Extracting new urban patterns in cities: Analysis, Models and Applications

Hitham Assem

A thesis submitted in fulfillment of the requirments
for the degree of
Doctor of Philosophy

School of Computer Science and Statistics
Trinity College Dublin

March 2018

Declaration

I, the undersigned, declare that this thesis has not been submitted as an exercise for a degree at this or any other university and it is entirely my own work. I agree to deposit this thesis in the University's open access institutional repository or allow the Library to do so on my behalf, subject to Irish Copyright Legislation and Trinity College Library conditions of use and acknowledgement.

Hitham Assem
March, 2018

Extracting new urban patterns in cities: Analysis, Models and Applications

Hitham Assem

Summary

Smart city initiatives rely on real-time measurements and data collected by a large number of heterogenous physical sensors deployed throughout a city. The data gathered by physical sensors can capably identify important events in cities, but seldom explain the underlying reasons behind such events. In other words, physical sensors can identify what happens, but may be unable to explain why or how specific events occur or patterns emerge. The rise of Location-based Social Networks (LBSNs) has allowed millions of dwellers and visitors of a city to share their observations, thoughts, feelings, and experiences, or in other words, their perceptions about their city through social media updates. LBSNs data represents a treasure which is still under explored, especially as the added location dimension on social networks bridges the gap between the physical world and the digital online social network services, potentially leading to the emergence of new types of applications.

This thesis shows how the use of this powerful LBSNs data coupled with machine learning techniques, can lead to the extraction of new urban patterns in cities. The thesis starts by leveraging the power of Deep Learning and in particular, Deep Belief Networks, for extracting a new urban pattern which is called *Socio-demographic Regional Patterns*. It is shown for the first time that it is possible to extract a unique pattern for various regions in cities of very close spatial proximity. The five boroughs in New York City are considered in a case study with an emphasis on extracting a unique pattern for each of the boroughs. Second, a new approach is introduced that discovers functional regions that not only change across space but time as well. It is shown with the proposed approach that it is possible to extract different functionalities for the same physical regions during the day. This type of new urban pattern is called, *Temporal Functional Regions Patterns*. Next, a new approach for *Recognizing Recurrent Crowd Mobility Patterns* in cities is introduced for illustrating how crowd shifts across space and time with various crowd level intensities. In addition, it is shown that the correlation between the extracted crowd mobility patterns and the temporal functional regions patterns provides further new insights into the motivation behind crowd mobility. Finally, it is shown how some of these extracted patterns can brought together to solve a domain specific challenge (*Network Demand Prediction*). To do so, a new deep learning based-approach (titled ST-DenNetFus) is introduced for fusing some of the extracted urban patterns with network demand data achieving a higher level of accuracy on predicting the network demand across cities compared to without fusing these patterns.

Acknowledgment

I would like first to express my gratitude to my supervisor Prof. Declan O'Sullivan for the amazing support, guidance, encouragement and feedback throughout my PhD work. During these years, he has patiently monitored and directed my work, allowing for the expression and development of the good qualities I may possess. Subsequently, I would like to express my gratitude to my colleagues in IBM and in particular my team, the Innovation Exchange. Sandra Buda, Gavin Shorten, Faisal Ghaffar, Lei Xu, and Bora Caglayan, thank you for the very fruitful technical discussions we had in the office that in lots of cases inspired me with new ideas in my PhD. Nevertheless, I have been so lucky to have Robert Mccarthy as my manager in IBM during my PhD who has been of great support and encouragement to me since I started my PhD and through all my journey in IBM. I would like also to thank Pat O'Sullivan who encouraged and helped me with the decision of doing my PhD while working in IBM.

I would not have contemplated this journey if not for my father and mother, Ahmed Assem and Azza Enany, who supported me unconditionally during my long educational journey. My father is my role model, who not only taught me how to act as a professional but as well taught me how to plan, think, and take decisions throughout all of my life. He is a big fan of engineering and constrained optimization which is somehow linked to this thesis, I learned from him how to leverage the scientific theories we study in real life situations, these qualities helped me a lot in accomplishing my PhD. My mother supported me with her wisdom and unconditional love. I know one of her dreams was to see me a PhD holder and that was obvious from her patience and support. I want also to thank my sister and brother, Dina and Mohamed for their encouragement and continuous support. I hope I made all of my family proud of me.

Last, but not least, thanks to my beloved wife, Rana Maher. If it was not for her, I would not have been able to study this degree, not writing this Thesis either and most likely you would not have known me. During these years, she has been very supportive, encouraging and understanding. During my long working hours that mostly reached twenty hours in several days including most weekends during these years, she has been always of a great support. However, I understand that it was very hard for her not seeing me lots of consecutive days and sometimes weeks due to my long working hours but hopefully I made her proud of me.

To my parents Assem and Azza, my sister Dina, my brother Mohamed, and my beloved wife Rana, for their infinite support.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Motivation

Mobile devices and smartphones are considered the most rapidly growing technologies in the world [1]. Market analysis estimates 1.75 billion smart-phones by 2014 in a world where location-based technologies is typically equipped in such devices [2]. An interesting survey in relation to smartphone usage in the USA shows as of February 2012 that 74% of smartphone owners utilize location based services such as Twitter or Foursquare on their phones. This is up from 55% in May 2011 indicating that the rise of the overall proportion of US adults who get location based information has almost been doubled over a short time frame [3].

In this way, the emergence the smartphone industry has led to the significant rise of what is called Location-based Social Networks (LBSNs) which empower people to share their activity related choices using their social networks (e.g., Facebook, Twitter, Foursquare). LBSNs lead to the emergence of a type of uncontrolled experimental context resulting in datasets with novel characteristics. This is in direct contrast with other previous methods used by scientists to collect datasets that can be useful for extracting urban patterns across cities: population survey methods (exploited typically by urbanists) have been of high economic cost and rather static in recording the temporal dimension of the extracted patterns; while sensor based methods instrumented by computer scientists in recent years could only be deployed and utilized by small number of participants and for a finite period of time. Furthermore, datasets which describe urban mobility owned by large telecom providers are very valuable and have become only sporadically to scientists due to privacy and economic concerns.

Although previous mobility data featured geographical coordinates of users, LBSNs come with fundamentally different attributes [4]: First, LBSNs not only report the geographical coordinates of the user but also identify the venues where users check-in such as restaurants, outdoor activities, or a stadium. In other words, it has the power to correlate the location of the user along with her activity. In addition, these broadcasts contain semantically rich information such as tips, comments or recommendations on the venues visited by the users. Finally, the scale of LBSNs data is all based on the user level participation which can itself gives some cultural, socio-demographic and behavioural insights within different cities [5].

LBSNs data represents a treasure which is still under explored especially as the added location dimension bridges the gap between the physical world and the digital online social network services. This data has stimulated the research community into identifying new human-generated patterns in cities that can find a natural application for not only predicting events and providing novel recommenders that facilitate users' choices and social interactions but can also help in discovering social commonalities among people [6]. In addition, by coupling real time social systems like Twitter[1], Facebook[2], and Google Buzz[3] with location sharing services like Foursquare[4], Gowalla[5] and Google Latitude[6], we can foresee an un-precedented access to activities, actions and footprints of millions of people [5]. This has the potential for deeper insights and better geospatial understanding of cities' unique characteristics and the collective consciousness of the people who reside, work and play within different regions in cities [7]. The research community is exploring the potential of harnessing the power of such data and its impact on different domain areas including urban planning [8], marketing [9][10], urban energy [11][12], and economy [13].

The motivation of the work in this thesis is inspired by four major shifts in thinking. First, gathering and analysing longer time duration of LBSNs datasets compared to previous state-of-the-art research work could yield the discovery of new urban patterns that were not possible before. This could potentially help in overcoming the challenge of the "the unreasonable effectiveness of data" [14]. Second, with the continuous growth of the number of users using location-based services, the sparsity challenge for inferring new insights about urban patterns is possibly lessened. Third, a shift of focus towards crowd behavioural analysis rather than user-centric behaviour may support novel classes of applications as those that will be discussed in this thesis. Fourth, the recent advancements in the data mining and machine learning such as deep learning techniques and topic models could potentially help in extracting new and finer urban patterns across cities.

[1] https://twitter.com/

[2] https://www.facebook.com/

[3] https://en.wikipedia.org/wiki/Google_Buzz

[4] https://foursquare.com/

[5] https://en.wikipedia.org/wiki/Gowalla

[6] https://google-latitude.en.softonic.com/web-apps

1.2 Thesis and its substantiation

1.2.1 Research Question

The main research question posed in this thesis is:

To what extent can new urban patterns be extracted from LBSNs data and be used in a new machine learning architecture that fuses diverse data sources to solve a spatio-temporal time series forecasting problem (Network Demand Prediction)?

In this thesis, *urban pattern* is defined as "a recurrent pattern in an urban environment that can be computed if the spatio-temporal feature of individuals' mobility data in cities extracted from distinct traces of higher level information (such as people commonalities, recurrent behaviours, or inference on upcoming events) is leveraged".

1.2.2 Research Objective and Technical Approach

As has been discussed in the previous section, geo-location data and especially LBSNs constitute a novel and online platform for capturing human mobility data associated with human activities through textual data (tweets in case of Twitter) or venue categories (check-ins venues in case of Foursqaure). They provide the ability to study what the individual is doing, when and where, as well as having no definitive end or number of participants being required. Further, the multiple layers of data that concurrently exist in these systems create a new ecosystem of information with promising implications for discovering new urban patterns that potentially can be useful in various applications.

Consequently, the argument outlined in this thesis is that the use of geo-location data over a longer time duration, and especially data from LBSNs, coupled with the utilization of advancements in machine learning will result in the extraction of new urban patterns. It is argued that these patterns can be of direct benefit to the development of effective applications and services, or can be of indirect benefit by utilizing these patterns as an external factors for solving particular domain problems, such as Network Demand Prediction.

The following objectives summarize the research of this thesis:

1. To identify novel urban patterns that can be extracted using LBSNs data.

2. To establish whether the correlation of various extracted urban patterns could bring further insights.

3. To establish in what way machine learning approaches would help extract the urban patterns.

4. To demonstrate the utility of the extracted urban patterns in solving a spatio-temporal time series forecasting problem (Network Demand Prediction).

The following points represent a step-by-step approach that has been followed to address the objectives. Figure 1.1 summarizes the followed approach.

1. **Investigate the state-of-the-art:** This theoretical investigation involved reviewing various machine learning approaches with a focus on its recent advancements as well as reviewing the previous work in extracting urban patterns. This step served as a basis for understanding what type of machine learning models could help in extracting new urban patterns.

2. **Identifying potential new urban patterns to extract:** The availability of LBSNs data (with longer time span and finer granularity) opened the possibility for new patterns to be extracted. This step involved identifying the potential intuitive new patterns that could be extracted using LBSNs data. For instance, the state-of-the-art showed several works in extracting static functional regions (functionality of regions that does not change across time), an intuitive new potential pattern in this example could be the ability to extract functional regions that change across time (someone could think of a region's functionality as a "Business" district in the morning, "Eating" in the afternoon, and "Night-life" at night).

3. **Dataset gathering and exploratory analysis:** This step involved gathering the LBSNs dataset and performing some exploratory statistical analysis for better understanding the dataset properties, this includes the frequency of tweets by users, the dataset's density analysis and the applications used.

4. **Extracting the identified new urban patterns:** In this step and for each of the identified potential new urban patterns, the following steps are carried out:

   a) **Develop machine learning algorithms for extracting the identified new urban patterns:** This step focused on researching and developing the machine learning models for extracting the potential patterns identified in step 2 using the dataset gathered in step 3.

   b) **Evaluate the extracted patterns:** This step involved evaluating and validating the extracted patterns. In some cases, subjective methods are used through a case study to try matching the extracted patterns to our understanding of the areas within cities. In other cases, objective methods are employed using well-known evaluation metrics for estimating the accuracy of the proposed machine learning models.

5. **Identify a potential domain problem that could benefit indirectly from the extracted patterns:** This involved researching what type of problems in a specific domain that indirectly could intuitively benefit from the extracted urban patterns. For instance, someone could think that the functionality of regions as well as crowd mobility patterns could be related to the network demand variation across a city.

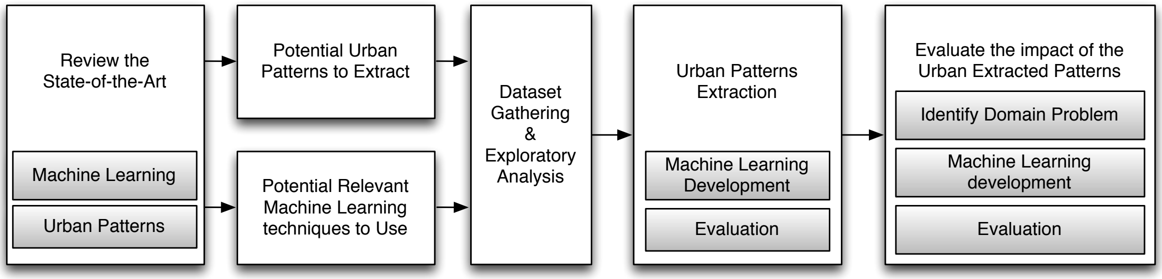

Review the State-of-the-Art
Machine Learning
Urban Patterns
Potential Urban Patterns to Extract
Potential Relevant Machine Learning techniques to Use
Dataset Gathering & Exploratory Analysis
Urban Patterns Extraction
Machine Learning Development
Evaluation
Evaluate the impact of the Urban Extracted Patterns
Identify Domain Problem
Machine Learning development
Evaluation

Figure 1.1: Technical Approach.

6. **Establish a machine learning architecture for spatio-temporal prediction problems:** This involved researching and developing a machine learning architecture for fusing some of the extracted urban patterns with another domain specific dataset in one architecture for showing the impact of the extracted urban patterns for the identified domain specific problem from step 5.

7. **Evaluate the impact of fusing the extracted patterns on solving the identified domain problem:** This involved objective quantitative evaluation for testing the accuracy of the developed machine learning architecture using well-known metrics that measures the accuracy of time series forecasting techniques.

1.3 Contributions & Publications

1.3.1 Contributions

The core contribution of this thesis is two-fold. *First, I extracted new urban patterns from LBSNs in ways that reveal the common attributes of users' behaviour across them. Through this thread of research, I have been able to extract three new urban patterns including: Socio-demographic Regional Patterns, Temporal Functional Regions, and Recurrent Crowd Mobility Patterns. In addition the last two patterns have been correlated together for deriving deeper insights into the motivation behind crowd mobility. Second, I have shown for the first time that some of these extracted urban patterns if fused as external data sources with network data have an impact on improving the accuracy of the Network Demand Prediction problem, which is crucial challenge in the Telecommunications service provider domain.* The following highlights the core contributions described in this thesis:

1. First, a new type of urban pattern (which is referred to in this thesis as "Socio-demographic Regional Patterns") has been identified and extracted using Deep-Belief-Networks. This unique pattern can be extracted for regions even if these regions are located within the same city. *To the best of our knowledge, there has been no previous research that shows the ability to extract a unique pattern of regions and moreover,*

Deep-Belief-Networks have not been leveraged previously for extracting urban patterns in cities.

2. Second, a new type of urban pattern called "Temporal Functional Regions" is introduced to recognize regions' functionalities that change across space and time and it is shown how clustering based techniques can be used to extract this pattern. *Extracting the functionalities of regions has been temporally static in the state-of-the-art which means that it has not been shown previously that it is possible to extract various functionalities for the same region based on the time of the day.*

3. Third, a new urban pattern called "Recurrent Crowd Mobility Patterns" is introduced and a new approach for recognizing these patterns across cities that highlights the level of crowdedness has been developed. Furthermore, it is shown that correlating these mobility patterns with the Temporal Functional Regions provides insights into the motivation behind crowd mobility. *Through our research, it is demonstrated that the proposed approach outperforms other baselines used in the state-of-the-art. Furthermore and for the first time, this correlation has the potential to provide new insights about cities, in that understanding the motivation behind such crowd shifts has the potential to empower several key city management applications, such as traffic management, urban planning, and public safety.*

4. Finally, it is demonstrated how the prior two extracted patterns (Temporal functional regions and Recurrent Crowd Mobility Patterns) could be of benefit to one of the challenges in the Telecommunications service provider domain, the Network Demand Prediction problem. The reason for choosing this particular problem is three-fold: First, intuitively someone could think that the Temporal Functional Regions as well as the Crowd Mobility patterns across space and time could impact the network usage patterns. Second, the Network Demand Prediction problem on itself is quite complicated problem for the telco operators due to the various external factors that could impact the network usage patterns [15]. Third, being able to predict more accurately the network demand has a quite positive impact on the telco operators as they could be able to allocate network resources adaptively according to the predicted demand rather than over-provisioning network resources which is a highly costly approach [16]. *In this thesis, a new deep learning based approach is introduced titled as ST-DenNetFus demonstrating for the first time how to fuse some of the extracted urban patterns with network data for increasing the accuracy of predicting network demand. ST-DenNetFus has the ability to fuse various external data sources of different dimensionalities which argued that other spatio-temporal prediction problems could leverage same approach.*

1.3.2 Publications

During my PhD studies, I have been involved in many fruitful collaborations that have yielded to 11 published works that span the areas of urban mobility, location-based social networks, topic modelling, urban activity, neighbourhood modelling and cognitive network management. In relation to this thesis, chapter 3 and in particular the machine learning TCDC-based recommenders is based on the work in [SmartCity 2015] [PUC 2016]. Sandra Buda and Lei Xu provided support on the design of the experiments and assisted in the writing of the paper. I carried out the design and implementation of the proposed TCDC framework for both the regression and classification models. Chapter 5 builds on [SIGSPATIAL 2016] in which I carried out the design, analysis, implementation and evaluation of these works as well as writing the paper. Chapter 6 builds on [ICTAI 2016] in which I carried out the design, analysis, implementation and evaluation of these works, whereas the co-authors contributed to the writing of the paper and provided support on refining technical aspects of the methodologies exploited around the clustering techniques employed. Chapter 7 is based on the work in [TIST 2017] where I designed the proposed framework and developed the algorithms and baselines whereas the co-authors helped in formalizing the algorithms and generating some map visualizations. Chapter 8 is based on the work in [TKDE 2018] in which I designed and developed the deep learning architecture while the coauthors helped in setting-up, conducting some experiments on GPUs for speeding up the training time and writing some sections in the paper.

Papers related to this thesis

1. Assem, H. and O'Sullivan, D., 2015, December. Towards bridging the gap between machine learning researchers and practitioners. In Smart City/SocialCom/SustainCom (SmartCity), 2015 IEEE International Conference on (pp. 702-708). IEEE.

2. Assem, H., Xu, L., Buda, T.S. and O'Sullivan, D., 2016. Machine learning as a service for enabling Internet of Things and People. Personal and Ubiquitous Computing, 20(6), pp.899-914.

3. Assem, H., Xu, L., Buda, T.S. and O'Sullivan, D., 2016, November. Spatio-temporal clustering approach for detecting functional regions in cities. In Tools with Artificial Intelligence (ICTAI), 2016 IEEE 28th International Conference on (pp. 370-377). IEEE.

4. Assem, H. and O'Sullivan, D., 2016, October. Discovering New Socio-demographic Regional Patterns in Cities. In Proceedings of the 9th ACM SIGSPATIAL Workshop on Location-based Social Networks (p. 1). ACM.

5. Assem, H., Buda, T.S. and O'Sullivan, D., 2017, August. "RCMC: Recognizing Crowd Mobility Patterns in Cities based on Location Based Social Networks Data." ACM Transactions on Intelligent Systems and Technology (TIST). ACM.

6. Assem, H., Xu, L., Buda, T.S. and O'Sullivan, D., 2016 "Cognitive Architecture and its applications in Smarter Cities." Springer Book Chapter.

7. Assem, H., Caglayan, B., Buda, T.S. and O'Sullivan, D., 2018, "ST-DenNetFus: A New Spatio-Temporal DenseNet-based architecture for Network Demand Prediction." ACM Transactions on Knowledge and Data Engineering (TKDE). ACM. (Submitted).

Other works during PhD study

1. Xu, L., Assem, H., Yahia, I.G.B., Buda, T.S., Martin, A., Gallico, D., Biancani, M., Pastor, A., Aranda, P.A., Smirnov, M. and Raz, D., 2016, June. CogNet: A network management architecture featuring cognitive capabilities. In Networks and Communications (EuCNC), 2016 European Conference on (pp. 325-329). IEEE.

2. Velez, G., Quartulli, M., Martin, A., Otaegui, O. and Assem, H., 2016, June. Machine Learning for Autonomic Network Management in a Connected Cars Scenario. In International Workshop on Communication Technologies for Vehicles (pp. 111-120). Springer International Publishing.

3. Buda, T.S., Assem, H., Xu, L., Raz, D., Margolin, U., Rosensweig, E., Lopez, D.R., Corici, M.I., Smirnov, M., Mullins, R. and Uryupina, O., 2016, April. Can machine learning aid in delivering new use cases and scenarios in 5G?. In Network Operations and Management Symposium (NOMS), 2016 IEEE/IFIP (pp. 1279-1284). IEEE.

4. Buda, T. S., Assem, H., and Xu, L. (2017, May). ADE: An ensemble approach for early Anomaly Detection. In Integrated Network and Service Management (IM), 2017 IFIP/IEEE Symposium on (pp. 442-448). IEEE.

1.4 Chapters outline

The thesis starts with some background in chapter 2, presenting an overview of the evolution of the different types of spatio-temporal datasets through to the growth of LBSNs. Further, the properties of LBSNs that make such datasets quite unique are discussed. In addition, the most important research findings investigated by scientists in the study of extracting urban patterns are presented. Chapter 3 introduces the various paradigms of machine learning with a focus on the techniques that will be further used in the core chapters to follows. Through this introduction, a new approach called **TCDC** is introduced which could be utilized for recommending the optimum supervised machine learning model. In chapter 4, an overview description, analysis and statistics on the LBSNs dataset is illustrated that will be used in the core chapters to follow.

The rest of the thesis and the core contributions are organized in the following way. Each of chapters 5, 6, 7, 8 are generally organized in subsections following the general pattern of motivation, state-of-the-art, what has been developed, and how it has been evaluated.

- In chapter 5, a new model for discovering new unique weekly urban patterns for various regions within a city is presented. The proposed model is entitled as **Socio-demographic Regional Pattern model**. In particular, the power of deep learning is leveraged for forming a complex automated feature hierarchy and **Deep-Belief-Nets** is employed to identify these unique weekly region-footprints patterns that have not been possible to discover before. Through the experiments, it has been demonstrated that it is feasible to discover unique patterns for each of the boroughs in New York City with nearly 70% accuracy. Furthermore, the existence and complexity of the extracted patterns (as well as to gain a better understanding of these unique patterns) is validated by applying **Latent Dirichlet Allocation (LDA)**.

- In chapter 6, a new approach is introduced for discovering for the first time **Temporal Functional Regions patterns, showing that the functionality of regions can vary temporally during the day compared to the prior state-of-the-art work that mostly focused on discovering static spatial functional regions that does not take the temporal factor into account**. Different time intervals for discovering the Temporal Functional Regions were studied and it is concluded that 2 and 4 Hours time intervals seem the most reasonable granularity for discovering Temporal Functional Regions. The resultant Temporal Functional Regions generated from the 4 Hours time interval, illustrating the morning, afternoon, evening and night functional regions, are analyzed and visualised. Furthermore, the results are analyzed subjectively by mapping some of the functional regions generated from the proposed approach to our common understanding of these regions' features.

- In chapter 7, a new approach is introduced that is capable of extracting **Recurrent Crowd Mobility Patterns** with an estimation of three levels of crowd intensity utilizing an approach combining **Kernel Density Estimation (KDE) and Non-negative Matrix Factorization (NMF)**. The extracted Recurrent Crowd Mobility Patterns show how crowd shifts from one area to another during each day across various time-slots. A detailed analysis is presented on the extracted crowd patterns with an exploratory visualization showing that the proposed approach can identify obvious mobility patterns that recur over time and space in the urban scenario outperforming other three baseline methods. Using the same time interval, it is further shown that **correlating the Temporal Functional Regions described in chapter 6 with the identified Recurrent Crowd Mobility Patterns** can yield a deeper understanding of the city dynamics and can produce insights around the motivation behind crowd mobility.

- In chapter 8, a new deep-learning based approach called **ST-DenNetFus** is proposed, to collectively forecast two types of telecommunications network throughput (uplink and downlink) in each and every region of a city. The designed architecture for ST-DenNetFus is based on unique properties of spatio-temporal data. More specifically, dense network neural network is employed to model the temporal closeness, period, and trend properties for the network demand. For each property, a branch of dense convolutional units is designed, each of which models the spatial properties of the network demand. **This aggregation is further combined with external factors including the two extracted patterns discussed in the previous two chapters showing their positive impact on the accuracy for the Network Demand Prediction problem which is very important in the telecommunications service provider domain**. An extensive experimental evaluation is further presented where it is found that ST-DenNetFus outperformed five well-known baselines.

Finally, in chapter 9, the main findings and directions for future work in extracting urban patterns research and its potential applications are summarized. Figure 1.2 summarizes the core contributions mapped to the chapters in the thesis.

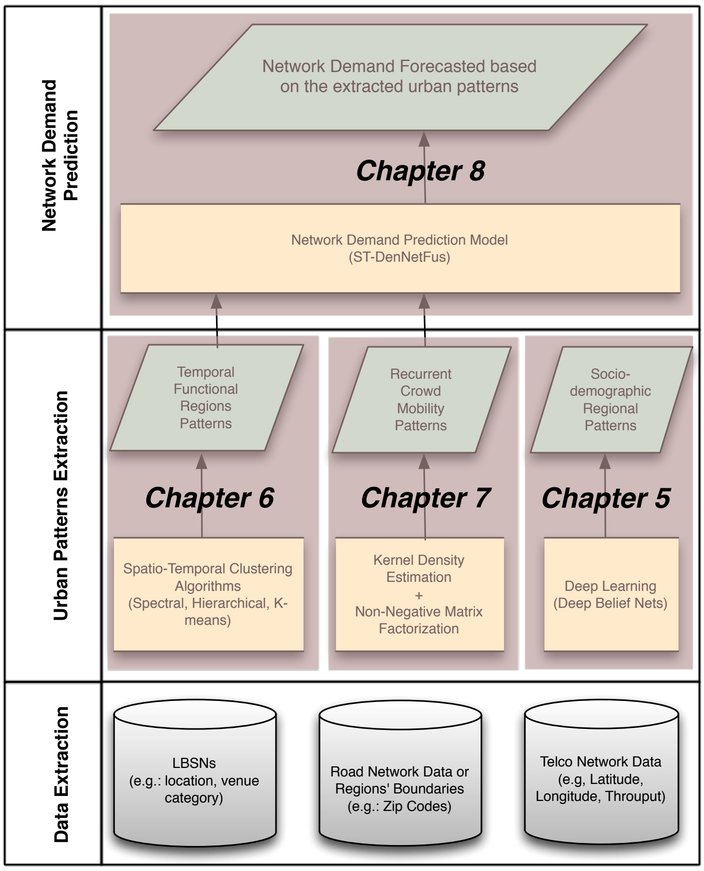

Network Demand Prediction
Network Demand Forecasted based on the extracted urban patterns
Chapter 8
Network Demand Prediction Model (ST-DenNetFus)
Urban Patterns Extraction
Temporal Functional Regions Patterns
Recurrent Crowd Mobility Patterns
Socio-demographic Regional Patterns
Chapter 6
Chapter 7
Chapter 5
Spatio-Temporal Clustering Algorithms (Spectral, Hierarchical, K-means)
Kernel Density Estimation + Non-Negative Matrix Factorization
Deep Learning (Deep Belief Nets)
Data Extraction
LBSNs (e.g.: location, venue category)
Road Network Data or Regions' Boundaries (e.g.: Zip Codes)
Telco Network Data (e.g, Latitude, Longitude, Throuput)

Figure 1.2: Core contributions mapped to chapters.

Chapter 2

Background and State-of-the-Art

Chapter overview: *This chapter gives a historic background of the spatio-temporal datasets that can be used for extracting urban patterns starting from survey-based methods, cellular networks data, until the rise of the location-based social networks. Further, it highlights the four main properties of location-based social networks that make this data source most unique: spatial accuracy, publicly availability, global accessibility, and venue categorization. Finally, an overview of the state-of-the-art for extracting urban patterns is discussed highlighting the techniques used and some interesting outcomes.*

2.1 Background - An historic perspective upon urban spatio-temporal datasets

In this section, a brief overview is provided on the spatio-temporal datasets that can be used for extracting urban patterns in cities. The first type of datasets were gathered using survey based methods and initiated empirical research work for describing human migration patterns. The second type of spatio-temporal data gathered have been from the communication interactions observed in cellular networks, with this type of data leading to important breakthroughs regarding extracting urban patterns due to the scale of acquiring records from large populations. Although these types of data were found to be very powerful in supporting analysis of urban patterns, the accessibility of these types of data have been found to be quite challenging and expensive by the research community.

2.1.1 Survey-based methods and census data

The first modern attempt for understanding and extracting some meaning from urban patterns took place in 1885 when E.G. Ravenstein published his work called "The Laws of Migration" [17] in the Journal of the Statistical Society. Ravenstein analyzed census data in the United Kingdom and highlighted important patterns and regularities of population mobility amongst the Irish, British and the Scottish Kingdoms. He supported his work empirically with census data where migration movements of millions of citizens were recorded. A large volume of research work utilizing survey-based methods followed aimed at analyzing and extracting urban patterns with the main focus being on human migration utilizing survey-based methods [17] [18] [19] [20]. However, the datasets collected using survey-based methods have provided only a very static viewpoint of human movements and hence, limits the potential for extracting urban patterns. Even today, this type of data could inform us for instance about the city or a country in which an individual resides, perhaps even the year of some occasion but they do not point to the exact place people go nor the timing of their visits. Thus, survey-based methods suffer from limited spatial and temporal granularity for the aim of extracting finer urban patterns. While research using such type of datasets can provide us meaningful insights into large scale human migration mobility, it is not able to capture any patterns related to the dynamicity between and within cities.

But then how can someone extract meaningful urban patterns in dynamic urban environments? One example of these patterns could be to understand how people move in cities across space and time. This pattern for instance has been motivated by the process of intensive urbanization that took place heavily in the second half of the twentieth century. As crowds of agricultural population started shifting towards the cities, resulting in a sudden increase in the population size, city planners became overwhelmed with big challenges in deploying and allocating cities' resources such as transportation infrastructure and providing administrative services to citizens. At that point, gaining knowledge about the following become vital: how people use urban spaces[1], are there any social commonalities between population residing in different regions[2]; and where the crowd is concentrated and how they shift from one area to another[3]. The principle method to acquire such knowledge has been to conduct representative population surveys [21]. These surveys have made it possible to acquire some of these knowledge such as the origins and destinations of trips in a city [22][23], and the transport means employed by commuters. However, it has been very challenging to infer other types of knowledge due to the limitations of such static methods as, described earlier.

2.1.2 Mobile Phones as Sensors

In the early 1990s, the launch of the second generation cellular technology (2G) in Finland promised a massive change in human communications. Although the idea of using mobile phone was an idea that has been around for a while, it was this time that it began to become mainstream. For the first time in our history, the human movement could potentially be tracked and predicted with per second granularity, highly geographic precision and for very large scale population. When a mobile user initiated a call or sent a text message, a session is created in the database with her position at the nearest Base Transceiver Station (BTS) that was handling the communication channel. While celebrating the opportunities that these massive urban sensing data brings to us, there was also fear. Privacy concerns

[1] This is referred to in this thesis as "functional regions".

[2] This is referred to in this thesis as "Socio-demographic Regional Patterns".

[3] This is referred to in this thesis as "Recurrent Crowd Mobility Patterns".

were raised about the possibility of the misuse of this data by the telecommunication providers. Researchers on the other side found it challenging to get access to such data due to legal restrictions and economic constraints. With little or no information about the social demographic characteristics of the individuals or the types of activities they are performing, the possibility of extracting meaningful urban patterns using such data is still inadequate [24]. For example, although the study provided by Becker et al. [25] promises to see plausible estimate of the estimates of the spatial distribution of residence by users of different phone usage patterns (e.g., classified "workers" or "partiers" based on call detail records (CDR)), a complete picture of human activities in non-home/work categories is very difficult to be inferred, as it is hard to validate those types of activities from purely the CDR data. Similar challenges are faced by other studies, In [26], Eagle et al. found it hard to differentiate between non-home/work activities. Other limitations of CDRs lie in their sparse temporal frequency where a sample is generated only when one of the prior mentioned transactions occurs, and on their rather coarse spatial granularity, as locations are based on the granularity of a cell tower [25]. In addition, cell towers vary in density from an area to another which affects the estimates and precision of the data recorded.

Despite the fact that the appearance of cellular data constituted a big step towards understanding human movement on a large scale compared to the data captured from survey-based methods, it did not provide the opportunity for researchers to shed light on extracting urban patterns taking place in cities captured by millions of people. For that a technology that would enable the recording of peoples' activities with spatial granularity of a few tens of meters was required to emerge.

2.1.3 The rise of location-based social networks (LBSNs)

The introduction of the World Wide Web in the early $1990s$ and the emergence of Internet services contributed in transcending the constraints imposed by the physical world and led to entering the digital era where information storage and exchange were becoming key to everything. New communication methods such as email let people interact and communicate reliably with their peers around the world, commercial activity moved also online in which money was able to be transferred from one party to another almost instantaneously. The improvements of web search engines were introduced allowing people to navigate effectively through massive amount of information and data.

The progress of this digital era shed the light on the expectation that people will care more about what was happening in the virtual universe rather than their real lives, this favoured further the introduction of the online social networks and in particular, Facebook which was launched in 2004. Facebook spread very fast across university students and today counts for more than one billion registered users [27]. The introduction of smart phone afterwards in the early $2000s$ however signalled a massive transition in the way people were accessing traditional web services. The classic image of a user who is using her desktop machine to access their email and web services began to fade and progressively a new type of web user emerged. That was the mobile user, carrying a computational device and capable of accessing the web from almost anywhere.

With the substantial and quick growth of mobile technologies, the idea of adapting social network platform to the mobile space emerged. The first online social networks that explicitly use location as their primary feature appeared in 2008. Foursquare[4] and Gowalla[5] took the lead in this new space and their service was based on a simple notion: *share with your friends information about the place where you are.* Despite concerns of sharing such private information, the thrill of exploring urban space in a new way attracted lots of users to the two location-based social networks and created a massive user base for each of them [28]. Twitter and Flicker were then first to allow the association of photos and messages (tweets) with the location/geographic information utilizing the GPS modules embedded in smartphones.

However, the activity of users in location-based social networks has brought into existence a completely new paradigm of crowdsourced data. These new type of datasets are expected to have a real impact on extracting interesting urban patterns that could potentially benefit various amount of applications including: public safety, transportation, urban planning and others. Moreover, due to the richness of this data comprising of text, images associated with the geographical information, it paves the way for computer science to deploy a new era of machine learning techniques that could make sense of this very special type of spatio-temporal data sources.

2.2 Background - Location-based social networks unique properties

The addition of spatial property in a location-based social networking service bridges the gap between the social networking services and the real-world social networks. Location-based Social Networks is formally defined as a type of social networking in which geographic services and capabilities such as geocoding and geotagging are used to enable additional social dynamics [29]. A comprehensive definition for the location-based social networks was given by Yu Zheng [30], as:

"A location-based social networks (LBSNs) do not only mean adding a location to an existing social network so that people in the social structure can share location-embedded information, but also consists of the new social structure made up of individuals connected by the interdependency derived from their locations in the physical world as well as their location-tagged media content, such as photos, video, and text. Here, the physical location consists of the instant location of an individual at a given timestamp and the location history that an individual has accumulated in a certain period. Further, the interdependency

[4] https://foursquare.com

[5] http://blog.gowalla.com

includes not only that two persons co-occur in the same physical location or share similar location histories but also the knowledge, e.g., common interests, behaviors, and activities, inferred from an individual's location (history) and location-tagged data."

The location-based social networks' unique properties can be summarized as follows:

- **Spatial Accuracy:** Location-based services provide for the first time the opportunity to record the geographic position (location) of an individual user in terms of GPS accuracy (10s of meters) as well as capturing the user' activity through the venue category (if using Foursquare) or may be through textual information (if using Twitter). This gives a clear advantage over cellular networks data (CDRs) that provides the accuracy of the user's location with respect to the nearest BTS but the user's exact location is missing. As will be discussed in later chapters, this precision of location attached with the user's activity may be used to infer interesting urban patterns across cities (for instance, where crowd shifts across space and time, and why).

- **Publicly available:** The second property of this type of data, the availability of such data to a researcher for gathering and analysis. The first route for gathering such data is using Twitter's Streaming API[6] which can be used for free and captures a sample of 1% of all Tweets. A second route for accessing this data could be through Twitter decahoses[7] which provides a sample of 10% of all Tweets but with a paid option. An alternative route to the previous two is through Foursquare's own API, yet the corresponding query limits yield much smaller datasets than Twitter. It is important to note that with all these routes, the tweets or check-ins that have been set to be private will not be captured. The merits of the public access of data is two-fold; first, the academic researcher can use the data to conduct new analyses, techniques and models, and secondly, research outputs can be reproduced by other researchers upon publication.

- **Global Accessibility:** The location-based social networks applications are deployed on the web, thus anyone at any place that has an Internet connectivity can access it. A remarkable feature in location-based social networks is the scale of capturing user activities. Unlike survey based methods that have been discussed before, location-based social networks allow for the collection of data that goes beyond the limitations posed by an experimental setting, both in terms of the scale of participation and duration of the experiment. Both of these features are especially relevant to the work presented in this thesis as it will allow us to extract more meaningful urban patterns that would not be feasible without the scale of the data and its time duration.

[6] https://dev.twitter.com/docs/streaming-apis

[7] https://gnip.com/realtime/decahose/

- **Venues Category:** Location-based social networks such as Foursquare are special for the multiple layers of information associated with them. For instance, the venue database constitutes the core of Foursquare in which not only the location of the user is recorded but also semantic information about the user's activity based on the exact place that is visited (e.g., American Restaurant, Library, or University). Today, there are already several type of applications that utilize such information through the Foursquare venue API[8]. In addition, several research work have been based on such type of unique information [31][32][33][34]. The initial set of venues when Foursquare was launched was at scale of hundreds of thousands of Points Of Interests (POIs). This has since been augmented and empowered through crowdsourcing in which users keep adding new venues every day. Overall, Foursquare enumerates more than 50 million venues globally, which span across the majority of countries around the world.

2.3 State-of-the-Art - Extracting urban patterns

This section provides just a brief general overview of relevant research in the state of the art related to the extraction of urban patterns. In each of chapters 4 to 6 to follow, the related work of each of the proposed core contributions will be presented in-depth highlighting how the proposed work progresses the state-of-the-art.

Urban computing [35] is emerging as a new paradigm where every vehicle, device, building, and person can be used as a sensor for probing city dynamics and further using advanced machine learning and data mining techniques for serving people and their cities. The work presented in this thesis is also a step towards urban computing. In this section, the state-of-the-art in extracting urban patterns in cities is reviewed pointing to the relevant key findings related to the work proposed in this thesis.

In recent years, many approaches have been introduced for identifying urban patterns in cities using mobility and LBSNs data. Bicoocchi et al. proposed in [36] an approach based on clustering and segmentation of GPS traces to infer the places of relevance to the user. In [26], Eagle et al. introduce Principle Component Based approach to infer places and mobility urban patterns on the basis of nearby RF beacons (e.g., WIF and GSM towers) where the top eigenvectors of the PCA represents human activities (termed as eigenbehaviors). Sigg et al. [37] compare various data mining techniques for extracting urban patterns from mobility data where they concluded that Independent Component Analysis (ICA) and PCA are the best suited for identifying human daily patterns.

The previous work was focusing on unsupervised learning methods and other clustering methods (e.g., K-means) which shows a great success in the prior art for detecting and extracting interesting urban patterns in cities through grouping together days that are similar for the whole 24 hours. However, there is another type of urban patterns

[8] https://developer.foursquare.com/

that could be of interest which results from clustering days for certain time interval only [38]. Topic models were found of a great benefit in extracting such type of urban patterns. Topic models are kind of probabilistic models (unsupervised methods) that are used for discovering semantic structure of a document collection. These unsupervised methods have been very useful in extracting useful semantic information in a variety of applications that requires identifying unique topics or concepts, such as distributional semantics [39], word sense induction ([40]; [41]), and information retrieval ([42]). In very recent years, we have seen topic models as a potential type of machine learning models for extracting individual recurrent urban patterns in cities. In [43], Laura et al. introduced Latent Dirichlet Allocation (LDA) based approach to automatically discover users' routine behaviour utilizing Google Latitude mobility dataset. The main objective was more towards extracting routine behaviours other than relevant places compared to what have been proposed in [36] and [26].

For the sake of predicting user's specific activity pattern, Samiul et al. [44] propose foundational tools that can be used to predict user's specific activity patterns. They address identified limitations and adapted a topic model that can extract the activity patterns without the socio-demographic details of the individuals. Felix et al. in [45] applied and trained LDA based topic model on a combined textual and movement data on averaged week activity for a check-ins dataset. They further identified, analyzed, and interpreted the output topics in space and time with a focus in analyzing city areas usage with temporally varying profiles which charcterized the intensities of within-day activities.

Among the various state-of-the-art that focused on extracting urban patterns in cities, there has been more focus recently within the research community on extracting and predicting urban mobility patterns but still with a main focus on individual's mobility rather than crowd mobility. In [46] and [47], the authors mainly forecast billions of individuals' mobility traces rather than the aggregated crowd flows. One challenge is that this task is computationally expensive, and predicting individual's mobility is not necessary useful to public safety and disaster management and other applications that could more benefit from crowd analysis. In [38], authors presented an approach based on LDA with the aim for detecting recurrent activities with the aim of identifying hotspots in city life. Although the approach is very useful in detecting mobility patterns, the recurrent urban patterns detected from LDA for the weekdays was with a probability of maximum 0.22 and for the weekends was around 0.35, which we think it is not still high enough to claim a clear detection for mobility recurrent crowd patterns. Another branch of research focuses on predicting traffic volume and travel speed on the road [48] [49] [50] [51], to the best of our knowledge most of the work reviewed in this area focuses on single or specific road segments rather than citywide approach for travel and speed. Recently, the research community has started to focus on city wide scale traffic flows prediction. In [52], the authors proposed an approach to predict crowd flows using human mobility data, weather conditions, and road network data utilizing Gaussian Markov random fields, to cope with noisy and missing data. In [53], the authors propose a deep-learning based model for forecasting the flow of crowds in each region using trajectory, weather and events data.

Having presented in this chapter background and an overview of state of the art with respect to urban patterns, the next chapter takes a similar to describing the background and state of the art with respect to the machine learning techniques of relevance to this research.

Chapter 3

Background on machine learning & the TCDC

Chapter overview: *This chapter gives an overview on the main machine learning paradigms with a focus on the techniques that will be used in the core work presented in this thesis. The main machine learning paradigms introduced in this chapter are supervised learning, unsupervised learning, and deep learning. First, the chapter introduces some of the most well-known supervised learning models by describing a new proposed approach called TCDC that helps in choosing an optimum supervised learning model for a particular task. This approach was developed at the beginning of the research work when reviewing the machine learning techniques. This new approach has been published in [SmartCity 2015] and [PUC 2016]. Then unsupervised learning is introduced with a focus on specific clustering based techniques as well as topic models that are used in this research. Finally, the chapter highlights the growth of a recent paradigm called "deep learning" in which it will be used in this research with several of its variations.*

3.1 Background

Over the past decade, machine learning has developed distinct wide theoretical and practical tracks that revolve around predictive analytics and improving performance with experience. Machine learning has incorporated different methods from different origins, some of them have their origins from artificial intelligence whereas others are coming from applied statistics. This can be observed from the first journal in Machine Learning in 1986 where the main focus was around trees and rule-based models. By the late 90's, the picture had drastically changed focusing more on the methods originated from artificial intelligence like multilayer neural networks.

By early 2000s, there has been great advancements in the supervised learning approaches for classification and regression for the sake of prediction and avoiding over-fitting where techniques like pruning trees, weight decay and penalties have been introduced to counter this effect. Another area of progress in this period relates to developing supervised learning algorithms to take into account domain knowledge during the induction process, by selecting the main useful features (often called *feature engineering* process) via dimension reduction methods (for example: PCA, Linear Discriminant Analysis). Other advances include different methods for dealing and handling missing data via different induction methods. In 2006, a breakthrough in neural networks showed that a kind of neural networks called *deep belief network* could be efficiently trained using a so-called greedy layer-wise pretraining as outlined in [54, 55, 56]. This breakthrough has introduced a new wave of research in machine learning called *deep learning*, which shows that researchers are able now to train deeper neural networks that had been impossible before.

3.2 Supervised learning

Supervised learning is a paradigm in machine learning that is used when the correct output is explicitly given for given inputs (features). A reasonable example of this kind of problem is a dataset that considers a hand-writing recognition problem where it is formed of a collection of images of hand-written digits, and for each image, a determination is made of what the true image is. Thus, a set of features, and an output are stated explicitly in the form (images, digit). There are different ways that the dataset can be presented to the supervised learning process, the most common in practice, is a dataset that is already presented complete and entire before the learning process. Other variations include *active learning* and *online learning* [57]. In active learning, the dataset is formed through queries that are made, while in online learning, the dataset is given one example at a time. The latter happens in problems where streaming data needs to processed in real-time. Supervised learning can be classified into *regression* or *classification* models. Regression models can be defined as the learning process for predicting a continuous numeric quantity while classification models are used for predicting a discrete categorical response.

In this section, the most popular supervised machine learning models are summarized through introducing a new recommender approach that I developed during the course of my PhD, to support being able to select the most optimum machine learning supervised learning model for a particular task. The evaluation of the proposed recommenders is discussed in depth in Appendix A.

Assuming there is a problem that it is thought that supervised learning can solve, the question one is faced with is: How to choose between the multiple existing wide range of machine leaning models [58]. This question cannot be answered easily since choosing the optimum machine learning model largely depends on: (a) the characteristics of the data and the type of questions that need to be answered (b) the metric used for selecting the model. Given this challenge, a high level general approach called TCDC (which stands for Train, Compare, Decide, and Change) was designed. The TCDC proposed approach is composed of the TCDC closed loop process and the TCDC decision metric as illustrated in Figure 3.1a and Figure 3.1b and presented in this section.

3.2.0.1 TCDC closed-loop process

The proposed TCDC approach is introduced as a closed loop process as shown in Figure 3.1a for selecting the optimum supervised machine learning model. The proposed approach passes through four main phases:

- **Train**: In this phase, two models get trained on the dataset: (a) The most flexible but least interpretable model available in the practitioner's toolbox such as Support Vector Machines (SVMs) (this model will be referred to as a *reference model* and will not change through the whole closed loop process) (b) The simplest model available in the practitioner's toolbox which is the least opaque and highly interpretable (e.g.: Multivariate Adaptive Regression Splines (MARS), Partial Least Squares (PLS), Naive Bayes).
- **Compare**: In this phase, the predictive performance of the two models trained from the previous step are compared. The accuracies will be measured using identical resampled bootstrap datasets, and hence, a paired t-test will be used to determine if the differences between the models are statistical significant [59, 60]. In Figure 3.3 and Figure 3.4, this comparison is referred to as ACPP (stands for Acceptable Comparable Predictive Performance) in which the selection will be towards the simplest model with a certain tolerance. The degree of tolerance will be specified by the user which will indicate the tradeoff between predictive accuracy and the TCDC Decision Metric described in the next section (This tolerance can be thought of as penalty for moving towards an optimum model).
- **Decide**: This phase checks if the predictive performance of the simpler model is acceptable compared to the reference model. The simpler model is selected in case it has a comparable acceptable performance. Otherwise, we move to the next phase.
- **Change**: If this phase is reached, it means that an optimum model has not yet been found with an acceptable performance compared to the reference model. So in this phase, a more complicated model is selected and the *train* phase is started again with an aim of finding the optimum model and at that stage, the closed loop will get broken. In this phase, the highest predictive performance model from two or more similarly complicated models might be compared before inputing the best to the next phase.

The closed loop of the TCDC approach is not an infinite loop since it will end up with the choice of the reference model if none of the other models was found to have an acceptable comparable performance. By employing the TCDC approach, the optimum machine learning model that has an acceptable predictive performance with the minimum computational complexity, highest interpretability and easiest to implement can be recommended to the user.

3.2.0.2 TCDC Decision Metric

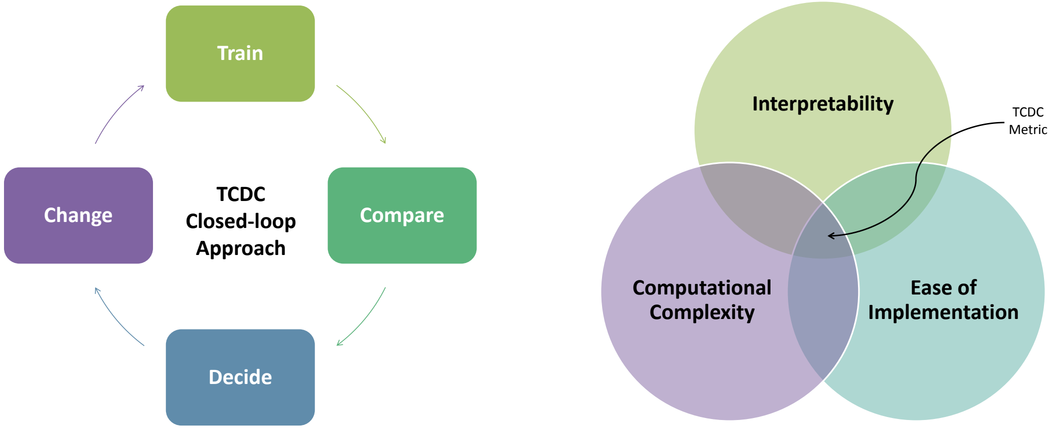
In the proposed TCDC approach, the objective is not only to choose a model with an acceptable predictive performance compared to the reference model but also the model that achieves a good trade-off between the factors illustrated below, that is referred to as *benefit metrics*, as shown in Figure 3.1b

- **Interpretability**: While a primary interest of machine learning models is to generate accurate predictions, a secondary interest may be the ability to interpret the model and understand why it works. The unfortunate reality is that, the more accurate model, the less interpretable it is. Hence, a tradeoff should be made between interpretability versus predictive performance [61]. Hence, trying out first the more interpretable models in our proposed TCDC approach is a great advantage if it is found to have an acceptable performance.

- **Computational complexity**: Different machine learning models suffer different levels of computational complexities [61]. This should be taken into account when choosing the model since a high computational complex model (e.g.: SVM) may not allow the prediction equation to be exported to a production system in practice.

- **Ease of implementation**: Some models are not easy to implement while others are very easy to develop. Trying first an algorithm which is easy to be implemented (e.g.: MARS model) is optimum for saving time especially if they yield a performance close to those models that requires more time to be implemented.

Sorting the machine learning models first according to the benefit metrics discussed above is one of the key ideas in the proposed general TCDC approach. In the proposed recommenders discussed in the next sections, SVM is chosen as the reference model in the TCDC approach since it has been found to be very powerful (i.e.: most accurate) across many problem domains, because it performs well in practice and because it is easy to use [62]. In addition, there are various methods and techniques for splitting and resampling the data for the sake of choosing the best performance model [61]. In the proposed TCDC approach, the recommendations summarized in Figure 3.2 are suggested based on what was concluded from [63, 64, 65].



(a) TCDC Closed-loop Process.

(b) TCDC Decision Metric (Benefit Metrics).

Figure 3.1: TCDC Proposed Approach

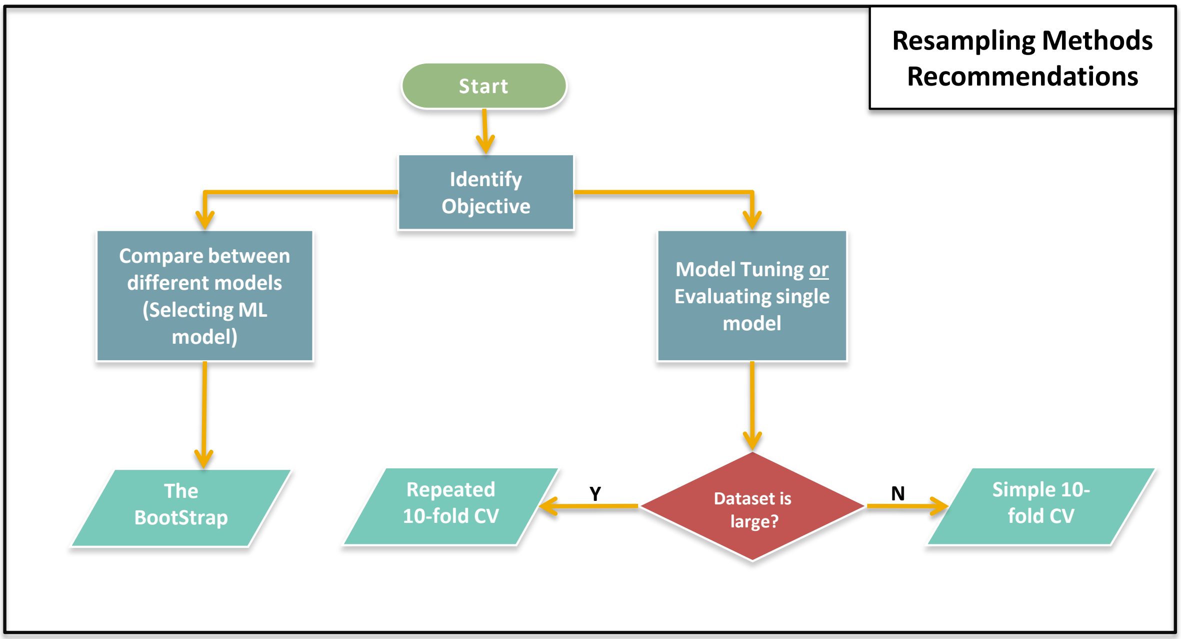

Resampling Methods Recommendations
Start
Identify Objective
Compare between different models (Selecting ML model)
Model Tuning or Evaluating single model
The BootStrap
Repeated 10-fold CV
Y
Dataset is large?
N
Simple 10-fold CV

Figure 3.2: Resampling Methods Recommendations.

3.2.1 Proposed TCDC Recommender Approach

3.2.2 TCDC-based Recommender for Regression Models

In this section, the focus will be on regression models, which are used for predicting a continuous outcome. As a first step when approaching a regression supervised learning problem, we recommend applying a flexible smoother model called *Loess* [66] in order to explore the relationship between the features and the outcome variable and hence, discover whether it is linear or non linear regression problem. Exploring visually the scatter plot generated from the applied loess model will allow us to identify the degree of linearity of the learning problem and hence, towards taking a decision as to which part of the workflow shown in Figure 3.3 to follow. A summary of the proposed recommender for

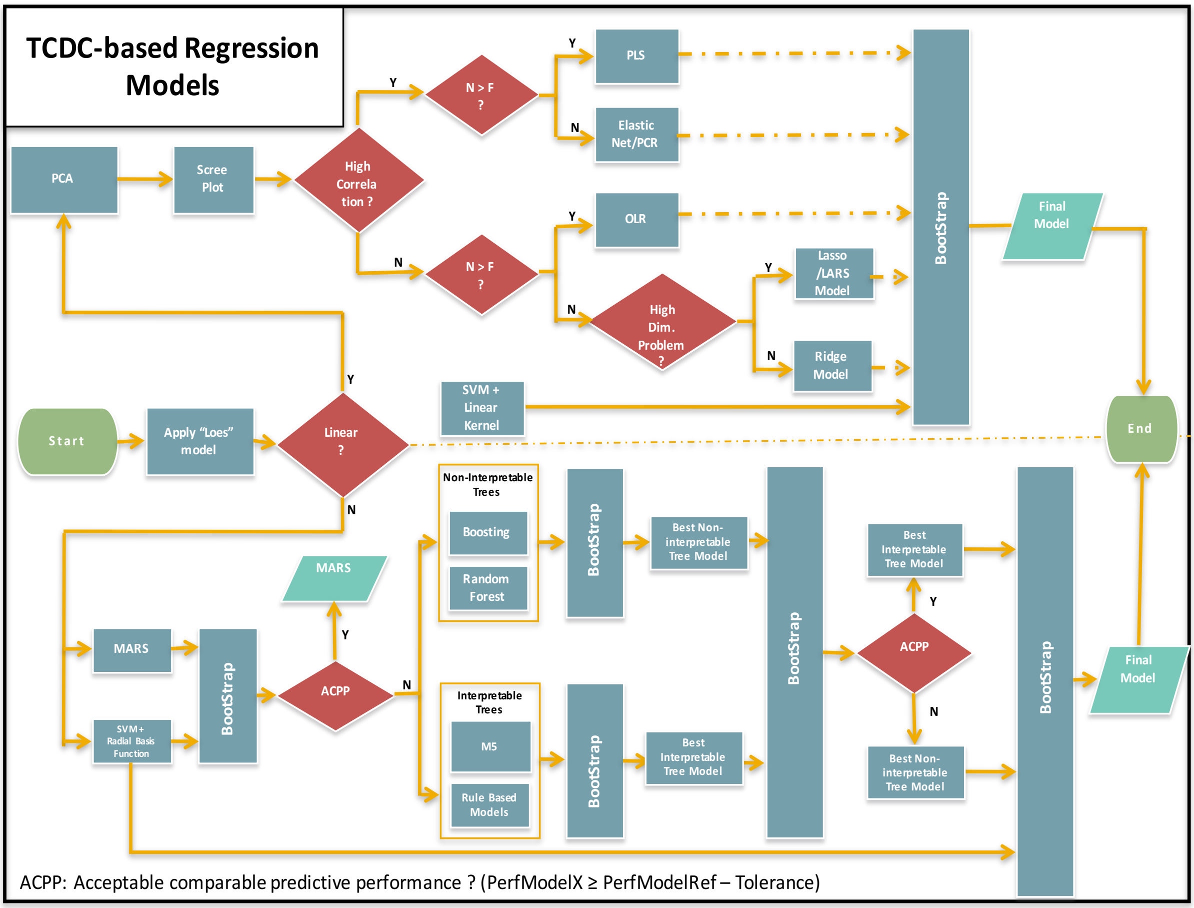


Figure 3.3: TCDC-based Recommender applied for supervised regression learning models.

the regression models is shown in Figure 3.3 and will be described in the following two sections.

3.2.2.1 Linear regression models

Assuming the relationship between the features and the outcome is linear, we recommend first investigating the degree of between-features correlations. We suggest using *Principle Component Analysis (PCA)* for evaluating the magnitude of the correlation problem on the full set of features, and hence, evaluating the percentage of variance accounted by each component visualized from the so-called *scree plot*. For instance, having 2-3 components that have relative contribution to the total variance indicates the existence of 2-3 relationships between features.

In case of having high between-features correlations, we recommend afterwards to check the number of features (F) against the number of samples (N) existing in the dataset. In case the number of samples exceeds the number of features, we recommend using a *Partial Least Square (PLS)* model. PLS can be described as a supervised technique that finds the components that describes the most variability in the data and at the same provides the maximum correlation (maximum covariance) with the outcome. PLS has one tuning parameter which is the number of components and it is recommended to use either simple or repeated 10-fold CV method according to the size of the data set to estimate this tuning parameter (refer to Figure 3.2). On the other hand, if the number of features is greater than the number of samples, we suggest as a first preference to deploy shrinkage parameter methods especially *elastic net*. The advantage of the elastic net is that it provides effective regularization using the ridge penalty and feature selection quality as the lasso penalty. As a second preference, the use of *Principal Component Regression (PCR)* is recommended. This is composed of applying PCA first and then regression, and has been used widely in problems with high correlation predictors and having more features than data samples. However, we do not recommend the PCR as a first preference because PCA is considered an unsupervised method that does not take into account any correlation with the response. Hence, if the variability is weak with the outcome, the PCR has a greater chance to perform poorly.

On the other side, and in case of having low between-features correlation with number of features more than the number of data samples, *Ordinary Linear Regression (OLR)* can be applied safely, as under these circumstances as there will be unique set of regression coefficients existing. However, if the number of features is more than the number of samples, we suggest the model based on the problem's dimensionality. If it is a high dimension problem, we recommend using the *Least Angle Regression (LARS)* model that can be used to fit *lasso* models more efficiently; lasso models provide regularization to improve the model and simultaneously conducts feature selection. If it is not a high dimensional problem, we recommend using the *ridge* Model. In effect, this model shrinks the estimates towards zero as the regularizer coefficient increases as shown in [67].

Using Bootstrap (refer to Figure 3.2), we compare the performance of one of the selected linear models based on the problem's characteristics as we described above with the Support Vector Machines (SVM) (as the reference model) using linear kernel. If the performance is equivalent or within an acceptable range of compared to the SVM, it is recommended to choose the linear models according to the proposed benefit metrics. Linear models allow for more interpretability and a less complex model that can be better exporter to a production system.

3.2.2.2 Non-linear regression models

Assuming the relationship between the features and the outcome was found to be nonlinear after the data exploration initial step performed by the loes model, we recommend training the *Multivariate Adaptive Regression Splines (MARS)* model and comparing it with the performance of SVM using Radial Basis function kernel. We suggest using the MARS model given that its performance is acceptable and comparable to that which SVM scored. The preference for choosing and trying MARS initially is actually due to several reasons.

First, the MARS model conducts feature selection as the model equation is independent of the features that are not involved with any of the final model features. Second, the MARS model is considered one of the most interpretable models. Third, the MARS model requires very little preprocessing (Data Cleaning), and additionally, the correlated predictors do not significantly affect the model performance. Hence, it makes it a very suitable model to try first based on the TCDC approach and its decision metric discussed in the previous section.

If the MARS model did not perform well compared to the SVM model, we still do not recommend to go and choose SVM directly. Instead, we recommend to change the model according to the TCDC loop and try different regression trees and rule based methods before choosing the final model. We recommend comparing the performance of the interpretable regression trees models (*M5* and *rule based methods*) with the non-interpretable methods (*Random Forests* and *Boosting*). In theory, the predictive performance of the non-interpretable models is expected to outperform the interpretable models. However, in practice and in some cases, the performance of the interpretable models is really comparable to the non-interpretable models. In such a case, the interpretable models will be preferred to avoid the disadvantages of the non-interpretable models including high computational cost, high memory requirements, and less interpretability.

Finally, the optimum regression tree model chosen from the previous step will be compared again with the predictive performance of the SVM with *Radial Basis Function (RBF)* kernel using Bootstrap and a decision should be made for the tree models if they perform well relative to SVM.

3.2.3 TCDC-based Recommender for Classification Models

In the previous section, the focus was on introducing the TCDC-based recommender for selecting a regression model when approaching a problem of a continuous outcome. In this section, the TCDC-based recommender that deals with the categorical outcome (classification problem) is introduced. The classification model aims to categorize the data samples into groups based on the characteristics of the features and the minimization problem for achieving this is different for each technique. Although many of the regression models discussed previously can be used for classification purposes, the performance metrics differ from the regressions models. A summary for the TCDC recommender for the classification models is shown Figure 3.4 and will be described in the following two sections.

3.2.3.1 Linear classification models

In this section, we explore a recommender for approaching the machine learning problems when the relation between the features and the outcome can be approximated by a linear function. Following the same approach applied previously for the regression models, we

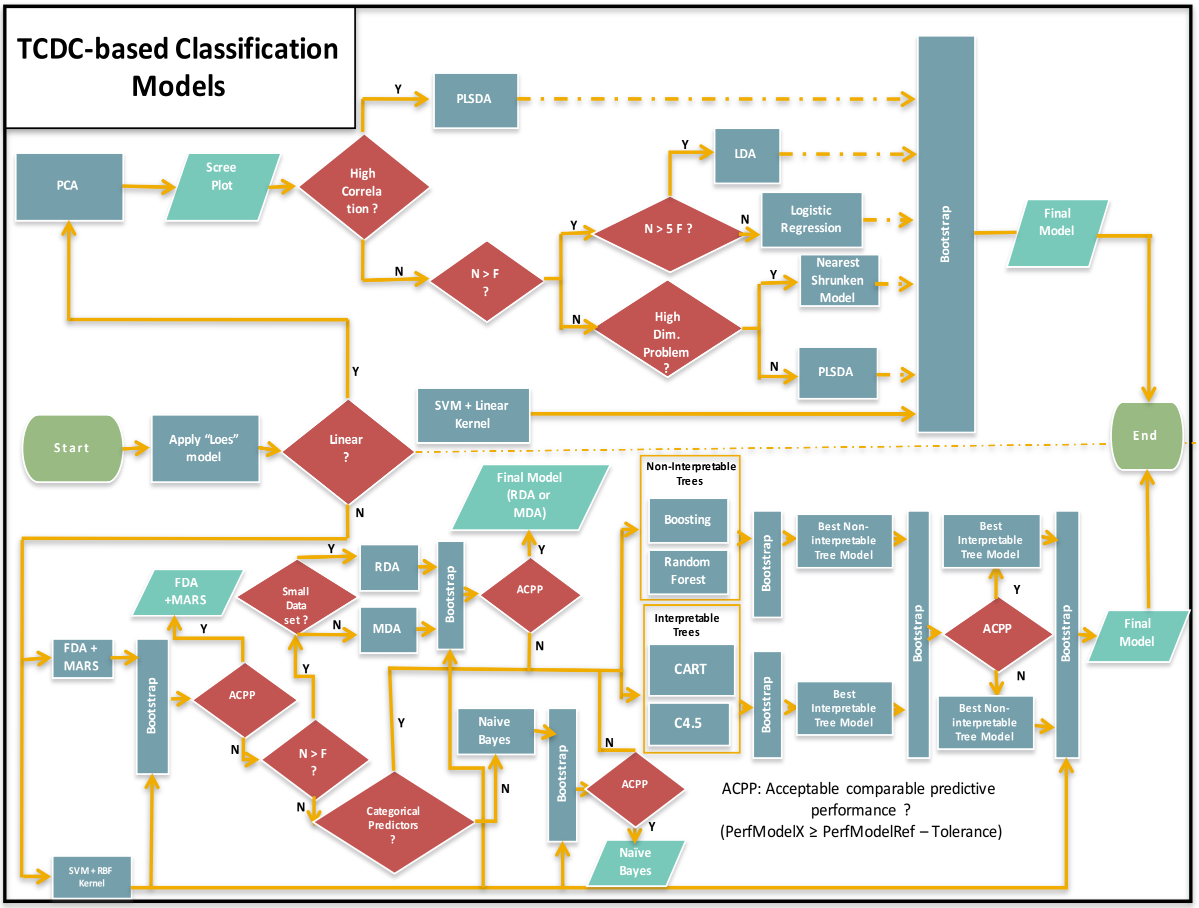


Figure 3.4: TCDC-based Recommender applied for different supervised classification learning models.

recommend as an initial step applying PCA and visualize the scree plot to characterize the magnitude of the between-features correlations. In case of high correlations, we recommend checking the number of samples (N) against the number of features (F). In case the number of samples exceeds the number of features, we recommend fitting *Partial Least Squares Discriminant Analysis (PLSDA)* model which is an evolution of applying PLS to classification problems. PLSDA aims to find the latent variables which will reduce dimensions whilst optimizing the correlation between categorical features. PLSDA does a good job in classification since it considers group information while trying to reduce dimension. In case of low between feature-correlations and having data samples at least 5 times more than the number of features, we recommend using *Linear Discriminant Analysis*. This model attempts to maximize a function that represents the difference between the means, normalized by a measure of the within-class variability, or the so-called *scatter*. Otherwise and in the case of having data samples less than 5 times the number of features, we recommend using *logistic regression*: a model which is somehow between the linear classification model, that uses hard threshold, and linear regression, that uses no threshold and it restricts the output smoothly to the probability range $[0, 1]$.

In the case of having the number of features that are more than the number of samples, we recommend checking the problem dimensionality. If it is a high dimensional problem, a linear classification model called *Nearest shrunken centroid* model (known as PAM) has been found to perform well in such kind of high dimensionality. The PAM model performs well for the problems with large number of features due to the fact that it has a built-in feature selection that is controlled via the shrinkage parameter [68]. In this low N and large F configuration, the data probably is difficult to be classified with a highly non-linear model and in this case, the PAM linear classification model is a good choice. On the other hand and in case of low dimensional problem but still where the number of features is larger than the number of samples, we suggest using PLSDA.

Based on the problem that we have, and the model we choose, we recommend comparing the predictive performance of the chosen model of SVM with linear kernel. Based on the TCDC approach introduced in Sect. 3.2.1, a decision will be favoured towards the chosen model if its performance is comparable and acceptable relative to the SVM model.

3.2.3.2 Non-linear classification models

In this section, the proposed TCDC-based recommendations for the non-linear classification machine learning problems are discussed. We recommend training *Flexible Discriminant Analysis (FDA)* model as it has several advantages based on our TCDC criteria discussed in Sect. 3.2.1. Hence, we believe it is worth exploring FDA as the first algorithm and selecting it if its performance was acceptable and comparable to the SVM with RBF kernel. FDA can be described as a process where for K classes, a set of K multivariate regression model using any model (we recommend this model to be MARS for the various reasons discussed in the previous section) can fit to a binary class indicators and using an optimal scoring technique, discriminant coefficients can be derived. If the predictive performance was found unacceptable compared to the SVM, then we recommend checking the number of features (F) against the number of samples (N).

In case the number of samples are more than the number of features, we then identify the dataset size and for a small dataset, we recommend using *Regularized Discriminant Analysis (RDA)*. RDA can be described simply as a method for providing nonlinear separating surface between Linear Discriminant Analysis and *Quadratic Discriminant Analysis (QDA)*. For medium to large datasets, we recommend using *Mixture Discriminant Analysis (MDA)*. MDA is a generalization of the Linear Discriminant Analysis; it allows the representation of each class by multiple Gaussian distributions and each of these distributions have same covariance structures but different means. A decision will be favoured towards choosing RDA or MDA (according to the dataset size) if they have acceptable performance compared to SVM.

In the case where the number of features more than the number of samples, we recommend checking if the features are categorical or the viability of converting them to categories. In the case where the features are categories or they are easily transformed, we recommend training *Naive Bayes (NB)* and choosing it as a final model in case its predictive performance is comparable to SVM. Naive Bayes (NB) classifier is a learning algorithm in which the features are discrete valued. NB is based on a strong assumption that the features are conditionally independent given the outcome. This is an extremely strong assumption yields a substantial reduction in the complexity of the calculations and so it is worth exploring the performance of NB. Laplace smoothing is a simple change to NB that in most cases makes NB works much better, especially for text classification problems.

In the previous scenarios, in case the performance of FDA, RDA, MDA or Naive Bayes was not satisfying compared to the SVM with RBF, we recommend at this stage training classification trees. The structure of the classification trees is similar to the structure of the regression trees. One of the reasons we recommend trying trees at this stage because if there was no model found with acceptable predictive performance from what have been tried previously. This may be due to the nature of the features and fortunately trees do a great job with handling different types of features as well as missing data.

For the trees, we recommend training and comparing the predictive performance between *Classification And Regression Trees (CART)* and *C4.5* classification models (Interpretable trees) which are considered the most widely used. They are similar models except that they are based on different splitting criterion, CART model is based on the Ginni Index criteria [69] whilst C4.5 is based on the cross-entropy criteria [70]. When training CART, it is recommended to create independent category features that may provide valuable interpretability regarding the relation between features and outcome. We do not think that the performance differs substantially between both models and hence a practitioner can rely only on trying one model but we have both in Figure 3.4 for a more comprehensive picture. Afterwards, we recommend comparing the predictive performance of the outperformed model from C4.5 and CART with the outperformed model from Boosting and Random forest (Non interpretable trees). The choice will be favoured towards the interpretable trees model if it has a comparable performance to the non interpretable model. Finally, the final model will be chosen after comparing the predictive performance of the selected tree model with the SVM model, favouring the tree model if it has an acceptable performance compared to the SVM. Eventually, if no model was found of an acceptable performance, SVM can be considered at this stage as the optimum model to be used.

3.3 Unsupervised learning

Unsupervised learning is a type of machine learning model used to draw inferences from datasets when the input data has no labeled responses (absence of ground truth). In this section, two of the most usable suite of unsupervised learning algorithms (clustering and probabilistic topic models) that will be used in the core chapters to follow are discussed.

3.3.1 Unsupervised Clustering Algorithms

Clustering is an unsupervised learning method. Given items $x_1, ..., x_n \in \mathbb{R}^D$, the goal is to group them into reasonable clusters. A pairwise distance/similarity function between items, and sometimes the desired number of clusters are needed. In this section, three of the most popular clustering techniques are reviewed that will be used for recognizing the "Temporal Functional Regions" introduced in chapter 6.

3.3.1.1 Agglomerative Hierarchical Clustering

The following steps summarize the procedures of the Agglomerative Hierarchical Clustering:

1. Assign each item $x_1, ..., x_n$ in its own cluster $c_1, c_2, ..., c_n$.
2. Repeat until there is only one cluster left:
3. Merge the nearest clusters, say $c_i \; and \; c_j$.

After performing the previous steps, the result will be a cluster tree in which it can be cut at any level to produce different clusters. For defining the "nearest clusters" in step 3, there is a need for defining a distance measure $d(x, x')$ between items. The following are some variations for measuring it:

- **Single-linkage:** This is equivalent to the minimum spanning tree algorithm. A threshold can be set and the clustering is stopped once the distance between clusters exceeds the threshold. Hence, the distance between clusters c_i and c_j is defined as: $d(c_i, c_j) = min_{x \in c_i, x' \in c_j} d(x, x')$.
- **Complete-linkage:** In this case the distance between clusters c_i and c_j is defined as follows: $d(c_i, c_j) = max_{x \in c_i, x' \in c_j} d(x, x')$. This usually generates compact and roughly clusters, equal in diameter.
- **Average-linkage:** In this case, the distance is somewhere between the single and complete and is defined as follows: $d(c_i, c_j) = \frac{\sum x \in c_i, x' \in c_j d(x, x')}{|c_i| \cdot |c_j|}$.

3.3.1.2 K-means Clustering

This is the most widely used type of clustering. It is an iterative algorithm which keeps track of the clustering centers (means) where the number of clusters k needs to be provided to the algorithm. The centers are in the same feature space of the input x. The following steps summarize the k-means algorithm:

1. Start by choosing random k centers $\mu_1, \mu_2, ..., \mu_k$.
2. Repeat until the clusters saturate:
3. Assign $x_1, ..., x_n$, to their nearest centers, respectively.
4. Update each center μ_i to the mean number of samples.

It is worth noting that step 3 is equivalent to creating a Voroni diagram under the current centers. Since k-means clustering is a special case of Gaussian Mixture Model (GMM) when the covariance of the Gaussian components tends to zero, it is unable to trace winding clusters. Hence, the next type of clustering is introduced that addresses this limitation.

3.3.1.3 Spectral Clustering

The previous discussed clustering techniques beside the generative models such as EM that are used to learn mixture density suffer from several drawbacks. First, harsh simplifying assumptions (e.g., that the density of each cluster is Gaussian) usually need to be made to use parametric density estimators. Second, iterative algorithms are needed to find a good solution since the log likelihood can have many local minima. A promising alternative that has recently emerged in a number of fields is to use spectral method for clustering.

Spectral clustering takes a graph W and the number of clusters C as input where graph nodes are $x_1, ..., x_n$ and the undirected edges have non-negative weights reflecting the similarity between nodes. The weights are symmetric: $w_{ij} = w_{ji}$ and $w_{ij} = 0$ if no edge. These weights can be represented in an $n \times n$ matrix W, which full specifies the graph. The graph is usually generated using one of these methods: (a) k-nearest-neighbor (KNN) graph. (b) Fully connected graph with RBF weights. (c) ϵNN graph.

From the formed weight matrix W, three different graph Laplacian matrices are defined: (a) Unnormalized Laplacian ($L = D - W$). (b) Normalized Laplacian ($L_{rw} = I - D^{-1}W$). (c) Another Normalized Laplacian ($L_{sym} = I - D^{\frac{-1}{2}} W D^{\frac{-1}{2}}$). It turns out that Laplacian's eigen values are always non-negative. In the ideal case, each cluster forms a connected component in graph W. Let U be the $n \times C$ matrix formed with these C eigenvectors as columns. x_i will be then represented by the $i - th$ row in U for $i = 1, ..., n$. Then all points within a cluster have the same new representation where clustering in these new space is trivial, with e.g. k-means. The following steps summarize the spectral clustering algorithm:

1. Input: graph W, number of clusters C.
2. Compute unormalized Laplacian ($L = D - W$) or normalized Laplacian ($L_{rw} = I - D^{-1}W$).
3. Compute matrix $U = [\phi_1 | ... | \phi_C]$ where C are the first eigenvectors.

4. Represent x_i by the i-th row in U.

5. Use k-means to cluster the new representation of x_i into C clusters.

3.3.2 Unsupervised Topic Modeling

Machine learning scientists have developed *probabilistic topic modeling*, a suite of algorithms that aim to discover and annotate large archives of documents with thematic information. Topic models are statistical methods that analyze the words of the original texts to discover the themes that run through them, how the extracted themes are connected to each other, and how they change over time. Since topic modeling algorithms do not require any kind of annotations to the documents or labeling of the documents, the topics extracted emerge from analyzing the original texts. The following highlights two of the most popular implementations of topic models that will be used in several sections in this thesis (mainly chapter 5 and chapter 7).

- **Latent Dirichlet Allocation (LDA)**: LDA [71] is a statistical model that reflects the intuition that documents exhibit multiple topics. Each document exhibits the topics with different proportion. As we described earlier, the objective of the topic modeling is to automatically discover the topics from a collection of documents. The central computational problem that LDA solves is to use the observed documents to find the *hidden structure*; the topics, per-document topic distribution, and the per-document per-word topic assignments. This can be thought of "reversing" the generative process in which the objective is to infer the hidden structure that likely generated the observed collection. In particular, this computational problem that LDA solves is the problem of computing the posterior distribution and the conditional distributions of the hidden variables given the documents.

- **Non-negative Matrix Factorization (NMF)**: The work of topic modeling has largely been focusing on the use of LDA. However, NMF can be also applied to textual data to reveal the hidden structure [72]. NMF decomposes a data matrix into factors that are constrained for not containing negative values. Given a document-term matrix $\mathbb{A} \in \mathbb{R}^{m \times n}$ representing m unique terms that exist in corpus of n documents, NMF provides a reduced rank-k approximation comprising of two non-negative factors $\mathbb{A} \approx \mathbb{W}\mathbb{H}$, where the objective is to minimize the reconstruction error between $\mathbb{A}$ and the low-dimensional approximation. The columns or basis vectors of $\mathbb{W} \in \mathbb{R}^{m \times k}$ can be interpreted as the topics, defined with non-negative weights relative to the m terms/words. The values in the matrix $\mathbb{H} \in \mathbb{R}^{k \times n}$ provide the per-document topic distribution.

LDA has been used widely in the literature for extracting individual recurrent patterns in cities [71][43][36]. However and to the best of our knowledge, NMF has not been used in the literature before for this objective. Nevertheless and for the first time, it is shown in this thesis that NMF is quite powerful for extracting urban patterns. In particular in chapter 7, NMF is utilized for extracting Recurrent Crowd Mobility Patterns where it was found that it surpasses the performance of LDA.

3.4 The rise of Deep Learning

Deep Learning is considered one of the most recent advancements in machine learning aiming at learning feature hierarchy formed by the composition of low level features [73]. A standard neural network (NN) consists of many simple connected processors called neurons where each produced a sequence of real-valued activations. First layer of neurons (input neurons) gets activated through sensors perceiving the environment, the following middle layers neurons get activated through the weighted connections from previous layers. Learning in neural networks is about finding weights that make the NN exhibits the desired behaviour. Depending on the NN architecture and how the neurons and layers are connected to each other, this may require long causal chains of computational layers where each layer performs a non-linear transformation for the aggregate activation for the whole network. Deep learning is about accurately assigning weights across many such layers.

Shallow NN models with few layers have been around for decades, models with few successive nonlinear layers of neurons date back at least to the 1960s and 1970s. An efficient gradient descent method called *backpropagation (BP)* was developed in the 1970s but have been applied to NN afterwards in 1981. However, BP has been found to be difficult to work with deep learning with many hidden layers in practise by 1980s and become a very important research subject by the early 1990s. DL became practically feasible to some extent through the help of unsupervised learning as a pre-training procedure through a new architecture called *Deep Belief Networks (DBNs)* introduced by Hinton et al. in 2006 [74], this was considered the first breakthrough in the DL field. DBNs use an unsupervised learning algorithm that greedily trains one layer at a time where each layer forms a Restricted Boltzmann Machine (RBM) [75]. Shortly afterwards, auto-encoders [55] were proposed exploiting the same concept of training intermediate levels of representations using unsupervised learning performed for each layer. Other algorithms have been introduced afterwards that follow the same principle that is neither RBMs nor auto-encoders [76]. Since 2006, deep networks have seen huge success in several tasks and applications including but not limited to, dimensionality reduction [77], natural language processing [78], classification tasks [79], and several more. In fact, since 2009, supervised deep NNs have won many official international pattern recognition competitions, outperforming alternative machine learning models such as kernel machines [80].

Another thing that helped deep learning to evolve was related to the progression of the hardware for training such computational expensive models. While the previous millennium saw several attempts at creating specific NN hardware such as the work proposed by Jackel et al. [81] and Ramacher et al. [82]) as well as exploiting standard hardware such as the work proposed by Anguita et al. [83] and Muller et al. [84], the new millennium brought a DL breakthrough in form of utilizing cheap Graphical Processing Units (GPUs). GPUs excels at the fast matrix and vector multiplications required not only for video games but also for training NNs, where they can speed up the learning process by a factor of 50 or more. In these days, most of the recent successes in contests for patterns recognition have been utilizing multiple GPUs for training deep NNs and outperforming existing state-of-the-art results. In the work presented in this thesis and in particular in chapter 5 and chapter 8, four of the recent advancements in deep learning are used entitled: Deep Belief Networks (DBNs), Convolutional Neural Networks (CNNs), Recurrent Neural Networks (RNNs) and Long-Short Term Memory (LSTMs). These techniques will be discussed in depth in the corresponding core chapters to follow.

Having presented in this chapter an overview on various machine learning techniques of relevance to the research presented in this thesis, the next chapter takes the first step towards the core of this thesis by describing the LBSNs dataset that will be used in the core contribution chapters.

Chapter 4

Dataset gathering and characteristics

Chapter overview: *The dataset considered in this thesis expands upon existing work on LBSNs datasets through its size and diversity. The dataset gathered tracks the activities of users on Twitter for a period of* 2 *years, starting from January* 2013*, and ending with December* 2014*. To the best of our knowledge, this dataset is the largest LBSNs data that has been analyzed compared to the maximum duration of* 1 *year LBSNs datasets that have been used before in the literature [6][8][5]. This chapter starts by describing the dataset format and then the initial filtering procedures on the original dataset needed for the core chapters to follow. Finally, some of the statistics related to the dataset density and application types used in the dataset are shown.*

4.1 Dataset format

The dataset used can be gathered from GNIP[1], a Twitter's enterprise API platform, delivering a wide range of APIs for gathering data from Twitter's Firehose. A similar publicly available dataset but for less time duration (spanning from late September 2010 to late January 2011) which can be used for generating some of the patterns discussed in this thesis is available from [5]. In the dataset, each tweet is stored as a tuple with the following attributes: userID, tweetID, time, text, statusesCount, followersCount, friendsCount, provider, expandedURL, location. An example of tuple is: *[824895, 3293997, 2013-05-01T01:00:00.000Z, Eating before the movie (@ Village Pourhouse - @pourhousedwntwn w/ 4 others) [pic]: http://t.co/iRKziigKvY, 9528, 385, 413, foursquare, http://4s.com/ZSsSMP, 40.731, -73.988].*

4.2 Dataset filtering

From this original Twitter dataset and for the purpose of achieving the core work presented in chapter 5, the whole dataset from NYC is filtered for each borough (Manhattan, The Bronx, Brooklyn, Queens, and Staten-Island). For performing this step, the boundaries

[1] https://gnip.com

specified by BetaNYC are used. BetaNYC works with New York City government, elected officials and other stakeholders to engage NYC "civic-minded" technology and design community[2]. Furthermore, and for the work proposed in chapter 6 and 7, we needed to further filter Manhattan data based on the boundaries of its zip-codes[3]. Figure 4.1a and Figure 4.1b show the boundaries used for filtering the dataset for the five boroughs including Manhattan and the zip-codes for Manhattan respectively.

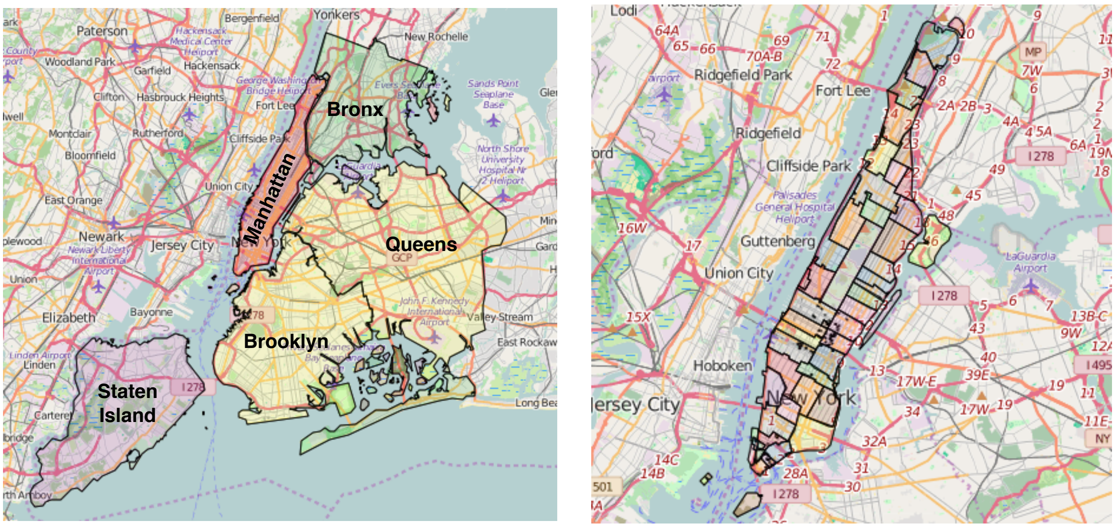
The Manhattan filtered dataset resulted in a total collection of $596,757$ users and $13,296,244$ geo-tagged tweets. The total size of the dataset is 3.39GB. We observed an average of 22.28 geo-labelled tweets per user, spanning from at least 1 tweet for $170,185$ users and up to a maximum of $89,999$ tweets for 1 user. This can be observed in more detail in Figure 4.2a. Moreover, it is noticed that between 10 and $1,000$ tweets is the most dense and typical in the dataset.

4.3 Dataset density based on different factors:

The density visualization for the filtered dataset is illustrated in Figure 4.3 for the average density for a week day compared to a weekend day. Figure 4.3a shows the spatial density for Monday as this is the day with the least tweets during the week. Moreover, the spatial density of regions on Sunday is illustrated in Figure 4.3b as compared to other days of the week, it is expected that on Sunday people will not work as much. In particular, it is worth noticing that Sunday is much more dense spatially than Monday. We expand on this result and show the density comparison between weekdays and weekends using a different visualisation in Figure 4.4. Moreover, the frequency of tweets per hourly basis is illustrated in Figure 4.2c. The lowest number of tweets are observed generally to occur at $9AM$, on Wednesdays, while the highest number of tweets typically occurs at $11PM$ also on Wednesdays. It is noticed that there is a drop in the users' activity starting from $02AM$, which decreases to low percentages especially between $05AM$ and $08AM$-$09AM$, suggesting a sleeping period. Further, the activity of the users increases steadily until $03PM$ and further reaching highest frequency during the evening and close to midnight, suggesting that most tweets occur outside of working hours. In addition, Figure 4.2d illustrates the tweets frequency per weekday, illustrating that the highest and lowest amount of tweets occur on Fridays and Mondays, respectively. Figure 4.2b expands on these results, combining an hourly visualization per weekday.

[2] The data can be found at [http://data.beta.nyc/dataset/nyc-borough-boundaries] and was last updated on January 5, 2015.

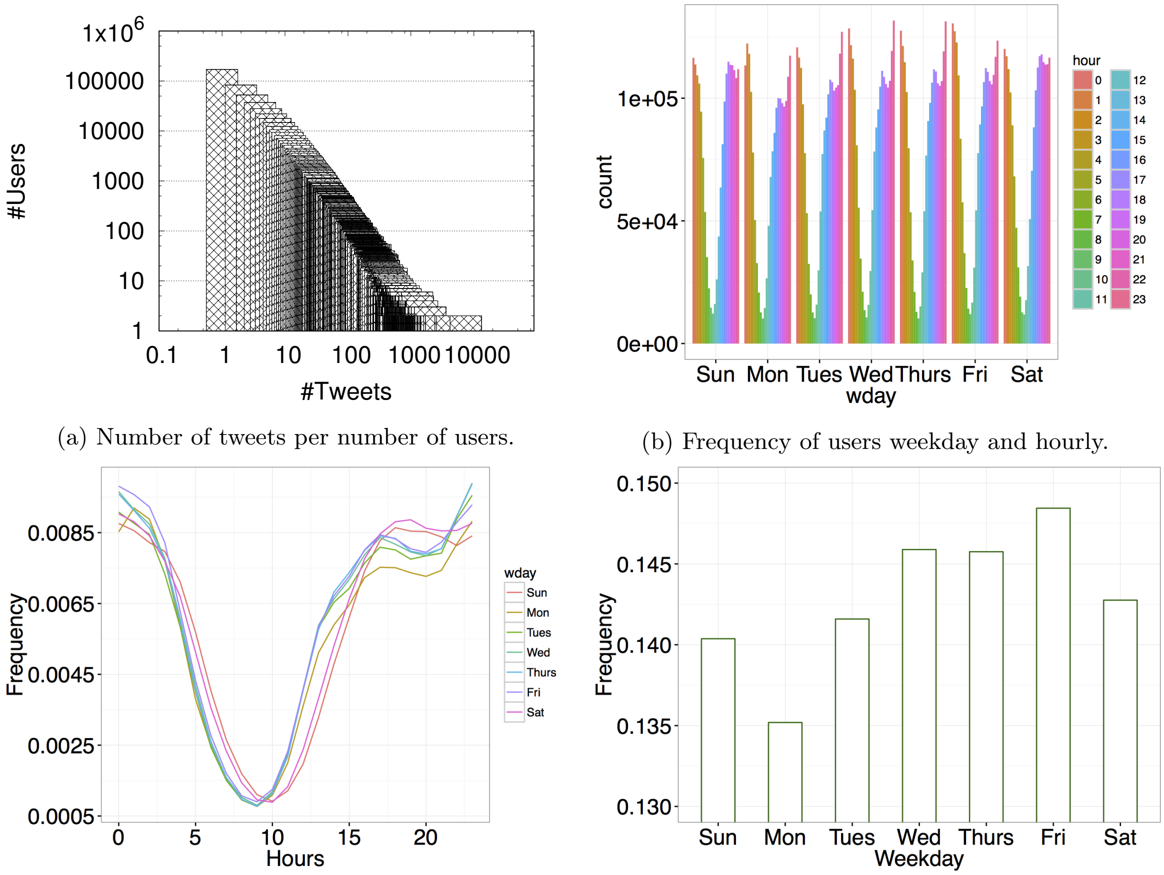
[3] The data for these zip-codes boundaries can be found at [https://data.cityofnewyork.us/Business/Zip-Code-Boundaries].



(a) NYC Boroughs Boundaries.

(b) Manhattan zip-codes boundaries.

Figure 4.1: NYC boroughs boundaries and Manhattan zip-codes.



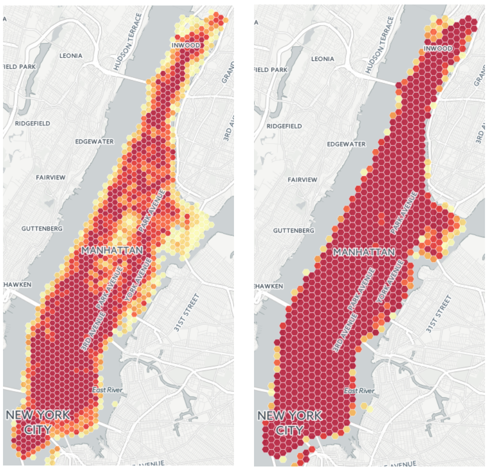
(a) Number of tweets per number of users.

(b) Frequency of users weekday and hourly.

(c) Hourly.

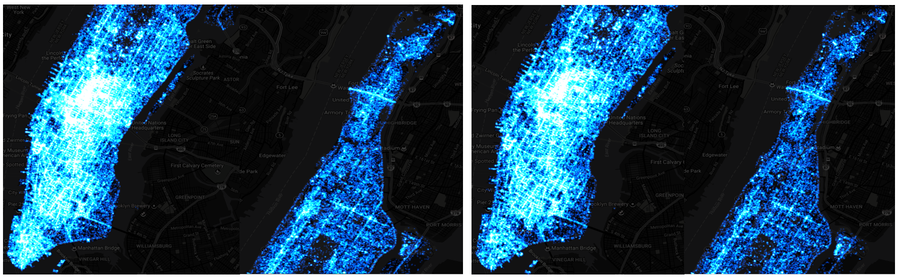
(d) Weekday.

Figure 4.2: Frequency of tweets and users.



(a) Monday. (b) Sunday.

Figure 4.3: First visualizations for the datasets' density maps.



(a) Weekday. (b) Weekend.

Figure 4.4: Second visualizations for the datasets' density maps.

4.4 Application types

The tweets were posted from 1115 applications, such as Twitter applications, Foursquare, Instagram, and dlvr.it. Their popularity can be observed in Table 4.1, with $\sim 52\%$ of the tweets originating from Twitter for iPhone. The second and third most popular sources are Instagram and Foursquare.

Name	Percentage
Twitter for iPhone	51.98%
Instagram	20.36%
Foursquare	10.97%
Twitter for Android	9.91%
Twitter for iPad	1.03%
dlvr.it	0.75%

Table 4.1: Distribution of Tweets' Sources.

Having presented in this chapter an overview on the dataset that will be used in the core contribution chapters to follow, the next chapter introduces the first urban pattern extracted and introduced in this thesis titled as *Socio-demographic Regional Patterns*.

Chapter 5

Extracting new Socio-demographic Regional Patterns

Chapter overview: *In this chapter, the first new type of urban pattern titled "Socio-demographic Regional Patterns" is introduced. The chapter also described how it can be extracted leveraging the power of deep learning using a Deep-Belief-Networks based model. This new approach has been published in [SIGSPATIAL 2016]. It is demonstrated in this chapter that this pattern can be extracted for each of the boroughs in NYC using the dataset discussed in chapter 4. Given weekly foot-prints captured from LBSNs, it is shown that the correct borough for these weekly-footprints can be predicted with an accuracy of up to 70%.*

5.1 Motivation

As highlighted in chapter 3, *Deep Learning* is considered one of the most recent advancements in machine learning aiming at learning feature hierarchy formed by the composition of low level features [73]. Deep Learning provides an automatic feature selection at multiple levels of abstractions allowing a system to learn very complex functions mapping the input to output directly without depending completely on human-crafted features. The concept of automating the feature selection process will become more important with the rise of machine learning applications and the availability of complex datasets in which building human-crafted features will become extremely expensive [73].

This chapter starts by demonstrating the first new type of urban pattern dicovered in this thesis which is referred to as *Socio-demographic Regional Patterns*. It is shown how this pattern has been inferred using DBN for detecting different patterns for very close proximity regions in cities. The *Socio-demographic Regional Patterns* is defined as patterns comprising individual activities in a certain region during certain time-slot for a particular day. For instance, for a particular area, the following spatial-temporal activities can be constructed from an LBSNs dataset: sat-Eating-12AM, sat-NightLife-01AM, sun-Traveling-15PM, mon-Shopping-17PM, and so on. This means as an example for a given region, there has been on Saturday an eating activity at $12AM$ and a NightLife activity at $01AM$. On Sunday, there has been a travelling activity at 15PM while on Monday, a shopping activity at $17PM$, and so on. Of course the aggregation of huge amounts of such spatial-temporal activities might form a unique pattern that varies from region to region which is precisely what the proposed model is trying to capture.

The power of DBN can be leveraged to infer such a possible unique pattern that can differentiate regions of close proximity to each other based on what is referred to as *region-footprints* captured from different regions. For example, by capturing crowd weekly activities from one of the boroughs in NYC (region-footprints), the proposed trained DBN model could potentially predict activities that might occur in this borough. This implies that such model can be leveraged for further understanding the socio-demographic commonalities between different regions across the globe. For example, figuring out which borough has the closest socio-demographic pattern to certain district in London can yield to better understanding of the commonalities between both and further exploring any relation with other external common factors including social, economic, and political that resulted in such commonality. Further extending this to more districts and cities across the globe may yield to new and deeper insights about our cities.

By applying the proposed approach, the following specific contributions are highlighted in this chapter:

1. DBNs for topic modeling can be successful for detecting new type of complex patterns in cities that has not been feasible before. In fact, in the previous state-of-the-art research, topic models have been seen as a great success for recognizing individual recurrent patterns [43][85] but to the best of our knowledge, there has been no prior work that applied and leveraged the power of DBNs for automatically extracting features for detecting new and more complex crowd patterns in cities.

2. The impact of applying different DBNs architectures is demonstrated for choosing the optimal K with the best performance for classifying different patterns among the boroughs within NYC where K is the latent representation for each region in the city.

3. The proposed *Socio-demographic Regional Patterns* model based on DBNs can be successfully applied on a sparse dataset coming from LBSNs. The proposed model has been validated on a geo-location dataset collected from New York City that was described in chapter 4. Finally, the proposed trained model has been shown to recognize unique pattern for each Borough within NYC based on the recurrent weekly region-footprint that recur over both space and time dimensions.

5.2 State-of-the-Art

The work presented in this chapter is based on applying DBN for topic modeling for the sake of detecting new activity patterns in cities. Hence, the state-of-the-art is reviewed from two perspectives: First is related to the prior work that utilizes LBSNs data for the sake of learning latent activity patterns in cities. The second is related to the evolution of DBNs until it has been utilized for topic models.

5.2.1 Activity Patterns in Cities

In recent years, many approaches have been proposed for identifying patterns using mobility and LBSNs data. In [36], Bicoocchi et al. proposed an approach based on clustering and segmenting GPS traces to infer the places of relevance to the user. In [26], Eagle et al. applied *Principle Component Analysis* (PCA) to infer places and mobility patterns on the basis of nearby RF beacons (e.g., WIF and GSM towers). The human activities termed as *eigenbehaviors* are represented as the top eigenvectors of the PCA. Similarly, the work presented by Sigg et al. in [37] compares different data mining techniques for extracting patterns from mobility data where they found that *Independent Component Analysis* (ICA) and PCA are best suited for identifying daily patterns of humans.

Although the previous unsupervised learning methods and other clustering methods (e.g., K-means) showed success for detecting patterns in cities through grouping together days that are similar for the whole 24 hours. However, there is a need for detecting patterns and clustering days for certain time interval only [38]. *Topic modeling* bridges this gap and several works has been based on topic modeling for extracting individual recurrent patterns. Topic models were introduced originally for finding underlying topics of words from a large collections of documents, with *Latent Dirichlet Allocation* (LDA) being one of the implementations of topic modeling that has been widely used for extracting individual recurrent patterns in cities [71].

In [43], Laura et al. presented a method based on LDA to automatically discover users routine behaviour extracted from Google Latitude mobility dataset. They focused more on extracting the routine behaviours other than relevant places compared to what have been proposed in [36] and [26]. In [38], authors presented an approach based on LDA for crowd detection that recur over time and space using Twitter posts of data in New York which contains a large set of users but in a sparse way. In [44], Samiul et al. provided foundational tools that can be used to predict user specific activity patterns. They addressed the main limitation for geo-location data for modeling individual behaviours and presented a topic model that can extract the activity patterns without the socio-demographic details of the individuals. Felix et al. in [45] combined textual and movement data and they applied topic models to the combined data on an averaged week activity in which they were able to show how city modalities evolve over time and space.

5.2.2 Deep Belief Nets for Topic Modeling

In this section, the motivation of applying DBN on document data is introduced. In the past, neural networks have had the drawbacks of the following: It requires labeled data that is difficult to obtain in most cases, the learning time does not scale well as it is very slow in networks with multiple hidden layers and it has the tendency to get stuck in a poor local minima [86]. Smolensky introduced the RBM [87] and Hinton introduced afterwards a learning algorithm called *Contrastive Divergence* for the training RBM [74]. Hinton and Salakhutdinov introduced the pretraining process by stacking a number of RBMs [77] and being able to train each RBM separately using the Contrastive Divergence algorithm. This was found to provide a crude convergence for the parameters in which can be used as an initialisation for the finetuning process. The finetuning process is very similar to the learning algorithms that have been used in the *Feed Forward Neural Networks* (FFNN). By using an optimisation model, the parameters converges to reconstruct the input. Hinton and Salakhutdinov validated their method on the popular MNIST dataset, where they demonstrated how they reduce the dimensionality of an input vector of 784-dimensions to 2-dimensions vector that well represent the data in the 2-dimensional space, in terms of the ability to spread the data based on labels in the output space [77].

Hinton and Salakhutdinov introduced the *Constrained Poisson Model* (CPM) as a core component of RBM to model word count data for performing a dimensional reduction on document data [88]. This approach has been replaced by the *Replicated Softmax model* (RSM) introduced by Hinton and Salakhutdinov due to the inability of the CPM to define a proper distribution over word counts [89]. Later RSM was introduced to act as the first component in the DBN pertaining process [90]. Hinton and Salakhutdinov validated their introduced approach on two datasets: Reuters Corpus Volume II and 20 Newsgroups. For measuring the similarity between documents using hamming distance, Hinton and Salakhutdinov introduced Semantic Hashing to produce binary values [88]. Based on the above findings, this work relied on using RSM as the first component in the DBN along with Semantic Hashing.

Though this section separates prior works on extracting activity patterns in cities and the huge advancements of DBNs for topic modeling in recent years, no prior work has attempted to apply DBNs for topic modeling for the sake of inferring new types of fine grained patterns within cities. In the rest of this chapter, it will be shown that it is possible to extract a unique pattern for different regions within cities leveraging the power of DBNs.

5.3 Socio-demographic Regional Patterns model

In this section, the problem of discovering Socio-demographic Regional Patterns is first defined along with some definitions which will be used in the chapter. Then the proposed

DBN-based model is described, which has been developed and trained for extracting the Socio-demographic Regional Patterns.

5.3.1 Notation and Definitions

Inferring unique patterns for different regions within the same city involves finding complex multi-weekly patterns from an individual's activities everyday (*individual-footprints*) within each region, we call these type of patterns, *Socio-demographic Regional Patterns*. *Region-footprint* is defined as a distribution of individual-footprints within a particular region in the city where each individual-footprint can be represented by day of the week, category of activity and time. Hence, the problem of inferring Socio-demographic Regional Patterns can be defined as: given a set of individual-footprints within a city for m weeks as r_1^1, r_1^2, r_1^3, ... $r_1^{n_1}$; r_2^1, r_2^2, r_2^3, ... $r_2^{n_2}$; ... ;r_m^1, r_m^2, r_m^3, ... $r_m^{n_m}$ where an individual on week $1, 2, ...m$ participates in activities $n_1, n_2, ..., n_m$ respectively, determine the K dimensional subspace through ϕ_k, $k \in 1, 2, ..., K$ where each ϕ_k is a distribution of region-footprint so that weeks with similar region-footprint lies next to each other. Figure 5.1 illustrates the Socio-demographic Regional Patterns inference problem.

5.3.2 Deep-Belief-Network Model

The main difference between the theory of the traditional FFNNs and DBNs is the training procedure as training DBNs is defined by two main steps: *pretraining* and *finetuning*. In the pretraining process, the neural network is separated pairwise to form two layered networks in which each form RBM. Each RBM is trained independently in which the output of the lower RBM is provided as input to the next higher-level RBM and so forth. The goal of this pretraining process is to perform rough approximations of the model parameters. These parameters are passed to the finetuning process. In the finetuning process, the network is transformed into a *Deep Autoencoder* (DA) by unrolling the whole DBN and by repeating the input and hidden layers and attaching it to the output of the DBN. By this structure, the DA can perform backpropagation on the unlabeled data by computing the probability of the input data $p(\hat{x})$ rather than computing the probability of the label given the input data $p(\hat{y}|\hat{x})$.

In the context of this chapter, the objective is to model a document[1] by its word[2] count vector. In other words, the main interest is in the number of times an individual-footprint appears in a week representing the document. Thus, bottom RBM layer is replaced by RSM and stochastic binary units are used for all the hidden layers. In the RSM, each input of the visible units $v_1, ..., v_D$ is scalar values. The inputs of the visible units as

[1] The document represents a week of region-footprint in a borough within NYC.

[2] The word represents individual-footprint captured for particular borough in NYC.

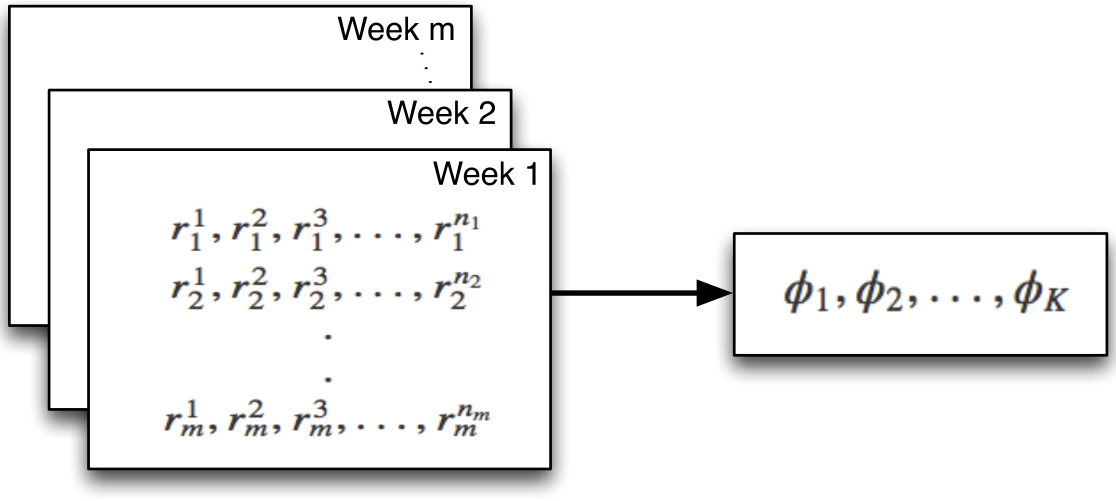

Week m
Week 2
Week 1
$r_1^1, r_1^2, r_1^3, \ldots, r_1^{n_1}$
$r_2^1, r_2^2, r_2^3, \ldots, r_2^{n_2}$
$r_m^1, r_m^2, r_m^3, \ldots, r_m^{n_m}$
$\phi_1, \phi_2, \ldots, \phi_K$

Figure 5.1: Socio-demographic Regional Patterns Inference Problem.

binary vectors forming a matrix U are defined as

$$U = \left\{\begin{matrix} u_{1,1} & u_{1,2} & \ldots & u_{1,D} \\ u_{2,1} & u_{2,2} & \ldots & u_{2,D} \\ . & . & . & . \\ . & . & . & . \\ u_{N,1} & u_{N,2} & \ldots & u_{N,D} \end{matrix}\right\} \tag{5.1}$$

where D indicates the size of the dictionary[3] and N represents the length of the document. Hence, the input vectors are represented as

$$\hat{u}_i = U_{:,i} = [u_{1,i}, ..., u_{N,i}] \tag{5.2}$$

The energy of the RSM is defined as

$$e(U,\hat{h};\boldsymbol{w}) \quad = \quad -\sum_{n=1}^{N}\sum_{j=1}^{M}\sum_{D}^{i=1} W_{ijn}h_jU_{n,i} \quad - \quad \sum_{n=1}^{N}\sum_{i=1}^{D} U_{n,i}b_{n,i} \quad - \quad \sum_{j=1}^{M} h_jaj \tag{5.3}$$

where W_{ijn} is the weight between visible unit i at location n in the document $U_{n,i}$, and hidden unit j [91]. $b_{n,i}$ is the bias of $U_{n,i}$. a_j is the bias of hidden unit j. The conditional distribution of the hidden units h_j and visible units can be computed as

$$p(h_j = 1|U) = \sigma(a_j + \sum_{n=1}^{N}\sum_{i=1}^{D} U_{n,i}W_{ijn}) \tag{5.4}$$

$$p(U_{n,i} = 1|\hat{h}) = \frac{e^{b_{n,i}+\sum_{j=1}^{M} h_jW_{ijn}}}{\sum_{q=1}^{d} e^{b_{n,q}+\sum_{j=1}^{M} h_jW_{qjn}}} \tag{5.5}$$

[3] The dictionary represents all different individual-footprints that exist in the whole dataset.

where σ represents the logistic sigmoid function and Eq. 5.5 denotes the softmax function. It is worth emphasizing that the softmax function can be applied to multinomial distribution only which is what is denoted by U.

The hidden units of RBM are stochastic binary units. The RBM's inference is conducted by finding representation of the hidden layer $\hat{h} = [h_1, ..., h_M]$ that minimizes the energy $e(\hat{v}, \hat{h}; \mathbf{w})$ with respect to the visible layer $\hat{v} = [v_1, ..., v_D]$ [92]. The energy is defined as in [93]

$$e(\hat{v}, \hat{h}; \boldsymbol{w}) = -\sum_{i=1}^{D} b_i v_i - \sum_{j=1}^{M} a_j h_j - \sum_{i=1,j=1}^{D,M} v_i h_j W_{ij} \tag{5.6}$$

where v_i is the state of the visible unit, h_j is the state of the hidden unit, b_i is the bias of the visible layer, a_j is the bias of the hidden layer, W_{ij} is the weight between v_i and h_j and which represents a matrix comprising all the weights and biases. A joint distribution can describe the visible (v_i) and hidden layers (h_j) as

$$p(\hat{v}, \hat{h}; \boldsymbol{w}) \quad = \quad \frac{1}{Z(\boldsymbol{w})} e^{-e(\hat{v},\hat{h};\boldsymbol{w})} \qquad \qquad where \quad Z(\boldsymbol{w}) \quad = \quad \sum_{\hat{v},\hat{h}} e^{-e(\hat{v}\hat{h};\boldsymbol{w})} \tag{5.7}$$

where $p(\hat{v}, \hat{v}; \mathbf{w})$ is called Boltzmann distribution and $Z(w)$ is the partition function used as a normalizing constant for the Boltzmann distribution. The probability the model reconstructs the visible vector $\hat{v}$ is calculated by [91]

$$p(\hat{v}; \boldsymbol{w}) = \frac{1}{Z(\boldsymbol{w})} \sum_{\hat{h}} e^{-e(\hat{v},\hat{h};\mathbf{w})} \tag{5.8}$$

The conditional distribution over the hidden units and visible units are calculated using Eq. 5.9 and Eq. 5.10 respectively.

$$p(h_j = 1|\hat{v}) = \sigma(a_j + \sum_{i=1}^{D} v_i W_{i,j}) \tag{5.9}$$

$$p(v_i = 1|\hat{h}) = \sigma(b_i + \sum_{j=1}^{M} h_j W_{i,j}) \tag{5.10}$$

For training, the derivative of the log-likelihood is calculated with respect to the model parameters $\mathbf{w}$ as illustrated in [91]. Contrastive Divergence is used for approximating the gradient of the objective function as suggested by Hinton in [74]. Hence, the RBM update of the weights and biases is done by

$$\Delta W = \epsilon(\mathbb{E}_{p_{data}}[\hat{v}\hat{h}^T] - \mathbb{E}_{p_{recon}}[\hat{v}\hat{h}^T]) \tag{5.11}$$

$$\Delta \hat{b} = \epsilon(\mathbb{E}_{p_{data}}[\hat{h}] - \mathbb{E}_{p_{recon}}[\hat{h}]) \tag{5.12}$$

$$\Delta \hat{a} = \epsilon(\mathbb{E}_{p_{data}}[\hat{v}] - \mathbb{E}_{p_{recon}}[\hat{v}]) \tag{5.13}$$

where $\mathbb{E}_{p_{data}}[.]$ is the expectation of the joint distribution of the real data, $\mathbb{E}_{p_{recon}}[.]$ is the expectation with respect to the reconstructions and ϵ is the learning rate. The distribution of p_{recon} is calculated using a Gibbs chain running for one iteration, since it has been shown that it works well [74].

After the pretraining is finished with the previous procedures, it is expected that the parameters estimated which will be passed to the finetuning process are already in proximity to a local minima on the error surface. The finetuning process will further apply an optimisation algorithm to adjust these parameters to ensure convergence. *Conjugate Gradient* is used as the optimization algorithm since it has been proved to be faster than the *Gradient Descent* and more robust [94].

5.4 Experiments setup

In this section, the dataset along with its preparations procedures is first described and then the evaluation metric that has been used to evaluate the proposed model is presented.

5.4.1 Dataset description and preparation

The Foursquare [95] is one of the most popular LBSNs services used. In terms of scale, Foursquare claims over 6 million registered users and around 1 million check-ins per day. Foursquare like other services allow users to *check-in* at different venues (e.g., restaurants, museums, home), write comments and tips, and upload pictures and videos about the visited venues. The check-ins filtered within NYC for the dataset introduced in chapter 4 can be viewed in Figure 5.2 as a heat map. From this filtered check-in dataset, a subset of all of these checkins are selected that lies in any of the five NYC regions/boroughs boundaries. The five boroughs of NYC were chosen in order to explore whether a unique socio-demographic patterns could be inferred for each borough, primarily because each borough has different characteristics and unique histories than each other [96]. The five boroughs in NYC are Manhattan, Brooklyn, Queens, Bronx, and Staten Island. Then, the data is is further processed and prepared in the following way:

- **Activity Categorization**: The type of the visited locations is categorized in the dataset for each check-in into 9 categories (Entertainment, Education, Night Life, Recreation, Social Services[4], Residence, Shopping, Travelling, and Eating) as per the Foursquare categorization. This was expected to help significantly decrease the sparsity problem when feeding the data to the DBN as it expects to increase the word count for specific individual-footprints within each borough. Hence, increasing the chances for finding a unique pattern for each borough.

[4] This category involves check-ins related to services provided for the benefit of the community, such as pharmacy, recycling facility, and Laundry service.

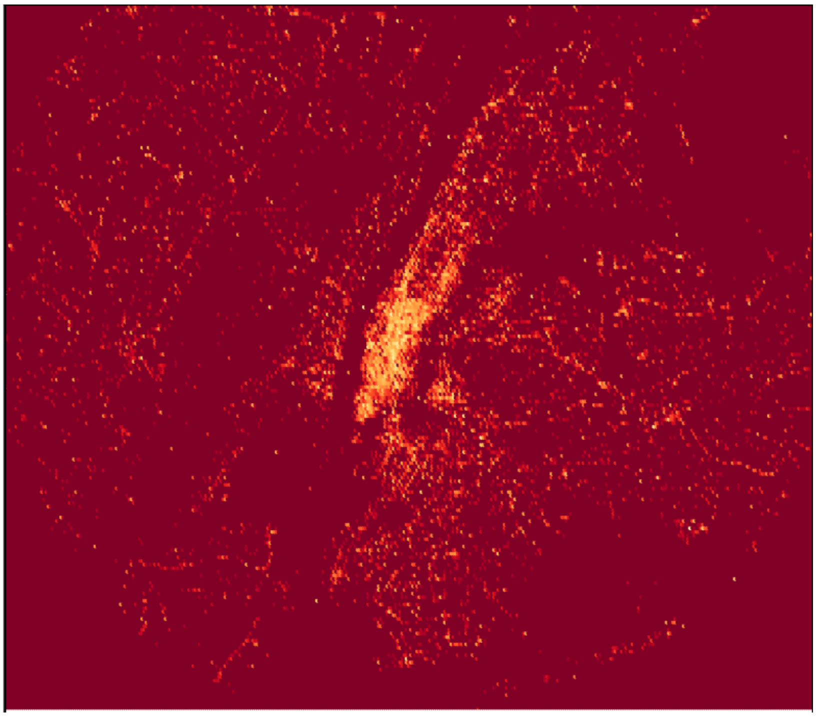


Figure 5.2: Check-ins in New York City (Heat map).

- **Time-span Aggregation**: In this step, each day is divided into 12 time-slots where each time-slot lasts for 2 hours each. In general the longer the slot, the finer the recurrent activity behaviour is captured. Based on the dataset and the conducted experiments, it was found that 12 time slots provides a good tradeoff between decreasing sparsity of the dataset and preserving the fine routine behaviours in the data required for capturing the Socio-demographic Regional Patterns.

In the methodology proposed in the previous section, DBN for topic modeling is developed and proposed to identify if a unique pattern for each of the five boroughs across NYC can be extracted. To do so, an analogy is drawn between discovering socio-demographic patterns of a region and the topic discovery of a document. Specifically in this work, a *word* is represented as the individual-footprint (day of the week, category of activity and time), while a *document* is comprised of the individual-footprints of one week in a borough and the *corpus* is formed of several documents of multi-weekly individual-footprints. There are K latent topics (socio-demographic weekly patterns) in the model, where K is the number of output units of the DBN. For example, a document can comprise the following words: TueTraveling9, TueResidence10, etc. This represents that there was an activity of Travelling on Tuesday from $06PM$-$8PM$ and Tuesday from $08PM$-$10PM$, there was an activity in relation to residence and so on.

5.4.2 Evaluation Methodology

In this section, the evaluation methodology is defined for measuring the accuracy of the proposed DBN model for discovering socio-demographic patterns. The data is split into training dataset 70% and test dataset 30% where each document represents one week activity pattern of one borough, this split ratio is common in several machine learning tasks [97][98][99]. The selection of the documents for the two sets was performed on the basis that both have an equal proportion of different labels. The test dataset will be used only for evaluating the model and not for training purposes. A forward-pass is performed for the test dataset on the trained DBN and generates a K dimensional vector with the length equivalent to the number of output units of the DBN. To compute the accuracy for a test document, the distance proximity is calculated between the nearest neighbours and the query/test document using *Euclidean distance.* Hence, the accuracy can be defined as the fraction between the number of neighbours belonging to same class of the test document to the total number of neighbours queried as

$$Accuracy = \frac{\text{no. of true labeled docs}}{\text{no of docs queried}} \tag{5.14}$$

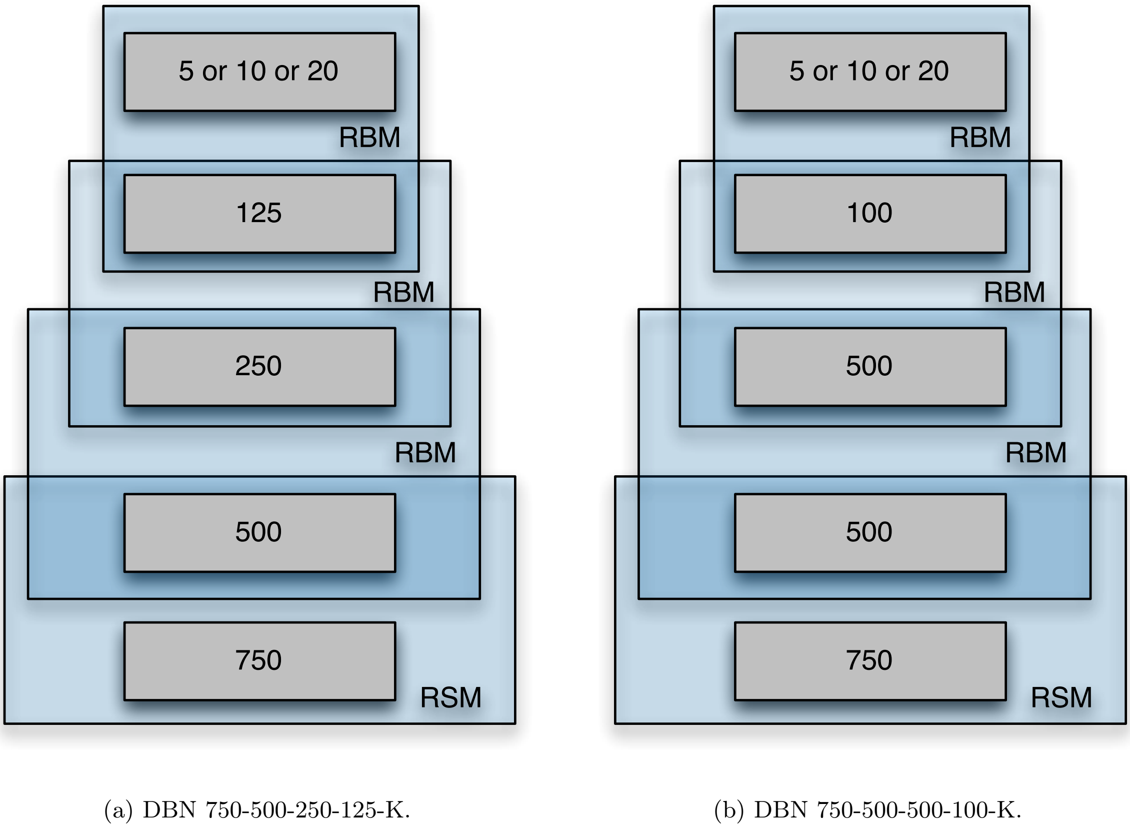
This evaluation is performed for a different number of neighbours on $\{1,\ 3,\ 7,\ 15\}$. The accuracy measurement evaluates the probability for similar week patterns (where each week is a document) taken from the same borough as the query week. This gives an indication of how well the weekly activity patterns of each borough can be spread. Hence, the higher the accuracy, the more obvious and unique socio-demographic patterns for each borough.

5.5 Results & validation

In this section, the proposed Socio-demographic Regional Patterns model is evaluated and the generated results is further validated using LDA and KL-Divergence. The choice of these is based on the fact that LDA is a well-known topic model that was used previously for extracting urban patterns [38] and KL-Divergence is a established method for choosing the optimum number of topics [100]. Further, the results of different DBNs architectures are shown highlighting the most accurate model. Last, the unique patterns extracted from applying DBN is validated and interpreted by applying LDA.

5.5.1 Results

The size of the dimensionality reduction of the DBN (number of output units in the DBN which we refer to as K) must be decided empirically. Hence, different DBN architectures are applied and the reconstruction errors are analyzed when applied to the dataset for choosing the optimal K. Four-layered DBN with two different architectures has been used in the experiment as illustrated in Figure 5.3. Based on our dataset, 750 input



(a) DBN 750-500-250-125-K.

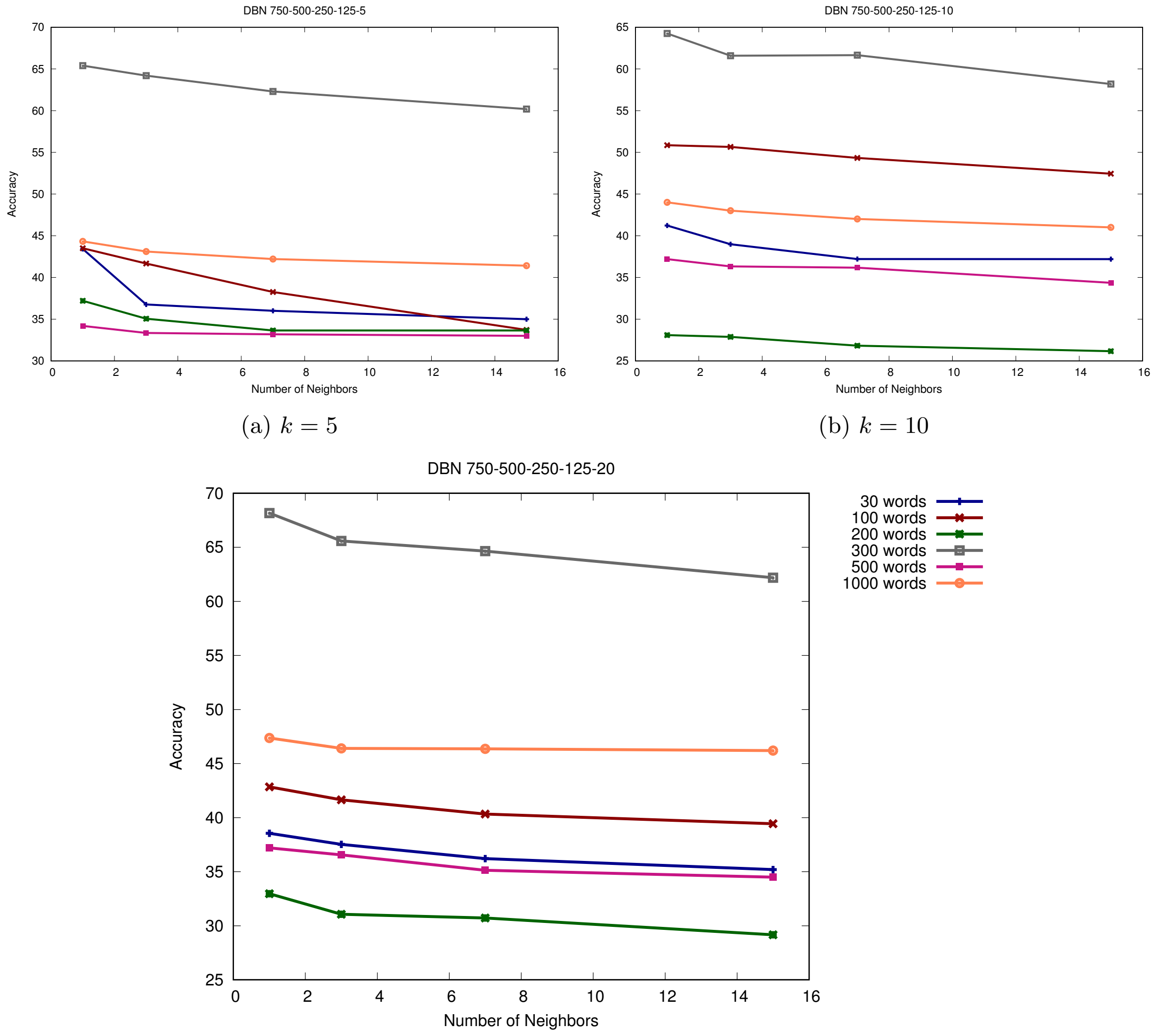
(b) DBN 750-500-500-100-K.

Figure 5.3: DBN Architectures.

data points $\{x_1, ..., x_{750}\}$ are passed through the DBN and the output layer is varied with 5, 10 and 20 units. These two DBNs architectures are referred to as 750-500-250-125-k and 750-500-500-100-k where k is the number of output units which represents the latent representation of a specific borough of NYC.

There are several input parameters that can be adjusted and tuned when training the DBN and there is no proof on an optimal adjustment for the structure of the DBN for best performance. Hence, different combinations are experimented of the following parameters seeking the optimum structure for our problem and dataset. The learning parameters of the pre-training are set with a learning rate $\epsilon = 0.01$, momentum $m = 0.9$, and a weight decay $\lambda = 0.001$. The weights are initialized with variance 0.01 from a 0-mean normal distribution. Furthermore, 50 epochs are repeated and all biases are initialized to 0. For finetuning, large batches are set to 10 and four line searches are performed for the Conjugate Gradient algorithm with 50 epochs. These parameters are found to be stable for our experiments and close to what has been used in [101] that have proved to be stable on different popular datasets. It is worth highlighting that when the number of epochs is increased to 100, there was not much difference in the performance observed and hence, the 50 epochs was found to be the optimum as it is obviously faster.

To further decrease the sparsity of the dataset and increase the possibility of finding unique region-footprint pattern for each of the boroughs, six samples from the original



(a) $k = 5$

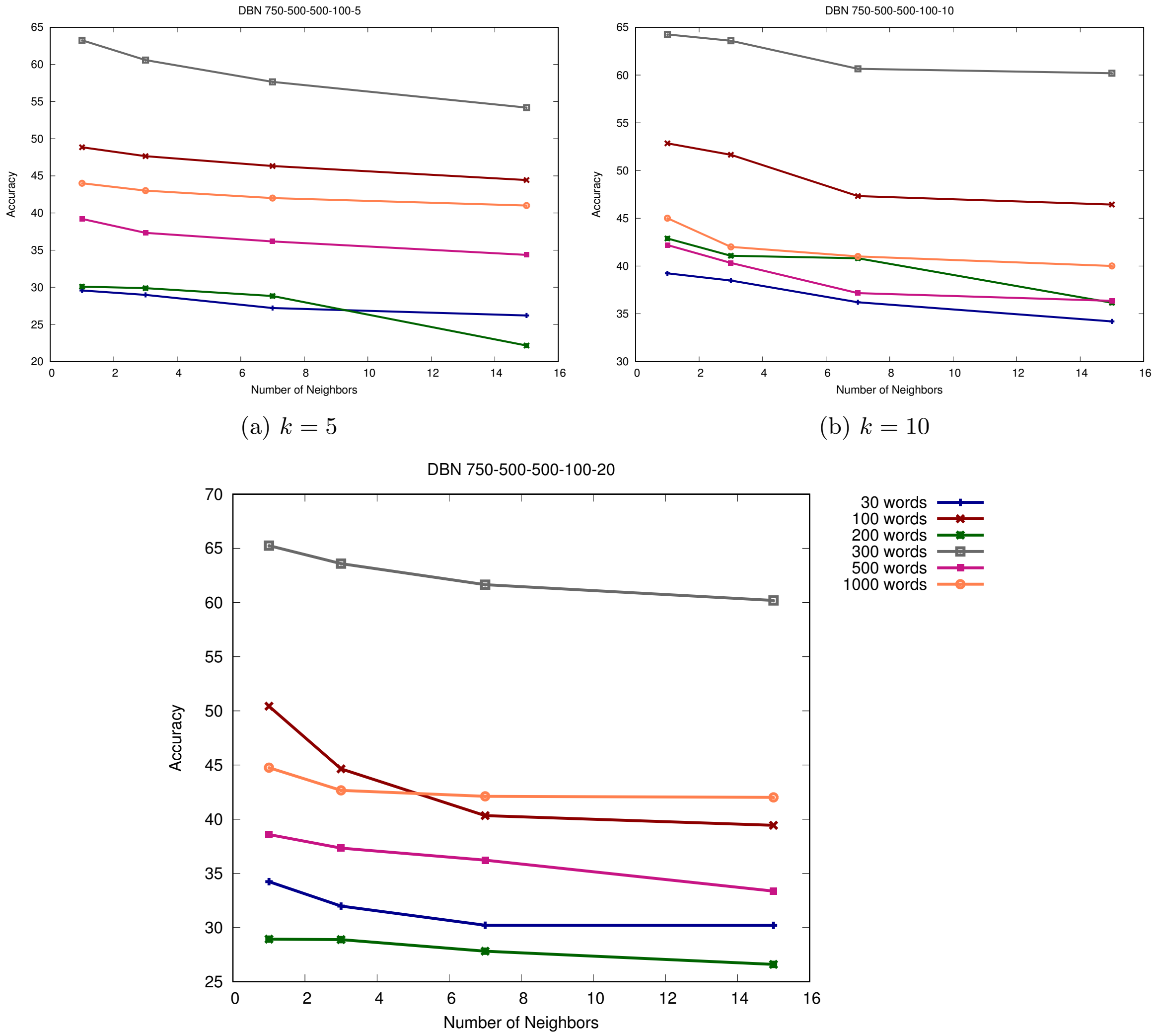
(b) $k = 10$

(c) $k = 20$

Figure 5.4: DBN 750-500-250-125-k.

dataset were created where each sample was filtered with a minimum number of words per each document (one week of individual-footprints). For example, the maximum words of 300 per document, indicates that all weeks/documents that have less than 300 individual-footprints are removed. The 6 new sampled datasets created were equivalent to documents of at least 30, 100, 200, 300, 500, and 1000 words.

Figure 5.4 and Figure 5.5 show the results for training the DBNs for both architectures. When the number of output units is increased from 5 to 10 for both architectures, an obvious increase is observed in the accuracy for the 5 sampled datasets. However, the increase is not that obvious when moving from 10 to 20 output units which indicates saturation in performance. Interestingly, the new sampled dataset with 300 minimum number of words is observed to outperform in both architectures in all of the cases regardless of the number of output units. This can be interpreted as being because of the sparsity problem. By removing the documents/weeks of less than certain number of individual patterns, as the sparsity problem decreases until reaching a certain limit and then it starts



(a) $k = 5$

(b) $k = 10$

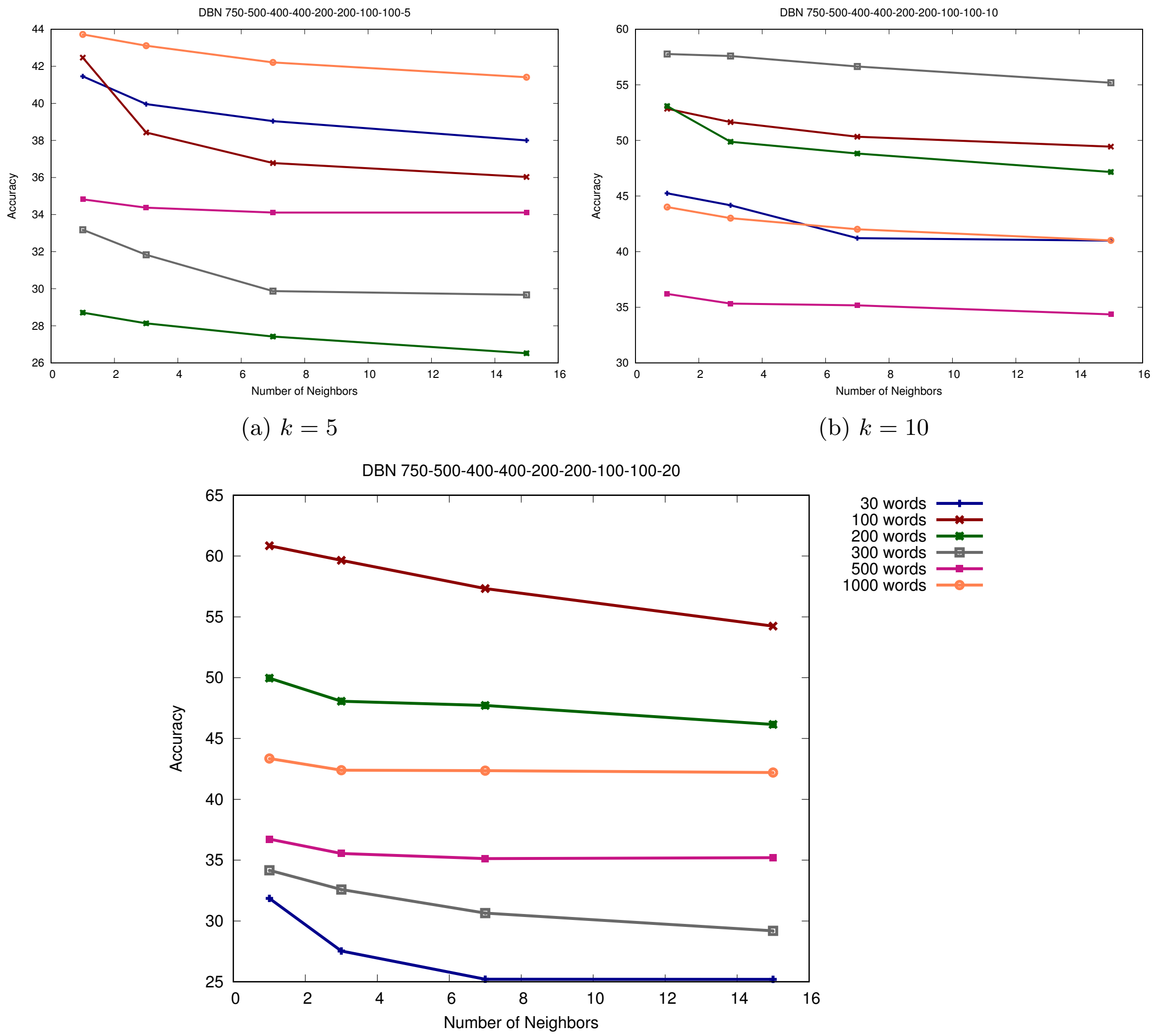
(c) $k = 20$

Figure 5.5: DBN 750-500-500-100-k.

to increase again due to the decrease in the number of documents to the extent that it impacts negatively upon the generalization and increases the chances of overfitting. It is worth emphasising that in the case of 300 words upwards, there were no documents to process for the Staten Island borough.

For the DBN 750-500-250-125-k, the best accuracy achieved is shown in Figure 5.4c with 68.17%, 65.59%, 64.65%, and 62.19% for the 1, 3, 5 and 7 neighborhoods respectively for the 300 words dataset. On the other hand, DBN 750-500-500-100-k shows slightly less accuracy equivalent to 65.25%, 63.59%, 61.66%, and 60.29% for the 1, 3, 5 and 7 neighborhoods for the 300 words dataset as shown in Figure 5.5c.

It is worth highlighting that when doubling the number of hidden layers from 3 to 7, there has been no any significant increase in the performance observed as illustrated in Figure 5.6 with a maximum accuracy of 61.03%. This indicates that there is no direct proportional relationship between the number of hidden layers and the accuracy.



(a) $k = 5$

(b) $k = 10$

(c) $k = 20$

Figure 5.6: DBN 750-500-400-400-200-200-100-100-k.

5.5.2 Validation

In this work context, it would be of interest to validate and better understand the unique patterns between the different boroughs within NYC. For doing so, LDA topic model is trained on the data filtered for 300 words which show the best accuracy in our experiments. For choosing the optimum number of topics for training LDA, the approach presented in [102] is applied where LDA is executed for number of topics ranging from $1 - 100$ and the symmetric *Kulback-Leibler (KL) divergence* is computed of the singular value distributions of matrix $M1$ and the vector distribution $L * M2$ where $M1$ and $M2$ can be viewed as the matrices generated from the LDA matrix factorization methods. These methods factorize the document-word frequency matrix and L is a vector containing the length of each document/weekly-footprints in the corpus formed of the full weekly-footprints of 1 borough. The higher the KL-Divergence, the less optimum number of topics. The advantage of this approach is that it makes use of the properties from both matrices $M1$ and $M2$ whereas other approaches proposed in [103] and [104] only consider the information in the stochastic topic-word and ignore the document-topic matrix. The results computed are shown in Figure 5.7a, Figure 5.7b, Figure 5.7c and Figure 5.7d for Bronx, Brooklyn, Manhattan and Queens boroughs respectively.

As it can be observed in the graphs of Figure 5.7, the optimal number of topics is relatively low for all boroughs datasets. It can be observed from Figure 5.7c and Figure 5.7d that the optimum number of topics is between 5 and 20. However, from Figure 5.7c and Figure 5.7d, the optimum number of topics seems to be between 5 and 15. Hence, 10 topics were selected to train the LDA for the 4 boroughs' datasets. Table 5.1 and Table 5.2 show the results when applying LDA for discovering the top 10 topics. Table 5.1 shows the first 5 topics results whilst Table 5.2 shows the last 5 topics. The results of three Boroughs are shown since Bronx results are quite close to Queens results. Some interesting common patterns found from the results are as follows:

- **Brooklyn**: The main pattern captured in Brooklyn was in relation to *eating* activities with all afternoon starting from $04PM$ onwards. The overall probability for the eating activities reaches 0.45.

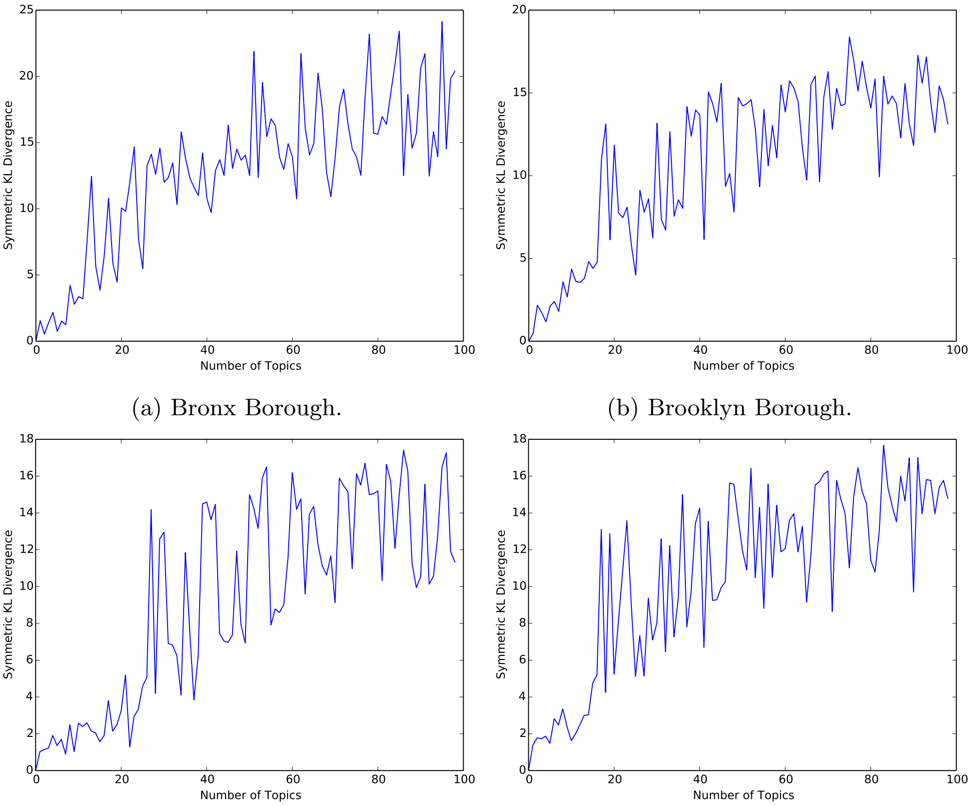
- **Manhattan**: The main pattern captured in Manhattan was in relation to *social services* activities with most captured around noon from $12PM$-$02PM$ with an overall probability of 0.38.

- **Queens**: The main pattern captured in Queens was in relation to *traveling* activities on weekdays and weekends with an overall probability of 0.47.

In summary, this section can be concluded with the following two main points:

1. In general, identifying unique patterns for different regions within cities is considered a complex problem that requires advanced machine learning models for being able to automatically learn features at multiple levels to learn the complex Socio-demographic Regional Patterns without depending on human crafted features. This was achieved via training the DBN with multiple layers. The complexity of inferring these patterns are validated by the low probabilities when LDA is applied for inferring regional unique weekly patterns within each borough. This is obvious from the results shown in Table 5.1 and Table 5.2.

2. Although, the low probabilities observed when LDA is applied indicates the complexity of detecting a unique pattern for each of the boroughs from the complete individual-footprints (e.g., friEating10), a clear unique pattern is observed in relation to the activity categories which validated that there is at least a different pattern for each of the boroughs.



(a) Bronx Borough.

(b) Brooklyn Borough.

(c) Manhattan Borough.

(d) Queens Borough.

Figure 5.7: Number of Topics Vs Symmetric KL Divergence for NYC Boroughs.

Table 5.1: Recurrent Region-footprint results for the first five topics (1-5).

Topics	**Topic 1**		**Topic 2**		**Topic 3**		**Topic 4**		**Topic 5**	
Boroughs	Word	Prob	Word	Prob	Word	Prob	Word	Prob	Word	Prob
Brooklyn	friEating10	0.017	monTraveling6	0.018	satEating0	0.001	thuEating11	0.011	sunEating8	0.010
	sunEating1	0.016	satEating11	0.016	sunEating8	0.001	sunEating9	0.011	thuEating0	0.008
	sunEating9	0.011	satEating10	0.011	satEating11	0.001	satEating0	0.011	satEating0	0.007
	satNighlife2	0.011	wedEating0	0.011	monTraveling6	0.001	thuShopping11	0.011	sunEating10	0.007
	satShopping11	0.010	sunEating0	0.011	satEating8	0.001	sunEating8	0.011	wedEating11	0.006
Manhattan	tueSocialServices6	0.015	thuSocialServices6	0.026	monSocialServices6	0.026	monSocialServices7	0.015	tueSocialServices6	0.019
	tueEating8	0.013	thuEating6	0.018	tueSocialServices6	0.015	friEating0	0.015	monSocialServices6	0.018
	tueEating9	0.012	monSocialServices6	0.017	sunEating8	0.015	thuEating0	0.015	thuEating8	0.016
	thuEating0	0.011	thEating8	0.015	sunEating9	0.014	thuEating9	0.013	tueEating8	0.015
	tueSocialServices7	0.011	wedEating11	0.013	monEating9	0.013	wedEating9	0.011	friSocialServices6	0.015
Queens	monTraveling6	0.015	monTraveling5	0.023	friTraveling5	0.009	tueTraveling6	0.025	wedTraveling6	0.017
	monTraveling5	0.010	tueTraveling6	0.018	satShopping9	0.009	monTraveling6	0.015	monShopping9	0.017
	monShopping10	0.009	monTraveling0	0.013	tueTraveling6	0.009	friTraveling6	0.014	wedTraveling5	0.016
	thuTraveling6	0.008	tueTraveling5	0.013	satEating9	0.009	monTraveling11	0.012	wedTraveling8	0.013
	friTraveling6	0.007	monTraveling6	0.012	thuTraveling6	0.008	sunShopping9	0.011	wedEducation6	0.013

Table 5.2: Recurrent Region-footprint results for the last five topics (6-10).

Topics	**Topic 6**		**Topic 7**		**Topic 8**		**Topic 9**		**Topic 10**	
Boroughs	Word	Prob	Word	Prob	Word	Prob	Word	Prob	Word	Prob
Brooklyn	sunEating8	0.027	tueTraveling5	0.012	huTraveling6	0.010	satEating11	0.001	satEating0	0.001
	satShopping8	0.026	sunEating0	0.011	wedTraveling6	0.010	sunEating0	0.001	sunEating0	0.001
	satEating8	0.024	tueEating0	0.09	thuEating9	0.009	sunEating8	0.001	satEating9	0.001
	satRecreation8	0.017	satEating11	0.08	monTraveling6	0.009	sunShopping9	0.001	satEating8	0.001
	satShopping10	0.016	sunShopping10	0.008	wedEating9	0.009	tueTraveling0	0.001	satShopping9	0.001
Manhattan	monSocialServices6	0.086	satEating0	0.016	wedEating8	0.018	monSoicalServices6	0.018	monSocialService6	0.014
	wedEating8	0.073	friEating0	0.013	tueSocialServices6	0.017	monEating8	0.018	friEating8	0.013
	monEating6	0.040	friEating8	0.013	wedSocialServices6	0.012	sunNightlife1	0.013	thuEating8	0.013
	wedShopping8	0.033	satNighlife2	0.013	sunEating9	0.011	wedSocialServices6	0.013	monEating8	0.012
	monShopping10	0.033	sunEating0	0.013	sunEating11	0.011	tueEating11	0.013	satEating0	0.011
Queens	tueTraveling6	0.018	thuTraveling5	0.013	monTraveling6	0.001	tueRecreation2	0.020	monTraveling6	0.001
	tueTraveling0	0.014	wedTraveling11	0.011	friTraveling5	0.001	tueTraveling9	0.015	wedTraveling6	0.001
	wedTraveling11	0.013	friTraveling5	0.010	tueTraveling6	0.001	tueRecreation1	0.014	friTraveling5	0.001
	tueTraveling5	0.013	tueTraveling11	0.009	monTraveling7	0.001	wedEating0	0.012	tueTraveling6	0.001
	wedTraveling6	0.011	thuTraveling6	0.001	tueRecreation0	0.001	tueRecreation0	0.011	wedTraveling11	0.001

5.6 Summary

In this chapter, the possibility of classifying new weekly pattern to one of the cities' regions has been explored using a proposed trained DBN based model. The five boroughs in New York City (NYC) have been considered as an example for demonstrating that it is feasible to extract unique patterns for each of the boroughs that differ from one another. To the best of our knowledge, there has been no previous research that shows the ability to extract such unique patterns of regions within the same city and moreover, DBNs have not been leveraged before for extracting urban patterns in cities. More specifically, in this chapter, it is shown that the best proposed trained model of DBN is able to achieve nearly 70% accuracy for classifying/predicting the region based on new/unseen weekly crowd activities in the city showing that discovering such type of patterns is feasible. The proposed pattern and extraction method is likely to find a natural application and impact for better understanding the commonalities between socio-demographics between regions across the globe. In addition it is anticipated that it will support the understanding as to whether there are any correlations between these patterns with other urban parameters such as energy consumption, economic development and industrialization.

Having presented in this chapter the first introduced contribution and urban pattern in this thesis referred to as *Socio-demographic Regional Patterns*, the next chapter introduces a clustering based approach for extracting the second type of urban pattern called *Temporal Functional Regions patterns.*

Chapter 6

Extracting the new Temporal Functional Regions patterns

Chapter overview: *In this chapter, the second type of new urban pattern proposed in this thesis is introduced titled as "Temporal Functional Regions patterns". It is described how it is possible to extract for the first time, using a clustering based approach, finer functional regions in cities that change across space and time. Various time intervals are studied for identifying the most suitable time granularity for extracting such Temporal Functional Regions. This new approach has been published in [ICTAI 2016]. Furthermore, It is demonstrated in this chapter as to how the approach has been applied to studying some specific regions in Manhattan, where it is validated that the proposed approach follows our intuitive understanding for the functionality of some of those regions.*

6.1 Motivation

Urban planning focuses on planning the land use in cities. Urban planners determine the functionality of regions within cities for being able to design urban environments and make the best use of the available spaces to increase the well being of citizens. For achieving this purpose, urban planners require a large amount of data on urban land use that is typically gathered from direct observations or questionnaires that captures how citizens interact with the urban environment. However, this approach has some obvious limitations in relation to the cost of running surveys and gathering such amounts of data, besides privacy concerns from citizens for providing such information (refer to chapter 2). An alternative approach for capturing functional regions and land use is Geographic Information Systems (GIS) which provides satellite imagery that has the possibility to capture land use through advanced vision techniques. Nevertheless, such techniques fail to capture real time information as satellite images are not captured frequently.

The approach presented in this chapter utilizes both spatial (geo-tagged location) and temporal (time-stamped) information, *highlighting for the first time how the functionality of a region can vary temporally.* As an example, a region within a city can exhibit shopping functionality in the afternoon, whereas at ight it can turn to be more of a residential area. In another region, it might be considered a business district in the morning if it is a home for lots of companies while later at night it might have night-life functionality due to the presence of lots of restaurants and bars in the same region. This logical and intuitive inference motivated my research into extending and applying clustering techniques to account for the temporal variations, in order to characterize the region during different time slots of the day. This is precisely the goal of the proposed approach in this chapter which is a step towards urban computing [35] and helps in understanding urban dynamics for cities. The following specific contributions of this part of the research are highlighted as follows:

1. It is shown for the first time that taking into account temporal variation to characterize a region's functionality at different slots during the day has the potential to detect variability of the functionality of the same region within the day. It is anticipated this may yield a better understanding of city dynamics compared to the default "no temporal split" approach.

2. Different time intervals have been studied in order to explore the optimum interval for detecting Temporal Functional Regions within the day. This has allowed the exploration of the tradeoff between sparsity of data and detecting meaningful functionalities of a region. The proposed approach has been successfully applied on a sparse dataset coming from LBSNs, as described in chapter 4.

3. Three different clustering based techniques are developed for detecting Temporal Functional Regions deriving the optimum number of clusters for each time interval.

4. Results of applying the approach to regions are likely to find a natural application in providing better personalized recommendations to users based on the variability of the regions' functionality they visit, as well as supporting the understanding of commonalities of socio-demographics between regions across the globe.

6.2 State-of-the-Art

Over a number of years, many approaches have been proposed for exploring functional regions in a city using socio-demographic data [105]. However, the process of acquiring and updating such data is very expensive and time consuming [105]. Hence, other alternative data sources have been attempted including human mobility data, survey data, and most recently LBSNs data as overviewed in chapter 2. In this section, the attempts within the state-of-the-art to discover functional regions from different types of data are focused upon.

6.2.1 Human Mobility & Survey Data

In this branch of research, there have been several attempts for utilizing the human mobility data to explore spatio-temporal patterns to infer the functions of regions using mobile phone Erlang data [106][107], taxicab data [108][109], wifi data [109] and smart card data [110]. Besides the human mobility data, some other studies have relied on using activity-based survey data to explore the spatial-temporal patterns and then derive the functional regions of a city [109]. Unfortunately, human mobility and survey data have their own limitations in the context of deriving functional regions. Human mobility data is subject to a lack of travel demand information which consequently impacts the detailed characteristics of regions that can be inferred. Hence, empirical analysis is needed to infer cluster of regions' type leading to the inability to distinguish between the likes of non-home/work activities [109]. To overcome these challenges, Yuan et al. used both information about human mobility among regions and Point Of Interests (POIs) located in a region [109]. They introduced a topic-based inference model which links the human mobility with POIs, however, this is not always be the case in real scenario. For instance, one user goes to a library that is beside a shopping center. If the library is not in the POI data, then her movement will be linked to shopping rather than the educational purpose.

6.2.2 LBSNs Data

With the rise of LBSNs such as Foursquare, Twitter and Flickr, it is feasible to record a user's surrounding along with their movement routes through what is so-called "check-in". Unlike data from cell phone and car trajectories data, check-in data not only contain the location but also the activity category of the user. False check-ins is one of the obvious challenges in relying on this data but Cheng et al. [109] proposed a series of rules to eliminate false check-ins and We et al. [111] introduced five criteria to discover the fake/untrusted check-in.

Check-in data has shown to be a great source for discovering functional regions compared to previous types of data. Justin et al. [112] relied on LBSNs data to discover sub-urban areas from foursquare data referred to as *Livehoods*. Felix et al. [45] combined both textual and movement data from LBSNs dataset to obtain semantically rich modalities of urban dynamics. One of their contributions was clustering city areas into functional regions relying on analogous results that can be obtained to represent areas of similar functions for particular time periods. Thiago et al. [113] measured dynamics of eight cities on a large scale using LBSNs data. However, they did not consider the interdependence between the functional regions and human activities which has been studied in detail by Ye et al. in [114].

Though, prior works for exploring functional regions are separated based on the data source (Human Mobility data, Survey data or LBSNs data), to our knowledge, no prior work using LBSNs data has attempted to study functional regions whilst taking into account temporal variations to characterise regions at different time slots during the day. In contrast in the research I have undertaken and described in this chapter, (a) it is shown how to derive finer functional regions during the day in a city by studying different time slots of the day and (b) the performance derived from using different clustering techniques is demonstrated.

6.3 Temporal functional regions patterns

In this section, the notation used in the proposed approach is introduced first. Then a description of the proposed approach for extracting Temporal Functional Regions is discussed highlighting the method for deriving the optimum: number of clusters, clustering method as well as the number of slots for splitting the dataset.

6.3.1 Notation and Definitions

The raw LBSNs dataset is denoted with D. Moreover, the dataset is split into d days, which are further split into s time slots of equal sizes, such that $d = \langle t_1, t_2, ...t_s \rangle$. In this work, the optimum number of slots is studied to split the initial dataset d from a set of candidates: $S = \{2, 4, 8, 12, 24\}$. For instance for $s = 2$, the dataset d is split into two time slots of $12h$ each. The dataset for a day d remains of $24h$ duration when s is equal to 1. Further, each time interval t_i has a duration of: $\|t_i\|= \frac{24}{s}$, $\forall i \in [1, s]$.

In order to determine the functional regions of a city, a set of 9 activity categories have been formulated (e.g., entertainment, education), as follows: $A = \{a_1, ...a_9\}$. These are also referred to as functions. Moreover, $act(r_j)$ denotes the main activity related to region r_j. In addition, a checkin within a region r_j and time-slot t_i is denoted with $c_{t_i}^{r_j}$. The activity related to the checkin $c_{t_i}^{r_j}$ is denoted with $act(c_{t_i}^{r_j})$.

Further, $\mathcal{C}_{t_i}^{r_j}(a_k)$ is defined as the set of relevant check-ins to a specific activity a_k within region r_j and during time-slot t_i as follows:

$$\mathcal{C}_{t_i}^{r_j}(a_k) = \{c_{t_i}^{r_j} \mid act(c_{t_i}^{r_j}) = a_k, \; \forall a_k \in A \;, \; \forall \; i \in [1, s] \;, \; \forall j \in [1, m]\} \tag{6.1}$$

The number of check-ins relevant to a specific activity a_k within region r_j and during time-slot t_i is referred by the number of elements within the associated set $\|\mathcal{C}_{t_i}^{r_j}(a_k)\|$.

Definition 1 (Checkins Matrix $\mathcal{C}_{t_i}$) *A Checkins Matrix is defined as a matrix containing the checkins relevant to each activity within each region r_j at interval t_i, as follows:*

$$\mathcal{C}_{t_i} = \begin{Bmatrix} \mathcal{C}_{t_i}^{r_1}(a_1) & \mathcal{C}_{t_i}^{r_1}(a_2) & ... & \mathcal{C}_{t_i}^{r_1}(a_9) \\ \mathcal{C}_{t_i}^{r_2}(a_1) & \mathcal{C}_{t_i}^{r_2}(a_2) & ... & \mathcal{C}_{t_i}^{r_2}(a_9) \\ . & . & . & . \\ . & . & . & . \\ \mathcal{C}_{t_i}^{r_m}(a_1) & \mathcal{C}_{t_i}^{r_m}(a_2) & ... & \mathcal{C}_{t_i}^{r_m}(a_9) \end{Bmatrix}, \; \forall\, i \in [1, s] \tag{6.2}$$

Moreover, the set of all checkins for d, $\forall t_i \in d,\; i \in [1, s]$ *is denoted by* $\mathcal{C}_d$*.*

Further, the input vector for each region r_j is represented during a time interval t_i as the cardinalities of each element of a row of the matrix $\mathcal{C}_{t_i}$, as follows:

$$\vec{\mathcal{C}}_{t_i}(r_j) =< \|\mathcal{C}_{t_i}^{r_j}(a_1)\| \;, ..., \; \|\mathcal{C}_{t_i}^{r_j}(a_9)\|>, \; \forall\, i \in [1, s] \; \wedge \; \forall\, j \in [1, m] \tag{6.3}$$

where $\|\mathcal{C}_{t_i}^{r_j}(a_k)\|$ represents the number of checkins relevant to each activity a_k within region r_j at interval t_i, $\forall j \in [1, m] \; \wedge \; \forall\, i \in [1, s]$.

Definition 2 (Static Functional Region fr^{a_k}) *A static functional region is defined as a set of regions that change their functionality over space only, depending on the spatial distribution of the checkins contained in dataset D. The functional region is associated to a single activity and does not consider any shifts in functionality across the time dimension.*

Definition 3 (Temporal Functional Region $fr_{t_i}^{a_k}$) *A temporal functional region is defined as a set of regions that change their functionality over space and time, depending on the activity shifts across the temporal variation of the different time slots considered. A functional region over an interval* t_i *for an activity* a_k *is denoted by* $fr_{t_i}^{a_k}$*, and defined as a set of regions* r_j *with the same function* a_k *over the time interval* t_i*, as follows:*

$$fr_{t_i}^{a_k} = \{r_j \mid act(r_j) = a_k, \; \forall j \in [1, m]\}, \; \forall\, i \in [1, s] \; \wedge \; \forall\, a_k \in A \tag{6.4}$$

Moreover, all the functional regions generated by the clustering techniques for an interval t_i are denoted by: $\mathcal{F}r_{t_i} = \{fr_{t_i}^{a_1}, fr_{t_i}^{a_2}, ..., fr_{t_i}^{a_9}\}, \; \forall\, i \in [1, s]$.

6.3.2 Proposed Approach

This section describes the proposed approach for generating spatio-Temporal Functional Regions from raw LBSNs data D. Three algorithms are described. The proposed approach is based on clustering the regions for detecting Temporal Functional Regions, deriving the optimum number of clusters for each time interval. This is achieved by applying the clustering techniques by varying the number of clusters on each of the time slots and generating the corresponding evaluation metrics – *silhouette coefficient*, *sc*. The latter, *sc*, is used as the metric for identifying the quality of the clustering results. The silhouette coefficient *sc* is chosen due to two obvious reasons. First, the silhouette score is bounded between -1 for incorrect clustering and $+1$ for highly dense clustering, while a score around 0 indicates overlapping clusters. Second, the silhouette score is high when the clusters are well separated and dense which follows the concept of *cluster*. Hence, if most objects have a high value, then it indicates an appropriate configuration for clustering. The silhouette coefficient sc for a single sample can be expressed as:

$$sc = \frac{b - a}{\max(a, b)} \tag{6.5}$$

where a is the mean distance between a sample and all other points in same cluster while b is the mean distance between a sample and all other points in the next nearest cluster. The silhouette coefficient is used to estimate the performance of the generated functional regions' clusters based on the euclidean pairwise difference of between and within cluster distances. Additionally, it is used to identify the optimal number of clusters, which is essential for the three clustering techniques used, by maximizing the value of this coefficient.

Optimum number of clusters (Algorithm 1): Given each day d of dataset D, initially, the input data will be aggregated for each time-slot and mapped into a number of keywords for each checkin. The keywords will further be processed over the given time slots and over the multiple physical regions to form a matrix of checkins per activity and region for each time-slot. This data preparation step will give us $\mathcal{C}_d$, the matrix containing the checkins relevant to each activity within each region for all time intervals in d (refer to the previous notation section). Further, this matrix is given to Algorithm 1, which based on the input matrix returns the number of optimal clusters for each time interval t_i of d, given a clustering technique *method* and the number of splits s for the time interval. The optimum number of clusters is explored by varying k between 2 and 14 (line 5), as it was found through experimentation that going further with k did not bring significant improvements in the sc results. Three of the most well known mature clustering techniques are explored in the experimentation, denoted by *method* in Algorithm 1 listing specifically: (i) Agglomerative hierarchical clustering, (ii) K-means clustering, and (iii) Spectral clustering. Agglomerative Hierarchical clustering considers that each item is in its own cluster, it merges nearest clusters and further iterates this process until only one cluster is left. Spectral clustering can be thought as a "pre-processing" step to change the feature representation before passing the new representation to K-means. K-means is a widely known technique which assumes that the number of clusters k is known and is an iterative algorithm which tracks the cluster means. It is worth highlighting that spectral clustering in particular have been widely used in urban computing generally and in particular for detecting functional regions as shown in [4][112]. Refer to chapter 3 for more details about the clustering algorithms used in this chapter.

Algorithm 1 calculates the optimum number of clusters for each time-slot such that it maximizes sc (lines 4 to 11). Moreover, the optimum number of clusters for all time slots is calculated by averaging the optimum number of clusters for each time-slot (line 12).

```
// Phase 1: Traverse time slots $t_i$ and find optimum number of clusters maximizing
    $sc$
$map(t_i, [method, opt_i, sc_{i_{max}}]) \leftarrow \emptyset$ ;
for $i \in [0, s]$ do
    $\mathcal{C}_{t_i} \leftarrow$ subset($\mathcal{C}_d$,$t_i$);
    // Initialize maximum silhouette score and optimal number of clusters for $t_i$
    $sc_{i_{max}} \leftarrow -1$;
    $opt_i \leftarrow -1$;
    // Find the optimal number of clusters $opt_i$ from $k \in [2, 14]$
    for $k \in [2, 14]$ do
        $\mathcal{C}lusters \leftarrow$ cluster($\mathcal{C}_{t_i}$, $k$, $method$);
        if $sc(\mathcal{C}lusters) > sc_{i_{max}}$ then
            $sc_{i_{max}} \leftarrow sc(\mathcal{C}lusters)$;
            $opt_i \leftarrow k$;
            $map_{t_i}.put(t_i, [method, opt_i, silc_{i_{max}}])$;
        end
    end
end
$opt \leftarrow$ round($\frac{1}{s} \cdot \sum_{i=1}^{s} opt_i$);
return $opt$, $map_{t_i}$;
```

Algorithm 1: opt#Clusters($\mathcal{C}_d$,*method*, s)

The algorithm also constructs a map denoted by map_{t_i}, which stores for each time interval t_i (i.e., the key of the map) a list of values: a clustering method, and its associated optimum number of clusters opt_i and the maximum silhouette coefficient $sc_{i_{max}}$. Its structure is: $map(t_i, [method, opt_i, sc_{i_{max}}])$. For instance, the map is illustrated in Figure 6.1 and Figure 6.2 for a slot $s = 4$ and $s = 2$ respectively (i.e., $d = \langle t_1, t_2 \rangle$ and the duration of each t_i is 12h or 6h) depicting the silhouette coefficient for each k. This information on the optimum number of clusters is later used to determine the optimal clustering method in Algorithm 2.

Optimum clustering method and number of slots to split the dataset (Algorithm 2): Further, in order to derive the optimum method for clustering the considered dataset and the optimum number of slots s to divide the dataset d, map_{t_i} is constructed for each potential number of slots s for splitting the d dataset by applying Algorithm 1 receiving as variation all potential values of s. This leads us to a second map, denoted further by $map^{s}_{t_i}$ which stores for each slot size s (i.e., the key) map_{t_i}. Its structure is: $map(s, map(t_i, [method, opt_i, sc_{i_{max}}]))$.

The algorithm for determining the optimum method and slots is presented in Algorithm 2 and receives as input the map $map^{s}_{t_i}$, and the set S, which contains the explored values of slots s. It traverses all potential values of s and further verifies the maximum sc coefficient across all number of slots s and all clustering methods considered (lines 4 to 9). If the difference between the maximum sc score achieved and closest but less optimal sc score is less than the threshold $\tau = 0.001$, the algorithm proceeds with the already found number of slots as described above (line 9).

```
// Phase 2: Traverse $map^s_{t_i}$ and find optimum number of splits and clustering
   method maximizing $sc$
$method_{opt} \leftarrow' \ ';$
$s_{opt} \leftarrow -1;$
for $s \in S$ do
    $sc_{max} \leftarrow -1;$
    $map_{t_i} \leftarrow map^s_{t_i}.get(s);$
    for $i \in [0, s]$ do
        $method_i \leftarrow map_{t_i}.get(t_i)[0];$
        $sc_i \leftarrow map_{t_i}.get(t_i)[2];$
        if $sc_i > sc_{max}$ and $sc_i - sc_{max} > \tau$ then
            $sc_{max} \leftarrow sc_i;$
            // Remember optimum $method$ and splits $s$
            $method_{opt} \leftarrow method_i;$
            $s_{opt} \leftarrow s;$
        end
    end
end
return $s_{opt}$, $method_{opt}$ ;
```

Algorithm 2: opt#Slots&Method($map^s_{t_i}$, S)

Functional region detection (Algorithm 3): Algorithm 3 returns the activity of each physical region $r_j, \forall j \in [0, m]$. After concluding the optimum number of clusters *opt* in Algorithm 1, the optimum clustering technique $method_{opt}$, and the optimum number of time slots s_{opt} to split the dataset in Algorithm 2, the clustering is then performed. For each region, the algorithm counts the percentage of keywords within the checkins registered in that region during the time-slot t_i by retrieving the vector $\vec{\mathcal{C}}_{t_i}(r_j)$ (See Section 6.3.1) from the checkins matrix (line 8). Afterwards, the top activity is identified by calculating the maximum percentage within the checkins registered within the functional region $\vec{\mathcal{C}}_{t_i}(fr_{t_i})$. Finally, this represents the most dominating functional feature of the functional region (line 10).

6.4 Experiments setup

The raw data introduced in chapter 4 was preprocessed and mapped into a suitable format that can be consumed by the selected clustering algorithm. The outcome of data preparation was a set of Spatial-temporal Functional Matrices, each of which covers a given time-slot. Every Spatial-temporal Functional Matrix illustrates the probability of each physical area belonging to functional categories. The preprocessing work can be decomposed into Activity Categorization, Time-slot Aggregation, and Geographic Region Categorization as follows:

- **Activity Categorization**: The visited location of each check-in is classified into 9 categories: Entertainment, Education, Night Life, Recreation, Social Services[1], Res-

[1] This category involves check-ins related to services provided for the benefit of the community, such as pharmacy, recycling facility, and Laundry service.

```
Phase 3: Apply $opt$ and $method$ to $\mathcal{C}_d$ divided by optimum $s$ and determine functional
 regions $\mathcal{F}r_{t_i}$ for each time interval $t_i$
for $i \in [0, s_{opt}]$ do
    $\mathcal{C}_{t_i} \leftarrow$ subset($\mathcal{C}_d$,$t_i$);
    $\mathcal{F}r_{t_i} \leftarrow$ cluster($\mathcal{C}_{t_i}$, $opt$, $method_{opt}$ );
    for $\forall fr_{t_i} \in \mathcal{F}r_{t_i}$ do
        $\vec{\mathcal{C}}_{t_i}(fr_{t_i}) \leftarrow \emptyset$ ;
        for $r_j \in fr_{t_i}$ do
            $\vec{\mathcal{C}}_{t_i}(r_j) \leftarrow$ subset($\mathcal{C}_{t_i}$,$r_j$);
            // Update the checkins count of each activity for $fr_{t_i}$
            $\vec{\mathcal{C}}_{t_i}(fr_{t_i}) \leftarrow$ update($\vec{\mathcal{C}}_{t_i}(fr_{t_i})$,$\vec{\mathcal{C}}_{t_i}(r_j)$ );
        end
        // Calculate the top activity for functional region $fr_{t_i}$
        $topActivityFr \leftarrow$ maxCount($\vec{\mathcal{C}}_{t_i}(fr_{t_i})$);
    end
end
```

Algorithm 3: functionalRegionDetection($\mathcal{C}_d$, opt, $method_{opt}$, s_{opt})

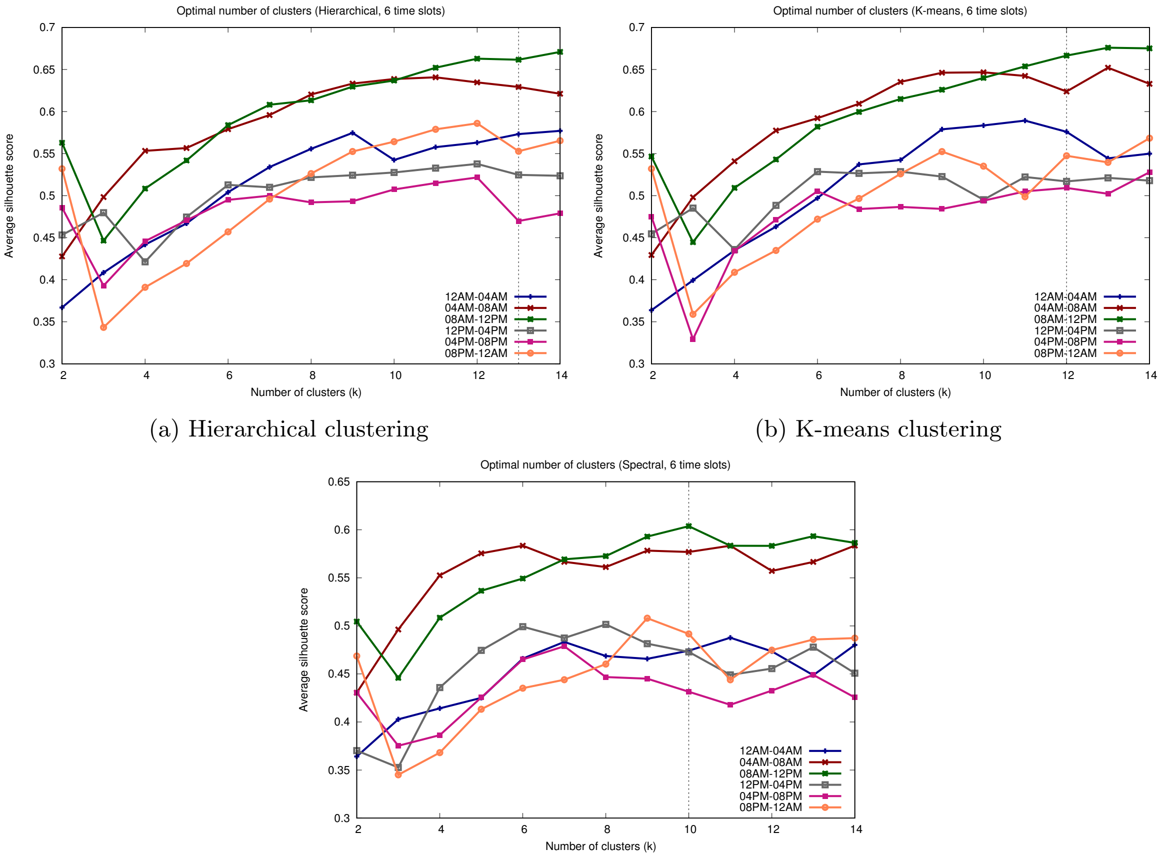
idence, Shopping, Travelling, and Eating, as used in Foursquare applications.These 9 activity categories will act as the functionalities of regions in our work. This preprocessing step will not only decrease the sparsity problem significantly when feeding the data to clustering techniques but also enable more meaningful functionalities of regions in this research to be used. In particular, this categorization mapped 532 different keywords in the dataset into 9 labels as functional regions to be mapped.

- **Time-slot Aggregation**: The daily records are then split into 24, 12, 6, 3 timeslots, each of which lasts for 1, 2, 4, and 8 hours respectively. In general the smaller the slot, the finer the region's functionality can be captured but with more sparsity challenges. The step aims to study the tradeoff between sparsity of the dataset and preservation of the meaningful functionality of a region.

- **Geographic Region Categorization**: The processed data, which contains checkins from Manhattanis then categorised into the zip codes boundaries The boundaries of Manhattan area and its zip codes are provided by NYC Department of Information Technology & Telecommunication as it was illustrated in chapter 4. This enables us to process the data as a matrix $\mathcal{C}_{t_i}$ defined in Equation 6.2.

6.5 Results and Validation

In this section, the results are shown for the proposed Temporal Functional Regions approach presented in section 6.3. The proposed approach is applied for identifying the optimum number of clusters k for each time interval by varying k from 2 to 14. As discussed previously, Figure. 6.1 and Figure. 6.2 show the results for the 4 and 2 hours intervals which resulted in 6 and 12 time slots respectively. From Figure 6.1a, Figure 6.1b, and Figure 6.1c, the optimum number of clusters for Hierarchical, K-means and Spectral



(a) Hierarchical clustering

(b) K-means clustering

(c) Spectral clustering

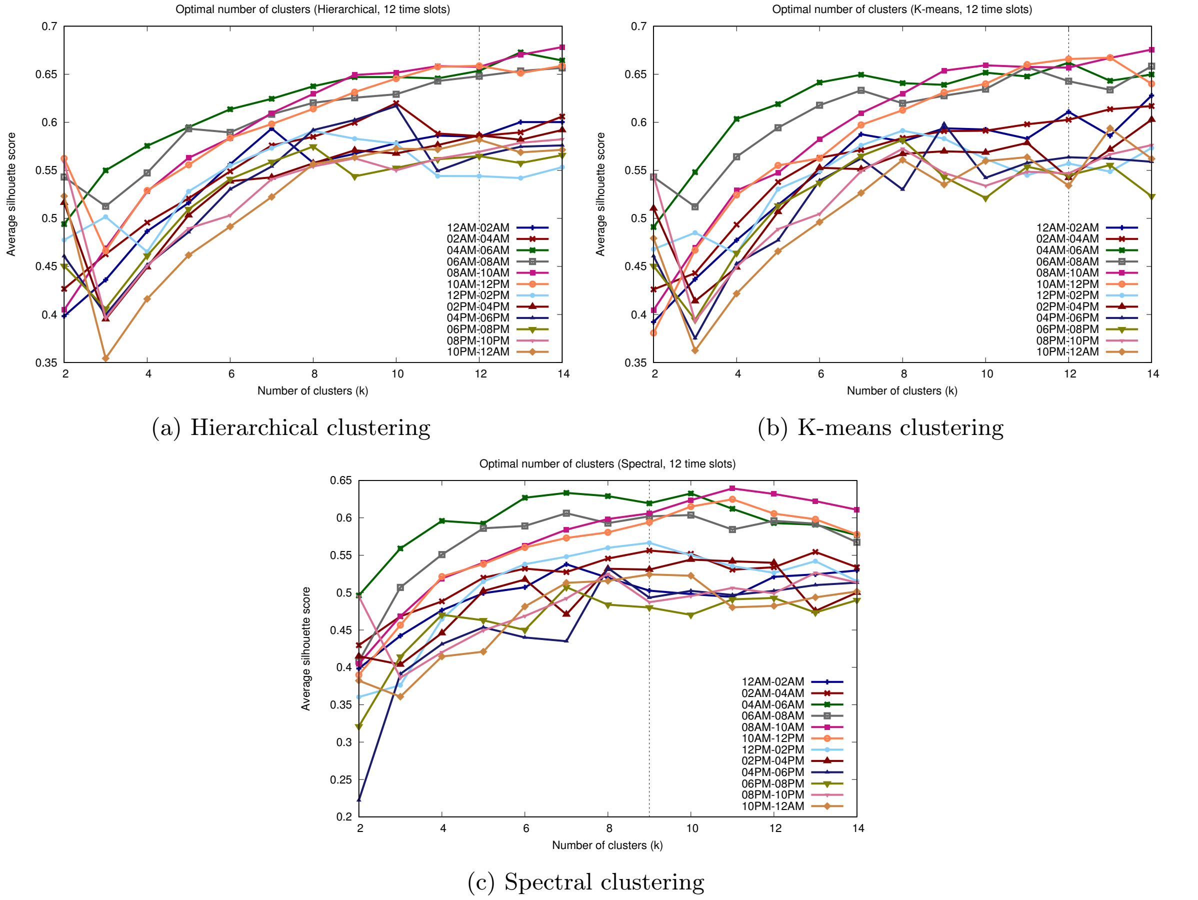
Figure 6.1: Exploring optimum number of clusters k for the 4 Hours interval resulting in 6 time slots.

clustering for the 4 Hours interval are 13, 12, and 10 respectively specified with a dashed line in the figures. Similarly, Figure 6.2a, Figure 6.2b, and Figure 6.2c shows that 12, 12 and 9 are the derived optimum number of clusters for the 2 Hours interval for the Hierarchical, K-means and Spectral clustering respectively. This process is repeated for the whole time intervals studied, 1 hour, 2 hours, 4 hours, 8 hours and with no temporal split at all. The resulting average silhouette scores are summarized and concluded for each temporal variation as illustrated in Figure 6.3. From this figure, it is worth emphasizing the following three points:

1. The “No Temporal Split” resulted in the least average silhouette scores which indicates the least quality of clustering compared to the Temporal Functional Regions.

2. There has been a substantial improvement in the silhouette scores for the 4 hours interval compared to the 8 Hours interval, however, the scores seem to stabilize going forward for the 2 hours and 1 hour time intervals.

3. Based on the previous point, the 4 hours slot was found to be the most reasonable when deriving the Temporal Functional Regions which will be illustrated and visualized in detail in the next section.



(a) Hierarchical clustering

(b) K-means clustering

(c) Spectral clustering

Figure 6.2: Exploring optimum number of clusters k for the 2 Hours intervals resulting in 12 time slots.

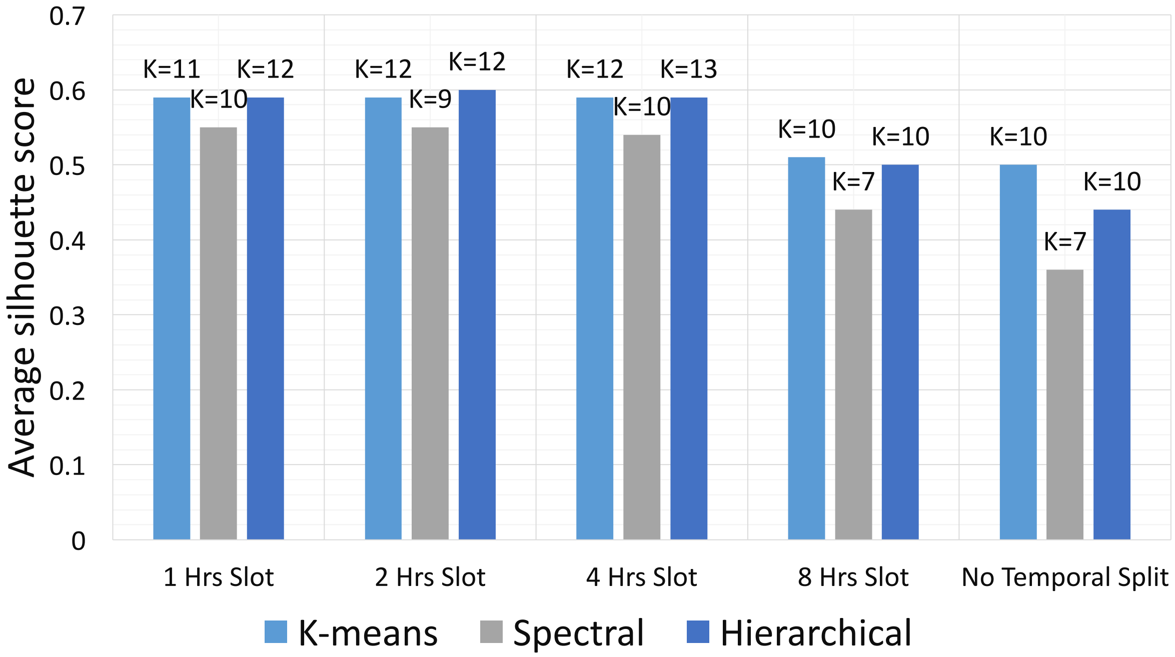


Figure 6.3: Temporal variation silhouette scores for different time intervals.

Table 6.1: Area clustering results for the morning activities.

cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	cluster 8	cluster 9	cluster 10	cluster 11	cluster 12	cluster 13
Recreation (0.54)	Eating (0.26)	Eating (0.38)	Traveling (0.35)	Recreation (1)	SocialServices (0.54)	Entertainment (0)	SocialServices (1)	Education (0.76)	Shopping (1)	NightLife (0.68)	Eating (1)	Residence (0.66)
Shopping (0.08)	Recreation (0.23)	Residence (0.22)	Eating (0.19)	Entertainment (0)	Eating (0.39)	Education (0)	Entertainment (0)	Eating (0.24)	Entertainment (0)	Eating (0.26)	Entertainment (0)	Traveling (0.21)
Eating (0.08)	SocialServices (0.13)	Traveling (0.1)	Recreation (0.12)	Education (0)	Recreation (0.03)	NightLife (0)	Education (0)	Entertainment (0)	Education (0)	Education (0.05)	Education (0)	Eating (0.11)
Traveling (0.07)	Traveling (0.11)	Shopping (0.09)	SocialServices (0.09)	NightLife (0)	Shopping (0.02)	Recreation (0)	NightLife (0)	NightLife (0)	NightLife (0)	Entertainment (0)	NightLife (0)	NightLife (0.01)
Residence (0.07)	Shopping (0.1)	Entertainment (0.09)	Entertainment (0.09)	SocialServices (0)	Residence (0.02)	SocialServices (0)	Recreation (0)	Recreation (0)	Recreation (0)	Recreation (0)	Recreation (0)	Recreation (0.01)
Education (0.06)	Residence (0.06)	Recreation (0.08)	Education (0.05)	Residence (0)	Education (0)	Residence (0)	Residence (0)	SocialServices (0)	SocialServices (0)	SocialServices (0)	SocialServices (0)	Shopping (0.01)
SocialServices (0.05)	NightLife (0.05)	SocialServices (0.03)	Shopping (0.05)	Shopping (0)	Traveling (0)	Shopping (0)	Shopping (0)	Residence (0)	Residence (0)	Residence (0)	Residence (0)	Entertainment (0)
Entertainment (0.04)	Education (0.04)	Education (0)	Residence (0.04)	Traveling (0)	Entertainment (0)	Traveling (0)	Traveling (0)	Shopping (0)	Traveling (0)	Shopping (0)	Shopping (0)	Education (0)
NightLife (0)	Entertainment (0.03)	NightLife (0)	NightLife (0.02)	Eating (0)	NightLife (0)	Eating (0)	Eating (0)	Traveling (0)	Eating (0)	Traveling (0)	Traveling (0)	SocialServices (0)

Table 6.2: Area clustering results for the afternoon activities.

cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	cluster 8	cluster 9	cluster 10	cluster 11	cluster 12	cluster 13
Eating (0.33)	Education (0.33)	SocialServices (0.4)	Traveling (0.51)	Recreation (0.4)	SocialServices (0.27)	Shopping (0.85)	Entertainment (0)	SocialServices (0.77)	SocialServices (0.62)	Recreation (0.75)	SocialServices (0.99)	Eating (0.95)
Shopping (0.19)	Eating (0.23)	Eating (0.2)	Residence (0.14)	Eating (0.22)	Traveling (0.23)	SocialServices (0.15)	Education (0)	Eating (0.16)	Recreation (0.38)	Residence (0.08)	Shopping (0.01)	SocialServices (0.04)
SocialServices (0.15)	SocialServices (0.11)	Recreation (0.12)	Eating (0.09)	Residence (0.08)	Eating (0.17)	Entertainment (0)	NightLife (0)	Education (0.03)	Entertainment (0)	Traveling (0.05)	Entertainment (0)	Traveling (0.02)
Recreation (0.11)	Recreation (0.1)	Shopping (0.1)	Recreation (0.08)	Entertainment (0.08)	Entertainment (0.1)	Education (0)	Recreation (0)	Traveling (0.02)	Education (0)	Eating (0.04)	Education (0)	Entertainment (0)
Traveling (0.1)	Traveling (0.08)	Education (0.07)	Shopping (0.07)	Shopping (0.07)	Shopping (0.09)	NightLife (0)	SocialServices (0)	Entertainment (0.01)	NightLife (0)	Education (0.03)	NightLife (0)	Education (0)
Residence (0.04)	Shopping (0.06)	Traveling (0.05)	SocialServices (0.06)	Traveling (0.06)	Education (0.06)	Recreation (0)	Residence (0)	Shopping (0.01)	Residence (0)	Entertainment (0.03)	Recreation (0)	NightLife (0)
Education (0.04)	Residence (0.05)	Residence (0.03)	Entertainment (0.04)	SocialServices (0.05)	Recreation (0.06)	Residence (0)	Shopping (0)	Recreation (0)	Shopping (0)	Shopping (0.02)	Residence (0)	Recreation (0)
NightLife (0.03)	Entertainment (0.03)	Entertainment (0.02)	Education (0)	Education (0.04)	NightLife (0.02)	Traveling (0)	Traveling (0)	NightLife (0)	Traveling (0)	SocialServices (0)	Traveling (0)	Residence (0)
Entertainment (0.02)	NightLife (0.01)	NightLife (0.01)	NightLife (0)	NightLife (0.01)	Residence (0.01)	Eating (0)	Eating (0)	Residence (0)	Eating (0)	NightLife (0)	Eating (0)	Shopping (0)

Table 6.3: Area clustering results for the evening activities.

cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	cluster 8	cluster 9	cluster 10	cluster 11	cluster 12	cluster 13
SocialServices (0.91)	Eating (0.26)	Recreation (0.42)	SocialServices (0.44)	Education (0.34)	Traveling (0.42)	Eating (0.6)	Shopping (0.55)	Eating (1)	Education (1)	Shopping (0.94)	Entertainment (0)	Eating (0.41)
Eating (0.04)	Shopping (0.18)	Eating (0.21)	Eating (0.4)	Eating (0.19)	Entertainment (0.15)	Recreation (0.1)	SocialServices (0.37)	Entertainment (0)	Entertainment (0)	SocialServices (0.06)	Education (0)	Shopping (0.18)
Education (0.02)	Recreation (0.14)	Education (0.09)	Shopping (0.14)	Residence (0.13)	Shopping (0.13)	Shopping (0.08)	Eating (0.05)	Education (0)	NightLife (0)	Entertainment (0)	NightLife (0)	SocialServices (0.11)
Recreation (0.02)	SocialServices (0.13)	Shopping (0.08)	Entertainment (0.02)	Shopping (0.12)	Eating (0.12)	Entertainment (0.07)	Education (0.04)	NightLife (0)	Recreation (0)	Education (0)	Recreation (0)	Recreation (0.09)
Shopping (0.01)	Traveling (0.1)	Traveling (0.06)	Education (0)	Recreation (0.1)	SocialServices (0.09)	SocialServices (0.06)	Entertainment (0)	Recreation (0)	SocialServices (0)	NightLife (0)	SocialServices (0)	NightLife (0.06)
Entertainment (0)	Residence (0.05)	SocialServices (0.04)	NightLife (0)	Traveling (0.06)	Education (0.05)	Traveling (0.05)	NightLife (0)	SocialServices (0)	Residence (0)	Recreation (0)	Residence (0)	Education (0.05)
NightLife (0)	Entertainment (0.05)	Entertainment (0.04)	Recreation (0)	SocialServices (0.04)	Recreation (0.03)	NightLife (0.03)	Recreation (0)	Residence (0)	Shopping (0)	Residence (0)	Shopping (0)	Entertainment (0.04)
Residence (0)	Education (0.05)	Residence (0.04)	Residence (0)	Entertainment (0.02)	NightLife (0.01)	Education (0.02)	Residence (0)	Shopping (0)	Traveling (0)	Traveling (0)	Traveling (0)	Traveling (0.04)
Traveling (0)	NightLife (0.03)	NightLife (0.02)	Traveling (0)	NightLife (0.01)	Residence (0)	Residence (0.01)	Traveling (0)	Traveling (0)	Eating (0)	Eating (0)	Eating (0)	Residence (0.02)

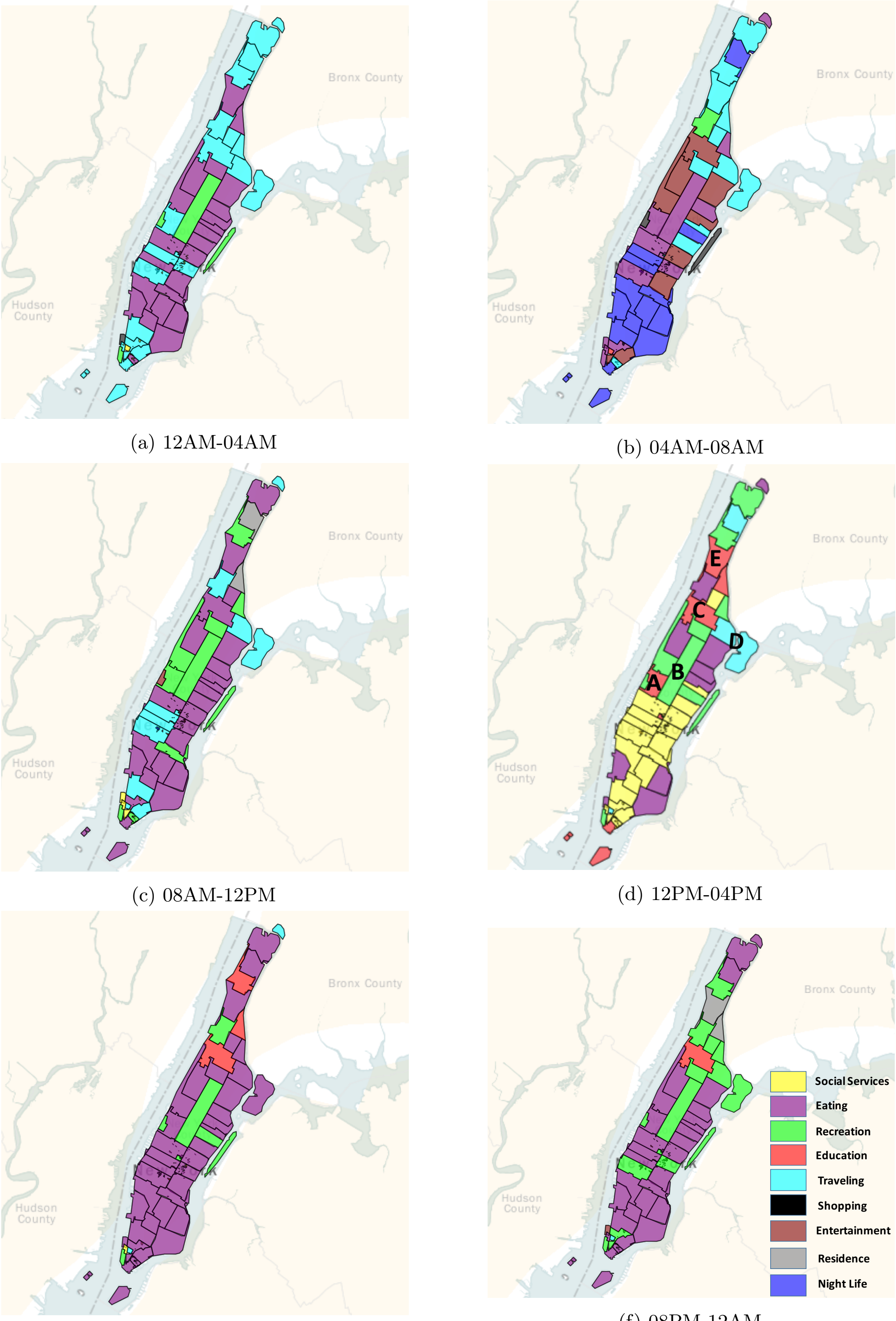
Table 6.4: Area clustering results for the night activities.

cluster 1	cluster 2	cluster 3	cluster 4	cluster 5	cluster 6	cluster 7	cluster 8	cluster 9	cluster 10	cluster 11	cluster 12	cluster 13
Recreation (0.65)	Entertainment (0.51)	Eating (0.28)	Traveling (0.84)	Education (0.44)	Residence (0.65)	Eating (0.59)	SocialServices (1)	Entertainment (0)	Shopping (0.93)	Education (1)	Eating (1)	Recreation (0.21)
Eating (0.09)	Eating (0.23)	Shopping (0.19)	Education (0.08)	SocialServices (0.13)	Recreation (0.12)	Recreation (0.11)	Entertainment (0)	Education (0)	SocialServices (0.04)	Entertainment (0)	Entertainment (0)	Traveling (0.21)
SocialServices (0.08)	Traveling (0.17)	Recreation (0.13)	Entertainment (0.06)	Shopping (0.12)	Eating (0.09)	SocialServices (0.09)	Education (0)	NightLife (0)	Eating (0.03)	NightLife (0)	Education (0)	Shopping (0.17)
Education (0.06)	SocialServices (0.03)	NightLife (0.1)	SocialServices (0.02)	Eating (0.12)	Education (0.06)	Shopping (0.07)	NightLife (0)	Recreation (0)	Entertainment (0)	Recreation (0)	NightLife (0)	Eating (0.14)
Residence (0.04)	Shopping (0.03)	Traveling (0.08)	Eating (0.02)	Recreation (0.09)	Shopping (0.04)	Traveling (0.06)	Recreation (0)	SocialServices (0)	Education (0)	SocialServices (0)	Recreation (0)	SocialServices (0.12)
Shopping (0.03)	Recreation (0.02)	SocialServices (0.07)	NightLife (0)	NightLife (0.05)	SocialServices (0.02)	Education (0.04)	Residence (0)	Residence (0)	NightLife (0)	Residence (0)	SocialServices (0)	Residence (0.05)
Traveling (0.03)	Education (0.01)	Entertainment (0.06)	Recreation (0)	Traveling (0.03)	Traveling (0.01)	Residence (0.03)	Shopping (0)	Shopping (0)	Recreation (0)	Shopping (0)	Residence (0)	NightLife (0.04)
Entertainment (0.02)	NightLife (0)	Education (0.05)	Residence (0)	Entertainment (0.01)	Entertainment (0)	NightLife (0.01)	Traveling (0)	Traveling (0)	Residence (0)	Traveling (0)	Shopping (0)	Education (0.03)
NightLife (0)	Residence (0)	Residence (0.04)	Shopping (0)	Residence (0.01)	NightLife (0)	Entertainment (0.01)	Eating (0)	Eating (0)	Traveling (0)	Eating (0)	Traveling (0)	Entertainment (0.03)

Due to the above, the results of the proposed approach are visualized and discussed focusing on the 4 Hours slot. For better understanding and validating the discovered functional regions, the following slots within the 4 Hours interval are defined: (a) Morning slot from 08AM-12PM. (b) Afternoon slot from 12PM-04PM. (c) Evening slot from 04PM-08PM. (d) Night slot from 08PM-12AM. Our results reported are for the Hierarchical Clustering output for a number of clusters K equals to 13 as it has the maximum silhouette score of 0.59 compared to 0.54 and 0.58 for Spectral and K-means clustering respectively (refer to Figure 6.3).

Table 6.1, 6.2, 6.3, and 6.4 show the results of Algorithm 3 for the morning, afternoon, evening, and night activities respectively. Each value in a cluster represents the number of check-ins with a specific functionality divided by the total number of check-ins observed in these zip-codes belonging to this specific cluster during certain time duration. Figure 6.4 shows the corresponding visualization for the Temporal Functional Regions illustrating how some regions can vary through the day. *Six* regions are highlighted in Figure 6.4d as A, B, C, D, E, and F as an example to better analyse and validate the generated functional regions. The following analysis as to why we observe the results we do for each region intuitively validates that the approach works:

- **Region A**: This region starts in the morning as a "recreation" functional area then in the afternoon it turns as an "educational" area due to the presence of several educational institutes in this area including Fordham university, Lincoln center, New York Institute of technology, and John Jay college of criminal justice. Then in the evening and night, the area turned to be "eating" functional area as it is considered attractive destination for several popular restaurants.



(a) 12AM-04AM

(b) 04AM-08AM

(c) 08AM-12PM

(d) 12PM-04PM

(e) 04PM-08PM

(f) 08PM-12AM

Figure 6.4: Temporal Functional Regions Visualizations.

- **Region B**: This region dedicates most of its space to the central park of Manhattan and is likely the reason that it is classified as a "recreation" functional region through the whole day with no any change over time.

- **Region C**: This region starts as an "eating" area in the morning and then turns to be an "educational" area until night and this can be interpreted due to the presence of one the biggest universities in US, Columbia university which indicates the influence of this university on the dominant functionality of this area not only in afternoon and evening but also in the night.

- **Region D**: This area starts in the morning and continues in the afternoon as a "travelling" functional region and that can be interpreted due to the presence of Triborough Bridge known officially as the Robert F. Kennedy which is a complex of three separate bridges in NYC that connects boroughs of Manhattan, Queens and the Bronx via Randalls and Wards Islands. In the evening, like most of other regions this area turns to be more "eating" area. At night, it seems as a more "recreation" area and that is due to the presence of two of the most popular parks Wards Island and Randalls parks which are very popular parks for hosting out door night events.

- **Region E**: This region starts in the morning as an "eating" functional and then its functionality changes in the afternoon to an "educational" due to the presence of The City of College in New York and then its dominant function turned to an "eating" area again like most of other areas at evening. At night this area has been found to be a "residential" area which might make sense for an area with no many outgoing night activities.

6.6 Summary

Discovering the functionality of regions in cities can enable new types of valuable applications that can benefit different end users: Urban planners could better identify the proximity of existing functional regions and hence, can contribute to better future planning for the cities. Tourists could differentiate scenic areas from other business and residential areas which will help in reducing effort for trip planning. Moreover, local people can better understand each part of their cities by understanding the different functionality of areas. With the rise of Location-Based Social Networks which attract lots of new users everyday with the potential of bridging the gap between the physical world and digital online social network services, it was shown in this chapter that identifying functional regions taking into account temporal variations of geographic user activity has become possible and is more sensible when identifying functional regions.

In this work, a novel approach is proposed to modeling functional areas taking into account temporal variation by means of place categories. The proposed approach compares between three clustering algorithms (Hierarchical, K-means, and Spectral) on areas and users of Manhattan borough in New York City using a dataset from one of the most vibrant LBSNs, Foursquare. In addition, the impact of different temporal variations splits on the quality of the clustering algorithms comparing it to the default approach with no temporal variation has been studied. The proposed approach introduced in this chapter may not only lead in deeper understanding of a complex city but also may offer finer personalized recommendations based on regions' functionality that changes over space and time. In the next chapter, a novel approach for recognizing Recurrent Crowd Mobility Patterns will be introduced and further will be correlated with the Temporal Functional Regions introduced in this chapter for showing the feasibility of extracting deeper insights around the motivation behind crowd mobility when correlating both extracted patterns.

Having presented in this chapter the second introduced contribution and urban pattern in this thesis referred to as *Temporal Functional Regions patterns*, the next chapter introduces a new approach for extracting an urban pattern called *Recurrent Crowd Mobility Patterns*. In addition, it will be shown that correlating both patterns could result in further insights into the motivation behind crowd mobility.

Chapter 7

Recognizing Recurrent Crowd Mobility Patterns

Chapter overview: *This chapter introduces a new approach for extracting* ***R****ecurrent* ***C****rowd* ***M****obility patterns in* ***C****ities (titled as RCMC) using a combination of the Kernel Density Estimation (KDE) and Non-negative Matrix Factorization (NMF) algorithms. KDE is used initially for extracting the crowd concentration variability across space and time whereas its output is fed to the NMF for extracting the "recurrent" behaviour of the crowd mobility. Using the same time interval as that used for extracting Temporal Functional Regions (as discussed in the previous chapter), it is further shown that the correlation between both patterns can provide interesting insights into the motivation behind crowd mobility. This new approach has been published in [TIST 2017].*

7.1 Motivation

Recognizing crowd mobility patterns in cities is very important for public safety, traffic management, disaster management, and urban planning [115]. In this chapter, a novel approach is introduced for recognizing Recurrent Crowd Mobility Patterns in cities. In addition, it is shown that the correlation between the extracted crowd mobility patterns with those Temporal Functional Regions introduced in chapter 6, supports the derivation of deeper insights into the motivation behind crowd mobility. The correlation between both patterns could potentially empower several applications such as traffic management, urban planning, and public safety. The following points summarize the contributions for the proposed approach along with some exemplar applications that could potentially benefit from the proposed approach:

1. A novel KDE/NMF based approach is proposed for extracting Recurrent Crowd Mobility Patterns where individuals mobility (represented by latitude and longitude) is fed into a Kernel Density Estimation (KDE) to infer crowded areas. Then the crowded areas act as the input to a Non Negative Matrix Factorization (NMF) based approach for recognizing the Recurrent Crowd Mobility Patterns and determining

when and where the crowd shifts during the various days of a week. The same crowded areas may span for consecutive several time slots and hence, the proposed approach intuitively also reveals also the level of crowdedness through the use of the NMF based approach. We refer to the level of crowdedness as the “crowd intensity” for each region.

Exemplar application: Being able to determine that a recurrent crowd mobility pattern for day d shifts from region r_1 at time-slot t_1 to region r_2 at time-slot t_2 could allow city planners to dynamically allocate cities resources accordingly (e.g., energy resources). Similarly, the telco operators may be able to dynamically migrate network resources from r_1 to r_2. This is in contrast to existing solutions that over-provision cities' various regions with similar amount of network resources, which has been found to be a very expensive solution [116]. Another obvious benefit for extracting such patterns, is the ability of detecting anomalous and rare events. For instance, detecting that region r_3 is crowded at time-slot t_2 could raise an emergency alarm of an anomalous behaviour occurring compared to the expected pattern.

2. Furthermore, in this research some specific regions have been studied to highlight how further the correlation between the extracted Recurrent Crowd Mobility Patterns with their Temporal Functional Regions might help in better understanding the motivation behind the crowd mobility and capturing the dynamicity of such regions.

   Exemplar application: Marketing and advertisement is a key natural domain for such correlated patterns. For instance, customized recommendation services could be built based on this correlation on *where crowd will go and for what purpose.* If the crowd at time t_1 is located in region r_1 identified as “Eating” functional area, the recommendations could be adapted for the purpose of suggesting restaurants that suits time-slot t_1 (e.g., lunch restaurants or dinner restaurants) and if the crowd is expected to move to region r_2 at time-slot t_2 with an expected dominant functionality of “Education”, the recommendations could then be dynamically adapted towards students' deals/offers and so on.

7.2 State-of-the-Art

Among the various state-of-the-art that focused on extracting urban patterns in cities, there have been a more recent focus within the research community on extracting and predicting mobility patterns. In [46] and [47], the authors mainly forecast billions of individuals' mobility traces rather than the aggregated crowd flows. The individual mobility patterns are those concerned with predicting the future locations of individuals. One challenge in this task is that it is computationally expensive, and predicting individual's mobility is not necessarily useful to public safety, disaster management and other applications that could more benefit from crowd mobility. Another branch of research focuses on predicting traffic volume and travel speed on the road [48] [49] [50] [51], to the best of our knowledge the majority of the work reviewed in this area focuses on single or specific road segments rather than citywide scale approach for travel and speed. Recently, the research community has started to focus on city wide scale traffic flows prediction. In [52], the authors proposed an approach to predict crowd flows including human mobility data, weather conditions, and road network data utilizing Gaussian Markov random fields, to cope with noisy and missing data. In [53], the authors propose a deep-learning based model for forecasting the flow of crowds in each region using trajectory, weather and events data. The proposed approach in this chapter is different from the two previous approaches as their proposed models focus naturally on an individual region and not the city, and they do not partition the city using zip codes which could be more meaningful partitioning to city planners. In addition, the proposed approach estimates crowd intensity which provides a finer understanding of the level of crowdedness. Nevertheless and for the first time, I show in this chapter that correlating the extracted Recurrent Crowd Mobility Patterns with the Temporal Functional Regions using same time interval has the potential for deeper insights about cities as well as understanding the motivation behind such crowd shifts.

7.3 Recurrent crowd mobility pattern

7.3.1 Notation and Definitions

In a similar manner to the previous chapter, the raw LBSNs dataset is denoted with D. Moreover, the dataset is split into d days, which are further split into s time slots of equal sizes, such that $d = \langle t_1, t_2, ...t_s \rangle$. The dataset for a day d remains of $24hour$ duration when s is equal to 1. Further, each time interval t_i has a duration of: $\|t_i\|= \frac{24}{s}$, $\forall i \in [1, s]$.

Definition 4 (Region r_j) *The dataset D contains geo-tagged data, which is further divided into m physical regions such that $R = \{r_1, ...r_m\}$, where R represents the collection of all regions r_j, $\forall j \in [1, m]$. Each region r_j is associated with a zip-code and represents a collection of points each denoted by $p_l^{r_j}$ and identified by a latitude and longitude: $p_l^{r_j} =< lat_l^{r_j}, long_l^{r_j} >$, $\forall j \in [1, m] \wedge l \in [1, n]$.*

In order to identify the spatial areas where the relative intensity of tweets in a certain timeslot and day is significantly higher compared to the overall spatial tweeting distribution, a crowded area is defined as follows:

Definition 5 (Crowded Area $\mathcal{CA}$) *A crowded area $\mathcal{CA}$ is defined as a set of points p_l, $p_l =< lat_l, long_l >$, which are equal to a density estimation larger than or equal to the* *mean of the overall density estimates of the spatial distribution for t_i in day d:*

$$\mathcal{CA} \ = \ \{p \mid \rho_K(p) \geq \sum_{l=1}^{n} \frac{\rho_K(p_l)}{n} \ , \ p_l = < lat_l, long_l >\} \tag{7.1}$$

Moreover, since crowded areas change over time, therefore the crowded areas over a time interval t_i is referred as $\mathcal{CA}_{t_i}$, and over a dataset d as $\mathcal{CA}_d$. The footprint of an individual $\mathcal{I}$ during a time-slot t_i is denoted by $\mathcal{IF}_{t_i}$, where a footprint is represented by a tuple between a time interval t_i during the day d and the region $r_j^{\mathcal{I}}$ (i.e., zip-code) where individual $\mathcal{I}$ is located (i.e., the location of the individual's' checkin): $\mathcal{IF}_{t_i} =< t_i, r_j >$. Further, a crowd footprint is defined as follows:

Definition 6 (Crowd Footprint $\mathcal{CF}_{t_i}$) *A crowd footprint during a time-slot t_i is defined as a set containing the individual footprints of all individuals during time interval t_i, denoted further by $\mathcal{CF}_{t_i}$. Moreover, $\mathcal{CF}_d$ is denoted by the sequence of all crowd footprints within the day d: $\mathcal{CF}_d = \langle \mathcal{CF}_{t_1}, ..., \mathcal{CF}_{t_s} \rangle$*

For instance, a day can be translated into the following sequence of crowd footprints: $\mathcal{CF}_d = \langle < t_1, zip10002 >, < t_1, zip10014 >, < t_2, zip10011 >, < t_3, zip10003 >, ..., etc. \rangle$. This suggests that crowds exist in zip-code areas 10003 and 100014 in the time-slot t_1. In the second and third time slots (t_2 and t_3), the crowd has been shifted to zip codes 10011 and 10003 respectively. It is worth emphasizing that since the objective is not to detect the mobility of a particular crowd in time, the $\mathcal{IF}_{t_i}$ and $\mathcal{IF}_{t_i}$ might not contain same individuals.

7.3.2 Proposed Approach (RCMC)

The proposed approach for detecting Recurrent Crowd Mobility Patterns, is based on three phases as follows:

(1) Crowd detection Phase: The crowded areas are detected at certain time interval given a certain geographical area. The proposed crowd detection approach is based on the spatial distribution of the social life traces captured from the LBSNs dataset as introduced in chapter 4. The proposed approach for crowd detection can be applied to any dataset that represents a reasonable[1] spatial-temporal distribution for the life traces in a certain geographical area.

(2) Data preparation Phase: After detecting crowded areas from the previous phase, data is transformed in this phase into a data structure that is suitable for processing by the third phase. In particular, a document is formulated for each day of the week which consists of the crowd footprints generated from the crowded areas at a certain time-slot. The 7 documents comprising crowd footprints generated in this phase will be the input data structure for input into the next phase.

[1] A reasonable spatial-temporal dataset is the one that has an acceptable level of sparsity preserving the real-life spatial-temporal distribution of citizens across a city.

(3) Recurrent crowd mobility patterns recognition Phase: The 7 documents generated from the previous phase act as the input to this final phase with the aim of extracting the Recurrent Crowd Mobility Patterns (i.e., topics) over space and time by applying topic-based models. Such detected crowd mobility patterns can be further correlated with the Temporal Functional Regions to understand the motivation behind crowd mobility. The following sections describe each of these phases in depth.

7.3.2.1 Crowd detection phase

The main objective of this phase is to detect the crowded areas at a given time-slot of each day in the week. Hence the following approach is applied:

(1) Each day of the week is divided into s time-slots lasting for $\frac{24}{s}$ hours each. Here each time-slot is still the raw data but only filtered temporally. Figuring out the optimum s may vary from dataset to another as it was illustrated previously in the previous chapter when deriving the Temporal Functional Regions (see Figure 6.3 in chapter 6). In general, the bigger is s, the finer mobility patterns can be detected.

(2) For each time-slot t_i, multi-variate kernel density estimation is applied with the aim of detecting the most crowded areas. Kernel Density Estimation was selected as it is a well-established non-parametric statistical technique, due to its computationally efficiency and scalability for processing streaming data. A kernel is a positive function $K(x, h)$ which is controlled by the bandwidth parameter h. In the proposed approach, gaussian kernel is utilized since it is the most widely used and have shown success across various applications [117]. Given this kernel form, the density estimate at a point p within a group of points $p_l; l \in [1, n]$ is given by:

$$\rho_K(p) = \frac{1}{n}\sum_{l=1}^{n} K\left(\frac{p - p_l}{h}\right) \ , \ where\ K(x,h) = \ \alpha \cdot exp(\frac{x^2}{2 \cdot h^2}) \tag{7.2}$$

The bandwidth h here acts as a smoothing parameter, controlling the tradeoff between bias and variance in the result. A large bandwidth leads to a very smooth (i.e. high-bias) density distribution. A small bandwidth leads to an unsmooth (i.e., high-variance) density distribution.

Algorithm 4 presents the approach of this phase. The algorithm checks from lines 5 to 7 for all points $p \in p_l^{r_j}$ whether their density estimation is greater or equal to the mean of the overall density estimates of the spatial distribution in r_j for time-slot t_i in day d (calculated at line 4). If so, the point is added to $\mathcal{CA}$, the algorithm constructs the tuples $\mathcal{CF}_d$, by traversing the constructed $\mathcal{CA}$ and building a tuple of the associated zip-code and the time interval the point belongs to.

To sum up, in this phase, it is identified where and when the city is most crowded and how the crowd shifts temporally based on the time-slot during the day. This phase only focuses on city dynamics and disregards the citizens' behaviour dimension.

```
// Phase 1: Identify crowded areas using Multi Variate Kernel Density Estimation,
   mvkde function, with gaussian kernel
$\mathcal{CA}_d \leftarrow \emptyset$;
for $t_i \in d$ do
    $\mathcal{CA}_{t_i} \leftarrow \emptyset$;
    for $r_j \in R$ do
        // Calculate the mean of the overall density estimates of the spatial
           distribution for time-slot $t_i$ in day $d$
        $mean \leftarrow \sum_{l=1}^{n} \frac{\rho_K(p_l)}{n}$;
        for $p \in p_l^{r_j}$ do
            // Apply mvkde
            $\rho_K(p) \leftarrow$ mvkde($p$,$p_l$);
            // Add the point $p$ to $\mathcal{CA}_{t_i}$ if density greater or equal to $mean$
            if $\rho_K(p) \geq mean$ then
                $\mathcal{CA}_{t_i} \leftarrow \mathcal{CA}_{t_i} \cup \{p\}$;
            end
        end
    end
    $\mathcal{CA}_d \leftarrow \mathcal{CA}_d \cup \mathcal{CA}_{t_i}$;
end
return $\mathcal{CA}_d$;
```

Algorithm 4: generateCrowdedAreas(d)

7.3.2.2 Data preparation phase

After pre-processing the data and identifying the crowded areas in the previous phase, the data is further processed in a representation that is suitable for topic models and precisely Non-negative Matrix Factorization (NMF) which will be the main pillar for recognizing crowd mobility patterns as will be illustrated in the phase to follow. In order to recognize the crowd mobility patterns of a city, the footprints of individuals $\mathcal{IF}_{t_i}$ for each day is created during each time-slot t_i , as $\mathcal{IF}_{t_i} =< t_i, r_j >$. In particular, given the crowded areas $\mathcal{CA}_{t_i}$ for each time-slot t_i from the previous phase (crowd detection phase), each of these points $p_l^{r_j}$ is mapped with their associated zip-code[2] r_j for constructing the tuples $\mathcal{IF}_{t_i}$.

Further, 7 documents are generated, where each document d represents a day of the week and each word in the document is a $\mathcal{IF}$ of an individual. For each document, the crowd footprints $\mathcal{CF}_d$ are generated, which represents the set of all individual footprints within dataset d by merging all individuals' footprints for all time slots t_i, $\forall i \in [0, s]$.

This phase is concluded by generating the set of all crowd footprints $\mathcal{CF}_{t_i}$ for those points that Algorithm 4 marked as belonging to crowded areas, $p_l \in \mathcal{CA}_{t_i}$ for each time-slot t_i within the d dataset. This phase aggregates the crowd footprints across all intervals and returns the associated crowd footprints for the day d, $\mathcal{CF}_d$.

[2] The approach of assigning zip codes is considered, however, this can be replaced with any other boundaries definition.

7.3.2.3 Recognition phase

In the proposed methodology, NMF [117] technique is employed for recognizing crowd mobility patterns for discovering crowd shifts from one area to another in a city. NMF refers to an unsupervised family of algorithms from linear algebra that simultaneously perform dimension reduction and clustering. While NMF has become popular as a tool for data exploration in bioinformatics, and was used in the past for clustering documents, it has only recently been recognised as a useful tool for topic modeling [118]. Topic models as introduced in chapter 3 are powerful tools initially developed to characterise text documents, but can be extended to other collections of discrete data (e.g., mobility data).

NMF seeks to decompose a data matrix into factors that are constrained so that they will not contain negative values. Given a document-term matrix $\mathbf{M} \in \mathbb{R}^{uxd}$ representing u unique terms present in a corpus of d documents, NMF generates a reduced rank-k approximation as the product of two non-negative factors:

$$\mathbf{M} \approx \mathbf{WH} \textit{ such that } \mathbf{W} \geq 0,\ \mathbf{H} \geq 0 \tag{7.3}$$

where the objective is to minimize the reconstruction error between $\mathbf{M}$ and the low-dimensional approximation. In the case of text data, the columns or basis vectors of $\mathbf{W} \in \mathbb{R}^{uxk}$ can be interpreted as topics, defined with non-negative weights relative to the u terms. The entries in the matrix $\mathbf{H} \in \mathbb{R}^{kxd}$ provide document memberships with respect to the k topics. Note that, unlike LDA which operates on raw frequency counts, NMF can be applied to a non-negative matrix $\mathbf{M}$ that has been previously normalized using common pre-processing procedures such as TF-IDF [119] term weighting and document length normalization. As with LDA [120], document-topic assignments are not discrete, allowing a single document to be associated with more than one topic. Formally, the entity termed *word* is the basic unit of discrete data defined to be an item from a vocabulary of size V. A *document* comprises of a sequence of w words. A corpus D if the whole set of the collection of d documents. In the context of this work, the *word* is represented by a location(zip-code) and time-slot which is depicted by $\mathcal{IF}$ as introduced in the previous section. A document is a day of the city with the aggregation of all the $\mathcal{IF}_{t_i}$, $\forall t_i \in d$ that exists in the dataset of this particular day. Hence, the corpus is formulated of 7 documents where each represents one day of the week. The final objective is to extract “topics” that each represents the crowd mobility patterns in cities. Each topic represents a sequence of crowd mobility concentration that recur across space and time for a particular day, for instance, one topic for Saturdays could indicate crowd concentration in area zip-10003 from $08AM$-$12PM$ then from $12PM$-$04PM$, the crowd concentration shifts towards zip-10026 and so on. A summary of the analogy from document-topics to crowd-mobility patterns is shown in Figure 7.1.

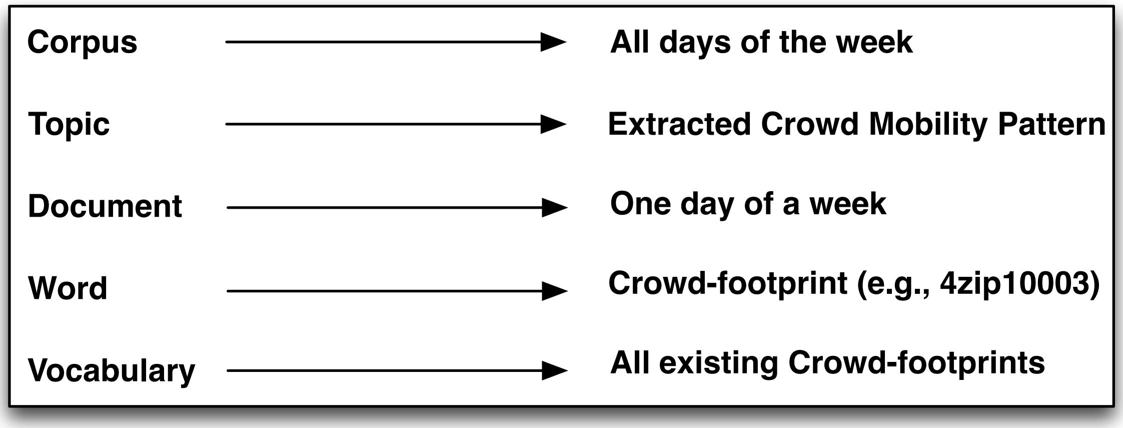

Corpus
All days of the week
Topic
Extracted Crowd Mobility Pattern
Document
One day of a week
Word
Crowd-footprint (e.g., 4zip10003)
Vocabulary
All existing Crowd-footprints

Figure 7.1: Analogy from document-topics to crowd-mobility patterns.

```
// Phase 3:
$\mathbf{M} \leftarrow$ genMatrix($\mathcal{CF}_D$);
$\mathbf{M} \leftarrow$Tf-IDF($\mathbf{M}$);
// Initial factors generated using the NNDSVD below
$\mathbf{W}, \mathbf{H} \leftarrow$ NMF($\mathbf{M}$);
// Calculate reconstruction error $\epsilon$ based on euclidean distance ed between M,W,H
$\epsilon \leftarrow$ ed ($\mathbf{M}, \mathbf{W}, \mathbf{H}$);
// Minimize $\epsilon$ through EM
$\mathbf{W}^{‘}, \mathbf{H}^{‘} \leftarrow$ EM($\mathbf{W}, \mathbf{H}, \epsilon$);
```

Algorithm 5: crowdPatternDetection($\mathcal{CF}_D$)

The following steps and Algorithm 5 summarize the proposed approach based on NMF:

(1) From the 7 documents generated from the previous procedure of data preparation, the document-term matrix $\mathbf{M}$ is constructed.

(2) TF-IDF term weighting and unit length normalisation is applied to the document-term matrix $\mathbf{M}$ (algorithm 5, line 2).

(3) NMF algorithms are often initialized with random factors. However, this can lead to many different "unstable" results, depending on the random values. To ensure a single definitive output, initial factors are generated using the Non-negative Double Singular Value Decomposition (NNDSVD) approach proposed by Boutsidis and Gallopoulos [121] (algorithm 5, line 3).

(4) The standard Euclidean formulation of NMF is applied for measuring the reconstruction error ϵ between $\mathbf{M}$ and the approximation $\mathbf{WH}$, using the initial factors from step 3 (algorithm 5, line 4).

(5) Furthermore, an optimization process based on EM algorithm is applied to refine $\mathbf{W}$ and $\mathbf{H}$ in order to minimize the objective function (Equation 7.4) by iterating between two multiplicative update rules (Equation 7.5) until convergence for a number of iterations (algorithm 5, line 5).

$$\frac{1}{2}||\mathbf{M} - \mathbf{WH}||_2^F = \sum_{i=1}^{d}\sum_{j=1}^{u}(\mathbf{M}_{ij} - (WH)_{ij})^2 \tag{7.4}$$

$$H_{cj} \longleftarrow H_{cj}\frac{(W\mathbf{H})_{cj}}{(W\mathbf{WH})_{cj}} \qquad W_{ic} \longleftarrow W_{ic}\frac{(\mathbf{MH})_{ic}}{(W\mathbf{HH})_{ic}} \tag{7.5}$$

(6) The resulting k topics are defined by: (a) topic descriptions as given by the top-ranked terms in the columns of the factor **W**; (b) document membership weights as given by the values in the rows of **H**.

Figure 7.2 shows the NMF application for detecting crowd mobility patterns where matrix **M** depicts the document-term matrix where each row indicates a day of the week and each column represents a $\mathcal{CF}_d$. Matrix **W** represents the latent relationship between each day of the week and the topics where each can indicate a different crowd mobility pattern. Finally matrix **H** shows the relationship and correlation between each of the topics (crowd patterns) and the $\mathcal{CF}_d$.

7.4 Experiments setup

The following explains the setup used in the different phases during the experiments that were undertaken.

Data preparation phase: Based on the dataset gathered and introduced in chapter 4, we follow the proposed approach for detecting the Recurrent Crowd Mobility Patterns in Manhattan comprising of three phases as introduced in the previous section.

For the crowd detection phase, each day of the week is divided into 6 time-slots lasting for 4 hours each (i.e., $s = 6$, $\|t_i\| = 4$). The reasons for choosing s as 6 are two-fold:

1. Intuitively, the objective is to try to extract recurrent patterns following the common terminology used by the public as well as urban planners such as morning, evening, and night activities. Hence, following terminology is considered: Late Night ($12AM$-$04AM$), Early Morning ($04AM$-$08AM$), Late Morning ($08AM$-$12PM$), Afternoon ($12PM$-$04PM$), Evening ($04PM$-$08PM$), Early Night ($08PM$-$12AM$).

2. In chapter 5 (section 3.4), it was shown that 6 time-slots are the most suitable granularity for detecting *Temporal Functional Regions* so choosing same time interval for detecting Recurrent Crowd Mobility Patterns will facilitate the correlation between both[3] and hence, deriving further insights around the motivation behind crowd mobility.

Detecting crowded areas phase: Multi-variate kernel density estimation is applied for detecting crowded areas. Gaussian kernel is used as they are the most commonly used proving success in various applications [117]. For setting the bandwidth h used for

[3] Despite this, the proposed approach has the potential to be applied to smaller time intervals for the purpose of detecting finer mobility patterns if needed.

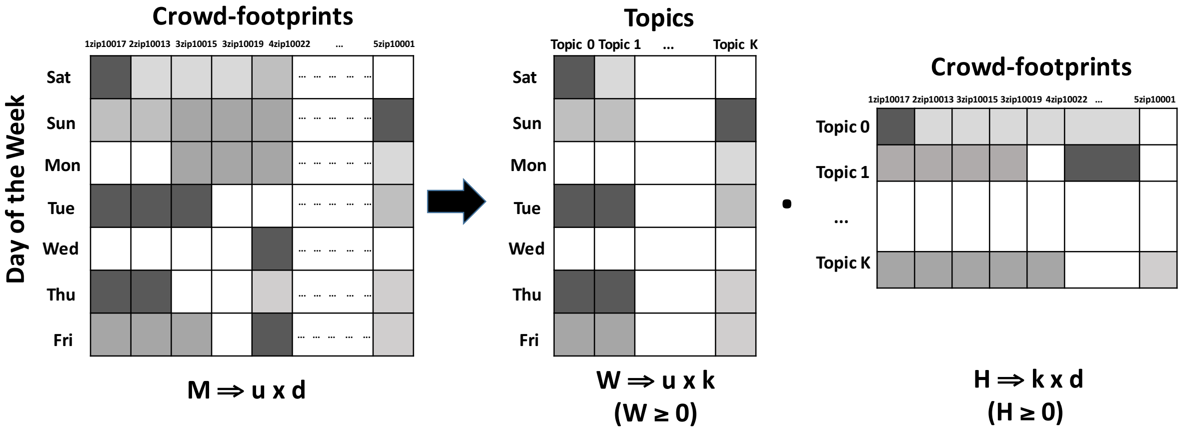
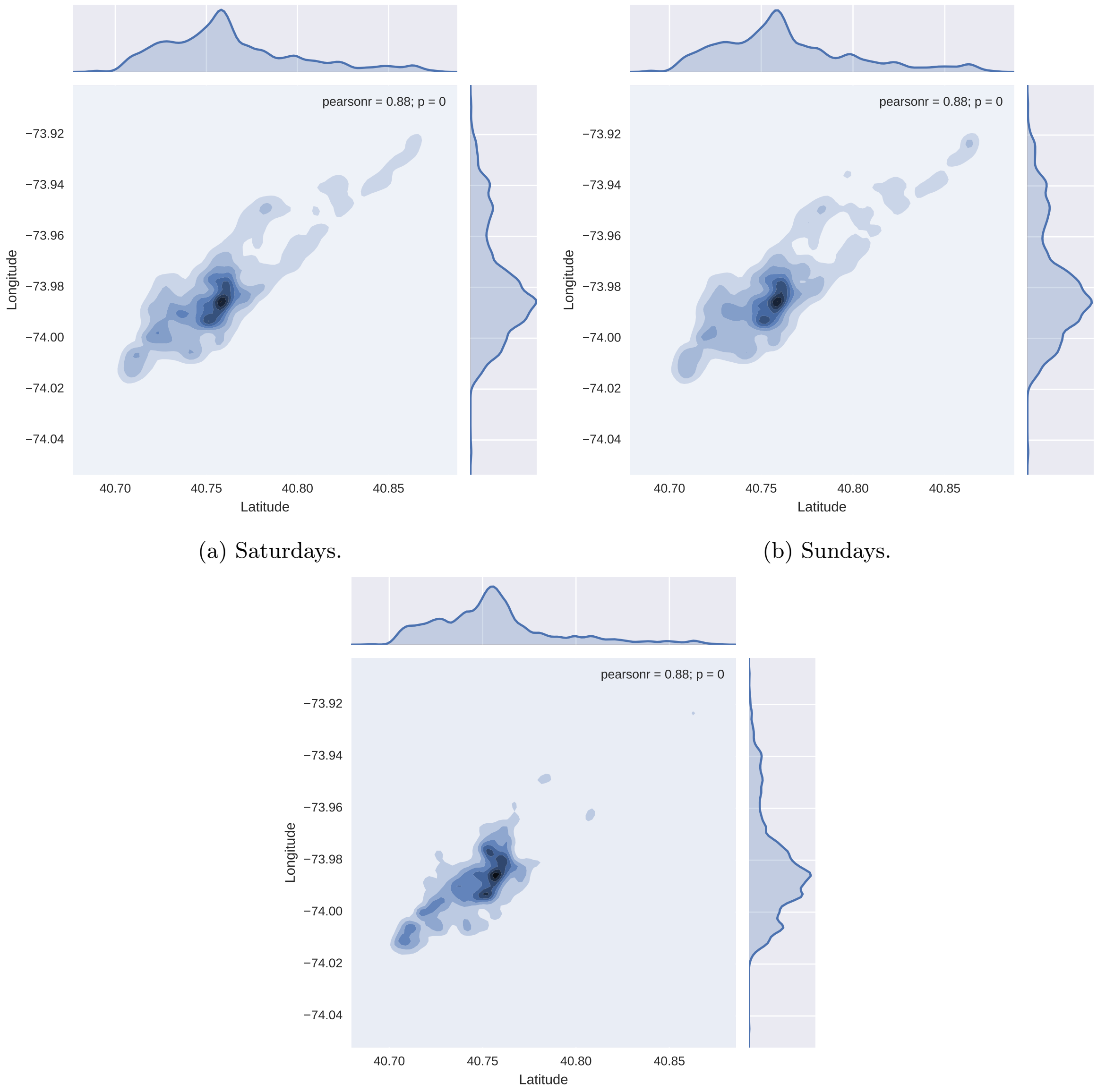


Figure 7.2: NMF: Given a non-negative matrix **M**, find k-dimension approximation in terms of non-negative factors **W** and **H**.

controlling the tradeoff between bias and variance, an exhaustive search was performed over a range of $[-1, 1]$ parameter values for an estimator through a cross-validated grid-search over a parameter grid. It was concluded that 0.1 is the optimum bandwidth for the KDE gaussian kernel fitted model. Figure 7.3 shows an example of the KDE results for one time-slot ($12PM$-$04PM$) where the x axis represents the latitude and the y axis represents the longitude in which the darker the blue, the more crowded areas. From this Figure, it can be observed that there is a difference in the crowd spatial distribution between the weekends (Figure 7.3a and Figure 7.3b) and the Weekdays (e.g., Wednesdays, Figure 7.3c). This follows intuitive thinking.

Recognition phase: The dataset samples are further processed into individual-footprints that are equal or larger than the mean of the overall density estimate of the spatial distribution for each of the time slots for every day. Aggregating the samples based on the day of the week resulted in 7 documents where each represents a day of the week comprising a set of individual-footprints resulted from the crowd detection step. In order to detect the "recurrent" crowd mobility patterns, the approach presented in detail in Section 7.3 and precisely in Algorithm 5 is applied. Although, there are several objective methods [122] for choosing the optimum number of topics, a subjective approach is applied in this research which is still the most reliable approach to date through interpreting the results taking into account what makes sense from the domain knowledge perspective [120]. Therefore, Algorithm 5 was first applied with the maximum number of topics that is bounded to 7 due to the dimensionality of matrix **W** with 7 rows (see Figure 7.2). Table 7.1 shows the results of applying Algorithm 5 using 7 as the number of topics. Each topic represents a different recurrent mobility pattern and the numbers in the table indicates *the probability of a particular day belonging to a certain topic*. In other words, it indicates the significance of the pattern/topic for a particular day. It is worth highlighting two points from this table: (a) Our proposed approach can discover of maximum three different patterns where each of the weekend days has a different pattern (Topic 3 and Topic 4) and the rest



(a) Saturdays.

(b) Sundays.

(c) Wednesdays.

Figure 7.3: KDE Gaussian Kernel Results (12PM-04PM).

of the weekdays share a common pattern (Topic 0). (b) The dominant pattern extracted for each day, represented by the probability highlighted in blue.

The optimum number of topics in this case is defined as *the least number of topics that could still discover all the different patterns captured with the maximum number of topics*. In our case, the maximum number of patterns are 3, extracted from the maximum number of topics, 7. Hence, the proposed approach was applied on 6, 5, 4, and 3 topics and it was concluded that 4 topics is the least number of topics that can discover 3 different patterns. Besides, the results for the model trained on 2 topics is of interest as well, as this investigation might reveal one common pattern for the weekends and another for the weekdays, where each is represented by different topic. Therefore, in the following sections, the extracted Recurrent Crowd Mobility Patterns will be discussed in depth

Table 7.1: Probability of each day belonging to certain topic (Topics = 7).

	Topic 0	Topic 1	Topic 2	Topic 3	Topic 4	Topic 5	Topic 6
Saturday	0.23102539	0.18439917	0.	0.57040886	0.01416657	0.	0.
Sunday	0.28908052	0.71091948	0.	0.	0.	0.	0.
Monday	0.41321829	0.02970232	0.39386167	0.11050093	0.04348833	0.	0.00922846
Tuesday	0.36381954	0.02375433	0.11015824	0.18867259	0.	0.3135953	0.
Wednesday	0.52313879	0.00003962	0.03673921	0.02144045	0.01621852	0.00797331	0.39445011
Thursday	0.62695577	0.	0.10387766	0.	0.11728928	0.15187729	0.
Friday	0.40699213	0.09169875	0.	0.11397975	0.29060095	0.06595809	0.03077032

when the number of topics is 2 and 4.

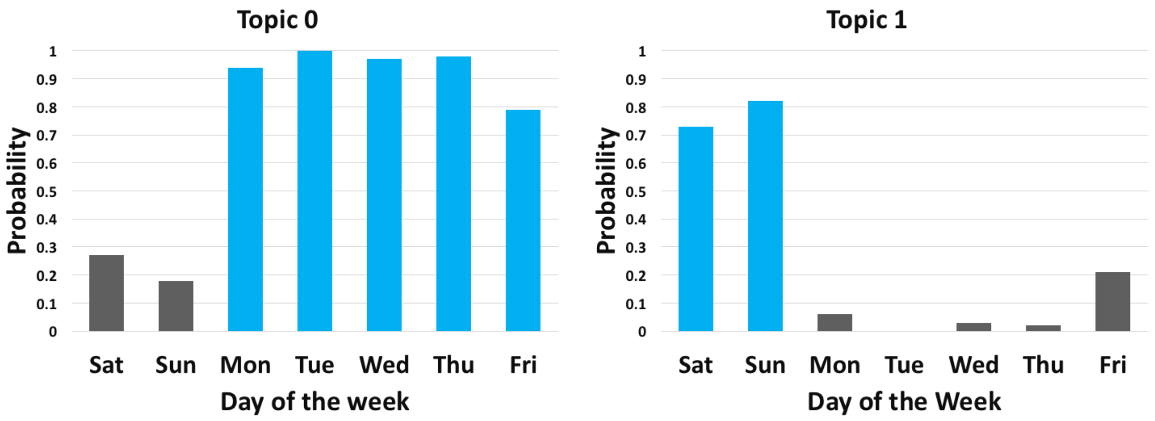
Moreover, for estimating the crowd intensity, the top-k words are divided for each of the extracted topics into three portions based on the per-word topic assignments probabilities indicating: (i) “Low” ($[0-0.33]$), (ii) “Medium” ($]0.33-0.66]$) and (iii) “Highly” crowded regions ($[0.66-1]$). Our argument that these three equal divided portions could indicate an estimate of the crowd intensity.

7.5 Results and Validation

In this section, the topic distributions are discussed when selecting 2 and 4 topics for training the proposed approach. Then, an evaluation metric called “topics stability” is introduced with the aim of evaluating the effectiveness of our proposed approach compared to other two baselines.

Topic distribution when utilizing two topics: Figure 7.4 shows the topic distribution across each day where the x axis indicates the day of the week and the y axis represents the probability of a particular day belonging to a certain topic. It can be observed that Topic 0 in Figure 7.4a represents the weekdays' recurrent crowd mobility pattern with at least 0.8 probability for all weekdays. Figure 7.4b represents the weekends' recurrent crowd mobility pattern with 0.73 probability for Saturdays and 0.8 probability for Sundays. Hence, our proposed approach trained on two topics clearly extracts obvious Recurrent Crowd Mobility Patterns for the weekdays and weekends following the intuitive understanding. Figure 7.5 and Figure 7.6 visualize the top-150 words (choosing 150 words will be justified later in Section 7.5) representing the Recurrent Crowd Mobility Patterns for the weekdays and weekends respectively with an estimation of the crowd intensity using three different grades of red colour where light-red, medium-red, and dark-red indicate low-crowded, medium-crowded, and highly-crowded areas respectively.

In the following discussion, the observed results are intuitively validated based on analysis of the areas involved by the author of this thesis. It is observed from the weekdays pattern visualized in Figure 7.5 that the most crowded time during the weekdays is from $08PM$-$12AM$ with the crowd concentrating in the midtown and lower Manhattan. Intuitively this is likely due to the presence of the most popular venues in NYC in these areas such as Times Square, Carnegie Hall, One World Trade Center, and Chinatown. In

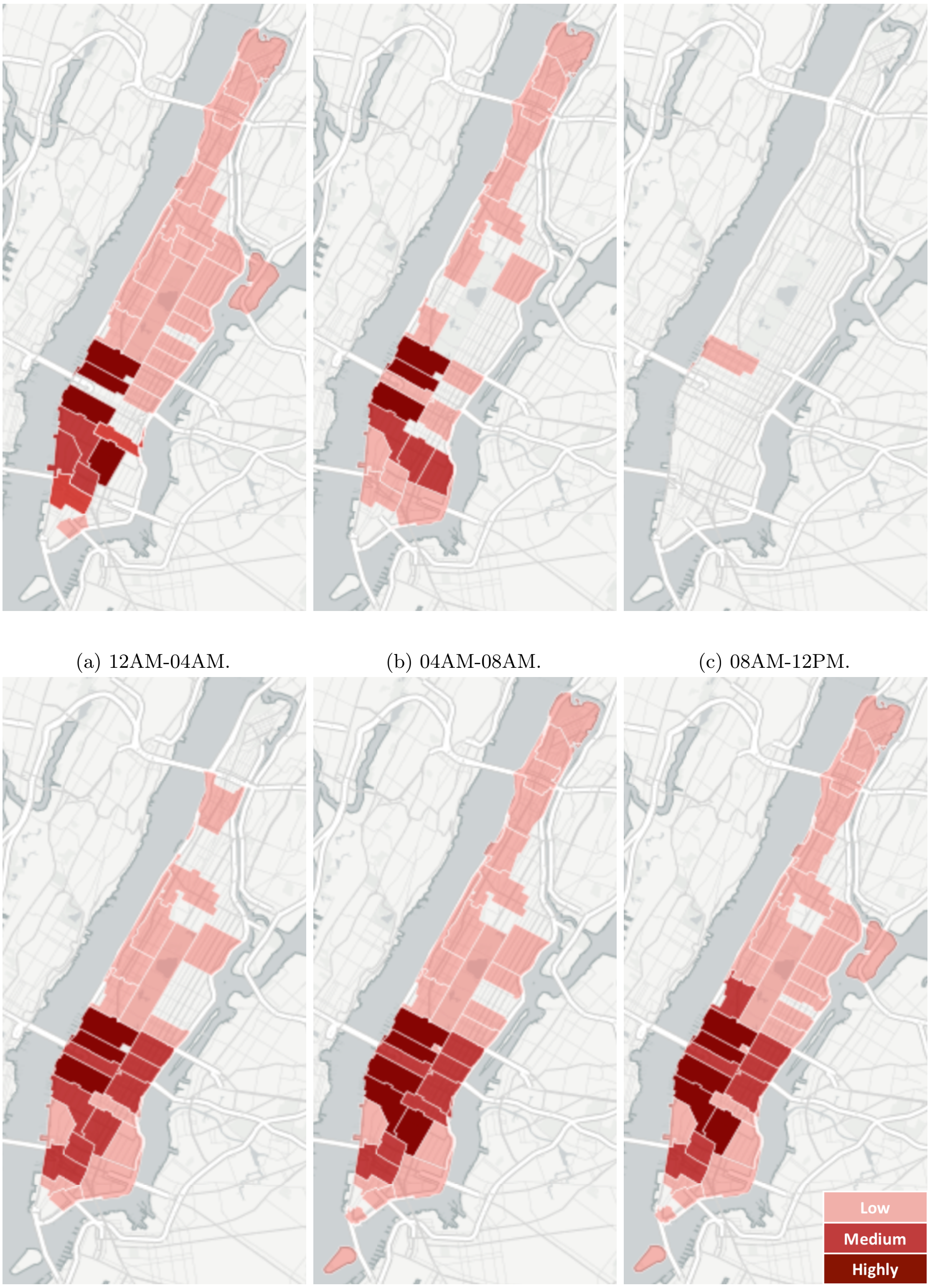


(a) Weekdays pattern.

(b) Weekends pattern.

Figure 7.4: Probability of each day belonging to certain topic (Topics = 2).

addition, it is observed that in this time-slot, the upper west side of Manhattan is crowded. This could be interpreted as being due to the presence of very popular theatres as well as the metropolitan opera; one of the most popular venues in NYC. Furthermore, upper manhattan at this time-slot was found to be still crowded but with lower crowd intensity. From $04PM$-$08PM$, the concentration of crowd is similar to the prior time-slot for the highly crowded areas with an obvious difference that the intensity of the crowd is less for the upper and upper west side areas. From $12PM$-$04PM$, generally, it can be observed that the crowd concentration is less from the prior time-slot sustaining the highly crowded areas for the most popular areas in midtown and lower manhattan. From $08AM$-$12PM$, this could be seen as the least crowded time-slot during the weekdays. By going earlier, it is observed that crowd starts again to shift to lower and midtown of Manhattan. Figure 7.6 shows the extracted recurrent crowd mobility pattern for the weekends. One of the main differences compared to the weekdays pattern is that the most crowded time-slot is from $04PM$-$08PM$. This can be explained by an intuitive understanding that people start going out on weekends earlier than weekdays. In addition and following the common sense, the Central Park Manhattan area can be seen to be crowded from $12PM$-$04PM$ and from $04PM$-$08PM$ unlike the weekdays pattern. In addition and although that from $08AM$-$12PM$ is still the least crowded time-slot but it is more crowded if compared to the weekdays patterns. It is worth noticing that the most recurrent crowded time-slot across the whole week is found to be from $04PM$-$08PM$ on the weekends. It is interesting to gather further insights about this observation when extracting different patterns for Saturdays and Sundays when the number of topics equals to 4. This is illustrated in the next section.



(a) 12AM-04AM.

(b) 04AM-08AM.

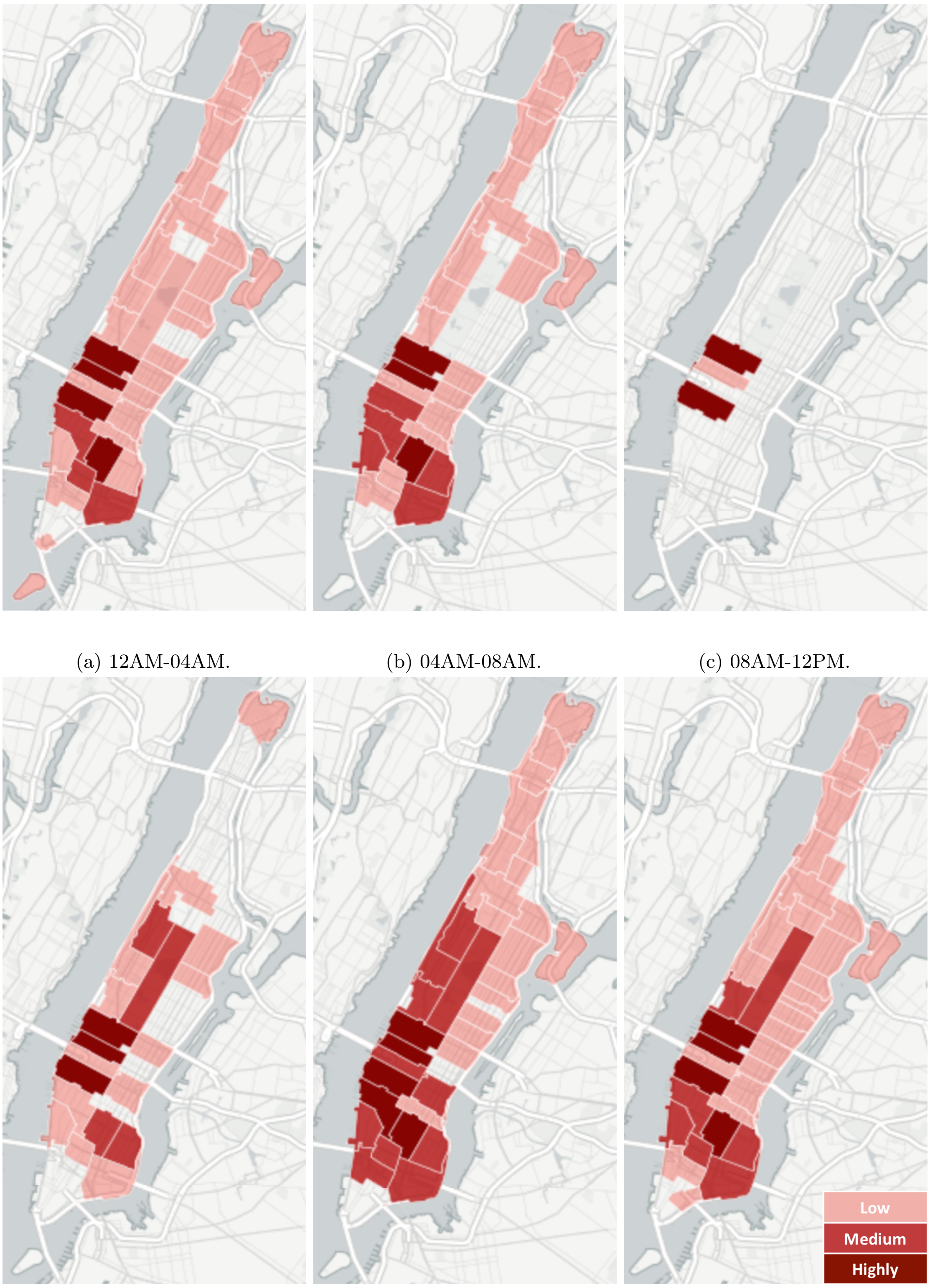
(c) 08AM-12PM.

(d) 12PM-04PM.

(e) 04PM-08PM.

(f) 08PM-12AM.

Figure 7.5: Topic 0 (weekdays).



(a) 12AM-04AM.

(b) 04AM-08AM.

(c) 08AM-12PM.

(d) 12PM-04PM.

(e) 04PM-08PM.

(f) 08PM-12AM.

Figure 7.6: Topic 1 (weekends).

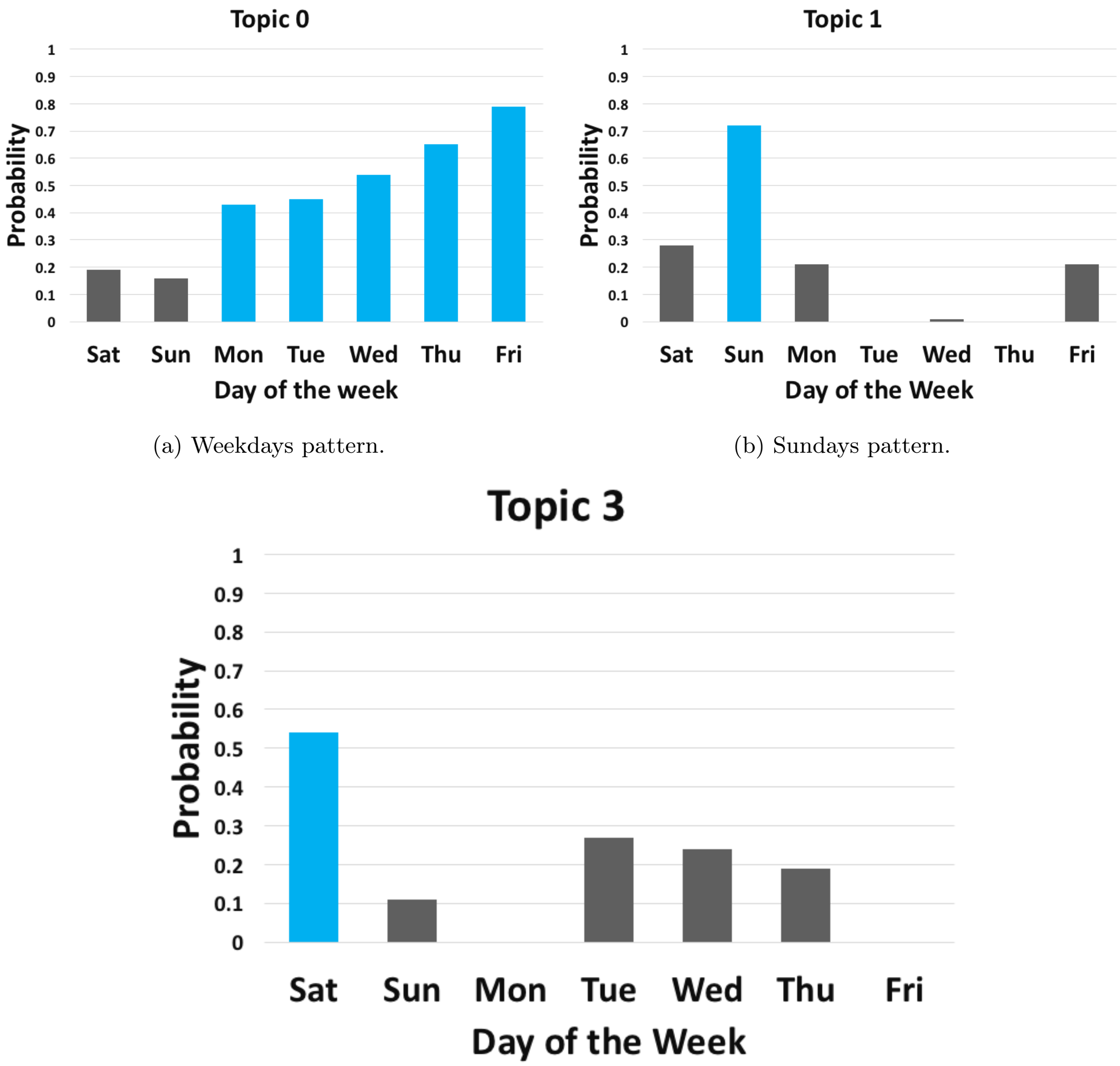
Topic distribution when utilizing four topics: Figure 7.7 shows three different patterns extracted when setting the number of topics to 4. Similarly to the previous section, the x axis indicates the day of the week and the y axis represents the probability of a particular day belonging to a certain topic. As shown from Figure 7.7a, the pattern depicted by Topic 0 has a probability greater than 0.5 for Wednesdays, Thursdays, and Fridays and around 0.45 for Mondays and Tuesdays. Compared to the extracted pattern for weekdays in case of 2 topics, the weekdays pattern extracted from 4 topics is less significant but at the expense of extracting finer patterns for the weekends. Topic 1 as shown in Figure 7.7b highlights the pattern for Sundays with more than 0.7 significance and Figure 7.7c represents the Saturdays pattern with more than 0.5 significance. To sum up, in the case of 4 topics, there was success in extracting the maximum number of patterns with the maximum number of topics (7 topics as shown in Table 7.1). In addition, compared to 2 topics, the significance of the pattern is less, but resulted in extracting different pattern for each of the weekend days (Topic 1 and Topic 3). Figure 7.8, Figure 7.9, and Figure 7.10 visualize the top-150 words representing the extracted Recurrent Crowd Mobility Patterns for the weekdays (Topic 0), Saturdays (Topic 1), and Sundays (Topic 3) respectively.

It is observed from the weekdays pattern visualized in Figure 7.8 that it follows a very similar crowd mobility pattern to that of the weekdays extracted when the number of topics equals 2 (refer to Figure 7.5). When the number of topics equals 4, it is possible to extract different recurrent crowd mobility pattern for each of the weekend days that had not been possible when the number of topics is set to 2. From Figure 7.9, it is observed that the most crowded time-slot on Saturdays is from $08PM$-$12AM$ compared to $04PM$-$08PM$ on Sundays (see Figure 7.10). Again this is validated by common sense as people prefer to return earlier usually on Sundays for getting ready for the start of the working week. For the $04PM$-$08PM$ time-slot on Sundays, nearly all areas in the upper west side are crowded. For Sundays, it is observed that the central park of Manhattan is getting crowded in the afternoon, evening, and early night.

It is worth highlighting from the visualized maps in all cases that the time-slot with the least number of crowded areas is from $08AM$-$12PM$ which follows the same statistics of the dataset (refer to chapter 4, Figure 4.2c)[4].

Topics Stability: Due to the difficulty of assessing the accuracy of the extracted recurrent mobility patterns due to the lack of ground truth, the purpose of this discussion is to validate the extracted topics through performing a topics/patterns stability analysis. This requires introducing a topic stability metric for assessing the ability for a topic-based algorithm to extract similar patterns on new unseen dataset. Since the main purpose of the proposed approach is to extract "recurrent" crowd mobility patterns, it is expected that such patterns will be similar even in a smaller time duration. However, the motivation

[4] The dataset statistics shows that from nearly $7AM$ to $12PM$ is the least frequency of check-ins for all days of the week



(a) Weekdays pattern.

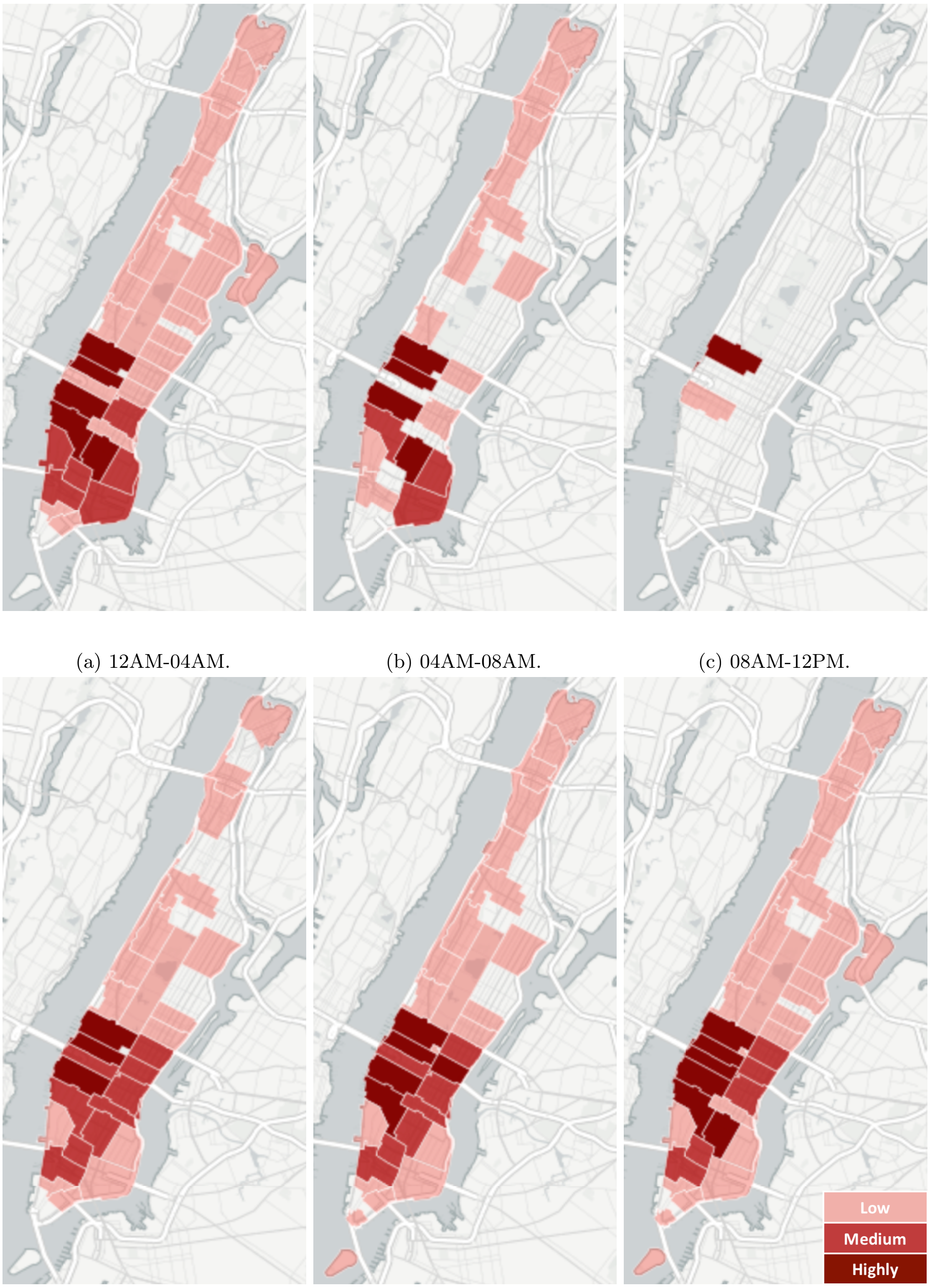
(b) Sundays pattern.

(c) Saturdays pattern.

Figure 7.7: Probability of each day belonging to certain topic (Topics = 4).

for extracting the patterns on a bigger dataset is obviously for accuracy perspective. For this purpose, the dataset D, introduced in chapter 4 comprising of 24 months is divided into 18 months for the training dataset and 6 months for the testing dataset.

In order to define the topic stability, we define a “topic” first as a set consisting of three other subsets, where each subset represents a certain level of crowd intensity, i..e, “Low” (L), “Medium” (M), or “High” (H), as follows: $Topic = \{L, M, H\}$. For instance, a topic can be represented as follows: Topic 0 (from training dataset) = {{2zip10002, 3zip10003, 4zip10019}, {1zip10001, 4zip10018}, {2zip10003, 5zip10022}}, where for instance L = {2zip10002, 3zip10003, 4zip10019}. This is generated across both the training and test dataset, denoted further by TR_X and TE_X, $X \in Topic$, respectively, where each subset consists of a set of words (e.g., $2zip10002$).



(a) 12AM-04AM.

(b) 04AM-08AM.

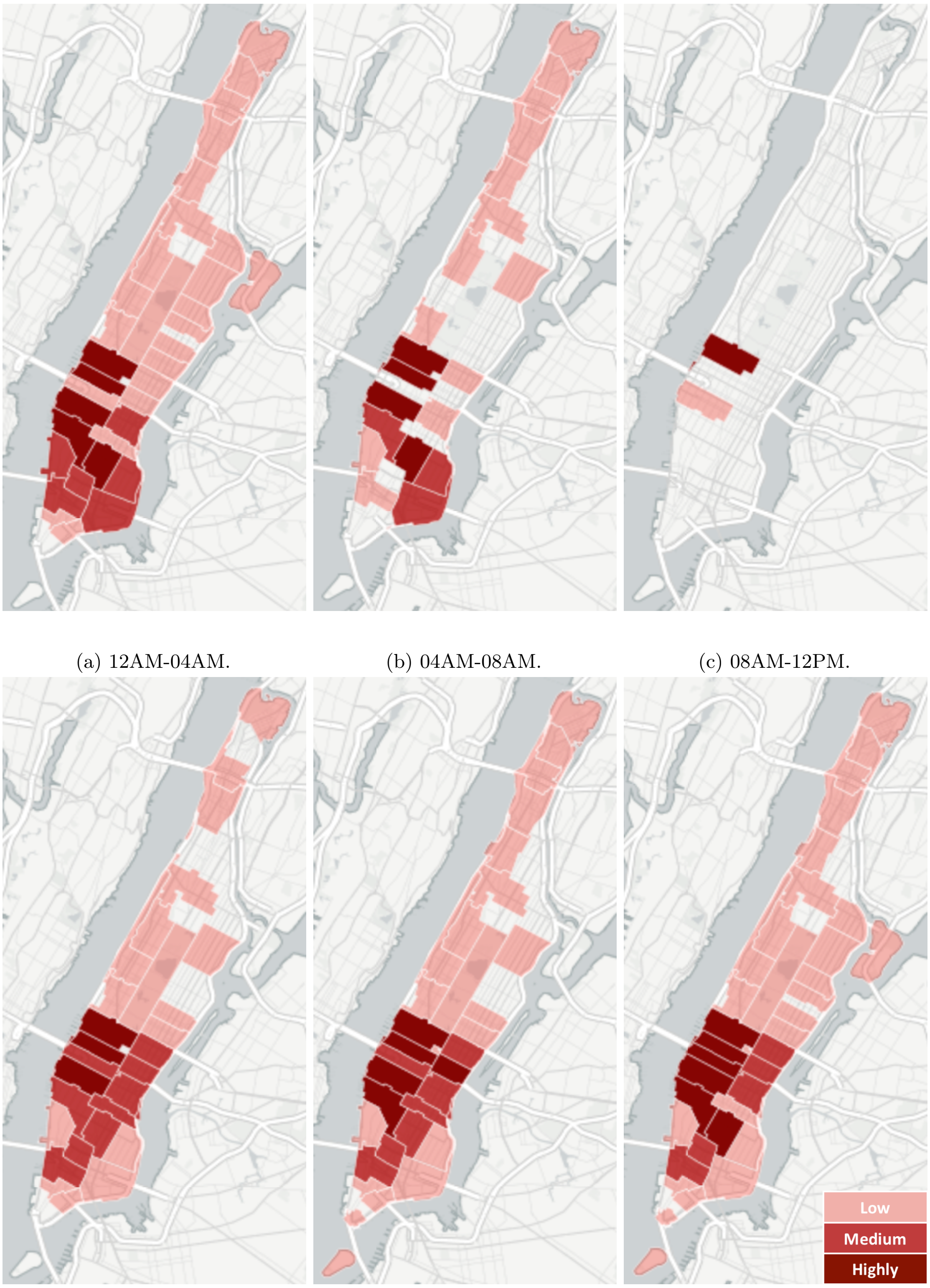
(c) 08AM-12PM.

(d) 12PM-04PM.

(e) 04PM-08PM.

(f) 08PM-12AM.

Figure 7.8: Topic 0 (weekdays).



(a) 12AM-04AM.

(b) 04AM-08AM.

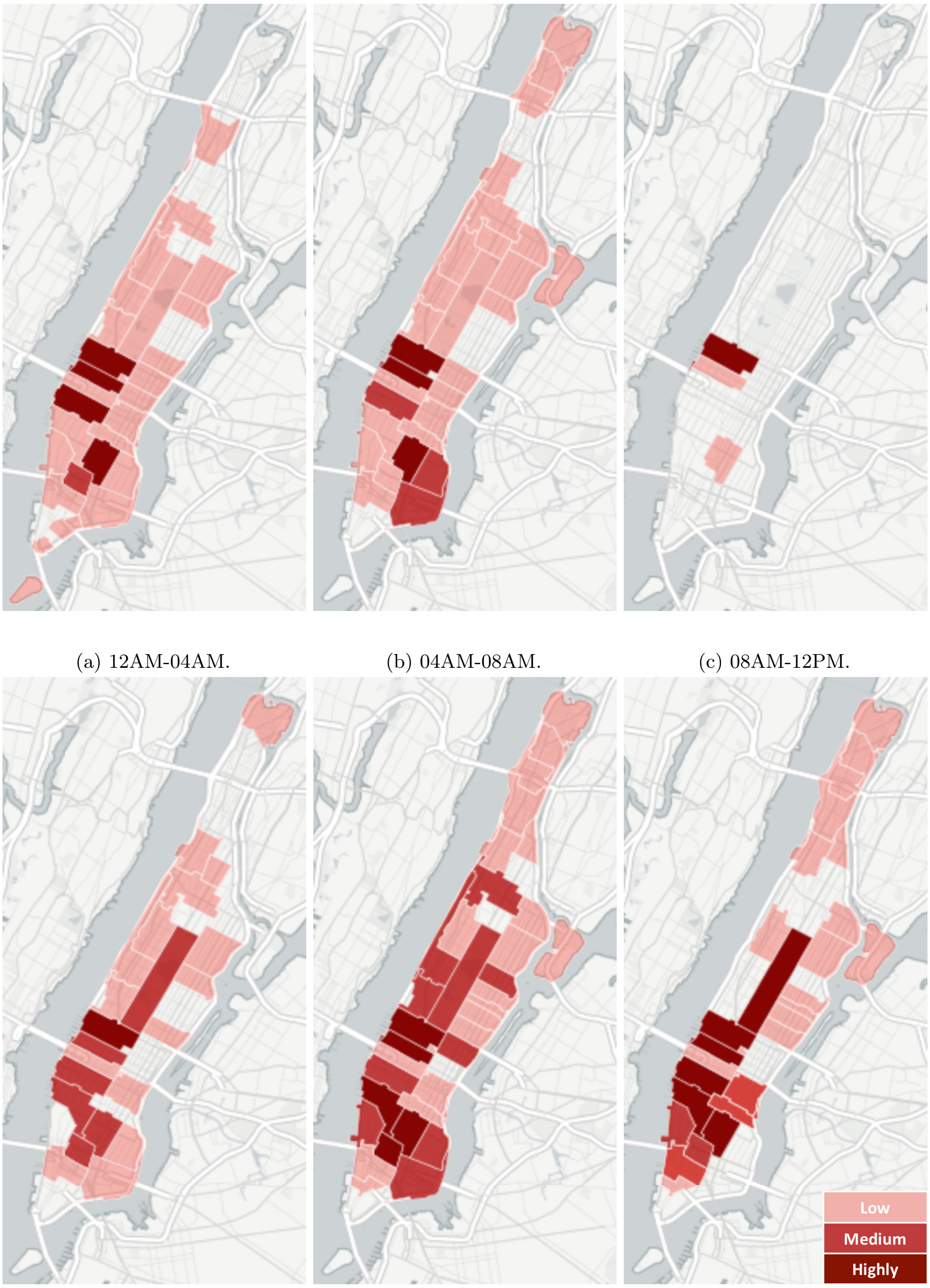
(c) 08AM-12PM.

(d) 12PM-04PM.

(e) 04PM-08PM.

(f) 08PM-12AM.

Figure 7.9: Topic 1 (Saturdays).


(a) 12AM-04AM. (b) 04AM-08AM. (c) 08AM-12PM.

(d) 12PM-04PM. (e) 04PM-08PM. (f) 08PM-12AM.

Figure 7.10: Topic 3 (Sundays).

We further introduce the stability metric $\mathcal{ST}$ of a subset X as follows:

$$\mathcal{ST}(X) \ = \ \{\frac{\|TR_X \cap TE_X\|}{\|TE_X\|} \cdot 100\} \ , \ \forall X \in Topic \tag{7.6}$$

where $\|X\|$ represents the number of elements in the set X. For instance, $\|TE_X\|$ and $\|TR_X\|$ represent the number of words in the training and test set of X, respectively.

The stability of a topic is the set of stabilities of each subset X:

$$\mathcal{ST}(X) = \{\mathcal{ST}(L), \mathcal{ST}(M), \mathcal{ST}(H)\}$$

Let us consider the following example for clarification: Topic 0 (from training dataset) = {{2zip10002, 3zip10003, 4zip10019}, {1zip10001, 4zip10018}, {2zip10003, 5zip10022}} and Topic 0 (from testing dataset) = {{2zip1002, 3zip1003, 4zip10024}, {1zip10022,4zip10018}, {2zip10015,5zip10023}}. Both topics resulted as the pattern for a certain day. By taking the first set in each topic, it was observed that 2 words are common from the 3 words that exist in the test dataset. Hence, in this case $\mathcal{ST}(L) = \frac{2}{3} \cdot 100 = 66.67\%$. Following the same process, the topic stability across the three sets is: $\mathcal{ST}(Topic0) = \{66.67\%, 50\%, 0\%\}$.

In order to have a fair comparison between 2 topics where each is extracted from a different dataset, the number of words in each should be equivalent to same amount of variance that is captured by each topic. For example, if 100 words capture 90% of the variance of Topic 0 extracted from dataset D_1, then we should be looking for the number of words that captures same amount of variance in Topic 0 extracted from dataset D_2, which might be different than 100 depending on the size of the dataset. In this research, capturing 90% of variance is considered sufficient for claiming that most of the pattern is captured by the model and hence, the top 150 and 60 words are utilized for the selected training and test dataset, respectively. This is shown in Figure 7.11, where the x axis indicates the number of words and the y axis represents the variance captured by the topic from certain number of words. Similarly, the same process is followed when the number of topics equals to 4, and similar number of words are concluded for the training and test datasets.

Baseline Comparisons: The RCMC proposed approach for extracting Recurrent Crowd Mobility Patterns has been compared with the following baselines for evaluating its effectiveness:

- **LDA:** In this baseline, the crowded areas are captured by counting the number of Twitter posts sent by users and then Latent Dirichlet Allocation (LDA) is used for extracting Recurrent Crowd Mobility Patterns. LDA [71] as introduced in chapter 3 is an unsupervised learning algorithm that models each document as a mixture of topics. The model generates topics in terms of a discrete probability distribution over words for each topic, and then the per-document discrete distributions over topics is

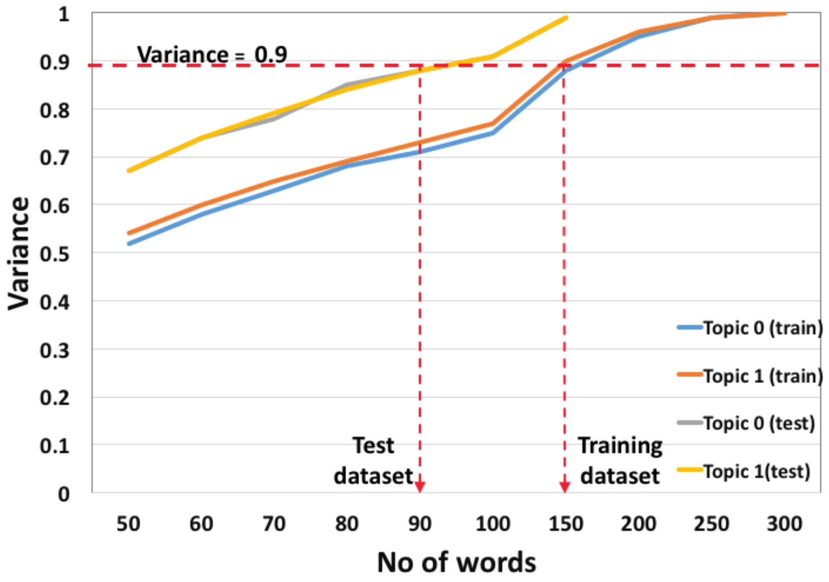


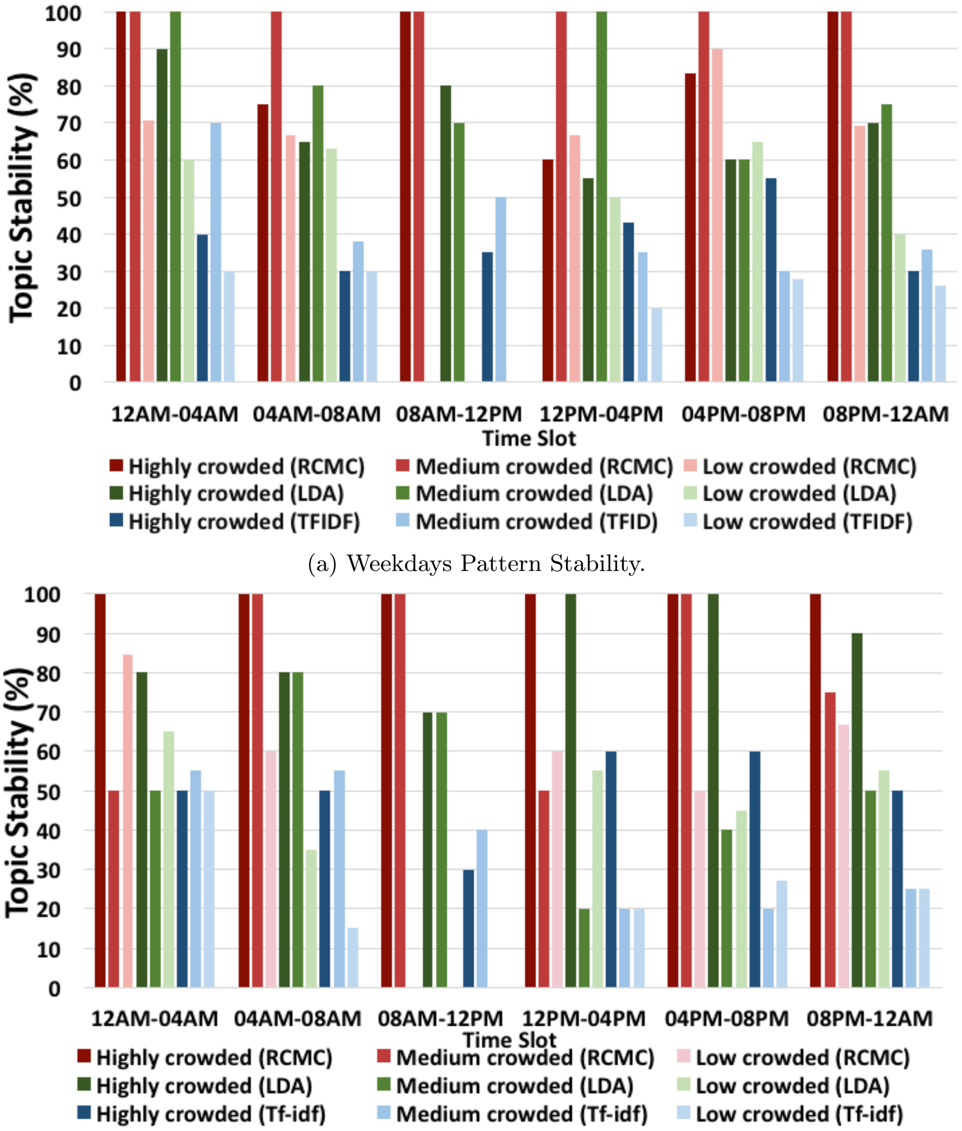
Figure 7.11: Total variance captured by topics with various number of words (Number of Topics = 2).

inferred. It is worth highlighting that this baseline is nearly the same approach to what has been proposed by Laura et al. [38].

- **Tf-idf:** Similar to the previous baseline, the crowded areas are captured by counting the number of Twitter posts sent by users then the Term Frequency-Inverse Document Frequency (Tf-idf) is used for extracting the crowd patterns. Tf-idf is a well-known method for scoring the importance of words in a document based on how frequently they appear across several documents.

Figure 7.12 and Figure 7.13 show the results of the topics/patterns stability when the number of topics equals 2 and 4 respectively comparing our proposed approach with the baselines. The proposed comparisons show that RCMC consistently and significantly outperform the baselines. From Figure 7.12, it is worth highlighting the following points: (a) The patterns extracted by RCMC are 100% stable for the "Medium" crowded areas for all time-slots for the weekdays. LDA performed similarly only on two time slots ($12AM$-$04AM$ and $12PM$-$04PM$) while it showed at least 20% worse performance compared to RCMC. Tf-idf has very bad performance with 50% topic stability across all time-slots except for the $12AM$-$04AM$ with 20% stability (See Figure 7.12a); (b) For the weekends and as shown in Figure 7.12b, the patterns extracted by RCMC are 100% stable for the "Highly" crowded areas for all time-slots. LDA scored similarly only for two time slots while it performed worse compared to RCMC with at least 20% on the rest of the time-slots. Similarly to the weekdays patterns, Tf-idf performed the worst.

From Figure 7.13, the following points are observed: (a) From Figure 7.13a, the RCMC approach performed the best across all time-slots except the $08AM$-$12PM$ for the "High" and "Low" crowded areas. (b) For Mondays and as shown in Figure 7.13b, the RCMC



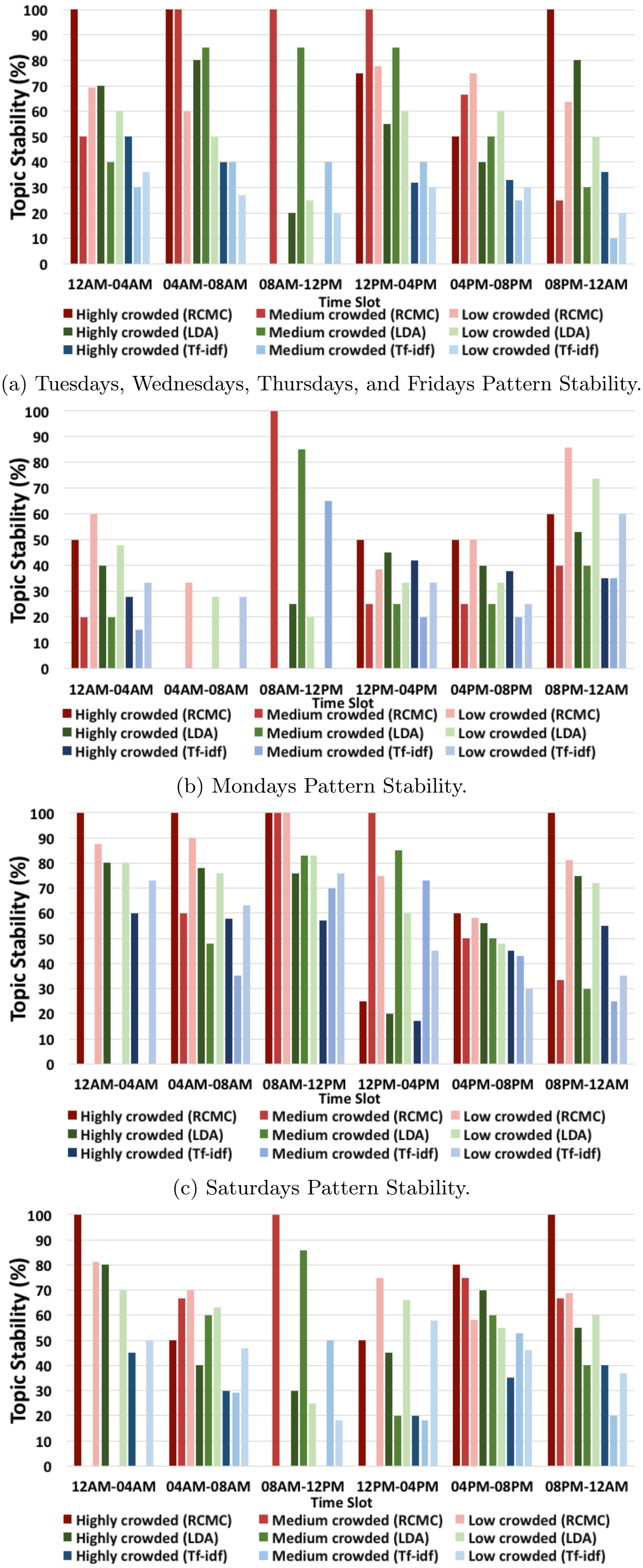
(a) Weekdays Pattern Stability.

(b) Weekends Pattern Stability.

Figure 7.12: Topics Stability (Number of Topics = 2).

performed still the best across all time-slots but with similar observation for the $08AM$-$12PM$ in which baselines performed better for the “High” and “Low” crowded areas. (c) For Saturdays and as shown in Figure 7.13c, RCMC performed the best with at least of $10 - 20\%$ difference compared to the baselines. (d) For Sundays and as shown in Figure 7.13d, RCMC outperformed all baselines but with an exception again for the $08AM$-$12PM$ time-slot.

Generally, for all of the previous discussed points, it is clear that RCMC outperforms the baselines in the extraction of recurrent mobility patterns compared to the baselines. The $08AM$-$12PM$ time-slot is an exception when the number of topics equals 4 in some cases, there is no clear scientific justification of this observation except that generally the $08AM$-$12PM$ time-slot seems its crowd pattern is not too obvious and that was clear from the visualisations maps presented before. LDA performed the second after RCMC



(a) Tuesdays, Wednesdays, Thursdays, and Fridays Pattern Stability.

(b) Mondays Pattern Stability.

(c) Saturdays Pattern Stability.

(d) Sundays Pattern Stability.

Figure 7.13: Topics Stability (Number of Topics = 4).

and Tf-idf performed the worst in nearly all cases. As shown from our experiments the overall stability when the number of topics is 2 is better compared to 4 topics. This was intuitively expected that the more number of topics, the less resulted stable topics that can be detected from the test dataset compared to the training dataset. This is due to the fact that the size of the testing dataset is usually less than the training dataset and hence, it would be harder to capture similar patterns from the testing dataset.

7.6 Case study - Correlating Temporal Functional Regions patterns with Recurrent Crowd Mobility Patterns

In this section, a few physical regions in Manhattan are highlighted as examples of how correlating Temporal Functional Regions pattern with recurrent crowd mobility pattern could deliver useful insights about the motivation for crowd mobility in cities. In particular, the focus is on the extracted recurrent mobility patterns when the number of topics is 2 (see Figure 7.5 and Figure 7.6). Figure 7.14 highlights the regions[5] selected for analysis:

- **Zip-10025**: This region has been categorized with "Eating" functionality during all of the time-slots in the day except from $04AM$-$08AM$, when it belongs to "Night Life" which intuitively corresponds with the functionality expected at such a time. It can be observed that this area has been ranked as a "Low" crowded area during all time slots in the weekdays and "Medium" crowded for the afternoon and evening time slots, naturally for the purpose of "Eating".

- **Zip-00083**: This region corresponds to the central park in Manhattan which the introduced Temporal Functional Regions approach from the previous chapter categorized it as expected as a "Recreation" area in nearly all time slots during the day. It is identified as "Low" or not crowded area during the weekdays and "Medium" crowded area over the weekends for the evening and early nights durations which follows what is expected from such an area with this functionality.

- **Zip-10036**: This region is identified by RCMC as "Highly" crowded during most of the time-slots for the weekends and weekdays except from $08AM$-$12PM$ time-slot during the weekdays which could be justified as it is late sleeping area and hence, people's activities are expected to start a bit late at least in the weekdays. This is not surprising for a region that comprises of Times Square, which is a major commercial intersection and neighbourhood in Manhattan, sometimes referred to as The Center of the Universe, and the heart of the world. As highlighted from the previous chapter, the temporal functional region introduced approach categorized this area with dominant functionality of "Traveling" from $12AM$-$04AM$ and from $12PM$-$04PM$ which follows an expected pattern temporally and spatially for one of

[5] We tried to select the most dynamic regions/zip-codes based on our research and understanding of the popular/dynamic areas in NYC.

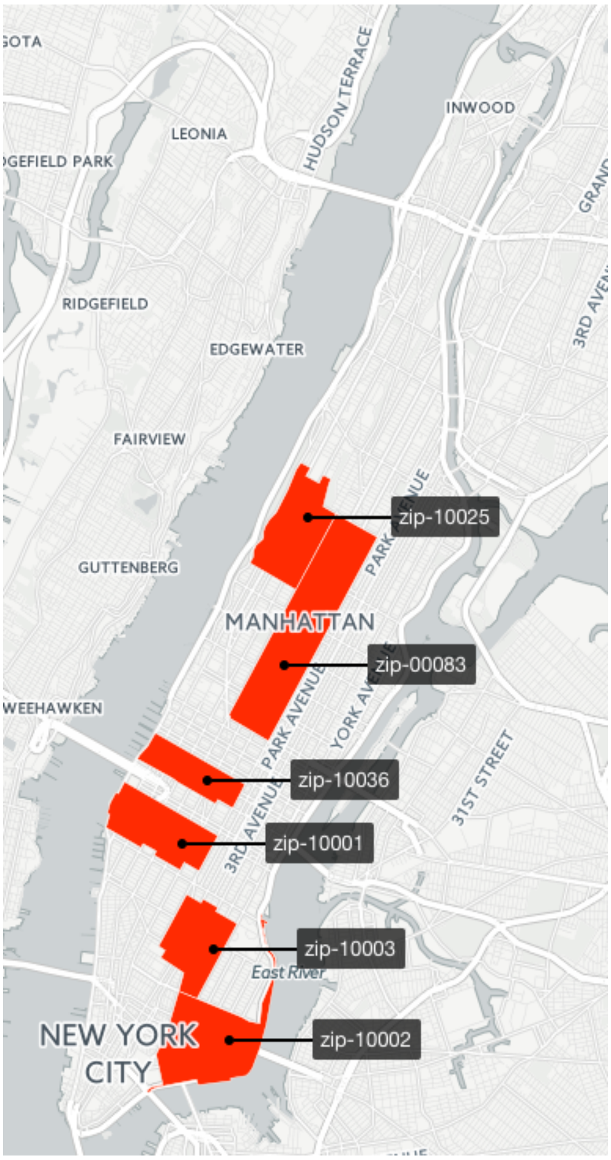


Figure 7.14: Case study for some particular zip-codes.

the world's most visited touristic areas in the world. From 04AM-08AM, this area is classified as "Night Life" area which is not surprising for an area that comprises very popular night life spots including clubs, bars, etc. In Evening and Early Night, its functionality is mainly "Eating" due to the huge amount of cafes and restaurants in this area, while from 12PM-04PM, this area is found as "Social Services" area which follows the temporal expectation for such a time interval. As it can be seen, this area is very rich in its temporal functionalities variations due its classification as always "Highly" crowded area by the RCMC approach.

- **Zip-10001**: From 12AM-04AM, this region is categorized as "Highly" crowded and this could be understandable due to that its temporal functionality at this time-slot is found to be "Traveling" as it owns one of the main train stations in NYC that is open for 24 hours (Pennsylvania Station as well as 33rd Street train station) that connects

Manhattan with Brooklyn and west NY. Similarly from $08AM$-$12PM$, the region is classified as “Highly” crowded in weekends for the same functionality (“Traveling”). From $12PM$-$04PM$, the region is classified as “Highly” crowded during weekdays and weekends for the sake of “Social Services” which follows most of the neighbourhood areas’ functionalities at the same time-slot. It is worth highlighting that this area contains Union Square, which is very popular with its dinning and eating venues and hence follows the motivational crowds shift patterns detected in this region.

- **Zip-10003**: This region starts from $12AM$-$04AM$ in the weekdays as “Highly” crowded for “Eating” functionality. Then “Medium” crowded from $04AM$-$08AM$ for “Night Life” functionality compared to Zip-10036 which is “Highly” crowded at same time-slot for same purpose. As it is a region with late sleeping patterns, the area tends to be “Low” or not crowded from $08AM$-$12PM$ before the crowd shifts again towards it for “Eating” from $12PM$-$04PM$ (the expected time for lunch) and then “Highly” crowded during Evening and Early Night.

- **Zip-10002**: For this region, it is observed that is “Low” or not crowded during the weekdays but it is “Medium” crowded during the weekends for the purpose of “Eating” with an exception of the time-slot from $08AM$-$12PM$ which is not crowded. This follows the features of the region as crowd shifts towards it mainly in weekends for “Eating” and hence, no crowd in weekends for the morning slots.

7.7 Summary

In this chapter, a new approach for Recognizing Crowd Mobility Patterns in Cities (titled RCMC) using LBSNs data has been introduced. The proposed approach was shown to be capable of extracting *Recurrent Crowd Mobility Patterns* with an estimation of the crowd intensity utilizing a KDE/NMF based approach. The proposed approach is evaluated on the LBSNs data that was introduced in chapter 4. It is further shown that the proposed approach outperforms two baseline topic-based models through a “stability” evaluation metric. Through a case study for some particular regions, it is further shown in this chapter that the correlation between the extracted Recurrent Crowd Mobility Patterns with the Temporal Functional Regions (introduced in the previous chapter) can provide further insights around the motivation behind crowd mobility. The extracted patterns from the proposed RCMC approach have the potential to benefit a wide variety of applications. For example, from the urban planners perspective, it can help them understand how crowd shifts across space and time allowing them to better allocate cities’ resources. As well personalized recommendations for activity could be provided to people based on the region’s functionality during certain times of the day. New home buyers could use the correlation between Recurrent Crowd Mobility Patterns and Temporal Functional

Regions to understand investment values for real estate through understanding when crowd mobility shifts to a particular region and for what purpose.

Having presented in this chapter the third extracted urban pattern, the next chapter introduces a new deep learning based approach called *St-DenNetFus* that fuses two of the previous extracted patterns with network data for showing the potential impact of such extracted urban patterns in helping to solve one of the challenges in the telecommunications service provider domain (Network Demand Prediction).

Chapter 8

Network demand prediction using the extracted urban patterns

Chapter overview: *This chapter introduces a deep-learning based approach called, ST-DenNetFus, to forecast network demand (i.e. uplink and downlink throughput) in every region of a city. ST-DenNetFus is an end to end structure for capturing unique properties from spatio-temporal data. ST-DenNetFus employs various branches of dense neural networks for capturing temporal closeness, period, and trend properties. For each of these properties, dense convolutional neural units are used for capturing the spatial properties of the network demand across various regions in a city. Furthermore, ST-DenNetFus introduces extra branches for fusing external data sources of various dimensionalities, in our case, these external factors are the crowd mobility patterns, Temporal Functional Regions, and the day of the week. This new approach has been submitted to [TKDE 2017]. The proposed approach not only shows the indirect benefit that could result from the extracted patterns discussed in the previous chapters on one of the most important challenges in the telecommunications domain, Network Demand Prediction, but also argues that the proposed framework could be leveraged for any other spatio-temporal prediction problems that requires fusing external data sources of various dimensionalities.*

8.1 Motivation

Mobile data traffic has increased dramatically in the last few decades [123]. Besides, the increase in the number of devices accessing the cellular network, emerging social networking platforms such as Facebook and Twitter has further added to the mobile data traffic [124]. This led to the need to increase the network resources provided to end-users and consequently this has caused a huge cost increase on the operators. The mobile network operators are striving for solutions to reduce the OPEX and CAPEX costs of such emerging demand. Reducing the OPEX and CAPEX cost is not only of importance to the operators but as well to the environment. The statistics show that the total CO2 emissions from the information and communication technology (ICT) infrastructure contributes for

2% of total CO2 submissions across the globe in which the telecommunication industry is a major part of it [125].

In this chapter, two types of network throughput are predicted: downlink and uplink. Downlink is the total downloaded network throughput in a region during a given time interval. Uplink denotes the total uploaded network throughput in a region during a given time interval. Both types of throughput track the overall Network Demand[1] of a region. There are significant spatial and temporal variations in cellular traffic [126] and the cellular system is designed using the philosophy of worst case traffic (such as to fulfil the quality of service (QoS) in case of a peak traffic). Hence, there is a growing need to have a spatio-temporal prediction based model for the Network Demand Prediction problem [127].

As discussed in chapter 3, deep learning [128] has been applied successfully in many applications, and is considered one of the most cutting edge techniques in Artificial Intelligence (AI). There are two types of deep neural networks that tries to capture spatial and temporal properties: a) Convolutional Neural Networks (CNNs) for capturing spatial structure and dependencies. b) Recurrent Neural Networks (RNNs) for learning temporal dependencies. However, it is still very challenging to apply these type of techniques to the spatio-temporal Network Demand Prediction problem due to the following reasons:

1. Spatial dependencies:
   Nearby - The downlink throughputs of a region might be affected by the uplink throughputs of nearby regions and vice versa. In addition, the downlink throughputs of a region would affect its own uplink throughputs as well.
   Distant - The network demand of a region can be affected by the network demand of distant regions especially if both are supported by same Base Station geographically.

2. Temporal dependencies:
   Closeness - Intuitively, the network demand of a region is affected by recent time intervals. For instance, a high network demand occurring due to a crowded festival occurring at $9PM$ will affect that of $10PM$.
   Period - Network demand during morning or evening hours may be similar during consecutive weekdays, repeating 24 hours.
   Trend - Network demand may increase as summer approaches especially on weekends. Recent study showed that the summer usage increases in the evening and early morning hours from about midnight to $4AM$, which indicates that teens and young adults are not putting down mobile devices just because it is a summer break.

[1] In our terminology, we refer to both types of throughputs uplink and downlink as Network Demand.

3. External Factors:
   Multiple - Some external factors may impact the network demand such as the Temporal Functional Regions, crowd mobility patterns and day of the week. For example, a business functional region may rely on the wireless networks more than the cellular networks. In addition, a highly crowded area has a higher chance for more network usage.
   Dimensionality - The external factors may vary in the dimensionality of the data. For example, the day of the week data will be in 1-dimensional space since it varies across time only but crowd mobility or Temporal Functional Regions data will be in 2-dimensional space since it varies across space and time.

To tackle the above challenges, in this chapter a spatio-temporal deep learning based architecture called ST-DenNetFus is proposed that collectively predict the uplink and downlink throughputs in every region. The proposed contributions in this chapter are five-fold:

1. ST-DenNetFus employs convolutional-based dense networks to model both nearby and distance spatial dependencies between regions in cities.

2. ST-DenNetFus employs several branches for fusing various external data sources of different dimensionality. The architecture proposed is expandable according to the availability of the external data sources needed to be fused.

3. ST-DenNetFus uses three different dense networks to model various temporal properties consisting of temporal closeness, period, and trend.

4. The proposed approach has been evaluated on a real network data extracted from NYC and in particular Manhattan, for 6 months. The results reinforces the advantages of the new approach compared to 4 other baselines.

5. For the first time, it is shown that the extracted urban patterns (specifically crowd mobility and Temporal Functional Regions) when fused as an external data sources for estimating the network demand, leads to more accurate prediction results.

8.2 State-of-the-Art

In this section, the state-of-the-art is reviewed from two different perspectives. First, the recent advancements of convolutional neural networks are discussed and then an overview on the Network Demand Prediction related work is presented.

8.2.1 Convolutional neural networks advancements

In the last few years, deep learning has led to very good performance on a variety of problems, such as visual recognition, speech recognition and natural language processing [129]. Among different types of deep neural networks, convolutional neural networks (CNN) have been most extensively studied. CNN is inspired by the natural visual perception mechanism of living creatures [130]. Hubel & Wiesel [131] in 1959 found that cells animal visual cortex are responsible for detecting light in receptive fields. In 1980 and inspired by this discovery, Kunihiko Fukushima [132] proposed neocognitron which could be seen as the predecessor of CNN. The first modern framework of CNN is later introduced in 1990 by LeCun [132] and has been further improved in [133]. LeCun's paper introduces a neural network called LeNet-5 of multiple layers and is trained using the backpropagation algorithm [134] for classifing handwritten digits. LeNet-5 has shown effectiveness in extracting representation of the original image, which makes it possible to recognize visual patterns directly from raw pixels with little preprocessing. A parallel study by Zhang et al. [135] introduced a network called SIANN which stands for shift-invariant artificial neural network to recognize characters from an image. However, it was found to be very challenging for both networks to perform well on more complex problems such as video classification due to the lack of training data the computational power available a that time [136].

Since 2006, various methods have been developed to overcome the limitations and challenges encountered in training deep CNNs. The most notable work started by Krizhevsky et al. when they introduced an architecture called AlexNet [137]. The overall architecture of AlexNet is similar to LeNet-5 but with deeper structure and showed significant improvements compared to LeNet-5 on the image classification task. With the success of AlexNet, several successful architectures have evolved, ZFNet [138], VGGNet [139], GoogleNet [140] and ResNet [141]. One of the main typical trends with these evolving architectures is that the networks are getting deeper. For instance, ResNet, the winner of ILSVRC 2015 competition got deeper 20 times more deeper than AlexNet and 8 times deeper than VGGNet. This typical trend is because networks can better approximate the target function when they are deeper. However, the deeper the network the more complex it is, which makes it more difficult to optimize and easier to suffer overfitting. Of course, various methods are proposed to deal with these problems in various aspects. Recently in 2016, a new architecture has been introduced called DenseNets [142] that exploits the potential of the network through feature reuse, yielding condensed models that are easy to train and highly parameter efficient. DenseNets obtain significant improvements over most of the state-of-the-art networks to date, whilst requiring less memory and computation to achieve high performance [142]. Hence, in this work we rely mainly on leveraging the dense blocks as a core part of the proposed ST-DenNetFus architecture as will be described in the sections to follow. To the best our knowledge, this is the first work to show the effectiveness of DenseNet on a different domain than computer vision.

8.2.2 Network Throughput Prediction

Cellular network throughput prediction plays an important role in network planning. This section overviews some of the approaches that focus on traffic and throughput prediction. In previous related work, AutoRegressive models have been very popular in application of Network Demand Prediction but without taking spatial dependencies into account. To name a few works, in [16], ARIMA and exponential smoothing model are used for predicting the network demand for a single cell and whole region scenarios. ARIMA was found to outperform for a whole region scenario while the exponential smoothing model had better performance for the single cell scenario. In [143], a hybrid method using both ARMA and FARIMA is introduced to predict the cellular traffic where FARIMA found to work effectively on the time series that holds long range dependence. For long time prediction, the authors in [144] presented an approach with 12-hour granularity that allows to estimate aggregate demands up to 6 months in advance. Shorter and variable time scales are studied in [145], [146] adopting AutoRegressive Integrated Moving Average (ARIMA) and Generalized AutoRegressive Conditionally Heteroskedastic (GARCH) techniques.

There are several pieces of work focused on taking into account the spatio-temporal parameters when predicting network demand. In [127], regressors are introduced for different performance indicators at different spatio-temporal granularity for mobile cellular networks focusing on per-device throughput, base station throughput and device mobility. Similar to this scope, the authors in [147] focus more on core network measurements where mobile device traffic data is collected from a cellular network operator and is used to classify IP traffic patterns of mobile cellular devices. The work presented in [148] studied traffic prediction in cloud analytics and showed that optimizing parameters and metrics can lead to accurate prediction even under high latency at the application/TCP layer to improve the performance of the application avoiding buffer overflows and/or congestion. More recently, researchers started to exploit external sources. In [15], the authors propose a dynamic network resources allocation framework to allocate downlink radio resources adaptively across multiple cells of 4G systems. Their introduced framework leverages three types of context information: user's location and mobility, application-related information, and radio maps. A video streaming simulated use case is used for evaluating the performance of the proposed framework. Another interesting work presented in [149] focuses on building Geo-localized radio maps for a video streaming use-case in which the streaming rate is changed dynamically on the basis of the current bandwidth prediction from the bandwidth maps. The empirical collection of geo-localized data rate measures is also addressed in [150] which introduces a dataset of adaptive Hypertext Transfer Protocol (HTTP) sessions performed by mobile users. To the best our knowledge, in the field of telecommunications, end-to-end deep learning for Network Demand Prediction fusing external cities patterns has not yet been undertaken.

8.3 Network demand prediction - ST-DenNetFus

8.3.1 Notation and Definitions

Definition 7 (Region) *There are many definitions that could highlight a region such as zip-code boundary [151] or defining regions by roads networks [8]. In this work, a region is defined as a grid after partitioning the city into an $I \times J$ grid map based on the longitude and latitude for generating image like matrices so that they can be further processed by convolutional neural networks.*

Definition 8 (Downlink/Uplink Throughput) *In this work, downlink and uplink throughput is defined as the maximum downlink and uplink throughput observed in the current pixel. For a grid (i, j) that lies at the i^{th} row and j^{th} column, the downlink and uplink throughput are defined at the time interval t respectively as*

$$x_t^{down,i,j} = \sum_{\forall n \in (i,j)} |N_n^{down}| \tag{8.1}$$

$$x_t^{up,i,j} = \sum_{\forall n \in (i,j)} |N_n^{up}| \tag{8.2}$$

Problem 1 *Given the historical observations of the network demand $\{\mathbf{X_t} | t = 0, ..., s-1\}$, predict $\mathbf{X_s}$.*

At the t^{th} time interval, downlink and uplink throughputs in all $I \times J$ regions can be denoted as a tensor $\mathbf{X_t} \in \mathbb{R}^{2 \times I \times J}$ where $(\mathbf{X_t})_{0,i,j} = x_t^{down,i,j}$, $(\mathbf{X_t})_{1,i,j} = x_t^{up,i,j}$. For a grid map of dimensions $I \times J$, there are two types of network demand (Downlink/Uplink Throughput) in each grid over time thus the observation at any time can be represented by a tensor $\mathbf{X} \in \mathbb{R}^{2 \times I \times J}$.

8.3.2 Deep Dense Networks

Dense Convolutional Network (DenseNet) has been introduced recently and it has been proved that it can scale naturally to hundreds of layers without exhibiting optimization challenges [142]. It introduces direct connections between any two layers with the same feature-map size. DenseNets has achieved state-of-the-art performances with fewer parameters and less computation [142]. The main idea of DenseNet is to improve the information flow between layers by proposing a different connectivity pattern, direct connections from any layer to all the subsequent layers. This means that the l^{th} layer receives the feature-maps of all preceding layers, $\mathbf{X_0}, ..., \mathbf{X_{l-1}}$, as input:

$$\mathbf{X_l} = \mathbf{H}_l([\mathbf{X_0}, ..., \mathbf{X_{l-1}}]), \tag{8.3}$$

where $[\mathbf{X_0}, ..., \mathbf{X_{l-1}}]$ refers to the input and the concatenation of the feature-maps produced in layers $0, ..., l-1$. For the ease of implementation, the multiple inputs of $\mathbf{H}_l(.)$ are concatenated into a single tensor.

8.3.3 Deep Spatio-Temporal Dense Network with Data Fusion (ST-DenNetFus)

Figure 8.1 presents the proposed architecture of ST-DenNetFus. Based on definition 7 and definition 8, each of the downlink and uplink network throughputs at time t is converted to a 32×32 of 2-channel image-like matrix spanning over a region. Then the time axis is divided into three fragments denoting recent time, near history and distant history. Then these 2-channel image-like matrices are fed into three branches on the right side of the diagram for capturing the trend, periodicity, and closeness and output $\mathbf{X}_{in}$. Each of these branches starts with convolution layer followed by L dense blocks and finally another convolution layer. These three convolutional based branches capture the spatial dependencies between nearby and distant regions. Then there is a number of branches that fuse external factors based on their dimensionality. In our case, the Temporal Functional Regions and the crowd mobility patterns are 2-dimensional matrices ($\mathbf{X}_{Ext-2D}$) that change across space and time but on the other side, the day of the week is 1-dimensional matrix that change across time only ($\mathbf{X}_{Ext-1D}$). At that stage a data fusion layer that fuses the $\mathbf{X}_{in}$, $\mathbf{X}_{Ext-2D}$, and $\mathbf{X}_{Ext-1D}$. The output is $\mathbf{X}_{in-Ext}$ which is fed to *tanh* function to be mapped to $[-1, 1]$ range. This helps in faster convergence in the backpropagation learning compared to a standard logistic function [152].

8.3.3.1 Network Throughput input data

The network throughput data for both uplink and downlink are fed into the first three branches (shown in blue in Figure 8.1).

Convolution Design. Since a city usually has a very large size with many regions, and intuitively the network demand may be affected by nearby as well as distant regions, convolutional neural network can handle this effectively as it captures the spatial structure through convolutions. In order to capture the dependency between regions, there is a need to design many convolutional layers. Subsampling techniques have been introduced to preserve distant dependencies and avoid the loss of resolution especially in video sequence generating tasks [153]. Unlike with the common approach to CNN, we do not use subsampling but instead rely only on convolutions [154]. Support for such an approach can be found in [115], where the authors were trying to capture the spatial dependencies at a citywide scale similar to our problem here. They concluded that one convolution naturally captures spatial near dependencies, and a stack of convolutions afterwards can further capture the spatial distant citywide dependencies. The closeness, periodicity, and trend components adapt 2-channel image-like matrices according to the time interval as follows, $[\mathbf{X}_{t-l_c}, ..., \mathbf{X}_{t-1}]$, $[\mathbf{X}_{t-l_p.p}, ..., \mathbf{X}_{t-p}]$, and $[\mathbf{X}_{t-l_r.r}..., \mathbf{X}_{t-r}]$ respectively. l_c, l_p and l_r

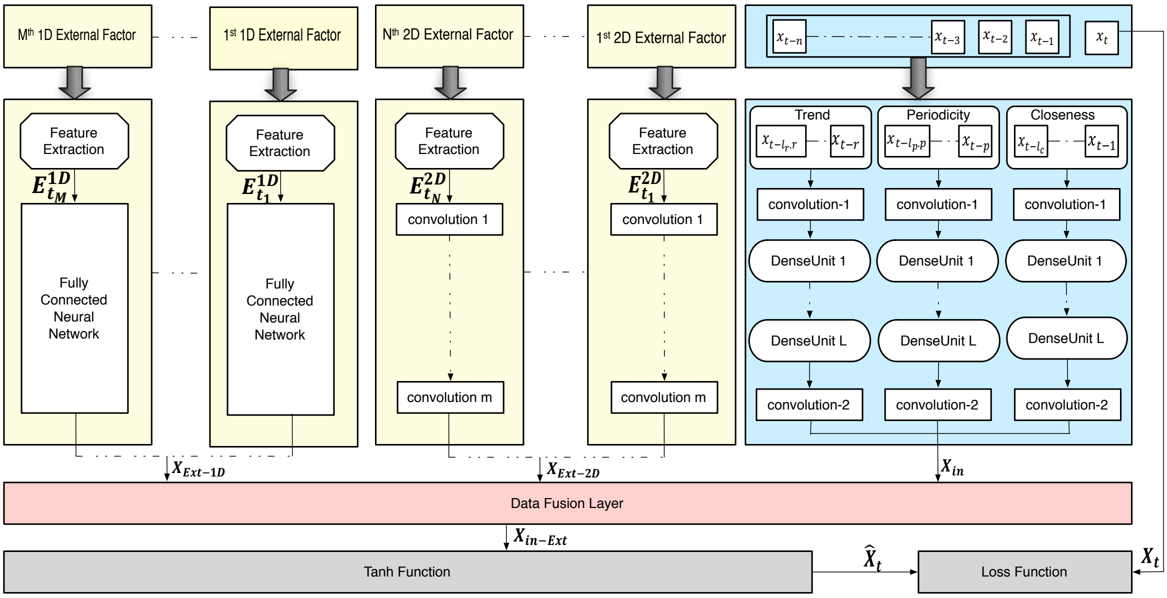


Figure 8.1: ST-DenNetFus Architecture.

represent the length of the dependent sequence for the closeness, period and trend while c, p and r depicts their span respectively. In our detailed implementation, the p captures one day to capture daily periodicity while the r is equal to one week that reveals the weekly trend of the network demand.

Each of these inputs is concatenated across the first axis (time interval) as tensors, $\mathbf{X}_c^{(0)}$, $\mathbf{X}_p^{(0)}$, and $\mathbf{X}_r^{(0)}$ and then followed by a convolution (convolution-1 in Figure 8.1) for each branch as follows:

$$\mathbf{X}_c^{(1)} = f(\mathbf{W}_c^{(1)} * \mathbf{X}_c^{(0)} + \mathbf{b}_c^{(1)}) \tag{8.4}$$

$$\mathbf{X}_p^{(1)} = f(\mathbf{W}_p^{(1)} * \mathbf{X}_p^{(0)} + \mathbf{b}_p^{(1)}) \tag{8.5}$$

$$\mathbf{X}_r^{(1)} = f(\mathbf{W}_r^{(1)} * \mathbf{X}_r^{(0)} + \mathbf{b}_r^{(1)}) \tag{8.6}$$

where $*$ denotes the convolution operation, $f(.)$ is an activation rectifier function [137], and the $(\mathbf{W}_c^{(1)}, \mathbf{b}_c^{(1)})$, $(\mathbf{W}_p^{(1)}, \mathbf{b}_p^{(1)})$, and $(\mathbf{W}_r^{(1)}, \mathbf{b}_r^{(1)})$ are the learnable parameters for this first layer for the three branches.

Since our objective to have the final output size as same as the size of the input (size of the grid map), a specific type of convolution called "same convolution" is employed which allows the filter to go outside of the border of the input padding each outside the border with a zero.

Dense blocks Design. Since in our case, there is a need to capture large citywide dependencies for increasing the accuracy in predicting network demand, a very deep network will be required. This will place both computational power and complexity burden on its implementation. To address this issue, DenseNet has been employed with some modifications that exploits the potential of the network through *feature reuse*, yielding condensed models that are easily trained and highly parameter efficient [142]. In our proposed ST-DenNetFus architecture, each of the outputs from the first convolution layer (shown as convolution-1 in Figure 8.1), $\mathbf{X}_c^{(1)}$, $\mathbf{X}_p^{(1)}$ and $\mathbf{X}_r^{(1)}$ is passed through L layers, each of which implements a non-linear transformation $\mathbf{H}_l(.)$, where l depicts the layer. In our implementation, $\mathbf{H}_l(.)$ is defined as a composite function of two consecutive operations of Rectified Linear Unit (ReLU) followed by a 3×3 convolution. On top of the L^{th} dense block, a convolutional layer is appended (shown as convolution-2 in Figure 8.1). The final outputs of each of these branches after convolution-2 are $\mathbf{X}_c^{(L+2)}$, $\mathbf{X}_p^{(L+2)}$ and $\mathbf{X}_r^{(L+2)}$.

8.3.3.2 External Factors & Fusion

Network demand can be affected by many complex external factors. One of the main contributions of this research is to show how some of the extracted patterns as discussed in the previous chapters can be of an impact on one of the most important challenges in the telecommunications domain, Network Demand Prediction. Intuitively, the thought was that there might be a relation between mobile data utilization and the functionality of the regions. Thinking about a business district, then intuitively one could expect that most companies will be empowered by a WiFi network and hence people once they arrive to their work will probably rely on the WiFi network more than the cellular network. In contrast, in a shopping district, the cellular network might be expected to be used more than the WiFi network as usually people are walking in streets or in shops in which WiFi is not universally or freely available. Another external factor that intuitively could impact the network demand is the crowd mobility patterns as it is obvious that the more crowded an area is, the higher network demand. In addition and as shown before in the literature [127], the day of the week is of an impact to the network demand variation. The simple example is that people typically rely on their cellular networks in a different pattern on the weekends compared to the weekdays.

To predict the network demand at time t, the prior three external factors: Temporal Functional Regions patterns, day of the week and the crowd mobility patterns can be already obtained. However, the challenge in embedding these external factors into a model is that they vary in their dimensionality. In other words, the Temporal Functional Regions and the crowd mobility patterns are both 2-dimensional features that vary across time however, the day of the week is 1-dimensional feature that varies across the time. For addressing this challenge, various branches have been introduced in the ST-DenNetFus architecture to fuse the external features according to their dimensionality as shown in the yellow branches of Figure 8.1. Let $[E_{t_1}^{1D},...,E_{t_N}^{1D}]$ and $[E_{t_1}^{2D},...,E_{t_M}^{2D}]$ depict the features vectors for the 1-dimensional and 2-dimensional features respectively, M and N indicates the number of the external 1-dimensional and 2-dimensional features respectively. Formally and for the 1-dimensional features, fully-connected layers are stacked and for the

2-dimensional features, convolutional layers with 5×5 filter are stacked for capturing the spatial dependencies of these features employing the "same convolution", for preserving the final output size to be the same as the size of the input.

After the network demand data is input and output $\mathbf{X}_{in}$ is generated, and the other branches for the external data sources $\mathbf{X}_{Ext-2D}$ and $\mathbf{X}_{Ext-1D}$ for the 2-dimensional and 1-dimensional features are produced, then a fusing layer (shown in red in Figure 8.1) is used. The output $\mathbf{X}_{in-Ext}$ is further fed to a *tanh* function to generate $\hat{\mathbf{X}}_t$ which denotes the predicted value at the t^{th} time interval. These operations are summarized with the following equations:

$$\hat{\mathbf{X}}_t = tanh(\mathbf{X}_{in-Ext}) \tag{8.7}$$

$$\mathbf{X}_{in-Ext} = \mathbf{X}_{in} + \mathbf{X}_{Ext-2D} + \mathbf{X}_{Ext-1D} \tag{8.8}$$

The ST-DenNetFus architecture can then be trained to predict $\hat{\mathbf{X}}_t$ from the Network Throughput input data and the external features by minimizing mean squared error between the predicted demand and the true demand matrix:

$$\kappa(\varepsilon) = ||\mathbf{X}_t - \hat{\mathbf{X}}_t||^2 \tag{8.9}$$

where ε represents all the learnable parameters in the whole ST-DenNetFu architecture.

A summary of the procedures for training the proposed ST-DenNetFu architecture is shown in Algorithm 6.

8.4 Experiments setup

In this section, an overview on the network (telco) dataset used is given, the baselines to be used for comparison presented and finally the evaluation metrics that will be used in the Results section will be discussed.

8.4.1 Dataset

The dataset used in this chapter is an application and network usage dataset gathered from Truconnect LLC[2], a mobile service provider based in US. The raw data contains more than 200 billion records of mobile sessions that spans across 6 months from July 2016 to December 2016 in NYC. All of the mobile sessions are geo-tagged with longitude and latitude. Mobile sessions can be any type of application usage on the phone that uses mobile network. These sessions might include session types such as Youtube video views, application downloads and updates, and web browsing sessions.

Each sample in the dataset is created due to one of the following: (a) every hour, (b) every change of a pixel (lat, long of 4 digits with a resolution of 10×10), (c) every

[2] https://www.truconnect.com

```
Input: Historical Observations: $\{\mathbf{X}_0, ..., \mathbf{X}_{s-1}\}$;
External 1D Features ($\mathbf{E}_t^{1D}$): $\{\mathbf{E}_{t_1}^{1D}, ..., \mathbf{E}_{t_M}^{1D}\}$;
External 2D Features ($\mathbf{E}_t^{2D}$): $\{\mathbf{E}_{t_1}^{2D}, ..., \mathbf{E}_{t_N}^{2D}\}$;
lengths of closeness, period, trend sequences: $l_c, l_p, l_r$
span of period, trend: $p, r$
Output: STDenNetFus model $\mathbf{M}$
// Construct training dataset
$\mathbf{TR} \leftarrow \emptyset$
for $1 \leq t \leq s-1$ do
    $TD_c \leftarrow [\mathbf{X}_{t-l_c}, ..., \mathbf{X}_{t-1}]$
    $TD_p \leftarrow [\mathbf{X}_{t-l_p.p}, ..., \mathbf{X}_{t-p}]$
    $TD_r \leftarrow [\mathbf{X}_{t-l_r.r} ..., \mathbf{X}_{t-r}]$
    create a sample of ($\{TD_c, TD_p, TD_r, \mathbf{E}_t^{1D}, \mathbf{E}_t^{2D}\}$, $\mathbf{X}_t$) in training data $TD$
end
// Train STDenNetFus model $\mathbf{M}$
$\mathbf{X}_{in}^c \leftarrow$ DenseNet($TD_c$), $\mathbf{X}_{in}^p \leftarrow$ DenseNet($TD_p$), $\mathbf{X}_{in}^r \leftarrow$ DenseNet($TD_r$)
$\mathbf{X}_{in} \leftarrow \mathbf{X}_{in}^c + \mathbf{X}_{in}^p + \mathbf{X}_{in}^r$
$\mathbf{X}_{Ext-1D}^1$,...,$\mathbf{X}_{Ext-1D}^M \leftarrow$ FCN($\mathbf{E}_{t_1}^{1D}$),...,FCN($\mathbf{E}_{t_M}^{1D}$)
$\mathbf{X}_{Ext-1D} \leftarrow \mathbf{X}_{Ext-1D}^1 + ... + \mathbf{X}_{Ext-1D}^M$
$\mathbf{X}_{Ext-2D}^1$,...,$\mathbf{X}_{Ext-2D}^N \leftarrow$ CNN($\mathbf{E}_{t_1}^{2D}$),...,CNN($\mathbf{E}_{t_M}^{2D}$)
$\mathbf{X}_{Ext-2D} \leftarrow \mathbf{X}_{Ext-2D}^1 + ... + \mathbf{X}_{Ext-2D}^N$
$\mathbf{X}_{in-Ext} \leftarrow \mathbf{X}_{in} + \mathbf{X}_{Ext-1D} + \mathbf{X}_{Ext-2D}$
$\hat{\mathbf{X}}_t \leftarrow \tanh(\mathbf{X}_{in-Ext})$
// Optimize parameters $\varepsilon$ for $\kappa$, see equation 8.9
Initialize the parameters $\varepsilon$
for Stopping condition NOT met do
    Select a batch of instances $TD_b$ from $TD$
    Find parameters $\varepsilon$ that minimizes $\kappa$
end
Conclude the learned STDenNetFus model $\mathbf{M}$
```

Algorithm 6: Training the STDenNetFus architecture.

application used within this pixel and hour results in a new record in the dataset. On average per mobile device, there are between $1000 - 1200$ records per day. The main features that are filtered and used within this dataset are as follows:

- **Ueid:** This feature represents a mobile device unique identifier.
- **Latitude:** This value represents the latitude of the bottom-right corner of the pixel with resolution of 0.0001 degree.
- **Longitude:** This value represents the longitude of the bottom-right corner of the pixel with resolution of 0.0001 degree.
- **MaxRxThrpt:** This value represents the maximum downlink throughput observed on the network interface of the mobile device in the current pixel (in bps).
- **MaxTxThrpt:** This value represents the maximum uplink throughput observed on the network interface of the mobile device in the current pixel (in bps).

The network demand data has strong periodical patterns for both MaxRxThrpt and MaxTxThrpt. The weekly recurrence of the network demand values are shown in Figure 8.2. From this figure, it could be observed that the network demand has two strong

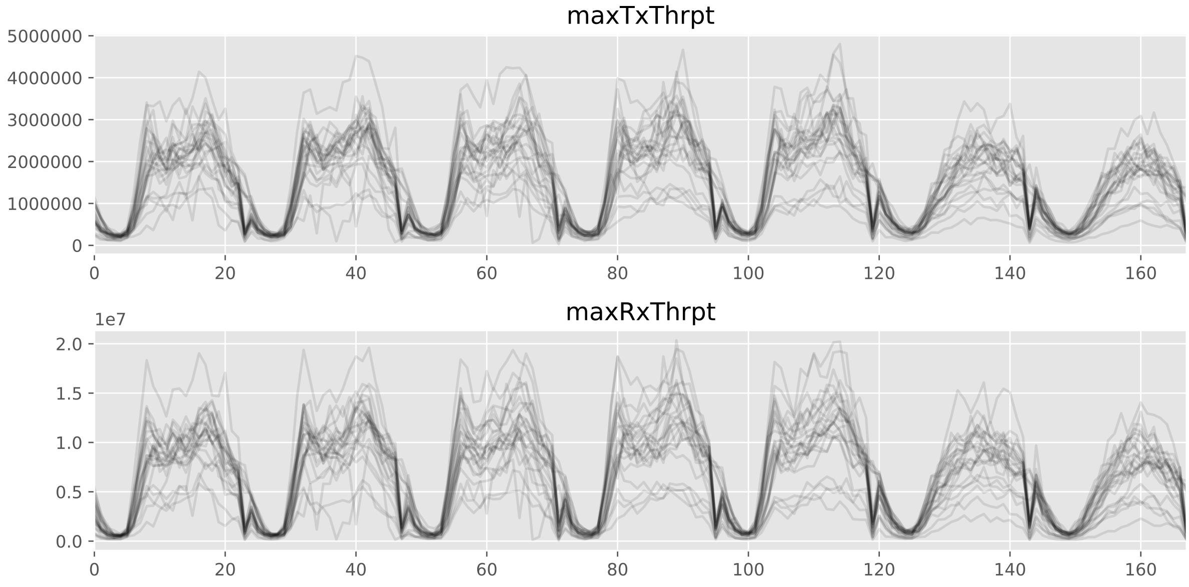


Figure 8.2: Network Demand Data periodicity.

patterns: a) Daily, b) Weekly. In the daily recurrent, a maximum demand during peak hours is observed during the day while the demand drops significantly after midnight.

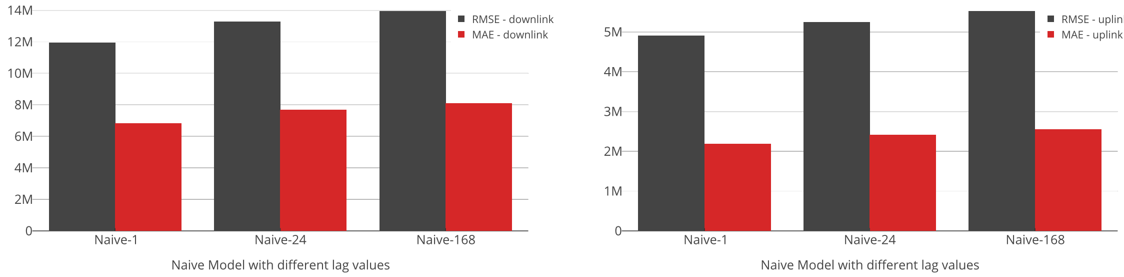
The network data available spans for the last 6 months of 2016 while the LBSNs data used in the previous chapters spans across 2 years in 2013 and 2014 (refer to chapter 4). So for achieving the research objective of this chapter, LBSNs data is gathered across the same 6 months of the network data using the Twitter streaming APIs[3]. This data is used for extracting both the Temporal Functional Regions and Recurrent Crowd Mobility Patterns by applying the approaches described in chapter 6 and 7 respectively.

8.4.2 Baselines

The proposed ST-DenNetFus approach has been compared with the following 4 baselines:

- **Naive:** A naive model [155] works by simply setting the forecast at time t to be the value of the observation at time $t-l$ where l is the lag value. Several lag values are tested considering l equals to 1, 24, and 168 that corresponds to hourly, daily or weekly which we refer to as Naive-1, Naive-24, and Naive-168 respectively. For instance, in case of daily, the network demand at time t on Monday is considered same as time t on Sunday. In case of weekly, the network demand at time t on Monday is considered same as time t on previous Monday and for hourly, the network demand at time t is considered same as time $t-1$. These comparisons are shown in Figure 8.3. From these comparisons, Naive-1 shows the best accuracy and following that Naive-24 and Naive-168 respectively.

[3] https://dev.twitter.com/streaming/overview



(a) Lag value impact on throughput downlink.

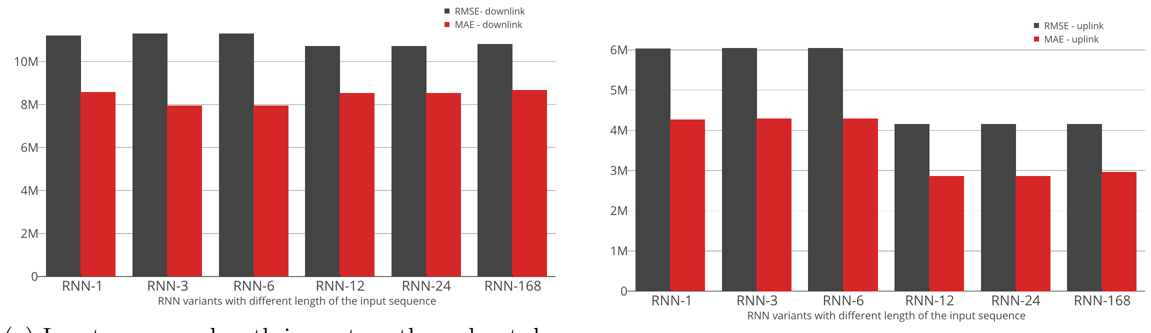
(b) Lag value impact on throughput uplink.

Figure 8.3: Lag value impact for the Naive Model. The smaller the better.

- **ARIMA:** An ARIMA model [156] is a well-known model for analyzing and forecasting time series data. ARIMA is an acronym that stands for AutoRegressive Integrated Moving Average and is considered a generalization of the simpler AutoRegressive Moving Average and adds the notion of integration. A nonseasonal ARIMA model is classified as titled as ARIMA(p, d, q) where p is the number of autoregressive terms, d is the number of nonseasonal differences needed for stationarity, and q is the number of lagged forecast errors in the prediction equation. In our trained model, p, d, q are set to 1.

- **RNN:** Recurrent Neural Networks or RNNs [157] are a special type of neural network designed for sequence problems. Given a standard feedforward Multilayer Perceptron network, a recurrent neural network can be thought of as the addition of loops to the architecture. For example, in a given layer, each neuron may pass its signal latterly (sideways) in addition to forward to the next layer. The output of the network may feedback as an input to the network with the next input vector. And so on. In our experiments, the length of the input sequence is fixed to one of the $\{1, 3, 6, 12, 24, 48, 168\}$. Figure 8.4 summarizes the comparison between these variants and concludes that the best accurate model is RNN-12.

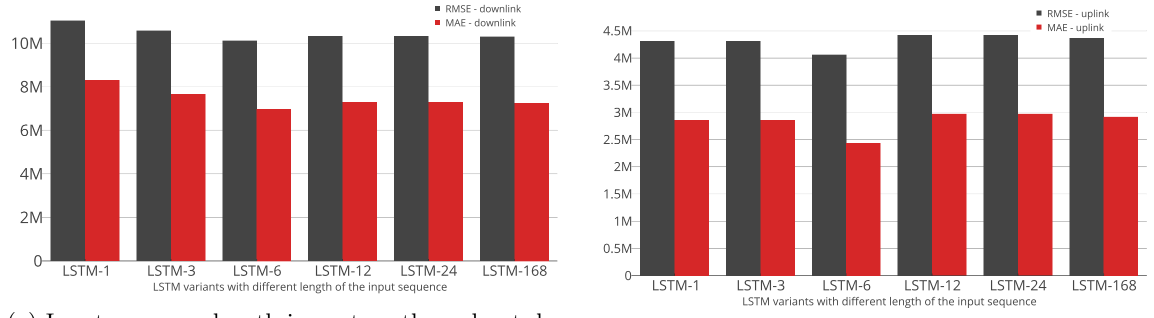
- **LSTM:** The Long Short-Term Memory or LSTM [158] network is a recurrent neural network that is trained using Back propagation Through Time and overcomes the vanishing gradient problem. As such it can be used to create large (stacked) recurrent networks, that in turn can be used to address difficult sequence problems in machine learning and achieve state-of-the-art results [159]. Instead of neurons, LSTM networks have memory blocks that are connected into layers. The experiments are conducted on 6 LSTM variants following the same settings of RNN, including, LSTM-1, LSTM-3, LSTM-6, LSTM-12, LSTM-24, LSTM-48, LSTM-168. Figure 8.5 summarizes the comparison between these variants, it can be concluded from this figure that LSTM-6 is the most accurate model.



(a) Input sequence length impact on throughput downlink.

(b) Lag value impact on throughput uplink.

Figure 8.4: Input sequence length impact for the RNN Model. The smaller the better.



(a) Input sequence length impact on throughput downlink.

(b) Input sequence length impact on throughput uplink.

Figure 8.5: The input sequence length impact for the LSTM Model. The smaller the better.

Going forward in this chapter, the best performing baseline models, Naive-1, RNN-12, and LSTM-6 are referred to as simply Naive, RNN, and LSTM respectively.

8.4.3 Evaluation Metric

ST-DenNetFus is evaluated by the Mean Absolute Error (MAE) and Root Mean Squared Error (RMSE) as they are two of the most common metrics used to measure the accuracy of continuous variables. Since RMSE has the benefit of penalizing large errors more so it can be more appropriate in some cases, for example, if being off by 10 is more than twice as bad as being off by 5. But if being off by 10 is just twice as bad as being off by 5, then MAE is more appropriate. Both metrics are used for comprehensively evaluating the ST-DenNetFus approach, the following defines both metrics:

Mean absolute error (MAE): Is a quantity used to measure how close forecasts or predictions are to the eventual outcomes. The mean absolute error is given by:

$$MAE = \frac{1}{n}\sum_{j=1}^{n}||y_i - \hat{y}_i|| \tag{8.10}$$

Root mean squared error (RMSE): Is a quadratic scoring rule that also measures the average magnitude of the error. It's the square root of the average of squared differences between prediction and actual observation. The root mean squared error is given by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{8.11}$$

In both Equations (8.10) and (8.11), y_i and $\hat{y}_i$ depict the actual values and the corresponding predicted values respectively; n is the number of the available samples in the dataset.

8.5 Results

8.5.1 Preprocessing

Manhattan region is divided into a grid (refer to Definition 7) for the pre-processing of the data. For fitting a grid map to Manhattan region, it is needed to rotate the grid map by an angle. For doing so, the longitude and latitude pairs of the grid map is processed to find lon' and lat' with a rotation matrix as follows:

$$\mathbf{R} = \begin{bmatrix} 0.9153 & 0.4027 \\ 0.4027 & 0.9153 \end{bmatrix} \tag{8.12}$$

$$lon_{center} = -73.9576 \tag{8.13}$$

$$lat_{center} = 40.7878 \tag{8.14}$$

$$< lon', lat' > = (< lon, lat > - < lon_{center}, lat_{center} >) * R \tag{8.15}$$

After doing such transformation, the rotated grid appeared as shown in Figure 8.6. For each grid, the median maximum throughput for users is derived for both the uplink and downlink throughputs. The reason for using the median throughput is because telco operators are more concerned with the experience of an average user network demand when planning resources [160].

Since the $tanh$ function is used in the output of the ST-DenNetFus to map the output between $[-1, 1]$, the data is preprocessed using the min-max normalization to scale the network demand data between between -1 and 1. When evaluating the performance, this transformation is inversed back later on to the original scale to be compared with the ground-truth. For the external factors, one-hot encoding [161] is used to transform the day of the week, functional regions and the crowd mobility patterns into binary vectors and then they are fed normally to their corresponding branches in the ST-DenNetFus as described in the previous section.

8.5.2 Hyperparameters

The learnable parameters of the ST-DenNetFus are initialized using a uniform distribution in keras [162]. The convolutions of convolution-1 and convolution-2 use 24 and 2 filters of

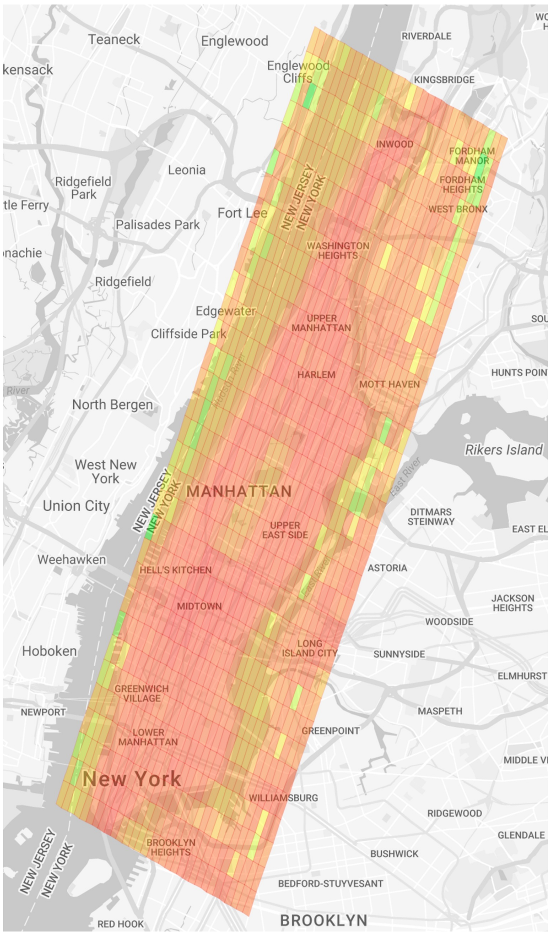
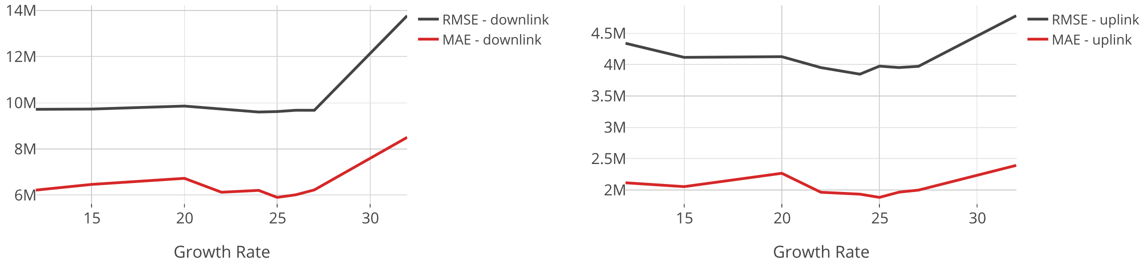


Figure 8.6: Manhattan rotated grid map.

size 3×3 respectively. Convolution-2 uses 2 filters to match the desired number of outputs needed for the downlink and uplink throughput. Adam [163] is used for optimization, and the batch size is set to 15 for fitting the memory of the GPU used in the experiments. The number of dense blocks is set to 5. For p and r, they are empirically set to capture one-day and one-week respectively where l_c, l_p and $l_r \in \{1, 2, 3, 4\}$. The dataset is partitioned as 80% for the training dataset and the remaining 20% for the test dataset (this ration in partitioning is one of the common practices in machine learning [164]) which is used for evaluating the performance of the final selected model. From the training dataset, 90% is selected for training each model and the remaining 10% for the validation dataset which is used for choosing the best model as well as to early-stop the training algorithm if there is no improvement found after 5 consecutive epochs.

Impact of growth rate: If each function $\mathbf{H}$ produced k feature-maps as output, then the l_{th} layer will have $k \times (l-1) + k_0$ input feature-maps, where k_0 is the number of channels in the input matrix. To prevent the network from growing too wide and to improve the parameters efficiency, k is limited to a bounded integer. This hyperparamater is referred to as *growth rate* [142]. Figure 8.7a and Figure 8.7b show the impact of increasing the growth rate on the prediction's accuracy for both the throughput downlink and uplink



(a) Growth rate impact on throughput downlink.

(b) Growth rate impact on throughput uplink.

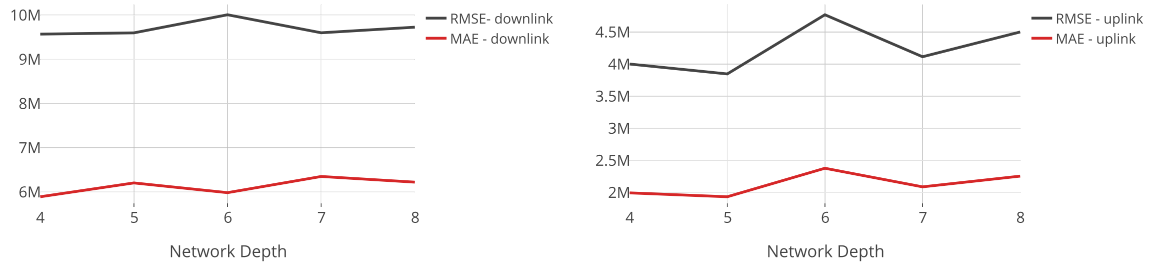
Figure 8.7: Growth rate impact on the Network Demand Prediction. The smaller the better.

respectively. From these figures, it is obvious that the accuracy improves by increasing the growth rate until reaching certain point in which widening further the network starts to have a counter impact on the accuracy. Hence it was concluded that the optimum *growth rate* is 24 for both downlink and uplink throughputs.

Impact of network depth: Figure 8.8a and Figure 8.8b show the impact of the network depth on the predictions' accuracy for the downlink and uplink throughputs respectively. It was concluded from these figures that a network depth of 5 has the optimum results which shows it can sufficiently capture with this depth the spatial dependence as well as the distant one. However, when the network is very deep, training becomes more difficult.

Impact of parallel branches for external features: In ST-DenNetFus, each of the external features are input into a separate branch unlike the traditional approach. The traditional approach of fusing external features of same dimensionality merges these features first and then fuses them into one branch. However, in our proposed approach, a separate branch for each of the external features is used and then merge their outputs in a later stage after the feedforward execution of each of the branches (shown in yellow in Figure 8.1). In our case, although both the Temporal Functional Regions and crowd mobility patterns are of same dimensionality where both are 2-dimensional matrices that change across time (1-hour time-interval), they are each input on a separate branch and then fused later. The impact of feeding the external features in this way is demonstrated in Figure 8.9a and Figure 8.9b for the downlink and uplink throughput prediction respectively. As it can be seen and compared to the traditional approach, our approach performs 10% RMSE and 8% MAE better for the downlink throughput prediction and 8% RMSE and 7% MAE better for the uplink throughput prediction.

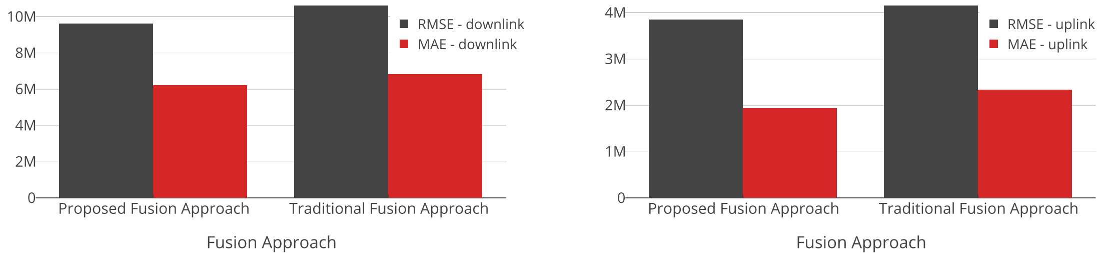
Impact of temporal closeness, period and trend: In order to determine the optimum length of closeness, period and trend for the network demand dataset. The length of period and trend are set to 1 and then the length of closeness is varied from 0 to 5 where $l_c = 0$ indicates that the closeness component/branch is not employed. Figure 8.10a and Figure 8.10b summarize such analysis for both the downlink and uplink throughputs prediction respectively. From these figures, it can be concluded that l_c equals to 3 has



(a) Network depth impact on throughput downlink.

(b) Network depth impact on throughput uplink.

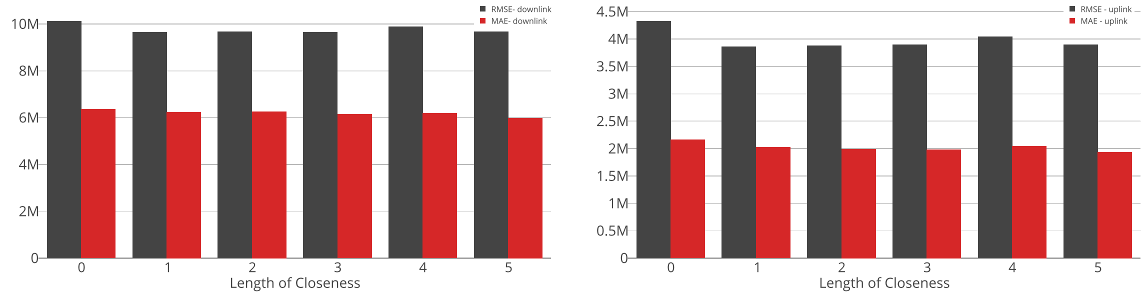
Figure 8.8: Network depth impact on the Network Demand Prediction. The smaller the better.



(a) External factors data fusion method impact on throughput downlink.

(b) External factors data fusion method impact on throughput uplink.

Figure 8.9: External factors data fusion method impact on the Network Demand Prediction. The smaller the better.

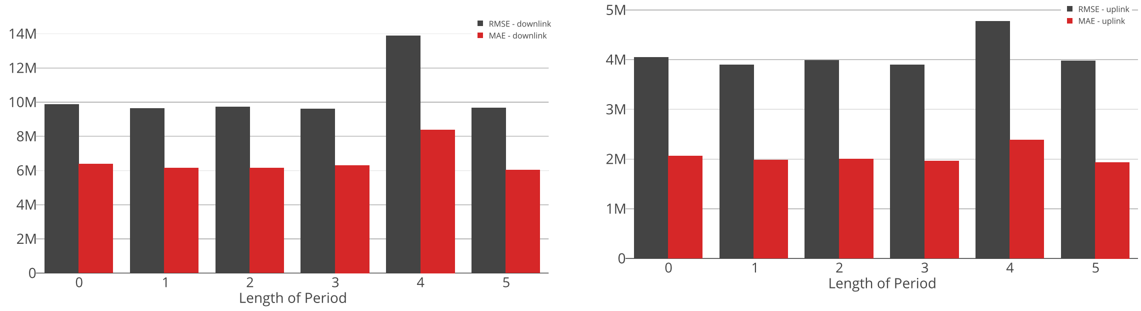


(a) Temporal closeness impact on throughput downlink.

(b) Temporal closeness impact on throughput uplink.

Figure 8.10: Temporal closeness impact on the Network Demand Prediction. The smaller the better.

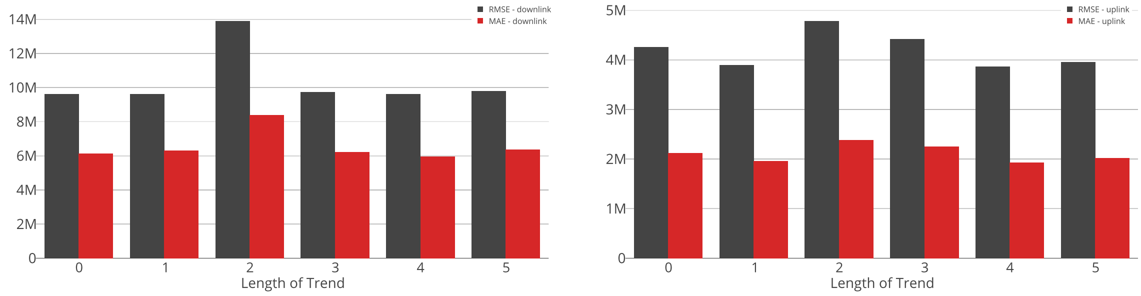
the lowest RMSE and MAE shown on the y axis for both types of throughputs and $l_c = 0$ has the highest RMSE and MAE. Then, l_c is set to 3 and l_r is set to 1 and then l_p is varied from 0 to 5. From Figure 8.11a and Figure 8.11b, it is concluded that the best performance is when l_p equals to 3. Finally, l_c and l_p are set to 3 and l_r is varied in which it is concluded from Figure 8.12a and Figure 8.12b that its best value is at 4. Based on this analysis, it is concluded that the best configuration for the $\{l_c, l_p, l_r\}$ is $\{3, 3, 4\}$.



(a) Temporal period impact on throughput downlink.

(b) Temporal period impact impact on throughput uplink.

Figure 8.11: Temporal period impact on the Network Demand Prediction. The smaller the better.



(a) Temporal trend impact on throughput downlink.

(b) Temporal trend impact on throughput uplink.

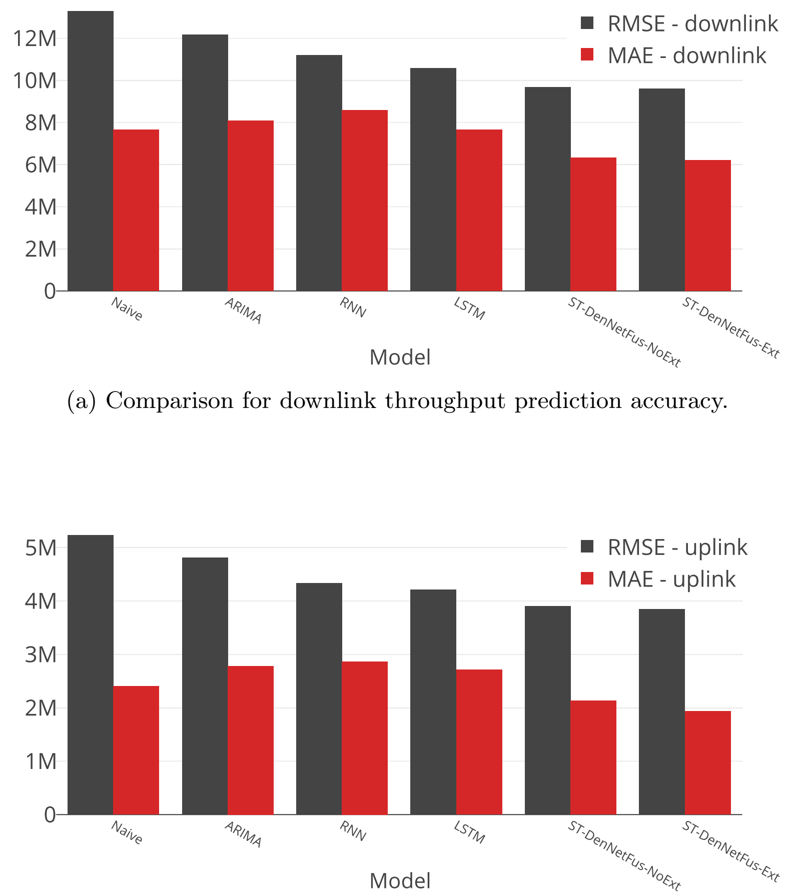
Figure 8.12: Temporal trend impact on the Network Demand Prediction. The smaller the better.

8.5.3 Discussion

Table 8.1 shows the ST-DenNetFus Network Demand Prediction accuracy comparisons with the baselines for both the throughput downlink and uplink. As shown, the proposed ST-DenNetFus consistently and significantly outperforms all baselines. Specifically, the results for the downlink throughput prediction demonstrate that ST-DenNetFus (with 5 dense-blocks) is relatively 30% RMSE and 20% MAE better than the Naive model, 20% RMSE and 23% MAE better than ARIMA, 15% RMSE and 30% MAE better than RNN and 10% RMSE and 20% MAE better than LSTM. For the uplink throughput prediction, ST-DenNetFus is 27% RMSE and 20% MAE better than the Naive model, 20% RMSE and 30% MAE better than ARIMA, 12% RMSE and 33% MAE better than RNN, and 10% RMSE and 30% MAE better than LSTM. ST-DenNetFus-NoExt is our proposed version of ST-DenNetFus-Ext that does not consider external factors (e.g. Temporal Functional Regions). It can be seen that ST-DenNetFus-NoExt is worse than the ST-DenNetFus-Ext indicating that external factors and patterns fused are always beneficial. Intuitively, the models in RMSE can be ranked as illustrated in Figure 8.13. It is worth emphasizing that although the improvement in the predictive accuracy with fusing the extracted urban patterns (ST-DenNetFus-Ext) compared to without fusing the urban patterns (ST-DenNetFus-NoExt) is better but the improvement is not obviously significant.

Table 8.1: Prediction accuracy comparisons with baselines

	Evaluation Metric (Downlink Throughput)		Evaluation Metric (Uplink Throughput)	
Model	**RMSE**	**MAE**	**RMSE**	**MAE**
Naive	13278936.747	7667966.397	5237542.197	2406522.277
ARIMA	12177307.197	8073921.767	4816366.977	2783808.635
RNN	11199525.956	8576942.055	4335734.302	2865784.639
LSTM	10580656.522	7660093.113	4216037.533	2713482.051
ST-DenNetFus-NoExt	9675762.836	6315907.039	3907936.380	2131071.282
ST-DenNetFus-Ext	**9600259.526**	**6206750.047**	**3847875.555**	**1933871.466**



(a) Comparison for downlink throughput prediction accuracy.

(b) Comparison for uplink throughput prediction accuracy.

Figure 8.13: Model ranking for Network Demand Prediction. The smaller the better.

However, this small improvement is considered quite a good achievement due to the fact that: (a) Fusing external data sources usually in most of the prior state-of-the-art is quite challenging task and in lots of cases it results in ingesting noise to the models leading to worse accuracy [165] [166] [167]. (b) In most of the machine learning tasks, after trying several techniques and reaching the best possible accuracy, it is quite challenging for any further small improvement and at that level any kind of small improvement is considered significant [168] [169].

8.6 Summary

In this chapter, a new deep learning based approach called ST-DenNetFus is proposed for forecasting the network demand (throughput uplink and downlink) in each and every region of a city. For the first time, it has been shown that fusing some external patterns such as Temporal Functional Regions and crowd mobility patterns improves the accuracy of the forecasting due to their intuitive correlation with the network demand variation. Compared to other 4 baselines, the proposed approach outperforms, confirming that the proposed approach is better and more applicable to the Network Demand Prediction problem. The introduced ST-DenNetFus is capable of learning the spatial and temporal dependencies. In addition, it employs various branches for fusing external data sources of various dimensionalities. In this chapter, it is concluded that besides the direct benefit of the extracted patterns introduced in the previous chapters, theses patterns could have a potential indirect impact on some of the challenges in other domains such as the Network Demand Prediction problem in the telecommunications domain. In addition, the author of this thesis argue that the introduced ST-DenNetFus architecture could be leveraged for solving other spatio-temporal prediction problems that requires fusing external data sources such as energy demand forecasting and others.

Having presented in this chapter the fourth and last introduced contribution in this thesis showing how some of the urban patterns extracted in the previous chapters could bring benefit to other domain problems (such as the Network Demand Prediction problem in the telecommunications domain), the next chapter concludes the thesis, summarizing the contributions and highlighting future work.

Chapter 9

Reflections and outlook

Chapter overview: *This chapter concludes the thesis by first summarizing the core contributions presented in the core chapters including the Socio-demographic Regional Patterns (chapter 5), Temporal Functional Regions Patterns (chapter 6), Recurrent Crowd Mobility Patterns (chapter 7), and the proposed approach for the Network Demand Prediction problem utilizing some of the extracted urban patterns (chapter 8). Second, a summary of the challenges encountered in this research are discussed. Future directions are then presented for each of the core contributions highlighting potential areas for extending the work presented in this thesis. Then, I shed the light on a new domain that can emerge from this thesis that focuses on reasoning from challenges that our cities face today from correlating various extracted urban patterns. Finally, a brief outlook is given on the whole work.*

9.1 Summary of contributions

With the rise of online social networks and services that provided another useful source where location data and human activity or relationships are being described, a classic question this thesis attempts to address is how far new urban patterns can be extracted harnessing the power of this type of data. The recent advancements and technologies in machine learning is a key enabler for extracting the potential new urban patterns. Through my journey of reviewing, studying and understanding various machine learning techniques in the literature to see what might be the most suitable techniques for extracting urban patterns, I came up with a recommender approach called TCDC discussed briefly in chapter 3 (whereas its validation and experiments are detailed in Appendix A). TCDC can help machine learning researchers and practitioners to choose the optimum supervised machine learning model for the classification and regression problems. Afterwards and for the aim of studying and extracting interesting urban patterns in cities, LBSNs data is gathered and preprocessed for two full years as described in chapter 4, to the best of our knowledge, this is the longest time duration LBSNs data that has been studied in the literature so far.

The **first core contribution** was presented in chapter 5 and focused on extracting new urban pattern called *Socio-demographic Regional Patterns.* The proposed model based on DBNs was able to extract a unique pattern for each of the boroughs in NYC utilizing the weekly footprints comprising of individuals' activities captured from the LBSNs dataset. For an unseen weekly footprints capturing individuals' activities, the best trained model was able to classify the borough that such weekly footprints belongs to with nearly 70% accuracy. Since in general, the DBNs trained models suffer from interpretability, I went further and applied a well-known topic based model called LDA for better understanding what could be the underneath unique pattern extracted for each of the boroughs. The outcome results from LDA validated that there is an observable different pattern for each of the boroughs and it proved that detecting such unique patterns is a complex task.

Since, the first core contribution showed an example of a new urban pattern that can be extracted using LBSNs, the second core contribution focused on extracting new type of functional regions that changes across space and time compared to the *static functional regions* that was the focus in the previous state-of-the-art. This leads to the **second core contribution** introduced in chapter 6 which focused on extracting what I refered to as *Temporal Functional Regions patterns* using clustering based techniques. In this context, I studied the optimum time interval for extracting such variation in the functionalities of the regions in which four hours were found to be the most suitable time interval for capturing the variation in the regions' functionalities. Although the experiments show objectively that the quality of clustering is better for the *Temporal Functional Regions* compared to the *static functional regions*, I went further and validated subjectively for some selected regions in Manhattan that the extracted Temporal Functional Regions follow the intuitive understanding of the features of these regions.

Subsequently, in chapter 7, I went further to understand how crowd flows to the extracted Temporal Functional Regions from the previous chapter. This leads to the **third core contribution** presented in this thesis, a new approach for *Recognizing Recurrent Crowd Mobility patterns in Cities called RCMC* based on a combination of KDE/NMF algorithms. I have adapted the KDE with gaussian kernel for extracting the crowded areas which further acted as an input to the NMF-based algorithm for extracting the Recurrent Crowd Mobility Patterns. For providing further insights on the level of crowdedness recognized, the output of the NMF-based algorithm is expanded for recognizing three levels of crowd intensity. The proposed approach succeeds to extract three different Recurrent Crowd Mobility Patterns, first for weekdays, second for Saturdays and third for Sundays. However, when adapting two topics, it was possible to extract two different unique patterns, one for the weekends and the other for the weekdays. The proposed approach was tested objectively and subjectively. Due to the difficulty of assessing the accuracy of the extracted Recurrent Crowd Mobility Patterns objectively due to the lack of ground truth, I went further and introduced an evaluation metric called *topic stability* for assessing the ability for a topic-based algorithm to extract similar patterns using an unseen dataset which is a key requirement for the proposed approach to detect "recurrent" patterns. It was shown that the proposed approach outperforms in most cases two baselines based on LDA and Tf-idf algorithms respectively. Using the same time interval (4 hours) as the one used for recognizing the Temporal Functional Regions, some key areas in Manhattan are selected and the mobility patterns extracted are correlated with the Temporal Functional Regions, this correlation helped in deriving insights into the motivation behind crowd mobility.

For illustrating an example of the impact of such extracted urban patterns in solving another domain challenges, I thought to fuse these patterns as an external data sources for exploring the possibility of improving the accuracy for the **Network Demand Prediction** problem. The reason for choosing the Network Demand Prediction problem is three-fold: First, intuitively I thought that the Temporal Functional Regions as well as the crowd patterns across space and time could impact the network usage patterns. Second, the Network Demand Prediction problem on itself is quite complicated problem for the telco operators due to the various external factors that could impact the network usage patterns [15]. Third, being able to predict the network demand reliably and accurately can allow the operators to allocated network resources adaptively according to the predicted/expected demand from the model. This subsequently can reduce significantly the costs accompanied with the current over-provisioning model for the network resources for guaranteeing SLAs (Service Level Agreements). This concludes the **fourth core contribution** in chapter 8 that introduced a deep learning convolutional neural networks based architecture for predicting the network demand. Interestingly and for the first time, I proved that the proposed approach improves the accuracy for the network demand when fusing some of the extracted urban patterns introduced in this thesis. In addition and for showing the effectiveness of the proposed approach, it outperformed other baselines that utilizes popular recent time-series forecasting techniques

9.2 Summary of research challenges

Throughout the work presented in this thesis, the following challenges have been encountered:

Challenge-1: What are the main principles in which the author of this thesis relied on for extracting new urban patterns? Intuitively and for addressing such broad challenge, first, a shift of focus in the research carried out was more towards crowd behavioural analysis rather than user-centric behaviour (that had more focus in the state-of-the-art work). This shift in focus may help in extracting novel urban patterns and supporting new classes of applications. This is coupled with working on longer time duration LBSNs data as in most cases and intuitively, more obvious recurrent urban patterns can be extracted from longer time duration data. In addition, studying and utilizing some of the recent advancements in machine learning such as deep learning based models (e.g., Deep-Belief-Networks) helped in extracting new urban patterns.

Challenge-2: Using LBSNs data, how can the sparsity challenge be lessened for extracting new urban patterns? The sparsity challenge is one of the major challenges when analysing LBSNs datasets as it was shown in various studies before [170][170][171]. For overcoming such challenge in our research, the data gathered and analysed in this thesis is of longer time duration (spanning across two full years as presented in chapter 4) compared to other LBSNs datasets used in the previous state-of-the-art research work. The argument that this could yield the discovery of new urban patterns that were not possible before by lessening the sparsity challenge.

Challenge-3: What are the most suitable machine learning models that can be used for extracting urban patterns? In chapter 1 and in particular in section 1.2.1, the urban pattern is defined as a recurrent pattern in an urban environment that can be extracted if the spatio-temporal feature of individuals' mobility data in cities is used. Through the research carried out in this thesis, it is worth emphasizing that the spatio-temporal feature in LBSNs data adds a new dimension for the machine learning models that needs to be considered for extracting urban patterns. At the start of the research work, various machine learning techniques have been studied in depth as introduced in chapter 3. However, throughout the research work, it was found that some models were useful in particular for extracting urban patterns such as Clustering-based techniques, Convolutional Neural Networks, Deep-Belief-Networks, and Topic Models (such as LDA, NMF). The author of this thesis believes that there is still a huge effort needed for researching, developing and adapting new and existing machine learning techniques to suit the spatio-temporal prediction problems in general and for extracting urban patterns in particular.

Challenge-4: How can some of the extracted urban patterns bring indirect benefit to solve a spatio-temporal time series forecasting problem? After it is shown (in chapter 5, 6, and 7) how it is possible to extract new urban patterns leveraging the power of LBSNs data utilizing various machine learning models, another challenge encountered in this thesis is how to show that such extracted urban patterns can be of an impact to other problems. For addressing such challenge, there was a need to come up with an approach that can fuse some of these extracted urban patterns with another domain specific data source (network data). Additional challenge that some of the external data sources are of different dimensionalities which required a new approach for fusing all together in one architecture. For addressing this challenge, various models and techniques are researched which led to the development of the introduced ST-DenNetFus architecture in chapter 8. It is built using multiple branches for fusing various dimensional data sources. Through the developed architecture, it is shown that the various patterns can bring benefit in terms of higher accuracy for the Network Demand Prediction problem. Additionally, the author of this thesis argues that the developed architecture can be of benefit to other spatio-temporal prediction problems that fusing various data sources could boost its prediction accuracy such as crowd flow forecasting or energy demand forecasting across cities.

9.3 Future directions

The Socio-demographic Regional Pattern model introduced in chapter 5 could potentially be extended in the future and applied to various cities across the globe. This means that such model can be leveraged for further understanding the socio-demographic commonalities between different regions across the globe. In addition, this model could be used for understanding how specific cities evolve across time through capturing their weekly citizens' footprints from the LBSNs and identifying the city with the closest pattern to it. This could be useful for exploring evolution of cities that various factors such as social, economical, and political could be of relevance to such extracted patterns.

In chapter 6, the proposed work showed that the concept of Temporal Functional Regions could be very useful for understanding how the functionality of regions could not only change across space but also temporally, across time. This work could be extended in the future to enrich and improve the accuracy of the extracted Temporal Functional Regions by fusing other relevant data sources with LBSNs that intuitively could be correlated to regions' functionalities such as: Points Of Interests (POIs) and mobility patterns (trajectory datasets). This could be seen as extending the work presented by Jing Yuan et al. in [172] but for extracting the new notion of Temporal Functional Regions introduced in this thesis. In addition and for further testing the effectiveness of the proposed approach, it would be very useful to apply it to other cities than NYC that has less dense check-ins and see how realistic the extracted Temporal Functional Regions will be. Furthermore, the proposed approach extracted the regions' functionalities based on the zip-codes boundaries, in the future, it is worth exploring the quality of clustering on other type of boundaries. This might be a tuning parameter that could further improve the extracted functional regions. However, it is worth noting that choosing the type of boundary might vary from application to another.

While our findings in chapter 7 allow for extracting Recurrent Crowd Mobility Patterns in cities, in the future and as a first potential extension to the work, an anomalous real-time recognition system could be built and experimented on the top of the proposed approach. In summary, if there is an area that has been detected in real-time as highly crowded but the output of the proposed RCMC approach recognized that it should be not or less crowded, then an alarm could be raised to the corresponding authorities indicating abnormal event occurring. Another potential extension to the work could be trying to mine from the text (e.g., tweets) in the LBSNs according to the time and location of the abnormal event further insights around the reason behind such event. In the proposed approach, the zip-codes boundaries are used as a way to define areas for extracting mobility patterns. As a future direction, it is worth applying the proposed approach on different types of boundaries (e.g., grid cell boundaries or road networks boundaries) and experiment if there is any improvement in the stability of the patterns extracted. In addition, with the proposed approach, three various crowd mobility patterns were extracted for weekdays, Saturdays and Sundays. In the future, it is worth exploring how topic based models such as NMF and LDA could be extended to extract finer and more various patterns for the weekdays.

Finally, and to show an example of the indirect benefit for some of the prior extracted patterns, it was shown in chapter 8 that when fusing some of the extracted urban patterns as external factors using a new deep learning based architecture titled as ST-DenNetFus, the Network Demand Prediction was more accurate than without fusing such patterns. This demonstrates clearly the value of such patterns in a problem that has never considered fusing such external factors before in the state-of-the-art. In the future, and for solving this particular problem (Network Demand Prediction), it might be worth ingesting more external sources that could be related to the network consumption such as weather conditions. In addition, the data fusion mechanisms still need lots of future research work and is crucial element in the future for making sense from various data sources. For example, extending the proposed solution for fusing various sources using weighted merge method might further improve the accuracy.

In all of the extracted patterns, the research has been carried on a LBSNs dataset spanning over two complete years (2013 and 2014). In the future, it might be interesting to test the limitations of extracting such patterns on a shorter time duration data and whether it is possible to overcome any deterioration in the stability of the extracted patterns by fusing other external data sources. In addition, it will be very interesting to extract the urban patterns introduced in this thesis on other time durations for tracking how cities progress. For instance, extracting the *Temporal Functional Regions Patterns* using a data gathered in 2016 could be of interest to see the evolution in urban planning and regions' functionalities that have been made since then and similarly, for the *Recurrent Crowd Mobility Patterns* and *Socio-demographic Regional Patterns*. In addition, it is argued that the generated patterns and approaches presented in this thesis can be applied on any other LBSNs data subjective to the density of samples per region. Hence, it is argued that someone could adapt the region's definition to generate stable urban patterns using the approaches introduced in this thesis which is another interesting research path to explore.

9.4 Towards the emergence of urban reasoning

Urban computing is an interdisciplinary field where computer sciences meet conventional city-related fields, like transportation, civil engineering, environment, economy, ecology, and sociology in the context of urban spaces. Urban computing aims to tackle these issues by using the data that has been generated in cities (e.g., traffic flow, human mobility, and geographical data). Urban computing connects urban sensing, data management, data analytics, and service providing into a recurrent process for an unobtrusive and continuous improvement of people's lives, city operation systems, and the environment. The term urban computing has been introduced initially in [173] but has been further formalized in depth in [35].

The contributions presented in this thesis is considered a step towards progressing this emerging field. As presented in this thesis, the main focus of this research was mainly towards extracting new urban patterns that could be of benefit to several types of applications such as transportation, economy, and telecommunications. Through my research work, I have concluded that some of these patterns when correlated with each other, they could further provide deeper insights on the causality behind some challenges that faces cities. This was obvious when correlating the Temporal Functional Regions with the recurrent crowd mobility patterns for the aim of understanding the motivation/reason behind crowd mobility. Similarly, the Temporal Functional Regions and crowd mobility patterns were proved to be correlated with the network demand in cities, this correlation could help in providing insights into the reasons of the variation of network demand across space and time in cities. Hence, I hope that the findings in this thesis might open the door in the future for extending the urban computing field to include the "reasoning" element for further giving insights on the reasons behind the challenges that our cities face today. Such extension could be referred to as "urban reasoning" and Figure 9.1 present my thoughts on where urban reasoning could complement the urban computing framework in the future. In other words, urban reasoning could be described as a multi-stage analytics process comprising of three phases: Domain Knowledge Data Analytics (DKDA) phase, City Wide Scale Data Analytics (CWDA) phase, and the correlation phase between the DKDA and CWDA which will not only aim to provide deeper insights about our cities' dynamicity but more importantly focuses on reasoning from certain challenges that our cities face today.

9.5 Outlook

The emergence of a new generation of mobile web services and applications has generated huge amounts of mobility data of unprecedented geographic scale and spatial granularity. This data is accompanied by other layers of information including social interactions between users and natural language expressions capturing users' activities and opinions.

The fact that every piece of online information is being geo-tagged brings not only new opportunities for extracting new urban patterns in cities or offering better services and new type of applications for users, but comes with challenges for adapting the recent advancements in machine learning to such type of spatio-temporal datasets. It will take some time until the value of this data is fully digested by academics, government institutions and industry, and when this happens, more data, more interesting extracted patterns and hence, more questions will emerge.

In this thesis, I have attempted to take a step towards extracting new and interesting urban patterns and showing the power of LBSNs data as well as the potential impact of some of the extracted patterns on one of the common challenges in the telecommunications domain (Network Demand Prediction). I hope that the patterns that were extracted, will inspire researchers in various industries and academic disciplines to build and provide new type of applications to users and government institutions that will make our cities better in the future. In addition, the concept of the correlation between various extracted urban patterns shown in this thesis for the aim of understanding and reasoning behind some challenges that our cities face today could hopefully emerge new field focusing on “urban reasoning”.

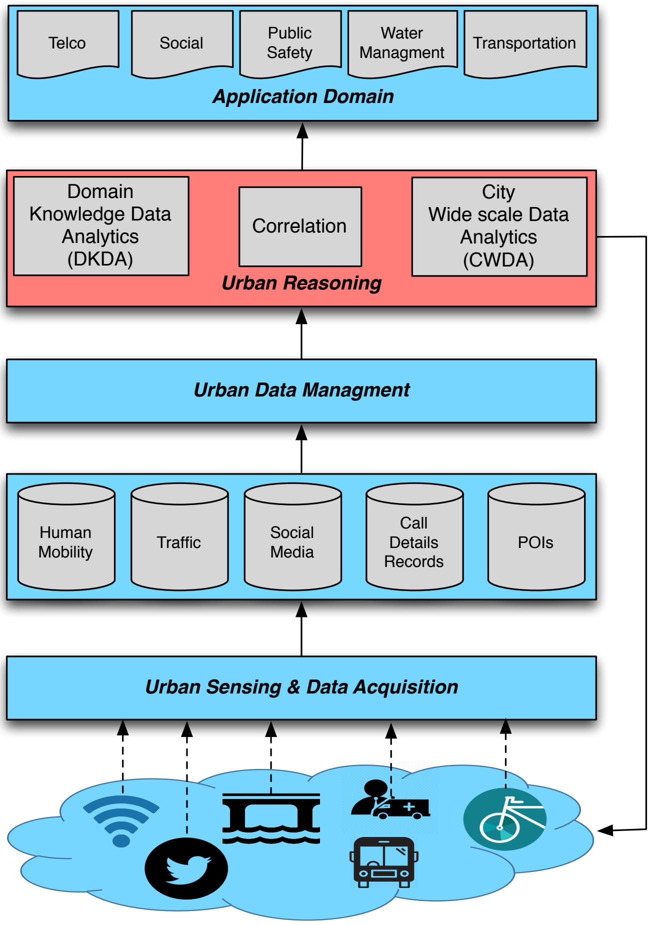


Figure 9.1: Empowering urban computing framework with urban reasoning.

Appendix A

Evaluation of the TCDC Proposed Approach

In this Appendix, the evaluation and validation results for the TCDC-based recommenders applied to the regression and classifications models introduced in chapter 3 are discussed. In the first subsection, we present our methodology for evaluating the TCDC approach describing: (a) The metrics used in our evaluation for measuring the predictive performance of the various models used; (b) The specifications of the datasets used. In the second subsection, we show and discuss the testing results obtained from running the TCDC-based recommenders on the various datasets. Finally, we summarize our evaluation results in the last subsection.

A.1 Evaluation Methodology

A.1.1 Predictive Performance Metric

There are many methods for measuring the predictive performance of the regression and classification models. For the machine learning models that are used for predicting continuous outcome, we need some measure of accuracy in order to evaluate the model. However, there are plenty of ways for measuring the performance of the regression models, one of the most commonly used method for measuring the predictive performance of a model with a continuous response is the *Root Mean Square Error (RMSE)*. The RMSE is evaluated by taking the square root of the Mean Square Error (MSE) so that units of the original data are sustained. We used the RMSE for evaluating the predictive performance of the regression models in the latter sections of the paper.

We now turn to measuring the predictive performance for models with a categorical outcome. Although, there are lots of machine learning methods that are common between regression and classification models, the way that we measure the predictive performance is necessarily very different from both models as a metric like RMSE that is used in regression will not be suitable in the classification context. For evaluating the classification models used in the paper, we used the *Accuracy* as the metric for measuring performance that is the number of correct predictions ($\textit{True positives} + \textit{True negatives}$) from all predictions made. It is common knowledge that classifiers are biased towards preferring classes with categories. Thus, the accuracy may not evaluate the performance of a classifier very well if the classifier works on a highly un-balanced dataset. The ROC/AUC curve is more appropriate to identify the performance of a classifier in such a case. However, applying ROC/AUC to a multiclass classification problem will bring extra complexity to the evaluation work that is outside the scope of this work.

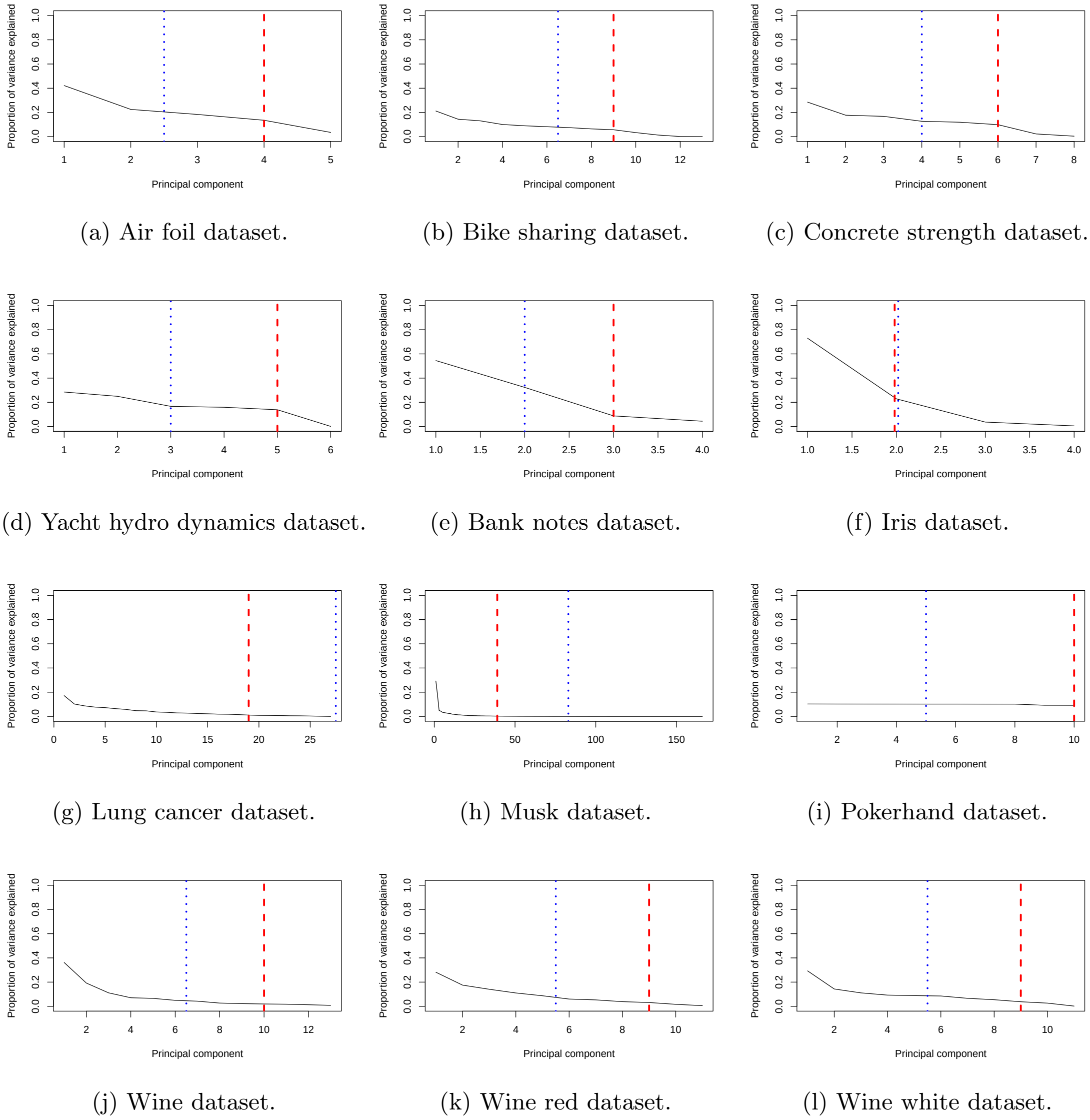
A.1.2 Datasets specifications

In order to show the impact of the proposed TCDC approach, we selected the datasets for the testing and validation tasks that fulfils the following criteria: (a) Widely-known datasets that have been used for a long time already for benchmarking within the machine learning community (e.g.: Iris dataset); (b) Publicly and open source datasets, most of the datasets used can be found in [174]; (c) Diverse datasets with respect to different domains (health, agriculture, environmental, etc), this will facilitate demonstrating the impact of our approach and its applicability on different domains. We implemented and applied the proposed TCDC approach on 12 datasets and we illustrate in the paper their results. Table A.1 summarizes the datasets' specifications used in the paper where N specifies the number of samples in the datasets, F is the number of features exist in the dataset, the fourth column represent the number of principle components that capture 95% of the variances of the features, the high dimensional column captures whether the number of features is greater than 100 or not, and finally the high correlation columns identify the degree of between-features correlation.

Next in order to determine the degree of between-features correlation, we applied PCA on the full set of transformed features of all datasets shown in Table A.1 and the variances accounted for by each component was determined. In Figure A.1 scree plots show a profile of the variability accounted by each component. We define a *Low correlation* dataset as one when more than half the number of features can capture at least 95% of the variances between the features. In Figure A.1, this is illustrated with the red line showing half the number of features that exist in the dataset and the blue line outlining the number of principle components capturing 95% of the variances. In another words, when the blue line comes after the red line, it indicates low correlation between features and vice versa. Before applying PCA, and in order to avoid summarizing feature scale information, we have transformed skewed features in the dataset and then center and scale the features before performing PCA. We choose to use PCA for determining the degree of between feature correlations because when the datasets become high dimensional (too many features), it starts to be harder to visually examine the correlation matrix of the training set. For this reason we believe this method is more suitable in big datasets.

Table A.1: A summary of datasets and some of their characteristics.

Dataset	Used for	N	F	PC(95%)	N >5F	High Dimensional	High Correlation
Airfoil	Regression	1503	5	4	Yes	No	No
Bank notes	Classification	1372	4	3	Yes	No	No
Bike sharing	Regression	17379	15	9	Yes	No	No
Concrete compressive strength	Regression	1030	8	6	Yes	No	No
Iris	Classification	150	4	2	Yes	No	Yes
Lung cancer	Classification	32	56	19	No	No	Yes
Musk	Classification	6598	168	39	Yes	Yes	No
Poker hand	Classification	1025010	10	10	Yes	No	No
Wine	Classification	178	12	10	Yes	Yes	No
Wine red	Classification	1599	11	9	Yes	No	No
Wine white	Classification	4898	11	9	Yes	No	No
Yacht hyrdodynamics	Regression	252	6	5	Yes	No	No



(a) Air foil dataset.

(b) Bike sharing dataset.

(c) Concrete strength dataset.

(d) Yacht hydro dynamics dataset.

(e) Bank notes dataset.

(f) Iris dataset.

(g) Lung cancer dataset.

(h) Musk dataset.

(i) Pokerhand dataset.

(j) Wine dataset.

(k) Wine red dataset.

(l) Wine white dataset.

Figure A.1: Scree plots for datasets used in regression and classification evaluation.

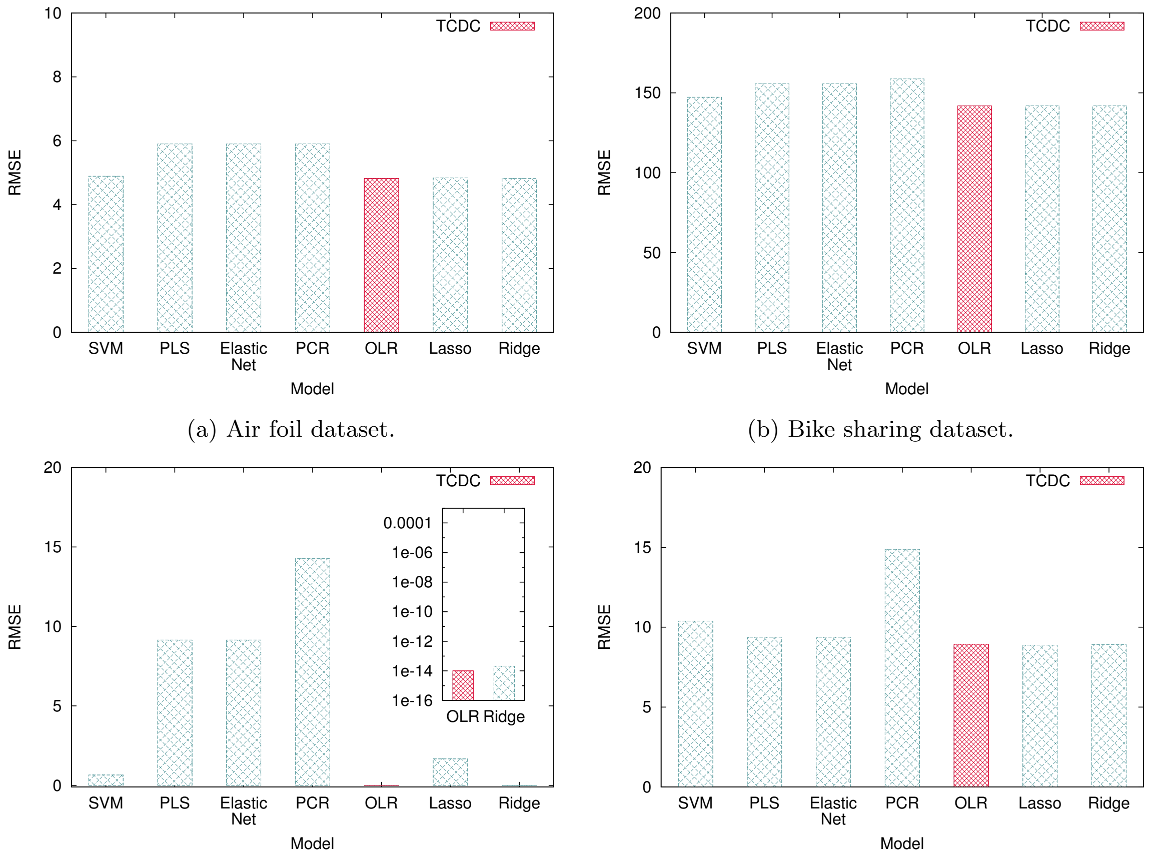
A.2 Testing Results

In this subsection, we show the results of applying the TCDC-based recommenders for both regression and classification to the datasets. In order to produce unbiased and realistic comparison, we did not perform any kind of feature selection for all models trained and we used the default settings for all algorithms with no fine tuning. In addition, we applied the required features transformations (e.g.: centring, scaling, removing missing values) for all models before training. As shown in table A.1, we have performed our testing on 4 datasets for regression and 8 datasets for classification. We considered the recommenders to be blinded as to whether the datasets are linear or non-linear and hence, we performed on each dataset both paths of linearity and non-linearity tests as shown. Besides, we considered the tolerance factor discussed before in chapter 3 (see Sec. 3.2.1) to be equal zero in order to have unbiased results towards comparing only the predictive performance.

A.2.1 Regression Results

Figure A.2 shows the linear regression results applied for the 4 datasets used for regression where the x axis indicates the dataset name and the y axis shows the predictive performance using the RMSE metric as described in the previous section. The model which is selected via the TCDC approach is highlighted in red. From the introduced TCDC-based recommender for regression (refer to chapter 3, Figure 3.3), it can be seen that the recommendation path leads to the OLR model being recommended for all datasets used (Airfoil, Bike sharing, Concrete compressive strength and Yacht hydro dynamics). This is because these datasets have low between-features correlation and their number of samples is greater than the number of features. In Figure A.2a, Figure A.2b and Figure A.2d, OLR outperforms along with the Lasso and Ridge models and in Figure A.2c, OLR outperforms along with the Ridge model compared to all other models. Obviously, TCDC resulted in the outperforming model (OLR) for all of the prior datasets along with more benefits rather choosing the other comparable performance models (Lasso or Ridge). OLR comes with the benefit of being very attractive due to its interpretability of its coefficients and its low computational complexity.

Figure A.3 shows the non-linear regression results for the predictive performance using RMSE for all of the models used in the lower part of TCDC-based recommender for regression (refer to chapter 3, Figure 3.3) with highlighting in red the chosen model from the TCDC approach. For the Air foil dataset results shown in Figure A.3a, Random Forests outperformed all other models with the smallest RMSE which coincides with the outcome from the TCDC approach. Random Forest was chosen by the TCDC due to the fact that MARS performed worse than SVM with RBF kernel, leading to progressing and trying trees. Comparing the predictive performance of the Interpretable (M5 & Rule based models) and Non-Interpretable trees (Boosting and Random Forest), it was found that the Random Forests is the best predictive performance model. In Figure A.3b, Figure A.3d and Figure A.3c, the best predictive performance model was found to be Random Forest for the first two and Model Trees for the latter one. However, the TCDC approach selects



(a) Air foil dataset.

(b) Bike sharing dataset.

(c) Concrete strength dataset.

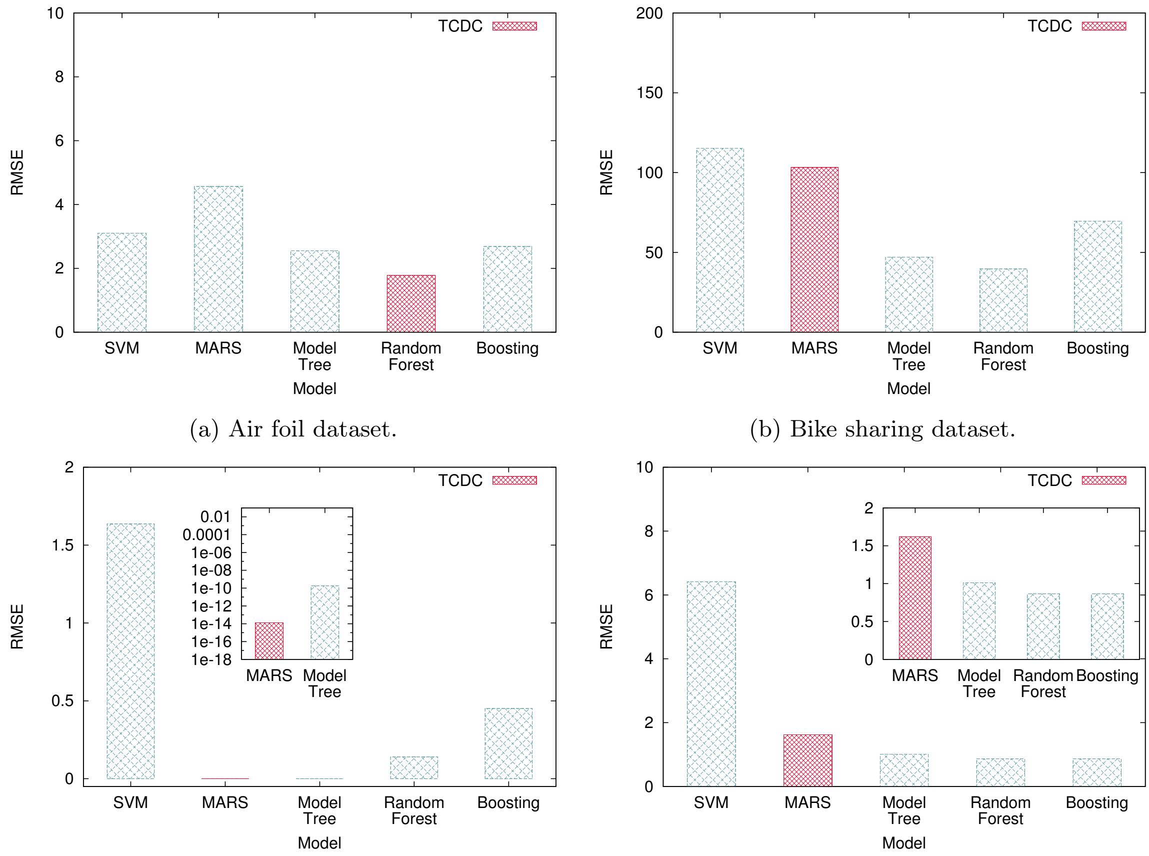
(d) Yacht hydro dynamics dataset.

Figure A.2: TCDC-based Linear Regression Results

the MARS model due to the fact that MARS performed better than SVM. This was due to the fact that we used the default settings of SVM, however, SVM is from the models that requires careful parameter selection unlike Bagging which works well with the most decision tree types and requires little fine tuning [175]. Although TCDC choose MARS, it still comes with the benefit of being able to conduct an automatic feature selection and considered a high interpretable model in which correlated features do not drastically impact its performance.

A.2.2 Classification Results

Figure A.4 shows the linear classification results applied for the 8 datasets used for classification as described in table A.1 where x axis indicates the dataset name and the y axis shows the predictive performance using the Accuracy metric as described in the previous section. In a similar manner to the regression results, the model which is selected by the TCDC approach is highlighted in red. Figure A.4a shows the results outcome of the Bank notes data set. According to the TCDC-based recommender for classification (refer to chapter 3, Figure 3.4), the LDA model is selected, trained and its performance is compared with SVM with linear kernel. This is due to the fact that the between-features correlations are low and the number of samples is greater than 5 times the number of fea-



(a) Air foil dataset.

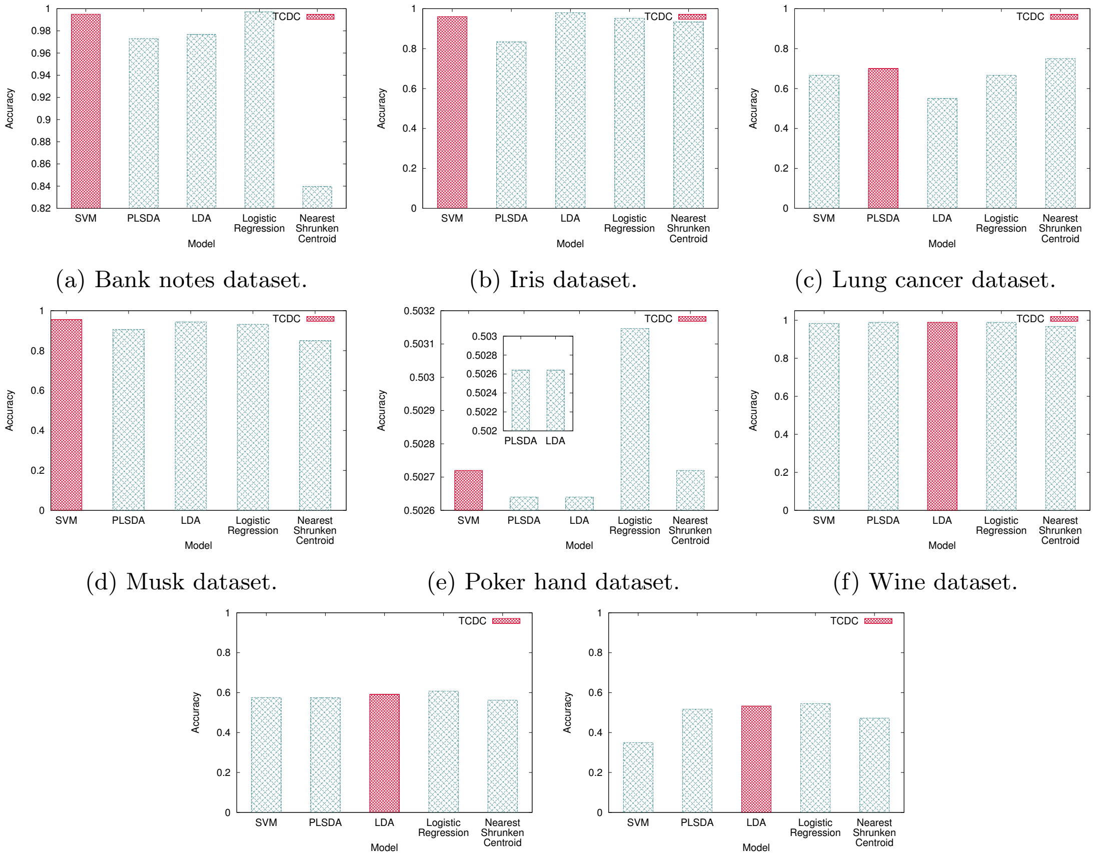
(b) Bike sharing dataset.

(c) Concrete strength dataset.

(d) Yacht hydro dynamics dataset.

Figure A.3: TCDC-based Non-linear Regression Results

tures in the Bank notes dataset. The outcomTCe of this comparison resulted in choosing SVM which is actually the best performing model. In Figure A.4b, the TCDC approach choses the PLSDA to train and compare its performance with SVM. This is due to that the Iris dataset is of high between-features correlation. The SVM outperforms compared to PLSDA and hence, it has been chosen. TCDC selects SVM as the best model for the Iris dataset and actually it performs nearly very close compared to LDA which was the best predictive performance model. In Figure A.4c, TCDC selects PLSDA to train and be compared to SVM due to the high between-features correlations in the dataset. PLSDA performs very close to the best model (Nearest Shrunken Centroid) and it comes with more interpretability and less complexity for being deployed. For the Musk dataset illustrated in Figure A.4d which is high dimensional dataset with more than 100 features and number of samples more than the features, the Nearest Shrunken Centroid model is trained and its performance is compared with SVM. The SVM outperformed all others and hence, it has been chosen from TCDC matching the selection of the best predictive performance model on this dataset. Similarly SVM was chosen by the TCDC recommender for the Poker hand dataset shown in Figure A.4e as it outperformed the LDA model. However, the Logistic Regression is the best model performing on this dataset with a difference of 2% compared to SVM. For the Wine, Wine red and Wine white datasets illustrated in



(a) Bank notes dataset.

(b) Iris dataset.

(c) Lung cancer dataset.

(d) Musk dataset.

(e) Poker hand dataset.

(f) Wine dataset.

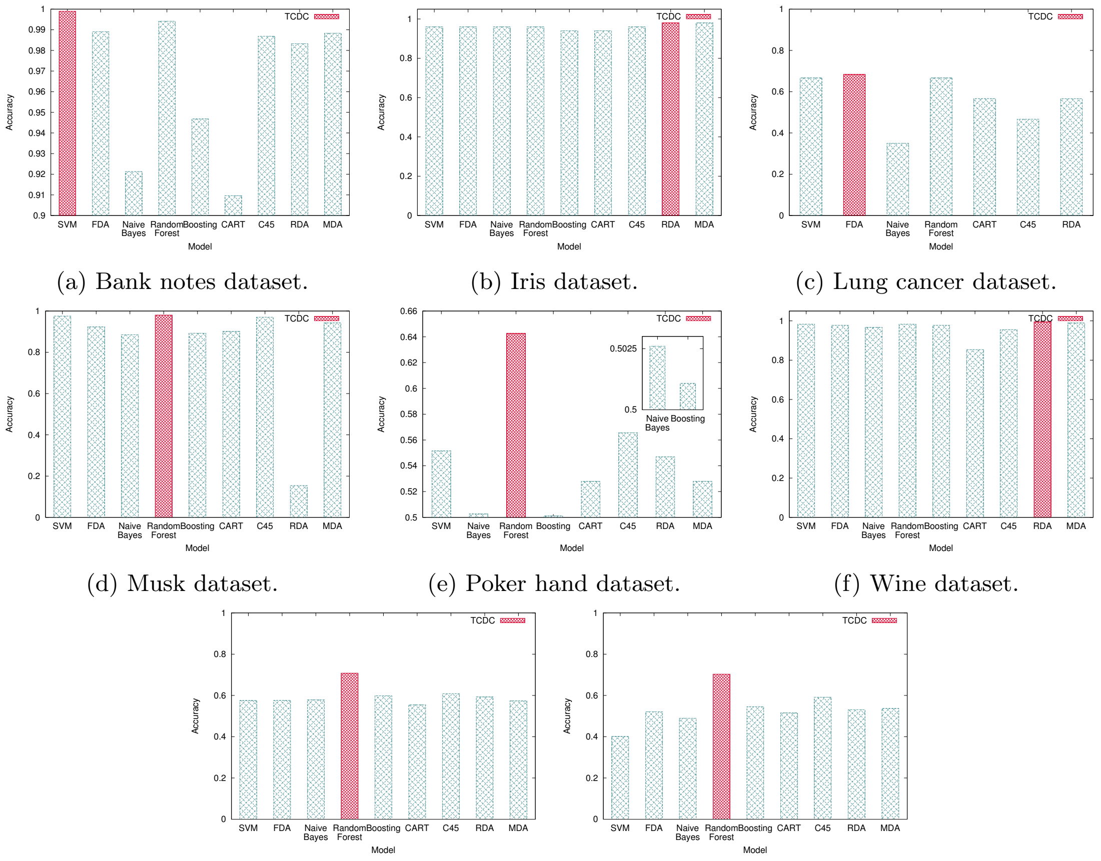
(g) Wine red dataset.

(h) Wine white dataset.

Figure A.4: TCDC-based Classification Linear Classification Results

Figure A.4f, Figure A.4g and Figure A.4h respectively, the LDA which is the best performing model has been chosen by TCDC since the between-features correlations is low with the number of samples is greater than 5 times the number of features. Besides, LDA outperformed when it is compared with SVM.

Figure A.5 shows the non-linear classification results performed on the 8 datasets for classification. Figure A.4a shows the results of the different ML models applied to the Bank notes dataset. Clearly, SVM with RBF kernel is the best model in terms of predictive accuracy. TCDC approach chooses SVM as well since it moved through the whole path as shown in the lower part of TCDC-based recommender for classification (refer to chapter 3, Figure 3.4) and SVM still outperformed all other models trained. For the Iris and Wine datasets shown in Figure A.5b and Figure A.5f, the best model performed was RDA which comes exactly with the same outcome from the TCDC recommender since both datasets are considered small datasets and the performance of RDA outperforms SVM. On the other hand and for the Lung cancer dataset with number of samples exceeding 10 times the number of features, the FDA model was selected by the TCDC approach since it outperformed SVM and it was found to be the best model. For the Musk, Poker hand, Wine red, and Wine white datasets visualised in Figure A.5d, Figure A.5e, Figure A.5g and Figure A.5h respectively, they all followed the same path in the TCDC approach.



(a) Bank notes dataset.

(b) Iris dataset.

(c) Lung cancer dataset.

(d) Musk dataset.

(e) Poker hand dataset.

(f) Wine dataset.

(g) Wine red dataset.

(h) Wine white dataset.

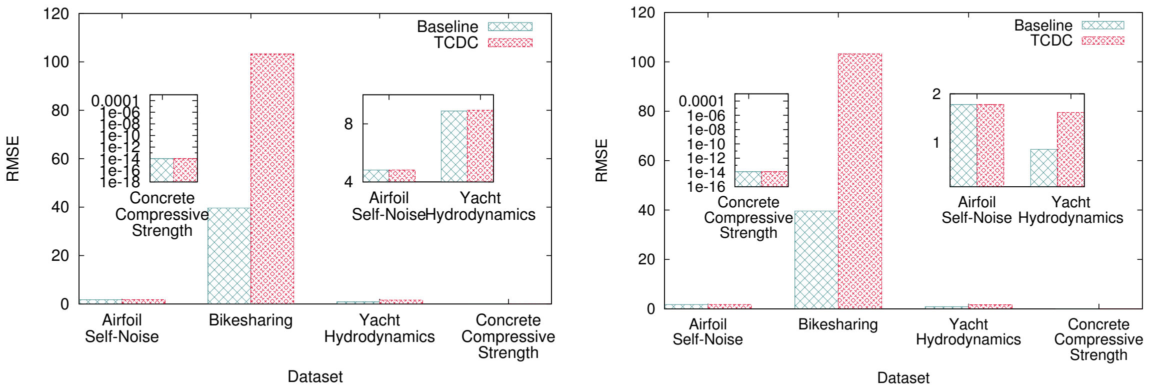
Figure A.5: TCDC-based Classification Non-linear Classification Results.

Firstly, they have been trained against FDA and the resulted performance was found to be less than SVM with RBF kernel. Secondly, they have been trained against MDA since they are not considered small datasets and again their performance on MDA was found less than SVM. Finally, they got trained against interpretable and non-interpretable trees and Random Forest was selected with the best performance. This coincides with the fact that Random Forest performs the best for these datasets compared to all models trained as clearly demonstrated in the figures.

A.3 Evaluation Summary

From the testing and evaluation discussed in the previous section, it clearly illustrates that choosing the best performing model would be a very time consuming process if we took the default approach of trying lots of different models and comparing them blindly regardless of the dataset's specification. We have shown in the previous section that in many cases, the TCDC recommenders can select the best performing model with less trials and shorter steps. To summarize our evaluation results, we need first to define two terms which are: (a) Baseline Approach: This is the default approach in which someone tries to train the set of machine learning models randomly and chooses the best one in terms of predictive performance. (b) TCDC Approach: This is the approach introduced in the paper which tries to select the optimum machine learning model in a smarter way by taking into account the dataset's specification before training and taking into account the predictive performance along with the introduced TCDC benefit metrics when two models are very close in their performance.

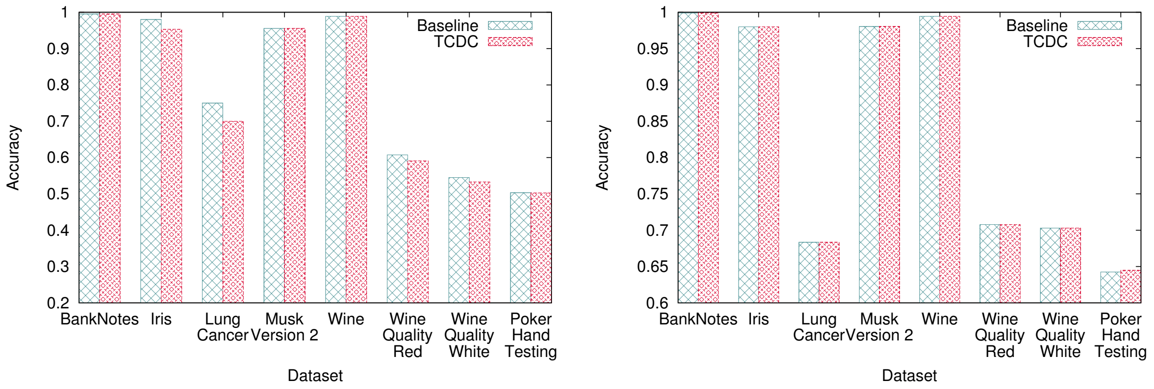
Figure A.6 and Figure A.7 show the summary of results comparing the predictive performance for the Baseline and TCDC approaches for all the 12 datasets we used in the testing and evaluation in the paper. The x axis indicates the dataset name and the y indicates the predictive performance metric whether it is RMSE for regression or Accuracy for classification. The TCDC approach was found to select the best model in terms of predictive performance in 62.5% for all the regression tests performed (75% for linear regression and 50% for non-linear regression) and 75% for all the classification tests (50% for linear classification and 100% for non-linear classification). In most cases in which the TCDC approach did not match the baseline, the difference in the predictive performance was very close.



(a) Linear Regression.

(b) Non-linear Regression.

Figure A.6: TCDC-based Regression Results Summary



(a) Linear Classification.

(b) Non-linear Classification.

Figure A.7: TCDC-based Classification Results Summary.

Appendix B

Demonstrator screenshots

In this Appendix, some screenshots are presented for the demonstrator that is developed during the course of the PhD for visualizing some of the patterns extracted and introduced in this thesis. The developed demonstrator allows end-users (e.g., cities' authorities) to visualize such patterns in realtime and take some mitigation actions when something abnormal happens. It is worth noting that in the developed demonstrator, some more features are implemented that are not discussed in the core chapters in this thesis for showing some examples of how these patterns could be useful to end-users.

Figure B.1 shows the temporal functional regions changing across time. As in the demo, a grid map is used for defining the regions boundaries for Manhattan rather than zip-codes, it was needed to decrease the number of functionalities to overcome some sparsity challenges when using smaller areas for regions boundaries. Figure B.2 illustrates the crowd mobility across space and time while Figure B.3 shows the network demand prediction across space and time taking into account the prior extracted patterns using the ST-DenNetFus proposed approach as presented in chapter 8.

Figure B.4 visualizes the locations with abnormal patterns shown in red, these abnormality could be as a result of unexpected network demand in certain area or crowd anomalous behaviour. Furthermore, the end-user could specify the warning communication mode as shown in Figure B.5 whether they would like to receive warnings through an email, sms, or a webhook. In case of abnormal crowd patterns, the warning mode might be "calling police station". Finally and based on the network demand prediction, suggestions for the location of portable base station could be recommended based on the areas that require more network coverage. Figure B.6 shows screenshot of the demo for this part.

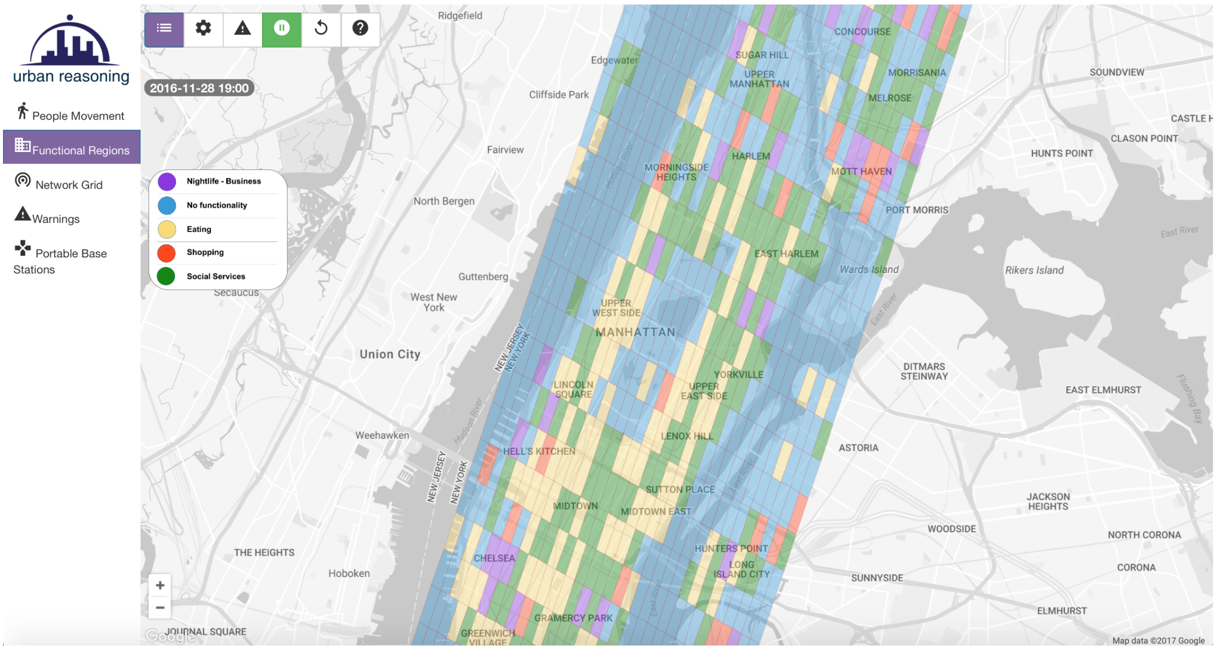


Figure B.1: Temporal Functional Regions variation in realtime.

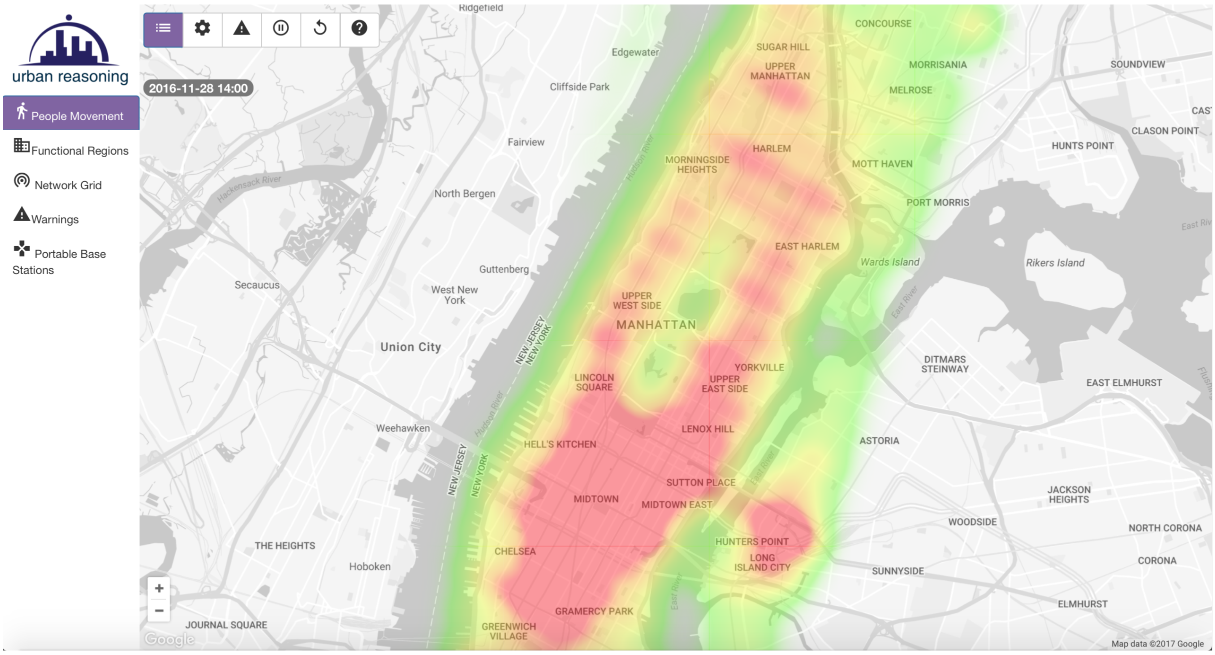


Figure B.2: Crowd Mobility Patterns in realtime.

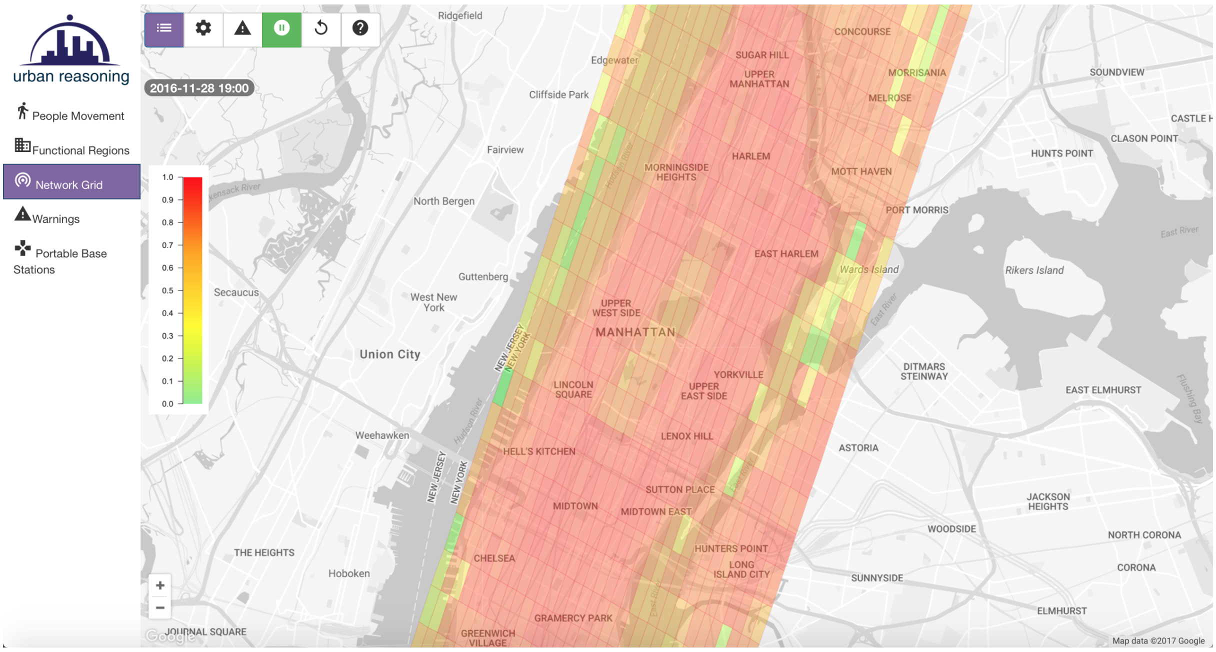


Figure B.3: Network Demand Prediction in realtime taking into account various external factors.

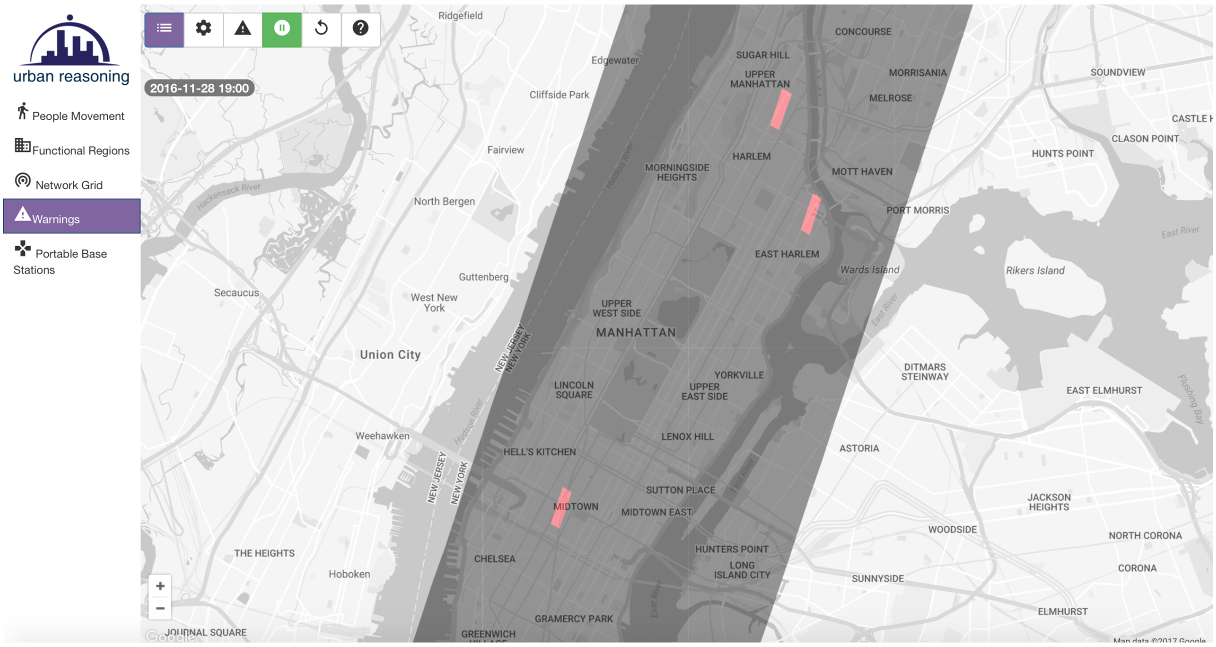
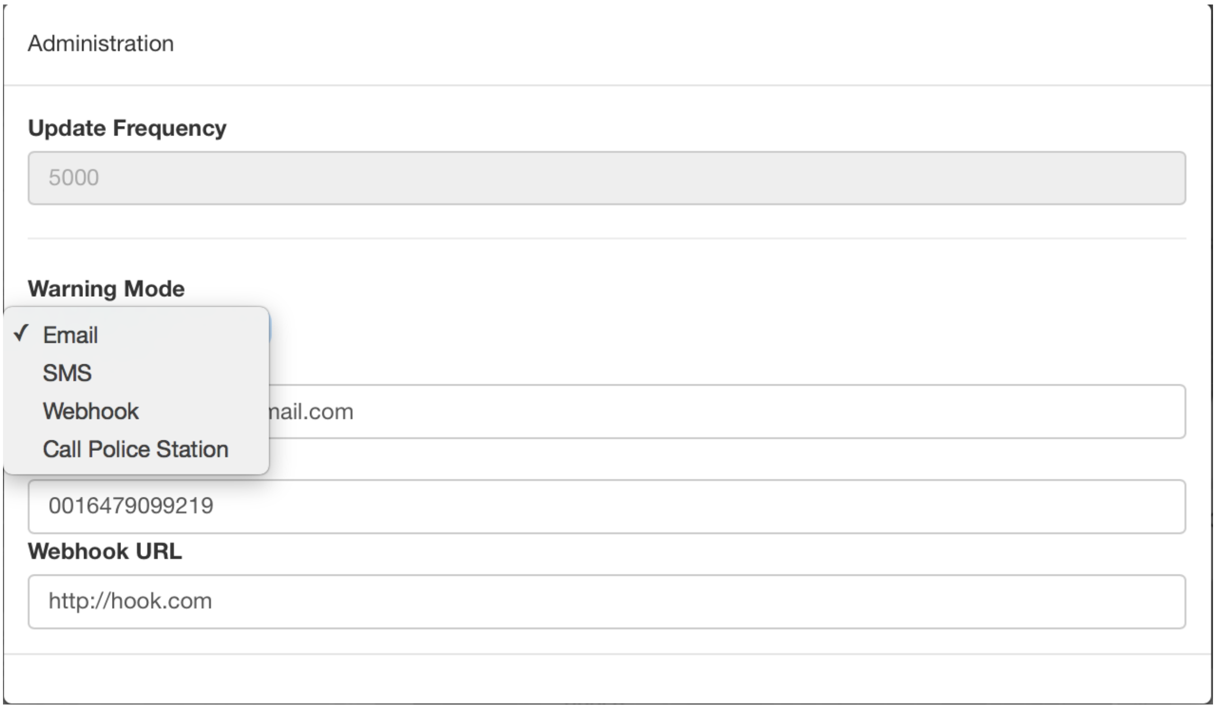


Figure B.4: Abnormal patterns in realtime.



Administration

Update Frequency

5000

Warning Mode

✓ Email

SMS

Webhook

Call Police Station

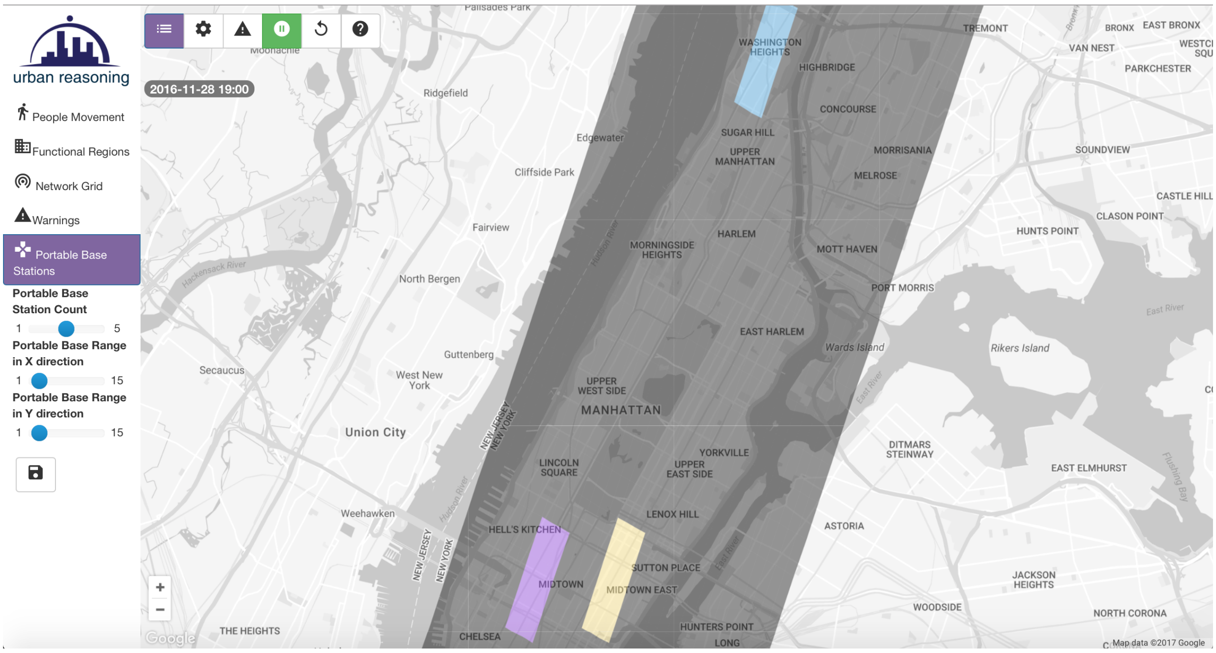
nail.com

0016479099219

Webhook URL

http://hook.com

Figure B.5: Warning modes.



Portable Base Station Count 1 5

Portable Base Range in X direction 1 15

Portable Base Range in Y direction 1 15

Figure B.6: Optimum location for the portable base stations based on the predicted network demand.

Bibliography

[1] Richard Heeks. Ict4d 2.0: The next phase of applying ict for international development. *Computer*, 41(6):26–33, 2008.

[2] Smartphone Users Worldwide Will Total. 1.75 billion in 2014. *Mobile users pick up smartphones as they become more affordable, 3G and 4G networks advance*, 2014.

[3] Kathryn Zickuhr. Three-quarters of smartphone owners use location-based services. *Pew Internet & American Life Project*, 2012.

[4] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. Exploiting semantic annotations for clustering geographic areas and users in location-based social networks. *The social mobile web*, 11:02, 2011.

[5] Zhiyuan Cheng, James Caverlee, Kyumin Lee, and Daniel Z Sui. Exploring millions of footprints in location sharing services. *ICWSM*, 2011:81–88, 2011.

[6] Jie Bao, Yu Zheng, David Wilkie, and Mohamed Mokbel. Recommendations in location-based social networks: a survey. *Geoinformatica*, 19(3):525–565, 2015.

[7] Hui-Huang Hsu, Chuan-Yu Chang, and Ching-Hsien Hsu. *Big Data Analytics for Sensor-Network Collected Intelligence*. Morgan Kaufmann, 2017.

[8] Nicholas Jing Yuan, Yu Zheng, Xing Xie, Yingzi Wang, Kai Zheng, and Hui Xiong. Discovering urban functional zones using latent activity trajectories. *IEEE Transactions on Knowledge and Data Engineering*, 27(3):712–725, 2015.

[9] Huiji Gao, Jiliang Tang, and Huan Liu. gscorr: modeling geo-social correlations for new check-ins on location-based social networks. In *Proceedings of the 21st ACM international conference on Information and knowledge management*, pages 1582–1586. ACM, 2012.

[10] Craig Collins, Samiul Hasan, and Satish V Ukkusuri. A novel transit rider satisfaction metric: Rider sentiments measured from online social media data. *Journal of Public Transportation*, 16(2):2, 2013.

[11] Jingbo Shang, Yu Zheng, Wenzhu Tong, Eric Chang, and Yong Yu. Inferring gas consumption and pollution emission of vehicles throughout a city. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1027–1036. ACM, 2014.

[12] Fuzheng Zhang, David Wilkie, Yu Zheng, and Xing Xie. Sensing the pulse of urban refueling behavior. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 13–22. ACM, 2013.

[13] Yanjie Fu, Hui Xiong, Yong Ge, Zijun Yao, Yu Zheng, and Zhi-Hua Zhou. Exploiting geographic dependencies for real estate appraisal: a mutual perspective of ranking and clustering. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1047–1056. ACM, 2014.

[14] Alon Halevy, Peter Norvig, and Fernando Pereira. The unreasonable effectiveness of data. *IEEE Intelligent Systems*, 24(2):8–12, 2009.

[15] Hatem Abou-Zeid and Hossam S Hassanein. Predictive green wireless access: Exploiting mobility and application information. *IEEE Wireless Communications*, 20(5):92–99, 2013.

[16] Xin Dong, Wentao Fan, and Jun Gu. Predicting lte throughput using traffic time series. *ZTE Communications*, 4:014, 2015.

[17] Ernest George Ravenstein. The laws of migration. *Journal of the Statistical Society of London*, 48(2):167–235, 1885.

[18] Everett S Lee. A theory of migration. *Demography*, 3(1):47–57, 1966.

[19] Michael J Greenwood. Research on internal migration in the united states: a survey. *Journal of Economic Literature*, pages 397–433, 1975.

[20] Wilbur Zelinsky. The hypothesis of the mobility transition. *Geographical review*, pages 219–249, 1971.

[21] Juan de Dios Ortuzar, Luis G Willumsen, et al. *Modelling transport*. Wiley New Jersey, 1994.

[22] Martin J Beckmann. On the theory of traffic flow in networks. *Traffic Quarterly*, 21(1), 1967.

[23] Geoffrey M Hyman. The calibration of trip distribution models. *Environment and Planning A*, 1(1):105–112, 1969.

[24] Shan Jiang, Joseph Ferreira Jr, and Marta C Gonzalez. Discovering urban spatial-temporal structure from human activity patterns. In *Proceedings of the ACM SIGKDD international workshop on urban computing*, pages 95–102. ACM, 2012.

[25] Richard A Becker, Ramon Caceres, Karrie Hanson, Ji Meng Loh, Simon Urbanek, Alexander Varshavsky, and Chris Volinsky. A tale of one city: Using cellular network data for urban planning. *IEEE Pervasive Computing*, 10(4):18–26, 2011.

[26] Nathan Eagle and Alex Sandy Pentland. Eigenbehaviors: Identifying structure in routine. *Behavioral Ecology and Sociobiology*, 63(7):1057–1066, 2009.

[27] Andreas M Kaplan and Michael Haenlein. Users of the world, unite! the challenges and opportunities of social media. *Business horizons*, 53(1):59–68, 2010.

[28] Anastasios Noulas, Cecilia Mascolo, and Random Walk Restart. *Human urban mobility in location-based social networks: analysis, models and applications.* PhD thesis, University of Cambridge, UK, 2013.

[29] Daniele Quercia, Neal Lathia, Francesco Calabrese, Giusy Di Lorenzo, and Jon Crowcroft. Recommending social events from mobile phone location data. In *Data Mining (ICDM), 2010 IEEE 10th International Conference on*, pages 971–976. IEEE, 2010.

[30] Yu Zheng and Xiaofang Zhou. *Computing with spatial trajectories.* Springer Science & Business Media, 2011.

[31] Henriette Cramer, Mattias Rost, and Lars Erik Holmquist. Performing a check-in: emerging practices, norms and'conflicts' in location-sharing using foursquare. In *Proceedings of the 13th international conference on human computer interaction with mobile devices and services*, pages 57–66. ACM, 2011.

[32] Salvatore Scellato, Anastasios Noulas, Renaud Lambiotte, and Cecilia Mascolo. Socio-spatial properties of online location-based social networks. *ICWSM*, 11:329–336, 2011.

[33] Anastasios Noulas, Salvatore Scellato, Cecilia Mascolo, and Massimiliano Pontil. An empirical study of geographic user activity patterns in foursquare. *ICwSM*, 11:70–573, 2011.

[34] Kenneth Joseph, Chun How Tan, and Kathleen M Carley. Beyond local, categories and friends: clustering foursquare users with latent topics. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 919–926. ACM, 2012.

[35] Yu Zheng, Licia Capra, Ouri Wolfson, and Hai Yang. Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):38, 2014.

[36] Nicola Bicocchi, Gabriella Castelli, Marco Mamei, Alberto Rosi, and Franco Zambonelli. Supporting location-aware services for mobile users with the whereabouts diary. In *MobilWare*, 2008.

[37] Stephan Sigg, Sandra Haseloff, and Klaus David. An alignment approach for context prediction tasks in ubicomp environments. *PERCOM*, 2010.

[38] Laura Ferrari, Alberto Rosi, Marco Mamei, and Franco Zambonelli. Extracting urban patterns from location-based social networks. In *SIGSPATIAL*, 2011.

[39] David Jurgens and Keith Stevens. The s-space package: an open source package for word space models. In *Proceedings of the ACL 2010 System Demonstrations*, pages 30–35. Association for Computational Linguistics, 2010.

[40] Tim Van de Cruys and Marianna Apidianaki. Latent semantic word sense induction and disambiguation. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 1476–1485. Association for Computational Linguistics, 2011.

[41] Samuel Brody and Mirella Lapata. Bayesian word sense induction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 103–111. Association for Computational Linguistics, 2009.

[42] David Andrzejewski and David Buttler. Latent topic feedback for information retrieval. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 600–608. ACM, 2011.

[43] Laura Ferrari and Marco Mamei. Discovering daily routines from google latitude with topic models. In *PERCOM*, 2011.

[44] Samiul Hasan and Satish V Ukkusuri. Urban activity pattern classification using topic models from online geo-location data. *Transportation Research Part C: Emerging Technologies*, 44:363–381, 2014.

[45] Felix Kling and Alexei Pozdnoukhov. When a city tells a story: urban topic analysis. In *SIGSPATIAL*, 2012.

[46] Zipei Fan, Xuan Song, Ryosuke Shibasaki, and Ryutaro Adachi. Citymomentum: an online approach for crowd behavior prediction at a citywide level. In *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 559–569. ACM, 2015.

[47] Xuan Song, Quanshi Zhang, Yoshihide Sekimoto, and Ryosuke Shibasaki. Prediction of human emergency behavior and their mobility following large-scale disaster. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 5–14. ACM, 2014.

[48] Afshin Abadi, Tooraj Rajabioun, and Petros A Ioannou. Traffic flow prediction for road transportation networks with limited traffic data. *IEEE Transactions on Intelligent Transportation Systems*, 16(2):653–662, 2015.

[49] Po-Ta Chen, Feng Chen, and Zhen Qian. Road traffic congestion monitoring in social media with hinge-loss markov random fields. In *2014 IEEE International Conference on Data Mining*, pages 80–89. IEEE, 2014.

[50] Ricardo Silva, Soong Moon Kang, and Edoardo M Airoldi. Predicting traffic volumes and estimating the effects of shocks in massive transportation systems. *Proceedings of the National Academy of Sciences*, 112(18):5643–5648, 2015.

[51] Yanyan Xu, Qing-Jie Kong, Reinhard Klette, and Yuncai Liu. Accurate and interpretable bayesian mars for traffic flow prediction. *IEEE Transactions on Intelligent Transportation Systems*, 15(6):2457–2469, 2014.

[52] Minh X Hoang, Yu Zheng, and Ambuj K Singh. Forecasting citywide crowd flows based on big data. *ACM SIGSPATIAL*, 2016.

[53] Junbo Zhang, Yu Zheng, and Dekang Qi. Deep spatio-temporal residual networks for citywide crowd flows prediction. *ACM SIGSPATIAL*, 2016.

[54] Geoffrey E Hinton, Simon Osindero, and Yee-Whye Teh. A fast learning algorithm for deep belief nets. *Neural computation*, 18(7):1527–1554, 2006.

[55] Yoshua Bengio, Pascal Lamblin, Dan Popovici, Hugo Larochelle, et al. Greedy layer-wise training of deep networks. In *NIPS*, 2007.

[56] Christopher Poultney, Sumit Chopra, Yann L Cun, et al. Efficient learning of sparse representations with an energy-based model. In *Advances in neural information processing systems*, pages 1137–1144, 2006.

[57] Ian H Witten and Eibe Frank. *Data Mining: Practical machine learning tools and techniques*. Morgan Kaufmann, 2005.

[58] Pedro Domingos. A few useful things to know about machine learning. *Communications of the ACM*, 55(10):78–87, 2012.

[59] Torsten Hothorn, Friedrich Leisch, Achim Zeileis, and Kurt Hornik. The design and analysis of benchmark experiments. *Journal of Computational and Graphical Statistics*, 14(3):675–699, 2005.

[60] Manuel JA Eugster, Torsten Hothorn, and Friedrich Leisch. Exploratory and inferential analysis of benchmark experiments. 2008.

[61] Max Kuhn and Kjell Johnson. *Applied predictive modeling*. Springer, 2013.

[62] Yann LeCun, LD Jackel, L Bottou, A Brunot, C Cortes, JS Denker, H Drucker, I Guyon, UA Muller, E Sackinger, et al. Comparison of learning algorithms for handwritten digit recognition. In *International conference on artificial neural networks*, volume 60, pages 53–60, 1995.

[63] Annette M Molinaro, Richard Simon, and Ruth M Pfeiffer. Prediction error estimation: a comparison of resampling methods. *Bioinformatics*, 21(15):3301–3307, 2005.

[64] J Kent Martin and Daniel S Hirschberg. Small sample statistics for classification error rates ii: Confidence intervals and significance tests, 1996.

[65] Bradley Efron and Robert Tibshirani. Bootstrap methods for standard errors, confidence intervals, and other measures of statistical accuracy. *Statistical science*, pages 54–75, 1986.

[66] William S Cleveland. Robust locally weighted regression and smoothing scatterplots. *Journal of the American statistical association*, 74(368):829–836, 1979.

[67] Jerome Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

[68] Robert Tibshirani, Trevor Hastie, Balasubramanian Narasimhan, and Gilbert Chu. Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proceedings of the National Academy of Sciences*, 99(10):6567–6572, 2002.

[69] LBJFR Olshen, Charles J Stone, et al. Classification and regression trees. *Wadsworth International Group*, 93(99):101, 1984.

[70] Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 2012.

[71] David M Blei, Andrew Y Ng, and Michael I Jordan. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022, 2003.

[72] Charu C Aggarwal and ChengXiang Zhai. *Mining text data*. Springer Science & Business Media, 2012.

[73] Yoshua Bengio. Learning deep architectures for ai. *Foundations and trends® in Machine Learning*, 2(1):1–127, 2009.

[74] Geoffrey E Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002.

[75] Yoav Freund and David Haussler. Unsupervised learning of distributions of binary vectors using two layer networks, 1994.

[76] Hossein Mobahi, Ronan Collobert, and Jason Weston. Deep learning from temporal coherence in video. In *ICML*, 2009.

[77] Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *Science*, 313(5786):504–507, 2006.

[78] Ronan Collobert and Jason Weston. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *ICML*, 2008.

[79] Hugo Larochelle, Dumitru Erhan, Aaron Courville, James Bergstra, and Yoshua Bengio. An empirical evaluation of deep architectures on problems with many factors of variation. In *ICML*, 2007.

[80] Bernhard Scholkopf and Alexander J Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond.* MIT press, 2001.

[81] LD Jackel, B Boser, HP Graf, JS Denker, Y Le Cun, D Henderson, O Matan, RE Howard, and KS Baird. Vlsi implementations of electronic neural networks: An example in character recognition. In *Systems, Man and Cybernetics, 1990. Conference Proceedings., IEEE International Conference on*, pages 320–322. IEEE, 1990.

[82] Ulrich Ramacher, Wolfgang Raab, J Anlauf, Ulrich Hachmann, Jörg Beichter, Nico Brüls, Matthias Wesseling, Elisabeth Sicheneder, Reinhard Männer, Joachim Gläß, et al. Multiprocessor and memory architecture of the neurocomputer synapse-1. *International journal of neural systems*, 4(04):333–336, 1993.

[83] Davide Anguita, Giancarlo Parodi, and Rodolfo Zunino. An efficient implementation of bp on risc-based workstations. *Neurocomputing*, 6(1):57–65, 1994.

[84] Ura A Muller, Anton Gunzinger, and Walter Guggenbuhl. Fast neural net simulation with a dsp processor array. *IEEE Transactions on Neural Networks*, 6(1):203–213, 1995.

[85] Katayoun Farrahi and Daniel Gatica-Perez. Discovering routines from large-scale human locations using probabilistic topic models. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 2(1):3, 2011.

[86] Anil K Jain, Jianchang Mao, and KM Mohiuddin. Artificial neural networks: A tutorial. *Computer*, (3):31–44, 1996.

[87] Paul Smolensky. Parallel distributed processing: explorations in the microstructure of cognition, vol. 1. chapter information processing in dynamical systems: foundations of harmony theory. *MIT Press, Cambridge, MA, USA*, 15:18, 1986.

[88] Ruslan Salakhutdinov and Geoffrey Hinton. Semantic hashing. *International Journal of Approximate Reasoning*, 50(7):969–978, 2009.

[89] Geoffrey E Hinton and Ruslan R Salakhutdinov. Replicated softmax: an undirected topic model. In *NIPS*, 2009.

[90] Geoffrey Hinton and Ruslan Salakhutdinov. Discovering binary codes for documents by learning deep generative models. *Topics in Cognitive Science*, 3(1):74–91, 2011.

[91] Geoffrey Hinton. A practical guide to training restricted boltzmann machines. *Momentum*, 9(1):926, 2010.

[92] Yann LeCun, Sumit Chopra, Raia Hadsell, M Ranzato, and F Huang. A tutorial on energy-based learning. *Predicting structured data*, 1:0, 2006.

[93] John J Hopfield. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences*, 79(8):2554–2558, 1982.

[94] Yuichiro Anzai. *Pattern Recognition & Machine Learning*. Elsevier, 2012.

[95] Foursquare application. `http://www.foursquare.com/`.

[96] 5 boroughs: Compare and contrast. `http://www.newyork.com/articles/real-estate/5-boroughs-compare-and-contrast-17601/`. Accessed: 16-01-2016.

[97] Alexander Golbraikh and Alexander Tropsha. Predictive qsar modeling based on diversity sampling of experimental datasets for the training and test set selection. *Molecular diversity*, 5(4):231–243, 2000.

[98] Yoav Freund, Robert E Schapire, et al. Experiments with a new boosting algorithm. In *Icml*, volume 96, pages 148–156, 1996.

[99] Ilya P Ioshikhes and Michael Q Zhang. Large-scale human promoter mapping using cpg islands. *Nature genetics*, 26(1):61, 2000.

[100] Loulwah AlSumait, Daniel Barbará, and Carlotta Domeniconi. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 3–12. IEEE, 2008.

[101] Lars Maaloe, Morten Arngren, and Ole Winther. Deep belief nets for topic modeling. *arXiv preprint arXiv:1501.04325*, 2015.

[102] R Arun, Venkatasubramaniyan Suresh, CE Veni Madhavan, and MN Narasimha Murthy. On finding the natural number of topics with latent dirichlet allocation: Some observations. In *KDD*. 2010.

[103] Juan Cao, Tian Xia, Jintao Li, Yongdong Zhang, and Sheng Tang. A density-based method for adaptive lda model selection. *Neurocomputing*, 72(7):1775–1781, 2009.

[104] Elias Zavitsanos, Sergios Petridis, Georgios Paliouras, and George A Vouros. Determining automatically the size of learned ontologies. In *ECAI*, 2008.

[105] Charlie Karlsson and Michael Olsson. The identification of functional regions: theory, methods, and applications. *The Annals of Regional Science*, 40(1):1–18, 2006.

[106] Jonathan Reades, Francesco Calabrese, and Carlo Ratti. Eigenplaces: analysing cities using the space–time structure of the mobile phone network. *Environment and Planning B: Planning and Design*, 36(5):824–836, 2009.

[107] Jameson L Toole, Michael Ulm, Marta C González, and Dietmar Bauer. Inferring land use from mobile phone activity. In *Proceedings of the ACM SIGKDD international workshop on urban computing*, pages 1–8. ACM, 2012.

[108] Guande Qi, Xiaolong Li, Shijian Li, Gang Pan, Zonghui Wang, and Daqing Zhang. Measuring social functions of city regions from large-scale taxi behaviors. In *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2011 IEEE International Conference on*, pages 384–388. IEEE, 2011.

[109] Yu Liu, Fahui Wang, Yu Xiao, and Song Gao. Urban land uses and traffic ?source-sink areas?: Evidence from gps-enabled taxi data in shanghai. *Landscape and Urban Planning*, 106(1):73–87, 2012.

[110] Marie-Pier Pelletier, Martin Trépanier, and Catherine Morency. Smart card data use in public transit: A literature review. *Transportation Research Part C: Emerging Technologies*, 19(4):557–568, 2011.

[111] Lun Wu, Ye Zhi, Zhengwei Sui, and Yu Liu. Intra-urban human mobility and activity transition: evidence from social media check-in data. *PloS one*, 9(5):e97010, 2014.

[112] Justin Cranshaw, Raz Schwartz, Jason I Hong, and Norman Sadeh. The livehoods project: Utilizing social media to understand the dynamics of a city. In *International AAAI Conference on Weblogs and Social Media*, page 58, 2012.

[113] Thiago H Silva, Pedro OS de Melo, Jussara M Almeida, Juliana Salles, and Antonio AF Loureiro. Visualizing the invisible image of cities. In *Green Computing and Communications (GreenCom), 2012 IEEE International Conference on*, pages 382–389. IEEE, 2012.

[114] Ye Zhi, Yu Liu, Shaowen Wang, Min Deng, Jing Gao, and Haifeng Li. Urban spatial-temporal activity structures: a new approach to inferring the intra-urban functional regions via social media check-in data. *arXiv preprint arXiv:1412.7253*, 2014.

[115] Junbo Zhang, Yu Zheng, Dekang Qi, Ruiyuan Li, Xiuwen Yi, and Tianrui Li. Predicting citywide crowd flows using deep spatio-temporal residual networks. *arXiv preprint arXiv:1701.02543*, 2017.

[116] Michael Menth, Rüdiger Martin, and Joachim Charzinski. Capacity overprovisioning for networks with resilience requirements. In *ACM SIGCOMM Computer Communication Review*, volume 36, pages 87–98. ACM, 2006.

[117] David W Scott. *Multivariate density estimation: theory, practice, and visualization*. John Wiley & Sons, 2015.

[118] Da Kuang, Jaegul Choo, and Haesun Park. Nonnegative matrix factorization for interactive topic modeling and document clustering. In *Partitional Clustering Algorithms*, pages 215–243. Springer, 2015.

[119] Akiko Aizawa. An information-theoretic perspective of tf–idf measures. *Information Processing & Management*, 39(1):45–65, 2003.

[120] David M Blei. Probabilistic topic models. *Communications of the ACM*, 55(4):77–84, 2012.

[121] Christos Boutsidis and Efstratios Gallopoulos. Svd based initialization: A head start for nonnegative matrix factorization. *Pattern Recognition*, 41(4):1350–1362, 2008.

[122] Mark Steyvers and Tom Griffiths. Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424–440, 2007.

[123] I Cisco. Cisco visual networking index: Forecast and methodology, 2011–2016. *CISCO White paper*, pages 2011–2016, 2012.

[124] Ziaul Hasan, Hamidreza Boostanimehr, and Vijay K Bhargava. Green cellular networks: A survey, some research issues and challenges. *IEEE Communications surveys & tutorials*, 13(4):524–540, 2011.

[125] Latif Ullah Khan. Performance comparison of prediction techniques for 3g cellular traffic. *International Journal of Computer Science and Network Security (IJCSNS)*, 17(2):202, 2017.

[126] Eunsung Oh, Bhaskar Krishnamachari, Xin Liu, and Zhisheng Niu. Toward dynamic energy-efficient operation of cellular network infrastructure. *IEEE Communications Magazine*, 49(6), 2011.

[127] Utpal Paul, Anand Prabhu Subramanian, Milind Madhav Buddhikot, and Samir R Das. Understanding traffic dynamics in cellular data networks. In *INFOCOM, 2011 Proceedings IEEE*, pages 882–890. IEEE, 2011.

[128] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015.

[129] Maryam M Najafabadi, Flavio Villanustre, Taghi M Khoshgoftaar, Naeem Seliya, Randall Wald, and Edin Muharemagic. Deep learning applications and challenges in big data analytics. *Journal of Big Data*, 2(1):1, 2015.

[130] David H Hubel and Torsten N Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195(1):215–243, 1968.

[131] Kunihiko Fukushima. Neocognitron–a self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *NHK*, (15):p106–115, 1981.

[132] Yann LeCun, Bernhard E Boser, John S Denker, Donnie Henderson, Richard E Howard, Wayne E Hubbard, and Lawrence D Jackel. Handwritten digit recognition with a back-propagation network. In *Advances in neural information processing systems*, pages 396–404, 1990.

[133] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998.

[134] Robert Hecht-Nielsen et al. Theory of the backpropagation neural network. *Neural Networks*, 1(Supplement-1):445–448, 1988.

[135] Wei Zhang, Kazuyoshi Itoh, Jun Tanida, and Yoshiki Ichioka. Parallel distributed processing model with local space-invariant interconnections and its optical architecture. *Applied optics*, 29(32):4790–4797, 1990.

[136] Jiuxiang Gu, Zhenhua Wang, Jason Kuen, Lianyang Ma, Amir Shahroudy, Bing Shuai, Ting Liu, Xingxing Wang, and Gang Wang. Recent advances in convolutional neural networks. *arXiv preprint arXiv:1512.07108*, 2015.

[137] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[138] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[139] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[140] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[141] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[142] Gao Huang, Zhuang Liu, Kilian Q Weinberger, and Laurens van der Maaten. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2016.

[143] Yanhua Yu, Meina Song, Yu Fu, and Junde Song. Traffic prediction in 3g mobile networks based on multifractal exploration. *Tsinghua Science and Technology*, 18(4):398–405, 2013.

[144] Konstantina Papagiannaki, Nina Taft, Zhi-Li Zhang, and Christophe Diot. Long-term forecasting of internet backbone traffic. *IEEE transactions on neural networks*, 16(5):1110–1124, 2005.

[145] Nayera Sadek and Alireza Khotanzad. Multi-scale high-speed network traffic prediction using k-factor gegenbauer arma model. In *Communications, 2004 IEEE International Conference on*, volume 4, pages 2148–2152. IEEE, 2004.

[146] Bo Zhou, Dan He, Zhili Sun, and W Hock Ng. Network traffic modeling and prediction with arima/garch. In *Proc. of HET-NETs Conference*, pages 1–10, 2005.

[147] M Zubair Shafiq, Lusheng Ji, Alex X Liu, and Jia Wang. Characterizing and modeling internet traffic dynamics of cellular devices. In *Proceedings of the ACM SIGMETRICS joint international conference on Measurement and modeling of computer systems*, pages 305–316. ACM, 2011.

[148] Zulfiquar Sayeed, Qi Liao, Dave Faucher, Ed Grinshpun, and Sameer Sharma. Cloud analytics for wireless metric prediction-framework and performance. In *Cloud Computing (CLOUD), 2015 IEEE 8th International Conference on*, pages 995–998. IEEE, 2015.

[149] Jun Yao, Salil S Kanhere, and Mahbub Hassan. Improving qos in high-speed mobility using bandwidth maps. *IEEE Transactions on Mobile Computing*, 11(4):603–617, 2012.

[150] Haakon Riiser, Paul Vigmostad, Carsten Griwodz, and Pål Halvorsen. Commute path bandwidth traces from 3g networks: analysis and applications. In *Proceedings of the 4th ACM Multimedia Systems Conference*, pages 114–118. ACM, 2013.

[151] Haytham Assem and Declan O'Sullivan. Discovering new socio-demographic regional patterns in cities. In *Proceedings of the 9th ACM SIGSPATIAL Workshop on Location-based Social Networks*, page 1. ACM, 2016.

[152] Yann A LeCun, Léon Bottou, Genevieve B Orr, and Klaus-Robert Müller. Efficient backprop. In *Neural networks: Tricks of the trade*, pages 9–48. Springer, 2012.

[153] Michael Mathieu, Camille Couprie, and Yann LeCun. Deep multi-scale video prediction beyond mean square error. *arXiv preprint arXiv:1511.05440*, 2015.

[154] Viren Jain, Joseph F Murray, Fabian Roth, Srinivas Turaga, Valentin Zhigulin, Kevin L Briggman, Moritz N Helmstaedter, Winfried Denk, and H Sebastian Seung. Supervised learning of image restoration with convolutional networks. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pages 1–8. IEEE, 2007.

[155] Spyros Makridakis, Steven C Wheelwright, and Rob J Hyndman. *Forecasting methods and applications*. John wiley & sons, 2008.

[156] JP Wu and Shuony Wei. *Time series analysis*. Hunan Science and Technology Press, ChangSha, 1989.

[157] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[158] Tomas Mikolov, Martin Karafiát, Lukas Burget, Jan Cernockỳ, and Sanjeev Khudanpur. Recurrent neural network based language model. In *Interspeech*, volume 2, page 3, 2010.

[159] Amit Sahu. Survey of reasoning using neural networks. *arXiv preprint arXiv:1702.06186*, 2017.

[160] Diane Tang and Mary Baker. Analysis of a local-area wireless network. In *Proceedings of the 6th annual international conference on Mobile computing and networking*, pages 1–10. ACM, 2000.

[161] Kevin P Murphy. *Machine learning: a probabilistic perspective*. MIT press, 2012.

[162] Francois Chollet. Deep learning library for python. runs on tensorflow, theano, or cntk. `https://github.com/fchollet/keras`. [Online; accessed 09-August-2017].

[163] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[164] Rong-En Fan, Kai-Wei Chang, Cho-Jui Hsieh, Xiang-Rui Wang, and Chih-Jen Lin. Liblinear: A library for large linear classification. *Journal of machine learning research*, 9(Aug):1871–1874, 2008.

[165] Shi Yu, Léon-Charles Tranchevent, Bart De Moor, and Yves Moreau. *Kernel-based data fusion for machine learning*. Springer, 2013.

[166] Hua Wang, Feiping Nie, and Heng Huang. Multi-view clustering and feature learning via structured sparsity. In *International Conference on Machine Learning*, pages 352–360, 2013.

[167] Yu Zheng. Methodologies for cross-domain data fusion: An overview. *IEEE transactions on big data*, 1(1):16–34, 2015.

[168] Sotiris B Kotsiantis, I Zaharakis, and P Pintelas. Supervised machine learning: A review of classification techniques, 2007.

[169] Cecilia Ovesdotter Alm, Dan Roth, and Richard Sproat. Emotions from text: machine learning for text-based emotion prediction. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 579–586. Association for Computational Linguistics, 2005.

[170] Mao Ye, Peifeng Yin, and Wang-Chien Lee. Location recommendation for location-based social networks. In *Proceedings of the 18th SIGSPATIAL international conference on advances in geographic information systems*, pages 458–461. ACM, 2010.

[171] Jie Bao, Yu Zheng, and Mohamed F Mokbel. Location-based and preference-aware recommendation using sparse geo-social networking data. In *Proceedings of the 20th international conference on advances in geographic information systems*, pages 199–208. ACM, 2012.

[172] Jing Yuan, Yu Zheng, and Xing Xie. Discovering regions of different functions in a city using human mobility and pois. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 186–194. ACM, 2012.

[173] Tim Kindberg, Matthew Chalmers, and Eric Paulos. Guest editors' introduction: Urban computing. *IEEE Pervasive Computing*, 6(3):18–20, 2007.

[174] UCI Machine Learning Repository. `http://archive.ics.uci.edu/ml/datasets.html`, 2015. [Online; accessed 19-December-2015].

[175] Rich Caruana and Alexandru Niculescu-Mizil. An empirical comparison of supervised learning algorithms. In *Proceedings of the 23rd international conference on Machine learning*, pages 161–168. ACM, 2006.