UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

# Characterization of Human Social Mobility Patterns Applied to Mobility Modelling and Opportunistic Networks

Danielle Lopes Ferreira Astuto

**Orientador**

Dr. Carlos Alberto Vieira Campos

**Co-orientador**

Dr. Katia Obraczka

RIO DE JANEIRO, RJ - BRASIL
AGOSTO DE 2019

Characterization of Human Social Mobility Patterns Applied to Mobility Modelling and Opportunistic Networks

Danielle Lopes Ferreira Astuto

TESE APRESENTADA COMO REQUISITO PARCIAL PARA OBTENÇÃO DO TÍTULO DE DOUTOR PELO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA DA UNIVERSIDADE FEDERAL DO ESTADO DO RIO DE JANEIRO (UNIRIO). APROVADA PELA COMISSÃO EXAMINADORA ABAIXO ASSINADA.

Aprovada por:

Campos

Dr. CARLOS ALBERTO VIEIRA CAMPOS — UNIRIO

Katia Obraczka

Dr. KATIA OBRACZKA — UCSC

Sidney C. de L

Dr. SIDNEY CUNHA DE LUCENA — UNIRIO

Dr. CARLOS EDUARDO RIBEIRO DE MELLO — UNIRIO

Artur Ziviani

Dr. ARTUR ZIVIANI — LNCC

Dr. CELIO VINICIUS NEVES DE ALBUQUERQUE — UFF

RIO DE JANEIRO, RJ - BRASIL
AGOSTO DE 2019

A859

Astuto, Danielle Lopes Ferreira
Characterization of Human Social Mobility Patterns Applied to Mobility Modelling and Opportunistic Networks / Danielle Lopes Ferreira Astuto, – Rio de Janeiro, 2019.
152f.

Orientador: Carlos Alberto Vieira Campos.
Coorientadora: Katia Obraczka.
Tese (Doutorado) - Universidade Federal do Estado do Rio de Janeiro, Programa de Pós-Graduação em Informática, 2019.

1. Opportunistic Network. 2. Community Identification. 3. Data Mining. 4. Mobility Model. 5. Clustering Algorithms. I. Campos, Carlos Alberto Vieira, orient. II. Obraczka, Katia, coorient. III. Título.

*TO MY PARENTS, HUSBAND AND DAUGHTERS*

# Acknowledgments

I would like to thank a number of people who contributed directly or indirectly to the elaboration of this thesis, with technical, emotional and financial support. I would like to express my gratitude to my advisors Prof. Carlos Alberto Vieira Campos (UNIRIO) and prof. Katia Obraczka (UCSC, USA) for their guidance and invaluable feedback on my work. Your contributions were really important to the direction and development of my research and I have learned a lot under your supervision. Many thanks to the members of my thesis committee, Professor Arthur Ziviani (LNCC, Petropolis, RJ), Celio de Albuquerque (UFF, Niteroi, RJ) for their time, comments and suggestions in the conclusion of my work. I also gratefully acknowledge the financial support from CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) during my doctoral program.

I would also like to thank the colleagues at PPGI UNIRIO, especially professors Sidney de Lucena, Carlos Eduardo de Mello, and Marcio Barros for their valuable comments and inputs on my research that has really been very helpful. Besides, prof. Mariano Pimentel who taught me new perspectives on teaching and research challenges. Also, I would like to thank colleagues at Lab. DR, especially Diego de Souza who helped me with the implementation of the DACCOR protocol and Tiago Saraiva who helped me with the SDWN architecture experiments.

I would particularly like to thank my colleague and husband Bruno Astuto for the tremendous emotional and technical support. Your contributions, feedback, and discussions about my research undoubtedly helped a lot in the outcome of my work. Your emotional support, love, and words of encouragement helped me in a great way to complete my thesis successfully in time. Thank you very much, my love.

Special thanks to my dear parents for the affection and support dedicated to me throughout my life. "*Obrigada mãe por estar sempre presente e por me ajudar com as meninas para que eu pudesse concluir o trabalho mais tranquila. Obrigada pai, pelo* *incentivo, orientação e amor.*”

Many thanks to my brothers, Jefferson and Débora, and my brother-in-law Flávio Camhi for the emotional support and love during this important stage, and to my parents in-law Marcos and Miriam for their support and encouragement throughout my study. Finally, my deepest thanks to those who give meaning to my life, my amazing and beloved daughters, Eduarda and Pietra, for the love, understanding and tolerance during this time.

# ABSTRACT

Understanding human behavior and mobility will play a vital role in urban and environmental planning as cities continue to grow. Ubiquitous geo-location and localization technology and availability of bigdata-ready computing infrastructure have enabled the development of more sophisticated models to characterize human mobility in urban areas.

We start discussing the scale-free properties of some important human mobility characteristics, namely spatial node density and mobility degree, and show that they exhibit behavior that can be described by a power-law. Based on their power law characteristics, we derive analytical models for the spatial node density and mobility degree and showed that the data generated by the proposed analytical models closely approach empirical data extracted from the real mobility traces. Another contribution of our work is to use the proposed analytical models to build a synthetic mobility regime that is suitable for simulations of intelligent transportation systems.

Then, we present a novel approach to identify user communities in communication networks by using cluster techniques based on their geographical preferences. We describe our user community identification methodology in detail including how mobility features can be extracted from real mobility traces. We present results obtained when using our approach to identify user communities in three different mobility scenarios as well as an evaluation study comparing the performance of different clustering algorithms. In addition, a validation methodology that uses image-based similarity metrics is proposed, in order to assess the quality of the identified communities.

As a next step, we improve our community identification methodology by introducing a novel deep autoencoder neural network framework. Our experiments show that the proposed deep autoencoder increases the measured contact times between users belonging to the same community by up to 80% when compared to the average contact time when not considering community structures, and by up to 150% when compared to user communities extracted from raw datasets, i.e., without using the encoding extracted from applying the autoencoder to the pre-processed data. Moreover, our approach also increases contact time between members of the same community from 10% up to 125%, when compared to an alternate community extraction approach that uses Principal Component Analysis (PCA) instead. To the best of our knowledge, our proposal is the first to consider Deep autoencoder NNs to perform automatic extraction of non-linear features and mobility patterns from real mobility datasets.

We hypothesize that users that have similar geographical preferences have also similar interests and as such we used a deep autoencoder to pre-process raw mobility datasets that was able to more accurately uncover community structures which identifies groups of users sharing common geographical interests and temporal relationships. Thus, based on the deep autoencoder results we propose a community based routing protocol named DACCOR, which uses geographical preference features for making routing decisions. DACCOR uses neural network to train on these features and make next hop selection decisions. The performance of the proposed protocol is evaluated and compared with Epidemic and Prophet routing protocols in terms of delivery probability, overhead ratio, hop count and dropped messages.

**Keywords:** Opportunistic Network, Mobility Model, Community Identification, Neural Network, Autoencoder, Real Mobility Records, Data Mining, Human Mobility, Clustering Algorithms.

# Contents

# List of Figures

# List of Tables

# 1. Introduction

teste According to the United Nations' Department of Economics and Social Affairs[1], it is estimated that 55% of the world's population currently lives in urban centers and will reach 68% by 2050. As such, the greatest wave of city migration is yet to come and together with it a wide range of challenges raised by the need to improve the style and quality of life of a growing urban population. According to [Calabrese et al. 2014], a better understanding of city dynamics would allow for improved services as well as minimized environmental impact resulting from urban expansion.

Urban mobility, defined as the displacement of people across an urban region over time [Boeing 2017], is critical to understand the dynamics of an urban center. As cities grow, the complexity of urban transportation and transit systems and the time people spend in transit will greatly increase. As a result, expanded- and new transportation services will be required demanding deeper investigation into urban mobility [Louf and Barthelemy 2014, Albino et al. 2015]. Additionally, understanding human mobility in urban areas is crucial to other city management and planning applications such as public health, emergency response, education, entertainment, shopping, etc [Hess et al. 2015a].

A notable example of "urban analytics" [Senaratne et al. 2018, Bocconi et al. 2015], i.e., the use of information technology applied to urban planning, and in particular, Smart Cities, is the study of human mobility. For instance, information about people with similar geographical and temporal mobility behavior is critical for efficient and environmentally-aware transportation and transit planning. Additionally, information on vehicle trajectory will also be used to plan location of fueling (e.g., gas-, electric, fuel-cell) stations [Niu et al. 2016]; car-, bike-, and scooter sharing services can also take advantage of human movement patterns to optimize their deployments [Liu et al. 2017, Behrendt 2016].

Capturing patterns in human mobility is also important to understand and account

[1] https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html

for social interactions which affect a range of important services such as public health (e.g., infectious disease management), law enforcement and emergency response, social services, recreation and entertainment, etc [Zhong et al. 2014]. Furthermore, periodic and occasional contacts between people (and their computing/communication devices, e.g., smart phones, smart watches) present themselves as opportunities to exchange and forward data. *Opportunistic communication* is especially attractive in scenarios where existing communication infrastructure is heavily loaded (e.g., densely populated areas, hot spots), or its coverage is insufficient due to sparse infrastructure deployment (e.g., suburban regions) [Conti and Giordano 2014]. It also becomes critical in emergency response and disaster recovery operations as the existing communication infrastructure may become completely overloaded and/or compromised.

So much so that, over the last decade, network researchers have dedicated considerable attention to user mobility modelling and characterization. The importance of node mobility in designing networks has motivated researchers and practitioners to try to use realistic scenarios to drive the design and evaluation of wireless network protocols. Thus, this work starts with human mobility modeling and characterization to propose a mobility model that takes into account the behaviour found in real world mobility records. Then, considering the behavior identified in the user's mobility, we propose a method for extracting communities that considers the clustering characteristics found in real traces based on the users' geographical preference. Finally, a message forwarding protocol in opportunistic networks that is able to represent the behavior identified in the movement of users in urban areas is proposed. The following sections present in general the contributions of the present work.

## 1.1 Scale-Free Properties of Human Mobility and Applications to Intelligent Transportation Systems

Studying mobility traces is crucial to understanding the properties of the human mobility with the aim of providing human mobility characterization and to design efficient data forwarding protocols. With that in mind, in Chapter 2 we start by showing empirically (using real mobility traces collected in a variety of scenarios) that spatial node density (the number of nodes located in a given unit area) observed in human mobility can be modeled by a *Power Law*. We then propose the *Scale-Free Stochastic Mobility (SFSM)*, a model to describe analytically the heavy tail behavior exhibited by spatial node density and mobility degree (number of cells visited by a mobile node). We verify that the proposed model closely approximates empirical spatial density distributions resulting from real mobility records (e.g., GPS coordinates traces and/or records). As an example application, we use SFSM to build a waypoint-based mobility regime, named *Scale-Free Mobility Regime (SFMR)*, that is capable of generating mobility traces whose spatial node density distributions closely resemble the ones measured in real human mobility scenarios with the advantage of not requiring to extract model parameters from empirical datasets. We use our mobility regime to simulate node mobility in ad hoc network scenarios and show that the resulting average spatial node density closely resembles spatial density behavior observed in real mobility traces.

The main contributions of this part of the work includes:

- The proposal of a power-law based analytical model[2] used to build a waypoint mobility regime that can be used when developing and evaluating ITS protocols and applications. The proposed mobility regime is capable of generating mobility traces whose spatial node density and mobility degree distributions closely approximate the ones measured in real human mobility scenarios.

- An important feature of the proposed mobility regime is the fact that its parameters do not need to be extracted from real traces. To the best of our knowledge, our work is the first to explore the viability of using an analytical model to generate realistic mobility regimes whose parameters need not be extracted from traces. In fact, the model parameters can be set so that synthetic traces generated by it are able to mimic a variety of mobility scenarios in terms of number of clusters, their size, as well as the nodes' mobility degree. The ability of the model to generate synthetic mobility traces for scenarios motivated by ITS applications is demonstrated in Chapter 2.

- We also evaluate our mobility regime in terms of how accurately it reproduces user mobility characteristics, i.e., spatial density and mobility degree. We conduct a comparative study using four well-known mobility regimes, namely: Random Waypoint mobility (RWP), Natural [Borrel et al. 2005], Clustered Mobility Model (CMM) [Lim et al. 2006], and Self-similar Least Action Walk (SLAW) [Lee et al. 2009]. Our results show that our mobility regime is the one that most closely approximates metrics collected from simulations carried by the use of real mobility traces.

- Additionally, we expand our study of node mobility degree behavior and show that,

[2] This analytical model was published on IEEE Transactions on Intelligent Transportation Systems [Ferreira et al. 2018].

similar to a campus scenario, mobility degree in a vehicular scenario also exhibits heavy tail behavior, i.e., follows a Power Law.

- Our proposed mobility model considers key features of human and vehicular mobility such as node clustering and node mobility degree, which have been shown to significantly impact performance of mobile networks and their protocols [Mota et al. 2014,Song et al. 2010a]. We conduct a comparative study of the proposed mobility regime when evaluating network routing and show that routing exhibits comparable performance under our mobility regime when compared to the real trace. We also show that our model's fidelity to the real trace is considerably higher when compared to existing mobility regimes (i.e., RWP, Natural [Borrel et al. 2005], CMM [Lim et al. 2006], and SLAW [Lee et al. 2009]).

## 1.2 Identifying User Communities Based on Geographical Preferences and Its Applications to Urban and Environmental Planning

Motivated by the findings identified in human mobility and modeled by the Scale-Free Stochastic Mobility analytical model presented in Chapter 2, we begin Chapter 3 by studying the behavior and correlations among users as members of a common group. We consider a community detection problem in a social network, over a diverse set of scenarios. In a city, users (people or vehicles) can belong to several social groups (or communities), in which members of the same community have stronger and denser social connections than users from different communities. For instance, in social networks, communities correspond to group of friends who attend the same school, or who come from the same hometown [McAuley and Leskovec 2012].

Thus, we start by studying user mobility characteristics - including time spent in a given locale, average time between movements, or pause time, and average mobility speed. Such features are usually available from user mobility records such as GPS traces and Wi-Fi access point association records. Then, we propose a user community identification approach based on user mobility characteristics. The proposed methodology uses clustering techniques to identify user communities based on similar mobility characteristics extracted from real mobility traces. We investigate different clustering algorithms, each representing four main categories of cluster classifiers proposed in the literature [Jain et al. 1999,Cebeci and Yildiz 2015,Hasnat et al. 2015], namely: *Exclusive-*, *Overlapping-*, *Hierarchical-*, and *Probabilistic* Clustering. Additionally, we use Principal Component Analysis (PCA) and index metrics, as well as spatio-temporal information from real mobility traces to evaluate the performance of the different clustering techniques.

Moving forward towards human mobility-based community detection, we take advantage of other mathematical and computing tools to improve the extraction of community structures. As computational resources become more widely available through cloud- and edge computing services, machine learning techniques, such as neural networks (NNs), which not too long ago were considered totally prohibitive in terms of their computational demands, have now become mainstream tools to handle the enormous amounts of data being generated by sensing devices embedded mostly everywhere. A special category of NNs named Deep Autoencoders have been applied in a variety of domains, ranging from data augmentation, de-noising, activity and speech recognition, computer vision, to name a few [Liu et al. 2016].

In this way, in Chapter 4, we explore deep autoencoder architectures applied to learning user geographical permanence patterns in a variety of urban scenarios. Our main goal is to be able to perform automatic feature extraction among users and identify *user communities* based on their geographical preference similarities. To this end, we improve the former community methodology by introducing a novel deep autoencoder framework. We use a diverse urban mobility datasets to validate and evaluate our framework. Our experiments show that the proposed deep autoencoder increases contact times between users belonging to the same community by up to 80% when compared to the average contact time when not considering community structures and by up to 150% when compared to user communities extracted from raw datasets, i.e., without running data through the autoencoder. Moreover, our approach also increases contact time between members of the same community from 10% up to 125%, when compared to an alternate community extraction approach that uses Principal Component Analysis [Bishop and Nasrabadi 2007] instead.

Overall, the main contributions of this part of the work includes:

- A methodology[3] for user community identification that relies solely on features extracted from real human and/or vehicular mobility traces (e.g, obtained through GPS or Wi-Fi technology), which eliminates the dependence on information often difficult or expensive to obtain, such as data from telecommunication providers and online social networks.

- A comparative performance study of four different categories of clustering algorithms for user community identification using real mobility traces.

[3] A preliminary version of this methodology was published on Workshop de redes P2P, dinâmicas, sociais e orientadas a conteúdo (WP2P+) [Ferreira et al. 2016]. In addition, a more complete version was submitted to the IEEE Transactions on Intelligent Transportation Systems [Ferreira et al. 2019c].

- Based on the user communities identified, extraction of common features within communities, e.g., user geographical preference that can be leveraged by Smart City services and applications, e.g., intelligent transit.

- A validation methodology based on novel image-based similarity metrics. the proposed metrics allow to quantitatively assess the quality of the identified communities.

- We develop a Deep autoencoder Neural Network based approach[4] to perform automatic feature extraction in order to characterize user mobility in a variety of urban mobility scenarios. We use both GPS and WLAN traces in different urban settings (such as downtown areas and an University campus) that incorporate a variety of modes of transportation, including private vehicles, buses, taxis, pedestrians, and bikes.

- We demonstrate that our approach to automatically extract user mobility features from mobility traces can be used as input to clustering algorithms for constructing communities that group users with similar spatial and temporal mobility patterns.

- We show quantitative evidence that dimensionality reduction methods, when applied to mobility data before clustering, dramatically increases the quality of the community structures identified. We discuss which of such methods perform better and why.

- We create different autoencoder models, including fully-connected and convolutional architectures for processing mobility data ahead of clustering. We show that they are able to achieve better performance than those based on PCA according to a number of metrics.

- We also compare the effectiveness of the different autoencoder architectures in finding user spatial and temporal similarities, as well as discussing their computational cost.

- Finally, we discuss the impact of different autoencoder architectures and parameters on the performance of our automatic feature extraction framework.

[4] This proposal was submitted to a Special Issue on Deep Learning For Spatial Algorithms and Systems on the ACM Transactions on Spatial Algorithms and Systems journal [Ferreira et al. 2019b].

### 1.3 Routing Protocol and Data Dissemination for Opportunistic Networking

Thanks to the penetration of smartphones and their sensors in everyday life, mobile communication technologies are no longer simply a means to connect a mobile device to the network infrastructure. The convenient short range communication functions integrated in smart devices (e.g. Bluetooth and Wi-Fi) have given birth to some emerging applications such as Intelligent Transportation Systems, recommender systems, mobile data offloading, device to device communication, vehicular ad hoc networking, internet of things among others. Application-oriented paradigms are also emerging such as people centric networking, that puts people in the center, as the network is built with the users' devices. In this paradigm billions of users' mobile devices can be used for location-aware data collection, instrumenting the real world and generating observations – *crowdsensing* – and also to offer cloud computing services.

In such scenarios, usually the environment is saturated with mobile devices, that can self-organize into networks for local communication amongst themselves. These networks are generally partitioned in disconnected islands, which can be connected by infrastructure network such as Wi-Fi or cellular networking, if they exist. However, even if such infrastructure exists the cost and energy consumption can be significant. Therefore, due to the pervasive nature of such environments, opportunistic networks emerge as a means to provide or extend communication.

In the previous chapters we show that the community structure from real human mobility was successfully extracted. One natural step forward should be to take advantage of this knowledge into making more educated decisions on how, when and to whom forward incoming data to. There is a great study opportunity in defining a function or a set of utility functions that compute the probability of forwarding a given message, based on the community structure and its relationships of the communicating counterparts. Thus, our ultimate goal is to develop and validate more efficient forwarding protocols, based on accurate estimation of the underlying social structure of the mobile networking nodes. Chapter 5 presents the proposed Deep AutoenCoder Community-based Opportunistic Routing protocol (DACCOR) for data forwarding in opportunistic networks. DACCOR takes into account the user mobility feature extraction and the community detection method presented in the previous chapters based on user geographical preference extraction using deep learning. The proposed DACCOR forwarding scheme uses community information to make forwarding decisions between members of different communities, and the computed user relationship metric (i.e., called SSIM metric) to make the forwarding decision within the community.

The main contributions of this part of the work includes:

- The introduction of an user mobility feature extraction method and then present a novel community detection method, based on user geographical preference extraction using deep learning.
- We discuss the user relationship metric used to identify similarities between members of a community and how it can be calculated.
- We propose a Deep AutoenCoder Community-based Opportunistic Routing protocol (DACCOR) [5] for data forwarding in opportunistic networks.
- We show the effectiveness of the proposed opportunistic protocol through extensive experimentation using one synthetic and two real mobility records representing diverse urban mobility scenarios.
- We show the improvements in performance when using DACCOR against other forwarding mechanisms, for several networking metrics, and under different network loads. Results show that DACCOR is able to outperform the other protocols, and is able to deliver more, faster, and using less networking resources (e.g., resources such as network bandwidth and battery power). In other words, by using less bandwidth and less radio, DACCOR is able to dramatically decrease energy consumption, optimizing battery life.

## 1.4 Publications during the doctoral program

The following papers were produced during the doctoral program at UNIRIO:

- FERREIRA, DANIELLE L.; NUNES, BRUNO A. A.; OBRACZKA, KATIA. *Scale-Free Properties of Human Mobility and Applications to Intelligent Transportation Systems*. IEEE Transactions on Intelligent Transportation Systems, 19(11): 3736-3748, 2018.
- FERREIRA, DANIELLE L.; NUNES, BRUNO A. A.; and CAMPOS, C. A. V., *Uma metodologia de identificação de estruturas sociais em registros reais de mobilidade humana e veicular*. WORKSHOP DE REDES P2P, DINÂMICAS, SOCIAIS E ORIENTADAS A CONTEÚDO (WP2P+), XXXIV Simpósio Brasileiro de Redes de Computadores, 2016.

[5] This work will be submitted to the 16th IEEE International Conference on Mobile Ad-Hoc and Smart Systems [Ferreira et al. 2019a].

- FERREIRA, DANIELLE L.; NUNES, BRUNO A. A.; VIEIRA, CAMPOS, C. A. V.; OBRACZKA, KATIA . *Using Real Mobility Records for User Community Identification in Smart Cities*, IEEE Transactions on Intelligent Transportation Systems, under review, 2019.

- FERREIRA, DANIELLE L.; NUNES, BRUNO A. A.; CAMPOS, C. A. V.; OBRACZKA, KATIA, *A Deep Learning Approach for Identifying User Communities Based on Geographical Preferences and Its Applications to Urban and Environmental Planning*, Special Issue on Deep Learning For Spatial Algorithms and Systems, ACM Transactions on Spatial Algorithms and Systems, under review, 2019.

- FERREIRA, DANIELLE L.; DE SOUZA CLAUDIO; CAMPOS, C. A. V.; OBRACZKA, KATIA, *Deep autoencoder based community detection and its application to data forwarding in opportunistic networks*. IEEE MASS 2019: The 16th IEEE International Conference on Mobile Ad-Hoc and Smart Systems, under work, 2019.

- DE SOUZA, CLÁUDIO; FERREIRA, DANIELLE L.; CAMPOS, C. A. V.; DE OLIVEIRA, ANTONIO; CARDOSO, KLEBER; and MOREIRA, WALDIR, *Employing Social Cooperation to Improve Data Discovery and Retrieval in Content-Centric Delay-Tolerant Networks*. IEEE Access, under review, 2019.

- BARROS, R.; MOURA, H.; FERREIRA, D. L.; NUNES, B. A. A.; Lucena, S.; CAMPOS, C. A. V., *Um Framework para Experimentos Realísticos em Redes Sem Fio Definidas por Software*. In: XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, 2018, Campina Grande. XXXVI Simpósio Brasileiro de Telecomunicações e Processamento de Sinais, 2018.

- Souza, C. D. ; Ferreira, D. L. ; CAMPOS, C. A. V. *DIRESC: Um protocolo para descoberta e recuperação de dados em redes centradas em conteúdo e tolerantes a atraso*. In: 35ª Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2017, Belém. Anais do 35ª Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC), 2017.

- Souza, C. D. ; Ferreira, D. L. ; CAMPOS, C. A. V. . *A protocol for data discovery and retrieval in content-centric and delay-tolerant networks*. In: IEEE 86th Vehicular Technology Conference, 2017, Toronto. Anais do IEEE 86th Vehicular Technology Conference, 2017.

# 2. Scale-Free Properties of Human Mobility and Applications to Intelligent Transportation Systems

Characterizing and modeling node mobility is of critical importance in building intelligent transportation systems and their applications. In this chapter, we discuss the scale-free properties of some important human mobility characteristics, namely spatial node density and mobility degree, and show that they exhibit behavior that can be described by a power-law. Based on their power law characteristics, we derive analytical models for the spatial node density and mobility degree and showed that the data generated by the proposed analytical models closely approach empirical data extracted from the real mobility traces. Another contribution of our work is to use the proposed analytical models to build a synthetic mobility regime that is suitable for simulations of intelligent transportation systems. Finally, through network simulations, we show that ad-hoc network routing behavior under our mobility regime closely approximates routing behavior when the corresponding real trace is used.

## 2.1 Introduction

As computing and sensing devices become more prevalent and embedded in everything around us and wireless communication more ubiquitous, they have enabled a variety of emerging applications such as Intelligent Transportation Systems, or ITS. According to the European Union's Directive 2010/40/EU [201 40oj], ITS embodies services that employ "information and communication technologies in the field of road transport, including infrastructure, vehicles and users, and in traffic management and mobility management, as well as for interfaces with other modes of transport". It also includes the use of information and communication technologies to improve public- and mass transit systems efficiency and safety.

Understanding how people move in different environments and at different time scales is thus critical to enable ITS applications and services. The need for a deeper understanding of user mobility in wireless network environments has been well recognized (e.g., [Conti and Giordano 2014]) and has captured considerable attention from the networking community. CRAWDAD [CRAWDAD 2015] is a notable example of an initiative funded by the US' National Science Foundation (NSF) whose goal is to make real traces of network user activity publicly available, including mobility records. However, even with such efforts, availability of real human and vehicular mobility traces is still quite limited and so is availability of real testbeds. As an alternative, a number of research efforts focus on extracting features from real mobility records (e.g., mobility traces) to build realistic mobility generators that will drive simulation platforms.

One of the main challenges in constructing mobility generators is developing models that can capture the complexity of human and vehicular mobility, and their key features, in real-world settings [Karamshuk et al. 2011, Mota et al. 2014, Lin and Hsu 2014]. Two such key features are *clustering*, which can be defined as the tendency of people to agglomerate [Newman 2004] and *geographical preference*, which refers to people's preferences for particular locales. The work in [Nunes and Obraczka 2011] proposes *spatial node density*, defined as the number of users located in a given unit area, as a way to measure the degree of clustering associated with a given user population. Spatial node density has considerable impact on fundamental network properties such as connectivity and capacity, which in turn have direct influence on core network functions like medium access and routing. In work [Nunes and Obraczka 2011] the authors showed that users tend to congregate and form clusters, rather than being homogeneously distributed over an area.

To date, only a few synthetic mobility regimes have attempted to model spatial node density. Some examples include [Bettstetter et al. 2003] and [Hyytia et al. 2006, Mitsche et al. 2014], which propose analytical models to study spatial node density under Random Waypoint (RWP) mobility. In [Nunes and Obraczka 2014], spatial node density has been modeled using first order ordinary differential equations (ODEs) whose parameters are extracted from real mobility traces. Using real traces to set values of model parameters is not ideal especially because of limited trace availability which may yield parameters that are specific to certain scenarios.

Song at al. [Song et al. 2010a] investigated users' geographical preferences and found that the number of distinct regions visited by users follows scaling laws. In addition, they observed that the power law parameter is related to the probability users visit new regions over time. Their study used data collected from cellular networks, which, due to its low resolution, may not provide accurate representation of user behavior [Hess et al. 2015b].

In this chapter we show empirically, using real mobility traces collected in a variety of scenarios, that spatial node density and node mobility degree (i.e., the number of distinct cells visited by an individual node) observed in human mobility can be modelled by a power law. We then proposed a model to analytically describe the heavy-tail behavior exhibited by spatial node density and mobility degree resulting from user mobility, and confirmed that the proposed model closely approximates empirical spatial density distributions found in real mobility traces. As an example application of this analytical model, we used it to derive a mobility regime and showed how the proposed mobility regime closely resembles the real trace and the analytical model.

## 2.2 Mobility Traces

In our study, we use real traces collected in scenarios that are quite diverse, namely: a public park in the city of Rio de Janeiro, Brazil, an university campus in the USA, and a vehicular trace of taxis moving around the city of San Francisco, California, USA. Two of the traces were collected using GPS devices, while the third records Wireless LAN (WLAN) users as they associate and disassociate with the WLAN's Access Points (APs). These traces are summarized in terms of number of users, trace duration, and data sampling period in Table 2.1.

| Trace | # users | # Cells | Duration | Data Sampling |
|---|---|---|---|---|
| Quinta [Campos et al. 2009] (GPS) | 97 | 16 | 900s | 1s |
| SF Taxis [Piorkowski et al. 2009] (GPS) | 483 | 1600 | 24 days | 1 to 3 mins |
| Dartmouth [Kotz et al. 2009] (WLAN) | 6524 | 1776 | 60 days | A/D events |

Table 2.1: Summary of user mobility traces considered in our study.

In Table 2.1, *Quinta* refers to the "Quinta da Boa Vista Park" trace [Campos et al. 2009] which is a GPS trace collected as people walked through the Quinta da Boa Vista public park in the city of Rio de Janeiro, Brazil. *SF Taxis* [Piorkowski et al. 2009] refers to the vehicular mobility trace collected in the city of San Francisco, California, USA, where a fleet of approximately 500 taxi cabs was equipped with GPS trackers and had their positions logged for a period of 24 days. The *Dartmouth* [Kotz et al. 2009] trace logs user access to Dartmouth College's WLAN, in the form of AP association and disassociation events (denoted as "A/D events" in Table 2.1's Data Sampling column). While the original trace spans 5 years, we only used about 60 days of activity in which we identified higher activity (in terms of number of users and active APs).

### Cells

The area in which mobile users move is divided into equal sized squares, or *cells*. When considering infrastructure-based wireless LAN (WLAN) traces, such as the Dartmouth trace, cells are defined by the range of the APs. Thus, every cell corresponds to an AP; as a result, for the Dartmouth trace, the number of cells is equal to the number of APs. We employ similar criteria for the GPS traces, i.e., the Quinta and SF Taxi traces and we used cell sizes that correspond to AP average transmission range. In our experiments we used $140m$-by-$140m$ cells. To validate our choice of cell size, we ran experiments varying the cell dimension a few tens of meters up and down [Nunes and Obraczka 2014]. We observe no significant impact on the results when considering cell dimensions that are not too small or too big.

### Node Spatial Density and Mobility Degree

We define *node spatial density* as the number of nodes located in a given cell. We compute node spatial densities from the traces based on the trace's cell size and sampling period. We define also:

- *node spatial density distribution* as the ratio of cells containing on average $\geq x$ nodes over time, while
- *mobility degree distribution* is the ratio of nodes that visit a number $\geq n$ of cells.

### Defining an Intensity Map

We extract the quantity we call the *intensity* of the cell as the number of mobile nodes that visits a cell during a given time interval. For a total number of cells $N$, each cell $i \in \{1..N\}$ dividing the mobility area, is assigned an intensity $\mu_{i,T_t}$ at interval $T_t$. The *Intensity Map (IM)*, for $t \in \{0, 1, 2, ...\}$ is a N-dimensional vector, where each element in this vector has a value $\{\mu_{i,T_t} \in \mathbb{R} \mid \mu_{i,T_t} \geq 0\}$ that indicates how intense the activity in cell $i$ is. The IM for interval $T_t$ represents the spatial node density for that interval of time.

Similarly, we extract an *User Intensity Map (UIM)* composed of $L$ elements, where each element has a value $\{\mu_{l,T_t} \in \mathbb{R} \mid \mu_{l,T_t} \geq 0\}$ that indicates how mobile a node $l \in \{1..L\}$ is. In other words, $\mu_{l,T_t}$ provides the number of distinct cells visited by node $l$ during $T_t$.

### Node Speed and Pause Time

The distributions for node speed and pause times are also computed from the traces and are based on the trace's sampling period. In the Quinta trace for example, where the sampling period is $T = 1$ seconds, we build histograms for speed and pause time, which provide the relative frequency at which each value occurs. We compute node speed as $\frac{d}{\Delta t}$ where $d$ is the distance traveled between two consecutive entries in the GPS trace at times $t_1$ and $t_2$ and $\Delta t = t_2 - t_1$. We compute pause time for the Quinta trace as we will use for the experiments reported in Section 2.7.2. Pause time is calculated as $P = \Delta t$, if $d <$ *threshold*, or zero otherwise. We use *threshold*$= 0.5m$ since, in the Quinta trace, data is sampled every 1sec and pedestrians do not typically move much in 1sec. Furthermore, the Quinta trace was post-processed to account for possible GPS errors, as indicated in [Campos et al. 2009].

## 2.3 Power Law and Human Mobility

As discussed previously, spatial node density and mobility degree have considerable impact on fundamental network properties such as connectivity and capacity, which in turn have direct influence on core network functions like medium access and routing. In this section, we show that both spatial node density and mobility degree resulting from human movement in different scenarios exhibits heavy tail behavior, i.e., follows a Power Law. Power law has been used to describe a number of phenomena in communication networks, such as, node inter-contact times [Karagiannis T 2010], human movement [Song et al. 2010a, Noulas et al. 2012], and Internet measurements [Mahanti et al. 2013], to name a few.

Power laws are expressions of the form $P(x) \propto x^{-\alpha}$, where $\alpha$ is a constant parameter and $x$ are the measurements of interest. Few physical phenomena follow a power law for all values of $x$ [Clauset et al. 2009]. Usually, only the tail of the distribution, i.e., starting from a given minimum value, $x_{min}$, follows a power law. Thus, given a set of values that correspond to the observed data and the hypothesis that the data was extracted from a distribution that follows a power law, we want to verify if this hypothesis is plausible.

Fitting empirical data into a distribution that follows a power law is not trivial due to issues such as: (1) fluctuations that occur in the tail of the distribution representing rare events, and (2) difficulty in identifying the part of the distribution that actually follows the power law, i.e., $x_{min}$.

We fit the data from our mobility traces into a power law and compute its parameters by following the statistical framework described in [Clauset et al. 2009]. We then apply a *goodness-of-fit* test also from [Clauset et al. 2009], which generates a $p$ value, used to test whether a distribution follows or not a power law distribution. In other words, the test checks if a distribution following a power law is a plausible fit for the empirical data. This test computes the distance between the empirical data distribution and the hypothesis of the model. This distance is computed through the statistical test of Kolmogorov-Smirnov (KS), and is compared with the distance of measurements taken from a set of synthetic data drawn from the same model. The value of $p$ is defined as a fraction of the distance of the synthetic data that is greater than the empirical distance.

In summary, if the computed $p$ value is high (i.e., close to $1$), then the differences between the empirical data and the model can be attributed to statistical fluctuations. In the case where $p$ is closer to $0$, the model is considered not to be a plausible fit. Following the recommendation from [Clauset et al. 2009], we use $p < 0.1$ to reject the hypothesis that the empirical data follows a power law.

### 2.3.1 Spatial Node Density

This section presents the hypothesis test that the spatial node density is well represented by a power law distribution. Figure 2.1 shows the cumulative distribution functions (CDFs) of spatial node density for the Quinta (Figure 2.1(a)), Dartmouth (Figure 2.1(b)), and the San Francisco cab traces (Figure 2.1(c)), along with the fitting of the data according to a power law. For the sake of comparison, Figure 2.1 also plots the fitting of the same data using the exponential and log-normal distributions, as suggested in [Clauset et al. 2009]. This is done in order to ensure that not only a power law distribution is a good fit for the data, but also provides better fit when compared to other distributions.

Additionally, the graphs in Figure 2.1 show the values for the parameters of the fitted curves. It also shows the values of $p$ for the power-law fit for all three traces studied. We observe that $p$ is well above the reference threshold of $0.1$ used in [Clauset et al. 2009] for all three traces, validating the hypothesis that the spatial node density distribution follows a power law with parameters $\alpha$ and $x_{min}$ approximately equal to 2.5 and $10-20\%$ of the upper density, respectively.

As pointed out in the previous section, the value $x_{min}$ which determines where the heavy tail behavior begins is sometimes imprecise. In our experiments we found that this value ranges from $10\%$ to $20\%$ of the upper density (i.e., the maximum value of density measured). These findings are consistent with the well known "80/20" rule [Newman

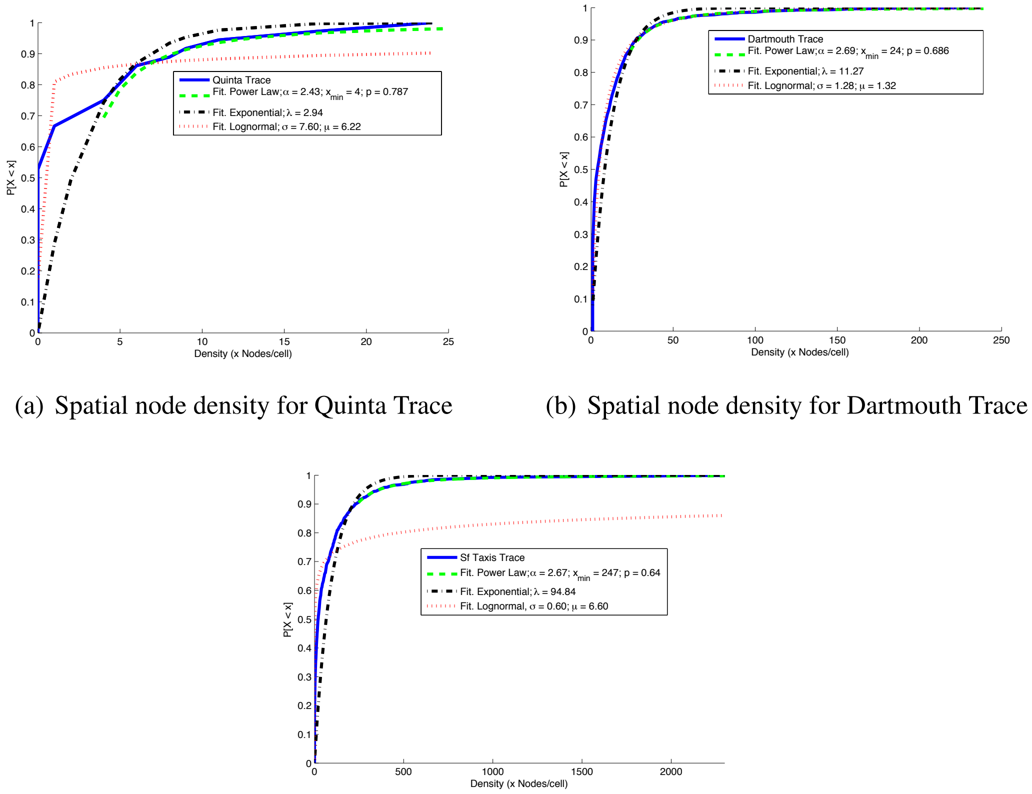
2005].

Here, the exponent $\alpha$ represents the slope of the curve, and can be extracted from the observed data by using the following formula [Clauset et al. 2009]:

$$\alpha = 1 + n[\sum_{i=1}^{n} \ln \frac{x_i}{x_{min}}]^{-1} \tag{2.1}$$

where $x_i$ are the measured values of $x$, and $n$ is the number of samples above $x_{min}$.

The parameters of the exponential and log-normal distributions were extracted from the data set by fitting the best curve that minimizes the distance to the real data, using Matlab's fitting toolbox. Table 2.2 compares the fitting errors between the different distributions (i.e., power law, exponential, and log normal) and the traces. The power law distribution fitting yields errors at least 2 orders of magnitude smaller than the fittings using the other distributions.



(a) Spatial node density for Quinta Trace

(b) Spatial node density for Dartmouth Trace

(c) Spatial node density for SF taxi trace

Figure 2.1: CDF of the spatial node density distribution for the Quinta, Dartmouth, and San Francisco cab traces.

### 2.3.2 Mobility Degree

*Node mobility degree*, or the number of different locations or cells that a node visits, is another important factor in mobile networks. For example, in disruption-tolerant networks (DTNs) or social networks, a node's degree of mobility will directly affect the node's node ability to relay messages since a node that visits a greater number of locations would potentially have more opportunities of contacts with other nodes. Thus, mobility degree can be used to decide whether a node is a good candidate to act as a message relay and/or how many copies of a message the node should carry.

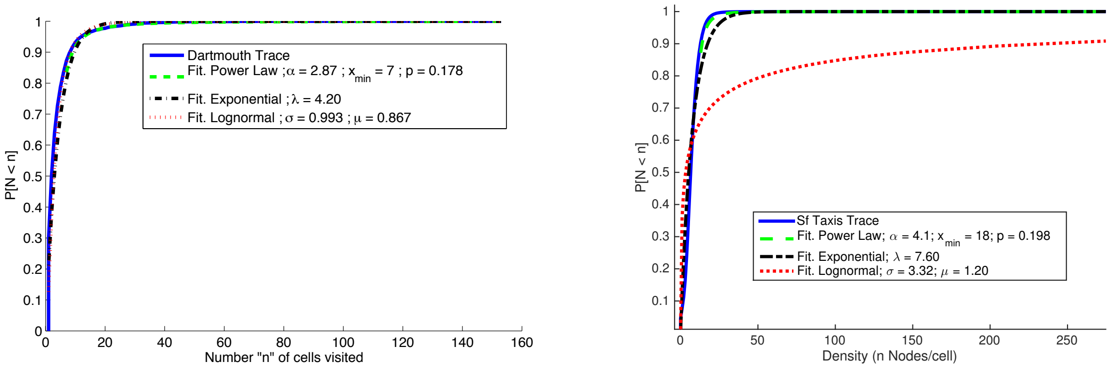
By applying the same method used in Section 2.3.1, we show that the cumulative distribution of the number of distinct locations visited by a node also presents a heavy tail behavior, i.e., the hypothesis that node mobility degree follows a power law distribution is also plausible.

Figure 2.2 shows the CDF of the distribution of the number of cells visited by users for the Dartmouth (Figure 2.2(a)) and SF Taxi traces (Figure 2.2(b)), along with the fitting of the data according to a power law (using the method described in [Clauset et al. 2009]), exponential, and log-normal distributions [1]. Here we can also observe that the curve that approaches the real data the most is the power law fit, which attests to the fact that most users tend to have low mobility or be stationary, while a small portion of users are highly mobile and visit a large number of locations. Table 2.2 shows the mean square error of each fit for the spatial node density metric, regarding Dartmouth and SF Taxi traces. Similar to the spatial node density results, the power law distribution also shows fitting errors for mobility degree at least 2 orders of magnitude smaller than the other distributions for both Dartmouth ad SF Taxis traces, as can be observed in Table 2.3.

| **Distribution** | **SF Taxis**<br>(density) | **Quinta**<br>(density) | **Dartmouth**<br>(density) |
|---|---|---|---|
| Power Law | **2.6306e-06** | **5.7428e-04** | **7.8624e-06** |
| Exponential | 0.0173 | 0.0390 | 0.00380 |
| Log-normal | 0.0154 | 0.0192 | 8.4840e-04 |

Table 2.2: Mean square error resulting from power-law, exponential, and log-normal fitting of the traces' spatial node density.

[1] Since nodes in the Quinta trace visit a relatively small number of locations, the trace does not exhibit enough mobility to be statistically representative of node mobility degree. As such, we do not use the Quinta trace in our mobility degree characterization.



(a) Node mobility degree for Dartmouth trace

(b) Node mobility degree for SF Taxis trace

Figure 2.2: CDF of the node mobility degree for the Dartmouth, and San Francisco cab traces.

| **Distribution** | **SF Taxis** | **Dartmouth** |
|---|---|---|
| | (mob. degree) | (mob. degree) |
| Power Law | **6.0948e-05** | **5.8607e-05** |
| Exponential | 0.0025 | 0.0022 |
| Log-normal | 0.0461 | 0.0014 |

Table 2.3: Mean square error resulting from power-law, exponential, and log-normal fitting of the traces' node mobility degree.

## 2.4 Scale-Free Stochastic Model

This section presents an analytical model for the spatial node density and node mobility degree, i.e., the number of cells visited by a mobile node. The proposed analytical model, named Scale-Free Stochastic Mobility (SFSM), is based on spatial node density's and node mobility degree's power-law behavior, as shown in Section 2.3. SFSM's contributions include the ability to: (1) express analytically these key features of human mobility which explains the formation and maintenance of clusters, and (2) generate mobility regimes that follow the observed power-law behavior of user mobility in real scenarios without the need to extract parameters from real traces. In Section 2.5, we exemplify SFSM's latter contribution by presenting an SFSM-based mobility regime.

### 2.4.1 Spatial Node Density as a Stochastic Process

Motivated by the empirical results presented in the previous section we now seek to model the spatial node density by means of a stochastic process. To this end, we divide the cells in groups such that cells with the same number of nodes belong to the same group.

Then, we find the transition probabilities for a cell to migrate from its current group to another, either denser or sparser, group. These transition probabilities allow us to derive the node density distribution in the cells. In [Champernowne 1953], a similar model was presented for modeling the income of people living in the UK in the early 50's.

We consider that the spatial node density distribution of countable groups of cells follow a stochastic process, and the stochastic matrix remains constant over time. In such context and provided certain specific conditions discussed below are satisfied, the distribution will tend towards an equilibrium distribution dependent on the stochastic matrix but not on the initial distribution. Table 2.4 summarizes SFSM's notation.

| **Param.** | **Description** |
|---|---|
| $X_r$ | number of cells in each range $R_r$ |
| $X_s$ | number of cells in each range $R_s$ |
| $p_{rs}(t)$ | probability of cell in range $R_r$ who shifts to range $R_s$ |
| $p_{ru}(t)$ | ratio of cells in range $R_r$ that jumps $u$ ranges |
| $b$ | root of g(z) |
| $N$ | total number of cells |
| $y_{min}$ | lowest cell density |
| $y_s$ | lower bound of the number of cells in range $R_s$ |
| $10^h$ | extent of each range |
| $F(y_s)$ | distribution of the number of cells exceeding $y_s$ |

Table 2.4: Summary of SFSM notation.

We assume that cell density, i.e. the number of mobile users populating a cell, is divided into a number of proportionally distributed ranges. For example, we consider ranges per time interval to be $[1,2)$ nodes, $[2,4)$ nodes, $[4,8)$ nodes, $[8,16)$ nodes, and so forth.

We use smaller ranges for lower density values and larger ranges for higher density values, due to the fact that higher densities do not occur as frequently. This is a reasonable assumption since sparse cells occur in much greater numbers than dense cells, i.e. it is not uncommon for a small subset of the cells to account for most of the nodes in the entire network.

We then consider that the change in node density distribution in any individual cell in a given interval depends on its state in the previous interval and on a random process. In other words, we consider node density variation across these ranges as being a stochastic process. In fact, as users move, there are always new users coming into some cell and other users leaving. An acceptable assumption to make is that for each user leaving a cell, there is a cell welcoming that user in the next instant of time, and vice-versa. This assumption will imply that cell density is approximately constant over time and that each mobile node decides where and when to move. We also assume that the total number of cells in the system does not change with time as the region under study remains fixed.

Under such assumptions, to describe the spatial node density distribution, we first define $X_r(0)$ as the number of cells in each range $R_r$, $r = 0, 1, 2, ...$ at initial time $T_0$, and a series of matrices $p^{'}_{rs}(t)$ as the probability of cells of $R_r$ at time $T_t$ who are shifted to range $R_s$ in the next interval time $T_{t+1}$. Then, the density distribution $x_r(t)$ will be generated according to Equation (2.2).

$$X_s(t+1) = \sum_{r=0}^{\infty} X_r(t) p^{'}_{rs}(t) \tag{2.2}$$

If we consider that the ranges are sorted by size, where the lowest cell density range is $R_0$, then we are able to define a new set of stochastic matrices

$$p_{ru}(t) = p^{'}_{r,r+u}(t) \tag{2.3}$$

and rewriting Equation (2.2) as

$$X_s(t+1) = \sum_{u=-\infty}^{s} X_{s-u}(t) p_{s-u,u}(t) \tag{2.4}$$

$p_{ru}(t)$ carries the information on the ratio of cells in range $R_r$ which jumps a number $u$ of ranges in $T_t$.

It is well known that dense locations tend to continuously attract other nodes keeping its high density characteristics, and sparse cells tend to remain sparse [Barabási and Albert 1999]. This assumption is also corroborated by our previous findings in [Nunes and Obraczka 2011], where we showed that cell density does not change over time.

As such, the frequency distribution of $p_{ru}(t)$ in $u$, is likely to be centered around $u = 0$.

In practice, this implies that the probability of cells shifting upwards and downwards across density ranges changes very little over time. We thus keep $p'_{r,r+u}(t) = p_{ru}(t)$ constant over time.

Given the discussion above, let us assume that, for all values of $t$ and $r$, and for some fixed integer $n$, we have

$$p'_{r,r+u}(t) = p_{r,u}(t) = 0 \quad \text{if} \quad u > 1 \quad \text{or} \quad u < -n \tag{2.5}$$

i.e., no cell can move upwards by more than one range or downwards by more than $n >= 1$ ranges at a time.

$$\begin{gathered} p'_{r,r+u}(t) = p_{r,u}(t) = p_u > 0 \\ -n =< u =< 1 \quad \text{and} \quad u > -r \end{gathered} \tag{2.6}$$

Equation (2.6) is our basic postulate, which follows from our findings from Section 2.3.1, that has tested the hypothesis that spatial node density follows a power law. What Equation (2.6) tells us is that the probabilities of a cell shifting up and down along the ranges of cell densities are distributed independently of the current cell density. This is true despite the imposed threshold forbidding that a cell descends below a given number of ranking positions and the frequency distribution of $p_{rs}(t)$ assumption discussed above. This will lead to a density distribution which obeys a Pareto's law, at least asymptotically, for high cell density values.

We also need to assume that for every value of $r$ and $t$

$$\sum_{s=0}^{\infty} p'_{rs}(t) = \sum_{u=-r}^{\infty} p_{ru}(t) = 1 \tag{2.7}$$

which, according to (2.6), also implies

$$\sum_{u=-n}^{1} p_u = 1 \tag{2.8}$$

The assumption described by Equation (2.7) tells us that cell density preserve their identity over time, as described in Section 2.4.1 above.

We also need to make sure that the cell density process is not dissipative. In other words, cell density does not increase indefinitely without reaching an equilibrium distribution.

We can then denote

$$g(z) \equiv \sum_{u=-n}^{1} p_u z^{1-u} - z \tag{2.9}$$

Thus, our stability assumption is as follows:

$$g^{'}(1) \equiv -\sum_{u=-n}^{1} u p_u \quad \text{is positive.} \tag{2.10}$$

This means that for all cells, initially in any one of ranges $R_n, R_{n+1}, R_{n+2}...$, the average number of ranges shifted during the next time is negative.

Now we determine the equilibrium distribution corresponding to any matrix $p^{'}_{r,r+u}(t) = p_{r,u}(t)$ according to our assumptions. Owing to the uniqueness theorem mentioned above in Section 2.4.1, it will be sufficient to find any distribution which remains exactly unchanged under the action of the matrix $p'_{rs}(t)$ over time. Such distribution, when found, must be (apart from an arbitrary multiplying constant) the unique distribution which will be approached by all distributions under the repeated action of the matrix multiplier $p'_{rs}(t)$ over time.

If $X_s$ is the desired equilibrium distribution, we need by (2.3), (2.5), (2.6)

$$X_s = \sum_{u=-n}^{1} p_u X_{s-u} \quad \text{for all} \quad s > 0 \tag{2.11}$$

and

$$X_0 = \sum_{u=-n}^{0} q_u X_{-u} \quad \text{where} \quad q_u = \sum_{v=-n}^{u} p_r \tag{2.12}$$

We need only satisfy (2.11), since (2.11), (2.5), (2.6) and (2.7) ensure the satisfaction of (2.12) as well.

Now a solution of (2.11) is

$$X_s = b^s \tag{2.13}$$

where $b$ is the real positive root other than unity of the equation

$$g(z) \equiv \sum_{u=-n}^{1} p_u z^{1-u} - z = 0 \tag{2.14}$$

where $g(z)$ was already defined in (2.9).

Descartes' rule of signs establishes that (2.14) has no more than two real positive roots: since unity is one root, and $g(0) = p_0 > 0$, and $g'(1) > 0$ by (2.10), the other real positive root must satisfy

$$0 < b < 1 \tag{2.15}$$

Hence the solution in (2.13) implies a total number of cells by

$$N' = \frac{1}{1-b} \tag{2.16}$$

and, to arrange for any other total number $N$, we need merely modify (2.13) to the form

$$X_s = N(1-b)b^s \tag{2.17}$$

We can now assume that the proportionate extent of each range is $10^h$, and that the lowest cell density is $y_{min}$, then $X_s$ is the number of cells in the range $R_s$ whose lower bound is given by

$$y_s = 10^{sh} y_{min} \quad \text{from where} \quad \log_{10} y_s = sh + \log_{10} y_{min} \tag{2.18}$$

By summing a geometrical progression, using (2.17), we now find that in the equilibrium distribution of the number of cells exceeding $y_s$ is given by

$$F(y_s) = N.b^s \quad \text{from where} \quad \log_{10} F(y_s) = \log_{10} N + s.\log_{10} b \tag{2.19}$$

Now put

$$\alpha = \log_{10} b^{-1/h} \quad \text{and} \quad \gamma = \log_{10} N + \alpha \log_{10} y_{min} \tag{2.20}$$

Then it follows from (2.18) and (2.19) that

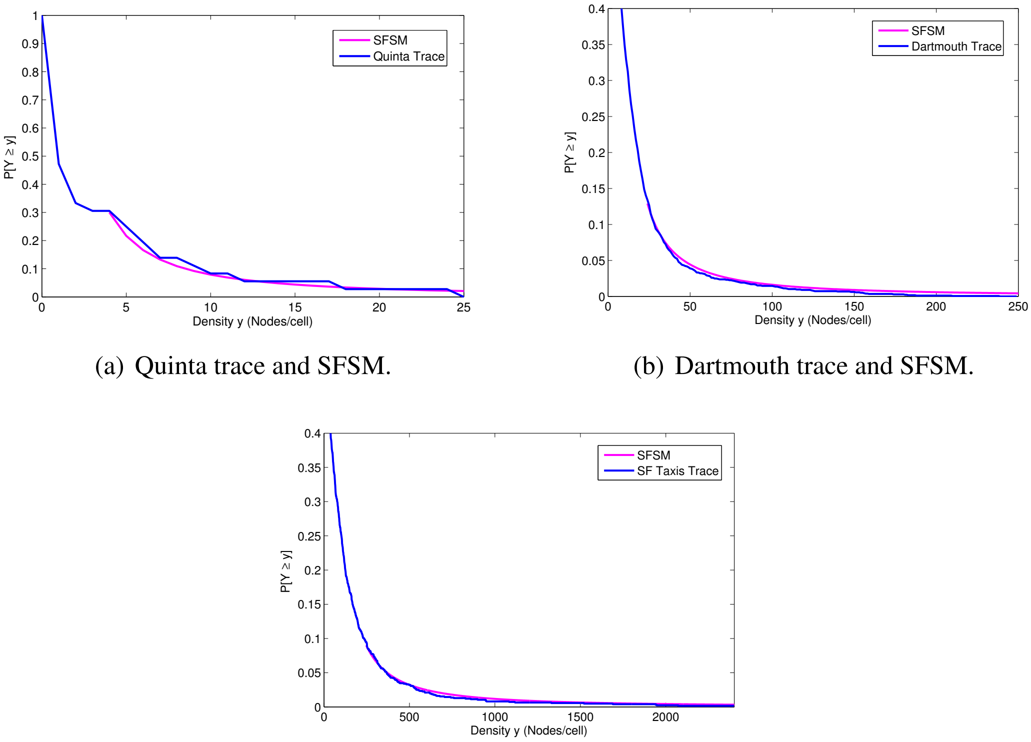
$$\log_{10} F(y_s) = \gamma - \alpha \log_{10} y_s \tag{2.21}$$

This means that for $y = y_0, y_1, y_2 ...$, the logarithm of the number of cells exceeding $y$ is a linear function of $y$. This states Pareto's law in its exact form [Clauset et al. 2009].

Thus, if all ranges are equal proportionate extent, our simplifying assumptions ensure that any spatial node density initial distribution will, with time, approach the exact Pareto distribution given by Equations (2.20) and  (2.21).

We validate the proposed SFSM model for spatial node density empirically by comparing it with mobility recorded in the Quinta, Dartmouth, and SF Taxis traces (which are summarized in Section 2.2). The graphs in Figure 2.3 show, for each trace, the probability of finding a cell that was visited by $y$ or more mobile users. They were computed by extracting the number of users visiting each cell during a given interval, i.e. $[800s, 900s]$ for the Quinta trace, and a random non-interrupted 24 hour interval for the Dartmouth and San Francisco traces. These intervals were chosen based on results presented in [Nunes and Obraczka 2011], which show that node density distribution does not change over time.

Figure 2.3 also shows the graphs obtained by running SFSM for each trace. The coefficients of the stochastic matrix (i.e., the probability $p_u$ of a cell changing $u$ ranges between two consecutive time intervals) used to parameterize SFSM were extracted from the traces so that we could compare to the empirical density and validate our model. The SFSM curves start at $x_{min} = 4, 24, 247$ for Quinta, Dartmouth and SF Taxis traces, respectively, and are derived in Section 2.3 and shown in Figure 2.1. To quantify SFSM's fidelity to the empirical spatial node density for values of density greater than a $y_{min}$, we define the *modeling error* as a perceptual difference between the distribution obtained from the real traces and the one computed from SFSM. In other words, the modeling error is calculated as the absolute difference between SFSM-derived spatial node density distribution and the distribution computed for the real trace, taken at each point in the x-axis in the tail of the distribution (i.e., ($> y_{min}$), divided by the corresponding value from the real trace density distribution. We computed the mean error and confidence intervals with a $95\%$ confidence level for the three traces studied. We average the errors computed for all points in the horizontal axis for values $> y_{min}$. The mean error and confidence interval for the Quinta trace shown in Figure 2.3(a) are $0.16\%[0.15\%, 0.19\%]$, respectively. Figure 2.3(b) shows the Dartmouth trace results, for which the mean error and confidence interval are $1.17\%[1.38\%, 0.96\%]$, respectively, and Figure 2.3(c) shows results for the San Francisco Taxi dataset with mean error and confidence interval of $0.43\%[0.47\%, 0.38\%]$, respectively.



(a) Quinta trace and SFSM.

(b) Dartmouth trace and SFSM.

(c) SF Taxi trace and SFSM.

Figure 2.3: Spatial node density distribution for the Quinta, Dartmouth, and SF Taxi traces compared against node density distributions generated by SFSM.

### 2.4.2 Mobility Degree as a Stochastic Process

Following the observation that, similarly to the spatial node density, mobility degree also exhibits power law behavior (see Section  2.3), we follow the same methodology used in Section 2.4.1 to derive a stochastic model for user mobility degree.

Recall that mobility degree is defined as the *number of cells visited by a mobile user over a given period of time*. As such, a user with low mobility visits a small number of cells, while a very mobile user visits a larger number of cells. In order to describe the mobility degree distribution, we define $\Theta_d(0)$, as the number $\Theta_d(0)$ of users in each mobility degree range $D_d$, $d = 1, 2, ...$ at the initial time $T_0$, and a series of matrices $p'_{dv}(t)$ as the probability of users in the range $D_d$ at time $T_t$ who shifted to range $D_v$ in the following interval time $T_{t+1}$. Then, the mobility degree distribution $\theta_d(t)$ will be generated according to

$$\Theta_{\nu}(t+1) = \sum_{d=0}^{\infty} \Theta_{d}(t)p'_{d\nu}(t) \tag{2.22}$$

Just as we did before, consider that the ranges are ordered by their size, where the lowest range of number of cells visited per user is $C_0$, then we can define a set of stochastic matrices such as

$$p_{df}(t) = p'_{d,d+f}(t) \tag{2.23}$$

where $p_{df}(t)$ indicates the ratio of users in $D_d$ who jumps over a number $f$ of ranges in $T_t$. Then, Equation (2.22) becomes:

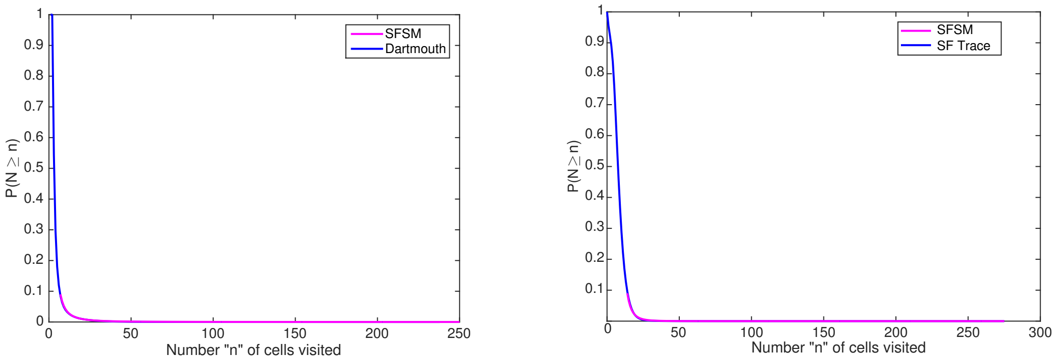
$$\Theta_{\nu}(t+1) = \sum_{f=-\infty}^{\nu} \Theta_{\nu-f}(t)p_{\nu-f,f}(t) \tag{2.24}$$

Following analogous derivations as in Section 2.4.1, we are then able to find the equilibrium distribution $F(\omega_{\nu})$ of the number of users whose number of visited cells exceeds $\omega_{\nu}$.

We validate the proposed SFSM model for node degree distribution empirically by comparing it with mobility recorded in the Dartmouth, and SF Taxis traces. Figure 2.4 shows the probability of a node visiting $n$ or more cells in a single trip, and by running SFSM for each trace. They were computed by counting the number of cells each upropmted user visits during the trace duration. The coefficients of the stochastic matrix (i.e., the probability $p_f$ of a user changing $f$ ranges between two consecutive time intervals) used to parameterize SFSM were extracted from the traces so that we could compare to the empirical density and validate our model.

## 2.5 Generating Scale-Free Mobility Regimes

Intelligent Transportation Systems have leveraged research and technology motivated by vehicular ad-hoc networks, or VANETs. In fact, many ITS services rely on the provision of an effective communication platform between vehicles, as well as between vehicles and road infrastructure (e.g., road-side units, sensors, etc). Also, communicating devices, such as laptops, smart phones, and even sensors now often carried by drivers and passengers can also be used to track vehicle mobility which is influenced by how humans



(a) Dartmouth trace and SFSM.

(b) SF Taxis trace and SFSM.

Figure 2.4: Node mobility degree distribution for Dartmouth and SF Taxi traces compared against mobility degree distributions generated by SFSM.

move, their habits, social links, and locality [Hossmann et al. 2011]. It is known that in the real world, nodes present clustering behavior and community structure [Newman 2004], with islands of connectivity and paths between clusters. For example, in VANETs, vehicles tend to group around traffic lights, junctions, toll, hazards, etc. The same behavior is also found in human mobility, where they tend to group in popular places, such as classrooms or cafeterias on campus, popular events, cafes, restaurants, etc.

As it is usually expensive and often logistically difficult to deploy and test ITS solutions in real world environments, network researchers and practitioners rely on simulation tools in order to develop and evaluate ITS services. Moreover, since we would like to be able to simulate realistic scenarios, mobility regimes that can closely represent real-world mobility are imperative in assessing the true impact and performance of ITS applications and protocols. In this section, we introduce the Scale-Free Mobility Regime (SFMR) that considers the previously discussed stochastic properties of node mobility, namely spatial node density and mobility degree, as well as nodes' geographical preferences. SFMR generates mobility regimes that reflect realistic human mobility behavior as characterized in Section 2.3. Next, we show how to use the Scale-Free Stochastic Model (SFSM) proposed in Section 2.4 to set SFMR's parameters.

In a nutshell, using SFMR to generate realistic mobility regimes works as follows: Before the simulation begins, cells with high node density (or clusters) are defined by specifying that the spatial node density in these cells is greater than a given threshold $y_{min}$; in other words, for these high density regions, we use the tail of the spatial density distribution to derive the probability that a node will choose a cell in the region. In the case of cells where density is below the $y_{min}$ threshold, we apply an uniform spatial density distribution, for simplicity. As shown in Section 2.7, our results indicate that uniform spatial node density is a reasonable approximation for low density regions. As part of our ongoing work, we have been studying more closely the impact of different known distributions to model cell density bellow $y_{min}$.

As we have previously discussed, one of SFMR's benefits is the ability to generate mobility regimes that result in spatial density distributions similar to the ones found in real mobile applications (as exemplified by the traces presented in Section 2.2) without the need to extract parameters from mobility traces. Below we provide a detailed description of SFMR, including how to set its parameters.

SFMR has two phases, namely initialization and movement. During the initialization phase (shown in Algorithm 1), nodes can be distributed in the geographic area according to an arbitrary‘ distribution. In the movement phase, for simplicity, we use a waypoint-based mobility regime, contending that simplicity is critical for wide adoption of any mobility regime. As such, the steps involved in the movement phase, as shown in Algorithm 2.

During initialization, described in Algorithm 1, some node $l$ may decide with probability $1 - P(\eta_l)$ if it will remain in the same cell, or if it will choose a destination with another cell with probability $P(\eta_l)$. The number of different cells $\eta_l$ visited by node $l$ is defined *a priori* by sampling from the computed distribution $F(\omega_v)$. $F(\omega_v)$ can be obtained as described in Section 2.4.2. The probability $P(\eta_l)$ that a user $l$ will leave a cell is computed in Equation 2.25, and this value of $P(\eta_l)$ is kept constant for every node $l$ during the simulation.

$$P(\eta_l) = \frac{\eta_l}{\sum_m \eta_m} \forall m \in \{1..L\} \tag{2.25}$$

When the simulation is in the movement phase, nodes behave as described in Algorithm 2. For every node, using a probability distribution given by $F(\omega_v)$, the node decides with probability $P(\eta_l)$ if it is going to move to another cell, as mentioned earlier. If the node decides to move, it chooses its next cell using a probability distribution given by $F(y_s)$. A $(x, y)$ destination is picked randomly inside the chosen cell. Then the node moves to that destination at a randomly chosen speed, uniformly distributed between $[V_{min}, V_{max}]$. When the node reaches its destination it pauses for some time, and repeats. We discuss how the values for $V_{min}$, $V_{max}$, and pause time are chosen below.

The decision of which cell is going to be the next destination is made with probability $P(\mu_i)$. We assume that the probability $P(\mu_i)$ that a node would choose cell $i$ as a next destination depends on the cell intensity $\mu_i$, that can be obtained by sampling from the computed distribution $F(y_s)$, of every cell $i$. The probability $P(\mu_i)$ is computed as in Equation 2.26, and this value of $P(\mu_i)$ (i.e., the probability that cell $i$ is chosen, given its intensity $\mu_i$) is kept constant for each cell $i$ during the simulation. Table 2.5 summarizes SFMR's notation.

| **Param.** | **Description** |
|---|---|
| $F_{y_s}$ | Distribution of the numb. of cells exceeding $y_s$ |
| $F_{\omega_v}$ | Distribution of the numb. of cells visited by a user that exceeds $\omega_v$ |
| $\nu_l$ | Numb of different cells visited by node $l$ |
| $\mu_i$ | Numb of different cells visited by node $l$ |
| $P_{\nu_l}$ | Prob. that node $l$ chooses to leave a cell |
| $P_{\mu_i}$ | Prob. that a node chooses cell $i$ as destination |
| $y_{min}$ | Lowest cell density |

Table 2.5: Summary of SFMR notation.

$$P(\mu_i) = \frac{\mu_i}{\sum_j \mu_j}, \forall j \in \{1..N\} \tag{2.26}$$

**Algorithm 1** SFMR: Initialization phase

```
*Distribute L nodes over the simulation area according to any given distribution*
**for** *each node* **do**
    Attribute the node degree probability $P(\eta_l)$, drawn from $F(\omega_v)$
**end**
```

As discussed previously, it is worth pointing out that the parameters for the proposed mobility regime do not need necessarily to be extracted from real mobility traces. In fact, the model parameters can be set and tuned in order to generate a variety of mobility scenarios in terms of number of clusters, their size, as well as the nodes' mobility degree. In the proposed model we need to set only 4 parameters, namely the speed range, pause time range, $y_{min}$, and the set of coefficients for the generating function in Equation 2.9. The tuning of these parameters will depend on the parameters for the scenario itself (e.g. total area, cell size, number of nodes, cluster size, etc). For the simulation results presented in the next section, we extracted the parameters from the traces for the sake of having a baseline (i.e., a real trace scenario) for a fair comparison of all the mobility regimes considered in our evaluation. That also shows that it is possible to mimic specific real world scenarios.

From the statistical study presented in Section 2.3, $y_{min}$ was found to typically fall

**Algorithm 2** SFMR: Movement phase

```
for each node do
    if node decides to move to another cell with probability $P(\eta_l)$ then
        Select next cell with probability, $P(\mu_i)$, drawn from $F(y_s)$
        Moves to destination using randomly speed between $[V_{min}, V_{max}]$
        pauses for a pause-time
    end
end
```

between 10% to 20% of the largest cluster (the highest node density). The coefficients of Equation 2.9 can be set according to the shape of the target density curve, considering: (1) the sizes of the clusters one wants to simulate and (2) the total population of nodes, which will provide an estimate of how many clusters of each size can be simulated. Equation 2.9 depends on the probability matrix of cells changing to another range (higher or lower). Depending on the scenario we would like to simulate, this probabilities can be set differently. For dense scenarios, where clusters are fewer and larger, such probabilities should be higher. For sparser scenarios, on the other hand the probability of choosing a given cell should vary little over the range of $i$.

## 2.6 Evaluation Methodology

We evaluate the proposed Scale-Free Mobility Regime (SFMR) in terms of how accurately it reproduces real user mobility according to spatial density and mobility degree when compared against real mobility traces. In our study, we also compare SFMR against four well-known mobility regimes, namely: Random Waypoint mobility (RWP), Natural [Borrel et al. 2005], Clustered Mobility Model (CMM) [Lim et al. 2006], and Self-similar Least Action Walk (SLAW) [Lee et al. 2009]. Our rationale for choosing these mobility models for our comparative performance study of SFMR is as follows. RWP, despite its limitations, has been widely used to evaluate wireless networks and their protocols. Natural and CMM were selected as representatives of the class of mobility regimes that follow the preferential attachment principle. As discussed in greater detail in Section 2.9, more recently proposed models have extended CMM, e.g., HCMM [Boldrini and Passarella 2010] and ECMM [Vastardis and Yang 2014] but preserve CMM's core preferential attachment based features; as such we use CMM, along with Natural, to represent preferential attachment based mobility regimes in our comparative analysis.

Similarly, SLAW is a well-known, widely cited mobility regime that accounts for social structure and social features. As described in Section 2.9, SLAW has inspired and has been extended by successors like SMOOTH [Munjal et al. 2011] and MobHet [Silveira et al. 2016]. This prompted us to select SLAW to represent mobility models that consider social interactions.

Additionally, we evaluate SFMR's fidelity to real user mobility by investigating how it affects network routing behavior, and consequently the efficiency of message dissemination in ITS, when compared to real mobility traces as well as to the mobility regimes listed above. We then start by describing these mobility regimes.

### 2.6.1 Random Way-Point Mobility

Random Way-Point (RWP) mobility is one of the simplest mobility regimes, and because of that, one of the most used when simulating mobile networks. According to RWP, during the initialization phase, nodes are placed in the simulation area using any given distribution. Mobile nodes stay in their current positions for *pause time* $P$ time units. This pause time period is usually drawn from a uniform distribution over an interval $[0, P_{max}]$, where $P_{max}$ is a pre-configured parameter. The pause time is drawn for every node individually and once it is over, each mobile node independently and uniformly chooses a new destination $(x_d, y_d)$ over the simulated area. The node then moves in a straight line to the newly chosen destination with a speed, also uniformly chosen over the interval $[v_{min}, v_{max}]$, where both $v_{min}$ and $v_{max}$ are also pre-configured parameters. Once the new destination is reached, the node pauses for some random time, chooses another destination and velocity as before and repeats this process until the simulation is over.

### 2.6.2 Preferential Attachment Based Mobility

Several synthetic mobility models rely on the so called preferential attachment principle [Barabási and Albert 1999]. As such, we also compare SFMR against mobility regimes that follow this principle, which is based on "attraction points". As representatives of the "preferential attachment" family of mobility regimes, we use for comparison the *Natural* [Borrel et al. 2005] mobility model and the Clustered Mobility Model (CMM) [Lim et al. 2006].

In Natural, a node's attraction to a given location is proportional to the location's "popularity". This popularity is proportional to the number of nodes already populating or moving towards this location. It is also inversely proportional to one's distance to the specific location. Thus, the probability $\Pi(a_i)$ that a node chooses an attraction point $a_i$ among all possible attractors is:

$\Pi(a_i) = \frac{\mathcal{A}_{a_i,z_k}}{\sum_j \mathcal{A}_{a_j,z_k}}$. $\mathcal{A}_{a_i,z_k}$ is defined as:

$$\mathcal{A}_{a_i,z_k} = \frac{(1 + \sum_{z_j \in \mathbb{Z}, z_j \neq z_k} B(a_i, z_k))}{\sqrt{(X_{a_i} - X_{z_k})^2 + (Y_{a_i} - Y_{z_k})^2}} \tag{2.27}$$

where $B(a_i, z_k)$ is a Bernoulli variable that is $B = 1$ if the user $z_k$ already populating or moving towards the attractor $a_i$ or 0 otherwise. The tuple ($X_{z_k}$, $Y_{z_k}$) defines the coordinates for the mobile user $z_k$ and ($X_{a_i}$, $Y_{a_i}$) the coordinates of an attractor $a_i$ .

We implemented this mobility model by dividing the simulation area into equally sized squared-cells and considering each cell to be an attraction point. The coordinates $(X_{a_i}, Y_{a_i})$ mentioned above mark the center of the $i$-th squared attraction point. Once node $z_k$ chooses its new destination, it travels towards to a randomly selected position inside the cell centered at $(X_{a_i}, Y_{a_i})$, with a velocity uniformly selected over $[v_{min}, v_{max}]$. Upon arrival to its destination, a pause time is randomly chosen over $[0, P_{max}]$. A new destination is selected at the end of the pause period, and this process repeats itself until the end of the simulation time.

In CMM, the simulation area is divided into a set of subareas used as attractors. Movement occurs similar to a RWP behavior, where speed and pause are randomly selected within a range. During a initial phase, called the “growth phase”, nodes are placed one-by-one in each subarea. The probability of each node being placed in each subarea changes at each node drop, and is proportional to the population of each subarea. As the growth phase goes on, the probability of assigning a node to each area changes, until all nodes are placed. During the mobility phase, called rewiring, nodes move from one subarea to another, following the preferential attachment principle, where the attractiveness of the area is determined by the current number of nodes assigned to that area. This is similar to Natural, but not taking into account the distance to the destination. In CMM, nodes also decide to move or stay in the same subarea according to a parameter called “mobility factor” $\epsilon$, in which nodes decide to change cells with a probability equal to $\epsilon$ and decide to stay in the same subarea with probability equal to $(1 - \epsilon)$. This is similar to our mobility degree, but they do not take into account the power law characteristics intrinsic to this decision.

### 2.6.3 Self-similar Least Action Walk

In the Self-Similar Least Action Walk (SLAW) mobility regime, fractal waypoints are generated using Brownian Motion [Lee et al. 2009]. Fundamental fractal properties are used to generate power-law flights. The degree of self-similarity (scale-independence) of waypoints can be controlled by adjusting the *Hurst* parameter [Lee et al. 2009]; a greater Hurst value represents a higher degree of self-similarity in the network. Even though the Hurst parameter is well-defined mathematically, it is highly difficult to estimate it from a given data sample. As such, a variety of Hurst parameter estimation techniques have been proposed [Berzin et al. 2014].

In SLAW, Waypoints work similarly to the attractors in the previous models, and once they are placed, clusters are formed via transitive closure, i. e., the waypoints that are connected to each other through multiple links form one cluster [Lee et al. 2009]. In the beginning of the simulation, every node chooses a set of clusters and a fraction of waypoints to visit within each of the selected clusters. Then, a trip planning algorithm called Least Action Trip Planning (LATP) is used and combined with a "walker" model restricts the mobility of each walker to a predefined sub-section of the total area. We used an available MATLAB implementation of the SLAW mobility model to generate mobility traces that follow SLAW mobility [Lee et al. 2009]. Moreover we followed the guidelines for setting up SLAW's simulation parameters found in [Lee et al. 2009]. We fixed the simulation area to be the same as the region covered in the mobility trace and the number of clusters to be the same as measured in the real trace.

### 2.6.4 Simulation Setup

We conducted two types of simulations: (1) first, we modified the Scengen [The Scenario Generator ] scenario simulator to generate traces according to RWP (already implemented), Natural and CMM (implemented at Scengen), SLAW (MATLAB implementation), and SFMR (also implemented in Scengen). Once the simulator was able to generate the mobility traces we computed the spatial node density distribution results presented in Section 2.7.1. (2) in the second type of simulation experiments, once the synthetic mobility traces were generated as described above, these and the real traces were fed to the Qualnet network simulator [Scalable Network Technologies ] in order to evaluate their impact to core network functions, such as routing and message dissemination for example.

For the first type of experiments, in order to compare synthetic traces generated with RWP, Natural, CMM, SLAW, and SFMR to real user mobility traces, we adjusted the Scengen simulation parameters according to information extracted from the real trace for all mobility models. For example, velocity range $[v_{min}, v_{max}]$ is set such that average node velocity (assuming that the velocity of each node is randomly chosen from a uniform distribution of values between $[v_{min}, v_{max}]$), matches the average node velocity extracted from the trace. In particular, for the RWP regime, in order to address the steady-state stationarity problem reported in [Yoon et al. 2003], we followed the recommendations mentioned in that work. More specifically, the velocity range was set to be ±, the standard deviation measured in the real traces, around the measured average velocity. Then, velocities were chosen *uniformly* within that range in which the lower limit was greater than zero and where the mean matches the one measured in the real trace.

Similarly, the pause time was chosen uniformly in the range $[0, P_{max}]$, where the value of $P_{max}$ is such that the average pause time matches the one measured in the real traces. The dimensions of the rectangular simulation area are set to be the same as in the traces. Moreover, in our simulation scenarios, we use the same initial positions for the nodes found in the real traces, except for SLAW which has its own initialization procedure (as described in Section 2.6.3).

In the RWP simulations using Scengen, a node's next destination $(x_d, y_d)$ is randomly chosen over the simulated area according to a uniform distribution. For SFRM, the choice of $(x_d, y_d)$ is given by Equation 2.26, where the intensity values $\mu$ are set by the initialization procedure as described in Section 2.4. For Natural and CMM, the probability of choosing the next destination is computed "on-the-fly", based on the destination's popularity as described in Section 2.6.2. SLAW follows its own initialization procedure which is detailed in Section 2.6.3

For the second type of experiments, synthetic mobility traces generated using Scengen as described above, as well as the real traces were fed to the Qualnet network simulator [Scalable Network Technologies ]. As previously pointed out, efficient message dissemination is critical to road safety and transportation efficiency in ITS. Thus, the goal of these experiments is to evaluate how close to the real trace are the synthetic mobility regimes as far as their impact on routing and data dissemination.

| **Parameter** | **Quinta** |
|---|---|
| Average Velocity ($\pm\sigma$)(m/s) | 1.2 (±0.53) |
| Average Pause Time Duration (sec) | 3.6 |
| Area Dimensions (meters x meters) | 840 x 840 |
| Duration of Simulation (sec) | 900 |
| Number of users | 97 |
| Number of CBR flows | 20 |

Table 2.6: Simulation parameters.

Data traffic scenarios used in these experiments try to simulate nodes communicating with one another in ITS scenarios (e.g., vehicle-to-vehicle, vehicle-to-infrastructure). We use 20 Constant Bit Rate (CBR) flows between randomly chosen source-destination node pairs. Flows start at randomly chosen times and stay active during the course of the whole simulation generating traffic at a rate of 4 packets per second. We use the Ad-hoc On-Demand Distance Vector (AODV) [Perkins et al. 2003] routing protocol, an Internet standard for routing in wireless multi-hop ad-hoc networks, and the IEEE 802.11g data link layer protocol with radio range of 150m and data rate of 54.0 Mbps. Table 2.6 summarizes other simulation parameters used in these experiments.

## 2.7 Results

Results are reported here for the Quinta trace with a $90\%$ confidence interval over 10 runs. For the runs using the real trace, since we cannot vary mobility, we randomize the traffic scenarios by varying the source and destination pairs of the flows in each of the 10 runs. The same traffic patterns were used to feed the RWP, Natural, CMM, SLAW and SFMR simulations, but in these cases, we generated 5 mobility traces with each model, giving a total of $10 \times 5 = 50$ simulation runs for each synthetic mobility regime.

### 2.7.1 Spatial Node Density

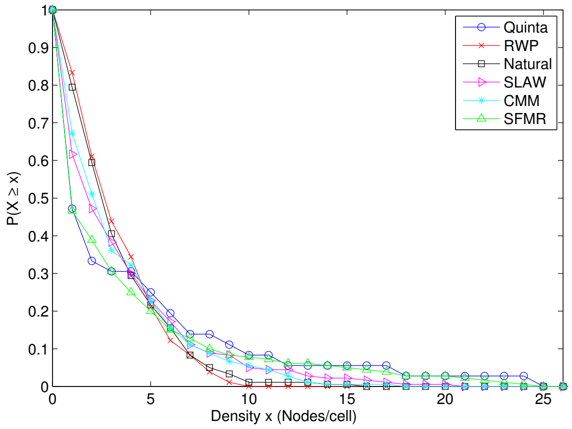


Figure 2.5: Node Spatial Density Distribution.

In order to study spatial node density behavior, we define the *Node density distribution* metric as the ratio of cells containing $\geq n$ nodes. Each curve in Figure 2.5 shows the density distribution for the Quinta trace and each mobility model, namely SFMR, RWP, Natural, CMM, and SLAW. The curves shows the distribution at the end of the trace collection interval, which is at 900 seconds for Quinta.

| Mobility Model | Mean | Confidence Interval |
|---|---|---|
| SFMR | 0.0161616 | [0.00749751 0.0248257] |
| SLAW | 0.0396465 | [0.0234087 0.0558843] |
| CMM | 0.0492424 | [0.0282684 0.0702164] |
| Natural | 0.070202 | [0.0364066 0.103997] |
| RWP | 0.813131 | [0.0442555 0.118371] |

Table 2.7: Normalized difference between the spatial distribution resulting from mobility models and the empirical distributions computed from the real trace: mean and lower and upper values of the 95% confidence interval.

From these plots we observe that SFMR's density distribution closely follows the distribution of the real trace. In the case of RWP, the majority of cells (i.e., more than 80%) present a similar number of nodes (i.e., one or more nodes), and no cells contain significantly greater concentration of nodes (i.e., no cell contains more than 9 nodes). This is also the case for Natural, CMM and SLAW. In order to quantitatively compare how close the node density distributions resulting from the synthetic mobility regimes are to the real trace, we compute the average normalized difference between the synthetic traces' spatial node density distribution and that of the real trace as follows: for each data point, we compute the absolute value of the difference between the density distribution resulting from the synthetic model and that of the real trace, divided by the latter. We average over all data points and Table 2.7 reports these averages as well as lower and upper values of their 95% confidence interval. Table 2.7 confirms that SFMR's spatial density distribution is the closest to the real trace's when compared to the other mobility regimes studied.

### 2.7.2 Performance Evaluation of SFMR

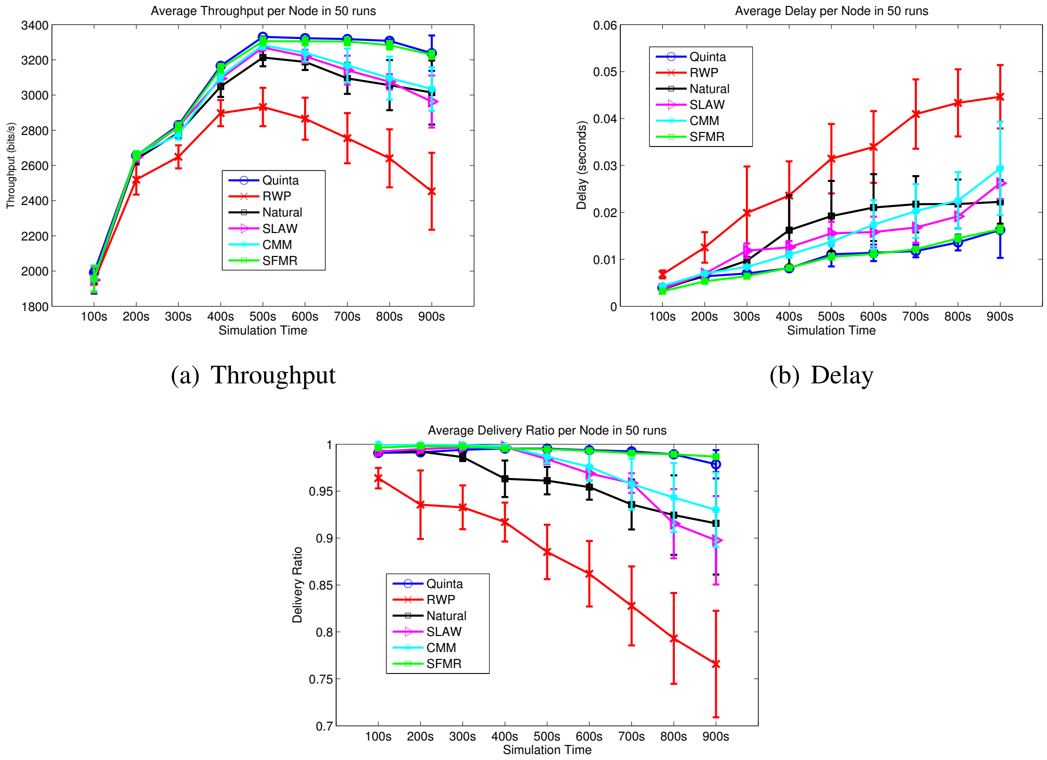
Mobility models are frequently used for simulation purposes when new communication-based vehicular and human mobile services are being investigated. One key factor researchers and developers must take into account when evaluating solutions through simulations of mobile scenarios such as V2V and V2I applications is realistic mobility patterns. In fact, mobility models play a vital role in determining the performance of various wireless mobile systems, such as Vehicular Ad-Hoc Network (VANET) [Hou et al. 2016], Wireless Sensor Network (WSN) AND Body Sensor Networks (BSNs) [Sadiq et al. 2018], etc. In ITS an efficient message dissemination scheme is critical to its applications, such as road safety and urban traffic status. Thus, in order to evaluate SFMR in such dynamic scenarios we focus on the study of the impact of different mobility models in an infrastructureless network, when compared to real mobility extracted from a real mobility trace.

We report results comparing performance for the AODV wireless ad-hoc network routing protocol under our mobility regime, the Quinta mobility trace, as well as mobility regimes proposed in the literature and discussed in Section 2.6. The objective here is not to evaluate a proper ITS system or a real application, but rather evaluate the ability of our proposed model to deliver realistic node movement and how a network simulation can be affected by realistic and non realistic mobility. We compute the following metrics in our study:

- *Throughput*: is defined as the total number of bytes received at the destination node divided by the time elapsed between the reception of the first byte of the first data packet and the reception of the last byte from the last data packet. This quantity is measured at all nodes and averaged before reported.
- *End-to-End Delay*: is measured as the time elapsed between the moment a packet is sent and the instant it is received at the destination. This quantity is then averaged for all packets transmitted by all nodes in the network.
- *Delivery Ratio*: is computed as the ratio between the total number of packets received by all nodes and the total number of packets transmitted by these nodes.

The above described metrics for throughput, delay, and delivery ratio are reported in Figures 2.6(a), 2.6(b) and 2.6(c) respectively, over time for the Quinta scenarios. There is a notable discrepancy between the results for the real trace and results for RWP. Also noteworthy is how the discrepancy widens over time which can be explained by RWP's inability to maintain the trace's spatial node density distribution over time which directly impacts routing performance. SFRM, on the other hand, allows the formation and preservation of clusters of nodes, which, in the case of this scenario, resembles closely the real trace curves. As the clusters are bigger for the realistic scenarios and SFRM, information delivery is also more efficient, as more nodes are closer together in the clusters.

In the case of Natural, CMM, and SLAW, we notice that routing performance under these mobility regimes stay close to the real trace up until around 300s for Natural and around 500s for SLAW and CMM. Up until then, the probabilities of choosing each cell are based on the initial non-uniform spatial densities, and the mobility regimes are capable of maintaining some level of node clustering. However, later in the experiment, nodes start to spread out as the probability of choosing a new cell starts approaching a uniform distribution. This behavior causes the clusters to dissipate and routing performance starts to diverge from the real traces.



(a) Throughput

(b) Delay

(c) Delivery Ratio

Figure 2.6: Network routing performance for the Quinta trace.

## 2.8 Generating ITS-Inspired Traces with SFMR

As previously pointed out, one of the distinguishing features of SFMR is its ability to generate mobility traces without the need to prime its parameters using existing traces. In this section, we demonstrate this feature of SFMR by using it to generate mobility traces for ITS-inspired scenarios.

### 2.8.1 Mobility in Urban Scenarios

Suppose we want to simulate mobility in an urban scenario, such as the downtown area of a large metropolitan region. We could then consider two different types of mobility, namely pedestrian- and vehicle mobility.

*Spatial Density* Pedestrians tend to congregate in locations like malls, markets, cafes, schools, etc. Since pedestrian density tends to be relatively high in most downtown areas (e.g., compared to rural or even American suburban areas), the mobility model used to represent spatial node density of pedestrians in urban centers could then be assigned a lower value for $\alpha$. This means that the power-law curve representing spatial node density of pedestrians in downtown areas would have a longer tail to indicate that a relatively higher percentage of cells have higher concentration of nodes.On the other hand, if we are now interested in simulating vehicle mobility in a city center, we could consider fewer nodes (e.g., in some cities, only public transportation is allowed to circulate in the city's downtown area) compared to pedestrians. Assuming that public transportation vehicles are moving most of the time, except for high traffic congestion spots or bus depots, most cells would have lower concentration of nodes. As such, we could use a power-law distribution with longer tail, i.e., a higher value of $\alpha$, to represent spatial density of vehicles in a city center.

*Mobility Degree* To model pedestrian mobility degree, we would assume that most pedestrians would typically visit less cells due to their limited mobility and thus exhibit lower mobility degree relative to vehicles. This means that pedestrian's mobility degree would follow a power law that decays quickly, i.e., with higher $\alpha$.

For vehicles, since they can cover longer distances and, as a result, visit more cells, the tail of the power law describing their mobility degree distribution would be longer when compared to pedestrians'.

### 2.8.2 Mobility in Suburban Areas

In the case of American suburbs, we could still consider different mobility regimes for pedestrians and vehicles. However, unlike urban scenarios, suburbs are typically less densely populated and there are less people walking than driving.

*Spatial Density* For pedestrians, there would likely be only a few areas with higher pedestrian density like parks and street malls, while most everywhere else would present low densities. As such, we could use a higher $\alpha$ value to simulate spatial density of pedestrian mobility in American suburban areas.

We could also envision similar behavior for the spatial density of vehicle mobility in suburban settings, i.e., that most cells will exhibit low vehicle density. As such, we could use higher $\alpha$ values to model vehicle spatial density in American suburban scenarios.

*Mobility Degree* In American suburban areas, we could envision scenarios where a reasonable number of vehicles circulate only locally but a good number travels longer distances, e.g. when people commute to work. As such, we would use lower $\alpha$ values for the mobility degree power law distribution.

In the case of pedestrians, we may consider people spending most of their time inside their property and going out to move around the streets for a few sporadic activities, e.g., jogging, walking the dog, go to the playground or store close by. For that reason, we would recommend using a higher value of $\alpha$ for simulating pedestrians in this conditions.

### 2.8.3 Sample ITS Mobility Regime

Here we use a sample ITS-inspired mobility scenario to illustrate how SFMR can be used to generate synthetic mobility traces without the need to extract parameters from existing traces. The goal is to show how to use SFMR to simulate a given ITS scenario and validate the resulting spatial density and mobility degree distributions by comparing them to the ones obtained using our analytical model SFSM derived in Section 2.4. We use our implementation of SFMR on the Scengen [The Scenario Generator ] simulator to generate SFMR mobility traces.

In particular, this example simulates $3,000$ vehicles moving around a large metropolitan region of size 8km-by-6km. Vehicle speeds vary uniformly over a range of $15$ to $40$ km/h [2]. The duration of the simulation is set to $100.000$ seconds (i.e., around 27 hours, or a little more than a day). We wanted to keep the network always mobile and for that reason we set pause time to be 0 at all times. Table 2.8 summarizes the simulation parameters and their values. We simulated two scenarios by essentially changing the value of $\alpha$. We first use an alpha of $1.4$ for both mobility degree and density, and then increase $\alpha$ to 2.4.

| **Parameter** | **Value** |
|---|---|
| Velocity Range (km/h) | [15 - 40] uniform |
| Average Pause Time Duration (sec) | 0 |
| Area Dimensions (meters x meters) | 8000 x 6300 |
| Duration of Simulation (sec) | 100000 |
| Number of nodes | 3000 |
| $\alpha$ for Mobility Degree | 1.4 and 2.4 |
| $\alpha$ for Spatial Density | 1.4 and 2.4 |

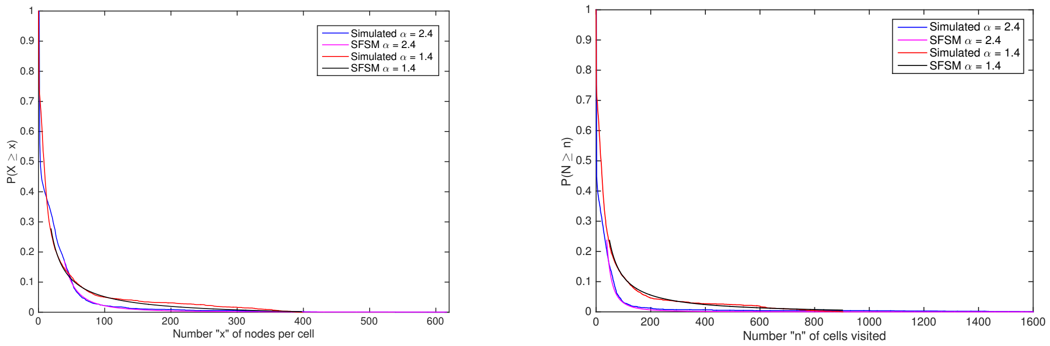
Table 2.8: SFMR parameters and their values for sample ITS mobility regimes.

The data points in the SFMR curves in Figure 2.7 are averaged over 20 simulation runs; the graphs also show the SFSM with the previously mentioned values of $\alpha$. Fig-

[2] These parameter values were set based on real scenarios as reported in "http://infinitemonkeycorps.net/projects/cityspeed/"

ure 2.7(a) shows the spatial density distribution for two values of $\alpha$. In the example scenario described in Section 2.8.2 above, most cells present low density of vehicles with a small number of cells exhibiting high vehicle densities (e.g., shopping malls, supermarkets, school campuses, etc); we would use $\alpha = 2.4$ in this case. The value of $x_{min}$ was set to 45 for SFSM with $\alpha = 2.4$. The value of $x_{min}$ was then set to 25 in the case of SFSM with $\alpha = 1.4$. We observe that both curves match closely the SFSM curves.

One of the curves in Figure 2.7(b), i.e., the one with $\alpha = 2.4$, shows an example of high mobility degree where few mobile nodes visit $> 1500$ cells. This mobility degree behavior can mimic the behavior of vehicles in a city center as described in Section 2.8.1. When $\alpha = 1.4$ the decay of the curve is slower and more nodes have lower and more uniform mobility degrees, meaning that 25% of the nodes visit from 85 ($x_{min}$) to 900 cells. This could be true if we wanted to simulate for example, vehicles moving on the suburban neighborhood scenario mentioned before in Section 2.8.2.



(a) Spatial Density.

(b) Mobility Degree.

Figure 2.7: Mobility degree and spatial density distribution for ITS-inspired mobility regime generated by SFMR and SFSM.

## 2.9 Related Work

Intelligent Transportation Systems (ITS) have been receiving considerable attention from both academia and industry, e.g., in the context of Smart Cities and Internet of Vehicles (IoV) ( [Datta et al. 2017, Botta et al. 2014]), where vehicles are considered connected resources through cloud and edge computing technology as well as wireless communication devices (e.g., smart-phones).

As previously pointed out, ITS has leveraged research and developments from the vehicular networking (VANET) community to address important transportation problems such as road safety, congestion control, travel reliability, and infotainment. Some characteristics of vehicular- and human mobility, such as contact frequency and duration, locations visited, as well as how far they travel within a given geographical region in both space and time are crucial to the design and performance of ITS services and their protocols.

Motivated by ITS and, more broadly, by Smart Cities applications and services, there has been increased interest in studying human mobility in order to develop more realistic mobility models. Since there is no commonly agreed definition of what "realistic mobility" means, existing models try to mimic real mobility using different approaches and criteria. For example, a number of models use the *preferential attachment* principle which states that "the more connected a node is, the more likely it is to receive new links". The seminal work of Barabási and Albert [Barabási and Albert 1999] proposes a model that generates scale-free networks, i.e., networks whose node degrees follow a power law distribution. Several mobility models have been inspired by the Barabási-Albert preferential attachment principle (e.g., [Lim et al. 2010, Kosta et al. 2014, Hsu et al. 2009]). The work reported in [Noulas et al. 2011], investigates node mobility patterns in a large number of cities across the world and found that the distribution of node displacements in the dataset is well approximated by a power-law with exponent $\beta = 1.50$ and a threshold $\delta r_0 = 2.87$. The Community-based Mobility Model (CMM) [Lim et al. 2006], which we use in our comparative study and describe in Section 2.6.2, is another model derived from applying Barabási's preferential attachment principle. In [Boldrini and Passarella 2010], an extension of CMM, called Home-cell Community-based Mobility Model (HCMM) takes into consideration both node- and location attraction. Under HCMM, nodes are assigned to a specific community and have social ties with members of this community. Additionally, some nodes also have social relations with nodes outside of the community. Nodes then choose an attraction region with a probability proportional to the number of ties with nodes in that region. In [Vastardis and Yang 2014], an extension to HCMM called Enhanced Community Mobility Model (ECMM) is proposed, which includes new features, such as pause periods and group mobility support.

Authors in [Nunes and Obraczka 2011] show that using the preferential attachment principle to model node mobility leads to undesirable long-term behavior. More specifically, preferential attachment based mobility regimes do not preserve the original spatial node density distribution and lead to steady-state behavior similar to random mobility as exemplified by the RWP model. Instead, real node mobility exhibits invariant density heterogeneity.

Other approaches study mobility patterns through data captured from WLAN- or cellular infrastructure [Lin and Hsu 2014]. For example, in [Song et al. 2010a], quantitative models that can account for the statistical characteristics of individual human trajectories are proposed. They show that human trajectories follow several highly reproducible scaling laws, such as the number of distinct locations visited by a randomly moving object, as well as the probability of a user to visit a given location and the resulting mean square displacement. However, user locations were recorded from GSM-tower cells which provide low spatial resolution (in the order of few hundreds of squares meters to several square kilometers). Also, paths taken between two points are often indicated by a series of discontinuous sudden jumps and thus are hardly observable with sufficient granularity [Lin and Hsu 2014]. Thus, mobility models derived using such approaches may remain biased [Hess et al. 2015b].

More recent efforts to model human mobility have tried to incorporate social structure and features into their mobility models. For example, the work in [Ribeiro et al. 2012] proposes to account for attraction between nodes when modeling pause times. The work in [Karamshuk et al. 2014] proposes a mobility framework that allows the generation of different mobility models by modifying properties such as inter-contact time. The proposed framework uses as input a social graph in order to detect and map communities. Users visit these communities over time based on a configurable stochastic process. Other examples of mobility models that account for social interactions include the work presented in [Harfouche et al. 2010] and [Rhee et al. 2011]; under their proposed mobility regimes, node movement is influenced by the strength of social ties and the choice of an attraction point is based on the history of visits of other nodes to that location. The SLAW mobility model [Lee et al. 2009, Lee et al. 2012] expresses socially-aware mobility patterns using fractal waypoints and heavy-tail flights between waypoints. According to [Lee et al. 2009], inter-contact time and pause time distributions extracted from real traces are shown to fit a truncated Pareto distribution. In [Lee et al. 2009], it is also shown that SLAW exhibits social structure present among people sharing common interests or those in a single community such as a university campus, companies, and theme parks. SMOOTH [Munjal et al. 2011] is a waypoint placement technique proposed to mimic the Hurst effect used to model self-similarity in SLAW. SMOOTH is validated against SLAW by comparing statistical features extracted from its synthetic traces against the same features from synthetic traces generated by SLAW.

Leap Graph [Dong et al. 2013] uses mobile phone data, such as user ID, time calls start and end, and phone GPS coordinates, to predict the next region where a user will be located given the user's current location. In order to predict user mobility, MobHet [Silveira et al. 2016] employs SMOOTH's techniques to assess the popularity of a particular geographic area and Leap Graph to determine the frequency at which nodes transition between these areas. The proposed mobility model was evaluated in two scenarios, one using data from a single source (mobile phone call log or GPS location from Twitter) and another combining the two types of sources. The results show that MobHet exhibit adequate accuracy when it exploits features from all available data sources. They also show that both SMOOTH using just GPS data and Leap Graph just using mobile phone data achieve comparable performance to MobHet.

A framework to characterize and classify Point of Interests (PoIs) according to their relevance to individual mobile users is proposed in  [Jahromi et al. 2016]. The accuracy of the framework, in terms of spatio-temporal regularity in visiting PoIs and also connectivity properties of human mobility, is evaluated by comparing against real traces as well as synthetic traces generated by SLAW.

## 2.10 Conclusion

In this chapter, we showed the scale-free properties of some important human mobility characteristics, namely spatial node density and mobility degree. In our study we analyzed a set of real mobility traces collected in diverse scenarios motivated by ITS, namely a city park, a University campus, and taxis in the downtown area of a major city. We demonstrated that both spatial node density and mobility degree exhibit power law behavior which then allowed us to derive analytical models for these two mobility features. We showed that the proposed analytical model closely matches the empirical data extracted from the real mobility traces. Another contribution of our work was to use the proposed analytical models for spatial node density and mobility degree to build a waypoint-based mobility regime capable of generating synthetic mobility traces whose spatial node density and mobility degree closely resembles the ones measured in real human mobility scenarios. As such, the proposed mobility regime can be employed to test and evaluate ITS services and protocols. Finally, using a network simulator, we evaluated a wireless ad-hoc network routing protocol and showed that its performance under our mobility regime and under the real trace is very similar.

Mobility models based on real mobility traces not only benefit the wireless and mobile networking design, but also have research impact in other areas, allowing more features about human behavior to be uncovered. For example, studying human mobility can help us understand the constraints of opportunistic communication and to design practical and effective forwarding strategies. As presented in this chapter, SFSM model has the ability to express analytically the behavior of users to tending to congregate and form clusters, where some regions may be quite dense while others completely deserted. Another interesting behavior found by SFSM is that there are few nodes that have very high mobility visiting many places in the trace, while the majority of users have a sedentary behavior, that is, they visit only few places. In the following chapters, we will use these findings, as it has considerable impact on fundamental network properties such as connectivity and capacity, to design a practical and effective forwarding strategy in opportunistic networks. To that end, in the next section, we are going to apply the principles of SFSM to identify communities based on the real behavior described by our proposed analytical model.

# 3. Using Real Mobility Records for User Community Identification in Smart Cities

Motivated by Smart City applications and services, this chapter presents a novel approach to identifying user communities in communication networks. The proposed approach uses clustering techniques to group users in communities based on their geographical preferences such as time spent in certain locales, and mobility-related features, namely mobility speed and time between consecutive movements. We describe our user community identification methodology in detail including how mobility features can be extracted from real mobility traces. We present results obtained when using our approach to identify user communities in three different mobility scenarios as well as an evaluation study comparing the performance of different clustering algorithms. In addition, a validation methodology that uses image-based similarity metrics is proposed, in order to assess the quality of the identified communities.

## 3.1 Introduction

According to the United Nations' Department of Economics and Social Affairs[1], it is estimated that 55% of the world's population currently lives in urban centers and will reach 68% by 2050. Consequently, modern cities face tremendous challenges including the need for efficient mass transit and transportation systems, communication infrastructure, power and water distribution systems, to name a few.

Leveraging advances in computing and communication, *Smart Cities* have emerged as a way to address such challenges. Smart Cities make use of ubiquitous information, computing, and communication infrastructure to ensure efficient and sustainable city operations and services [Alavi et al. 2018, Wikipedia contributors 2019]. A notable example

[1] https://www.un.org/development/desa/en/news/population/2018-revision-of-world-urbanization-prospects.html

of “urban analytics” [Senaratne et al. 2018, Bocconi et al. 2015], i.e., the use of information technology applied to urban planning, and in particular, Smart Cities, is the study of human mobility. For instance, information about people with similar geographical and temporal mobility behavior is critical for efficient and environmentally-aware transportation and transit planning. Additionally, information on vehicle trajectory will also be used to plan location of fueling (e.g., gas-, electric, fuel-cell) stations [Niu et al. 2016]; car-, bike-, and scooter sharing services can also take advantage of human movement patterns to optimize their deployments [Liu et al. 2017, Behrendt 2016].

Capturing patterns in human mobility is also important to understand and account for social interactions which affect a range of important services such as public health (e.g., infectious disease management), law enforcement and emergency response, social services, recreation and entertainment, etc [Zhong et al. 2014]. Furthermore, periodic and occasional contacts between people (and their computing/communication devices, e.g., smart phones, smart watches) present themselves as opportunities to exchange and forward data. *Opportunistic communication* is especially attractive in scenarios where existing communication infrastructure is heavily loaded (e.g., densely populated areas, hot spots), or its coverage is insufficient due to sparse infrastructure deployment (e.g., suburban regions) [Conti and Giordano 2014]. It also becomes critical in emergency response and disaster recovery operations as the existing communication infrastructure may become completely overloaded and/or compromised.

Prior work uses social relationships among users in a network to decide when and to whom messages should be opportunistically forwarded [Yuan et al. 2016, Alajeely et al. 2017, Li and Wu 2009, Chuah and Coman 2009]. Our work focuses on capturing social relations among users by identifying *user communities*. Some existing approaches to user community identification in communication networks use features such as radio frequency signatures (e.g., whether users are communicating using Bluetooth), encounter history, as well as contact time and duration [Yuan et al. 2016, Yang et al. 2013, Nguyen et al. 2017, Eagle and (Sandy) Pentland 2006]. Some use information obtained through online social network services or data from communication service providers, e.g., cell phone call records [Phithakkitnukoon et al. 2012].

In this chapter, we propose a user community identification approach based on user mobility characteristics, including time spent in a given locale, average time between movements, or pause time, and average mobility speed. Such features are usually available from user mobility records such as GPS traces and Wi-Fi access point association traces. The proposed methodology uses clustering techniques to identify user communities based on common mobility characteristics extracted from real mobility traces. We investigate different clustering algorithms, each representing four main categories of cluster classifiers proposed in the literature [Jain et al. 1999, Cebeci and Yildiz 2015, Hasnat et al. 2015], namely: *Exclusive-*, *Overlapping-*, *Hierarchical-*, and *Probabilistic* Clustering. Additionally, we use Principal Component Analysis (PCA) and index metrics, as well as spatio-temporal information from real mobility traces to evaluate the performance of the different clustering techniques.

It is important to mention that, despite the fact that the methodology does not require such data, this data can be added to the feature matrix, if it is available.

### 3.2 Clustering for user community identification

As previously discussed, identifying communities or clusters of users has applications in a range of Smart City services, including opportunistic networking. Human social networks are known to have strong clustering characteristics [Newman 2004]. For example, people spend more time with family, friends and co-workers than with strangers. However, it is often quite challenging to find and/or access labeled records of such relationships in order to use them to identify user communities. Alternatively, human mobility also exhibits intrinsic patterns that are guided by habit, social links, and geographical preference [Hossmann et al. 2011], and arguably human mobility attributes such as geographic location, speed, direction of movement, and pause time may be easier to record and access. As such, our premise in this work is to use such mobility features that can be extracted from more widely available traces of real user mobility to infer social structure. User communities can then be identified by clustering users with similar mobility features.

Clustering is a well-known unsupervised learning method that can be used to find structure in a collection of unlabelled data, organizing data items into groups with similar features. Choosing an adequate clustering algorithm for a particular dataset depends on factors such as dataset size, structure, as well as specific goals and constraints (e.g., energy efficiency, geographic proximity, etc) [Cebeci and Yildiz 2015, Bora and Gupta 2014]. Clustering algorithms can be generally grouped into 4 categories: *exclusive-*, *overlapping-*, *hierarchical-*, and *probabilistic* clustering [Jain et al. 1999, Duda et al. 2001, Cebeci and Yildiz 2015].

- In *exclusive* clustering, data is grouped in an exclusive way, in other words, a data item can belong to only one cluster. The K-means algorithm is a well-known representative of exclusive clustering.

- *Overlapping* clustering uses fuzzy sets to cluster data, which means that, unlike exclusive clustering, a item data can belong to two or more clusters with different degrees of membership. The appropriate membership value will be associated to the data; fuzzy c-means is an example of an overlapping clustering algorithm.
- At the start of *hierarchical* clustering execution, each data observation is considered to belong to its own cluster. Then, at each iteration of the algorithm, the two nearest clusters are merged. When the specified number of clusters is reached, the algorithm terminates. We use the hierarchical clustering algorithm to represent this category of clustering algorithms.
- *Probabilistic* clustering, as the name conveys, uses a completely probabilistic approach to cluster data. Gaussian mixture clustering is well-known probabilistic clustering algorithm.

In this chapter we use four of the most widely used clustering algorithms representative of the categories described above, namely: k-means, fuzzy c-means, hierarchical and gaussian mixture clustering. We discuss each one of these algorithms in more detail below.

### 3.2.1 Hierarchical clustering

Hierarchical clustering algorithms are one of the most popular clustering techniques. It recursively find groups following two strategies: (1) agglomerative - a bottom-up approach where each data point starts in its own cluster and the most similar pair of clusters are merged successively to form a cluster hierarchy; (2) divisive - top-down approach where all the data points start in one cluster and recursively each cluster is divided into smaller clusters.

One advantage of Hierarchical clustering is that it does not require the number $k$ of clusters to be specified a priori. This is the case in user community identification using unlabeled mobility traces. We discuss how to select the value for $k$ in Section 3.3. On the other hand, the disadvantage is that the most common hierarchical clustering algorithms have a complexity that is at least quadratic in the number of data points compared to the linear complexity of $K$-means and MBC [JAI 2010].

Algorithm 3 shows the Agglomerative hierarchical clustering algorithm, where a similarity metric $x_{i,j}$ is used to evaluate the distance between attributes of nodes $i$ and $j$.

Example of node attributes extracted from mobility traces include location, speed, pause time, etc.

**Algorithm 3** Hierarchical clustering

```
Assign each item $x_1, ..., x_n$ to $K$ clusters, where $K = n$
Given a distance function $x_{i,j}$ for items $i$ and $j$
while number of clusters $! =$ single cluster of size $n$ do
    Find the closest pair of clusters (min($x_{i,j}$)) and merge them into a single cluster
    Compute similarities between the new cluster and each of the old clusters
end
```

*Step 5* of the algorithm can be done in different ways. The method we use in this work is based on the *Ward* approach [Murtagh and Legendre 2014], where, at each step, the criterion of choice for the next merge is based on an optimal value of the objective function. This function can be any function that reflects the desired similarity between resulting clusters. We used the minimum variance criterion, which minimizes the total variance within the cluster. That is, at each step, the pair of groups that leads to the lowest increase of the total variance of the merged cluster is found. This increase must be proportional to the square of the Euclidean distance between the cluster centres (centroids), given by

$$x_{i,j} = \sqrt{\sum_{l \neq i,j} (A_{il} - A_{i,j})^2} \tag{3.1}$$

where $A_{i,j}$ is the adjacency matrix element for vertices $i$ and $j$. Note that Euclidean distance is actually a measure of dissimilarity between nodes, being zero for vertex pairs that are structurally equivalent and larger for vertex pairs that do not share similarities.

Also note that the *Ward* method tends to combine clusters that have a small number of observations, in addition to allocating clusters that are the same in size and spherical [Chris Fraley 2002].

### 3.2.2 K-means Clustering

K-means clustering falls under the exclusive clustering category which allows no overlap between clusters. Similarly to other cluster techniques, K-means clustering partitions data samples into a pre-defined number $k$ of clusters so as to minimize the sum of the square of the distances between samples in the cluster. The algorithm for k-means clustering operates as follows: (1) $k$ centroids are selected amongst the data samples; the initial choice of centroids can be made randomly, or can be pre-specified; (2) the Euclidean distances between each data samples and each cluster centroid are calculated, and the node is assigned to the nearest centroid's cluster; (3) the centroid is recalculated based on the cluster's new membership. The steps of the algorithm are repeated until there are no further cluster membership changes or until there are no further changes in the centroids.

K-means algorithm requires that some parameters be specified during initialization, such as: the number $k$ of clusters, centroids' initial positions, and distance metric. The most critical choice is the number of clusters. We discuss a few ways for choosing $k$ in Section 3.3. The cluster initialization is generally accomplished by running the algorithm with different initial partition and choosing the partition that gives the smallest square error. Finally, in terms of distance metric, the one most commonly used to compute the distance between data points and centroids is the Euclidean metric.

K-means is one of the most widely used algorithms for clustering due its easy implementation, simplicity, efficiency, and empirical success [JAI 2010]. The complexity of the algorithm is linear and most of it comes from the time spent in computing vector distances. This means that k-means is more time efficient than the hierarchical algorithms.

### 3.2.3 Model-Based Clustering

Model-Based Clustering (MBC) is a representative of a probabilistic model approach for data clustering that models the density function by a probabilistic mixture model. This method assumes that the data is generated by a mixture distribution and the clusters are defined by one or more mixture components [Dasgupta and Raftery 1995]. Each cluster, can be model by a normal or Gaussian distribution that has three parameters: mean vector, covariance matrix and an associated probability in the mixture, where each point has a probability of belonging to each cluster. The Expectation-Maximization (EM) algorithm, initialized by hierarchical model-based clustering, is often used for estimating the parameters of the model, where clusters are centered at the mean value, and the geometric features (shape, volume, and orientation) are given by the covariance matrix.

The MBC consists of three main steps: (1) During initialization, it is necessary to specify the number of clusters and randomly initialize the distribution parameters for each group. The agglomerative hierarchical clustering is used to obtain the initial partitions of the data. (2) Then, compute the probability that each data point belongs to a particular cluster. (3) Application of the EM (Expectation-Maximization) algorithm, which is based on a maximum likelihood estimate, used to estimate the likelihood of the mixture parameters. (3) Finally, once the covariance matrix of the components lead to different models, the BIC technique (Bayesian Information Criterion) is used to choose the best model.

Model-Based Clustering (MBC) has linear complexity and attempts to deal with a more arbitrary shaped clusters. Due to the standard deviation parameter, the clusters can take on any ellipse shape, rather than being restricted to circles. This solve problems found in hierarchical and k-means algorithms, which tend to produce spherical and same size groups.

### 3.2.4 Fuzzy C-Means Clustering

Fuzzy clustering generalizes partition clustering algorithms (such as k-means and hierarchical) by allowing data to belong to two or more groups. In k-means and, hierarchical clustering, each data is a member of only one cluster. In Fuzzy clustering, each cluster is associated with a membership function that expresses the degree (i.e., the probability that a data is classified into a cluster) to which the data belong to a cluster. The algorithm performs clustering by iteratively searching for a set of fuzzy clusters and the associated cluster centers that best represent the structure of the data.

The fuzzy c-means algorithm works as follows: (1) Initialize the membership function matrix ($u_{ij}$); (2) At K-step, calculate the centers vectors $c_j$ with $u_{ij}$, (3) Update $u_{ij}^k$, $u_{ij}^{k+1}$ and (4) If $||u_{ij}^{k+1} - u_{ij}^k||$ is less then a certain termination criterion, between $0$ and $1$, then stop, otherwise return to (2).

Where, the membership $u_{ij}$, i.e, the degree to which data point $x_i$ belongs to cluster $c_j$, is given by,

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left( \frac{||x_i - c_j||}{||x_i - c_k||} \right)^{\frac{2}{m-1}}} \tag{3.2}$$

where $m$ is the hyper- parameter that determine the level of cluster fuzziness. In other words, the higher $m$ is, the fuzzier the cluster will be in the end. The cluster center $c_J$ is

$$c_j = \frac{\sum_{i=1}^{N} u_{ij}^m . x_i}{\sum_{i=1}^{N} u_{ij}^m} \tag{3.3}$$

The fuzzy c-means and k-means clustering are very similar once they attempt to minimize the same objective function. They differ from the addition of the membership $u_{ij}$ and the parameter $m$. When $m = 1$, the values of memberships assume $0$ or $1$, and the fuzzy cluster is reduced to the k-means algorithm.

Fuzzy c-means clustering has a complexity that is close to K-means clustering, but it still requires more computation time due to the fuzzy measurement calculations [Bridges et al. 2000]. The Fuzzy clustering has the advantage that it does not force every object into a specific cluster. On the other hand, it has much more information to be interpreted.

### 3.2.5 Discussion

The clustering classification and clustering algorithms described in this section suggest that there is no one-size-fits-all solution, i.e., no one clustering algorithm is universally applicable [Jain et al. 1999]. The efficiency of the algorithm is greatly dependent on the application and its available datasets. Additionally, information about cluster attributes such as size and scope is often not available. In the next section, we conduct an evaluation study comparing the performance of the clustering algorithms described above applied to user community identification based on features extracted from GPS and WiFi mobility traces. This comparative performance study is carried out as part of the user community identification methodology we propose and describe in detail below.

## 3.3 User Community Identification Methodology

As illustrated in Figure 3.1, our proposed user community identification methodology consists of four steps, namely:

(1) pre-processing the datasets, (2) determining the number of user communities, (3) clustering users into communities, and (4) validation and visualization. Each of these steps are described in detail below.

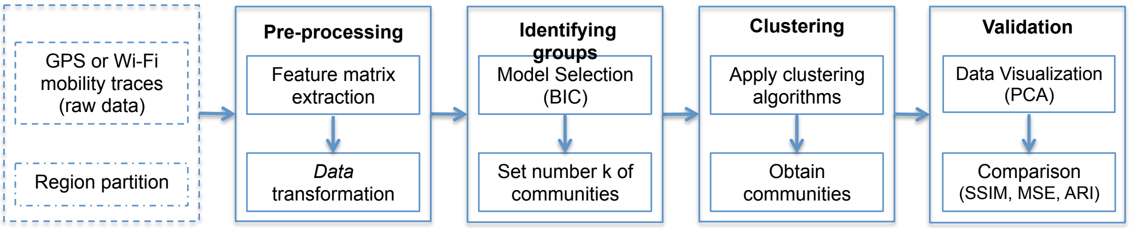


Figure 3.1: Method for community discovery in real mobility traces

- **Step 1 - Mobility Trace Pre-processing and Feature Extraction**

  The datasets used in this study are described in more detail in Section 3.4. The dataset pre-processing step consists of extracting desired features from raw mobility traces and construct a *feature matrix*.

In some of the mobility traces, user trajectories are recorded in latitude-longitude geographical coordinates. For these traces, geolocation coordinates were converted to two-dimensional UTM Cartesian coordinates [Langley 1998]. In the case of Wi-Fi datasets the location of a user was set to the location of the access point to which the user was associated at the time.

**Cell Division:** The geographical region containing all the users as they move around is divided into *cells* of size 300-by-300 meters. To validate our choice of cell size, we ran experiments varying the cell dimension a few tens of meters up and down. We observe no significant impact on the results when considering cell dimensions that are not too small or too big. It is worth mentioning that decreasing the cell size will result in increasing the number of rows in the feature matrix as explained below.

**Feature Matrix:** User spatio-temporal features are organized using a feature matrix $SM$ consisting of $N$ rows, where $N$ is the number of users, and $C+2$, where $C$ is the number of cells. $SM(i,j)$ contains the average time user $i$ spent in cell $j$. Two additional columns log the average speed of each mobile node and the user's average pause time.

**Pause Time:** The pause time is typically defined as the time between two consecutive movements by the same user. For our study, we consider that a user is not moving if the user does not move more than 1 meter. As such, if a node's displacement is less that 1 meter between two consecutive records, the time interval between these two records is added to the node's pause time. If the following record meets the same displacement limit criteria, then the time between those subsequent records are also counted towards the node's pause time, and so on. A pause interval ends when this threshold criterion is no longer valid. A node's average pause time is then calculated as the average over all pause times for that node. Note that since the WiFi dataset does not have enough granularity to allow us to calculate the user's average pause time and speed accurately, the feature matrix for the WiFi trace only has C columns containing the average time users spend in the different cells.

**Normalization:** The feature matrix must be normalized to avoid that higher values of a given attribute skew the cluster analysis. To this end, we normalize the average time a node spends in a cell by the total time the node appears in the trace and represent it as a percentage. Pause time and speed are normalized by the maximum pause time and speed encountered in the trace.

**Logit Transformation:** We apply the Logarithmic Likelihood *L*ogit function [Jaeger 2008] to the normalized feature matrix. The Logit function performs a nonlinear transformation that converts the values of the feature matrix' attributes which are between $(0,1)$ to values between $(-\infty,\infty)$ that are symmetric at $0.5$. Besides highlighting the differences and similarities between observations for each variable, this transformation also improves similarity detection by the clustering algorithms.

The *Logit()* function is defined as a logarithm of relative probabilities. If $p$ is the probability of an event, then $(1-p)$ is the probability of not observing the event, and the relative probabilities of the event are $p/(1-p)$. Hence, the *Logit* of $p$ between $0$ and $1$ is given by

$$\mathrm{Logit}(p) = \log\left(\frac{p}{1-p}\right) \tag{3.4}$$

Note that *L*ogit is not defined when $p=0$ or $p=1$. One solution to this problem is to add some small value, $\epsilon$ to the numerator and to the denominator of the *L*ogit function.

- **Step 2 - Determining the Number of User Communities**

  Some clustering algorithms rely on specifying a priori the number of clusters. We use the Bayesian information criterion (BIC) [Hasnat et al. 2015] estimator to determine the number of clusters $k$. BIC is a well-known technique that has been extensively used to find the most appropriate number of parameters for a model.

  Each combination of the number of clusters corresponds to different statistical models, reducing the problem of finding the best number of clusters to comparisons between a set of possible clusters. Therefore, if several models $M_1,...,M_k$ are considered, with a priori probabilities $p(M_k), \forall k = \{1,...,K\}$, then according to the Bayes theorem, the posteriori probabilities of the model $M_k$, given by data $D$, can be obtained by [Chris Fraley 2002]:

$$p(M_k|D) \propto p(D|M_k)p(M_k). \tag{3.5}$$

  According to the Bayesian model selection form, if the $p(M_k)$ probabilities are equal, then the choice of the model is given by the maximum likelihood between the models. This likelihood can be approximated by the BIC, such that:

$$BIC_k = 2\log p(D|\theta_k, M_k) - K\log(n) \propto 2\log p(D|M_k) \tag{3.6}$$

  where $K$ is the number of independent parameters that must be estimated (e.g., the number of clusters) for the $M_k$ model and $\theta_k$ is the parameter vector that maximizes the likelihood function for the $M_k$ model.

After the BIC calculation for several group sizes, one must decide the first local maximum, which shows the number of clusters $k$ that is more appropriate to the model.

- **Step 3 - Clustering Users into Communities** Once the number of clusters $k$ is determined, the next step is to apply the different clustering techniques described in Section 3.2, namely

  *Hierarchical Clustering*, *K-means*, *Model-Based Cluster* (MBC) and *Fuzzy C-means* clustering, in order to identify similarities amongst users and group them into communities accordingly.

  Once user communities are identified using the different clustering algorithms, the next step is to evaluate how similar users within the same community are according to their mobility features, i.e., geographic preference, average speed and average pause time, and, considering the same features, how different users across communities are.

- **Step 4 - Validation and Visualization**

  Recall that our goal is to identify user communities based on the users' mobility characteristics. As such, there is no user community organization that can be considered "ground truth" and that can be used as baseline to evaluate how well the different clustering approaches can identify user communities.

  Therefore, we used Principal Component Analysis (PCA) and also some metrics [Chakraborty et al. 2017] to identify the similarities between users to evaluate the algorithms.

  PCA is an statistical method for extracting relevant information from usually highly dimensional data. The goal of PCA is to reduce the data set dimensionality that consist of a large number of interrelated variables, while retaining as much as possible of the variance present in the data set. In order to achieve that, the data set is transformed in a new set of variables, named principal components (PCs), which are uncorrelated. The first few PCs retain most of the variance present in all of the original variables. Thus, if a good representation of the data exists in a small number of dimensions then PCA will find it, i.e. if we plot the values for each observation of the first two PCs, we get the best possible two-dimensional plot of the data. It will give a straightforward visual representation of what the data look like.

  PCA helps to find structure among objects which could not be visualized otherwise. Therefore, we use PCA in order to verify whether the clusters were able to distinguish individuals with similar mobility characteristics and geographical preferences in the same group, and distinct ones in different groups. Closeness in the score plot indicates similar “behavior” between samples.

Therefore, by applying PCA to our data we are able to (1) explore the high dimensional data and identify patterns on them; (2) visualize the data when their dimensionality is reduced to $\mathbb{R}^3$ or even at the $\mathbb{R}^2$; (3) analyze the data through the use of statistical tools such as probability density, clustering or classification.

In addition, to quantitatively evaluating the quality of clusters defined by the clustering algorithms, we proposed a validation method using the users positions over time to generate images of their trajectory along the mobility trace. Then, we compared the image pairs of users belonging to the same cluster and to different clusters using three image comparison metrics, namely: Mean Square Error (MSE), Structural SIMilarity Index (SSIM) and Adjusted Rand Index (ARI). More details on the evaluation method are presented in Section 3.4.3.

## 3.4 Experimental Results

We evaluate the proposed methodology and the clustering algorithms in two steps: first, we use PCA to reduce the number of data dimensions and assist in community structure visualization. Second, we use three metrics to compare the performance of the different clustering algorithms studied.

### 3.4.1 Experimental Datasets

The datasets used in this study represent different mobility scenarios, namely Wi-Fi association traces from the Dartmouth college campus’ WLAN [Kotz et al. 2009], the San Francisco cabs [Piorkowski et al. 2009] GPS trace, and the GeoLife [Zheng et al. 2010] dataset. The Dartmouth Wi-Fi trace logs user access to Dartmouth College’s campus WLAN using Access Point (AP) association and disassociation events. It has 6,524 users over 60 days.

The GeoLife trace captures various modes of transportation in the city of Beijing, China, (e.g. walking, cycling and driving). It includes trajectories of 182 users collected over a period of three years and sampled every 5 seconds. User trajectories are represented in the dataset by a sequence of latitude and longitude coordinates over time. The dataset contains 17,621 trajectories with a total distance of 1,292,951 kilometers and total duration of over 50,000 hours. The San Francisco cab trace consists of GPS trajectories of

483 cabs in the City of San Francisco, USA. It was collected during 24 days with samples ranging from 1 to 3 minutes.

The community structure extracted from the mobility traces was obtained by applying the community identification methodology presented in Section 3.3 which identified communities of different sizes by applying the four clustering algorithms described in Section 3.2. Table 3.1 shows, for each clustering algorithm, the number of users in each community for all traces, i.e., GeoLife (GL), San Francisco (SF), and Dartmouth (DT).

In the table, communities are shown ordered by size so that it is possible to compare the results obtained through the different cluster algorithms. We can observe from Table 3.1 and Figure 3.2 that the groups identified by the four algorithms studied for the Geolife and SF traces are similar in size and shape. This is not true for the Dartmouth trace, as Table 3.1 shows a much bigger variation in the size of the clusters. This may be caused by the fact that the Dartmouth trace has only access points association / disassociation information, i.e. it does not have the whole trajectories of nodes that the other traces do. Also, Dartmouth does not have the spacial restrictions that are present in Geolife and SF traces. These last two traces are restricted by roads and pathways while, by the nature of the Dartmouth trace, any node can transition from one location to the other without passing through other locations. It should be noted that the Fuzzy algorithm, in this scenario, was not able to find similarities to divide the groups into 9 communities, as indicated by the methodology through the BIC. Finally, the quality of the clusters presented in Table 3.1 is discussed in Sections 3.4.3 and 3.4.2.

| Trace | Community | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|---|
| GL | K-means | 32 | 31 | 28 | 27 | 21 | 20 | 10 | 8 | – |
| GL | HC | 37 | 33 | 28 | 24 | 20 | 17 | 10 | 8 | – |
| GL | MBC | 33 | 31 | 28 | 22 | 19 | 19 | 17 | 8 | – |
| GL | Fuzzy | 43 | 36 | 34 | 18 | 18 | 11 | 9 | 8 | – |
| SF | K-means | 82 | 54 | 50 | 44 | 43 | 41 | 37 | 37 | 37 |
| SF | HC | 85 | 71 | 58 | 47 | 39 | 36 | 34 | 28 | 27 |
| SF | MBC | 85 | 54 | 53 | 46 | 42 | 41 | 39 | 36 | 29 |
| SF | Fuzzy | 82 | 64 | 59 | 46 | 41 | 37 | 37 | 31 | 28 |
| DT | K-means | 334 | 283 | 267 | 265 | 229 | 219 | 229 | 167 | 73 |
| DT | HC | 403 | 289 | 258 | 245 | 244 | 155 | 150 | 148 | 112 |
| DT | MBC | 537 | 331 | 313 | 304 | 148 | 138 | 111 | 66 | 56 |
| DT | Fuzzy | 1060 | 886 | 51 | 4 | 3 | – | – | – | – |

Table 3.1: Community size for each clustering algorithm applied to the Geolife (GL), San Francisco (SF), and Dartmouth (DT) mobility traces.

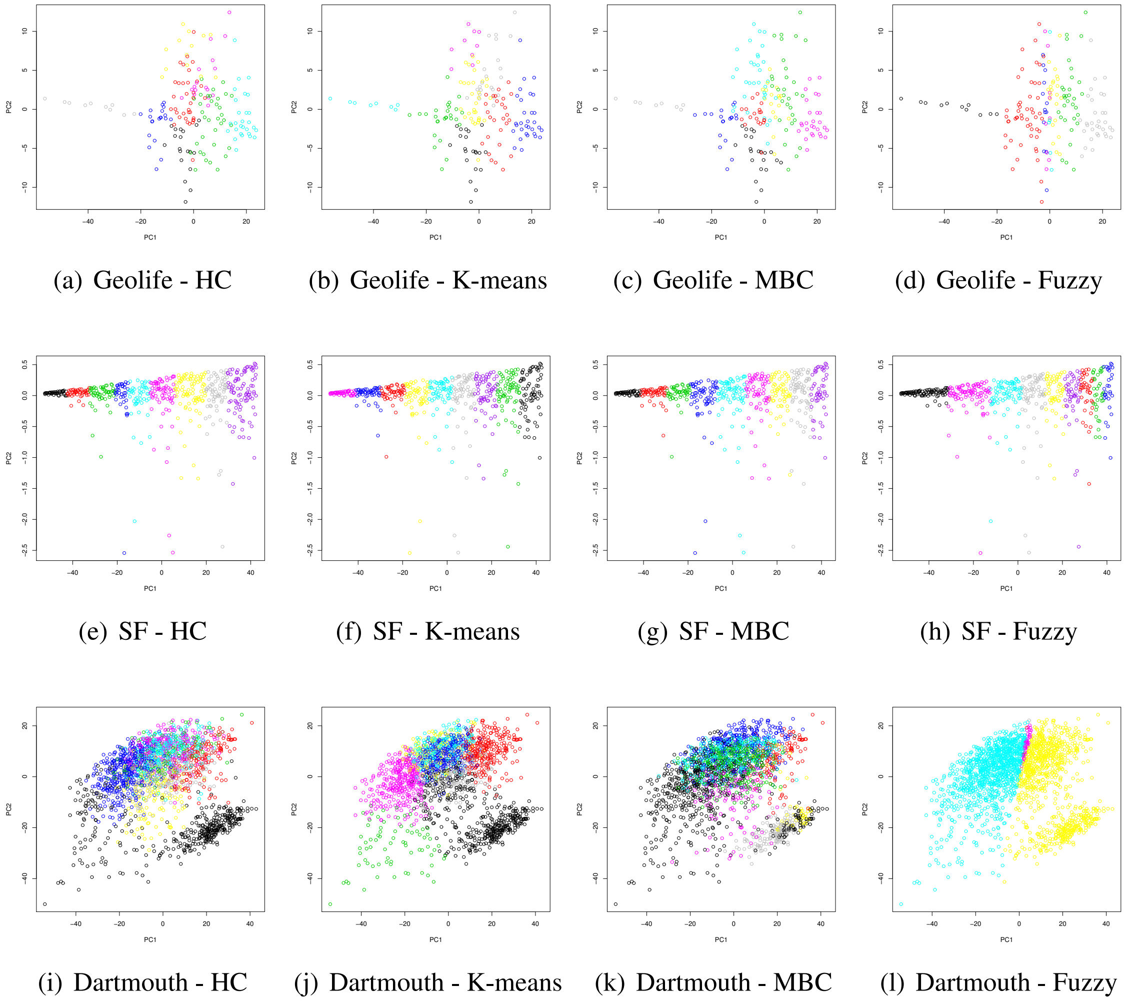
### 3.4.2 Data Visualization

In order to verify that the clustering algorithms were able to capture the geographical preferences and mobility characteristics of the users in different communities, a study of the characteristics presented in each community was conducted. Initially, users' geographical preferences were calculated using PCA. The 126 attributes (GeoLife), 734 attributes (San Francisco), and 138 attributes (Dartmouth) relative to the total time each user remains in each cell have been reduced to 2 principal components, which together represent around 80% of the data variance. The first principal component accounts for around 60% of this variance, thus offering a good representation of the users' geographical preferences.

Figure 3.2 shows the cluster visualization for GeoLife, San Francisco and Dartmouth traces, where each dot displayed in the plots correspond to an observation (i.e., a user). The visualization method presented considers the two principal components, called PC1 and PC2, to display the preference of the user of each community for a given geographical area. These plots show the first principal component (PC1) on the y-axis and the second (PC2) on the x-axis. It is possible to observe in these figures that through only two main components, it is already possible to visualize the communities quite distinctly and identify intervals in PC1 and PC2 where each community resides.

Figure 3.2 shows that nodes are separated into groups, but it is not possible to visualize the nodes attributes that are contributing to the cluster formation. Thus, in order to understand the similarities between the nodes attributes that constitute each cluster, we have reduced the dimensionality of the attribute matrix through PCA, to capture the nodes geographic preference of the different communities. Thus, Figure 3.3 presents the empirical probability density function of the PC1 of the geographical preferences for the four clustering algorithms, in all mobility traces studied. It is possible to observe that the method of community identification was able to differentiate the user's preferences of different communities by certain geographic regions. For example, in Figure 3.3(b), K-means is able to differentiate five different geographic regions of interest from communities 4, 5, 6, 7, 8. Communities 1, 2 and 3, however, were not differentiated only with PC1, since their density curves are very overlapping. On the other hand, considering the second principal component - PC2, Figure 3.4(b), K-means is able to differentiate the geographical preferences of communities 1, 2 and 3.

Figure 3.5 shows the density of the user average speed for MBC, K-means, Fuzzy, and HC clustering algorithms for GeoLife and San Francisco traces. It is possible to notice by the figure that the average speed of the users belonging to the the same cluster are



(a) Geolife - HC (b) Geolife - K-means (c) Geolife - MBC (d) Geolife - Fuzzy

(e) SF - HC (f) SF - K-means (g) SF - MBC (h) SF - Fuzzy

(i) Dartmouth - HC (j) Dartmouth - K-means (k) Dartmouth - MBC (l) Dartmouth - Fuzzy

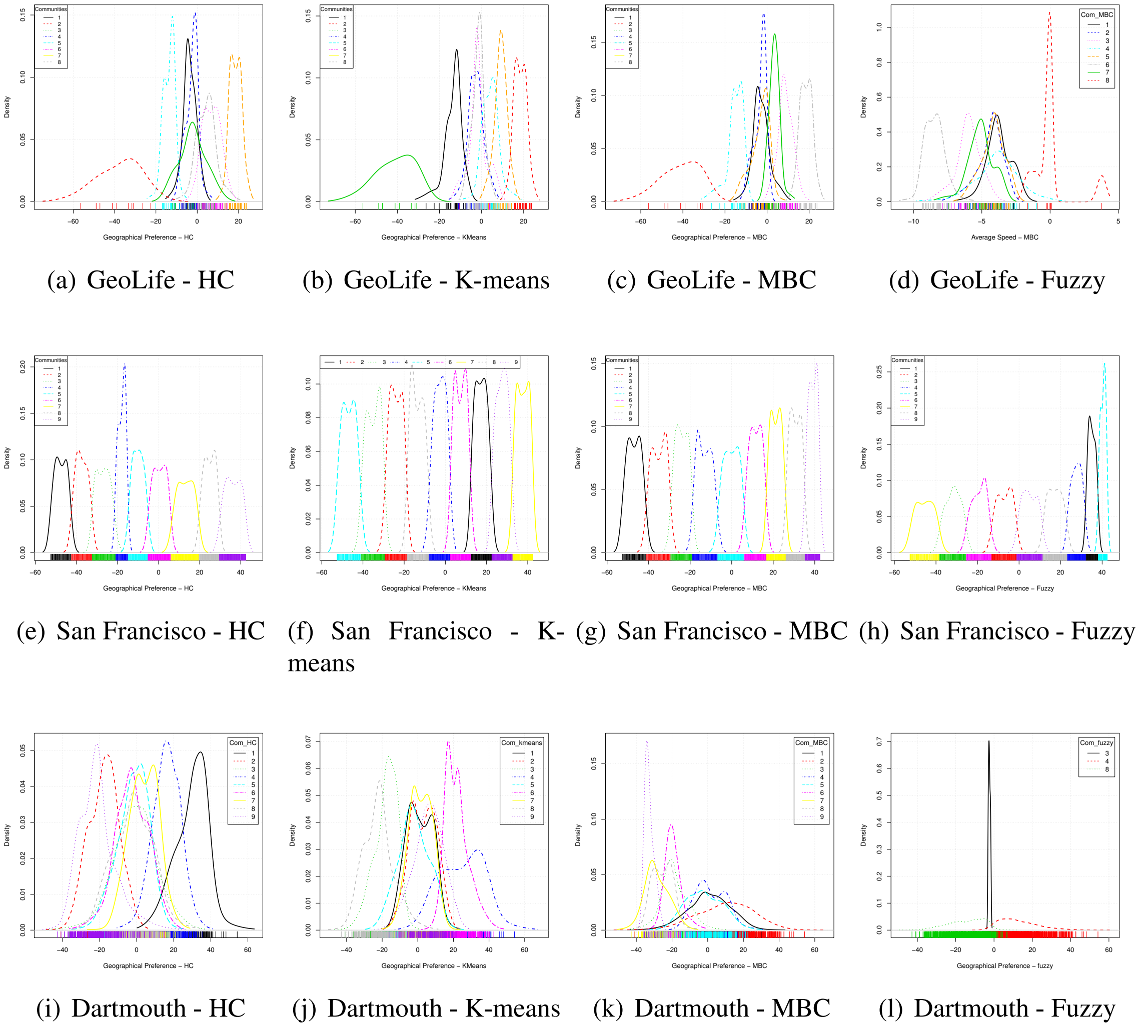
Figure 3.2: Different analyses of the clustering algorithms studied for Geolife, San Francisco cabs and Dartmouth mobility traces.

similar to each other, and dissimilar intra-clusters. Thus, the methodology was able to differentiate the different characteristics of the user mobility.

Figure 3.6 shows an example of clustering using the cluster algorithms without applying the *logit* transformation. We can observe that none of the cluster algorithms were able to extract the similarities and dissimilarities of the mobility traces. Thus, this step of the methodology is fundamental for grouping users in different communities successfully.

### 3.4.3 Evaluation

In this sub-section we complement the results presented in Section 3.4.2 to answer the following questions: Was the proposed methodology able to increase the average similarity metric between nodes that belong to the same group and also the average dissimilarity between nodes belong to different groups? What is the impact of different cluster algo-



(a) GeoLife - HC (b) GeoLife - K-means (c) GeoLife - MBC (d) GeoLife - Fuzzy

(e) San Francisco - HC (f) San Francisco - K-means (g) San Francisco - MBC (h) San Francisco - Fuzzy

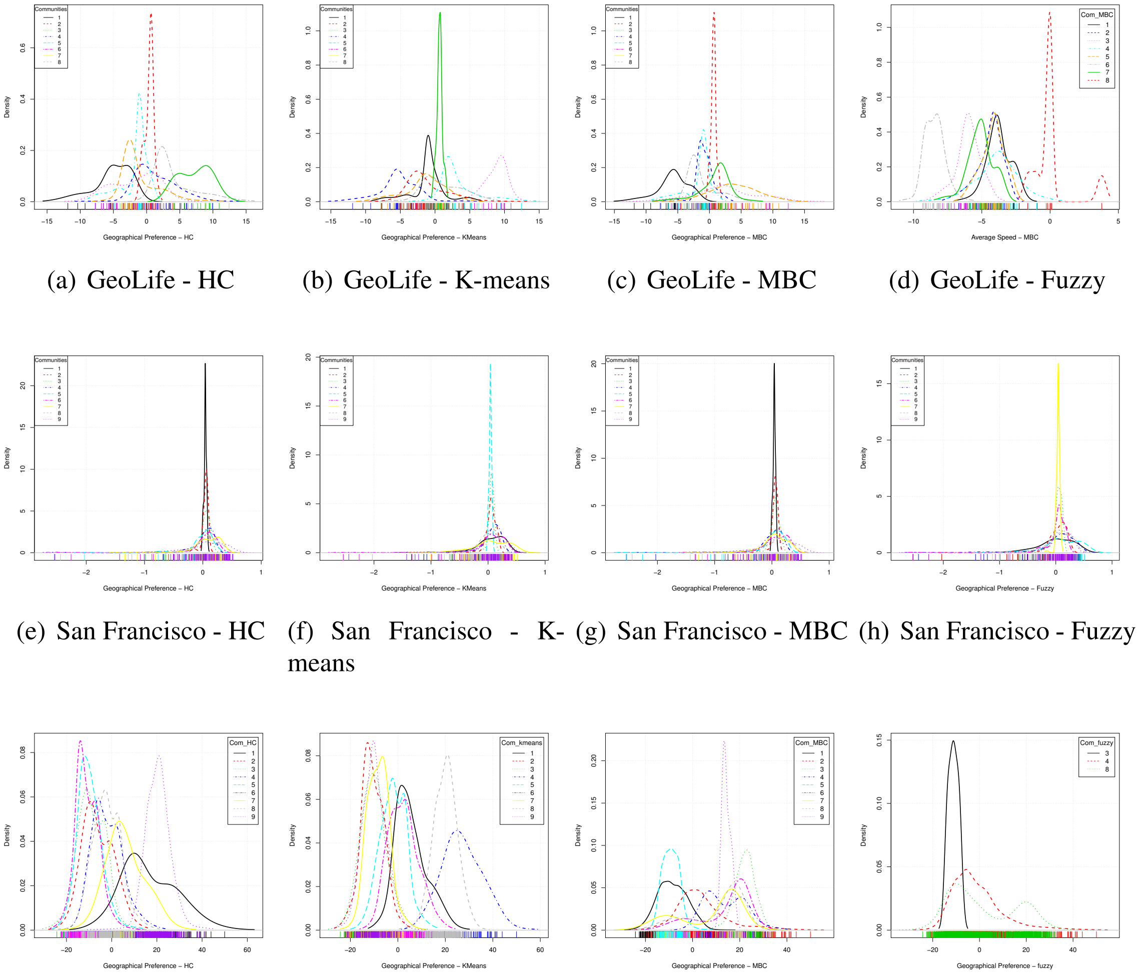
(i) Dartmouth - HC (j) Dartmouth - K-means (k) Dartmouth - MBC (l) Dartmouth - Fuzzy

Figure 3.3: Probability density function of geographical preferences (PC1) of the reduced data using PCA for all clustering algorithms, for Geolife, San Francisco cabs and Dartmouth mobility traces.

rithms in the capacity of the methodology to identify communities from mobility data?

We introduce tree metrics used to evaluate how well the algorithms split the input data into different clusters by looking at the similarities in geographical preferences between elements of each cluster. In order to do that, we propose to use the data on the geographical positions and generate images for each node. This would generate something similar to a heat map of the geographical preferences of that node. Once we have images for each node, we can compare pairs of images, towards determining how similar they are.

An image for each node is generated as follows: First we rearrange into a matrix, the features used in the previous section for classifying each node. If we then look at these matrices as images, where each position of the matrix is a pixel, the values of each position would be the intensity of that pixel. Then we are able to compare the similarity



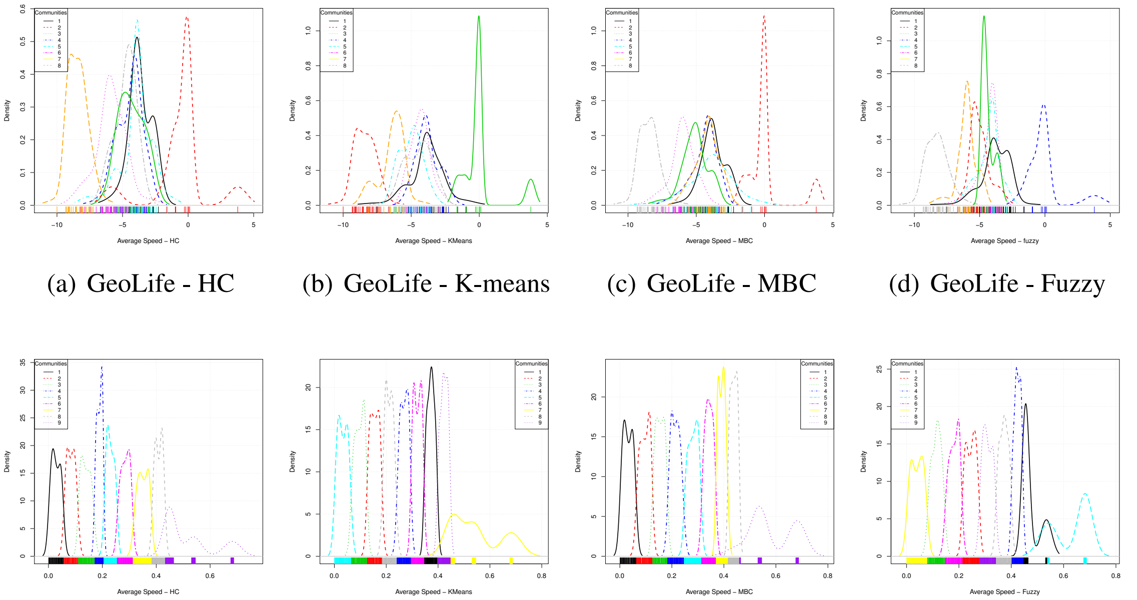
(a) GeoLife - HC (b) GeoLife - K-means (c) GeoLife - MBC (d) GeoLife - Fuzzy

(e) San Francisco - HC (f) San Francisco - K-means (g) San Francisco - MBC (h) San Francisco - Fuzzy

(i) Dartmouth - HC (j) Dartmouth - K-means (k) Dartmouth - MBC (l) Dartmouth - Fuzzy

Figure 3.4: Probability density function of geographical preferences (PC2) of the reduced data using PCA for all clustering algorithms, for Geolife, San Francisco cabs and Dartmouth mobility traces.

in the behaviour or geographical preferences of two nodes, by comparing the similarities between the two images (i.e. trajectories of node pairs and intensity of the pixel, that would be related to the time spent at that position). Figure 3.7 shows an example of three different images generated by the feature matrix of three different nodes. We can observe by visual inspection that Figures 3.7 (b) and (c) show greater similarity to each other. In other words, the two users tend to visit similar locations (pixels) and spend more time in the same places (yellow regions). On the other hand, users (a) and (b) show greater dissimilarity, since they prefer to visit and spend time in different places, i.e. user (a) spends most of the time in the top region (left-center), while user (b) prefers the middle center. Such visual inspection can be confirmed by the metrics shown in the Figure 3.7 and explained below.

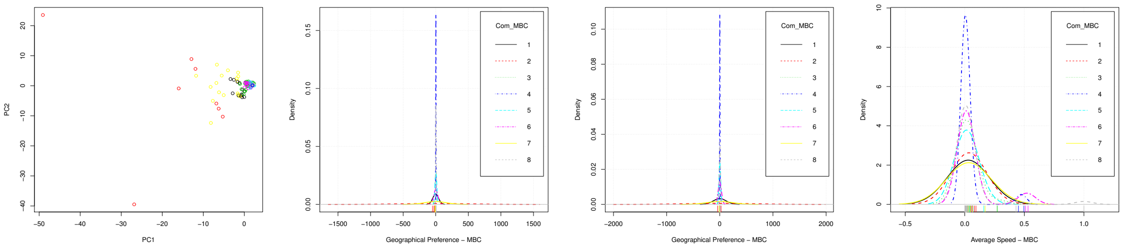


(a) GeoLife - HC (b) GeoLife - K-means (c) GeoLife - MBC (d) GeoLife - Fuzzy

(e) San Francisco - HC (f) San Francisco - K-means (g) San Francisco - MBC (h) San Francisco - Fuzzy

Figure 3.5: Average Speed for all clustering algorithms, for Geolife and San Francisco cabs mobility traces.

There are few methods that can be used to find how similar two images are. The simplest quality metric is the *mean square error* (MSE), computed by averaging the squared differences in the intensities between the pixels of two images. However, two images with the same MSE may have different types of errors. *Structural SIMilarity* (SSIM) Index identifies the information structures found in the images and therefore appears as an alternative to the previous method. A third metric to evaluate the clustering results of the proposed method is *Adjusted Rand Index* (ARI) [Santos and Embrechts 2009]. ARI is the corrected-for-chance version of the Rand index (RI), which measures the percentage of decisions (cluster assignments of all pair of users) that are made correctly.



(a) Clustering (b) PC1 (c) PC2 (d) Speed

Figure 3.6: MBC data clustering without using logit transformation - Geolife mobility trace.

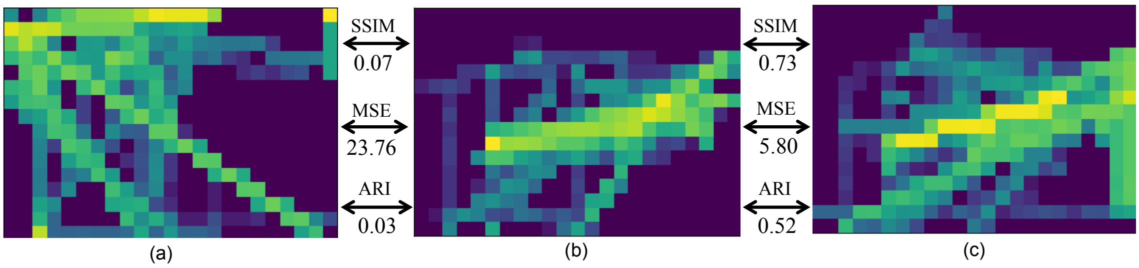


Figure 3.7: Trajectories and geographical preferences of three different nodes and the comparative metrics between each image pair. Images (b) and (c) are visually more similar, and present higher similarity metrics (e.g., SSIM and ARI) and smaller dissimilarity metrics (e.g., MSE), than (b) and (a).

The SSIM metric compares two images aligned and in the same scale, pixel-by-pixel. Three similarity functions are computed on the image data: luminance, contrast, and structural similarities.

The similarity for two images X and Y can be calculated as follows:

$$\mathrm{SSIM}(x,y) = [l(x,y)]^{\alpha} \cdot [c(x,y)]^{\beta} \cdot [l(x,y)]^{\gamma} \tag{3.7}$$

where the luminance comparison function $l(x,y)$ is a functions of the mean intensity of image $x$ and $y$ and is given by

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C1}. \tag{3.8}$$

The contrast comparison function $c(x,y)$ is the comparison of the standard deviation intensity of image $x$ and $y$ and is given by

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C2}. \tag{3.9}$$

Finally, the structure comparison function $s(x,y)$ is defined as

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x + \sigma_y + C3}, \tag{3.10}$$

where $\sigma_{xy}$ is the covariance of $x$ and $y$, that can be estimated as

$$\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{3.11}$$

Also, $\sigma_x$ and $\sigma_y$ are the standard deviation and $\mu_x$ and $\mu_y$ the average value for $x$ and $y$ respectively.

The constants C1, C2 and C3 are small constants that provide stability when the denominator approaches zero. More details about the SSIM algorithm can be obtained in [Wang et al. 2004].

Another successful cluster validation index for measuring agreement between two partitions in clustering analysis is Adjusted Rand Index (ARI) [Santos and Embrechts 2009]. ARI measures the relation between pairs of dataset elements, and can be calculated as follow:

$$ARI = \frac{\sum_{i,j}\binom{n_{ij}}{2} - [\sum_i \binom{n_i}{2} \sum_j \binom{m_j}{2}]/\binom{n}{2}}{\frac{1}{2}[\sum_i \binom{n_i}{2} + \sum_j \binom{m_j}{2}] - [\sum_i \binom{n_i}{2} \sum_j \binom{m_j}{2}]/\binom{n}{2}} \tag{3.12}$$

where $n$ is the number of elements in a given set S, $n_{ij}$ denotes the number of pairs in common between two partitions, $U = u_1, ..., u_R$ and $V = v_1, ..., v_C$, of this set. $n_i$ and $m_j$ are defined from the contingency table bellow.

| $U \setminus V$ | $V_1$ | $V_2$ | ... | $V_C$ | Sums |
|---|---|---|---|---|---|
| $U_1$ | $n_{11}$ | $n_{12}$ | ... | $n_{1C}$ | $n_1$ |
| $U_2$ | $n_{21}$ | $n_{22}$ | ... | $n_{2C}$ | $n_2$ |
| ... | ... | ... | ... | ... | ... |
| $U_R$ | $n_{R1}$ | $n_{R2}$ | ... | $n_{RC}$ | $n_R$ |
| Sums | $m_1$ | $m_2$ | ... | $m_C$ | |

Table 3.2: Number of pairs in common between two partitions $U$ and V of a set S.

ARI has a score between -1.0 and 1.0, where an ARI close to 0 means that the two partitions do not agree on any pair of points, and 1 stands for perfect match.

We use MSE, ARI and SSIM to evaluate the performance of the cluster algorithms, through the following steps:

1. Compute the values for the attributes for each mobile user and construct an image with the values of these attributes.

2. Compute the MSE, ARI and SSIM index for each pair of nodes in the network, each with its previously calculated image in the previous step.

3. Compute the MSE, ARI and SSIM mean for all node pairs belonging to the same community and belonging to different communities (for the 4 applied cluster algorithms).

Note that the more the values of ARI and SSIM approaches 1, the more similar are the attributes. The hypothesis to be tested is that nodes belonging to the same community have ARI and SSIM closer to 1, than nodes belonging to different communities. For the MSE results, the lower the value, the more similar the nodes are, and vice-versa. On the other hand, the computed similarity metrics are expected to be lower when computed between nodes belonging to distinct communities.

Table 3.3 displays the average value of SSIM, MSE and ARI without clustering the nodes in different communities. In other words, the similarity value between each pair of nodes in the network was calculated and the mean value was extracted. From this, we want to establish a benchmark to evaluate whether the proposed methodology was able to increase the similarity and dissimilarities between nodes belonging to the same and different communities, respectively.

| | SSIM | MSE | ARI |
|---|---|---|---|
| GeoLife | 0.24 | 20.70 | 0.13 |
| San Francisco | 0.73 | 0.077 | 0.38 |
| Dartmouth | 0.21 | 25.25 | 0.20 |

Table 3.3: Metrics Benchmark - Total average of metrics computed for all traces without considering the community structure output by the proposed methodology.

In Tables 3.4, 3.5 and 3.6, the values displayed in the *same* columns were obtained by calculating the mean values between each pair of nodes belonging to the same community, for all communities, and divided by the total number of communities. The values shown in the *diff* columns were obtained by summing the metric results between each pair of nodes belonging to different communities, divided by the total number of possible combinations between the pairs of nodes belonging to different communities.

Table 3.4 displays the SSIM, MSE and ARI values for the four algorithms studied for the GeoLife trace. We can observe that the proposed methodology was able to increase similarities (SSIM > 0.24, MSE < 20.70, and ARI > 0.13) for all the clustering algorithms, and dissimilarities (SSIM < 0.24, MSE > 20.70, and ARI < 0.13) for most of the algorithms. The K-means and Fuzzy algorithms presented the SSIM and/or ARI metrics that were slightly lower than the benchmark. The MBC algorithm showed the best results for all the metrics studied in this trace.

SSIM, MSE and ARI results for the four algorithms studied for the San Francisco trace are shown in the Table 3.5. In this scenario, the proposed methodology was also able to increase similarities and dissimilarities when compared to our proposed benchmark (see Table 3.3), for all four clustering algorithms. In this trace the MBC algorithm was slightly better at identifying similarities and dissimilarities between the node com-

| Algorithms<br>Communities | SSIM<br>(same) | SSIM<br>(diff) | MSE<br>(same) | MSE<br>(diff) | ARI<br>(same) | ARI<br>(diff) |
|---|---|---|---|---|---|---|
| K-means | 0.32 | **0.22** | 10.58 | 33.23 | 0.17 | **0.12** |
| HC | 0.38 | 0.25 | 9.25 | 37.72 | 0.19 | 0.13 |
| MBC | **0.446** | 0.27 | **7.34** | **38.79** | **0.24** | 0.15 |
| Fuzzy | 0.27 | 0.24 | 14.07 | 21.41 | 0.14 | **0.12** |

Table 3.4: Performance Evaluation - Geolife Trace

munities. With the exception of the ARI metric, which did not capture any difference between the clustering algorithms.

| Algorithms<br>Communities | SSIM<br>(same) | SSIM<br>(diff) | MSE<br>(same) | MSE<br>(diff) | ARI<br>(same) | ARI<br>(diff) |
|---|---|---|---|---|---|---|
| K-means | 0.962 | 0.67 | 0.001 | 0.088 | 0.386 | 0.377 |
| HC | 0.965 | 0.655 | 0.001 | 0.086 | 0.386 | 0.377 |
| MBC | **0.966** | **0.649** | 0.001 | **0.091** | 0.386 | 0.377 |
| Fuzzy | 0.957 | 0.698 | 0.001 | 0.084 | 0.386 | 0.377 |

Table 3.5: Performance Evaluation - San Francisco Trace

Finally, Table 3.6 displays the comparative results for the statistical metrics in the Dartmouth trace. In this scenario the proposed methodology was also able to increase the intra-cluster similarity for all algorithms studied. The MBC algorithm, once again, was the one that obtained the best results, being the algorithm that was able to meet all the metrics and overcome the benchmark for all similarities and almost all dissimilarities.

| Algorithms<br>Communities | SSIM<br>(same) | SSIM<br>(diff) | MSE<br>(same) | MSE<br>(diff) | ARI<br>(same) | ARI<br>(diff) |
|---|---|---|---|---|---|---|
| K-means | 0.38 | 0.19 | 18.60 | **27.02** | **0.29** | 0.18 |
| HC | 0.37 | 0.18 | 19.4 | 26.7 | **0.29** | **0.17** |
| MBC | **0.39** | **0.17** | **16.78** | 23.51 | **0.29** | **0.17** |
| Fuzzy | 0.27 | 0.23 | 22.4 | 23.4 | 0.24 | 0.26 |

Table 3.6: Performance Evaluation - Dartmouth Trace

### 3.4.4 Discussion

According to [Xu and Wunsch 2005], the time complexity of the K-means clustering algorithm is $O(NKd)$ and its storage space complexity is $O(N+K)$, where $N$ is the number of $d$-dimensional vectors and $k$ is the number of clusters. Hierarchical Clustering's time and storage space complexities are both $O(N^2)$, and Fuzzy C-Means' complexity is $O(N)$ [Xu and Wunsch 2005]. Since MBC represents a class of algorithms based on models, its computational complexity analysis will depend on the details of each model, but most models present a complexity around $O(N^2M_1)$, where $M_1$ is the number of iterations used for model estimation [Zhong and Ghosh 2003]. Thus, if an application requires low complexity approaches due to limited processing, storage, and energy capabilities, Fuzzy C-Means and K-means are the most indicated.

Based on our evaluation results, we observe that the MBC algorithm performed best for all traces evaluated. Consequently, MBC is preferred when accuracy is the main concern. However, K-Means is the most cost/time effective. Finally, if for a given application, a node can belong to multiple communities, then Fuzzy clustering would be the best choice.

## 3.5 Related Work

Social networks consist of nodes connected by socially meaningful relationships. The work in [Girvan and Newman 2002] and [Newman 2004] propose techniques to extract social relationships between users and also their social communities. In community-aware opportunistic networks, nodes are divided into several communities according to their relationships. Some works use data mining to detect user interests in certain geographic areas. For instance, in [Khetarpaul et al. 2011], a method to analyze users' aggregate GPS locations, and extract and rank users' location interests was proposed. In [Zheng et al. 2009a] GPS user trajectories are also used as a way to mine location interests and movement trajectories. Similarly, in [Giannotti et al. 2007], sequences of user trajectories are mined in order to find trajectory patterns and regions of interest.

Our work differs from efforts such as the ones outlined above since we focus on identifying user user communities based on features such as user geographical preferences and mobility characteristics.

The work reported in [Eagle and (Sandy) Pentland 2006] is a notable example of an approach that seeks to recognize social patterns in daily user activity through traces generated from mobile devices. It explores mobility profile and user behavior to propose a methodology for community identification based on the similarities found among different users. However, they use data from Bluetooth encounters and user location is inferred from cell tower location, thus losing spatial resolution as well as trajectory information.

Authors in [Ferrari et al. 2011] extract social networking patterns based on users' location in New York City, using the Twitter application. [Tang et al. 2012] proposes a method for extracting similarities among users of different social networks, in order to group them into communities. However, social networks provide information about user location or interests with low granularity, since information is only recorded when users actually use the social network. For example, uploading images in Instagram, or check-in using Foursquare.

In [Marbach 2016], a mathematical model to study communities in social networks is proposed. It is assumed that there is a population of agents who are interested in obtaining different types of content. The communities are formed in order to maximize their utility for obtaining and producing content. However, as stated by the authors, the model fails to capture some properties of information communities that have been observed in practice.

## 3.6 Conclusions

In this chapter, we proposed a methodology for identifying user communities based on users' geographical preferences and mobility attributes, such as speed and pause time. We hypothesize that users that have similar geographical preferences, e.g., frequent the same locales for similar amounts of time, have similar interests and could be classified as belonging to the same community. We show that the proposed methodology is able to identify similarities and dissimilarities between users belonging to the same and different communities, respectively.

The proposed methodology includes extracting mobility features from real human and/or vehicular mobility traces (e.g, obtained through GPS or Wi-Fi technology). It focuses on datasets that can be easily collected (e.g., through any *smartphone*), which eliminates the dependence on information often difficult to obtain, such as data from mobile operators, online social networks, or demographics.

Overall, the contributions of this work can be summarized as follows: (1) we developed a methodology for user community identification that relies solely on features extracted from user mobility traces; (2) we conducted a comparative performance study of four different categories of clustering algorithms for user community identification; (3) we validated the proposed methodology using a novel image-based similarity metric which allows to quantitatively assess the quality of the identified communities.

# 4. A Deep Learning Approach for Identifying User Communities Based on Geographical Preferences and Its Applications to Urban and Environmental Planning

Understanding human behavior and mobility will play a vital role in urban and environmental planning as cities continue to grow. Ubiquitous geo-location and localization technology and availability of bigdata-ready computing infrastructure have enabled the development of more sophisticated models to characterize human mobility in urban areas. In this chapter, our main goal is to extract geographical preference similarities among users and identify user communities based on such preferences. To this end, we introduce a novel deep autoencoder framework and use diverse urban mobility datasets to validate and evaluate our framework. Our experiments show that the proposed deep autoencoder increases contact times between users belonging to the same community by up to 80% when compared to the average contact time when not considering community structures and by up to 150% when compared to user communities extracted from raw datasets, i.e., without running data through the autoencoder. Moreover, our approach also increases contact time between members of the same community from 10% up to 125%, when compared to an alternate community extraction approach that uses Principal Component Analysis (PCA) instead. To the best of our knowledge, our proposal is the first to consider Deep autoencoder NNs to perform automatic extraction of non-linear features and mobility patterns from real mobility datasets.

## 4.1 Introduction

Currently, about fifty percent of the world's population lives in urban areas and the forecast is that by 2050 this percentage will grow to approximately seventy percent [Calabrese et al. 2014]. As such, the greatest wave of city migration is yet to come and together with it a wide range of challenges raised by the need to improve the style and quality of life of a growing urban population. According to [Calabrese et al. 2014], a better understanding of city dynamics would allow for improved services as well as minimized environmental impact resulting from urban expansion.

Urban mobility, defined as the displacement of people across an urban region over time [Boeing 2017], is critical to understand the dynamics of an urban center. As cities grow, the complexity of urban transportation and transit systems and the time people spend in transit will greatly increase. As a result, expanded- and new transportation services will be required demanding deeper investigation into urban mobility [Louf and Barthelemy 2014, Albino et al. 2015]. Additionally, understanding human mobility in urban areas is crucial to other city management and planning applications such as public health, emergency response, education, entertainment, shopping, etc [Hess et al. 2015a].

As computational resources become more widely available through cloud- and edge computing services, machine learning techniques, such as neural networks (NNs), which not too long ago were considered totally prohibitive in terms of their computational demands, have now become mainstream tools to handle the enormous amounts of data being generated by sensing devices embedded mostly everywhere. A special category of NNs named Deep autoencoders have been applied in a variety of domains, ranging from data augmentation, de-noising, activity and speed recognition, computer vision, to name a few [Liu et al. 2016].

In this work, we explore deep autoencoder architectures applied to learning user geographical permanence patterns in a variety of urban scenarios. Our main goal is to be able to extract geographical preference similarities among users and identify *user communities* based on such preferences. To this end, we introduce a novel deep autoencoder framework and use diverse urban mobility datasets to validate and evaluate our framework. Our experiments show that the proposed deep autoencoder increases contact times between users belonging to the same community by up to 80% when compared to the average contact time when not considering community structures and by up to 150% when compared to user communities extracted from raw datasets, i.e., without running data through the autoencoder. Moreover, our approach also increases contact time between members of the same community from 10% up to 125%, when compared to an alternate community extraction approach that uses Principal Component Analysis (PCA) [Bishop and Nasrabadi 2007] instead.

To the best of our knowledge, our proposal is the first to consider Deep autoencoder NNs to perform automatic extraction of non-linear features and mobility patterns from real mobility datasets.

## 4.2 Background and Related work

This section reviews previous work on mobility characterization as well as presents a brief overview of autoencoders and our rationale for using them to extract features from raw user mobility datasets.

### 4.2.1 Mobility Characterization

In recent years, the wide availability of localization devices and techniques, such as the Global Positioning System (GPS), cellular base-station and Wi-Fi positioning systems have enabled human mobility data to be captured and recorded. The availability of such positioning data not only enabled a variety of services and applications including road navigation, intelligent transportation systems, ride sharing, etc, but also motivated a large body of research on user mobility characterization and modeling. Below, we briefly describe some examples.

User mobility was found to be highly predictable and largely independent of the distance users cover on a regular basis [Song et al. 2010b], where most users visit the same places and share the same probability density function for places visited [Gonzalez et al. 2008]. Additionally, the probability of a user to visit new locations or returning to previously visited locations follows a scaling law pattern, i.e., the probability of users visiting a new place decreases over time, while the chances of returning to places they frequently visit increases. Also, it is well known that members of the same social group also present similar mobility behavior [Eagle and Pentland 2006]. Moreover, human mobility exhibits strong non-linear dynamics and hence can not be described by linear stochastic models [Domenico et al. 2013].

The study of movement patterns has applications in a wide range of fields, such as urban planning [Bastani et al. 2011,Nair et al. 2013,Zheng et al. 2014], identify similarities among individuals [Zheng et al. 2009b,Siła-Nowicka et al. 2016], evaluating and proposing mobile network protocols [Ferreira et al. 2018,Hong et al. 2010], predicting health condition [Mehrotra and Musolesi 2018,Canzian and Musolesi 2015], among others.

More recently, machine learning has been used to identify patterns within large-scale, high-dimensional mobility data [Toch et al. 2019,Zheng et al. 2014]. Most of the work to-date uses supervised learning to map data instances to labels and predict new, unlabelled data. However, much of the data available from positioning technologies is not labelled (e.g., GPS records).

Most efforts based on unsupervised learning use clustering algorithms to group instances that have similar behavior [Khoroshevsky and Lerner 2016,Zhu et al. 2014,Zheng 2015, Toch et al. 2019]. Clustering has been extensively explored in machine learning and data analytics. It tries to organize data observations into groups with similar features, and its performance highly depends on the quality of the input data [Aljalbout et al. 2018, Min et al. 2018]. However, these works do not consider the design of non-linear feature extraction, neither pre-processing and data transformations before clustering the data. Unfortunately, the expressive power of linear features is very limited: they cannot be stacked to form deeper, more abstract representations since the composition of linear operations yields another linear operation [Bengio et al. 2012].

Our work addresses this gap by using autoencoders to automatically learn from non-linear data representations, which can be stacked into deeper networks to better map input data into a feature space with improved data representation for clustering. As it will become clear from our experimental results, the ability to adequately represent non-linear features as well as the pre-processing transformations on the raw positioning data are crucial steps towards extracting valuable and meaningful mobility features from available human mobility datasets.

### 4.2.2 Deep Autoencoders

Deep learning (DL) models have been widely employed in recent years by researchers and practitioners to solve a plethora of different problems in many areas [LeCun et al. 2015, Bengio et al. 2012, Liu et al. 2016]. In the literature, it is possible to find DL architectures that fit specific purposes. For example, the use of U-Nets in medical imaging segmentation problems in  [Ronneberger et al. 2015], the use of Convolutional Neural Networks (CNNs) for semantic segmentation and image classification and recognition problems [Krizhevsky et al. 2012, Yu et al. 2017a, Wang et al. 2017, Long et al. 2015], application of Fully Connected (FC) neural networks for regression and classification problems [Rocha et al. 2007], Generative models for style transfer and data augmentation [Yu et al. 2017b, Antoniou et al. 2017, Reed et al. 2016], and Recurrent Neural Networks (RNNs) applied to sequential and temporal data analysis [Ziat et al. 2017, Graves et al. 2007, Cho et al. 2014, Che et al. 2018, Young et al. 2018], to name a few.

Most examples mentioned above are applied to supervised learning problems, where there are labels or ground truth (GT) values that can be used to train neural networks to be able to identify such labels automatically. While autoencoders (AE) are also an example of DL architectures, they can be applied to problems where a supervised approach is not possible, e.g., where there are no labels or GTs.

Identifying user community structures from raw mobility data requires unsupervised learning approaches since, most of the time, there is no previous knowledge from these raw records about the nature of the relationship between users, whether they belong to certain communities, etc. Principal Component Analysis (PCA) [Bishop and Nasrabadi 2007] is a class of algorithms that has been widely used for unsupervised learning problems, especially for dimensionality reduction. PCA applies linear transformations to the data, rotating axis on the directions of where the data presents the higher variability. This way, it is possible to describe most of the variability in the data with only a few variables, instead of using the raw higher-dimensional data. Autoencoders are another well known class of algorithms that can be applied to unsupervised representation learning and has been used in a number of applications, such as pattern identification and dimensionality reduction [Charte et al. 2018]. One of our the main reasons behind applying AEs to our work, instead of PCA, is the non-linear nature of the activations on the output of the AE layers [Wetzel 2017]. Our hypothesis was that AEs are able to capture non-linearities intrinsic to the data, that PCA cannot represent, given its linear nature.

An AE is a neural network architecture designed to learn data encodings in an unsupervised fashion. As mentioned before, it is typically used for dimensionality reduction, where the complexity and variability of the data is reduced into an encoded, more compact representation. Along with data reduction, there is also a reconstruction step that tries to reconstruct a representation as close as possible to the original input. In other words, the AE takes a set of unlabeled data $x \in R^n$ and tries to learn an approximation to the identity function to force the output to be as similar as possible to the input.

Autoencoders consist of three basic general components: (1) the encoder, that is the portion before the most compressed layer (or code) of the architecture. It compresses the input vector $x$ into a latent representation $h$ using a weight matrix $\omega$; (2) the code, $h$, or latent space representation, is a lower-dimensionality representation of the input. This reduced representation allow us to discover interesting structures about the data; and (3) the decoder, the portion after the code, maps $h$ back to the input, reconstructing it to obtain $x'$ with another weight matrix $\omega'$. Parameter optimizations are used to minimize the average reconstruction error between $x$ and $x'$. Usually, the input and output layers have the same dimensionality.

Training the network means learning the weight matrix $\omega'$ associated with all the neurons in the network. The basic unit of computation in a neural network is the neuron, often called a node or unit. During the training, each unit located in any layer in between input and output layers, also called hidden layers, receives several inputs from the preceding layer. The unit computes the weighted sum of these inputs and eventually applies an activation function, to produce the output. The most popular activation functions are *Linear, Logistic, ReLU, SELU,* and *Tanh*. The non-linear behavior of neural networks comes from the choice of these activation functions. After these steps, the output $x'$ is compared to the input $x$, and the error will be propagated to every individual unit using the back-propagation algorithm [LeCun et al. 2015]. Finally, each weight's contribution to the error is calculated and the descendent algorithm is adopted to adjust the parameters at each layer (i.e., update the weights). A typical loss function is the mean squared error or cross-entropy, when input values are binary or modeled as bits.

Another category of neural network that is widely used for image processing tasks is the convolutional autoencoder (CAE). CAEs are designed to process data inputs in the form of multidimensional arrays, e.g. images composed of 2D arrays containing pixel intensities in color channels. Some examples of data inputs are: 1D signals and time series, 2D for images and 3D for video or images. CAEs use the same principle as the traditional autoencoders discussed above, but instead of fully-connected layers, it contains convolutional layers in the encoder part and deconvolution layers in the decoder part.

CAE architectures are structured in several stages of convolutional and pooling layers [Sze et al. 2017]. The units in a CAE are organized in features maps, also known as convolutional filters, that are connected through a set of weights between the layers. Again a non-linearity activation function, such as ReLU, is passed through the weights. In this way, CAEs are able to detect local groups of values in an array of images that are often highly correlated, and also detect spatial invariance patterns. In other words, if a pattern is identified in a part of an image, it could appear also in other parts. Hence, the convolutional layer is responsible for detecting patterns from the previous layer, and the pooling layer for merging semantically similar features to one. In CAE, the pooling layer is responsible for reducing the dimension of the representation and creating an invariance to small shifts and distortions on the images. Usually, CAEs contain two or three stages of convolutional layers, non-linearity and pooling stacked, followed by more convolutional and/or fully-connected layers. The back-propagation algorithm is also used to training CNNs [LeCun et al. 2015].

The vast majority of applications of convolutional neural networks focus on image data, and so does the present work. Different from previous work, our proposed method is based on convolutional autoencoders, applied to extract features from real mobility data and identifying communities automatically.

## 4.3 Deep Learning Assisted User Community Identification

This section presents our proposed approach for extracting mobility patterns from GPS and Wi-Fi raw data using deep autoencoder models and clustering algorithms. Figure 4.1 summarizes the steps involved in the methodology. It also presents the key elements needed to the design of such approach, which will be described in detail in the subsections below.

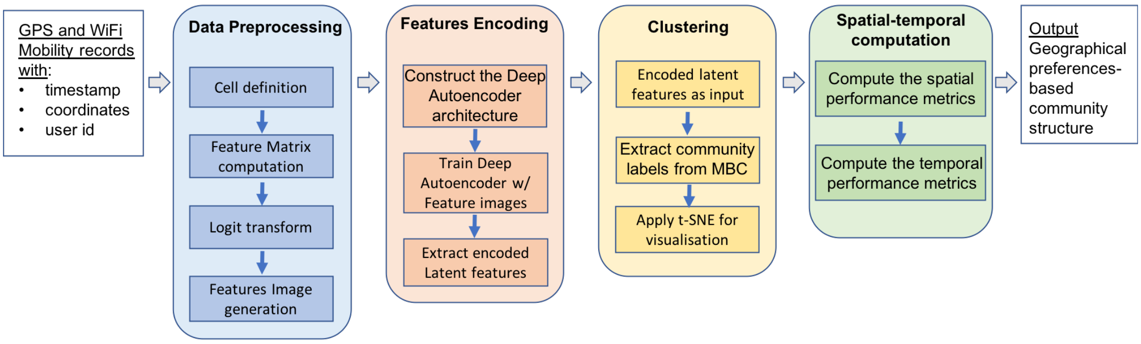

GPS and WiFi Mobility records with:
• timestamp
• coordinates
• user id
Data Preprocessing
Cell definition
Feature Matrix computation
Logit transform
Features Image generation
Features Encoding
Construct the Deep Autoencoder architecture
Train Deep Autoencoder w/ Feature images
Extract encoded Latent features
Clustering
Encoded latent features as input
Extract community labels from MBC
Apply t-SNE for visualisation
Spatial-temporal computation
Compute the spatial performance metrics
Compute the temporal performance metrics
Output
Geographical preferences-based community structure

Figure 4.1: The proposed community structure identification methodology illustrating all its components and data-flow.

### 4.3.1 Proposed Methodology

The Data Pre-processing steps to compute the mobility features for training the network by using the mobility traces, work as follows:

(Step 1) Cell Definition - Divide the geographical region in small parts, containing all the users as they move around into the *cells*. *Cell* division is described in more details in Section 4.4.1.

(Step 2) Feature Matrix Computation - Compute the user spatio-temporal features by defining a feature matrix consisting of $N$ rows and and $C$ columns, where $N$ is the number of users and $C$ the number of cells present in the dataset. Each position in the feature matrix $FM(i, j)$ contains the average time which user $i$ spent in cell $j$.

(Step 3) Logit Transformation - Apply the Logarithmic Likelihood *Logit* [Jaeger 2008] nonlinear transformation in the matrix $FM(i, j)$ to normalize the data. This step is important in order to evidence the similarities between each user.

(Step 4) Feature Image Generation - Generate an image for each node, by reshaping the mobility feature matrix into a 2D-image. For constructing this image, we simply take each position of the feature matrix as a pixel, and the values of each position is the intensity of that pixel. The size of the images are described in Section 4.4.1.

The encoding of features towards extracting user mobility features using deep-autoencoder consists of the following steps:

(Step 5) Deep Autoencoder Architecture - Construct the deep autoencoder architecture for each trace. Section 4.4.1 are detailed the parameters. It is not possible to train a single architecture for general use, since the models depend on the size of the input, and it varies with the application scenario (i.e., the size of the area, number of cells and feature matrix changes from scenario to scenario).

(Step 6) Deep Autoencoder Training - Train the deep autoencoder by using the input image representation obtained from the mobility features described above. In this work we trained three different autoencoders architectures in order to compare their impact on the experiments results.

(Step 7) Extract Encoded Latent Features - Once the network is trained, extract the reduced features from the autoencoder latent representation space. These features can be used to make predictions, and comparing the original input with the reconstructed image. The size of the reduced feature space depends on the autoencoder architecture and on the dataset. Section 4.4.1 presents the latent space size for all architectures and the 3 datasets.

We than cluster the latent variables from the autoencoder as follows:

(Step 8) Encoded Latent Features as input - Use the reduced latent feature representation obtained from Step 7 as input for the MBC clustering algorithm;

(Step 9) Extract community labels from MBC - Extract the community labels for every user from the MBC results;

(Step 10) Apply t-SNE technique for clusters visualisation. Apply t-SNE to reduce the extracted feature to a 2D plan for cluster vizualization.

Finally, we compute our evaluation metrics for measuring the spatio-temporal impact of the final clustering;

(Step 11) Compute the Spatial Evaluation Metrics - Use the similarity metrics to compute the quality of the cluster obtained in Step 9. Compute the MSE, ARI and SSIM index for each pair of images of nodes trajectories generated at Step 4. Compare the similarities metrics average for all node pairs belonging to the same community and belonging to different communities. We expect to see higher average similarity metrics (e.g., SSIM and ARI) between nodes belonging to the same community and higher average dissimilarity metrics (e.g., MSE that computes the difference between two samples) for nodes belonging to different communities. The similarity metrics are described in Section 4.4.3.

(Step 12) Compute the temporal evaluation metric - Compute the contact time between users belonging to the same and different communities, also based on the labels obtained in Step 9. We compute the average total time spent together in the same cell for pairs of nodes belonging to the same community and different communities. We expect to see higher contact time values between nodes from the same communities and lower values for nodes belonging to different communities.

Figure 4.2 summarizes proposed mobility characterization approach based on a 2D deep autoencoder architecture consisting of four parts: (I) Input of pre-processed data (II) training the model and making predictions, (III) clustering the code (IV) spatial and temporal computation. This figure shows the architecture for Conv/Dense model with parameters of GeoLife trace, as described in Section 4.4.1.

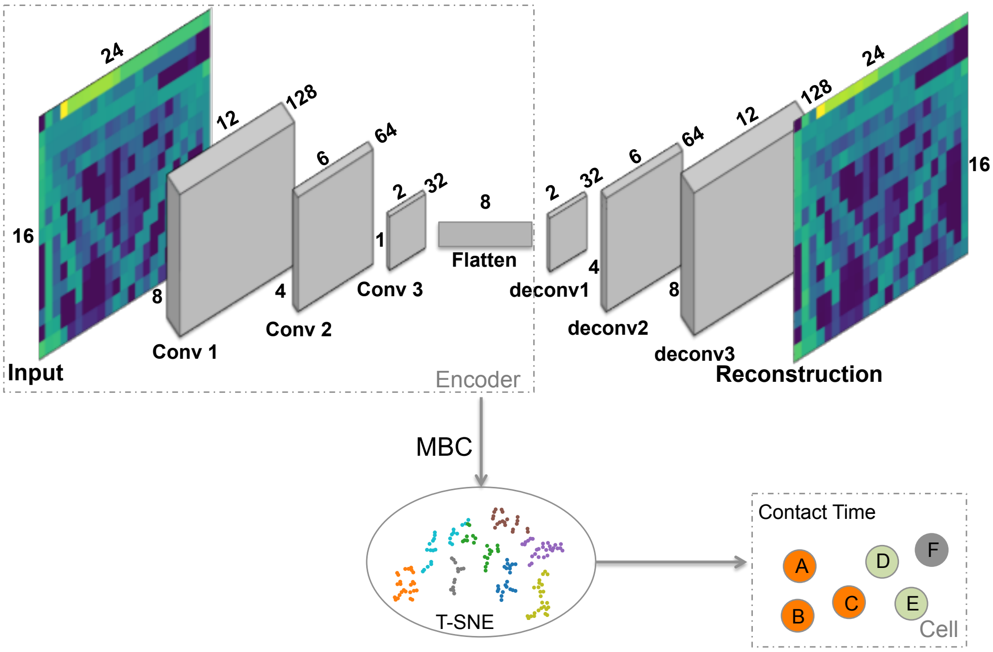


Figure 4.2: The architecture of the proposed deep autoencoder for GeoLife. There are three convolutional layers followed by a fully connected layer. The embedded layer is composed by only 8 neurons. Then a three deconvolutional layers reconstruct the Input. The 8 embedded features represent the encoding of the inputs and are used as input for the MBC clustering. The node's contact time within a cell is obtained from clusters labels.

### 4.3.2 Clustering the embedded layer

Two of the most popular feature-based clustering methods are K-means and Gaussian Mixture Model (GMM) [Jiang et al. 2017]. K-means updates the cluster centroids, by minimizing the within-cluster sum of squared errors, to generate K clusters, which are represented by the centroids. Another representative of model-based clustering is GMM which assumes that the original data consists of several Gaussian distributions. Data obeying the same independent Gaussian distribution is considered to belong to the same cluster [Xu and Tian 2015].

In this work we applied the Model-Based Clustering (MBC) for detecting the community structures from the data samples. MBC is a representative of a probabilistic model approach for data clustering that models the density function by a probabilistic mixture model. This method assumes that the data is generated by a mixture distribution and the clusters are defined by one or more mixture components [Dasgupta and Raftery 1995]. Each cluster, can be modeled by a Gaussian distribution that has three parameters: mean vector, covariance matrix and an associated probability in the mixture, where each point has a probability of belonging to each cluster. The Expectation-Maximization (EM) algorithm, initialized by hierarchical model-based clustering, is often used for estimating the parameters of the model, where clusters are centered at the mean value, and the geometric features (shape, volume, and orientation) are given by the covariance matrix.

The MBC consists of three main steps: (1) During initialization, it is necessary to specify the number of clusters and randomly initialize the distribution parameters for each group. The agglomerative hierarchical clustering is used to obtain the initial partitions of the data. (2) Then, compute the probability that each data point belongs to a particular cluster. (3) Application of the EM (Expectation-Maximization) algorithm, which is based on a maximum likelihood estimate, used to estimate the likelihood of the mixture parameters. (3) Finally, once the covariance matrix of the components lead to different models, the BIC technique (Bayesian Information Criterion) is used to choose the best model.

Model-Based Clustering (MBC) has linear complexity and attempts to deal with a more arbitrary shaped clusters. Due to the standard deviation parameter, the clusters can take on any ellipse shape, rather than being restricted to circles. This solve problems found in hierarchical and k-means algorithms, which tend to produce spherical and same size groups.

### 4.3.3 Visualizing data structure with t-SNE

t-SNE (t-Distributed Stochastic Neighbor Embedding) is a commonly used technique for the visualization of high-dimensional data in scatter-plot [Van Der Maaten 2014]. The technique aids in visualizing high-dimensional data by giving each data-point a location in a two or three low-dimensional data representation in such a way, that nearby points correspond to similar objects and that distant points correspond to dissimilar objects. It aims to preserve as much of the significant structure of the high-dimensional data as possible in the low-dimensional data.

Other techniques, such as Principal Components Analysis (PCA), also aims to preserve such structure. The difference between the two is that they differ in the type of structure they preserve. PCA is a linear technique, which keeps the low-dimensional representations of dissimilar data-points far apart. On the other hand, t-SNE is a non-linear technique that keep the low-dimensional representations of similar data-points close together. t-SNE is also capable of revealing global structure such as the presence of clusters in the data.

t-SNE computes an NxN similarity matrix in both the original data high dimension and in the low-dimensional latent space. The similarity matrix contains the probabilities given by a Student-t distribution between two data-points, where high probability means a pair of similar objects and low probability a pair of dissimilar ones. The low-dimensional embedding is learned by minimizing the Kullback-Leibler divergence between the probability distributions, in the original high dimensional and the low-dimensional data space, with respect to the locations of the points in the latent.

## 4.4 Experimental Methodology

We performed experiments on three mobility datasets to evaluate the performance of the proposed autoencoder architecture and other variants of autoencoders as well as PCA. The datasets we used in our experiments are described in Section 4.4.1. The setup of our experiments is presented in Section 4.4.2, and the evaluation metrics in Section 4.4.3.

### 4.4.1 Experimental Datasets

The experiments were performed on three datasets selected to represent different mobility scenarios: (1) GeoLife, (2) San Francisco cabs, and (3) Dartmouth. We briefly describe each of the datasets as well as the pre-processing applied on them.

#### 4.4.1.1 Geolife dataset

The GeoLife trace, refers to mobility in various scenarios in the city of Beijing, including different modes of transportation (e.g. walking, cycling and driving) [Zheng et al. 2010]. The trace contains GPS trajectories of 182 users, collected over a period of three years and sampled every 5 seconds. The user trajectory is represented in the dataset by a sequence of latitude and longitude set of coordinates over time, containing 17,621 trajectories, over 50,000 hours. The user trajectories are recorded in latitude-longitude geographical coordinates. We converted the geolocation coordinates to two-dimensional UTM Cartesian coordinates [Langley 1998]. A 2D-image was generated for each node, where each position of the matrix is a pixel. The 182 images in this dataset have 15 x 23 = 345 pixels, which corresponds to the number of cells for this trace. The values of the feature matrices were individually normalized so that the values of the pixels for every user would remained between 0 and 1. These images with normalized pixel values were used as input for training our architectures. All architecture were trained for 1000 epochs on the same dataset.

To extract features from the images for this dataset, we trained three different autoencoder architectures:

(1) Dense: fully connected network with five dense layers on the encoder part and five symmetric dense layers to reconstruct the input. This network has the following structure: the five encoding layers contain 512, 256, 64, 32 and 16 units, respectively. The latent mid layer is composed by an 8-unit feature vector that is used as input for the MBC clustering algorithm.

(2) Conv: convolutional network with four 2D convolutional layers on the encoder and four 2D convolutional layers to reconstruct the input. This network has the following structure: the four convolutional layers contain 128, 64, 32 and 16 filters of size (3x3). The latent layer contain 1 filter of size 3x3. We used the output of the activation in the middle layer, of shape [1, 2] as input features for the clustering algorithm.

(3) Conv/Dense: convolutional network with three 2D convolutional layers on the encoder, followed by a fully-connected layer in the latent space, and three simetric 2D convolutional layers to reconstruct the input. This network has the following structure: the three convolutional layers contain 128, 64 and 32 filters of size (3x3), strides over 2x2-pixel regions, respectively. The latent layer contains a flatten fully-connected layer with 8 units. The reshape image has size N = 1 x 2 and 128 filters. This leads to feature representations of dimensionality D = 8, which were used as input into the clustering algorithm.

#### 4.4.1.2 San Francisco dataset

The San Francisco trace is a vehicular trace and consists of GPS trajectories of 483 cabs in the City of San Francisco, USA [Piorkowski et al. 2009]. It was collected during

24 days with intervals between sample positional records ranging from 1 to 3 minutes. Similar to GeoLife trace, we generated an image for each node. The 425 images in this dataset have 12 x 58 = 696 normalized pixels. The number of epochs for training was set to 500.

To extract features from these images, we also trained three different autoencoder architectures:

(1) Dense: Same as GeoLife.

(2) Conv: convolutional network for San Francisco trace has the same architecture as described above for GeoLife trace except for the activations in the middle of the convolutional layers that have shape [1, 7]. This layer was again, used as input features for the MBC clustering algorithm.

(3) Conv/Dense: network architecture is the same as GeoLife Conv/Dense architecture described above.

#### 4.4.1.3 Dartmouth dataset

This dataset is a Wi-Fi association trace from the Dartmouth College campus' WLAN [Kotz et al. 2009]. The trace logs user access to Dartmouth College's campus WLAN using Access Point (AP) association and disassociation events. It has 6,524 users over 60 days. In this dataset the location of a user was set to the location of the access point to which the user was associated at the time. We worked with a subset of this dataset, with only the busy days where the larger number of APs were active. After filtering, this dataset allowed 2004 images (one for each active user) with 26 x 18 = 468 normalized pixels. To extract features from these images, we also trained three different autoencoder architectures:

(1) Dense: It has the same architecture as Geolife.

(2) Conv: It has the same architecture as San Francisco, except for the number of filters in each layer. The values of the filters are 128, 64, 8 and 4 with size 2x2. The latent layer contains 1 filter of size 2x2, leading to a shape of [1, 2] after activations, to be used as input for the clustering algorithm.

(3) Conv/Dense: It has the same architecture as GeoLife and San Francisco traces except for the parameters. The parameters for the three convolutional layers are 128, 256 and 16 filters of size (2x2) and strides equal to 2x2. The encoded latent vector used as input to the MBC is the output of the activations of the fully-connected layer with 8 units.

For all experiments the activation function used for all layers was the *Rectified Linear Unit* (ReLU), except the activation function on the output layer, which is linear. All the networks for all traces were trained to minimize mean square error and the optimizer function was *Adam*. The batch size was set to be the same as the number of image samples, i.e., 182 for GeoLife, 425 for San Francisco, and 2004 for Dartmouth. The training error and number of parameters of the obtained network, and the time elapsed during the training are shown in Table 4.3.

### 4.4.2 Experimental Setup

We ran experiments with three different types of pre-processing: (1) **Raw data** - where we used the feature matrix without any pre-processing as input for the MBC clustering algorithm , i.e., only the cell user time, as defined in Section 4.3-Step-2, is considered. (2) **PCA** - experiments feed the clustering algorithm with transformed feature matrix data by applying dimensionality reduction technique using principal component analysis (PCA). The transformed feature matrix contains the first eight principal components values. (3) **Autoencoder** - transforms the feature matrix by applying the three autoencoder architectures detailed in Section 4.4.1, before clustering and t-SNE are performed.

In all experiments, the t-SNE technique was used for visualizing the clusters. The perplexity parameter [Van Der Maaten 2014] used to compute the input similarities varied from 5 to 25. Dartmouth dataset was set to be 5, and GeoLife and San Francisco to be 25. Typical values for this parameter are reported in [Van Der Maaten 2014] to be between 5 and 50.

The computation time presented in Table 4.3 were measured on a laptop computer with an 2.0 GHz quad-core Intel Core i7 processor and 8 GB of memory.

### 4.4.3 Evaluation Metrics

In this section we proposed a validation method based on a number of metrics, that reflect not only how well our method depicts spatial similarities, but also temporal. The images generated from the mobility trace extracted features reflect the actual user displacement in the studied scenarios and thus indicate the user movement patterns. If we compare the trajectories of different pairs of users we will have a measurement of the similarities for each pair. Then, we compared the image pairs of users belonging to the same cluster and to different clusters, using three image comparison metrics, in order to verify if our method was able to group users that share similar movement patterns.

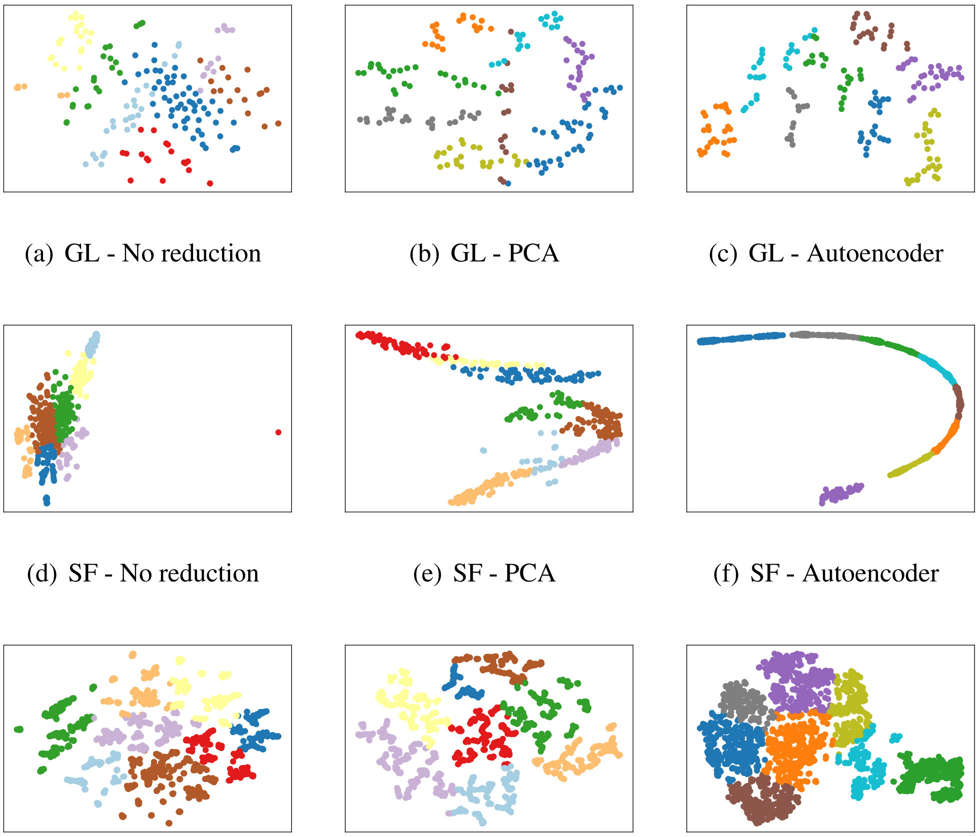
There are few methods that can be used to find how similar two images are. The simplest quality metric is the *mean square error* (MSE), computed by averaging the squared differences in the intensities between the pixels of two images. However, two images with the same MSE may have different types of errors. *Structural SIMilarity* (SSIM) [Wang et al. 2004] Index identifies the information structures found in the images and therefore appears as a better alternative to the previous method. A third metric to evaluate the clustering results of the proposed method is *Adjusted Rand Index* (ARI) [Santos and Embrechts 2009]. ARI is the corrected-for-chance version of the *Rand Index*, which measures the percentage of decisions (cluster assignments of all pair of users) that are made correctly.

Image quality comparison metrics work in our case as similarity indexes for spatial displacement, since the images we use as input for our proposed method, can be seen as heat-maps of each user's geographical preferences (i.e., time spent at a given location). Analogously, we seek to establish, rather then a spatial, also a temporal relationship metric to validate the proposed method. We argue that users belonging to the same group would spend more time together, as they would share similar interests, routes and geographical preferences, even though we only used data about users' individual geographical preferences in order to form groups. In this way, we calculate the encounter, or contact time between two nodes, visiting the same cell (i.e., sharing the same location at the same time), in a real trace. In other words, we summed for all nodes the duration of the encounters between all users present in the same cell at the same time. We then observed the total time spent together for members of the same group, and for members of different groups. The results are presented in next section.

## 4.5 Results

Figure 4.3 shows the output of the clustering step after applying t-SNE for visualization. It plots clusters for the three mobility datasets using three pre-processing/feature encoding methods, namely: Raw Data (i.e., no dimensionality reduction applied before clustering), PCA, and autoencoder. The results presented for the autoencoder approach were obtained using the Conv/Dense architecture. No significant visible differences were observed in the clustering visualization amongst the different autoencoder architectures.

As discussed before, one of the motivations behind applying autoencoders before clustering was the non-linear nature of the activations. The hypothesis was that the autoencoders would be able to represent non-linearities intrinsic to the data that PCA could not represent, given its linear nature. It is clear that the clusters become increasingly distinct as the linear and non-linear pattern identification techniques are performed and the best visual results are achieved when using autoencoders.



(a) GL - No reduction

(b) GL - PCA

(c) GL - Autoencoder

(d) SF - No reduction

(e) SF - PCA

(f) SF - Autoencoder

(g) DT - No reduction

(h) DT - PCA

(i) DT - Autoencoder

Figure 4.3: Clustering visualization after applying t-SNE for raw data, reduced data with PCA, and latent representation by Conv/Dense autoencoder architecture using GeoLife (GL), San Francisco (SF), and Dartmouth (DT) mobility datasets.

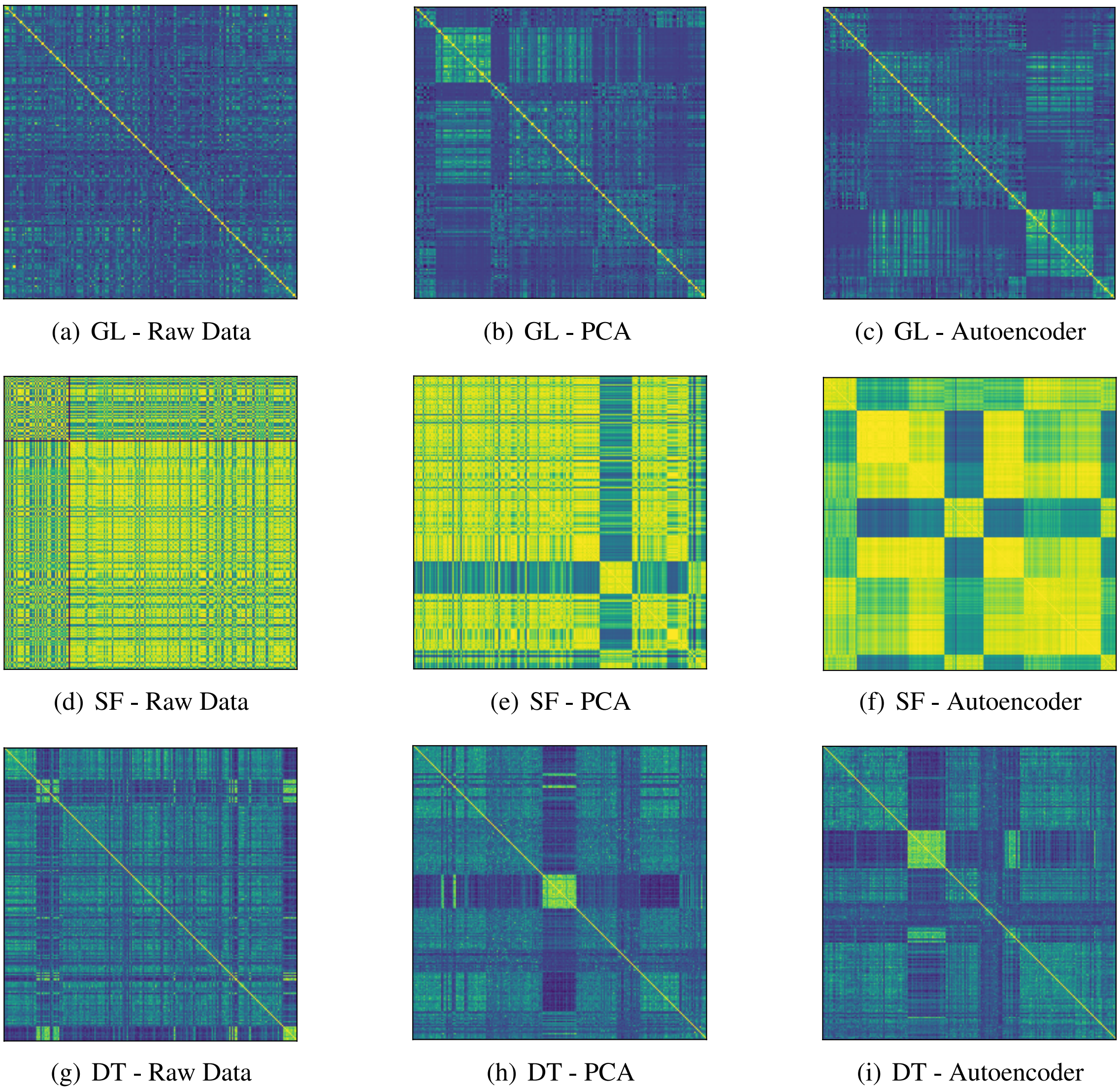
Figure 4.4 visualizes the matrices of pairwise SSIM values $ssim_{ij}$ between nodes $i$ and $j$, computed after considering the labels attributed to each user by our proposed methodology when using raw data, PCA, and autoencoder. The values of the 3 matrices computed for each dataset (i.e., each line of Figure 4.4) are in fact the same for all 3 images. What changes from image to image, for a specific dataset, is the reordering of the matrix rows and columns. Each matrix for each dataset has their rows and columns reordered and sorted according to their group labels (i.e., labels defined by the clustering algorithms) as they were rearranged in a way that nodes belonging to the same community are shown together in the matrices. In the figure, the brighter the color, the higher the similarity (i.e., SSIM value close to $1$), whereas darker colors indicate lower similarity.

We expect to see brighter colors along the main diagonal, as nodes belonging to the same community tend to have similar geographical preferences and as a result, a higher SSIM similarity metric among them. We find however, bright regions showing high similarity between different communities. This only means that sometimes, nodes of different communities can have also similar preferences. These interesting patterns reveled by the SSIM matrices can be interpreted as relationships between the different communities identified.

The noisier and less structured the patterns in the images, the less efficient in separating similar spatial preferences the method is. We can observe that images (a), (d) and (g), generated by applying clustering over the raw data, are the noisiest of them all, which means they are the less structured ones. Overall, cluster separation is more distinct when using the autoencoder and raw data yields less distinct clusters. The goodness of the method becomes visually more evident specially when looking at SF-Autoencoder results in Figure 4.4(f), where the different regions with different colors are much more evident and well defined when compared to the images resulting from the PCA and Raw Data. For these last two, we can still see more noisy and not well defined structured regions of SSIM values for different communities. PCA also represents better structures in these matrices when compared to Raw Data, however quantitative metric results will show that autoencoder is able to differentiate better the communities, not only in terms of spatial behavior, but temporal as well. We discuss this further below.

Quantitative metrics can help us better understand differences between each pattern identification approaches. Table 4.1 shows the performance of different data transformation methods. The table presents results for the three similarity metrics that are SSIM, MSE and ARI for GeoLife, San Francisco and Dartmouth mobility datasets. The results are shown in absolute values of similarity and also as the percentage difference in relation to clustering users using simply the raw data. Raw data clustering is then used as the baseline for assessing the quality of the clustering methodology and it only slightly and marginally improves the metrics when comparing to the metrics computed without considering community structures (i.e., no cluster). We observe that despite of the fact that PCA presents good results in some metrics, the autoencoder Conv/Dense architecture outperforms it in most metrics for all datasets.

In order to study the temporal impact of the groupings generated by the different methods of mobility patterns extraction, Table 4.2 presents results for the mean total time users spend together in a cell. We call this metric contact time. Table shows mean total time spent together and, $\mathrm{CI}_{min}$ and $\mathrm{CI}_{Max}$ represents lower and upper bounds respectively for a $95\%$ confidence interval.



(a) GL - Raw Data

(b) GL - PCA

(c) GL - Autoencoder

(d) SF - Raw Data

(e) SF - PCA

(f) SF - Autoencoder

(g) DT - Raw Data

(h) DT - PCA

(i) DT - Autoencoder

Figure 4.4: Similarity for each pair of nodes sorted by label groups, computed using the SSIM metric for GeoLife (GL), San Francisco (SF), and Dartmouth (DT) datasets. Yellow color shows high similarity and dark green low ones.

As we can see in Table 4.2, the Conv/Dense architecture presents results that are up to 85% better in relation to non-clustering, showing that the community detection method, when using Conv/Dense architecture, is able to extract community structures where the nodes belonging to the same community actually and consistently (i.e., for all datasets studied) spend more time together in the same location. On the other hand, nodes that do not belong to the same community tend not to meet, and the Conv/Dense architecture also managed to decrease contact time for members of different communities.

After presenting the results obtained in which the Conv/Dense architecture outperforms the alternatives in most metrics for all datasets, an analysis of computational complexity was performed by measuring the Time elapsed for training, training Loss and total

|  | SSIM<br>(same) | SSIM<br>(diff) | MSE<br>(same) | MSE<br>(diff) | ARI<br>(same) | ARI<br>(diff) |
|---|---|---|---|---|---|---|
| GL - No Cluster | 0.1264 (-) | - | 0.1074 (-) | - | 0.1140 (-) | - |
| GL - Raw Data | 0.1342 (-) | 0.1861 (-) | 0.1041 (-) | 0.0600 (-) | 0.1515 (-) | 0.1429 (-) |
| GL - PCA | 0.2512 (87%) | 0.0862 (54%) | **0.0683 (34%)** | **0.1512 (152%)** | 0.1942 (28%) | **0.0662 (54%)** |
| GL - Dense | 0.2266 (69%) | 0.1199 (46%) | 0.0776 (25%) | 0.1159 (93%) | 0.1866 (23%) | 0.0820 (43%) |
| GL - Conv AE | 0.2372 (77%) | 0.0959 (49%) | 0.0688 (34%) | 0.1349 (124%) | 0.1882 (24%) | 0.0971 (32%) |
| GL - Conv/Dense | **0.2784 (107%)** | **0.0860 (54%)** | 0.0687 (34%) | 0.1297 (116%) | **0.2120 (40%)** | 0.0816 (43%) |
| SF - No Cluster | 0.7632 (-) | - | 0.0347 (-) | - | 0.1501 (-) | - |
| SF - Raw Data | 0.7826 (-) | **0.7074** (-) | 0.0342 (-) | 0.0392 (-) | 0.1674 (-) | 0.1579 (-) |
| SF - PCA | 0.8747 (12%) | 0.7147 (1%) | 0.0159 (54%) | **0.0473 (21%)** | 0.1640 (2%) | 0.1498 (5%) |
| SF - Dense | 0.9528 (22%) | 0.7860 (11%) | 0.0010 (97%) | 0.0301 (23%) | 0.1618 (3%) | 0.1521 (4%) |
| SF - Conv | 0.9517 (22%) | 0.7370 (4%) | **0.0008 (98%)** | 0.0416 (6%) | 0.1637 (2%) | 0.1455 (8%) |
| SF - Conv/Dense | **0.9545 (22%)** | 0.7254 (3%) | **0.0008 (98%)** | 0.0410 (5%) | **0.1649 (1%)** | **0.1444 (9%)** |
| DT - No Cluster | 0.2018 (-) | - | 0.0198 (-) | - | 0.1804 (-) | - |
| DT - Raw Data | 0.2898 (-) | 0.1813 (-) | 0.0180 (-) | 0.0198 (-) | 0.2116 (-) | 0.1708 (-) |
| DT - PCA | 0.3242 (12%) | 0.1652 (9%) | 0.0161 (11%) | 0.0202 (2%) | 0.2435 (15%) | 0.1569 (8%) |
| DT - Dense | 0.3120 (8%) | 0.1931 (6%) | 0.0166 (11%) | **0.0207 (4%)** | 0.2444 (15%) | 0.1734 (2%) |
| DT - Conv | 0.3419 (18%) | 0.1829 (1%) | 0.0158 (12%) | 0.0204 (3%) | 0.2484 (17%) | 0.1709 (1%) |
| DT - Conv/Dense | **0.3580 (24%)** | **0.1625 (10%)** | **0.0157 (13%)** | 0.0202 (2%) | **0.2634 (24%)** | **0.1505 (12%)** |

Table 4.1: Performance evaluation using the three different metrics, SSIM, MSE, and ARI, for the three studied autoencoder architectures and data transformation, using MBC clustering on Geolife (GL), San Francisco (SF), and Dartmouth (DT) datasets.

number of parameters of each autoencoder architecture and for each dataset analyzed. Thus, as we can see in Table 4.3, the Time elapsed and Training Loss metrics for the Conv/Dense architecture are smaller than the ones measured for Conv architecture, even when the total number of network parameters was larger, as it was the case for the GL and DT datasets. In this way, the performance of the Time elapsed metric demonstrates the feasibility of using the our proposal in the identifying communities applied on smart mobility applications for urban and environmental planning.

## 4.6 Discussion and Application

An important characteristic of smart mobility applications such as Lime, Bird, Scoot, Lyft, and Uber-owned Jump is that users can either pick up or drop off equipment anywhere in the city. Attending to the needs of groups of users that share the same mobility pattern allows for (1) increasing the efficiency equipment usage (more shared bicycles/scooters available by certain route) and (2) reducing expenses with equipment relocation, by the equitable placement of shared vehicles.

Recently, some cities experimented issues with such vehicles (bikes/scooters) blocking sidewalks and building entrances, causing accidents (e.g., people tripping on scooters) and making public spaces less accessible to children and people with disabilities. The mobile pattern extracted by the proposed method may be used in the decision-making of where to make shared mobility equipment available. Therefore the application can make

|  |  | Mean | $\mathtt{CI}_{min}$ | $\mathtt{CI}_{Max}$ | # contacts |
|---|---|---|---|---|---|
| GL | No cluster | 4024842 | 3615976 | 4433707 | 32037 |
| GL - Raw Data | Same | 2863731 (-) | 2312482 (-) | 3414981 (-) | 17432 |
|  | Diff | 5410701 (-) | 4801901 (-) | 6019501 (-) | 14605 |
| GL - PCA | Same | 4795244 (67%) | 3672235 (59%) | 5918253 (73%) | 5470 |
|  | Diff | 3863132 (-29%) | 3428206 (-29%) | 4298059 (-29%) | 26567 |
| GL - Dense | Same | 6165288 (115%) | 4856236 (110%) | 7474340 (119%) | 4962 |
|  | Diff | 3632565 (33%) | 3212565 (-33%) | 4052565 (-33%) | 27075 |
| GL - Conv | Same | 5361019 (87%) | 4298002 (86%) | 6424036 (88%) | 4470 |
|  | Diff | 3872740 (-28%) | 3429318 (-28%) | 4316162 (-28%) | 27567 |
| GL - Conv/Dense | Same | **7175820 (151%)** | **5644526 (144%)** | **8707115 (155%)** | 4308 |
|  | Diff | 3535303 (-35%) | 3127447 (-35%) | 3943159 (-35%) | 27729 |
| SF | No cluster | 236227 | 230051 | 242404 | 176820 |
| SF - Raw Data | Same | 226120 (-) | 217844 (-) | 234398 (-) | 90246 |
|  | Diff | 246763 (-) | 237561 (-) | 255966 (-) | 86574 |
| SF - PCA | Same | 238084(5%) | 223890 (2%) | 252279 (7%) | 35454 |
|  | Diff | 235762 (-4%) | 228905 (-4%) | 242618 (-5%) | 141366 |
| SF - Dense | Same | 244589 (8%) | 229042 (5%) | 260136 (10%) | 28020 |
|  | Diff | 234653 (-5%) | 227923 (-4%) | 241383 (-6%) | 148800 |
| SF - Conv | Same | 251571 (11%) | 240800 (11%) | 262342 (12%) | 65842 |
|  | Diff | 227124 (-7%) | 219641 (-7%) | 234608 (-8%) | 110978 |
| SF - Conv/Dense | Same | **259195 (14%)** | **243613 (12%)** | **274777 (17%)** | 33792 |
|  | Diff | 230801 (-6%) | 224112 (-6%) | 237490 (-7%) | 143028 |
| DT | No cluster | 1006 | 985 | 1027 | 4018020 |
| DT - Raw Data | Same | 1129 (-) | 1075 (-) | 1183 (-) | 635442 |
|  | Diff | 983 (-) | 960 (-) | 1006 (-) | 3382578 |
| DT - PCA | Same | 1399 (24%) | 1335 (24%) | 1463 (24%) | 581186 |
|  | Diff | 940 (-4%) | 918 (-4%) | 962 (-4%) | 3436834 |
| DT - Dense | Same | 1493 (32%) | 1432 (32%) | 1554 (32%) | 607324 |
|  | Diff | 919 (-7%) | 897 (-7%) | 942 (-7%) | 3410696 |
| DT - Conv | Same | 1488 (32%) | 1404 (31%) | 1572 (33%) | 531164 |
|  | Diff | 933 (-5%) | 912 (-5%) | 954 (-5%) | 3486856 |
| DT - Conv/Dense | Same | **1557 (38%)** | **1485 (38%)** | **1629 (38%)** | 589824 |
|  | Diff | 911 (-7%) | 890 (-7%) | 933 (-7%) | 3428196 |

Table 4.2: Contact Time by users in the same cell on Geolife (GL), San Francisco (SF), and Dartmouth (DT) datasets.

more precise decisions regarding the distribution of equipment throughout the area visited by the same mobility group. Moreover, the job of collecting and reallocating assets still needs to be performed so that they are available to users in the appropriate parks and racks. This can also be optimized once users groups' movement patterns are known.

Carpool services, such as Waze, Scoop and Lyft are other examples of mobile-based applications. A carpooling activity consists of a matching process that enables drivers and passengers to be matched, and a daily route commute process that chooses the order at which passengers will be picked up and dropped off. The complexity of the problem of finding the length of the carpooling route, and select the best route scales dramatically according to the increase in the number of candidates and the number of passengers in the carpool [Xia J 2015]. Such applications improve the efficiency in the carpool sharing matching algorithm if they had knowledge about mobility patterns of groups/communities that share the same paths (fully or partially). A vehicle-to-passenger communication (V2P) approach to support communications between riders and drivers *that allows ride-sharing*, such as [Liu et al. 2010], could also benefit from our approach.

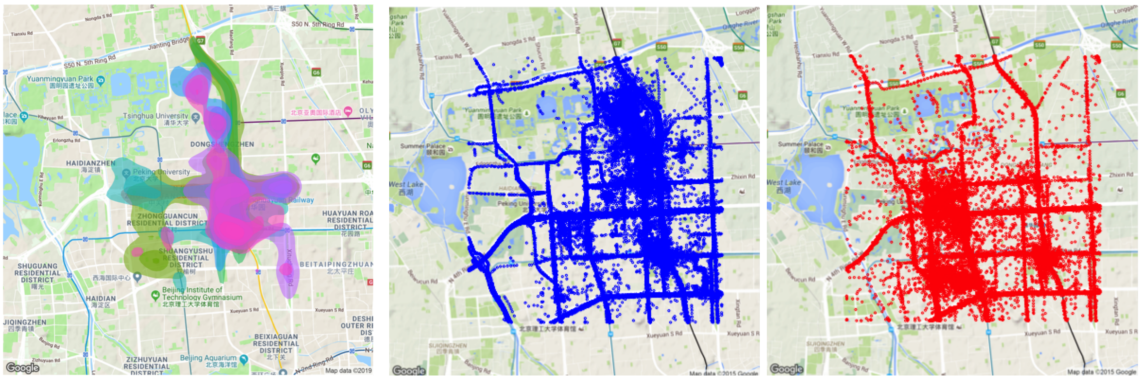
|  | Time elapsed (sec.) | Loss | Total Parameters |
|---|---|---|---|
| GL - Dense | 9.58 | 0.0046 | 483,481 |
| GL - Conv | 2184.21 | 0.0503 | 331,890 |
| GL - Conv/Dense | 268.36 | 3.1765e-04 | 495,017 |
| SF - Dense | 49.39 | 0.0245 | 1,015,600 |
| SF - Conv | 2500.02 | 6.9078e-05 | 388,017 |
| SF - Conv/Dense | 989.44 | 8.8735e-05 | 200,802 |
| DT - Dense | 78.34 | 0.0056 | 443,374 |
| DT - Conv | 1047.79 | 0.0072 | 72,122 |
| DT - Conv/Dense | 537.47 | 0.0044 | 266,073 |

Table 4.3: Time elapsed, Training Loss and total number of parameters for the three autoencoder architectures analyzed.

These applications normally rely on user id, home address and work address to perform matching. If the application knows a priory the information about the group structure, labeling users according to groups that reflect their geographical preferences, communication between drivers and riders and even the matching algorithms could be improved.

Message routing protocols on wireless networks can also take advantage of the mobility pattern identified by the proposed method, since users belonging to the same group spend more time together and can be good message forwarders. This characteristic was studied in several previous work [Yuan et al. 2016, Alajeely et al. 2017, Li and Wu 2009,Chuah and Coman 2009] and it was found that the time users spend together, in one region and at the same time, has a great impact on the probability of delivering messages in some types of networks. In this work, we show that the proposed group identification strategy increases the average total time spent together for members of the same group by up to 80% if compared to the average time pairs of users spend together without considering community structures. We show that our best autoencoder model increases average total contact time by up to 150% when comparing with an approach that clusters the users using raw mobility data. Moreover, the same metric is improved by 80% when considering our best approach, for one of the datasets in our study, when compared to the non-deep learning approach. On the other hand, when using our best model, we also find that users belonging to different groups (i.e, users who do not have strong relationship or common interests) have much smaller total contact then the same metric computed for groups extracted by other methods.

Figure 4.5(a) shows probability distribution of finding a member of a given group in a given position on the map, for all groups extracted by our method and plotted over the Beijing city map. As expected, different groups have different preferences for different


Yuanmingyuan Park
Tsinghua University
HAIDIANZHEN
Peking University
ZHONGGUANCUN RESIDENTIAL DISTRICT
SHUANGYUSHU RESIDENTIAL DISTRICT
HUAYUAN ROAD RESIDENTIAL DISTRICT
BEITAIPINGZHUANG
SHUGUANG RESIDENTIAL DISTRICT
HAIDIAN
Beijing Institute of Technology Gymnasium
JIQINGZHEN
ZIZHUYUAN RESIDENTIAL DISTRICT
Beijing Aquarium
BEIXIAGUAN RESIDENTIAL DISTRICT
Map data ©2019
Summer Palace
West Lake
Map data ©2015 Google

(a) All groups in the map (b) Blue group (c) Red group

Figure 4.5: Groups extracted by the proposed methodology plotted in the map of the city of Beijing.

regions of the city. This is more evident when looking at Figure 4.5(b) and Figure 4.5(c). These figures show the trajectories of all nodes belonging to groups 1 (blue) and 2 (red), respectively. The blue group visits much of the city, but has preference for the upper-right region of the map, while the red group preferences are more evident towards the lower-left region. Figure 4.6 also shows the probability distribution of finding a member of a group in the map, but filtered by week days. We can observe from the figure that geographical preferences can change over time. Hence, the proposed autoencoder architecture should be retrained at appropriated time intervals, according to the temporal sensitivity of the application applying the models, such as carpool sharing and V2P communications.

Smart mobility applications can distribute their equipment, parking lots and racks according to the groups' regions of preference, also considering groups' routes. The sharing algorithms of carpooling applications could run their matching searches between source and destination more efficiently within a group, increasing the probability of overlapping routs and origin and destination pairs. Also, algorithms for message forwarding or advertisement dissemination applications could take advantage of the location and encounters among same group users to forward their messages in order to reach a specific audience, increasing message delivery rate.

## 4.7 Conclusion

In this chapter we proposed an approach to automatically identify user community structures from real mobility records (e.g., GPS fixes and Wi-Fi network logs). We show that the proposed methodology which uses deep autoencoder to pre-process raw mobility datasets is able to more accurately uncover community structures which identifies groups

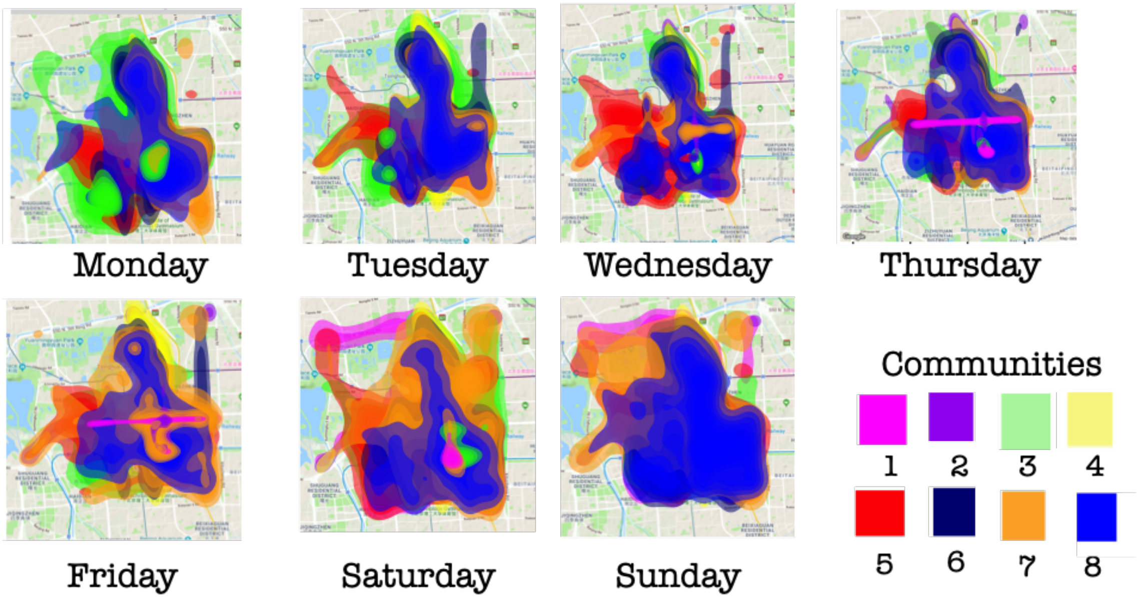


Figure 4.6: All communities by week days in the city of Beijing.

of users sharing common geographical interests and temporal relationships. The proposed methodology was built based on 3 main pillars: (1) geographical preferences feature generation, pre-processing and mobility data transformations, (2) deep autoencoders for dimensionality reduction and extraction of latent non-linear representations of the mobility data, and (3) clustering the output of the autoencoders and visualizing clusters by applying the t-SNE visualization technique.

Through extensive experimentation using three real mobility records representing diverse urban mobility scenarios we show the effectiveness of the proposed autoencoder-based methodology. Our results show that automatically extracted features lead to an improvement of the performance of spatial similarity metrics while increasing contact time for users in the same community from 30% up to 150%. Moreover, the proposed approach reduces the complexity of the features design task.

According to the results, we can notice that users belonging to the same community spend more time in certain geographic regions, which increases the probability of same community user meeting, when compared to users of different communities. We therefore expect that users belonging to the same community are potential opportunistic message forwarders. Also from the observation of the obtained results, users from different communities visit different geographic regions, although there is some intersection in some points. Since nodes in the same community tend to have a high frequency of encounters, the process of finding the best relay node and a good routing strategies can be improved by taking this social structure into account. Also the control messages overhead as well as the node storage table can be reduced if we consider only the group information instead of the individual. Thus, in next chapter we use the proposed community identification methodology that can assist in the choice of relay nodes considering the contact opportunities between nodes in opportunistic networks.

# 5. Deep autoencoder based community detection and its application to data forwarding in Opportunistic Networks

Many researchers are seeking solutions to the limitations (e.g., in terms of delay, delivery ratio, overhead, etc.) imposed by opportunistic connectivity, by developing schemes to data dissemination [Conti and Giordano 2014]. This chapter proposes a deep autoencoder community based routing protocol named DACCOR, which uses geographical preference features for making routing decisions. DACCOR uses a neural network trained on features extracted from real mobility records and uses its output to compute metrics that allows the protocol to make the next hop selection decisions. The performance of the proposed protocol is evaluated and compared with Epidemic and Prophet routing protocols in terms of delivery probability, latency, overhead, hop count and number of dropped messages. Finally, we show that by decreasing overhead and using less bandwidth and less radio, DACCOR is also able to dramatically and positively impact on energy consumption, optimizing mobile device's battery life.

## 5.1 Introduction

Thanks to the penetration of smartphones and their sensors in everyday life, associated with the tremendous volume of data to be exchanged between communicating devices, mobile communication technologies are no longer simply a means to connect a mobile device to the network infrastructure. The convenient short range communication functions integrated in smart devices (e.g. Bluetooth and WiFi) have given birth to some emerging applications such as Intelligent Transportation Systems (ITS), recommender systems, mobile data offloading, device to device communication, vehicular ad-hoc networking (VANET), internet of things (IoT) among others. Application-oriented paradigms are also emerging such as people centric networking, that puts people in the center, as the network is built with the users' devices. In this paradigm billions of users' mobile devices can be used for location-aware data collection, instrumenting the real world and generating observations – *crowd-sensing* – and also to offer cloud computing services.

In such scenarios, usually the environment is saturated with mobile devices, that can self-organize into networks for local communication amongst themselves. These networks are generally partitioned in disconnected islands, which can be connected by infrastructure network such as Wi-Fi or cellular networking, if they exist. However, even if such infrastructure exists the cost and energy consumption can be significant. Therefore, due to the pervasive nature of such environments, opportunistic networks emerge as a means to provide or extend communication.

Opportunistic networking is one of the most interesting evolution of the multi-hop networking paradigm. This success is mainly due to the fact that opportunistic networks do not consider node mobility a challenge but an opportunity to forward data [Conti and Giordano 2014]. In this network, temporary and occasional contacts between users and their devices present themselves as data transmission opportunities, while the user mobility can be seen as the transport media of this data. Opportunistic networking emerge as an area of growing interest with several challenging research issues.

In order to propagate messages properly, an efficient forwarding scheme should be able to send messages to specific and most suitable devices in such a way that chances of successful data recovery and delivery are increased. Another important concern should be to keep network overhead as low as possible, avoiding unnecessary transmissions. The importance of reducing overhead lies on the fact that mobile devices are limited in battery power, and reducing unnecessary radio transmissions can substantially decrease energy consumption, increasing battery life. Thus, the challenge is on how to forward data to relay nodes that have best chance to contact their destination, limiting also the number of copies being relayed in the network.

Opportunistic routing protocols seek to discover similar behavior or relationships among users in the network in order to use this information in the decision making of when and to whom to forward messages [Yuan et al. 2016, Alajeely et al. 2017, Li and Wu 2009, Chuah and Coman 2009]. It has been demonstrated that community scheme improve forwarding messages in specific scenarios [Yuan et al. 2016]. The first community-based proposed was BUBBLE [Hui et al. 2008], that uses the well known centrality metric and community structure to forward data. More recent works, such as [Xia et al. 2015, Yuan et al. 2014, Nguyen and Giordano 2012] to name a few, still proposing community based protocols to forwarding data. [Yuan et al. 2014] studied the impact of less popular nodes on the diffusion of messages on the network. More specifically, the authors removed nodes that were less “important" (low centrality) and found that message delivery performance was degraded.

Despite of the fact that taking community structure into account when making forwarding decisions, do improve performance, most community detection schemes extract these information via inter contact time between users [Eagle and (Sandy) Pentland 2006] (i.e. peer contact), or using information obtained through social networks or telecommunication company [Phithakkitnukoon et al. 2012, Motani et al. 2005], through calls made by mobile phones. Therefore, it is extremely useful to identify communities only through human mobility behavior, rather than depending on information obtained from external factors. Besides that, none of those works take into account the geographical preference and node density behavior described by our proposed mobility model and community identification methods.

This chapter presents a Deep AutoenCoder Community-based Opportunistic Routing protocol (DACCOR) for data forwarding in opportunistic networks. DACCOR takes into account the user mobility feature extraction and the community detection method presented in the previous chapters based on user geographical preference extraction using deep learning. The proposed DACCOR forwarding scheme uses community information to make forwarding decisions between members of different communities, and the computed user relationship metric (i.e., called SSIM metric, also presented in the previous chapter) to make forwarding decision within the community.

## 5.2 DACCOR - Deep AutoenCoder Community-based Opportunistic Routing protocol

This section provides the detailed design of the proposed Deep AutoenCoder Community-based Opportunistic Routing protocol (DACCOR) for data forwarding in opportunistic networks. We will introduce a user mobility feature extraction method and then present a novel community detection method, based on user geographical preference extraction using deep learning. Then, we discuss the user relationship metric used to identify similarities between members of a community and how it can be calculated. Finally, we present our social community based forwarding scheme used in DACCOR.

Mobility traces provide information about mobile devices location over time. By analyzing such information it is possible to obtain mobility characteristics of mobile users, including the distance traveled by the user, the distance between other devices, the time they spend together (i.e., contact time), etc. Since people move with a certain purpose

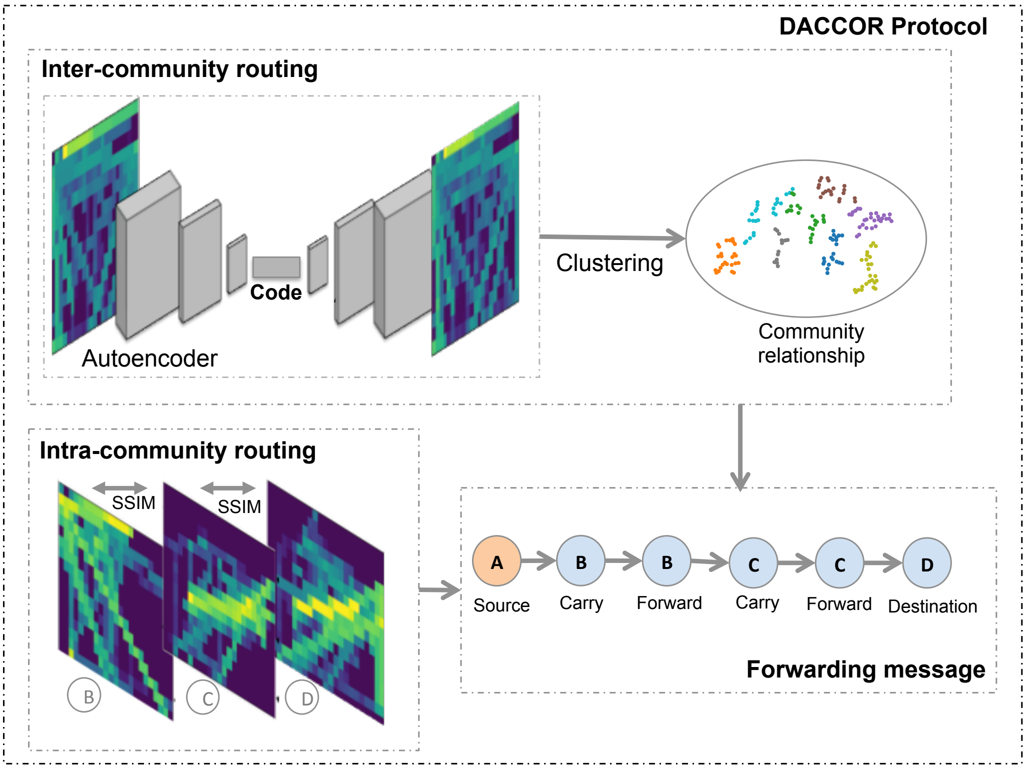


Figure 5.1: Parts from the DACCOR protocol, including the phases: inter-comnunity routing, intra-community routing and forwarding of messages.

(e.g. work to home), we assume that their locations and mobility characteristics may involve their interests and preferences. On the other hand, in an analogous way, we assume that if users do not share features and places in common, they have different interests and thus are less likely to have social relationships. Once such characteristics are found, it is possible to group individuals with similar mobility behaviours and geographical preferences/interests. Extracting and understanding such information may enable the successful elaboration of more realistic mobility models, increase the efficiency of message routing algorithms for opportunistic networks, development of context-based applications, among other applications.

In this work we are interested in extracting user mobility characteristics and transmission opportunities between devices. In order to do so we propose a set of steps to pre-process the raw mobility data from GPS and WiFi records. We start by extracting the two-dimensional maximum and minimum limits of the area defined in the raw data records and dividing this area into equal sized squared cells, constructing an spatial-temporal feature matrix. Such matrix holds the geographical and temporal characteristics of the users. The $i$-th row of the matrix represents the $i$-th user $\forall i \in [1, I]$, where $I$ is the maximum number of users. The $c$-th column of the matrix represents the $c$-th cell $\forall c \in [1, C]$, where $C$ is the maximum number of cells. Each of the $(i, c)$ matrix positions hold the time spent by user $i$ in that cell $c$. It is important to note that this feature matrix must be normalized so that an attribute with values higher than others does not dominate the distance metrics calculated during cluster analysis.

Once we construct the normalized feature matrix a nonlinear transformation is applied to the attribute matrix, using the logarithmic likelihood function *l*ogit. When applied to the normalized data this transformation modifies the proportions of the attributes matrix variables so that data between $(0, 1)$ takes real values, between $(-\infty, \infty)$, and is symmetric at $0.5$. This transformation, besides evidencing the differences and similarities between the observations for each variable, also improves pattern identification and learning, as it will become clear in the next section.

Finally, we generate the $i$-th image for each node $i$, by reshaping the $i$-th row of the mobility feature matrix into a 2D-image, that will reflect the dimensions of the area of the trace in cells. For constructing this image, we simply take each $c$-th position of the $i$-th row of the feature matrix as a pixel, where the values of each position is the intensity of that pixel. These images reflect the actual user displacement in the studied scenarios and thus indicate the user movement patterns and geographical preferences. Now the data is ready to be fed to the the neural network, as discussed in the next section.

### 5.2.1 Identifying user community structures using a deep learning approach

Deep learning (DL) models have been widely employed in recent years by researchers and practitioners to solve a plethora of different problems in many areas [LeCun et al. 2015, Bengio et al. 2012, Liu et al. 2016]. Identifying user community structures from raw mobility data requires unsupervised learning approaches since, most of the time, there is no previous knowledge from these raw records about the nature of the relationship between users, whether they belong to certain communities, etc. An autoencoder is a neural network architecture designed to learn data encodings in an unsupervised fashion. It is typically used for dimensionality reduction, where the complexity and variability of the data is reduced into an encoded, more compact representation [Charte et al. 2018]. Along with data reduction, there is also a reconstruction step that tries to reconstruct a representation as close as possible to the original input. In other words, the autoencoder takes a set of unlabeled data $x \in R^n$ and tries to learn an approximation to the identity function to force the output to be as similar as possible to the input.

Autoencoders consist of three basic general components: (1) the encoder, that is the portion before the most compressed layer (or code) of the architecture. It compresses the input vector $x$ into a latent representation $h$ using a weight matrix $\omega$; (2) the code, $h$, or latent space representation, is a lower-dimensionality representation of the input. This reduced representation allow us to discover interesting structures about the data; and (3) the decoder, the portion after the code, maps $h$ back to the input, reconstructing it to obtain $x'$ with another weight matrix $\omega'$. Parameter optimizations are used to minimize the average reconstruction error between $x$ and $x'$. Usually, the input and output layers have the same dimensionality. One category of neural network that is widely used for image processing tasks is the convolutional autoencoder (CAE). CAEs are designed to process data inputs in the form of multidimensional arrays, e.g. images composed of 2D arrays containing pixel intensities in color channels. CAEs use the same principle as the traditional autoencoders discussed above, but instead of fully-connected layers, it contains convolutional layers in the encoder part and deconvolutional layers in the decoder part. The vast majority of applications of convolutional neural networks focus on image data, and so does the present work. Our proposed methodology is based on convolutional autoencoders, applied to extract features from real mobility data and identifying communities automatically.

#### 5.2.1.1 Convolutional autoencoder design

Training the network means learning the weight matrix $\omega'$ associated with all the neurons in the network. The basic unit of computation in a neural network is the neuron, often called a node or unit. During the training, each unit located in any layer in between input and output layers, also called hidden layers, receives several inputs from the preceding layer. Analogously, convolutional autoencoder (CAE) architectures are structured in several stages of convolutional and pooling layers [Sze et al. 2017]. The units in a CAE are organized in features maps, also known as convolutional filters or even convolutional kernels, that are connected through a set of weights between the layers.

The unit computes the weighted sum of these inputs and eventually applies an activation function, to produce the output. The output of all, except the last of our convolutional layers are activated by a *Rectified Linear Unit* (ReLU) activation function, where the output is $f(x) = max(0, x)$. Only the last layer, the output layer, is activated by a linear function. The non-linear behavior of neural networks comes from the choice of these activation functions. Other popular ones are *Linear, Logistic, ReLU, SELU*, and *Tanh*. In this way, the convolutional autoencoder is able to detect local groups of values in an array of images that are often highly correlated, and also detect spatial invariance patterns. In other words, if a pattern is identified in a part of an image, it could appear also in other parts. Hence, the convolutional layer is responsible for detecting patterns from the previous layer, and the pooling layer for merging semantically similar features to one. In CAE, the pooling layer is responsible for reducing the dimension of the representation and creating an invariance to small shifts and distortions on the images.

After these steps, the output $x'$ (reconstructed node's trajectory image) is compared to the input $x$ (original node's trajectory image), and the error will be propagated to every individual unit using the back-propagation algorithm [LeCun et al. 2015]. Finally, each weight's contribution to the error is calculated and the gradient descendent algorithm is adopted to adjust the parameters at each layer (i.e., update the weights). We trained our autoencoder to minimize the mean square error and the optimizer used was *Adam*.

Usually, CAEs contain two or three stages of convolutional layers, non-linear activations and pooling layers, followed by more convolutional and/or fully-connected layers. Our proposed architecture contains a convolutional network with three 2D convolutional layers on the encoder, followed by a fully-connected layer in the latent space, and three symmetric 2D convolutional layers to reconstruct the input. This network has the following structure: the three convolutional layers on the encoder side contain 128, 64 and 32 filters with sizes that varies from (3x3) to (5x5) depending on the shape of the input image, and strides over 2x2-pixels. The latent layer contains a flatten layer with 8 units. The reshape image has size N = 1 x 2 and 128 filters. This leads to feature representations of dimensionality D = 8, which were used as input into the clustering algorithm. The deconvolutional layer is symmetric to the convolutional one with similar parameters.

#### 5.2.1.2 Clustering Encodings

In this work we applied the Model-Based Clustering (MBC) for detecting community structures from the lower-dimensionality representation of the input obtained from training the autoencoder. MBC is a representative of a probabilistic model approach for data clustering that models the density function by a probabilistic mixture model. This method assumes that the data is generated by a mixture distribution and the clusters are defined by one or more mixture components [Dasgupta and Raftery 1995]. Each cluster, can be modeled by a Gaussian distribution that has three parameters: mean vector, covariance matrix and an associated probability in the mixture, where each point has a probability of belonging to each cluster. The Expectation-Maximization (EM) algorithm, initialized by hierarchical model-based clustering, is often used for estimating the parameters of the model, where clusters are centered at the mean value, and the geometric features (shape, volume, and orientation) are given by the covariance matrix.

The MBC consists of three main steps: (1) During initialization, it is necessary to specify the number of clusters and randomly initialize the distribution parameters for each group. The agglomerative hierarchical clustering is used to obtain the initial partitions of the data. (2) Then, the probability that each data point belongs to a particular cluster is computed. (3) Finally the EM (Expectation-Maximization) algorithm is applied, which is based on a maximum likelihood estimate used to estimate the likelihood of the mixture parameters. (3) Finally, once the covariance matrix of the components lead to different models, the BIC (Bayesian Information Criterion) is applied to choose the best model.

### 5.2.2 Community member user relationship

Image quality comparison metrics work in our case as similarity indexes for spatial displacement, since the images we use as input for our autoencoder architecture, can be seen as heat-maps of each user's geographical preferences (i.e., time spent at a given location). Analogously, we seek to establish, rather then a spatial, also a temporal relationship metric. We argue that users belonging to the same group would spend more time together, as they would share similar interests, routes and geographical preferences, even though we only used data about users' individual geographical preferences in order to form groups.

In this way, we use the Structural SIMilarity (SSIM) Index to compute the similarity in the mobility behaviour of two nodes by comparing the similarities between the two user trajectory images. The SSIM index identifies the information structures found in the images and therefore is used to compute the similarity between a pair of nodes. The SSIM algorithm compares point by point two images aligned and scaled. Three similarity functions are computed on the image data: (1) luminance similarity, (2) contrast similarity, and (3) structural similarity. Note that the more the value of SSIM approaches 1, the more similar are the two images, and also, the more similar are the attributes of the two nodes.

The similarity for two images $X$ and $Y$ can be calculated as follows:

$$SSIM(x,y) = [l(x,y)]^{\alpha}.[c(x,y)]^{\beta}.[s(x,y)]^{\gamma} \tag{5.1}$$

where the luminance comparison function $l(x,y)$ is a functions of the mean intensity of image $x$ and $y$ and is given by

$$l(x,y) = \frac{2\mu_x\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C1}. \tag{5.2}$$

The contrast comparison function $c(x,y)$ is the comparison of the standard deviation intensity of image $x$ and $y$ and is given by

$$c(x,y) = \frac{2\sigma_x\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C2}. \tag{5.3}$$

And, the structure comparison function $s(x,y)$ is defined as

$$s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x + \sigma_y + C3} \tag{5.4}$$

where $\sigma_{xy}$ can be estimated as

$$\frac{1}{N-1}\sum_{i=1}^{N}(x_i - \mu_x)(y_i - \mu_y) \tag{5.5}$$

The constants C1, C2 and C3 are small constants that provide stability when the denominator approaches zero. More details about the SSIM algorithm can be obtained in [Wang et al. 2004].

In summary, the encoding of features towards extracting user mobility features using deep-autoencoder consists of the following steps: (1) generate the mobility image based on the feature matrix extracted from the real trace; (2) construct the deep autoencoder architecture for the trace. It is not possible to train a single architecture for general use, since the models depend on the size of the input, and it varies with the application scenario (i.e., the size of the area, number of cells and feature matrix changes from scenario to scenario); (3) train the deep autoencoder by using the input image representation obtained from the mobility features described above; (4) once the network is trained, extract the reduced features (code) from the autoencoder latent representation space. These features can be used to make predictions, and comparing the original input with the reconstructed image; (5) use the reduced latent feature representation as input for the MBC clustering algorithm; (6) extract community labels from the clustering algorithm; (6) extract node similarity from SSIM metric. We now have extracted mobility patterns and are in possession of community structure and relationship indicators between each pair of nodes. The community structure is indicated by the community labels given by the clustering algorithm. The relationship between every node is given by the SSIM index values computed for every pair of nodes, which indicates their geographical preferences similarity. We can take advantage of these information to make more intelligent and educated decisions on when and to whom forward a message to in the context of opportunistic and delay tolerant networks. The following sections introduce DACCOR, a new forwarding protocol that takes advantage of the community structure information.

## 5.3 DACCOR Forwarding Protocol

The proposed DACCOR forwarding scheme uses community information to make forwarding decisions between members of different communities, and the user relationship (SSIM metric) to make the forwarding decision within the community. This approach provides DACCOR with scalability by decreasing the vector size needed to carry neighboring information. It can be done by tuning the number of communities that exist on the network, i.e. the larger the number of communities the smaller the number of members belonging to any given community.

### 5.3.1 Community Affinity and SSIM metrics for communities and nodes similarities

We consider the following features in order to determine the suitability of a relaying node to carry a message to the destination:

1. Node community ID (cID) - The community label carried by each node and extracted by the proposed community identification using deep learning, as detailed in Section 5.2.1.

2. Community affinity (CA) - The relationship between the communities of the encountered node and destination node. It is computed by the average of SSIM for pairs of nodes belonging to the two communities (i.e., the community of the encountered node and the community of the destination node), as further discussed in the following Section 5.3.2.

3. Node Similarity - The similarity between the encountered node and destination node, if it is the case, i.e. if both nodes belong to the same community. It is computed using the SSIM index for each pair of nodes in the community, as detailed in Section 5.2.2.

We assume that all nodes know the information about which community it belongs to, the community relationship between all communities in the network, and the SSIM metric of all members of their own community.

The forwarding is carried by combining two strategies: (1) Inter-community based on social community membership and (2) Intra-community based on user node relationship.

### 5.3.2 Inter-community Routing

Our proposed social community is calculated based on the user mobility patterns extracted in Section 5.2.1 and shared by members of the network. The inter-community forwarding scheme uses the community affinity information to make the forwarding decision. In DACCOR, each node maintains an $N$ x $N$ community affinity matrix, which is defined as:

$$CA = \begin{bmatrix} C_{11} & C_{12} & ... & C_{1N} \\ C_{21} & C_{22} & ... & C_{2N} \\ ... & ... & ... & .... \\ C_{N1} & C_{N2} & ... & C_{NN} \end{bmatrix}, \tag{5.6}$$

where $C_{n,m}$ denotes the community affinity between each community $n$ and community $m$, for all $n$ and $m \in \{1..N\}$, where $N$ is the total number of communities. $C_{n,m}$ is obtained by

$$C_{nm} = \begin{cases} 1, & \text{if } n = m \\ E(\text{SSIM}_{n,m}), & \text{if } n \neq m. \end{cases}$$

$E(\text{SSIM}_{n,m})$ is the average of SSIM between pairs of nodes belonging to communities $n$ and $m$. Thus, the $CA$ matrix dictates the message forwarding between nodes belonging to different communities, i.e. nodes that have different communities ID (cID).

For example, assume that a node $i$ has a message to destination node $d$ and encounters with node $j$. Also, assume that node $i$ and node $d$ belong to different communities, $n$ and $m$, respectively, i.e. $C_{n,m} \neq 1$. If j is the destination (i.e., j = d), $i$ will forward the message $M$ to $j$. If node j belongs to the same community of node d, then node $i$ forwards the message to node $j$. If not, node $j$ does not belong to the same community of node $d$, and node $i$ will verify the community affinity matrix to decide if it is going to forward $M$ to node $j$ or not, i.e. if $C_{j,d} > C_{i,d}$ then $i$ is going to forward $M$ to $j$. Otherwise, $i$ will not forward it and continue to hold and carry message $M$. When the message has reached the destination's community, DACCOR looks for relay nodes that are more similar to the destination node, by using the SSIM metric as discussed in Section 5.2.2. Algorithm 4 shows the detailed forwarding algorithm for the first phase.

**Algorithm 4** Inter-community AE community based algorithm

```
Input: $node_d$, $node_i$, $node_j$, community ID (cID), $M$, CA
while $node_d$ without M do
    current $node_i$ encounters $node_j$ without $M$
    if $node_j$ is $node_d$ then
        $node_j$ accepts $M$ from $node_i$ and the forwarding process ends
    else
        if $node_j$ and $node_d$ in $C$ then
            $node_i$ forwards $m$ to $node_j$
        else
            if CA($cID_j$,$cID_d$) > CA($cID_i$,$cID_d$) then
                $node_i$ forwards $m$ to $node_j$
            end
        end
    end
end
```

### 5.3.3 Intra-community Routing

For intra-community routing, the proposed protocol uses the $SSIM$ metric, defined in Section 5.2.2 to determine the probability of message forwarding for each encountered node. Thus, the $SSIM$ metric dictates the message forwarding between members of the same community.

For instance, assume there is a node $i$ carrying a message, an encountered node $j$ and a destination node d, all belonging to the same community. DACCOR will search for the higher geographic similarity, by comparing the SSIM metric between the two nodes and the destination node. In other words, if SSIM($node_j$, $node_d$) > SSIM($node_i$, $node_d$) the message is forwarded from node $i$ to node $j$.

In addition, to increase the probability of message delivery, even if node $j$ and node $d$ are not part of the same community, the message can be forwarded using the community affinity metric. We understand that a node may have a social relationship with more than one community on the network and therefore if node $j$ has community affinity that is greater than the average community affinity for all communities in the network, the message will be forwarded. Algorithm 5 shows the detailed forwarding algorithm for the second phase.

**Algorithm 5** Intra-community AE community based algorithm

```
Input: $\mathrm{node_d}$, $\mathrm{node_i}$, $\mathrm{node_j}$, community ID (cID), $\mathrm{M}$, CA, SSIM, E[CA]
while $\mathrm{node_d}$ without m do
    current $\mathrm{node_i}$ encouters $\mathrm{node_j}$ without $\mathrm{M}$
    if $\mathrm{node_j}$ is $\mathrm{node_d}$ then
        $\mathrm{node_j}$ accepts $\mathrm{M}$ from $\mathrm{node_i}$ and the forwarding process ends
    else
        if $\mathrm{node_j}$ is in $\mathrm{C}$ and SSIM($\mathrm{node_j}$,$\mathrm{node_d}$) > SSIM($\mathrm{node_i}$,$\mathrm{node_d}$) then
            $\mathrm{node_i}$ forwards $\mathrm{M}$ to $\mathrm{node_j}$
        else
            if CA($\mathrm{cID_j}$,$\mathrm{cID_d}$) > E(CA) then
                $\mathrm{node_i}$ forwards $\mathrm{M}$ to $\mathrm{node_j}$
            end
        end
    end
end
```

### 5.4 Performance Evaluation

The performance of opportunistic networks may vary significantly, depending on several factors such as node mobility, population density and the distance from sender to receiver. Depending on the scenario delivery latency may vary from minutes to hours or even days, and delivery probability may range from close to 0 or go up to 1. The key factors are the routing algorithms used and how well their design assumptions match the actual mobility patterns.

In this way, we evaluate the proposed routing protocol against two benchmark protocols PRoPHET [Lindgren et al. 2003] and Epidemic [Vahdat et al. 2000] protocols in terms of delivery ratio, average latency, hop count and overhead, when using synthetic and real mobility records, to demonstrate the effectiveness of our community based protocol. Our rationale for choosing these routing protocols for our comparative performance study of DACCOR is as follows.

Epidemic, despite its limitations, has been widely used to evaluate opportunistic networks and their protocols and serves as the upper bound for delivery and cost. In Epidemic protocol, messages are stored locally and, during any encounter, the message is flooded through the network. Epidemic shows the upper bound of delivery ratio of any routing methods. However, it also shows the biggest message overhead. In contrast, DACCOR uses the community approach to select appropriate forwarders that presents greater geographical similarity with the message destination, and thus controls the number of hops, and costs less network resources (e.g., bandwidth, device battery, etc).

PRoPHET uses the delivery predictability metric based on historical contact frequency between nodes to choose the next relay nodes. The difference between DACCOR and PRoPHET is that DACCOR selects the next relay nodes based on the geographical similarity, whereas PRoPHET relies on node encounter history to estimate which node has the highest "likelihood" of being able to deliver a message to the final destination.

PRoPHET is a non-oblivious benchmark that has been evaluated against several previous works, including social-based protocols. For example, BubbleRap [Hui et al. 2008], the first social-based protocol, compares its performance against PRoPHET. As a result, the authors found a similar delivery ratio to PRoPHET with half of the PRoPHET cost.

BubbleRap protocol uses two network characteristics that are community and centrality. Some devices interact with more devices than others in a community and they are considered to have high centrality. This approach uses this metric, i.e. centrality, to relay nodes to forward node to the destination.

Also, more recently proposed protocols have extended PRoPHET [Pathak et al. 2017], but preserve the core node contact history-based features. As such, since PRoPHET has been evaluated against other algorithms before [Pathak et al. 2017], including the social-based ones [Hui et al. 2008], it is a good target to compare with DACCOR.

The following section presents the evaluation scenarios and experimental setups used in our evaluation.

### 5.4.1 Experimental Datasets

To illustrate our approach, the datasets used in this study were selected to cover a range of scenarios considering vehicular and human mobile networks: GeoLife [Zheng et al. 2010], San Francisco cabs [Piorkowski et al. 2009] and Helsink [Keränen et al. 2009]. The GeoLife trace, refers to mobility in various scenarios in the city of Beijing, including different modes of transportation (e.g. walking, cycling and driving). The trace represent GPS trajectories of 182 users, collected over a period of three years and sampled every 5 seconds. The user trajectory is represented in the dataset by a sequence of latitude and longitude set of coordinates over time. The dataset contains 17,621 trajectories with a total distance of 1,292,951 kilometers and total duration of over 50,000 hours. In our simulation we used 12 hours of the trace containing 169 nodes. A summary with the traces parameters are shown in Table 5.1

| Trace | # users | Type | Speed (km/h) |
|---|---|---|---|
| GeoLife [Zheng et al. 2010] | 169 | - | - |
| SF Taxis [Piorkowski et al. 2009] | 483 | - | - |
| Helsink [Keränen et al. 2009] | 80 | Pedestrians | 1.8 to 5.4 |
| | 40 | Cars | 10 to 50 |
| | 6 | Trams | 25 to 36 |

Table 5.1: Summary of user mobility traces considered in our study.

The vehicular mobility record is related to the movement of taxis in the city of San Francisco/USA. The SF trace represents GPS trajectories of 483 users, and was collected for 24 days with samples ranging from 1 to 3 minutes.

Helsink is a synthetic mobility trace available in the ONE simulator. Nodes move on the simulation area according to a mobility trace generator, where 80 are pedestrians, 40 are cars and 6 are trams. Cars run at uniformly distributed speed from 10 to 50 km/h and trams at 25 to 36 km/h with uniformly distributed pause times of 10 to 120 and 10 to 30 seconds, respectively. In this scenario, trams follow predefined routes defined by the simulator, while pedestrians and cars choose random destinations in their reach on the map and move towards theirs next destination by following a shortest path algorithm, such as Dijkstra algorithm.

| Parameters | values |
|---|---|
| Transmission rate | 2 Mbps |
| Radio range | 150m |
| TTL | 12h |
| Buffer size | 1GB |
| Simulation time | 12h |
| Message sizes | 500KB to 1000KB |

Table 5.2: Simulation parameters.

### 5.4.2 Experimental setup

We designed simulation experiments with the Opportunistic Network Environment (ONE) simulator. We assume that devices uses WLAN radios with a transmission rate of 2 Mbps and data range of 150 m. The radio range has minor impact and do not change the elementary interaction characteristics between the devices [Keränen and Ott 2007]. The mobile devices have 1 GB of free buffer space for storing and forwarding messages. Simulation time and message time-to-live (TTL) was define to last for 12 hours. Random source nodes generate messages to a randomly chosen destination on average once every interval of time. The length of such interval was varied in order to change network load conditions, i.e., smaller inter-message periods allow a greater load, while larger intervals decrease the load in the network. Inter-message periods are randomly chosen over the following average intervals: 6, 8, 12, 18, 60 and 600 seconds (uniformly distributed). Message sizes are uniformly distributed between 500 KB and 1 MB. Given average message sizes, inter-message transmission intervals and channel capacity, these intervals were chosen to represent the network load ranging from 0.1 to 1 proportion of the channel capacity.

In all experiments, we compare each protocol using the following routing metrics.

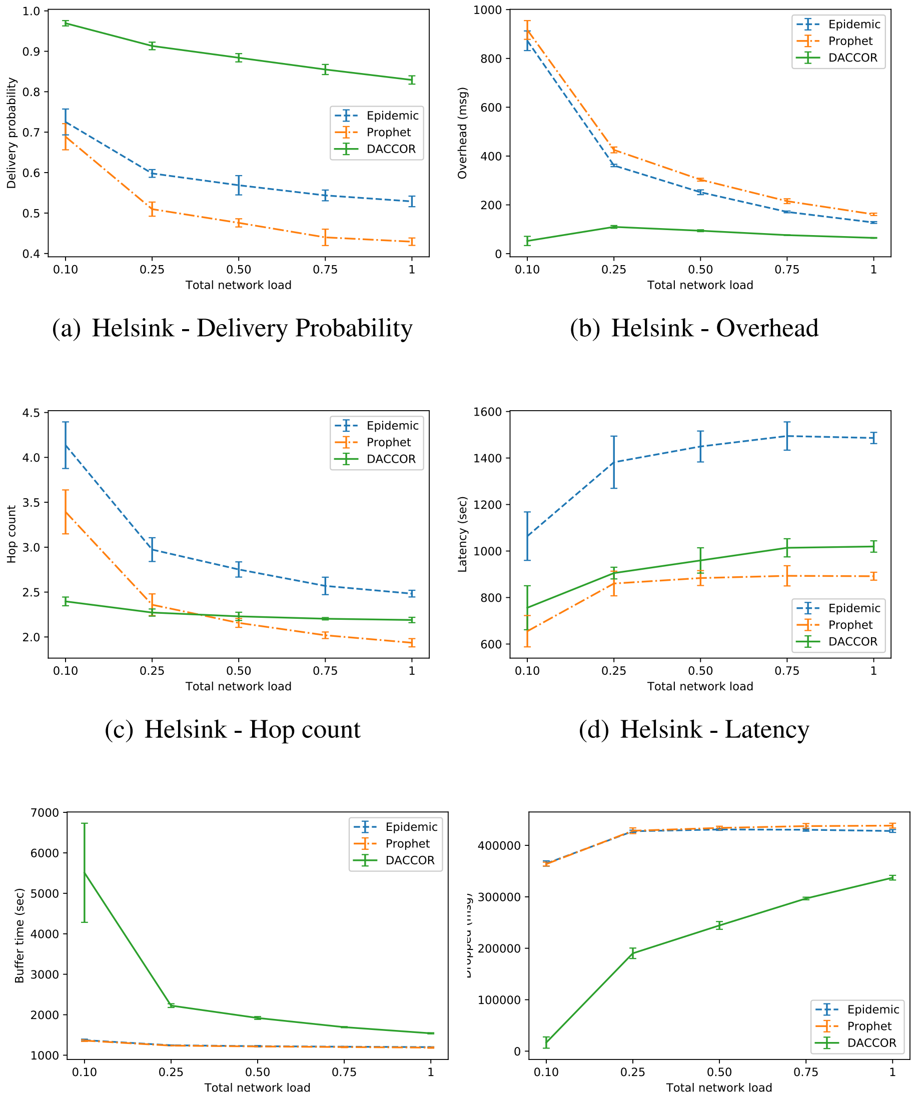
- *Delivery probability:* the probability of successfully delivering messages from source to the destination is computed as the total number of successfully delivered messages in the networks, divided by the total number of messages created.
- *Overhead:* the total number of relayed messages in the network, divided by the total number of successfully delivered message, i.e. the amount of transfers required to perform one successful delivery.
- *Latency:* the average elapsed time from the instant a message is generated to its successful delivery at the destination.
- *Hop count:* the average number of hops for each successful delivery.
- *Buffer time:* the average duration of time a message spend in a buffer.
- *Dropped Message:* number of messages dropped due to buffer overflow.

## 5.5 Results

Results are reported here for the Helsink, San Francisco and GeoLife mobility traces with a 95% confidence interval over 10 runs for each network load, giving a total of $10 \times 5 = 50$ simulation runs for each scenario. We randomize the traffic scenarios by varying the source and destination pairs of the flows in each of the 10 runs.

Figures 5.2, 5.4 and 5.3 show all the four defined metrics varying over the total network load (i.e., x-axis) for Helsink, GeoLife and San Francisco mobility scenarios. We can see from those figures that DACCOR achieves the highest delivery ratio with the lowest overhead and Hop count when compared to other protocols. The only exception is for

Geolife scenario considering a network load of 0.1. We argue that this reduced delivery ratio can be explained by the scenario's sparsity associated with the selective nature of DACCOR, which does not forward messages until it encounters a node with the same geographical preferences as the destination node. In fact for this scenario, where there are few messages to be forwarded and a low contact number between nodes, the best performing protocol is Epidemic. However, even though Epidemic achieves best delivery probabilities for low load, it costs more unnecessary transmissions and hops, dramatically increasing battery consumption of mobile devices and at the expense of network bandwidth, as shown in Figures 5.4(b) and 5.4(c), respectively.



(a) Helsink - Delivery Probability

(b) Helsink - Overhead

(c) Helsink - Hop count

(d) Helsink - Latency

(e) Helsink - Buffer time

(f) Helsink - Dropped

Figure 5.2: Performance evaluation for buffer size of 1GB and no TTL for DACCOR, Prophet, and Epidemic protocols under Helsink scenario.

It should be noted that the overhead metric considers only the messages delivered to calculate the number of messages replicated on the network. Therefore, we can observe that as the number of messages dropped on the network increases (Figures 5.2(f), 5.3(f), and 5.4(f)), the overhead on the network decreases (5.2(b), 5.3(b), and 5.4(b)). In other words, since there are fewer messages delivered on the network, there is also a smaller value of relayed messages counted in the overhead metric as the load increases. Table 5.3 helps to visualize the message overhead in the San Francisco trace scenario once it is not possible to view the message overhead values graphically. It is due to the large disparity between the DACCOR overhead and the other protocols.

Figures 5.2(e), 5.3(e), and 5.4(e) show the buffer time for messages that were not delivered to the network due to buffer overflow. The message discard policy is FIFO. We note that DACCOR has the longest buffer time, especially for low load, as it is more selective and therefore tends to buffer messages longer. We can argue that in the case of the epidemic protocol, which forwards messages at each encounter, when the node buffer fills the buffer time approaches the meeting time between nodes.

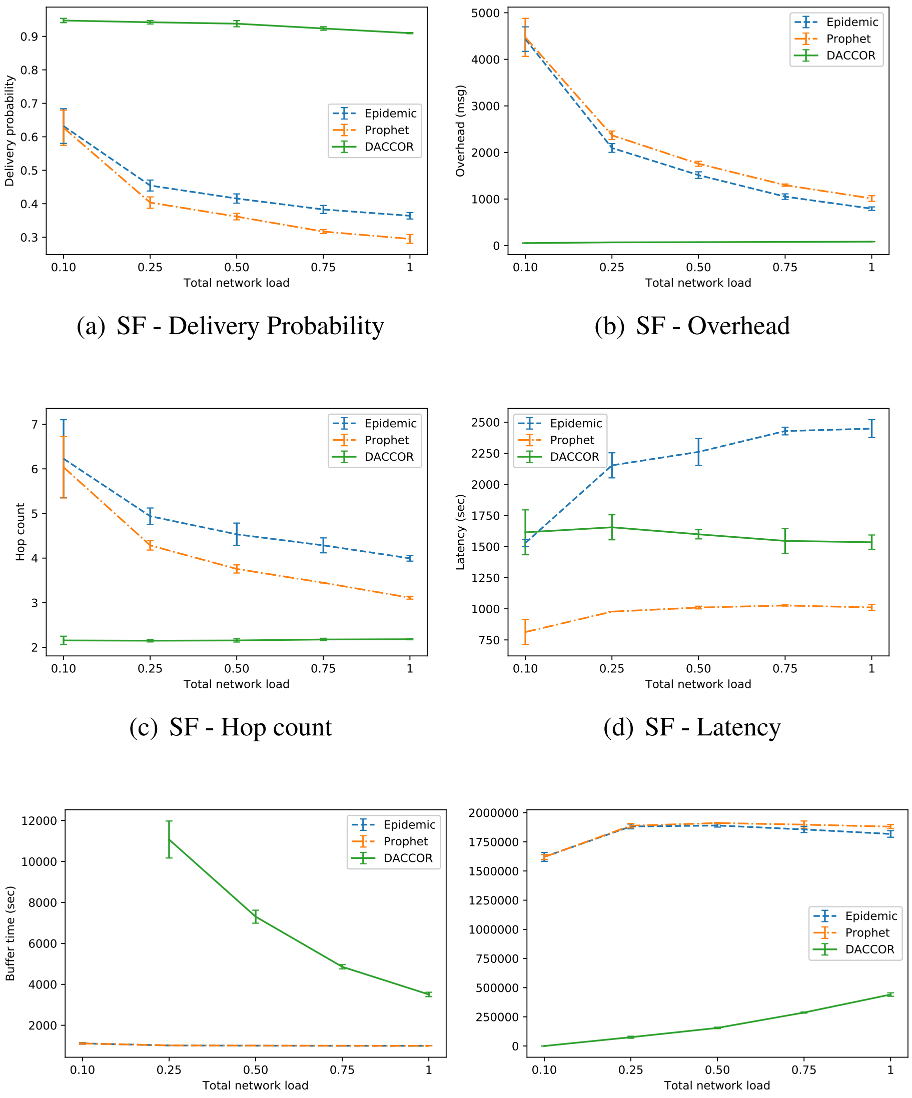
It is worth noting that Figures 5.3(e), and 5.4(e) do not have values for buffer time up to a load of 0.25 for DACCOR protocol. This is because DACCOR protocol does not reach the buffer limit, so there is no message loss until this message generation rate.

A possible downside for DACCOR due to its selective behavior when forwarding messages is the average latency for message delivery. However, the protocol has clear advantages over other metrics.

| **Load/**<br>Protocol | **0.1**<br>$\mu(\sigma)$ | **0.25**<br>$\mu(\sigma)$ | **0.50**<br>$\mu(\sigma)$ | **0.75**<br>$\mu(\sigma)$ | **1**<br>$\mu(\sigma)$ |
|---|---|---|---|---|---|
| Epidemic | 4434 (263) | 2095 (94) | 1512 (71) | 1051 (58) | 792 (38) |
| Prophet | 4470 (408) | 2368 (91) | 1756 (52) | 1297 (24) | 1011 (59) |
| DACCOR | 52.56 (5.09) | 68.65 (3.98) | 72.92 (0.96) | 78.13 (1.35) | 84.02 (2.82) |

Table 5.3: Average values ($\mu$) and standard deviation ($\sigma$) for the overhead metric for the San Francisco trace.

Results presented in this section confirm the efficiency of introducing geographical preference in the design of community based routing schemes. It is also worth to emphasize the extremely reduced energy fingerprint of DACCOR, as observed by the network overhead. With an overhead that is orders of magnitude lower than the other protocols, DACCOR activates the devices radio much less, saving energy and increasing battery life.



(a) SF - Delivery Probability

(b) SF - Overhead

(c) SF - Hop count

(d) SF - Latency

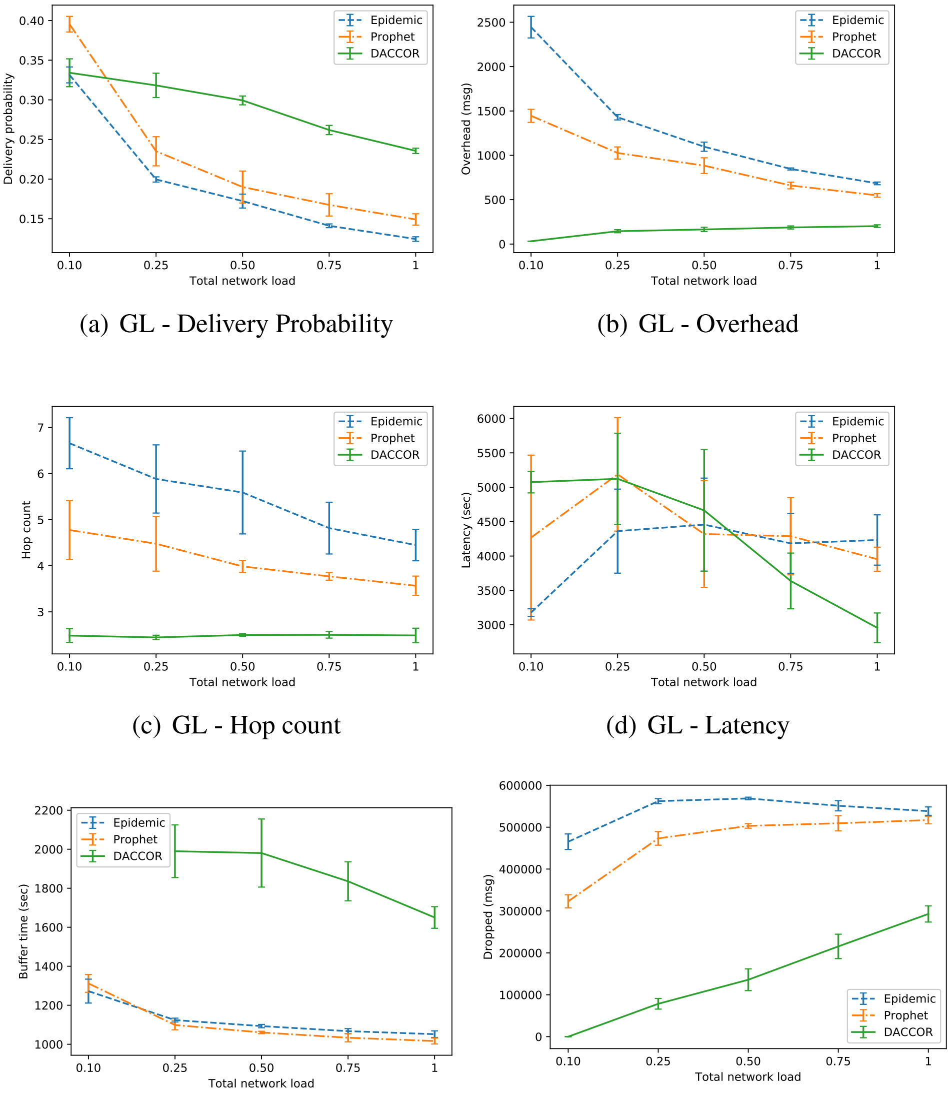
(e) SF - Buffer time

(f) SF - Dropped

Figure 5.3: Performance evaluation for DACCOR, Prophet, and Epidemic protocols under San Francisco scenario.

## 5.6 Related Work

Several forwarding messages protocols on opportunistic networks, including DACCOR, use the concept of communities as a strategy for choosing the next hop. Thus, this section presents works related to opportunistic networks protocols and social structures and clustering to form communities.



(a) GL - Delivery Probability

(b) GL - Overhead

(c) GL - Hop count

(d) GL - Latency

(e) GL - Buffer time

(f) GL - Dropped

Figure 5.4: Performance evaluation for DACCOR, Prophet, and Epidemic protocols under GeoLife scenario.

### 5.6.1 Opportunistic Network Routing Protocols

In opportunistic networks, best forwarding nodes are chosen based on chances (utility of node) they have to delivery a message to their destination. Next, a strategy to forward message to relay nodes with high utility and lower cost has to be taken.

Some protocols use social relationship information for selecting the best relay node [Yuan et al. 2016, Alajeely et al. 2017, Li and Wu 2009, Chuah and Coman 2009]. The first community-based proposed was BUBBLE [Hui et al. 2008], that uses the well known centrality metric and community structure to forward data. More recent works, such as [Xia et al. 2015, Yuan et al. 2014, Nguyen and Giordano 2012] to name a few, still proposing community based protocols to forwarding data. [Nguyen and Giordano 2012] uses relationship information containing node's profile (such as name, address, workplace, hobbies, etc.) to calculate the probability with destination.

In [Vangelis Angelakis and Yuan 2012] the relay node is selected in the neighbourhood based on the highest probability to reach the destination. The probability is calculated, using information that the sender knows about the destination, based on the behavior of repeating patterns at different times during day, week, and month. Authors in [Gao et al. 2014] proposed effective schemes that consider the existence of other relays carrying replicas of the same message in the network. The schemes eliminate this redundancy with some global network information. The authors found an interesting result, they observed that some messages replicas contribute little on improving the delivery ratio. Also, they show that forwarding performance after redundancy elimination was improved by 20%.

The impact of less popular nodes on the diffusion of messages on the network was studied in [Yuan et al. 2014]. More specifically, the authors removed nodes that were less "important" (low centrality) and found that message delivery performance was degraded.

Hui and others [Hui et al. 2007] proposed a distributed detection scheme for Pocket Switched Networks, where each device senses and detects its own community by analyzing the mobile device history it encountered. Just encounter events are used to build social relationship between them. These works use data obtained from opportunistic networks traces, which only contains information of the meetings between the mobile devices. More recently, [Chen and Lou 2016] proposed an expected encounter based routing protocol that makes the routing decision by comparing the minimum expected meeting delay to the destination. Besides, they proposed a community aware routing protocol using the expected number of encountering communities. The paper studies how the failures of some nodes in opportunistic networks can affect the performance of social-based forwarding strategies. These nodes can fail due to energy exhaustion, or intermittent connectivity where they can be out of communication range. It was shown that the non participation of only some important nodes can significantly degrade the performance of the entire network. The authors concludes that the community-based forwarding and routing methods in DTNs are really sensitive to the change of network communities.

### 5.6.2 Social Structures and Clustering

In a community aware opportunistic networks, nodes are divided into several communities according to their relationships. Social network consist of nodes connected by socially meaningful relationships. [Girvan and Newman 2002, Newman 2004] propose schemes to extract social relationships between users and also their social communities. Some works use data mining to detect interests in certain geographic areas by users. In [Khetarpaul et al. 2011], the authors propose a method to analyze users 'aggregate GPS location and extract users' location interests and rank them. In [Zheng et al. 2009a] authors also use GPS user trajectories to mine location interests and travel sequences. Authors in [Giannotti et al. 2007] mining similar sequences of user trajectories to find patterns of trajectories and regions of interest, applying different methods for pattern extraction. However, these papers do not consider the social characteristics of users and, therefore, do not address the problem of user clustering.

Eagle and others [Eagle and (Sandy) Pentland 2006] seek to recognize social patterns in the daily activity of users using traces generated from mobile devices (100 users using Bluetooth). This work explores mobility profile and user behavior to propose a methodology for community identification based on the similarities found among different users. However, they use data from Bluetooth encounters and the user location is inferred by the cell tower locations, thus losing the granularity of moving nodes.

Authors in [Ferrari et al. 2011] extract social networking patterns based on users' location in New York City, using the Twitter application. [Tang et al. 2012] proposes a method for extracting similarities among users of different social networks, in order to group them into communities. However, social networks provide information about user location or interests with high granularity, since information is only recorded when users actually use the social network. For example, uploading images in Instagram, or check-in using Foursquare.

A mathematical model to study communities in social networks is proposed in [Marbach 2016]. Authors assume that there is a population of agents who are interested in obtaining different types of content. The communities are formed in order to maximize their utility for obtaining and producing content. However, as stated by the author, the model fails to capture some properties of information communities that have been observed in practice.

As previously seen in this section, nodes tend to group into cluster structures. This behavior is dictated by their mobility behavior and geographical preferences, which we contend is also related to their community structure, assuming that similar individuals present similar behavior and also belong to the same community. Since nodes in the same community tend to have a high frequency of encounters, the process of finding the best relay node and a good routing strategies can be improved by taking this social structure into account. Also the control messages overhead as well as the node storage table can be reduced if we consider only the group information instead of the individual.

## 5.7 Conclusions

In this chapter we introduced DACCOR, a Deep AutoenCoder Community based Opportunistic Routing protocol. We hypothesize that users that have similar geographical preferences have also similar interests and as such we used a deep autoencoder to preprocess raw mobility datasets. This autoencoder approach was able to more accurately uncover community structures which identifies groups of users sharing common geographical interests and temporal relationships.

The proposed protocol used the community information and user relationship to make an efficient next hop selection decisions. Through extensive experimentation using one synthetic and two real mobility records representing diverse urban mobility scenarios we show the effectiveness of the proposed opportunistic protocol.

Our results show that the proposed deep autoencoder community based routing protocol lead to an improvement of the performance of the studied network metrics, i.e. delivery probability, overhead ratio, hop count, latency and dropped messages when compared with Epidemic and Prophet routing protocols. Finally, we show that DACCOR is able to outperform other opportunistic forwarding protocols, not only on the networks metrics, but also in the fact that, by using less bandwidth and less radio, DACCOR is able to dramatically decrease energy consumption, optimizing battery life.

# 6. Conclusion and Future Work

The contributions of this thesis can be divided into three parts: (1) a study on the Scale-Free Properties of Human Mobility, (2) the identification of User Communities Based on Geographical Preferences and Its Applications to Urban and Environmental Planning, and (3) the proposal of a Routing Protocol and Data Dissemination scheme for Opportunistic Networking. In the following, we summarize these contributions, providing also some possible research perspectives of each of the carried studies.

## 6.1Scale-Free Properties of Human Mobility

In this thesis, we started by showing the scale-free properties of some important human mobility characteristics, namely *spatial node density* and *mobility degree*. In our study we analyzed a set of real mobility traces collected in diverse scenarios motivated by ITS, namely a city park, a University campus, and taxis in the downtown area of a major city. We demonstrated that both *spatial node density* and *mobility degree* exhibit power law behavior which then allowed us to derive analytical models for these two mobility features. We showed that the proposed analytical model closely matches the empirical data extracted from the real mobility traces. Another contribution of our work was to use the proposed analytical models for *spatial node density* and *mobility degree* to build a waypoint-based mobility regime capable of generating synthetic mobility traces whose *spatial node density* and *mobility degree* closely resembles the ones measured in real human mobility scenarios. As such, the proposed mobility regime can be employed to test and evaluate ITS services and protocols. Finally, using a network simulator, we evaluated a wireless ad-hoc network routing protocol and showed that its performance under our mobility regime and under the real trace is very similar.

SFSM model has the ability to express analytically the behavior of users to tending to congregate and form clusters, where some regions may be quite dense while others completely deserted. Another interesting behavior found by SFSM is that there are few nodes that have very high mobility visiting many places in the trace, while the majority of users have a sedentary behavior, that is, they visit only few places.

In this work we presented simulation scenarios that demonstrate the independence of our model from real trace parameters and showed how one can use the proposed mobility model to evaluate their proposals. However, we did not evaluate the proposed mobility model in other applications. Mobile communication systems extensively use mobility models for predicting future user location. Also, mobility models are crucial to evaluate and test smart city applications using vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) in both real testbed, and simulated scenarios. Those aspects are becoming highly relevant, considering the growing of vehicles with V2V communication capabilities, sensors and smartphones. Also, the network simulations used only one real scenario (Quinta trace) and as future work, other scenarios could be considered.

## 6.2Identifying User Communities Based on Geographical Preferences and Its Applications to Urban and Environmental Planning

Our findings on the power law behavior of user density and mobility degree motivated further investigations, as it has considerable impact on fundamental network properties such as connectivity and capacity. How to design a practical and effective forwarding strategy in opportunistic networks? In order to answer that, we applied the principles of SFSM to identify communities based on the real behavior described by our proposed analytical model. Our proposed methodology identifies user communities based on users' geographical preferences and mobility attributes, such as speed and pause time. We showed that the proposed methodology is able to identify similarities and dissimilarities between users belonging to the same and different communities respectively, comparing four clustering algorithms.

We further improved our methodology by proposing a deep autoencoder-based approach to pre-process raw mobility datasets that is able to more accurately uncover community structures which identifies groups of users sharing common geographical interests and temporal relationships. The proposed methodology uses deep autoencoders for dimensionality reduction and extraction of latent non-linear representations of the mobility data, and then clusters the coded representation of mobility data, given by the autoencoders. Through extensive experimentation using three real mobility records, representing diverse urban mobility scenarios, we showed the effectiveness of the proposed autoencoder-based methodology. Our results show that automatically extracted features lead to an improvement of the performance of spatial similarity metrics while increasing contact time for users in the same community from 30% up to 150%. Moreover, the proposed approach reduces the complexity of the feature design task.

According to the results presented in this thesis, we notice that users belonging to the same community spend more time in certain geographic regions, which increases the probability of same community user meeting, when compared to users of different communities. We therefore expect that users belonging to the same community are potential opportunistic message forwarders. Also from the observation of the obtained results, users from different communities visit different geographic regions, although there is some intersection in some points. Since nodes in the same community tend to have a high frequency of encounters, the process of finding the best relay node and a good routing strategies can be improved by taking this social structure into account. Moreover, the control messages overhead as well as the node storage table can be reduced if we consider only the group information instead of the individual.

However, once mobility traces do not normally (or very rarely) present community labels or ground truth, we validated our proposal by using several index metrics (SSIM, MSE, and ARI), visualization (PCA and t-SNE) and the contact time between nodes belonging to the same community. As such, as a proposal for future work, we could use social traces and associate it with mobility traces to look for correlations between the communities extracted by geographical location and the relationships found in social networks.

### 6.3Routing Protocol and Data Dissemination for Opportunistic Networking

Finally, we proposed a forwarding protocol that can assist in the choice of relay nodes considering the contact opportunities between nodes in opportunistic networks. DAC-COR, a deep autoencoder community based routing protocol used the community information and user relationship to make efficient next hop selection decisions. Through extensive experimentation using one synthetic and two real mobility records representing diverse urban mobility scenarios we show the effectiveness of the proposed opportunistic protocol. Our results showed that the proposed deep autoencoder community based routing protocol lead to an improvement of the performance of the studied network metrics, i.e. delivery probability, overhead ratio, hop count, latency and dropped messages when compared with Epidemic and Prophet routing protocols.

Many message dissemination schemes have been proposed in the literature, however it is still challenging to disseminate messages to a target area in some environments. For example, in the context of urban transportation management, nodes can gather and locally processes content and then delivery it to a server in the cloud. Servers can be responsible for processing real time traffic data to take action when needed, e.g. incident response. In order to forward data, these nodes can rely on opportunistic connectivity with other various nodes (mobile devices, vehicles, smart-phones, etc.) and communicate with the application running in the cloud via multi-hop. Thus, as future investigation, a message dissemination scheme could be design to take advantage of the node degree behavior identified in SFSM and better select forwarding nodes to spread the message in a given urban area.

Also, the current design of the protocol only considers unicast message delivery. There may be environments where multicast message delivery is required. As future work, it will be interesting to explore the capability of DACCOR in delivering messages for a groups of nodes. Finally, we investigated DACCOR considering a pre-define autoencoder architecture and a specified number of communities. In the future, we could explore the impact of vary this hyper-parameters on evaluating the protocol.

# Bibliography

[JAI 2010] (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651 – 666. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

[201 40oj] (http://data.europa.eu/eli/dir/2010/40/oj). Directive 2010/40/eu of the european parliament and of the council of 7 july 2010 on the framework for the deployment of intelligent transport systems.

[Alajeely et al. 2017] Alajeely, M., Doss, R., and Ahmad, A. (2017). Routing protocols in opportunistic networks: A survey. *IETE Technical Review*, 0(0):1–19.

[Alavi et al. 2018] Alavi, A. H., Jiao, P., Buttlar, W. G., and Lajnef, N. (2018). Internet of things-enabled smart cities: State-of-the-art and future trends. *Measurement*, 129:589 – 606.

[Albino et al. 2015] Albino, V., Berardi, U., and Dangelico, R. (2015). Smart cities: Definitions, dimensions, performance, and initiatives. *Journal of Urban Technology*, 22(01):3–21.

[Aljalbout et al. 2018] Aljalbout, E., Golkov, V., Siddiqui, Y., and Cremers, D. (2018). Clustering with deep learning: Taxonomy and new methods. *CoRR*, abs/1801.07648.

[Antoniou et al. 2017] Antoniou, A., Storkey, A., and Edwards, H. (2017). Data Augmentation Generative Adversarial Networks. *arXiv e-prints*.

[Barabási and Albert 1999] Barabási, A.-L. and Albert, R. (1999). Emergence of scaling in random networks. *Science*, 286(5439):509–512.

[Bastani et al. 2011] Bastani, F., Huang, Y., Xie, X., and Powell, J. W. (2011). A greener transportation mode: Flexible routes discovery from gps trajectory data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, GIS ’11, pages 405–408, New York, NY, USA. ACM.

[Behrendt 2016] Behrendt, F. (2016). Why cycling matters for smart cities. internet of bicycles for intelligent transport. *Journal of Transport Geography*, 56:157 – 164.

[Bengio et al. 2012] Bengio, Y., Courville, A. C., and Vincent, P. (2012). Unsupervised feature learning and deep learning: A review and new perspectives. *CoRR, abs/1206.5538*, 1:2012.

[Berzin et al. 2014] Berzin, C., Latour, A., and León, J. (2014). *Inference on the Hurst Parameter and the Variance of Diffusions Driven by Fractional Brownian Motion*. Springer International Publishing.

[Bettstetter et al. 2003] Bettstetter, C., Resta, G., and Santi, P. (2003). The node distribution of the random waypoint mobility model for wireless ad hoc networks. *Mobile Computing, IEEE Transactions on*, 2(3):257 – 269.

[Bishop and Nasrabadi 2007] Bishop, C. M. and Nasrabadi, N. M. (2007). Pattern Recognition and Machine Learning. *Journal of Electronic Imaging*, 16(4).

[Bocconi et al. 2015] Bocconi, S., Bozzon, A., Psyllidis, A., Titos Bolivar, C., and Houben, G.-J. (2015). Social glass: A platform for urban analytics and decision-making through heterogeneous social data. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15 Companion, pages 175–178.

[Boeing 2017] Boeing, G. (2017). The structure and dynamics of cities: Urban data analysis and theoretical modeling, by marc barthelemy. *Journal of the American Planning Association*, 83(4):418–418.

[Boldrini and Passarella 2010] Boldrini, C. and Passarella, A. (2010). Hcmm: Modelling spatial and temporal properties of human mobility driven by users' social relationships. *Computer Communications*, 33(9):1056 – 1074.

[Bora and Gupta 2014] Bora, D. J. and Gupta, A. K. (2014). A comparative study between fuzzy clustering algorithm and hard clustering algorithm. *CoRR*, abs/1404.6059.

[Borrel et al. 2005] Borrel, V., de Amorim, M. D., and Fdida, S. (2005). On natural mobility models. In *WAC*.

[Botta et al. 2014] Botta, A., de Donato, W., Persico, V., and Pescapé, A. (2014). On the integration of cloud computing and internet of things. In *2014 International Conference on Future Internet of Things and Cloud*, pages 23–30.

[Bridges et al. 2000] Bridges, S. M., Vaughn, R. B., Professor, A., and Professor, A. (2000). Fuzzy data mining and genetic algorithms applied to intrusion detection. In *In*

*Proceedings of the National Information Systems Security Conference (NISSC*, pages 16–19.

[Calabrese et al. 2014] Calabrese, F., Ferrari, L., and Blondel, V. D. (2014). Urban sensing using mobile phone network data: A survey of research. *ACM Comput. Surv.*, 47(2):25:1–25:20.

[Campos et al. 2009] Campos, C., Azevedo, T., Bezerra, R., and de Moraes, L. (2009). An analysis of human mobility using real traces. In *Proceedings of the 2009 IEEE WCNC*.

[Canzian and Musolesi 2015] Canzian, L. and Musolesi, M. (2015). Trajectories of depression: Unobtrusive monitoring of depressive states by means of smartphone mobility traces analysis. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, UbiComp '15, pages 1293–1304.

[Cebeci and Yildiz 2015] Cebeci, Z. and Yildiz, F. (2015). Comparison of k-means and fuzzy c-means algorithms on different cluster structures. *JOURNAL OF AGRICULTURAL INFORMATICS*, pages 13–23.

[Chakraborty et al. 2017] Chakraborty, T., Dalmia, A., Mukherjee, A., and Ganguly, N. (2017). Metrics for community analysis: A survey. *ACM Comput. Surv.*, 50(4):54:1–54:37.

[Champernowne 1953] Champernowne, D. (1953). A model for income distribution. *Economic Journal*, 63:318–351.

[Charte et al. 2018] Charte, D., Charte, F., García, S., del Jesus, M. J., and Herrera, F. (2018). A practical tutorial on autoencoders for nonlinear feature fusion: Taxonomy, models, software and guidelines. *Information Fusion*, 44:78 – 96.

[Che et al. 2018] Che, Z., Purushotham, S., Cho, K., Sontag, D., and Liu, Y. (2018). Recurrent neural networks for multivariate time series with missing values. *Scientific reports*, 8(1):6085.

[Chen and Lou 2016] Chen, H. and Lou, W. (2016). Contact expectation based routing for delay tolerant networks. *Ad Hoc Networks*, 36(Part 1):244 – 257.

[Cho et al. 2014] Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734.

[Chris Fraley 2002] Chris Fraley, A. E. R. (2002). Model-based clustering, discriminant analysis, and density estimation. *Journal of the American Statistical Association*, 97(458):611–631.

[Chuah and Coman 2009] Chuah, M. and Coman, A. (2009). Identifying connectors and communities: Understanding their impacts on the performance of a dtn publish/subscribe system. In *2009 International Conference on Computational Science and Engineering*, volume 4, pages 1093–1098.

[Clauset et al. 2009] Clauset, A., Shalizi, C. R., and Newman, M. E. J. (2009). Power-law distributions in empirical data. *SIAM Rev.*, 51(4):661–703.

[Conti and Giordano 2014] Conti, M. and Giordano, S. (2014). Mobile ad hoc networking: milestones, challenges, and new research directions. *IEEE Communications Magazine*, 52(1):85–96.

[CRAWDAD 2015] CRAWDAD (P?gina visitada em 2015). http://crawdad.cs.dartmouth.edu/.

[Dasgupta and Raftery 1995] Dasgupta, A. and Raftery, A. E. (1995). Detecting Features in Spatial Point Processes with Clutter via Model-Based Clustering. *Journal of the Americ. Stat. Association*, 93:294–302.

[Datta et al. 2017] Datta, S. K., Haerri, J., Bonnet, C., and Costa, R. F. D. (2017). Vehicles as connected resources: Opportunities and challenges for the future. *IEEE Vehicular Technology Magazine*, 12(2):26–35.

[Domenico et al. 2013] Domenico, M. D., Lima, A., and Musolesi, M. (2013). Interdependence and predictability of human mobility and social interactions. *Pervasive and Mobile Computing*, 9(6):798 – 807.

[Dong et al. 2013] Dong, W., Duffield, N., Ge, Z., Lee, S., and Pang, J. (2013). Modeling cellular user mobility using a leap graph. In Roughan, M. and Chang, R., editors, *Passive and Active Measurement*, pages 53–62, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Duda et al. 2001] Duda, R., Hart, P., and Stork, D. (2001). *Pattern Classification*. Wiley.

[Eagle and Pentland 2006] Eagle, N. and Pentland, A. S. (2006). Reality mining: sensing complex social systems. *Personal and ubiquitous computing*, 10(4):255–268.

[Eagle and (Sandy) Pentland 2006] Eagle, N. and (Sandy) Pentland, A. (2006). Reality mining: Sensing complex social systems. *Personal Ubiquitous Comput.*, 10(4):255–268.

[Ferrari et al. 2011] Ferrari, L., Rosi, A., Mamei, M., and Zambonelli, F. (2011). Extracting urban patterns from location-based social networks. In *of the 3rd ACM SIGSPATIAL*, pages 9–16, Chicago, Illinois.

[Ferreira et al. 2019a] Ferreira, D. L., CAMPOS, C. A. V., and Obraczka, K. (2019a). Deep autoencoder based community detection and its application to data forwarding in opportunistic networks. *IEEE MASS 2019 : The 16th IEEE International Conference on Mobile Ad-Hoc and Smart Systems (under work).*

[Ferreira et al. 2019b] Ferreira, D. L., Nunes, B. A. A., , CAMPOS, C. A. V., and Obraczka, K. (2019b). A deep learning approach for identifying user communities based on geographical preferences and its applications to urban and environmental planning. *Special Issue on Deep Learning For Spatial Algorithms and Systems, ACM Transactions on Spatial Algorithms and Systems (submitted).*

[Ferreira et al. 2019c] Ferreira, D. L., Nunes, B. A. A., , CAMPOS, C. A. V., and Obraczka, K. (2019c). Using real mobility records for user community identification in smart cities. *IEEE Transactions on Intelligent Transportation Systems (submitted).*

[Ferreira et al. 2016] Ferreira, D. L., Nunes, B. A. A., and CAMPOS, C. A. V. (2016). Uma metodologia de identificação de estruturas sociais em registros reais de mobilidade humana e veicular. *WORKSHOP DE REDES P2P, DINÂMICAS, SOCIAIS E ORIENTADAS A CONTEÚDO (WP2P+),XXXIV Simpósio Brasileiro de Redes de Computadores.*

[Ferreira et al. 2018] Ferreira, D. L., Nunes, B. A. A., and Obraczka, K. (2018). Scale-free properties of human mobility and applications to intelligent transportation systems. *IEEE Transactions on Intelligent Transportation Systems*, 19(11):3736–3748.

[Gao et al. 2014] Gao, W., Li, Q., and Cao, G. (2014). Forwarding redundancy in opportunistic mobile networks: Investigation and elimination. In *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pages 2301–2309.

[Giannotti et al. 2007] Giannotti, F., Nanni, M., Pinelli, F., and Pedreschi, D. (2007). Trajectory pattern mining. In *Proc of 13th ACM SIGKDD*, KDD '07, pages 330–339, San Jose, California, USA.

[Girvan and Newman 2002] Girvan, M. and Newman, M. E. J. (2002). Community structure in social and biological networks. *Proceedings of the National Academy of Sciences*, 99(12):7821–7826.

[Gonzalez et al. 2008] Gonzalez, M. C., Hidalgo, C. A., and Barabasi, A.-L. (2008). Understanding individual human mobility patterns. *nature*, 453(7196):779.

[Graves et al. 2007] Graves, A., Fernández, S., and Schmidhuber, J. (2007). Multi-dimensional recurrent neural networks. In de Sá, J. M., Alexandre, L. A., Duch, W., and Mandic, D., editors, *Artificial Neural Networks – ICANN 2007*, pages 549–558, Berlin, Heidelberg. Springer Berlin Heidelberg.

[Harfouche et al. 2010] Harfouche, L., Boumerdassi, S., and Renault, E. (2010). Weighted social manhattan: Modeling and performance analysis of a mobility model. In *IEEE 21st International Symposium on Personal Indoor and Mobile Radio Communications (PIMRC)*.

[Hasnat et al. 2015] Hasnat, M. A., Velcin, J., Bonnevay, S., and Jacques, J. (2015). Simultaneous clustering and model selection for multinomial distribution: A comparative study. In Fromont, E., De Bie, T., and van Leeuwen, M., editors, *Advances in Intelligent Data Analysis XIV*, pages 120–131, Cham. Springer International Publishing.

[Hess et al. 2015a] Hess, A., Hummel, K. A., Gansterer, W. N., and Haring, G. (2015a). Data-driven human mobility modeling: A survey and engineering guidance for mobile networking. *ACM Computing Surveys*, 48(3):38:1–38:39.

[Hess et al. 2015b] Hess, A., Hummel, K. A., Gansterer, W. N., and Haring, G. (2015b). Data-driven human mobility modeling: A survey and engineering guidance for mobile networking. *ACM Comput. Surv.*, 48(3):38:1–38:39.

[Hong et al. 2010] Hong, S., Lee, K., and Rhee, I. (2010). Step: A spatio-temporal mobility model for humans walks. In *The 7th IEEE International Conference on Mobile Ad-hoc and Sensor Systems (IEEE MASS 2010)*, pages 630–635.

[Hossmann et al. 2011] Hossmann, T., Spyropoulos, T., and Legendre, F. (2011). Putting contacts into context: Mobility modeling beyond inter-contact times. In *Proceedings of the Twelfth ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '11, pages 18:1–18:11, Paris, France. ACM.

[Hou et al. 2016] Hou, X., Li, Y., Jin, D., Wu, D. O., and Chen, S. (2016). Modeling the impact of mobility on the connectivity of vehicular networks in large-scale urban environments. *IEEE Transactions on Vehicular Technology*, 65(4):2753–2758.

[Hsu et al. 2009] Hsu, W. J., Spyropoulos, T., Psounis, K., and Helmy, A. (2009). Modeling spatial and temporal dependencies of user mobility in wireless mobile networks. *IEEE/ACM Transactions on Networking*, 17(5):1564–1577.

[Hui et al. 2008] Hui, P., Crowcroft, J., and Yoneki, E. (2008). Bubble rap: Social-based forwarding in delay tolerant networks. In *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, MobiHoc '08, pages 241–250, Hong Kong, Hong Kong, China. ACM.

[Hui et al. 2007] Hui, P., Yoneki, E., Chan, S. Y., and Crowcroft, J. (2007). Distributed community detection in delay tolerant networks. In *2Nd ACM/IEEE MobiArch'07*, pages 7:1–7:8, Kyoto, Japan.

[Hyytia et al. 2006] Hyytia, E., Lassila, P., and Virtamo, J. (2006). Spatial node distribution of the random waypoint mobility model with applications. *IEEE Transactions on Mobile Computing*, 5(6):680–694.

[Jaeger 2008] Jaeger, T. F. (2008). Categorical data analysis: Away from anovas (transformation or not) and towards logit mixed models. *Journal of Memory and Language*, 59(4):434 – 446. Special Issue: Emerging Data Analysis.

[Jahromi et al. 2016] Jahromi, K. K., Zignani, M., Gaito, S., and Rossi, G. P. (2016). Simulating human mobility patterns in urban areas. *Simulation Modelling Practice and Theory*, 62:137 – 156.

[Jain et al. 1999] Jain, A. K., Murty, M. N., and Flynn, P. J. (1999). Data clustering: A review.

[Jiang et al. 2017] Jiang, Z., Zheng, Y., Tan, H., Tang, B., and Zhou, H. (2017). Variational deep embedding: An unsupervised and generative approach to clustering. In *Proceedings of the 26th International Joint Conference on Artificial Intelligence*, IJCAI'17, pages 1965–1972, Melbourne, Australia.

[Karagiannis T 2010] Karagiannis T, Le Boudec JY, V. M. (2010). Power law and exponential decay of intercontact times between mobile devices. *IEEE Transactions on Mobile Computing*, 9(10):1377–1390.

[Karamshuk et al. 2011] Karamshuk, D., Boldrini, C., Conti, M., and Passarella, A. (2011). Human mobility models for opportunistic networks. *IEEE Communications Magazine*, 49(12):157–165.

[Karamshuk et al. 2014] Karamshuk, D., Boldrini, C., Conti, M., and Passarella, A. (2014). Spot: Representing the social, spatial, and temporal dimensions of human mobility with a unifying framework. *Pervasive and Mobile Computing*, 11(0):19 – 40.

[Keränen and Ott 2007] Keränen, A. and Ott, J. (2007). Increasing reality for dtn protocol simulations. *Helsinki University of Technology, Tech. Rep.*

[Keränen et al. 2009] Keränen, A., Ott, J., and Kärkkäinen, T. (2009). The ONE Simulator for DTN Protocol Evaluation. In *SIMUTools '09: Proceedings of the 2nd International Conference on Simulation Tools and Techniques*, New York, NY, USA. ICST.

[Khetarpaul et al. 2011] Khetarpaul, S., Chauhan, R., Gupta, S. K., Subramaniam, L. V., and Nambiar, U. (2011). Mining gps data to determine interesting locations. In *Proc. of the 8th IIWeb'11*, pages 8:1–8:6, Hyderabad, India.

[Khoroshevsky and Lerner 2016] Khoroshevsky, F. and Lerner, B. (2016). Human mobility-pattern discovery and next-place prediction from gps data. In *IAPR workshop on multimodal pattern recognition of social signals in human-computer interaction*, pages 24–35. Springer.

[Kosta et al. 2014] Kosta, S., Mei, A., and Stefa, J. (2014). Large-scale synthetic social mobile networks with swim. *IEEE Transactions on Mobile Computing*, 13(1):116–129.

[Kotz et al. 2009] Kotz, D., Henderson, T., Abyzov, I., and Yeo, J. (2009). CRAWDAD data set dartmouth/campus (v. 2009-09-09). Downloaded from http://crawdad.cs.dartmouth.edu/dartmouth/campus.

[Krizhevsky et al. 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. In *Advances in Neural Information Processing Systems 25*, pages 1097–1105.

[Langley 1998] Langley, R. B. (1998). The utm grid system. *GPS world*, 9(2):46–50.

[LeCun et al. 2015] LeCun, Y., Bengio, Y., and Hinton, G. (2015). Deep learning. *Nature*, 521:436 EP –.

[Lee et al. 2009] Lee, K., Hong, S., Kim, S. J., Rhee, I., and Chong, S. (2009). Slaw: A mobility model for human walks. In *Proceedings of IEEE INFOCOM*.

[Lee et al. 2012] Lee, K., Hong, S., Kim, S. J., Rhee, I., and Chong, S. (2012). Slaw: Self-similar least-action human walk. *IEEE/ACM Trans. Netw.*, 20(2):515–529.

[Li and Wu 2009] Li, F. and Wu, J. (2009). Localcom: A community-based epidemic forwarding scheme in disruption-tolerant networks. In *6th IEEE SECON'09.*, pages 1–9.

[Lim et al. 2006] Lim, S., Yu, C., and Das, C. (2006). Clustered mobility model for scale-free wireless networks. In *Local Computer Networks, Proceedings 2006 31st IEEE Conference on*.

[Lim et al. 2010] Lim, S., Yu, C., and Das, C. R. (2010). A realistic mobility model for wireless networks of scale-free node connectivity. *IJMC*, 8(3):351–369.

[Lin and Hsu 2014] Lin, M. and Hsu, W.-J. (2014). Mining gps data for mobility patterns: A survey. *Pervasive and Mobile Computing*, 12(Supplement C):1 – 16.

[Lindgren et al. 2003] Lindgren, A., Doria, A., and Schelén, O. (2003). Probabilistic routing in intermittently connected networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 7(3):19–20.

[Liu et al. 2017] Liu, J., Sun, L., Li, Q., Ming, J., Liu, Y., and Xiong, H. (2017). Functional zone based hierarchical demand prediction for bike system expansion. In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, pages 957–966, Halifax, NS, Canada. ACM.

[Liu et al. 2010] Liu, N., Liu, M., Cao, J., Chen, G., and Lou, W. (2010). When transportation meets communication: V2p over vanets. In *2010 IEEE 30th International Conference on Distributed Computing Systems*, pages 567–576.

[Liu et al. 2016] Liu, W., Wang, Z., Liu, X., Zeng, N., Liu, Y., and E. Alsaadi, F. (2016). A survey of deep neural network architectures and their applications. *Neurocomputing*, 234.

[Long et al. 2015] Long, J., Shelhamer, E., and Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440.

[Louf and Barthelemy 2014] Louf, R. and Barthelemy, M. (2014). How congestion shapes cities: from mobility patterns to scaling. *Scientific reports*, 4(5561).

[Mahanti et al. 2013] Mahanti, A., Carlsson, N., Mahanti, A., Arlitt, M., and Williamson, C. (2013). A tale of the tails: Power-laws in internet measurements. *IEEE Network*, 27(1):59–64.

[Marbach 2016] Marbach, P. (2016). The structure of communities in information networks. pages 1–6.

[McAuley and Leskovec 2012] McAuley, J. and Leskovec, J. (2012). Learning to discover social circles in ego networks. In *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, NIPS'12, pages 539–547, Lake Tahoe, Nevada. Curran Associates Inc.

[Mehrotra and Musolesi 2018] Mehrotra, A. and Musolesi, M. (2018). Using autoencoders to automatically extract mobility features for predicting depressive states. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 2(3):127.

[Min et al. 2018] Min, E., Guo, X., Liu, Q., Zhang, G., Cui, J., and Long, J. (2018). A survey of clustering with deep learning: From the perspective of network architecture. *IEEE ACCESS*, 6:39501–39514.

[Mitsche et al. 2014] Mitsche, D., Resta, G., and Santi, P. (2014). The random waypoint mobility model with uniform node spatial distribution. *Wireless networks*, 20(5):1053–1066.

[Mota et al. 2014] Mota, V. F., Cunha, F. D., Macedo, D. F., Nogueira, J. M., and Loureiro, A. A. (2014). Protocols, mobility models and tools in opportunistic networks: A survey. *Computer Communications*, 48(Supplement C):5 – 19.

[Motani et al. 2005] Motani, M., Srinivasan, V., and Nuggehalli, P. S. (2005). Peoplenet: Engineering a wireless virtual social network. In *Proc. of the 11th MobiCom'05*, pages 243–257, Cologne, Germany.

[Munjal et al. 2011] Munjal, A., Camp, T., and Navidi, W. C. (2011). Smooth: A simple way to model human mobility. In *Proceedings of the 14th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, MSWiM '11, pages 351–360, New York, NY, USA. ACM.

[Murtagh and Legendre 2014] Murtagh, F. and Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: Which algorithms implement ward's criterion? *Journal of Classification*, 31(3):274–295.

[Nair et al. 2013] Nair, R., Miller-Hooks, E., Hampshire, R. C., and Busic, A. (2013). Large-scale vehicle sharing systems: Analysis of vélib'. *International Journal of Sustainable Transportation*, 7(1):85–106.

[Newman 2004] Newman, M. (2004). Detecting community structure in networks. *The European Physical Journal B - Condensed Matter and Complex Systems*, 38(2):321–330.

[Newman 2005] Newman, M. E. (2005). Power laws, pareto distributions and zipf's law. *Contemporary physics*, 46(5):323–351.

[Nguyen et al. 2017] Nguyen, C.-B., Yoon, S., and Kim, J. (2017). Discovering social community structures based on human mobility traces. *Mobile Information Systems*, 2017:17 pages.

[Nguyen and Giordano 2012] Nguyen, H. A. and Giordano, S. (2012). Context information prediction for social-based routing in opportunistic networks. *Ad Hoc Networks*, 10(8):1557 – 1569. Special Issue on Social-Based Routing in Mobile and Delay-Tolerant Networks.

[Niu et al. 2016] Niu, H., Liu, J., Fu, Y., Liu, Y., and Lang, B. (2016). Exploiting human mobility patterns for gas station site selection. In Navathe, S. B., Wu, W., Shekhar, S., Du, X., Wang, X. S., and Xiong, H., editors, *Database Systems for Advanced Applications*, pages 242–257, Cham. Springer International Publishing.

[Noulas et al. 2011] Noulas, A., Scellato, S., Lambiotte, R., Pontil, M., and Mascolo, C. (2011). A tale of many cities: universal patterns in human urban mobility. *CoRR*, abs/1108.5355.

[Noulas et al. 2012] Noulas, A., Scellato, S., Lambiotte, R., Pontil, M., and Mascolo, C. (2012). A tale of many cities: Universal patterns in human urban mobility. *PLoS ONE*, 7(5):e37027.

[Nunes and Obraczka 2011] Nunes, B. A. A. and Obraczka, K. (2011). On the invariance of spatial node density for realistic mobility modeling. MASS '11, pages 322–331.

[Nunes and Obraczka 2014] Nunes, B. A. A. and Obraczka, K. (2014). A framework for modeling spatial node density in waypoint-based mobility. *Wireless Networks*, 20(4):775–786.

[Pathak et al. 2017] Pathak, S., Gondaliya, N., and Raja, N. (2017). A survey on prophet based routing protocol in delay tolerant network. In *2017 International Conference on Emerging Trends Innovation in ICT (ICEI)*, pages 110–115.

[Perkins et al. 2003] Perkins, C., Belding-Royer, E., and Das, S. (2003). Ad hoc on-demand distance vector (aodv) routing.

[Phithakkitnukoon et al. 2012] Phithakkitnukoon, S., Smoreda, Z., and Olivier, P. (2012). Socio-geography of human mobility: a study using longitudinal mobile phone data. *PLoS ONE*, 7(6):e39253.

[Piorkowski et al. 2009] Piorkowski, M., Sarafijanovic-Djukic, N., and Grossglauser, M. (2009). CRAWDAD data set epfl/mobility (v. 2009-02-24). Downloaded from http://crawdad.cs.dartmouth.edu/epfl/mobility.

[Reed et al. 2016] Reed, S. E., Akata, Z., Yan, X., Logeswaran, L., Schiele, B., and Lee, H. (2016). Generative adversarial text to image synthesis. *CoRR*, abs/1605.05396.

[Rhee et al. 2011] Rhee, I., Shin, M., Hong, S., Lee, K., Kim, S. J., and Chong, S. (2011). On the levy-walk nature of human mobility. *IEEE/ACM Trans. Netw.*, 19(3):630–643.

[Ribeiro et al. 2012] Ribeiro, A., Sofia, R., and Zuquete, A. (2012). Modeling pause time in social mobility models. In *Wireless Communication Systems (ISWCS), 2012 International Symposium on*, pages 656–660.

[Rocha et al. 2007] Rocha, M., Cortez, P., and Neves, J. (2007). Evolution of neural networks for classification and regression. *Neurocomputing*, 70(16):2809 – 2816.

[Ronneberger et al. 2015] Ronneberger, O., Fischer, P., and Brox, T. (2015). U-net: Convolutional networks for biomedical image segmentation. In *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, pages 234–241, Cham.

[Sadiq et al. 2018] Sadiq, B. O., Adedokun, A. E., and Abubakar, Z. M. (2018). The impact of mobility model in the optimal placement of sensor nodes in wireless body sensor network. *CoRR*, abs/1801.01435.

[Santos and Embrechts 2009] Santos, J. M. and Embrechts, M. (2009). On the use of the adjusted rand index as a metric for evaluating supervised classification. In *Proceedings of the 19th International Conference on Artificial Neural Networks: Part II*, ICANN '09, pages 175–184, Berlin, Heidelberg. Springer-Verlag.

[Scalable Network Technologies ] Scalable Network Technologies. Qualnet 4.0.

[Senaratne et al. 2018] Senaratne, H., Mueller, M., Behrisch, M., Lalanne, F., Bustos-Jimenez, J., Schneidewind, J., Keim, D., and Schreck, T. (2018). *Urban Mobility Analysis With Mobile Network Data: A Visual Analytics Approach*, volume 19.

[Siła-Nowicka et al. 2016] Siła-Nowicka, K., Vandrol, J., Oshan, T., Long, J. A., Demšar, U., and Fotheringham, A. S. (2016). Analysis of human mobility patterns from gps trajectories and contextual information. *International Journal of Geographical Information Science*, 30(5):881–906.

[Silveira et al. 2016] Silveira, L. M., de Almeida, J. M., Marques-Neto, H. T., Sarraute, C., and Ziviani, A. (2016). Mobhet: Predicting human mobility using heterogeneous data sources. *Computer Communications*, 95:54 – 68. Mobile Traffic Analytics.

[Song et al. 2010a] Song, C., Koren, T., Wang, P., and Barabasi, A.-L. (2010a). Modelling the scaling properties of human mobility. *Nat Phys*, 6(10):818–823.

[Song et al. 2010b] Song, C., Qu, Z., Blumm, N., and Barabási, A.-L. (2010b). Limits of predictability in human mobility. *Science*, 327(5968):1018–1021.

[Sze et al. 2017] Sze, V., Chen, Y., Yang, T., and Emer, J. S. (2017). Efficient processing of deep neural networks: A tutorial and survey. *Proceedings of the IEEE*, 105(12):2295–2329.

[Tang et al. 2012] Tang, L., Wang, X., and Liu, H. (2012). Community detection via heterogeneous interaction analysis. *Data Mining and Knowledge Discovery*, 25(1):1–33.

[The Scenario Generator ] The Scenario Generator. http://isis.poly.edu/ qiming/scengen/index.html.

[Toch et al. 2019] Toch, E., Lerner, B., Ben-Zion, E., and Ben-Gal, I. (2019). Analyzing large-scale human mobility data: a survey of machine learning methods and applications. *Knowledge and Information Systems*, 58(3):501–523.

[Vahdat et al. 2000] Vahdat, A., Becker, D., et al. (2000). Epidemic routing for partially connected ad hoc networks.

[Van Der Maaten 2014] Van Der Maaten, L. (2014). Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res.*, 15(1):3221–3245.

[Vangelis Angelakis and Yuan 2012] Vangelis Angelakis, N. G. and Yuan, D. (2012). Probabilistic routing in opportunistic ad hoc networks. *Wireless Ad-Hoc Networks*.

[Vastardis and Yang 2014] Vastardis, N. and Yang, K. (2014). An enhanced community-based mobility model for distributed mobile social networks. *Journal of Ambient Intelligence and Humanized Computing*, 5(1):65–75.

[Wang et al. 2017] Wang, K., Gou, C., Zheng, N., Rehg, J. M., and Wang, F.-Y. (2017). Parallel vision for perception and understanding of complex scenes: methods, framework, and perspectives. *Artificial Intelligence Review*, 48(3):299–329.

[Wang et al. 2004] Wang, Z., Bovik, A. C., Sheikh, H. R., and Simoncelli, E. P. (2004). Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612.

[Wetzel 2017] Wetzel, S. J. (2017). Unsupervised learning of phase transitions: From principal component analysis to variational autoencoders. *Phys. Rev. E*, 96:022140.

[Wikipedia contributors 2019] Wikipedia contributors (2019). Smart city Wikipedia, the free encyclopedia. https://en.wikipedia.org/w/index.php?title=Smartcityoldid=880818358. [Online; accessed 31-January-2019].

[Xia et al. 2015] Xia, F., Yang, Q., Li, J., Cao, J., Liu, L., and Ahmed, A. M. (2015). Data dissemination using interest-tree in socially aware networking. *Comput. Netw.*, 91(C):495–507.

[Xia J 2015] Xia J, Curtin KM, L. W. Z. Y. (2015). A new model for a carpool matching service. *PLoS ONE*, 10(6):46–50.

[Xu and Tian 2015] Xu, D. and Tian, Y. (2015). A comprehensive survey of clustering algorithms. *Annals of Data Science*, 2(2):165–193.

[Xu and Wunsch 2005] Xu, R. and Wunsch, II, D. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, 16(3):645–678.

[Yang et al. 2013] Yang, J., McAuley, J. J., and Leskovec, J. (2013). Community detection in networks with node attributes. *2013 IEEE 13th International Conference on Data Mining*, pages 1151–1156.

[Yoon et al. 2003] Yoon, J., Liu, M., and Noble, B. (2003). Random waypoint considered harmful. In *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, volume 2, pages 1312–1321 vol.2.

[Young et al. 2018] Young, T., Hazarika, D., Poria, S., and Cambria, E. (2018). Recent trends in deep learning based natural language processing. *ieee Computational intelligenCe magazine*, 13(3):55–75.

[Yu et al. 2017a] Yu, S., Jia, S., and Xu, C. (2017a). Convolutional neural networks for hyperspectral image classification. *Neurocomputing*, 219:88–98.

[Yu et al. 2017b] Yu, Y., Gong, Z., Zhong, P., and Shan, J. (2017b). Unsupervised representation learning with deep convolutional neural network for remote sensing images. In Zhao, Y., Kong, X., and Taubman, D., editors, *Image and Graphics*, pages 97–108, Cham. Springer International Publishing.

[Yuan et al. 2016] Yuan, P., Fan, L., Liu, P., and Tang, S. (2016). Recent progress in routing protocols of mobile opportunistic networks: A clear taxonomy, analysis and evaluation. *Journal of Network and Computer Applications*, 62:163 – 170.

[Yuan et al. 2014] Yuan, P., Liu, P., and Tang, S. (2014). Exploiting partial centrality of nodes for data forwarding in mobile opportunistic networks. In *2014 IEEE 17th International Conference on Computational Science and Engineering*, pages 1435–1442.

[Zheng 2015] Zheng, Y. (2015). Trajectory data mining: an overview. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 6(3):29.

[Zheng et al. 2014] Zheng, Y., Capra, L., Wolfson, O., and Yang, H. (2014). Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):38.

[Zheng et al. 2009a] Zheng, Y., Xie, X., and Ma, W.-Y. (2009a). Mining interesting locations and travel sequences from gps trajectories. In *ACM WWW 2009*. ACM WWW 2009.

[Zheng et al. 2010] Zheng, Y., Xie, X., and Ma, W.-Y. (2010). Geolife: A collaborative social networking service among user, location and trajectory. *IEEE Data(base) Engineering Bulletin*.

[Zheng et al. 2009b] Zheng, Y., Zhang, L., Xie, X., and Ma, W.-Y. (2009b). Mining interesting locations and travel sequences from gps trajectories. In *Proceedings of the 18th international conference on World wide web*, pages 791–800. ACM.

[Zhong et al. 2014] Zhong, C., Arisona, S. M., Huang, X., Batty, M., and Schmitt, G. (2014). Detecting the dynamics of urban structure through spatial network analysis. *Int. J. Geogr. Inf. Sci.*, 28(11):2178–2199.

[Zhong and Ghosh 2003] Zhong, S. and Ghosh, J. (2003). A unified framework for model-based clustering. *Journal of Machine Learning Research*, 4:1001–1037.

[Zhu et al. 2014] Zhu, W., Peng, W., Hung, C., Lei, P., and Chen, L. (2014). Exploring sequential probability tree for movement-based community discovery. *IEEE Transactions on Knowledge and Data Engineering*, 26(11):2717–2730.

[Ziat et al. 2017] Ziat, A., Delasalles, E., Denoyer, L., and Gallinari, P. (2017). Spatio-temporal neural networks for space-time series forecasting and relations discovery. In *2017 IEEE International Conference on Data Mining (ICDM)*, pages 705–714.