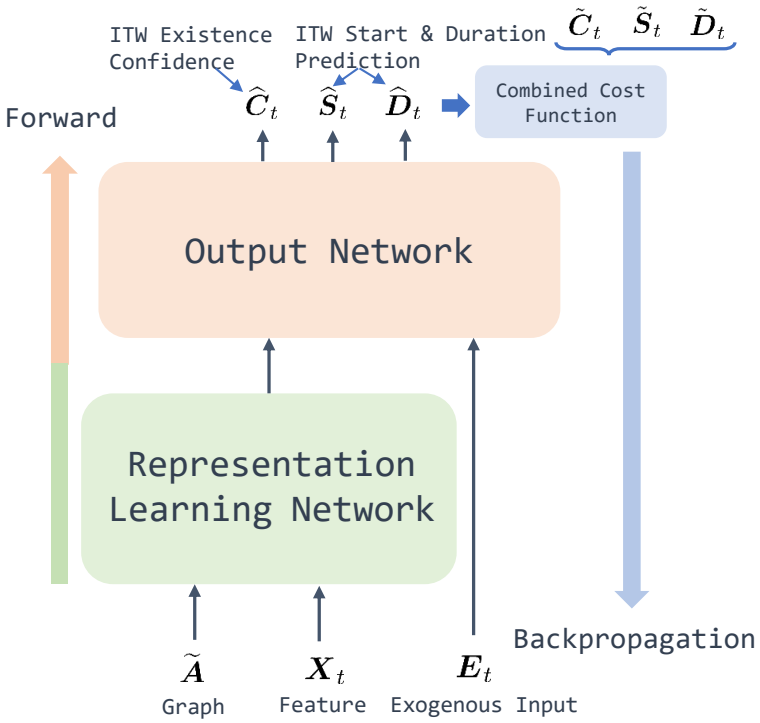


Figure 5.5: ITW Prediction Model

to predict their respective ITWs, as the powerful prediction model could simultaneously learn the representations of all cells in a mobile network.

5.4.1 Representation Learning Network

To learn the high-level representations from the input features $\boldsymbol{X}_t$, both the sequence and graph structures in $\boldsymbol{X}_t$ need to be sophisticatedly modeled, to prevent the overall prediction model from overfitting. As a result, we will discuss both the sequence and graph modeling for representation learning as follows.

A1. Temporal Modeling

In this work, we propose to employ the temporal convolutional network (TCN) to model the temporal structure of input $\boldsymbol{X}_t$ in our proposed representation learning network, which has been demonstrated as an excellent generic temporal (sequence) modeling architecture

in [97].

The TCN comprises two critical operations, namely the *dilated casual convolution (DC-Conv)* and the *residual connection*, both of which are aimed to deal with the training difficulty issue of very deep networks in different manners and discussed as follows.

Dilated Casual Convolution (DC-Conv): The dilated casual convolution operation takes the form as follows,

$$\boldsymbol{y}(t) = (\boldsymbol{Z} *_d \boldsymbol{F})(t) = \sum_{i=0}^{k-1} \boldsymbol{f}_i \times \boldsymbol{z}_{t-d*i} \tag{5.16}$$

where $\boldsymbol{Z} = [\boldsymbol{z}_1, \boldsymbol{z}_2, \cdots, \boldsymbol{z}_L]$ denotes a sequence of temporal signals, each of which $\boldsymbol{z}_i$ is a vector signal. And $\boldsymbol{F} = [\boldsymbol{f}_0, \cdots, \boldsymbol{f}_{k-1}]$ represents a trainable filter with size k, each tap of which $\boldsymbol{f}_i$ is also a vector with the same size as $\boldsymbol{z}_i$, as shown in Fig. 5.6(a). Thus, the output of a 1-D convolution is $\boldsymbol{Y} = [\boldsymbol{y}_i, \cdots, \boldsymbol{y}_F]$, where F denotes the number of filters. As illustrated in (5.16), the DC-Conv operation is different from the conventional 1-D convolution operation in terms of two important concepts, namely causality and dilation:

- *Causality*: The causality is a fundamental requirement for temporal signal processing, which prevents the leakage of future information to the past. In other words, the current signal is completely dependent on its past but not relies on its future.

- *Dilation*: The dilation is to relax the consecutiveness restriction in convolution operations, i.e., $\boldsymbol{f}_i \times \boldsymbol{z}_{t-d*i}$, in the DC-Conv operation as shown in (5.16), where d denote the dilation factor. That is, the conventional one-dimensional convolution is a special case of dilated convolution with $d = 1$.

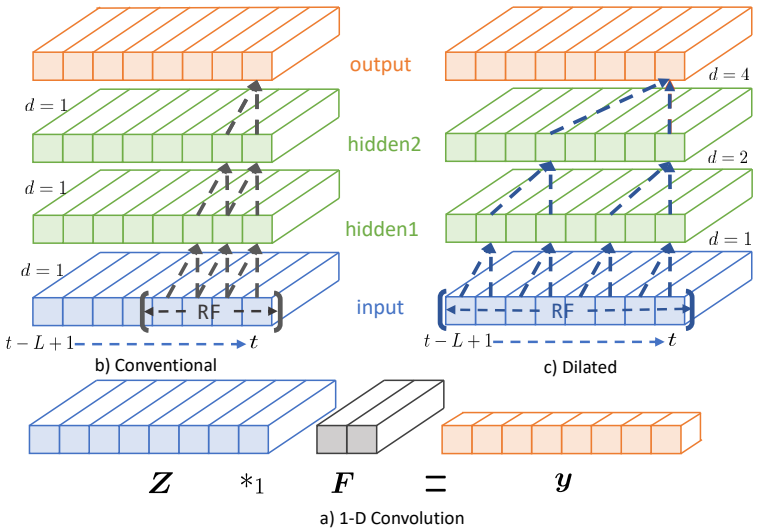


Figure 5.6: An example of dilated casual convolution: a) 1-D convolution with $d = 1$ and $k = 2$, b) Conventional convolution receptive field (RF) of a three-layer network with $d = 1$ and $k = 2$, c) Dilated convolution receptive field (RF) of a three-layer network $d = 2^{j-1}$ and $k = 2$.

The advantage of dilation is to enable a deep network to look back history of inputs much faster than that of the conventional 1-D convolution as shown in Fig. 5.6 [97, 99], as the dilation factor d is designed to grow exponentially with respect to the depth of the network. Accordingly, the DC-Conv networks with exponential dilation factor could achieve the same receptive field as the conventional 1-D convolution without a very deep network structure.

Residual Connection: The residual connection is to add a path bypassing some layers in a very deep network as shown in Fig. 5.7, which has become a prominent architecture in deep learning [100]. The mapping enabled by the residual path could be expressed as follows,

$$\boldsymbol{y} = \text{Activation}(\mathcal{M}(\boldsymbol{x})) = \text{Activation}(\mathcal{F}(\boldsymbol{x}) + \boldsymbol{x}) \tag{5.17}$$

where $\mathcal{M}(\cdot)$ denotes the underlying mapping, while $\mathcal{F}(\cdot)$ denotes the actual mapping to be learned in training. In (5.17), it can be observed that the term "residual" originates from

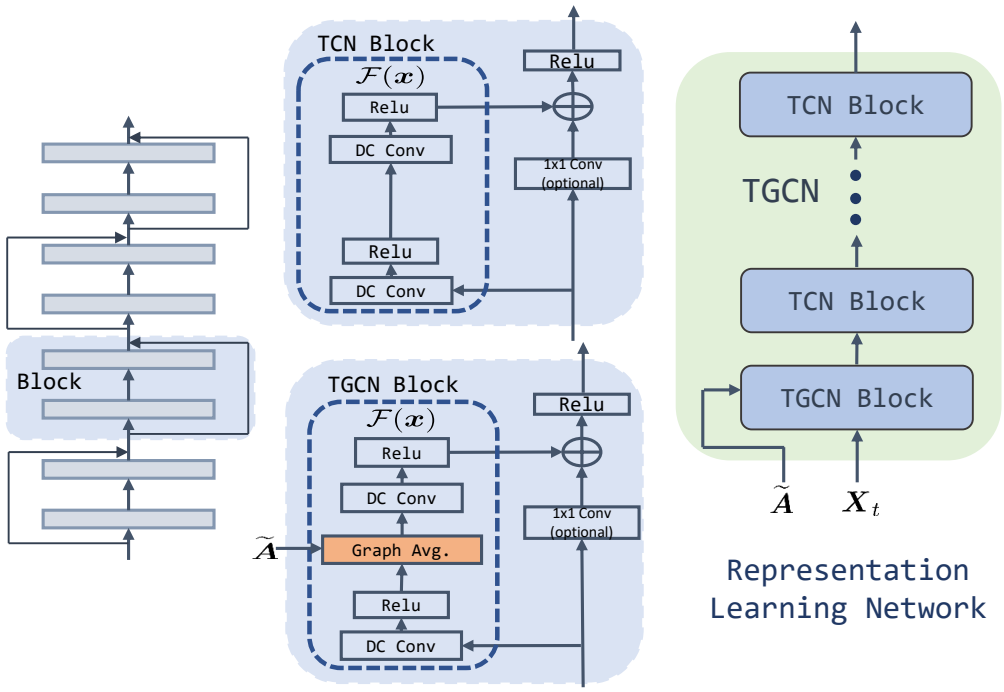


Figure 5.7: Representation Learning Network: Temporal Graph Convolutional Network (TGCN)

the mapping $\mathcal{F}$ actually learning the residual between $\mathcal{M}(\boldsymbol{x})$ and $\boldsymbol{x}$, i.e., $\mathcal{F}(\boldsymbol{x}) = \mathcal{M}(\boldsymbol{x}) - \boldsymbol{x}$, rather than the underlying mapping $\mathcal{M}(\boldsymbol{x})$. It is demonstrated in [101] that the residual connection would lead to the easy training of very deep networks with high accuracy gain, compared with the counterpart without the residual connection. With the residual shortcuts, the entire neural network could be formed in terms of blocks [97], each of which consists of DC-Conv layers and residual connection parallelly coupled as shown in Fig. 5.7. It is worth noting that two DC convolution layers share the same dilation factor in one TCN block. In addition, zero padding is employed to ensure the same sequence length in both the input and the output of TCN blocks.

A2. Spatiotemporal Modeling:

TCN blocks are employed to account for the temporal relevancy underlying structure of the input $\boldsymbol{X}_t$. However, the spatial relevancy among cells encoded by graph adjacency matrix

$\boldsymbol{A}$ is not yet touched so far. In our previous work [63], the graph convolutional networks (GCN) has been demonstrated that it can successfully capture the spatial relevancy on the task of mobile demand forecasting. In this work, we propose to further employ the concept beneath GCN combined with TCN blocks as discussed previously, to introduce the proposed *TGCN block* that can account for both the temporal and spatial structures of the input.

The approximated graph convolution operation proposed by Kipf and Welling in [76] takes the form as follows,

$$\boldsymbol{y} = g_\theta^{(1)}(\widetilde{\boldsymbol{L}}) \star \boldsymbol{Z} = \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \widetilde{\boldsymbol{A}} \widetilde{\boldsymbol{D}}^{-\frac{1}{2}} \boldsymbol{Z}\boldsymbol{\theta}, \tag{5.18}$$

where $\widetilde{\boldsymbol{A}} = \boldsymbol{I} + \boldsymbol{A} \in \mathcal{R}^{N \times N}$ and $\widetilde{\boldsymbol{D}}$ is a diagonal matrix, $\widetilde{\boldsymbol{D}}_{ii} = \sum_j \widetilde{\boldsymbol{A}}_{ij}$. Moreover, $\boldsymbol{Z} \in \mathcal{R}^{N \times F}$ is the node-based input matrix, each row of which is the feature vector of each node in the graph, while $\boldsymbol{\theta}$ denotes the graph filter. In (5.18), it can be observed that the approximated graph convolution is essentially to first filter the feature of each node independently (i.e., $\boldsymbol{z}_i\boldsymbol{\theta}$, where $\boldsymbol{z}_i$ is a row vector representing the feature vector of node i), and the output of each node by the graph convolution (5.18) is the average of the filtered results among itself and its neighbors as follow,

$$y_i = \frac{1}{|\mathcal{N}_i|} \sum_{j \in \mathcal{N}_i} \boldsymbol{z}_i \boldsymbol{\theta}, \tag{5.19}$$

where $\mathcal{N}_i$ denote the neighbor set of node i including itself. In this work, this operation is termed as *graph average*.

As a result, we propose to incorporate the graph convolution into the TCN block, to create a model that can simultaneously capture both the underlying temporal and the spatial structure of the input, termed as *temporal graph convolutional networks (TGCN)*. The TGCN

block is to add the graph average operation amid two dilated casual convolution layers in TCN blocks, as shown in Fig. 5.7. Let $\boldsymbol{H}^{j-1} \in \mathcal{R}^{N \times L \times F_{j-1}}$ denote the input of j-th TGCN block. The first DC-Conv layer in TGCN blocks will filter the input $\boldsymbol{H}^{j-1}$ along the temporal axis cell-by-cell and generate $\boldsymbol{H}^{j,1} \in \mathcal{R}^{N \times L \times F_j}$ after activation function. The output $\boldsymbol{H}^{j,1}$ will be further inputted to graph average based on the sparsity information provided by the graph adjacency $\widetilde{\boldsymbol{A}}$ as follows,

$$\boldsymbol{H}_i^{j,2} = \frac{1}{|\mathcal{N}_i|} \sum_{n \in \mathcal{N}_i} \boldsymbol{H}_n^{j,1}, \tag{5.20}$$

where $\boldsymbol{H}_i^{j,2} \in \mathcal{R}^{L \times F_j}$ denotes the output of node i after graph average operation and tensor $\boldsymbol{H}^{j,2} \in \mathcal{R}^{N \times L \times F_j}$ can be obtained by stacking $\boldsymbol{H}_i^{j,2}$ across all the nodes in the graph. Also, $\boldsymbol{H}_i^{j,2}$ will be further fed to the second DC-Conv layer and then combined with the result via the residual connection to generate the final output of TGCN $\boldsymbol{H}^j$. Details of TGCN is shown in Fig. 5.7.

5.4.2 Prediction Model Assembly

As observed in Fig. 5.4, both demand and mobility time series are more relevant to its history than the one from its neighbors. As a result, the proposed representation learning network first emphasizes the temporal relevancy by employing three TCN blocks to learn the high-level representation from inputs, as shown in Fig 5.7. Besides, one TGCN block is introduced as the first block in the representation network to embed the graph information in the proposed prediction model so that the spatial relevancy among cells could be accounted for by the proposed ITW prediction model.

The output network is a two-layer full-connection forward neural networks, in which the

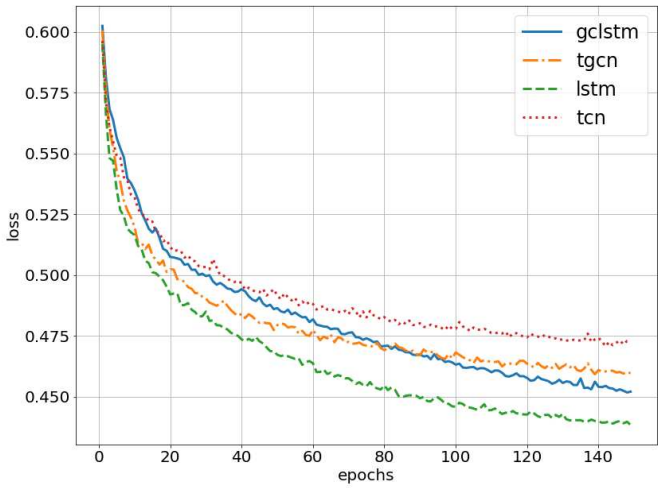
Table 5.1: TGCN, TCN, LSTM, and GCLSTM Specifications

	TGCN	TCN	LSTM	GCLSTM
Input	$\boldsymbol{X}_t, \boldsymbol{E}_t, \boldsymbol{A}$	$\boldsymbol{X}_t, \boldsymbol{E}_t$	$\boldsymbol{X}_t, \boldsymbol{E}_t$	$\boldsymbol{X}_t, \boldsymbol{E}_t, \boldsymbol{A}$
RL Networks	$[128, 128, 8]$	$[128, 128, 8]$	$[90, 90]$	$[90, 90]$
Kernel Size	2	2	N/A	N/A
Trainable Para.	$106,304$	$106,304$	$103,152$	$103,152$

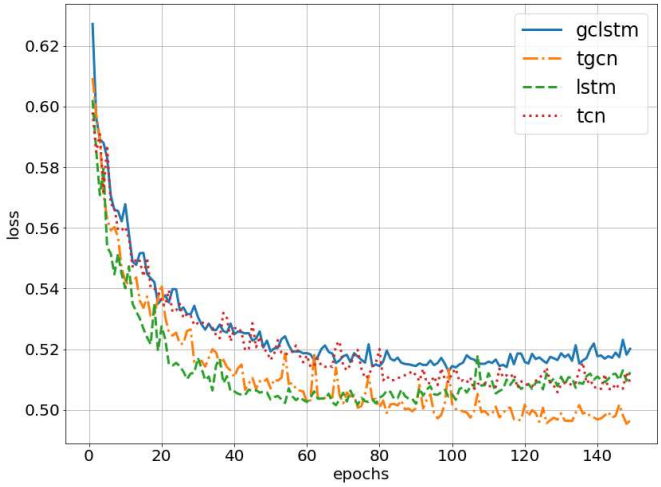
rectifier (ReLu) function is employed as the activation function in the input and hidden layers. Also, we employ sigmoid function as activation functions in the final layer, as the value of the desired outputs ($\boldsymbol{C}_t$, $\boldsymbol{S}_t$, and $\boldsymbol{D}_t$) is bounded between zero and one. In addition, it is worth noting that the output of representation learning network will be flattened cell-by-cell (i.e., the output will be formatted from $\mathcal{R}^{N\times L\times F_4}$ into $\mathcal{R}^{N\times F_f}$, where $F_f = L \times F_4$). So the output layer will generate the final result also in a cell-by-cell manner. Details of the proposed predictive model architecture are illustrated by Fig. 5.7.

5.5 Experiments Results

In this section, we validate the proposed problem formulation on cell ITW prediction and also compare the proposed temporal graph convolutional network (TGCN) with other sequence and sequence-graph models, namely long short-term memory (LSTM) [82], temporal convolutional networks (TCN) [97], and graph-convolutional LSTM (GCLSTM) [63]. The GCLSTM model in our previous work is a natural extension of the classic convolutional LSTM (convLSTM) [102] from grid-like spatiotemporal data to graph-based spatiotemporal data. The output network among compared models, including TGCN, TCN, LSTM, and GCLSTM, is kept the same as shown in Fig. 5.5 (one 32-unit hidden layer), while the specifications of their representation learning networks are detailed in Table 5.1. It is observed in



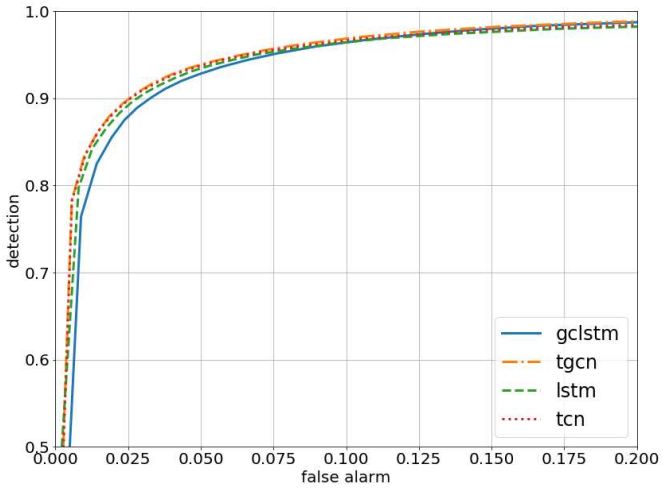
(a) training loss



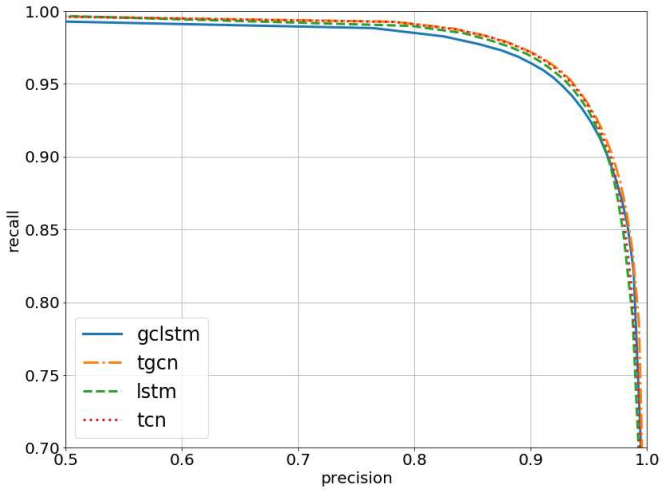
(b) validation loss

Figure 5.8: Training and validation comparisons in terms of designed cost and epochs.

Table 5.1 that the total trainable parameters are designed to be similar for fair comparisons. In addition, two baselines are also employed to validate the problem formulation and the performance of the proposed predictive model, namely the *baseline-Y* and *baseline-W*, whose performances are shown in Table 5.2. Specifically, the baseline-Y is to use the time window at the same time directly but one-day before as the ITW estimate, while baseline-W is to use the time window of one-week before. These two baselines are chosen based on the fact



(a) RoC



(b) Precision-Recall

Figure 5.9: RoC and precision-recall comparison

that both the demand and mobility time series are daily periodic and weekly periodic.

5.5.1 Model Training

All experiments in this work are carried out by the PyTorch deep learning framework [86]. In all experiments, the training-validation data are cleaned data extracted from the previously discussed data starting from Aug. 1st, 2016 to Nov. 30th, 2016, while the test data is extracted from Dec. 4th to Dec. 19th. The training and validation datasets are

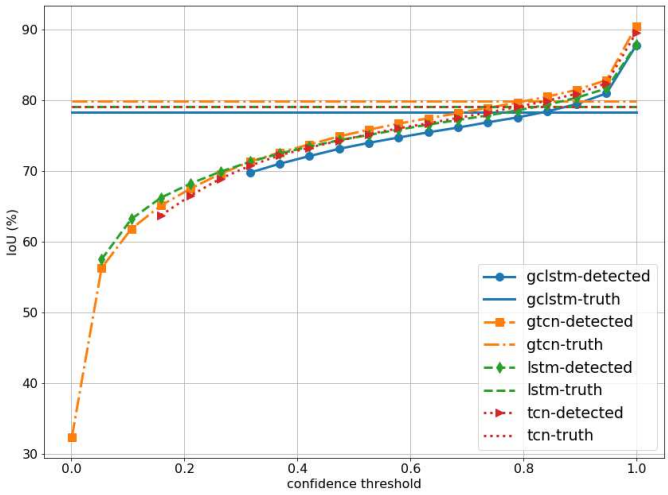
uniform-randomly selected from the entire train-validation data with the 95% and 5% of total samples, respectively. The length of history for ITW prediction is 6 or equivalently 2 hours, while the future horizon H is 18 or equivalently 6 hours. In this work, we employ the combine cost function, as stated in (5.9) to train all the compared models. The weight λ for ITW estimation in the cost function (5.9) is set to be 5.

Before training, all the features and inputs are first normalized by their mean and standard deviation. The dropout technique [103] is employed as the regularization during training the proposed predictive model. The optimization method utilized for training is Adam [104], which generally has a relatively better performance compared with the commonly used stochastic gradient descent (SGD) algorithm. In addition, weight normalization [105] is added to every DC-Conv layer in the TCN and TGCN to expedite the training speed as in [97]. Fig. 5.8 shows the loss of both the training dataset and validation dataset during training versus epochs, which suggests that the prediction model converge easily. It could be observed that the LSTM and GCLSTM could easily outperform the convolution-based models (TCN and TGCN) in terms of training loss. However, the validation loss suggests that the lower training loss does not necessarily lead to a good generalization. The LSTM and GCLSTM easily overfit starting around the 60th and 100th epoch, respectively.

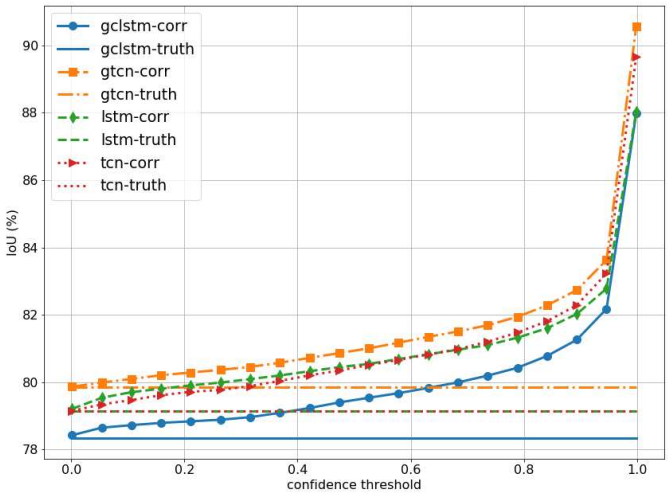
5.5.2 Testing: Performance Evaluation

In this work, two tasks are simultaneously fulfilled by our proposed ITW prediction model, namely ITW detection and ITW estimation. As a result, both the ITW detection and ITW estimation performance will be evaluated as follows.

ITW Detection

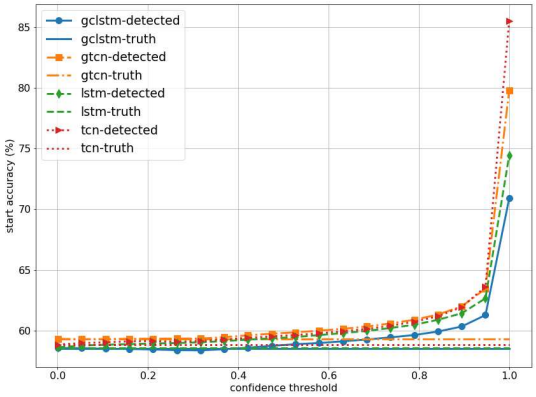


(a) ITW Detected: IoU

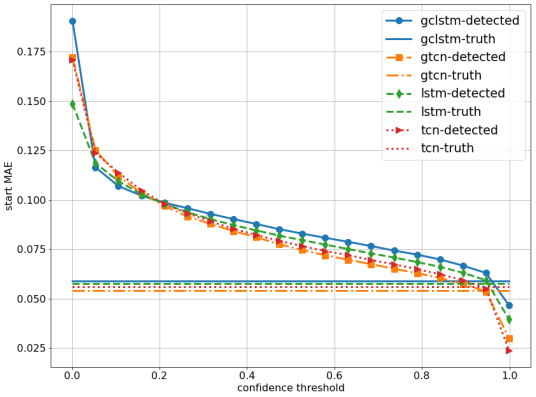


(b) ITW Correctly Detected: IoU

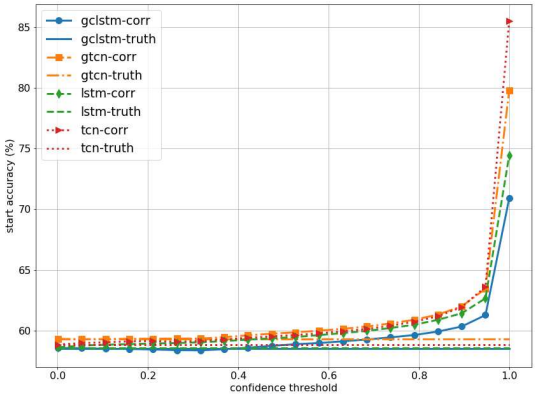
Figure 5.10: IoU comparison, where line legend "-detected", "-truth", and "corr" denote the ITW detected case, ITW presence case, ITW correctly detected case, respectively.



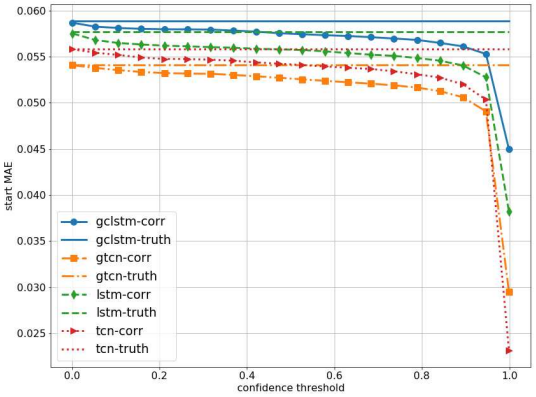
(a) ITW Detected: Start Accuracy



(b) ITW Detected: Start MAE



(c) ITW Correctly Detected: Start Accuracy



(d) ITW Correctly Detected: Start MAE

Figure 5.11: Accuracy and MAE comparisons of ITW estimation, where line legend "-detected", "-truth", and "corr" denote the ITW detected case, ITW presence case, ITW correctly detected case, respectively.

The receiver operating characteristics (RoC) and precision-recall curves are typically employed to assess the performance of detection or binary classification problems. The RoC curve consists of two core metrics, namely the detection and false alarm. In fact, the detection results could be categorized into the following 4 types (as shown in the table below), based

	True	
Predicted	True Positive	False Positive
	False Negative	True Negative

on which the evaluation metrics (detection/recall, false alarm, and precision) are defined as follows,

$$\begin{aligned} \text{Detection/Recall} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Negative}} , \\ \text{Precision} &= \frac{\text{True Positive}}{\text{True Positive} + \text{False Positive}} , \\ \text{False Alarm} &= \frac{\text{False Positive}}{\text{False Positive} + \text{True Negative}} . \end{aligned} \tag{5.21}$$

The F1 score is essentially the harmonic average of the recall and precision as follows,

$$\text{F1} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} . \tag{5.22}$$

Based on our proposed prediction model, one can obtain the RoC and precision-recall curves by adjusting the confidence threshold τ in the decision rule as stated in (5.4). Fig. 5.9 shows the RoC and precision-recall performance of all the compared models. It could be observed that all the models could well detect ITW in future horizons in terms of both RoC and precision-recall metrics, which verifies the validness of the proposed problem formulation, feature engineering, and prediction model structure. Also, the proposed TGCN can outperform others from the perspective of both the RoC and precision-recall.

Table 5.2: ITW Prediction Test Results

			GCLSTM	TGCN	LSTM	TCN	baseline-Y	baseline-W
Detection	Recall (%)		93.5899	94.1442	93.7525	**94.1974**	92.0651	92.0906
	Precision (%)		94.1415	**94.5600**	94.5171	94.3749	93.0336	92.5659
	F1 Score (%)		93.8649	**94.3516**	94.1332	94.2861	92.5468	92.3276
	False Alarm (%)		5.6931	**5.2942**	5.3162	5.4881	6.7387	7.2295
Estimation	Presence	IoU (%)	78.3439	**79.8565**	79.1544	79.1415	69.3181	68.8176
		s Acc. (%)	58.5265	**59.3068**	58.5818	58.8425	52.1011	50.6305
		d Acc. (%)	27.0026	**30.6853**	29.9848	30.3642	26.5762	25.6742
		s MAE	0.0589	**0.0541**	0.0577	0.0558	0.0991	0.1005
		d MAE	0.0924	**0.0840**	0.0886	0.0861	0.1080	0.1113
	Detected	IoU (%)	73.1486	**74.9044**	74.3670	74.2978	70.6629	69.6650
		s Acc. (%)	55.2978	**56.4903**	56.1036	56.1707	52.6492	50.8917
		d Acc. (%)	23.6972	**27.6653**	27.3187	27.5659	26.8558	25.8067
		s MAE	0.0851	**0.0775**	0.0819	0.0792	0.0928	0.0970
		d MAE	0.1060	**0.0949**	0.0992	0.0975	0.1076	0.1126
	Correctly	IoU (%)	79.3999	**80.8638**	80.4468	80.3472	77.8243	77.1938
		s Acc. (%)	58.7390	**59.7402**	59.3582	59.5186	56.5916	54.9790
		d Acc. (%)	25.1150	**29.2438**	28.8446	29.2013	28.8667	27.8793
		s MAE	0.0576	**0.0527**	0.0558	0.0542	0.0664	0.0684
		d MAE	0.0946	**0.0858**	0.0901	0.0878	0.0997	0.1032

ITW Estimation

As ITW start time and duration estimation are only meaningful when an ITW is predicted to appear, ITW estimation performance should be evaluated in terms of ITW detection. Hence, the ITW estimation performance will be evaluated in terms of three cases, namely *ITW Presence*, *ITW Detected*, and *ITW Correctly Detected*. The ITW Presence case is to assess the model on all the samples that ITW indeed presents, regardless of whether the prediction model can detect the ITW, which is aimed to evaluate the overall performance that how a prediction model can estimate ITW. The ITW detected case is to test the model on the samples that a prediction model claims ITW presence in the future horizon H, aimed to evaluate a prediction model in practice, while the ITW correctly detected is to assess a model on the samples that correctly identified. As the detection of all the models is not perfect, false alarms will appear in the ITW detected cases, which will be penalized when calculating IoU detailed later. In these three cases, the employed evaluation metrics on ITW estimation will be discussed as follows.

- *Accuracy.* As both the state time and duration of ITWs as defined in **Definition 5.1** are discretized, the accuracy is to assess how many start time and duration can be exactly predicted by each predictor as follows,

$$\text{sacc.} = \frac{\#(\tilde{S} = \hat{S})}{\text{total } \#} \text{ or } \text{dacc.} = \frac{\#(\tilde{D} = \hat{D})}{\text{total } \#} \tag{5.23}$$

- *Error.* The absoluble error between ground truth and a predict is also employed as assessment metrics, i.e., $|\hat{S} - \tilde{S}|$ or $|\hat{D} - \tilde{D}|$. To analysis the prediction error, we will show mean absolute error (MAE) of both the start time and the duration estimates.

- *Intersection over Union (IoU).* The previous two metrics only assess the start time and duration prediction independently, but not the quality of the overall ITW estimation. In this work, we borrow the intersection-over-union (IoU) metric from the object detection task from the field of computer vision [98] to assess how well the predicted time window overlaps with the ground truth as follows,

$$\begin{aligned} \text{Intersection} &= \min\{\hat{S}+\hat{D}, \tilde{S}+\tilde{D}\} - \max\{\hat{S}, \tilde{S}\} \\ \text{Union} &= \max\{\hat{S}+\hat{D}, \tilde{S}+\tilde{D}\} - \min\{\hat{S}, \tilde{S}\} \\ \text{IoU} &= \text{Intersection}/\text{Union} \end{aligned} \tag{5.24}$$

Among the above three evaluation metrics, the IoU shall be the most important one, as it directly reflects how a predictor performs in terms of window estimation. As for false alarms, the IoU will be directly set to be zero as the penalty, since the IoU metric when ITW is absent is meaningless.

Fig. 5.10 shows the IoU comparisons in three cases discussed previously in terms of confidence threshold τ. Intuitively, the IoU of all compared prediction models does not vary with the confidence threshold, which is also shown as a horizontal line in the figure. The tradeoff between precision and recall could be demonstrated by adjusting the confidence threshold. That is, the high confidence threshold suggests high precision performance but relatively low recall performance and vice versa. As a result, one can observe that the IoU in both the ITW detected and ITW correctly detected cases could grow with the increase of confidence threshold. Such phenomenon demonstrates the flexibility of our proposed prediction model facilitated by the designed confidence index. The IoU can reach 90% even

in the ITW detected case when the confidence threshold is high. Compared with other representation learning models, our proposed TGCN model is the best in terms of IoU.

Fig. 5.11 illustrates the ITW estimation performance in terms of start time and duration accuracy and MAE. It can be observed that the accuracy and MAE have a similar pattern as IoU performance for each compared prediction model. In Table 5.2, all the metrics discussed previously of all compared prediction models are compared with baselines. The confidence threshold for each prediction model is selected based on its optimal F1 score in Table 5.2. It can be observed that the proposed prediction model can have about 10% IoU improvement compared with two baselines employed (as shown by IoU in ITW presence case). By relaxing the detection performance to be the same level as the baselines via adjusting the confidence threshold, the IoU performance can be further improved according to the relationship between IoU and confidence threshold as shown in Fig. 5.10.

5.5.3 Discussion

The experiment results discussed previously have shown the validness and effectiveness of our proposed feature engineering scheme, ITW prediction model network structure, and temporal-graph convolutional networks. ITW can be well predicted by one-day or one-week ahead observations at the same time, due to the strong seasonality exhibited in mobile demand time series across the network. In this work, our proposed prediction model can effectively learn the pattern from recent history via representation learning network and project a better ITW detection and estimation by taking the periodic observations as exogenous inputs.

In the proposed ITW prediction model, the ITW existence confidence index plays an

important role, as it not only indicates the confidence of ITW presence during the model inference but also helps eliminate the negative impact of ITW non-presence samples on the ITW estimation performance during the model training. Such ITW estimation performance enhancement during model training is facilitated by the indicator function for ITW presence in the designed cost function (5.9) for model training as well as its tunable weight hyperparameter λ emphasizing ITW starting time and duration estimation. Besides, the confidence index threshold can be further employed to control the tradeoff between the recall performance and the IoU performance (or the precision performance) of the predictor, as shown in Figs. 5.10 and 5.11. In such a manner, our proposed model is flexible and can fulfill different robustness requirements in practice.

As for representation learning models, both the TCN and LSTM can effectively learn useful patterns for ITW prediction, as demonstrated by the experiment results discussed previously. Also, the spatial modeling by graph averaging employed in TGCN can further improve the ITW prediction, compared with the one by TCN and LSTM. However, due to the strong individual temporal relevancy of mobile demand time series—semivariogram is much smaller at 0 spatial distance as shown in Fig. 5.4—the overwhelming spatial modeling may lead to a deteriorated prediction performance. This is the reason why the GCLSTM has a worse performance compared with LSTM, where GCLSTM employs the graph averaging operation (5.19) in both two layers. TCGN employs one TGCN block (Fig. 5.7) involving the graph averaging operation (5.19) with other TCN blocks for temporal modeling, which could effectively capture both the spatial and temporal characteristics. Since the TGCN is similar to TCN yet with one additional graph averaging operation, the TGCN inherits the advantages and limitation of TCN. The advantages and disadvantages of TCN compared

with LSTM has been thoroughly discussed in [97] in terms of model training and inference. Overall, the proposed TGCN model demonstrates its superiority by experiment results by properly capturing both the spatial and temporal patterns.

5.6 Summary

In this work, we proposed to directly predict the idle time window based on subscribers' demand and mobility in mobile networks. A novel feature extraction scheme has been discussed to capture current trends of demand and mobility as well as the exogenous inputs accounting for the periodicity inherited in subscribers' demands. By modeling the spatial relevancy among cells as a graph, an ITW prediction model consisting of the representation learning network and the output network has been proposed, in which the temporal graph convolutional networks (TGCN) was further proposed to learn the high-level spatiotemporal patterns for the ITW prediction. Experiment results validated the effectiveness of the proposed idle time window prediction formulation and demonstrated the superiority of the proposed TGCN.

CHAPTER 6

CONCLUSIONS

In this dissertation, we studied the individual spatiotemporal learning in terms of privacy evaluation, while the aggregated spatiotemporal learning has been investigated in terms of predictive network management applications, namely the demand forecasting and the ITW prediction.

For privacy evaluation, we proposed a scalable multi-feature ensemble matching framework that integrates multiple matching results based on linear assignment problem formulation and the philosophy of majority voting. In addition, multiple spatiotemporal features were explored and exploited to distinctly characterize users in different semantic aspects to provide fuels for the proposed ensemble matching framework.

Next, we studied the traffic demand forecasting problem across the entire mobile network, which is considered as the aggregated behaviors of network users. In this work, we proposed to model the spatial dependency among cells by a dependency graph without the loss of spatial granularity. Hence, the graph convolutional networks (GCNs), the long short-term memory (LSTM) networks, as well as their integration GCLSTM have been employed to model the spatial and temporal dependencies for demand forecasting, respectively.

In addition to demand forecasting, we studied the per-cell idle time window (ITW) prediction in mobile networks based on subscribers' aggregated spatiotemporal behaviors. The ITW prediction was first formulated into a regression problem with an ITW presence confidence index. To predict the ITW, a deep-learning-based ITW prediction model was proposed,

consisting of a representation learning network and an output network. The temporal graph convolutional network (TGCN) was also proposed to implement the representation learning network, which effectively capture the graph-based spatiotemporal input features.

In summary, these spatiotemporal studies of mobile big data have shown that the success of data mining on mobile big data heavily relied on the specific spatiotemporal modeling for specific applications. It also suggested that graph is an excellent abstraction for spatial modeling. Although this dissertation has studied the individual and the aggregated spatiotemporal modeling for mobile big data application separately, it will be interesting to connect and generalize the individual and aggregated spatiotemporal modeling in the future.

REFERENCES

[1] J. Unnikrishnan, "Asymptotically optimal matching of multiple sequences to source distributions and training sequences," *IEEE Transactions on Information Theory*, vol. 61, no. 1, pp. 452–468, Jan. 2015.

[2] F. M. Naini, J. Unnikrishnan, P. Thiran, and M. Vetterli, "Where you are is who you are: User identification by matching statistics," *IEEE Transactions on Information Forensics and Security*, vol. 11, no. 2, pp. 358–372, Feb. 2016.

[3] M. Ficek and L. Kencl, "Inter-call mobility model: a spatio-temporal refinement of call data records using a Gaussian mixture model," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, Orlando, FL, Mar. 25–30, 2012, pp. 469–477.

[4] A. Ladd, K. Bekris, G. Marceau, A. Rudys, D. Wallach, and L. Kavraki, "Using wireless Ethernet for localization," in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, Lausanne, Switzerland, Sep. 30–Oct. 4, 2002, pp. 402–408.

[5] B. Ferris, D. Fox, and N. D. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proceedings of the 20th International Joint Conference on Artificial Intelligence (IJCAI)*, vol. 7, Hyderabad, India, Jan. 6–12, 2007, pp. 2480–2485.

[6] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor WiFi GraphSLAM," in *Proceedings of IEEE International Conference*

on Robotics and Automation (ICRA), Shanghai, China, May 9–13, 2011, pp. 1038–1043.

[7] M. Balakrishnan, I. Mohomed, and V. Ramasubramanian, "Where's that phone?: geolocating IP addresses on 3G networks," in *Proceedings of the 9th ACM SIGCOMM conference on Internet Measurement Conference*, Chicago, Illinois, Nov. 4–6, 2009, pp. 294–300.

[8] A. Metwally and M. Paduano, "Estimating the number of users behind IP addresses for combating abusive traffic," in *Proceedings of the 17th ACM International Conference on Knowledge Discovery and Data Mining*, San Diego, CA, Aug. 21–24, 2011, pp. 249–257.

[9] L. T. Le, T. Eliassi-Rad, F. Provost, and L. Moores, "Hyperlocal: Inferring location of IP addresses in real-time bid requests for mobile Ads," in *Proceedings of the 6th ACM SIGSPATIAL International Workshop on Location-Based Social Networks*, Orlando, FL, Nov. 5, 2013, pp. 24–33.

[10] X. Cheng, L. Fang, L. Yang, and S. Cui, "Mobile big data: The fuel for data-driven wireless," *IEEE Internet of Things Journal*, vol. 4, no. 5, pp. 1489–1516, Oct. 2017.

[11] ——, *Mobile Big Data*. Cham: Springer International Publishing, 2018.

[12] D. Z. Yazti and S. Krishnaswamy, "Mobile big data analytics: Research, practice, and opportunities," in *Proceedings of the 15th IEEE International Conference on Mobile Data Management (MDM)*, Brisbane, QLD, Jul. 14–18, 2014, pp. 1–2.

[13] Y. Park and E. Lee, "A new generation method of a user profile for information filtering on the internet," in *Proceedings of the 12th International Conference on Information Networking*, Tokyo, Japan, Jan. 21-23, 1998, pp. 261–264.

[14] R. J. Mooney and L. Roy, "Content-based book recommending using learning for text categorization," in *Proceedings of the 15th ACM Conference on Digital Libraries*, San Antonio, TX, 2000, pp. 195–204.

[15] M. Pazzani, J. Muramatsu, and D. Billsus, "Syskill & webert: Identifying interesting web sites," in *Proceedings of the Thirteenth National Conference on Artificial Intelligence*, Portland, OR, Aug. 4–8, 1996, pp. 54–61.

[16] W. Kim, L. Kerschberg, and A. Scime, "Learning for automatic personalization in a semantic taxonomy-based meta-search agent," *Electronic Commerce Research and Applications*, vol. 1, no. 2, pp. 150–173, Summer 2002.

[17] C. C. Tossell, P. Kortum, C. W. Shepard, A. Rahmati, and L. Zhong, "Getting real: a naturalistic methodology for using smartphones to collect mediated communications," *Human-Computer Interaction*, vol. 2012, no. 10, pp. 1–10, Apr. 2012.

[18] E. Nathan and A. Pentland, "Reality mining: sensing complex social systems," *Personal and Ubiquitous Computing*, vol. 10, no. 4, pp. 255–268, Mar. 2006.

[19] J. K. Laurila, D. Gatica-Perez, I. Aad, J. Blom, O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, "The mobile data challenge: Big data for mobile computing research," in *Proceedings of Nokia Workshop on Mobile Data Challenge in conjunction with International Conference on Pervasive Computing*, Newcastle, UK, Jun. 18–20, 2012.

[20] D. Wagner, A. Rice, and A. Beresford, "Device analyzer: Understanding smartphone usage," in *Proceedings of the 10th International Conference on Mobile and Ubiquitous*

Systems: Computing, Networking and Services, Tokyo, Japan, Dec. 2–4, 2013, pp. 195–208.

[21] D. T. Wagner, A. Rice, and A. R. Beresford, "Device analyzer: Large-scale mobile data collection," *ACM SIGMETRICS Performance Evaluation Review*, vol. 41, no. 4, pp. 53–56, Mar. 2014.

[22] S. Han, C. L. I, G. Li, S. Wang, and Q. Sun, "Big data enabled mobile network design for 5g and beyond," *IEEE Communications Magazine*, vol. 55, no. 9, pp. 150–157, Jul. 2017.

[23] D. Naboulsi, M. Fiore, S. Ribot, and R. Stanica, "Large-scale mobile traffic analysis: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 124–161, Firstquarter 2016.

[24] Q. Lv, Y. Qiao, N. Ansari, J. Liu, and J. Yang, "Big data driven hidden markov model based individual mobility prediction at points of interest," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 6, pp. 5204–5216, Jun. 2017.

[25] J. Johansson, W. A. Hapsari, S. Kelley, and G. Bodog, "Minimization of drive tests in 3GPP release 11," *IEEE Communications Magazine*, vol. 50, no. 11, pp. 36–43, Nov. 2012.

[26] F. Chernogorov and J. Puttonen, "User satisfaction classification for minimization of drive tests QoS verification," in *Proceedings of the 24th IEEE Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC)*, London, UK, Sep. 8–11, 2013.

[27] L. Fang, X. Cheng, L. Yang, and H. Wang, "Location privacy in mobile big data: User identifiability via habitat region representation," *Journal of Communications and Information Networks*, vol. 3, no. 3, pp. 31–38, 2018.

[28] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, Mar. 2008.

[29] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, Feb. 2010.

[30] X. Cheng, L. Fang, X. Hong, and L. Yang, "Exploiting mobile big data: Sources, features, and applications," *IEEE Network*, vol. 31, no. 1, pp. 72–79, January 2017.

[31] Y. De Mulder, G. Danezis, L. Batina, and B. Preneel, "Identification via location-profiling in GSM networks," in *Proceedings of the 7th ACM Workshop on Privacy in the Electronic Society*, Alexandria, Virginia, USA, 2008, pp. 23–32.

[32] Y.-A. de Montjoye, C. A. Hidalgo, M. Verleysen, and V. D. Blondel, "Unique in the crowd: The privacy bounds of human mobility," *Scientific Reports*, vol. 3, Mar. 2013.

[33] A. Cecaj, M. Mamei, and N. Bicocchi, "Re-identification of anonymized CDR datasets using social network data," in *Proceedings of IEEE International Conference on Pervasive Computing and Communication Workshops (PERCOM WORKSHOPS)*, Budapest, Hungary, Mar. 24–28, 2014, pp. 237–242.

[34] M. Gramaglia and M. Fiore, "Hiding mobile traffic fingerprints with GLOVE," in *Proceedings of the 11th ACM Conference on Emerging Networking Experiments and Technologies*, Heidelberg, Germany, Dec. 1–4, 2015, pp. 26:1–26:13.

[35] M. Gramaglia, M. Fiore, A. Tarable, and A. Banchs, "Preserving mobile subscriber privacy in open datasets of spatiotemporal trajectories," in *Proceedings of IEEE International Conference on Computer Communications (INFOCOM)*, Atlanta, GA, USA, May 1–4, 2017, pp. 1–9.

[36] B. C. M. Fung, K. Wang, R. Chen, and P. S. Yu, "Privacy-preserving data publishing: A survey of recent developments," *ACM Comput. Surv.*, vol. 42, no. 4, pp. 14:1–14:53, Jun. 2010.

[37] C. Riederer, Y. Kim, A. Chaintreau, N. Korula, and S. Lattanzi, "Linking users across domains with location data: Theory and validation," in *Proceedings of the 25th International Conference on World Wide Web*, Montreal, Quebec, Canada, Apr. 11–15, 2016, pp. 707–719.

[38] Z.-H. Zhou, *Ensemble methods: foundations and algorithms*. CRC press, 2012.

[39] W. Xu, X. Liu, and Y. Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, Toronto, Canada, Jul. 28 – Aug. 1, 2003, pp. 267–273.

[40] N. Gillis, "The why and how of nonnegative matrix factorization," *Regularization, Optimization, Kernels, and Support Vector Machines*, vol. 12, no. 257, 2014.

[41] L. Sweeney, "K-anonymity: A model for protecting privacy," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 10, no. 5, pp. 557–570, Oct. 2002.

[42] H. Zang and J. Bolot, "Anonymization of location data does not work: A large-scale measurement study," in *Proceedings of the 17th Annual International Conference on Mobile Computing and Networking*, Las Vegas, Nevada, USA, Sep. 19–23, 2011, pp. 145–156.

[43] Y. Song, D. Dahlmeier, and S. Bressan, "Not so unique in the crowd: a simple and effective algorithm for anonymizing location data," in *Proceedings of the 37th annual international ACM SIGIR conference*, Gold Coast, Queensland, Jul. 6–11, 2014, pp. 19–24.

[44] N. Takbiri, A. Houmansadr, D. L. Goeckel, and H. Pishro-Nik, "Limits of location privacy under anonymization and obfuscation," in *Proceedings of IEEE International Symposium on Information Theory (ISIT)*, Aachen, Germany, Jun. 25–30, 2017, pp. 764–768.

[45] L. Pappalardo and F. Simini, "Data-driven generation of spatio-temporal routines in human mobility," *Data Mining and Knowledge Discovery*, vol. 32, no. 3, pp. 787–829, May 2018.

[46] Z. Tu, F. Xu, Y. Li, P. Zhang, and D. Jin, "A new privacy breach: User trajectory recovery from aggregated mobility data," *IEEE/ACM Transactions on Networking*, vol. 26, no. 3, pp. 1446–1459, Jun. 2018.

[47] I. Liccardi, A. Abdul-Rahman, and M. Chen, "I know where you live: Inferring details of people's lives by visualizing publicly shared location data," in *Proceedings of the 2016*

CHI Conference on Human Factors in Computing Systems, San Jose, California, USA, 2016, pp. 1–12.

[48] A. Narayanan and V. Shmatikov, "Robust de-anonymization of large sparse datasets," in *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA, May 18–22, 2008, pp. 111–125.

[49] L. Rossi, J. Walker, and M. Musolesi, "Spatio-temporal techniques for user identification by means of gps mobility data," *EPJ Data Science*, vol. 4, no. 1, pp. 1–16, Aug. 2015.

[50] W. Cao, Z. Wu, D. Wang, J. Li, and H. Wu, "Automatic user identification method across heterogeneous mobility data sources," in *Proceedings of IEEE 32nd International Conference on Data Engineering (ICDE)*, Helsinki, Finland, May 16–20, 2016, pp. 978–989.

[51] R. Pellungrini, L. Pappalardo, F. Pratesi, and A. Monreale, "A data mining approach to assess privacy risk in human mobility data," *ACM Trans. Intell. Syst. Technol.*, vol. 9, no. 3, Dec. 2017.

[52] H. Wang, C. Gao, Y. Li, G. Wang, D. Jin, and J. Sun, "De-anonymization of mobility trajectories: Dissecting the gaps between theory and practice," in *Proceedings of The 25th Annual Network & Distributed System Security Symposium (NDSS'18)*, San Diego, CA, USA, Feb. 18–21, 2018.

[53] S. Chang, C. Li, H. Zhu, T. Lu, and Q. Li, "Revealing privacy vulnerabilities of anonymous trajectories," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 12, pp. 12 061–12 071, Dec. 2018.

[54] A. Pyrgelis, N. Kourtellis, I. Leontiadis, J. Serrà, and C. Soriente, "There goes wally: Anonymously sharing your location gives you away," in *Proceedings of IEEE International Conference on Big Data (Big Data)*, Seattle, WA, USA, Dec. 10–13, 2018, pp. 1218–1227.

[55] D. Kondor, B. Hashemian, Y. de Montjoye, and C. Ratti, "Towards matching user mobility traces in large-scale datasets," *IEEE Transactions on Big Data*, pp. 1–1, 2018.

[56] F. Xu, Z. Tu, H. Huang, S. Chang, F. Sun, D. Guo, and Y. Li, "No more than what i post: Preventing linkage attacks on check-in services," in *The World Wide Web Conference*, San Francisco, CA, USA, May 13–17, 2019, pp. 3405–3412.

[57] R. Jonker and T. Volgenant, "Improving the hungarian assignment algorithm," *Operations Research Letters*, vol. 5, no. 4, pp. 171 – 175, Oct. 1986.

[58] R. Jonker and A. Volgenant, "A shortest augmenting path algorithm for dense and sparse linear assignment problems," *Computing*, vol. 38, pp. 325 – 340, 1987.

[59] R. Sedgewick and K. Wayne, *Algorithms, 4th Edition.* Addison-Wesley, 2011.

[60] Jianbo Shi and J. Malik, "Normalized cuts and image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888–905, Aug. 2000.

[61] H. Zha, X. He, C. Ding, H. Simon, and M. Gu, "Bipartite graph partitioning and data clustering," in *Proceedings of the Tenth International Conference on Information and Knowledge Management*, Atlanta, Georgia, USA, Oct. 5–10, 2001, pp. 25–32.

[62] G. H. Golub and C. F. Van Loan, *Matrix computations.* JHU press, 2012.

[63] L. Fang, X. Cheng, H. Wang, and L. Yang, "Mobile demand forecasting via deep graph-sequence spatiotemporal modeling in cellular networks," *IEEE Internet of Things Journal*, pp. 1–1, 2018.

[64] F. Malandrino, C. F. Chiasserini, and S. Kirkpatrick, "Understanding the present and future of cellular networks through crowdsourced traces," in *Proceedings of the 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Macau, China, Jun. 12–15, 2017.

[65] M. Peng, D. Liang, Y. Wei, J. Li, and H. H. Chen, "Self-configuration and self-optimization in lte-advanced heterogeneous networks," *IEEE Communications Magazine*, vol. 51, no. 5, pp. 36–45, May 2013.

[66] O. G. Aliu, A. Imran, M. A. Imran, and B. Evans, "A survey of self organisation in future cellular networks," *IEEE Communications Surveys Tutorials*, vol. 15, no. 1, pp. 336–361, First 2013.

[67] R. Li, Z. Zhao, X. Zhou, G. Ding, Y. Chen, Z. Wang, and H. Zhang, "Intelligent 5G: When cellular networks meet artificial intelligence," *IEEE Wireless Communications*, vol. 24, no. 5, pp. 175–183, Oct. 2017.

[68] E. J. Kitindi, S. Fu, Y. Jia, A. Kabir, and Y. Wang, "Wireless network virtualization with SDN and C-RAN for 5G networks: Requirements, opportunities, and challenges," *IEEE Access*, vol. 5, pp. 19 099–19 115, Sep. 2017.

[69] H. Zhang, N. Liu, X. Chu, K. Long, A. H. Aghvami, and V. C. M. Leung, "Network slicing based 5G and future mobile networks: Mobility, resource management, and challenges," *IEEE Communications Magazine*, vol. 55, no. 8, pp. 138–145, Aug. 2017.

[70] D. Tikunov and T. Nishimura, "Traffic prediction for mobile network using Holt-Winter's exponential smoothing," in *Proceedings of the 15th International Conference on Software, Telecommunications and Computer Networks*, Split-Dubrovnik, Croatia, Sep. 2007.

[71] R. Li, Z. Zhao, X. Zhou, J. Palicot, and H. Zhang, "The prediction analysis of cellular radio access network traffic: From entropy theory to networking practice," *IEEE Communications Magazine*, vol. 52, no. 6, pp. 234–240, Jun. 2014.

[72] F. Xu, Y. Lin, J. Huang, D. Wu, H. Shi, J. Song, and Y. Li, "Big data driven mobile traffic understanding and forecasting: A time series approach," *IEEE Transactions on Services Computing*, vol. 9, no. 5, Sep. 2016.

[73] J. Zhang, Y. Zheng, D. Qi, R. Li, and X. Yi, "DNN-based prediction model for spatio-temporal data," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Burlingame, California, Oct. 31 - Nov. 3, 2016, pp. 92:1–92:4.

[74] J. Wang, J. Tang, Z. Xu, Y. Wang, G. Xue, X. Zhang, and D. Yang, "Spatiotemporal modeling and prediction in cellular networks: A big data enabled deep learning approach," in *Proceedings of IEEE Conference on Computer Communications (INFOCOM)*, Atlanta, GA, USA, May 1–4, 2017.

[75] N. Cressie and H.-C. Huang, "Classes of nonseparable, spatio-temporal stationary covariance functions," *Journal of the American Statistical Association*, vol. 94, no. 448, pp. 1330–1339, 1999.

[76] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," Paris, France, Apr. 24–26, 2017.

[77] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Advances in Neural Information Processing Systems 29*, D. D. Lee, M. Sugiyama, U. V. Luxburg, I. Guyon, and R. Garnett, Eds., 2016, pp. 3844–3852.

[78] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997.

[79] A. Sandryhaila and J. M. F. Moura, "Discrete signal processing on graphs," *IEEE Transactions on Signal Processing*, vol. 61, no. 7, pp. 1644–1656, Apr. 2013.

[80] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, May 2013.

[81] A. Sandryhaila and J. M. F. Moura, "Big data analysis with signal processing on graphs: Representation and processing of massive data sets with irregular structure," *IEEE Signal Processing Magazine*, vol. 31, no. 5, pp. 80–90, Sep. 2014.

[82] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[83] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured sequence modeling with graph convolutional recurrent networks," *eprint arXiv:1612.07659*.

[84] X. Shi, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, "Convolutional LSTM Network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds., 2015, pp. 802–810.

[85] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, May 2015.

[86] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," Long Beach, CA, USA, Dec. 9, 2017.

[87] X. Jian, R. A. Olea, and Y.-S. Yu, "Semivariogram modeling by weighted least squares," *Computers & Geosciences*, vol. 22, no. 4, pp. 387 – 397, May 1996.

[88] L. Fang, X. Cheng, H. Wang, and L. Yang, "Idle time window prediction in cellular networks with deep spatiotemporal modeling," *IEEE Journal on Selected Areas in Communications*, vol. 37, no. 6, pp. 1441–1454, Jun. 2019.

[89] M. Ismail, W. Zhuang, E. Serpedin, and K. Qaraqe, "A survey on green mobile networking: From the perspectives of network operators and mobile users," *IEEE Communications Surveys Tutorials*, vol. 17, no. 3, pp. 1535–1556, thirdquarter 2015.

[90] H. Ghazzai, M. J. Farooq, A. Alsharoa, E. Yaacoub, A. Kadri, and M. S. Alouini, "Green networking in cellular hetnets: A unified radio resource management framework

with base station on/off switching," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 7, pp. 5879–5893, Jul. 2017.

[91] S. Zhang, S. Zhao, M. Yuan, J. Zeng, J. Yao, M. R. Lyu, and I. King, "Traffic prediction based power saving in cellular networks: A machine learning method," in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, Redondo Beach, CA, USA, 2017, pp. 29:1–29:10.

[92] H. Shi and Y. Li, "Discovering periodic patterns for large scale mobile traffic data: Method and applications," *IEEE Transactions on Mobile Computing*, pp. 1–1, 2018.

[93] J. Ding, R. Xu, Y. Li, P. Hui, and D. Jin, "Measurement-driven modeling for connection density and traffic distribution in large-scale urban mobile networks," *IEEE Transactions on Mobile Computing*, vol. 17, no. 5, pp. 1105–1118, May 2018.

[94] C. Zhang and P. Patras, "Long-term mobile traffic forecasting using deep spatio-temporal neural networks," in *Proceedings of the Eighteenth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Los Angeles, CA, USA, 2018, pp. 231–240.

[95] Y. Liang, S. Ke, J. Zhang, X. Yi, and Y. Zheng, "Geoman: Multi-level attention networks for geo-sensory time series prediction," in *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 7 2018, pp. 3428–3434. [Online]. Available: https://doi.org/10.24963/ijcai.2018/476

[96] Y. Li, R. Yu, C. Shahabi, and Y. Liu, "Diffusion convolutional recurrent neural network: Data-driven traffic forecasting," in *Proceedings of International Conference*

on Learning Representations (ICLR), Vancouver, BC, Canada, Apr. 30 –May3, 2018. [Online]. Available: https://openreview.net/forum?id=SJiHXGWAZ

[97] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[98] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, "You only look once: Unified, real-time object detection," in *Proceedings of IEEE conference on computer vision and pattern recognition (CVPR)*, Las Vegas, NV, USA, Jun. 27–30, 2016, pp. 779–788.

[99] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *Proceedings of International Conference on Learning Representations (ICLR)*, San Juan, Puerto Rico, May 2–4, 2016.

[100] S. Jastrzebski, D. Arpit, N. Ballas, V. Verma, T. Che, and Y. Bengio, "Residual connections encourage iterative inference," in *Proceedings of International Conference on Learning Representations*, Vancouver, BC, Canada, Apr. 30 – May 3, 2018. [Online]. Available: https://openreview.net/forum?id=SJa9iHgAZ

[101] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[102] X. SHI, Z. Chen, H. Wang, D.-Y. Yeung, W.-k. Wong, and W.-c. WOO, "Convolutional LSTM network: A machine learning approach for precipitation nowcasting," in *Advances in Neural Information Processing Systems 28*, C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, Eds. Curran Associates, Inc., 2015, pp. 802–810.

[103] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[104] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proceedings of International Conference on Learning Representations (ICLR)*, San Diego, CA, May 7–9, 2015.

[105] T. Salimans and D. P. Kingma, "Weight normalization: A simple reparameterization to accelerate training of deep neural networks," in *Advances in Neural Information Processing Systems 29*, 2016, pp. 901–909.

LIST OF ABBREVIATIONS

APN	Access Point Name
ARIMA	Autoregressive Integrated Moving Average
CAPEX	Capital Expenditures
CBD	Central Business District
CDR	Call Detail Records
CN	Core Networks
CNN	Convolutional Neural Networks
CPT	Control-Plane Traffic
DC-Conv	Dilated Casual Convolution
DHR	Daily Habitat Regions
DS-Ensemble	Dual-Selection Ensemble
DS-Ensemble P&M	Dual-Selection Ensemble Partitioning and Matching
EMM	Eps Mobility Management
EPC	Evolved Packet Core
EPS	Evolved Packet System
GCLSTM	Graph Convolutional LSTM
GCN	Graph Convolutional Networks
ITW	Idle Time Window
IoT	Internet of Things
IoU	Intersection over Union
JV	Jonker-Volgenant
LAP	Linear Assignment Problem
LOWAN	Low Power Wide Area Network
LSTM	Long Short-Term Memory
MDT	Minimization of Drive Tests
MF-Ensemble	Matching-Filtered Ensemble
MME	Mobility Management Entity
MSC	Mobile Switching Center
NMF	Non-Negative Matrix Factorization
OPEX	Operational Expenditures
OTT	Over the Top
PAR	Peak-To-Average
PCEF	Policy Control Enforcement Function
PDN	Packet Data Network
PGW	Packet Data Network Gateway
PS	Core Packet Switched Core
RAN	Radio Access Networks
RB	Resource Block
RMR	Radio Measurement Reports
RNC	Radio Network Controller
RNN	Recurrent Neural Networks

RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
SARIMA	Seasonal ARIMA
SGD	Stochastic Gradient Descent
SGW	Serving Gateway
SON	Self-Organizing Networks
SSID	Service Set Identifier
TA	Tracking Area
TCN	Temporal Convolutional Networks
TGCN	Temporal Graph Convolutional Networks
UE	User Equipment
UPT	User-Plane Traffic
VFD	Visiting Frequency and Duration
VFO	Visiting Frequency Only
kLAP	k-Cardinality Linear Assignment Problem