Titre: Title:	Mobility Prediction in Self-Organizing Cellular Networks
Auteur: Author:	Nasrin Bahra
Date:	2022
Type:	Mémoire ou thèse / Dissertation or Thesis
Référence: Citation:	Bahra, N. (2022). Mobility Prediction in Self-Organizing Cellular Networks [Thèse de doctorat, Polytechnique Montréal]. PolyPublie. https://publications.polymtl.ca/10354/

URL de PolyPublie: PolyPublie URL:	https://publications.polymtl.ca/10354/
Directeurs de recherche: Advisors:	Samuel Pierre
Programme: Program:	Génie informatique

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Mobility Prediction in Self-organizing Cellular Networks

NASRIN BAHRA
Département de génie informatique et génie logiciel

Thèse présentée en vue de l'obtention du diplôme de *Philosophiæ Doctor*
Génie informatique

Avril 2022

POLYTECHNIQUE MONTRÉAL
affiliée à l'Université de Montréal

Cette thèse intitulée :

Mobility Prediction in Self-organizing Cellular Networks

présentée par **Nasrin BAHRA**
en vue de l'obtention du diplôme de *Philosophiæ Doctor*
a été dûment acceptée par le jury d'examen constitué de :

Foutse KHOMH, président
Samuel PIERRE, membre et directeur de recherche
Alejandro QUINTERO, membre
Abderrezak RACHEDI, membre externe

DEDICATION

To my parents Tahereh and Mohammad,
for their unconditional love and faith in me,
couldn't have done it without you...

To my husband Arash,
for his immeasurable love and support,
for your constant love, patience and friendship...

To my sisters Maryam and Mina,
you are my best friends...

To my niece Anita, and my nephew Aria,
you make my heart smile...

ACKNOWLEDGEMENTS

This thesis is completed with the kind help of many persons. I would like to express my sincere appreciation to all of them for their kind support throughout my PhD journey.

First and foremost, it is a real pleasure to express my genuine appreciation to my research director Professor Samuel Pierre for his respectable and permanent support during these years. Under his supervision, I learnt numerous invaluable things from both life and research aspects. His kind moral and scientific support, immense knowledge, patience and dedication is a true example for me. I cannot imagine completing this path without his continues guidance and I would like to warmly thank him for giving me this golden opportunity.

I am also sincerely thankful to my PhD committee members whose work I regard so highly. I would like to thank Professor Alejandro Quintero, Professor Foutse Khomh and Professor Abderrezak Rachedi for their insightful comments and constructive discussions.

I would like to acknowledge and give my warmest thanks to my colleagues at Mobile Computing and Network Research Laboratory (LARIM) for their helpful feedback and comments on my research. Thanks to Sanaz, Farnoush, Franjieh, Paran, Eric, Marieme, Lamia, Amir, Sepehr, Olson, Cloudy, Loic. Their kindness and collaboration make our work environment unforgettable for me.

I am eternally indebted to my parents, Tahereh and Mohammad for their endless love and support. I cannot thank them enough for the sacrifices that they have made for me. I would never have made it here without them. I am also very grateful to my lovely sisters, Maryam and Mina for their unconditional love and support. Finally, my deepest heartfelt thanks to my caring, loving and supportive husband, Arash. His constant encouragement was a valuable help for me throughout this journey. He is always a great source of inspiration and happiness for me.

RÉSUMÉ

La prédiction de la mobilité et la localisation en tant qu'éléments importants de la gestion de la mobilité jouent un rôle de premier plan dans les réseaux cellulaires auto-organisés. La prédiction de la mobilité est un catalyseur clé prometteur pour la gestion intelligente de la mobilité des futurs réseaux prévisibles dans lesquels les ressources disponibles sont critique limité. Les techniques de localisation offrent des opportunités en or pour un réseau intelligent gestion en termes de meilleure allocation des ressources, d'amélioration de la qualité de service (QoS) et réduction de la latence de transfert. De plus, les informations sur l'emplacement futur de l'utilisateur permettent à de nombreux applications et services mobiles de localisation. Cependant, il existe un certain nombre de problèmes non résolus dans ce domaine qui n'a pas encore été abordé par les travaux existants dans la littérature.

Dans cette thèse, nous passons en revue certains des travaux récents connexes et résumons les défis ouverts de perdition de mobilité dans les réseaux cellulaires. Nous proposons une nouvelle approche de prédiction de la mobilité qui est principalement basé sur le concept populaire de réseaux de neurones profonds (DNN). Notre travail non seulement se concentre sur l'amélioration des performances de précision des prédictions, mais répond également aux exigences de les limitations de complexité temporelle dans le contexte des futurs réseaux mobiles. Les approches proposées tirent pleinement parti des nombreux mérites des DNN afin de fournir une prédiction à long terme et de modéliser la mobilité de l'utilisateur même avec des motifs complexes et des irrégularités avec un temps d'exécution minimum. À cette fin, cette thèse met en évidence le besoin urgent de jeter les bases d'une gestion intelligente des réseaux. Pour atteindre cet objectif, notre travail est composé de trois phases principales.

Dans la première phase, nous proposons un modèle de mobilité des utilisateurs afin de prédire la trajectoire future des utilisateurs. Dans ce modèle, la première étape consiste à éliminer les informations inutiles de la trajectoire brute de l'utilisateur avant d'en extraire les motifs répétitifs. Comme étape potentielle de prétraitement, nous proposons d'appliquer des techniques de simplification des lignes à l'historique des mouvements de l'utilisateur et de ne conserver que les données pertinentes. Ensuite, en ayant la trajectoire modifiée de l'utilisateur, nous exploitons des variantes de réseaux neuronaux récurrents (RNN) pour apprendre le comportement de mobilité de l'utilisateur et prédire la trajectoire future. Les résultats de la simulation montrent une réduction considérable du temps d'exécution tout en réduisant l'erreur de prédiction. Dans le meilleur des cas, le temps d'exécution et les performances du modèle s'améliorent respectivement de 79 % et 31 %.

Dans la deuxième phase, nous proposons un modèle de prédiction de trajectoire bidirectionnelle (BTPM) pour améliorer le niveau de performance du modèle de mobilité. Dans le modèle BTPM, nous étudions les corrélations sous-jacentes de la trajectoire de l'utilisateur dans les deux sens (avant et arrière) à l'aide d'unités récurrentes à déclenchement (GRU). Cet apprentissage bidirectionnel nous donne l'avantage distinct d'avoir une compréhension plus profonde du comportement de mobilité de l'utilisateur. De plus, nous introduisons une nouvelle phase de préparation des données spécifiquement pour la prédiction de la trajectoire afin d'obtenir non seulement un prédicteur de mobilité très précis mais aussi de réduire le niveau de bruit des mesures et la complexité temporelle. Les résultats expérimentaux indiquent que le modèle proposé atteint une erreur de prédiction de 0,014 et une réduction de la complexité temporelle jusqu'à 97 %.

Enfin, nous exploitons les avantages potentiels de la prédiction de la trajectoire de l'utilisateur pour la gestion du handover (HO). Nous proposons une approche hybride de prédiction de la mobilité des utilisateurs pour la gestion du handover dans les réseaux mobiles. L'approche proposée comporte deux parties principales : (1) la prédiction de la mobilité de l'utilisateur et (2) la procédure de transfert prédictive. Dans la première partie, nous analysons l'historique des mouvements de l'utilisateur en utilisant un modèle statistique. Nous utilisons un modèle d'autorégression vectorielle (VAR) pour dériver les dépendances des échantillons d'entrée sur la base des étapes temporelles précédentes. Ensuite, nous introduisons les informations obtenues dans un réseau neuronal basé sur le GRU pour apprendre le modèle de mouvement de l'utilisateur. Dans la deuxième partie, après avoir prédit les emplacements futurs de l'utilisateur, nous réduisons le nombre de messages de signalisation nécessaires pour exécuter une procédure de transfert. Pour évaluer l'impact de la prédiction de la mobilité sur la gestion du transfert, nous étudions les coûts de traitement et de transmission du transfert. Les résultats montrent que les coûts de traitement et de transmission du transfert s'améliorent respectivement de 57,14 % et 28,01 % pour un transfert vertical.

ABSTRACT

Mobility prediction and localization as important parts of mobility management have leading roles in self-organizing cellular networks. Mobility prediction is a promising key enabler for the intelligent mobility management of foreseeable future networks in which available resources are critically limited. Localization techniques provide golden opportunities for smart network management in terms of better resource allocation, quality of service (QoS) improvement and handoff latency reduction. Moreover, the information of user's future location empower many location-aware mobile applications and services. However, there are a number of unresolved issues in this field that has not been addressed yet by the existing works in the literature.

In this thesis, we review some of the recent related works and summarize the open challenges of mobility perdition in cellular networks. We propose novel mobility prediction approaches that are mainly based on the popular concept of deep neural networks (DNNs). Our work not only focuses on the prediction accuracy performance improvement but also meets the demands of time complexity limitations in the context of future mobile networks. The proposed approaches takes full advantage of numerous merits of DNNs in order to provide a long-term prediction and model user's mobility even with complex patterns and irregularity with minimum execution time. To this end, this thesis highlights the urgent need for laying the foundation for providing intelligent network management. To fulfill this objective, our work is composed of three main phases.

In the first phase, we propose a user mobility model to predict the future trajectory of users. In this model the primary step is to eliminate the unnecessary information from the user raw trajectory before extracting the repetitive patterns. As a potential preprocessing step, we propose to apply line simplification techniques to the user movement history and keep only the relevant data. Next, having the modified user trajectory, we exploit recurrent neural network (RNN) variants to learn the user mobility behavior and predict the future trajectory. Simulation results show a huge reduction in execution time while reducing the prediction error. For the best case, execution time and model performance respectively improve by 79% and 31%.

In the second phase, we propose a bidirectional trajectory prediction model (BTPM) to enhance the performance level of the mobility model. In BTPM, we investigate the underlying correlations in user trajectory in both forward and backward directions using gated recurrent units (GRUs). This bidirectional learning gives us a distinct advantage of having a deeper

understanding of user mobility behavior. Moreover, we introduce a new data preparation phase specifically for trajectory prediction to obtain not only a highly accurate mobility predictor but also reduce measurement noise level and time complexity. Experimental results indicate that the proposed model achieves a prediction error of 0.014 and time complexity reduction up to 97%.

Lastly, we exploit the potential benefits of user trajectory prediction for handover (HO) management. We introduce a hybrid method for user trajectory prediction for handover management in mobile networks. The proposed approach has two main parts including (1) User mobility prediction and (2) Predictive handover procedure. In the first part, we analyze the user movement history using a statistical model. We employ a vector autoregression (VAR) model to derive input samples dependencies based on the previous time steps. Then, we feed the obtained information to a GRU-based neural network to learn the user movement pattern. In the second part, having predicted the user future locations, we reduce the number of required signaling messages for performing a handover procedure. To evaluate the impact of mobility prediction on handover management, we study the HO processing and transmission costs. Results prove that handover-related costs including processing and transmission costs improve respectively by 57.14% and 28.01% for a vertical HO.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS AND ACRONYMS

SON	Self-Organized network
OSI	Open Systems Interconnection
RNN	Recurrent Neural Network
GRU	Gated Recurrent Unit
BiGRU	Bidirectional Gated Recurrent Unit
DNN	Deep Neural Network
LSTM	Long Short-Term Memory
IoT	Internet of Things
QoS	Quality of Service
LBSN	Location-Based Social Network
POI	Point-of-Interest
CDR	Call Detail Records
HMM	Hidden Markov Model
R-W	Reumann-Witkam
V-W	Visvalingam-Whyatt
D-P	Douglas-Peucker
HO	Handover
BTPM	Bidirectional Trajectory Prediction Model
MSE	Mean Square Error
MAE	Mean Absolute Error
RMSE	Root Mean Square Error
UE	User Equipment
NG-RAN	NG Radio Access Network
SMF	Session Management Function
UTRAN	Open Street Map
UMTS	Universal Mobile Telecommunications System
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
PGW	Packet Gateway
SGSN	Serving GRPS Support Node
RNC	Radio Network Controller
PCRF	Policy and Charging Rules Function
S-GW	Serving Gateway
SeMMu	SDN enabled Mobility Management unit

OSM	Open Street Map
LR	Linear Regression
SVR	Support Vector Regression
LR	Linear Regression
KNN	K-Nearest Neighbour
G-means	Gaussian means
Lon	Longitude
Lat	Latitude
LAC	Location Area Code
MCC	Mobile Country Codes
MNC	Mobile Network Code
ANN	Artificial Neural Network
VANET	Vehicular Ad hoc Network
ST-RNN	Spatial Temporal Recurrent Neural Network
HMTP	Hidden Markov-based Trajectory Prediction
MIRACLE	Mobility prediction inside a coverage hole
CLSTERS	Challenging Localization Situation-aimed Trajectory Error Reduction System
EM	Expectation Maximization
MLP	Multilayer Perceptron
DAMP	Destination and Mobility Path prediction
DPM	Destination Prediction Model
PPM	Path Prediction Model
CEPR	Collaborative Exploration and Periodically Returning
PPM	Path Prediction Model
GTS-LP	Geographic-Temporal-Semantic-based Location Prediction
MGDPre	Mobility Gradient-based Destination Prediction
DPBMA	Distributed Prediction with Bandwidth Management Prediction
SLWE	Principal Component Analysis
PCA	Principal Component Analysis
AMF	Access and Mobility Management Function
UPF	User Plane Function
RAT	Radio Access Technology
EPS	Evolved Packet System
5G NGC	5G Next Generation Core
SGD	Stochastic Gradient Decent

BS	Base Station
HHO	Horizontal HO
VHO	Vertical HO
GSM	Global System for Mobile Communications
LTE	Long-Term Evolution

LIST OF APPENDICES

CHAPTER 1 INTRODUCTION

Mobility prediction and localization as important parts of mobility management have leading roles in self-organizing cellular networks. Mobility prediction is a promising key enabler for the intelligent mobility management of foreseeable future networks in which available resources are critically limited. A deep understanding of network traffic behavior is of crucial importance in today's rapidly growing mobile networks. Bandwidth restriction, high-speed packet transmission and communication reliability pose serious challenges in the network management. If the network can predict the mobile user's next location, based on its prior knowledge, it can better allocate resources and provide higher quality of service (QoS) for users. In mobile networks, acceptable QoS provisioning is quite challenging since mobility is an inherited characteristic in them and may cause several problems such as high signaling load. Therefore, a high-accuracy mobility model prediction has a profound impact on mobile networks management and user satisfaction.

Self-organizing cellular network (SON) is an emerging paradigm that is based on adoptive and independent network management. It is able to learn from past experiences and improve the network's performance based on the prior knowledge. The core concept of self-organizing networks consist of self-configuration, self-optimization and self-healing. Mobility management is a part of self-optimization function in self-organizing networking that can address the aforementioned issues [4].

Location awareness can be exploited in many ways in order to meet the increasing demands in the context of internet of things (IoT). Some of the potential research areas that can significantly benefit from location awareness are: resource allocation techniques, optimized handoff decision, call admission control optimization, bandwidth reservation, routing mechanisms and provision of high QoS. In particular, when a mobile node moves between different access points with an active session, seamless connectivity without interruption is essential while user is moving. Location awareness can make handoff process transparent to mobile users in dense networks and effectively eliminate the problem of QoS degradation [5]. Moreover, location information serves useful purposes for 5G networks in terms of energy consumption and latency reduction. Massive increase in the number of mobile devices and connected objects stress the needs for lower latency and high data rates. Context information and in particular location information are enabling technologies in order to satisfy 5G networks requirements. In this context, anticipating the mobility pattern of highly mobile users enables the provision of various services, especially location-based services, for users. This anticipatory mobile networking aims at dynamic network planning based on reliable and accurate predictions.

In this proposal, a broad overview of different mobility prediction techniques in self-organizing networks is provided and the potential drawbacks of the existing works are explained. Our work aims at proposing adoptive methods for mobility prediction that can address some of the issues in this field.

The rest of this introductory chapter is organized as follows. Section 1.1 provides an overview of basic concepts and definitions. Some open problems in this field is provided in Section 1.2. Section 1.3 and 1.4 respectively present our research objectives and research contributions. Finally, thesis outline is provided in Section 1.5.

1.1 Basic concepts and definitions

In this section, an overview of basic definitions and concepts are provided. First, we elaborate on the concept of mobility management and in particular mobility prediction and user mobility data types. Then, we present the concepts of widely used technologies in mobility prediction approaches.

1.1.1 Overview of the mobility management in self-organizing networks

The distinguishing feature of self-organizing networks (SONs) is the fact that they can dynamically adopt to possible changes in the network that aims at reducing manual efforts. SON is a scalable autonomous network that can cope well with performance degradation using the information acquired from the past experiences. In other words, these networks are able to independently interact with the environment and make decisions; moreover, they can learn how to enhance their overall performance based on the previous actions. The main advantages of SON is installation cost and time reduction, network performance enhancement and user experience improvement.

Self-organizing mobile networks can be classified into three main groups: self-configuration, self-optimization and self-healing [4]. Self-configuration procedure is mainly responsible for any probable configuration process that is needed for the network in order to operate properly. For example, when a new base station is added to the network self-configuration functions perform the IP configuration and neighbour cell list configuration. Self-optimization procedure is basically about constant optimization of network parameters aiming at network performance improvement. This optimization concerns many fields in the network including handover optimization, resource allocation optimization, mobility optimization, load balanc-

Table 1.1 Self-organizing functions and their use cases.

SON Functions	Use Cases
Self-configuration	Physical cell ID configuration- Neighbor cell list- Radio parameters
Self-optimization	Mobility - Handover- Resource optimization- Call admission control
Self-healing	Failure detection- Cell outage management- Failure classification and diagnose

ing and energy consumption optimization. In self-healing, the main purpose is to deal with any unexpected software or hardware failures in the network. Self-healing functions are responsible for both failure detection and also helping the network to recover faster and can function properly. Some of the popular use cases for each SON functions are summarized in Table 1.1.

Mobility management is a subsection of self-optimization functions. Self-optimization in terms of mobility management in mobile cellular networks is mainly based on the machine learning and deep learning techniques. The main idea is to estimate user's movement using machine learning approaches for a better network management. Learning techniques extract mobility patterns from user mobility history that can be useful for future network optimization [4].

1.1.1.1 Mobility prediction

User mobility is completely random and unplanned and is dependant on the user mobility behavior. However, it has been proven that there is a certain level of potential predictability hidden in user movement history [6]. Predicting user mobility is based on the available user movement history from previous time steps. In general, almost all the methods in the mobility prediction concept can fit into two main categories [7]:

1. User future path/trajectory prediction: The main objective in this category is to predict the user movement trajectory for a predefined number of time steps ahead. It is a regression task that predicts the geographical location of the user in the next time step. The prediction is the next geographical location point of the user (e.g., longitude and latitude). One popular example is user trajectory prediction (e.g, (lon1, lat1), (lon2, lat2), (lon3, lat3), ...).

2. User next location prediction: It is a classification task that predicts the next places that user will visit in the future steps. In the dataset, each location has an specific location id. The prediction is the next location id of the user. One popular example of its application is point-of-interest recommendation methods (e.g, restaurant, park, gas

station, ...).

The scope of our work fits to the first category and mainly concentrates on user trajectory prediction. Our primary focus is on modeling user mobility behavior based on the knowledge of previous trajectories and predicting the future trajectory of the user. There are several types of mobility data that can be effectively exploited in order to extract insights from existing patterns. In the following, we define user trajectory and trajectory prediction terms and ellaborate on mobility data types in the next section.

- *Trajectory Definition:* A trajectory for the user i is represented as $T_{useri} = (L_{t-n}, ..., L_{t-2}, L_{t-1}, L_t)$ which is a sequence of time-stamped location points. We denote L_t as $L_t = (x_t, y_t)$ where x_t and y_t provide the geographical information of the user location at time t (i.e., longitude and latitude).

- *Trajectory Prediction:* Given a trajectory of a user for the previuos n time steps (T_{user}), we want to predict the next n time steps ($\tilde{T}_{user}$). Where $T_{user} = (L_{t-n}, ..., L_{t-1}, L_t)$ and $\tilde{T}_{user} = (L_{t+1}, ..., L_{t+n})$.

Figure 1.1 provides an overall overview of the mobility prediction framework [8]. This structure shows the main five parts of mobility prediction procedure (data sources, collected information, mobility predictors, prediction results and performance metrics) and some examples of its application. This figure shows different mobility data types that generate several types of information for the input of the prediction algorithms. There are several prediction outputs based on the type of information that has been used as collected information that can be evaluated using several metrics. Details regarding these parts are given in the following chapters.

1.1.1.2 Different types of mobility data sources

In the area of mobility and understanding the existing patterns in order to predict future locations, the primary step is to have the past mobility data of a user. This mobility data can be generated from different types of mobility data sources. In this section, four types of mobility data produced by different tracking technologies are explained, namely Global positioning system (GPS) data, Wi-Fi data, cellular-based data and location-based social networks data [9].

GPS data: The great advantages of using GPS as mobility data are the high level of prediction accuracy and global outdoor environment coverage. However, this type of data falls short in terms of optimized energy consumption due to the continuous location requests.

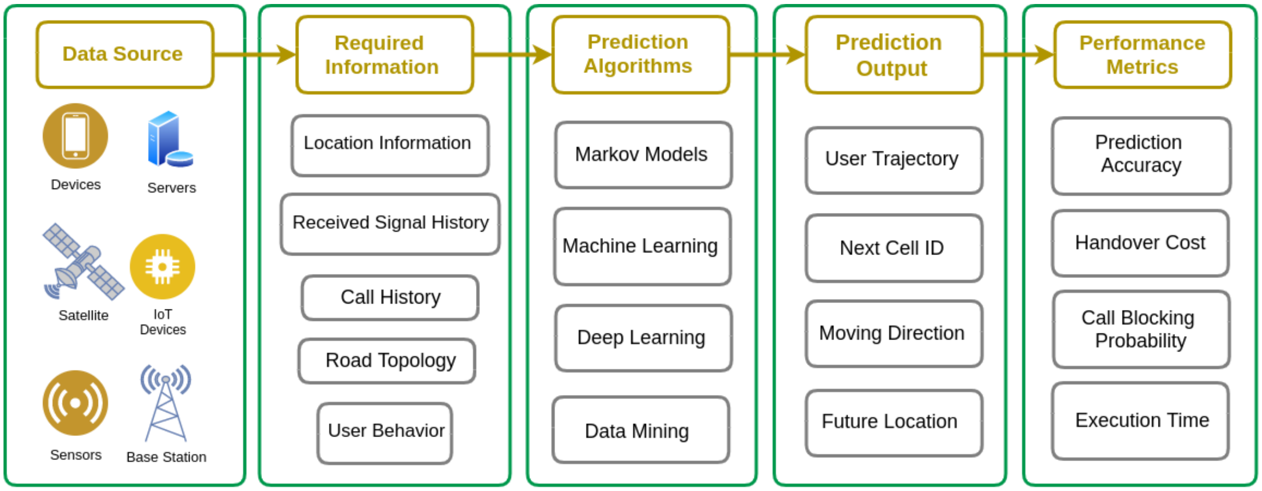


Figure 1.1 The overall overview of the mobility prediction framework.

Wi-Fi data: This type of data contains the record of Wi-Fi access points that user attached to them while moving. The location accuracy in this case is much lower than using GPS data. This is because the coverage area of an access point is usually a hundred meters, it means when a user attached to an access point its location is in a 100-meter-radius area and we do not know the exact place while for GPS data the radius is almost 10 meters. On the other hand, using Wi-Fi data for the mobility model needs lower energy compared to using GPS data.

Cellular network data: In this type, mobility data contains the record of cells that user passes while moving and it is based on the base stations that user's device attached to them while moving. Location accuracy using cellular data is strongly critical since cell coverage in a cellular network may vary from 200 meters up to several hundred meters and kilometers depending on the cell type, predicting next cell with a range of for example 500 meters is not desirable especially compared to the location accuracy that can be achieved using GPS data. On the other hand, energy consumption for this type is the lowest due to the fact that there is no need for any other additional process to collect the data.

Data from Location-based social networks (LBSN): This group of works take advantage of the rapidly growing popularity of smart devices and location-aware applications between users. LBSN is used for sharing locations of everyday activities and collects visited locations records that can open up plenty of opportunities for different location-aware applications. The location data accuracy in this case is closely related to the user's honesty since user provides the information about his/her location directly.

1.1.2 The concepts of widely used technologies in mobility prediction approaches

There are several technologies and approaches that have been deployed to address mobility prediction in mobile networks such as stochastic models and learning algorithms. Several stochastic models such as Markov models and autoregressive models have been introduced in this field. Moreover, learning approaches based on machine learning techniques and deep learning methods have gained a huge attraction in this area as well. Among these approaches Markov models and neural networks are the most popular techniques that could achieve an acceptable performance.

1.1.2.1 Neural networks

Artificial neural networks (ANNs) also commonly known as neural networks (NNs) are a popular subset of machine learning algorithms which is intuitively inspired by mimicking human brain's behavior. Neural network as a promising computing system is composed of layers of neurons that can learn complex patterns. A simple neural network model is shown in Figure 1.2. In this figure, layer 1 is the input layer, layer 3 is the output layer of the neural network and the middle layer is called hidden layer. When the number of hidden layers increases, a deep neural network (DNN) is obtained. DNNs are able to learn complex tasks and patterns in model training phase.

There are two phases regarding neural networks: training phase and testing phase. In the training phase, a set of training samples as the input of neural network is used to find the optimum values for the network parameters (weights) with the purpose of minimizing the cost function. Cost function represents the difference between the predicted output and the actual output. In the testing phase, network parameters that are obtained from the training phase are used for a set of unseen samples (testing set). The accuracy of the model is evaluated based on the errors of the predictions for the unseen data [10].

More precisely, for a set of d training samples $(x^1, y^1), (x^2, y^2) \ldots, (x^d, y^d)$, gradient decent is exploited to train the neural networks. The cost function for each training example is defined as:

$$J(W, x, y) = \frac{1}{2} \| h_w(x) - y \|^2, \tag{1.1}$$

where W denotes the network parameters, also h and y are respectively the predicted and actual outputs of the network. In order to minimize the cost function we need to calculate the partial derivatives of cost function in respect to the network weights which is usually done using back-propagation algorithm.

Two popular types of neural networks are convolution neural networks (CNNs) and Recurrent

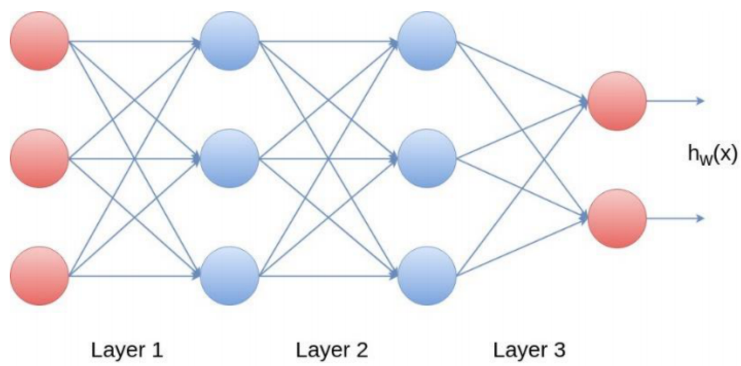


Figure 1.2 Neural network model.

neural networks (RNNs) that can potentially be deployed in wide range of applications. CNNs are mostly used to learn spatial features from image dataset. RNNs are a potential tool to investigate temporal dependency in sequences. RNN and its variant networks can potentially be utilized in our work.

1.1.2.2 Markov models

Markov models are mostly used when we have a sequential data in which there is a correlation between the data samples. Using the product rule, the joint distribution for a data sequence can be expressed as follows:

$$p(o_1, o_2, \ldots, o_N) = \prod_{i=1}^{N} p(o_i | o_1, \ldots, o_{i-1}) \tag{1.2}$$

where $o_1, o_2, \ldots, o_N$, $i = 1, .., N$ is the sequence of observations. It shows that each observation is dependent on all the previous observations. If we consider that each observation is only dependent on the last observation and is independent of other observations, we have a

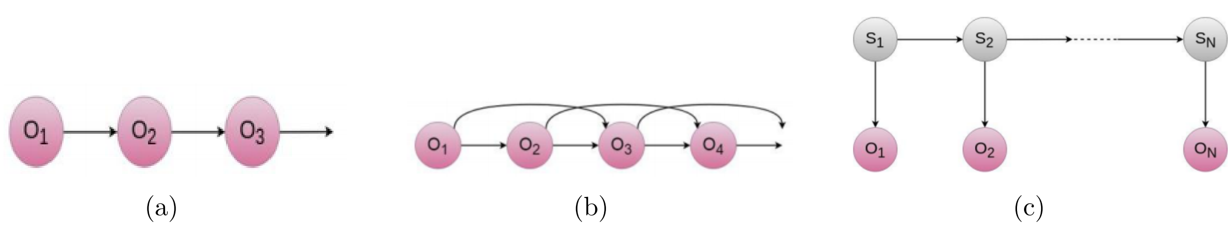


Figure 1.3 Different types of Markov models (a) First-order Markov model, (b) Second-order Markov model and (c) Hidden Markov model.

first-order Markov chain with the following joint distribution:

$$p(o_1, o_2, \ldots, o_N) = p(o_1) \prod_{i=2}^{N} p(o_i|o_{i-1}) \tag{1.3}$$

Figure 1.3(a) shows an example of a first-order Markov chain that is a simple model to treat sequential data [11]. However, ignoring the data correlation of past observations is not a realistic assumption in most cases and has a noticeable disruptive impact on the prediction results. To this end, second-order Markov is introduced in which the prediction is based on the last two previous observations as is shown in Figure 1.3(b). The joint distribution of the second-order Markov is given by

$$p(o_1, o_2, \ldots, o_N) = p(o_1)p(o_2|o_1) \prod_{i=3}^{N} p(o_i|o_{i-1}, o_{i-2}) \tag{1.4}$$

There is an important trade-off in Markov models that needs to be given a careful consideration. Higher orders of Markov chain results is a better prediction accuracy but also leads to an undesirable exponential growth in the number of model parameters. For an Mth-order Markov chain with d states, $d^{M-1}(d-1)$ parameters are needed. In order to have a model with limited number of parameters while not being restricted to the Markov assumption, we need to add latent variables to our model [11]. Figure 1.3(c) shows such a model. In this model,$s_1, s_2, \ldots, s_N$are the latent variables for N observations. The corresponding joint distribution of this model can be defined by

$$p(o_1, o_2, \ldots, o_N, s_1, s_2, \ldots, s_N) = p(s_1)[\prod_{i=2}^{N} p(s_i|s_{i-1})] \prod_{i=1}^{N} p(o_i|s_i) \tag{1.5}$$

The graphical model in Figure 3 is known as state space model, in the case where latent variables are discrete this graphical structure is called hidden Markov model (HMM). HMM is a popular technique that is exploited in many different applications such as speech recognition and natural language modeling. This model also is widely used in mobility prediction techniques in cellular networks, some of this Markov-based predictions are summarized in the next sections.

1.2 Open problems

The significance of anticipatory and self-organizing mobile networking and in particular mobility prediction and its advantages were explained in the previous parts. However, this promising field has some serious challenges that has not been addressed yet by the existing works in the literature. Thus, in order to benefit from the prior knowledge of the user location and improve the level of comfort for network management we need to pay serious attention to the remaining unsolved challenges in this field.

Some of the important unsolved open problems are:

- Prediction reliability: The accuracy of the predictors should be carefully controlled to ensure a reliable prediction. Even more major problems may arise when inaccurate predictions are deployed for the network management compared with the case with no prediction.

- Long-term prediction: Mostly, the reported accuracy for mobility prediction is acceptable for a short-term prediction. Anticipating the mobility patterns could not have beneficial impacts on optimization algorithms if it is too short. Therefore, it is essential to take the trade-off between accuracy and prediction length into consideration. Prediction errors make huge impacts on having a robust algorithm.

- Irregularity: Usually, mobility prediction techniques result in desirable prediction accuracy for predictable users. In other words, they are best suited for regular users that have repetitive habits. A generic model is required that can cope well with the irregularity of unpredictable users.

- Inequality of data access: In the concept of mobility prediction, prior knowledge and the amount of data that a researcher could have access is very important. Training a mobility model with a large dataset can remarkably improve the performance of the model. Unfortunately, in spite of the massive explosion of data generation, a limited number of researchers have access to this so called “Big data” (for example it is available

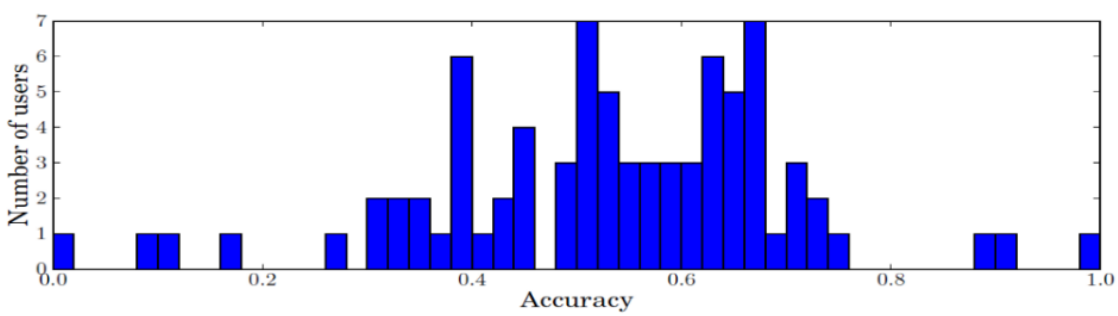


Figure 1.4 The variance of user predictability for a dataset [3].

for those who works in Google or Facebook labs). Hence, this inequality of data access and computational power holds an unfair competition between researchers and also makes the research reproducibility process very hard for a group of researchers with limited data access.

- Prior knowledge dependency: Mostly, predictions rely on different types of information about the users such as user's visited locations with its time and date, cell ID and also other contextual information. This prior knowledge dependency may raise the problems of energy consumption, device memory, high delay and in some cases lack of available data.

- Limited number of datasets: There are limited number of datasets in this field and collecting data for different scenarios is a challenging task.

- Trade-off between the model complexity and accuracy: User mobility is highly correlated to different variables such as spatial or temporal variables, taking all the variables into consideration for designing the mobility model leads to more accurate but also more complex model. We need an adoptive mobility prediction model that can dynamically fit to different scenarios with different priorities in terms of model accuracy and complexity.

- Sparse data: In the cases that available data is sparse, the variance of predictability is significantly high. Figure 1.4 depicts accuracy histogram of a data set [37]. As it is shown in the figure the variance of user predictability is very high since prediction accuracy is noticeably different for predictable and unpredictable users. The main reason of this poor performance is the lack of enough data for individual user mobility model that may also cause overfitting due to the few number of samples in the training process.

- Practical use cases: Despite the fact that there are a wide range of proposed mobility prediction models in the literature, most of these models has not been adopted for practical cases. The reason of this issue may lie in the fact that we need new protocols for users and mobile service providers in user-centric networks. Based on this new interaction protocols, mobile service providers can offer a wide range of location-aware services for users in a user-friendly network and also respect users privacy.

1.3 Research objectives

Our work aims to bridge the gap between achieving an ideal mobility prediction model in theory and practice with addressing some of the existing issues. This section presents the global objective of this thesis as well as the specific objectives.

1.3.1 Main objective

The main objective of our work is to propose mobility prediction models in self-organizing cellular networks that can best meet the demands of intelligent mobility management. Our work aims at proposing adoptive methods for mobility prediction that can address some of the issues particularly in terms of user mobility prediction accuracy improvement and execution time reduction.

1.3.2 Specific objectives

More specifically, we want to achieve the following specific objectives:

1. Propose novel generic mobility models for all types of users both predictable and unpredictable that can maximize the impact of data dependency of prior knowledge about user's past experiences on the mobility model. This objective highlights the problem of achieving poor performance in the case of irregularity and emphasizes the need for a generic approach for all users with any degree of predictability.

2. Optimize the proposed learning-based mobility prediction model regarding time complexity of the model in order to significantly reduce the execution time in comparison with the existing approaches with almost the same accuracy performance.

3. Propose a prediction-based handover management method to reduce handover related costs in the network.

4. Evaluate the performance of the proposed location prediction method using simulation tools in terms of robustness, prediction accuracy and time complexity.

1.4 Research contribution

In this thesis, we propose three mobility models for user trajectory prediction in mobile networks. We thoroughly exploit stochastic and learning methods to investigate and compare the impact of each approach on user mobility data. We propose a bidirectional analysis on the user trajectory to extract the mobility patterns. Moreover, we introduce particular preprocerssing phases before model training. The main contributions of this thesis are summarized in the following:

- Proposing a preprocessing phase to carefully prepare the mobility dataset specifically suitable for the trajectory prediction model. The main concern is to deal with the existing underlying issues related to the raw datasets. This phase helps to address two fundamental issues of redundancy and noisiness before training and inference steps. Kalman filter and line simplification algorithms are deployed to reduce the noise and obtain significantly lower execution time. This process helps to yield more accurate results.

- Proposing an RNN-based approach for user future trajectory prediction in which user location sequence is investigated chronologically. Recurrent neural networks and its variants are carefully compared to prove their performance in modeling user mobility.

- Proposing a novel trajectory prediction approach based on bidirectional recurrent neural networks that enables highly accurate predictions. The proposed mobility model analyzes both forward and backward correlations in the user past movement history which gives us a deeper understanding of user mobility behavior. To the best of our knowledge, it is the first time that a model based on bidirectional gated recurrent unit (BiGRU) has been deployed for user trajectory prediction.

- Proposing a mobility prediction technique to model the user mobility behavior. The proposed VAR-GRU model predicts the future trajectory (i.e., path) of the user. The core concept is to fully analyze the existing dependencies in the user past trajectories and extract general pattern in the data.

- Investigating the impact of user mobility prediction on the conventional handover signaling procedure. Handover processing and transmission costs are evaluated to compare the predictive and non-predictive scenarios.

- Conducting comprehensive experiments on data generated from real-world users to show the effectiveness of the proposed approaches. We deployed several metrics to evaluate the model's accuracy performance, reliability and execution time.

1.5 Thesis outline

In this chapter, first we provided a brief introduction bout the importance of our work and the main motivation behind it. Then, some of the basic definitions and concepts regarding a understanding our work is presented. Next, our research objectives are stated. In chapter 2, we summarize recent relevant works regarding mobility prediction and intelligent mobility management. We review these works from two points of view including types of mobility models and types of mobility data. Chapter 3 provides a high-level explanation regarding our general approach and its steps toward fulfilling our research objectives that are stated in section 1.4.

In Chapter 4, the full text of our article entitled as *RNN-Based User Trajectory Prediction Using a Preprocessed Dataset* is provided. This article was published in *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob).* This work proposes a user trajectory prediction scheme in mobile networks. Using user movement history as the prior knowledge, a learning model based on recurrent neural networks variations (i.e.,RNN, LSTM and GRU) are exploited to extract user movement patterns. Furthermore, we carefully investigate the impact of line simplification techniques on simplifying user raw trajectories before training phase.

In Chapter 5, the full text of our article entitled as *A Bidirectional Trajectory Prediction Model for Users in Mobile Networks* is provided. This article was published in *IEEE ACCESS* Journal. This work proposes a method that relies on bidirectional gated recurrent units. This approach aims at considering not only the correlations of the location sequence in the chronological order but also take into account the undeniable existing dependencies in the reverse order. User GPS trajectory is deployed as the model states. First, input sequence is compressed using line simplification techniques to eliminate the unnecessary data points and reduce the execution time. Then, we study the existing underlying dependencies and patterns in the sequence of consecutive visited locations.

In Chapter 6, the full text of our article entitled as *A Hybrid User Mobility Prediction Approach for Handover Management in Mobile Networks* is provided. This article was published in *Telecom Journal.* The proposed approach has four main parts: (1)mobility model training based on vector autoregression model, (2)model training using gated recurrent units, (3)inferring future trajectory of the user based on the extracted patterns and (4)improving handover signaling procedure and related costs.

In Chapter 7, we provide a general discussion on the thesis outcomes relating to the defined research objectives. Next, we analyze the obtained results and their effectiveness and impact. Finally, chapter 8 presents a summary of the proposed works in this thesis, underlying research limitations. Moreover, some potential solutions for the aforementioned limitations and some research directions as future works are provided at the end.

CHAPTER 2 LITERATURE REVIEW

This chapter carries out a fundamental review of the recent related works in the field of user mobility prediction. Mobility prediction and its applications have been extensively studied and there are a lot of wide-ranging works about them in the literature. First, we explain the importance of the mobility prediction in future mobile networks. Then, we review the recent related works in this field.

2.1 The importance of mobility prediction

Mobile computing has attracted a tremendous attention in recent years and has a leading role in information dissemination because of its numerous advantages. Despite the conventional mobile communication standards with low data rates such as GSM (global system for mobile communication), recent standards like LTE (long-term evolution) support higher data rates in order to meet the demands of users. Although, mobility adds a wide range of great services for users in the network, it can cause several critical issues at the same time. One common problem in cellular networks is managing the frequent handovers [12]. When a mobile node changes its location to a new cell, available bandwidth is not guaranteed in the new coverage area especially if the new area is congested. This may lead to an increase in call dropping probability and QoS degradation which is not acceptable since the user has paid for this service.

The only way for dealing with call dropping and supporting a seamless connection is passive bandwidth reservation. In this way, a part of bandwidth is reserved for mobile node in some of the cells which are most likely to be passed by the user during his/her movement. This technique does not depend on any type of technology or infrastructure and can be applied to several situations [12]. Therefore, predicting the future locations of user in the network is of a great importance in order to provide the information of user location for the network such that it can passively reserve bandwidth for the user in future locations with the purpose of call dropping probability reduction. The undeniable growing number of mobile devices with the key feature of mobility highlights the importance of smart mobility management in the network.

Therefore, a deep understanding of network traffic behavior is of crucial importance in today's rapidly growing mobile networks. There are several surveys that provide a comprehensive overview of the existing works with different points of view. In [13], an extensive survey is conducted which is basically composed of classification, prediction and optimization techniques for mobile networking. This survey reviews a list of works that are about geographic-based predictions in mobile networks such as next location prediction, trajectory prediction and mobility-assisted handover optimization. Another survey [4], reviews some of the works in this field based on the learning algorithms in self-organizing cellular networks. Another recent literature review of mobility and geolocation prediction is provided in [9] which mainly reviews geolocation prediction techniques in the context of mobile big data such as mining personal trajectories. In [3], an overview of geolocation prediction for mobile big data is provided. This work only focuses on the prediction based on GPS data due to several reasons including high location accuracy, easy data collection and the fact that it is commonly used in the literature.

Existing methods in literature can be categorized from different points of view. Figure 2.1 shows and example of dividing mobility models into main four categories of mobility models based on mobility data sources, estimation approach, learning-based methods and stochastic models. In the following, we review and classify the recent current works from two different points of view. First, we conduct a review of mobility prediction approaches using machine learning techniques applied to self-organizing cellular networks. Our work substantially relies on the learning methods and mainly is about learning-based prediction techniques which have gained a tremendous attention in this field. Then, a thorough review of the prediction approaches is provided that is based on different kinds of mobility data sources in the network generated by the user device while moving including GPS data, cellular networks data, Wi-Fi data and social networks data.

2.2 Related works

In this section, we overview the recent existing works using machine learning and deep learning techniques. Next, mobility prediction approaches using different mobility data types are briefly presented.

2.2.1 Mobility prediction approaches using machine learning and deep learning techniques applied to self-organizing cellular networks

There are a huge number of works that fits to this group. Learning-based mobility prediction approaches has attracted a tremendous attention in the recent years due to the popularity of machine learning techniques and their numerous advantages for different applications. Here, we summarize some of these approaches.

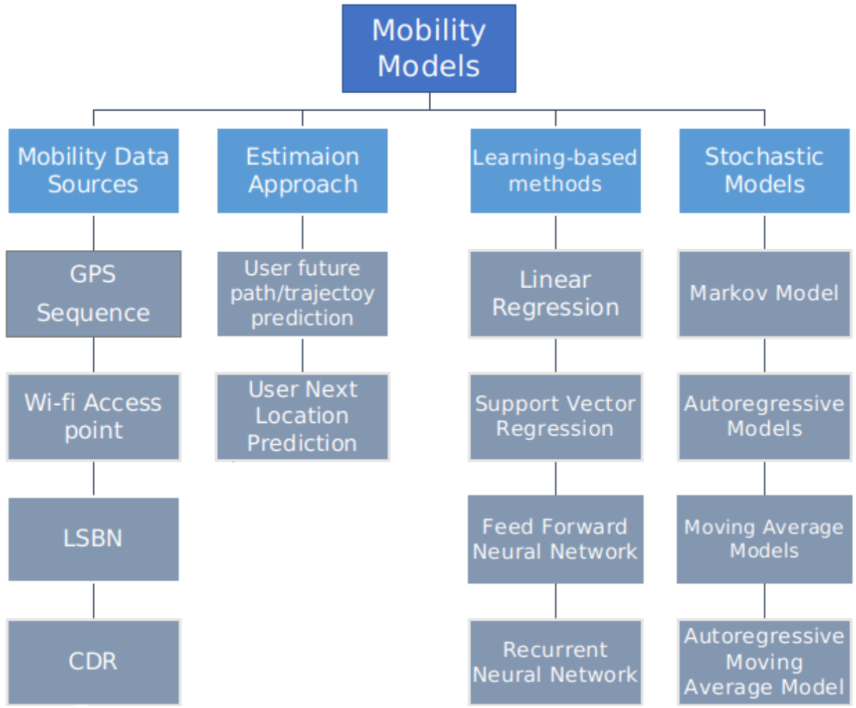


Figure 2.1 Mobility prediction models.

A mobility prediction method was proposed in [14]. This mobility model predicts users trajectory based on an online learning process using mobility data history. The main purpose of this work is to reduce handover-related signaling load and minimize the latency using a predictive handover scheme. In [15], authors proposed a location prediction technique based on the user's activity pattern. Their approach has three main steps. In the first step, supervised learning is used in order to infer the current activity type of each user using contextual features. Then, individual's next activity is predicted. Next activity prediction is based on a dynamic model for the user's activity pattern. In last step, next location of user is predicted using the information from step 1, 2 and past visited places.

A huge part of localization techniques and mobility predictors are based on different types of Markov models. The movement path of each user is modeled based on his/her record of visited cells or geographic information and Markov predictor is deployed in order to predict the future cells. In [16], a Markov-based trajectory prediction technique called Hidden Markov model-based Trajectory Prediction (HMTP) was presented that can estimate the complete path and not only a part of it. To deal with the changing speed, parameters of the algorithm are chosen actively and important parameters are automatically adopted with the changes. Moreover, for trajectory anticipation in a larger scale, a partitioning algorithm was provided that has an acceptable performance. Simulation results show that in the case of changing speed HMTP, has a better accuracy performance compared to traditional methods.

In [17], authors proved that Markov models are not capable of capturing criticality in user mobility. In other words, Markov-based models cannot accurately model human mobility. In this work, it was shown that Markov models fail to capture long-term correlations in a user trajectory over a long distance.

A proactive multicast approach based on mobility prediction was proposed in [18]. Multicast is used for controlling the large mobile populations to provide seamless connection for mobile users in the heterogeneous networks. This work mainly focused on the post-handover delay issue for cases that are sensitive to delay. The core idea is to exploit a Markov model to extract mobility patterns of users in the cellular network. A semi-Markov model was used to predict both the next cell transition and also the time interval between the cell transitions in the cellular network. In this work, authors model user mobility behavior using a Markov model in which the Markov states are the ID of visited cells and the time duration in each state relies on current and next state of the user in the network. Moreover, the proposed learning-based prediction technique can be applied for both offline and online learning. In offline learning, a training phase is needed prior the using our mobility model in order to obtain the best parameters for our model and learn the pattern. In online learning, prediction probabilities are dynamically updated for any changes in the movement behavior. Performance evaluation results show that the proposed mobility-based proactive multicast has a better performance in comparison with other methods such as blind proactive multicast.

A novel approach with a different perspective was introduced in [19]. This work aims at predicting mobility inside a tracking region and does not accept the assumption of mobile node being static in a tracking region. This method is called mobility prediction inside a coverage hole (MIRACLE) and its main goals are to anticipate mobility inside a coverage region and also if the target's mobility pattern changed, it should be able to predict the possible transitions between different mobility patterns. MIRACLE is mainly based on stochastic learning weak estimation (SLWE) mechanism which can predict a parameter based on its observations. Five different mobility models were deployed to evaluate the performance of the MIRACLE. It was shown that this method can achieve an average accuracy of approximately 60% and 67% for respectively mobility model prediction and mobility model transition.

One challenging part of deploying user movement history is the fact that this information needs pre-processing before we can use it for our mobility model since it is raw data and should be labeled first. In [20], a supervised learning method was used for location prediction since no data labeling is required when using check-in data bases. It is worth noting that check-in data bases contain the information of visited locations. Another machine learning technique for location prediction was proposed in [21]. This work focuses on the concept of data labeling for mobility model training. Considering the fact that data labeling needs great effort, authors believe that a reasonable accuracy performance can be achieved with very little or even zero labeled data. They used an expectation maximization (EM) technique for location prediction.

Authors in [22] proposed a spatiotemporal mobility prediction method in self-organizing cellular networks. In this method, mobility traces in LTE network were utilized in a Markov-based mobility model in order to predict next base station. This method achieved an acceptable accuracy performance up to 90%. A novel distributed prediction with bandwidth management algorithm (DPBMA) was proposed in [23]. Instead of having one centralized predictor for the whole network, DPBMA uses several distributed predictors for each coverage cell. These distributed predictors are based on Markov models that can predict local mobility. In [24], a long-term mobility estimation scheme was proposed in order to learn the mobility patterns based on the previous trajectories. For the mobility model, trajectories with a new representation, were clustered based on the trajectory similarities for the offline learning process. The key purpose of this article is to estimate a group of possible future paths for each individual in the network and save the information at potential sensor nodes in the path. This technique can significantly reduce routing costs, optimize energy usage and improve network lifetime.

In [25], an approach called smartDC was introduced. This scheme has three main parts: mobility learner, mobility predictor and adoptive duty cycling. In the first part that is an unsupervised mobility learner, user's movement history is collected. Mobility predictor is responsible for estimating the departure time of the user towards the next location, the proposed location predictor is based on Markov model and time series analysis. In the last part (i.e., adoptive duty cycling), the main purpose is to increase the mobility monitoring accuracy and take into account the energy budget of user's devices.

In [26], a method was proposed for mobility monitoring aiming at minimizing the energy needed for mobility data collection. The proposed method is a location inference model based on HMM. Using cell connection patterns, the inference model detects the change of places with no need for sensor activation when visiting the previously visited places.

Another Markov-based method was proposed in [27]. This novel approach is called destination and mobility path prediction (DAMP) and consists of two main parts: destination prediction model (DPM) and path prediction model (PPM). DPM is responsible for estimating the end location deploying second-order Markov chain based on information about user's habits and movement direction and contextual knowledge. Then, PPM predicts the path toward the destination that is anticipated by DPM. At each road intersection, PPM algorithm calculates probabilities regarding the candidates for the next road (using a second-order Markov chain). The road with the highest probability is then chosen as the next road. This method results in a higher accuracy performance compared to previous works based on simulation results using real trajectories.

An approach for predicting next place in cellular network was proposed in [28]. This approach is composed of a pattern classifier and a location predictor. Pattern classification is based on the information about the user's current location and movement history. Two classifiers were deployed in this work: K-nearest neighbor and decision tree. Location prediction is performed based on the results of trajectory classification. Two types of predictor were introduced in this method namely 12 Micro and Macro location predictor, the key difference between them is the way that trajectory classification is performed, considering all the visited cells for classification or just the neighboring cells. In [29], a destination prediction approach was proposed based on the concept of LSTMs for modeling long-term and short-term effects of user consecutive locations. In this works, authors proposed a location-aware mechanism for mobility prediction and predicting the user destination.

2.2.2 Mobility prediction approaches using different mobility data types

As explained earlier in the previous parts, there are different technologies that can be utilized as tracking mechanisms using mobile devices for user location recognition. These various localization methods result in generating various kinds of mobility dataset. In this section, we categorized some of the existing works in the literature into four groups according to the four popular types of mobility data sources including: GPS data, cellular networks, Wi-Fi data and social networks.

Table 2.1 Mobility data types.

Mobility data	Location accuracy
GPS data	Very good
cellular network	Poor (depending on the cell range)
Wi-Fi data	Good
social network	Depends on the user's honesty

2.2.2.1 Prediction techniques based on GPS data

This section studies localization methods based on GPS data. This source of mobility data is known for achieving a high predictability accuracy and small location error. Here, some of the GPS-based mobility models are reviewed.

In [30], a moving destination prediction method was proposed for the sparse data sets which is called MGDPre (Mobility Gradient-based Destination Prediction). This work deals with the fact that the available trajectory data sets are mostly sparse because of various reasons such as privacy issues. Data sparsity may lead to high prediction accuracy degradation. Instead of finding similar trajectories in the sparse available data records, authors proposed to take advantage of the fact that the final destination of the user is closely related to his/her purpose of movement. In this paper, firstly, a destination prediction technique was proposed based on the mobility gradient decent. In the second step, after predicting the final destination, the focus was on examining the mobility behavior of query trajectory. More precisely, the changes of distance between ongoing trajectory and the final destination was investigated to choose the route toward the target. In the third step, two strategies were introduced to improve the MGDPre prediction accuracy in which the main idea is to minimize the search area for finding the final destination. A GPS dataset containing taxis trajectories for a period of 10 days was deployed to conduct experiments for performance evaluation of the presented approach. Results show that MGDPre has a better performance compared to conventional methods in the state-of-the-art in terms of accuracy and scalability.

A location prediction system using GPS data was proposed in [31]. This approach contains three main parts: location extraction, location recognition and location prediction. In the first part (i.e., location extraction), in order to improve the prediction performance a Gaussian-means (G-means) algorithm was used instead of k-nearest neighbor (KNN). Moreover, to further optimize the proposed method in terms of model complexity and execution time reduction, G-means clustering was used only for stay points of GPS data. In the second part, KNN and decision tree algorithms were exploited for location recognition in both indoor and outdoor environments. Last part predicts next location using hidden Markov models (HMM). The reported average accuracy performance is approximately 90% based on the simulation results.

A bandwidth reservation method with the use of mobility prediction technique in the network was proposed in [32]. Authors used a GPS trajectory for a period of two months in order to predict the mobility path of users. Later, they used this information to estimate the handoff time and control the bandwidth allocation. This can significantly help to reduce the handoff latency and active session dropping. In [33], a comprehensive survey of mobility pattern prediction based on GPS data mining was provided. This work gives a general perspective on user's mobility pattern using GPS datasets that is classified to two subsections: mining mobility patterns and introducing mobility models. In this work, an overview of mobility prediction based on GPS data was provided that mainly covers the recent works on mining the GPS trajectories datasets. The main purpose of this work is to investigate the mobility model of each user individually which can be classified to two groups, first groups is extracting useful information from mobility patterns and the second one is to build a mobility model. In this survey, authors overviewed different techniques for location inference, several prediction algorithms and probabilistic models in the context of individual future location prediction. They believe that a few works has discussed about the maximum possible prediction accuracy and a high-accuracy generic mobility model is still missing. Furthermore, this paper describes a trade-off between spatial resolution and prediction accuracy that has an important impact on the mobility model. Another important factor in each user movement behavior is user's transportation mode that can directly affect the speed of user while moving. Lastly, authors suggested some future research directions including: applying data compression techniques to the mobility data, building new mobility models based on real trajectories, performing empirical study based on real trajectories and prediction performance improvement.

In [34], a method called Geographic-Temporal-Semantic-based Location Prediction (GTS-LP) was introduced which is a mining-based future place prediction. The main concept of this work is about location recommendation models which are based on next location prediction. Authors believe that predicting future location of the user's depends on not only their interests but also it is closely related to their intentions. More precisely, users tend to go to different places not only because of their interest but also because they have to do something there such as workplace. This method extract useful information (user mobility pattern) from raw GPS data with paying special attention to three types of users intentions: semantic-triggered, geographic-triggered and temporal-triggered intentions. Evaluations results show an acceptable performance for this approach.

In [16], GPS trajectory was used for path prediction. A dynamic self-adaptive algorithm was proposed to tackle the problem of rapid changes for moving devices and mobile users. Furthermore, an algorithm for trajectory partitioning was proposed to improve the efficiency. In [1], authors proposed a trajectory prediction method using LSTM model. In the second part of their work, a region-oriented multi-user mobility model was introduced using sequence to sequence LSTM model. In [35], a trajectory representation learning approach was proposed which is called Trembr. This method focused on capturing temporal and spatial features in the trajectory using a RNN-based model.

2.2.2.2 Prediction techniques based on cellular networks and Wi-Fi data

This group of works are based on using cellular and Wi-Fi information that are known as energy-efficient techniques while having a higher location error. Basically, cellular networks are partitioned into some smaller areas called cells with minimum one base station in each cell. Each cell has a unique ID that can be used for tracking the user in the network based on the visited cells. In this context, the movement history of each user is define by the cell ID that has been recorded during his/her move.

In [36], authors used cellular networks information in their work since they believe continuous GPS data collection is not energy-efficient. Although cellular-based mobility models are more efficient in terms of energy usage, they fall short of location error in many cases because of different types of signal fading and other disruptive effects in cellular networks. To deal with this problem, authors proposed to combine cellular information with nearby smart phones aiming at location error reduction. For example in the case of public transportation (e.g. a bus), cellular information of all the users in the bus can be utilized to increase the location accuracy. They used a fingerprint algorithm in their method that takes the cell ID and signal strength as input and gives the best matched location according to that.

In [37], one of the most popular cellular-based datasets was used which is called “MIT reality mining dataset” that is composed of a million GSM traces and almost 110 calls. This paper investigates the mobility prediction problem from a new point of view and is based on social interplay.

It was shown that there is a close relationship between call patterns of two arbitrary users in the same cell at the same time and social relationship has an undeniable impact on cellular calls. Relying on this correlation, an algorithm called NextCell was proposed, focusing on prediction accuracy improvement. This approach consists of two main predictors that are respectively based on regular behaviors and social interplay. This method can anticipate next locations of the person for the following 6 hours. Hence, social interplay between users and the corresponding cellular calls significantly affect user movement behavior.

More specifically, NextCell has three main parts including periodicity predictor, social interplay predictor and self-adjust learner. In the first part, periodicity predictor, user mobility history (in this case cellular information) was investigated in terms of temporal and spatial regularity. This predictor extracts a time-table regarding the user’s frequent visited locations. Then, a probabilistic model for periodic behavior is computed and future location is estimated based on this probabilistic model. In social interplay predictor, next location prediction is performed based on social relationships. Using GSM traces, social interplay between two arbitrary individual is detected. Social interplay refers to the possibility of co-occurrence in the same cell at the same time. This predictor mainly tries to answer to two questions in order to discover the social correlation for future prediction. These question are: who will be the possible candidates for the next co-cell occurrence? and where will be that cell? The answers for these questions can be found based on likelihood estimation. In the self-adjust learner, the two predictions obtained from the last two steps are aggregated to have prediction with a high accuracy. In order to evaluate NextCell, MIT dataset was used. This dataset is collected for a period of approximately 18 months form 106 users and consisting 32,579 cell towers with almost 112000 mobile calls. The results of experiments show that the proposed approach achieves a higher prediction performance. It was shown that, for most cases social interplay has a remarkable impact on prediction result. However, there are some users with very low regularity and also as the prediction time increases the accuracy decreases.

Trajectory datasets are widely used in localization techniques. Some of these datasets may contain large errors depending on the type of localization technique. As mentioned earlier, cellular-based localization methods are more prone to location accuracy degradation in comparison with using GPS datasets. Localization error based on cellular information is noticeably higher than GPS. The reason behind this is the fact that location accuracy using cellular network information has a close relationship with the cell coverage area which can vary from hundreds of meters to several kilometers (depending on whether it is an urban area or rural). Figure 2.2 shows an example of a cell with a range of 2Km which leads to low accuracy. There are several works in the literature that focus on dealing with this problem using different techniques.

In [38], an approach called CLSTERS was proposed in order to reduce the errors in a noisy

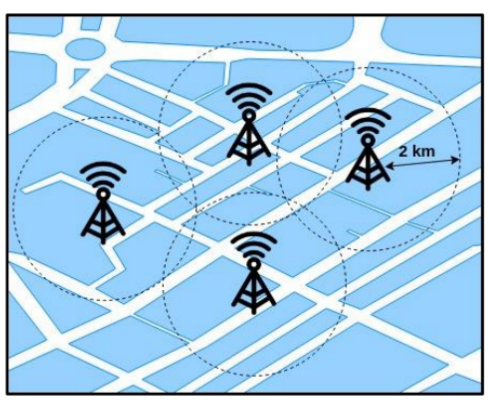


Figure 2.2 A cell with a range of 2Km which leads to low location accuracy.

dataset. CLSTERS is the short form of "Challenging Localization Situation-aimed Trajectory Error Reduction System" and has four main steps including filtering, route inference, candidate construction and interpolation. In filtering step, datasets points with unusual large errors are removed. Then, in route inference step, an HMM model is used to inference the route from trajectories based on the road network information. In candidate construction phase, the points with small error in dataset is chosen to further reduce the error using Bayesian soothing technique. In the last step, interpolation, the remaining points in the dataset are calibrated based on the constructed candidates from the last step. To evaluate the performance of CLSTERS algorithm, a number of experiments were carried out. For these experiments, three different cellular-based noisy datasets with different characteristics were chosen. The difference between these datasets are in terms of duration, trajectory length, cellular network type (4G/3G), localization technique, etc. Results of experiments show that CLSTERS can significantly improve the localization and decrease the errors up to 40% in compared with original errors.

In another work [39], a Markov-based map matching method was proposed for cellular trajectories. Localization based on cell ID is highly prone to errors because of the fact that the cell coverage may be up to several kilometers and the exact location of the user within that area is not clear. Moreover, user may be in the coverage area of more than one cell and this can cause the ping-pong effect which is not desirable. Map matching is the process of mapping a set of erroneous points to the accurate place on the road network. This technique is widely used for location-based applications and can address the aforementioned issues. For performance evaluation, cellular-based location information was used that is obtained based on the cell tower information and simulation results show significant improvements. In [40], a cloud-based mobility prediction approach was proposed for mobile big data in telecom cloud. The core concept of this work is based on the idea that call patterns are closely related to co-location patterns and extracting this correlation using cellular call history can be very influential for network management. The proposed method is called NextMe and its main goal is to estimate user's future location. It is composed of five main units: data preprocessor, periodicity module, social interplay module, self-adjust learner and call pattern recognition. To evaluate the proposed system a dataset of cellular call records containing almost a million cellular calls was used and results show that NextMe can improve the prediction accuracy up to 14 percent. A hybrid technique using both data types was introduced in [26] that is an energy-optimized movement monitoring method called FreeTrack. This method focus on the fact that consistent mobility monitoring leads to energy shortage for the exiting energy-constraint devices.

The core idea of FreeTrack is to learn the frequent patterns and in the case of observing a previously visited pattern it can infer the location without any sensor activation. More precisely, user's mobility pattern is learnt by a Markov model and this mobility model is able to infer user's location based on the history of cell tower connections. For evaluation, mobility data generated by cellular module, Wi-Fi and GPS in a one period was used and results show a 68% improvement in terms of energy consumption reduction while achieving a reasonable accuracy performance. In [41], authors proposed a method based on machine learning algorithms using WiFi data. Their proposed model is based on the concepts of principal component analysis (PCA) and gated recurrent unit (GRU). The hybrid PCA-GRU trajectory prediction approach optimize the performance of predictions in comparison with PCA-LSTM and PCA-MLP models.

2.2.2.3 Prediction techniques based on social networks

This group of works takes advantage of the rapidly growing popularity of smart devices and location-aware applications between users. Location-based social network (LBSN) is an emerging paradigm that has attracted tremendous attention recently. LBSN is used for sharing locations of everyday activities and opens up plenty of opportunities for different location-aware applications. The collected visited location's records helps to better extract users mobility patterns; therefore, results is a more accurate future location estimation since the user provide the exact location. To this end, a model called CEPR (Collaborative Exploration and Periodically Returning) was proposed in [20]. CEPR exploits the potentials of LBSN in a novel way in order to improve the accuracy performance of location prediction techniques. First, they investigated the exploration prediction problem that is a binary classification aims at dealing with user's unvisited locations. This classifier anticipates whether their future location is new or already existed in the record. Three classifiers were trained on two LBSN datasets for exploration prediction problem. Then, CEPR model was used based on the results of the previous part. This model works as both location prediction and recommendation and could improve the performance up to 30% in comparison with conventional methods. In [42], a novel model proposed to aggregate the social relationships and mobility behavior of users which is based on neural networks. This approach has two main parts, first part is to find social correlations and build a social network and the second part is to model mobility trajectories. It was shown that taking into account the social relationship is very helpful for mobility prediction performance. In another work [43], authors studied the frequency that users tend to share their locations of their visited places. They used Foursquare, a LBSN, to investigate the relationship between user's personalities and types of visited places (mobility pattern). They found some correlations between individual's personality and his/her record of previous visited places. This work concluded that user's personality has an important impact on the type of visited places for each person and can significantly help to a more accurate future location estimation.

In [44], a RNN-based model was introduced called DeepJMT. This model predicts future location and the corresponding time using several components including: an encoder, periodicity/spatial/temporal context extractor. This context-aware method deploy social relationships information to improve the performance. In [45], authors presented a method to predict the future point-of-interest. The proposed method, VANext, deploys convolutional neural networks to discover long-term dependencies.

CHAPTER 3 THE GENERAL APPROACH OF THE ENTIRE RESEARCH PROJECT

This thesis highlights the importance of mobility prediction in future mobility-aware self-organized networks. In the previous chapters, we provided the main motivation behind our work, our research objectives, a review of the recent works and the remaining open problems. Our work tries to address some of the aforementioned unsolved issues. We specifically emphasize the importance of two points: (1) mobility data preparation (2) deep correlation extraction between user locations. In this chapter, we present the entire methodological approach implemented towards fulfilling the defined research objectives in section 1.4. In particular, we highlight the links existing between the various objectives set out in Section 1.4 and the scientific articles presented in the next chapters of the manuscript. In the following, we explain the general procedure in three main phases, each of which results in a scientific article.

3.1 Phase 1: Unidirectional mobility model using a one-step preprocessed mobility data

Future location prediction has been studied in many areas and several works. In the existing works, user raw data were used to predict future location with either no or minimum data preprocessing. In this phase, we introduced a specifically designed data preparation with the main objective of eliminating irrelevant data samples in user movement history. Then, we exploit the modified information for modeling user mobility behavior. This approach is elaborated in the article entitled as "RNN-Based User Trajectory Prediction Using a Preprocessed Dataset", presented in chapter 4.

3.1.1 User trajectory preparation

In the most existing works, the main focus is on the mobility model and its features. However, user data is of great importance and needs to be considered as well. User raw mobility data contains some inherent uncertainties that will result in inevitable negative consequences. The main reason is the fact that this data is collected from real-world mobile users and face the problems of noisy measurements, redundancy, sparsity, etc. Here, in this phase, we propose to modify user movement history with the main purpose of simplifying it and eliminating the irrelevant data. Adding this step before model training and inference has highly beneficial effects on the model time complexity and network parameters.

3.1.2 RNN-based user trajectory prediction

Mobility model states are highly correlated and it is vital to discover this dependency between sample inputs of the model. We deploy recurrent neural networks since they have memory to capture states correlations. The main objective is to get the modified trajectory of the user we plan to predict the future trajectory of the user for the next time steps. The proposed model investigates the correlation between them in order to find repetitive patterns and predict the future trajectory.

3.1.3 Experimental results

To comprehensively evaluate the accuracy of the proposed approach, we provided a series of experiments. We set a threshold for the trajectory simplification step to control the degree of the data elimination. We investigate the impact of this threshold on dataset size, model time complexity and mobility model prediction error. We analyze three variants of recurrent neural networks as mobility predictors. Results show that our approach improves the prediction error with a remarkably lower time complexity.

3.2 Phase 2: Bidirectional mobility model using a two-step preprocessed mobility data

Existing works focus on exploring dependencies in user consecutive visited locations only in chronological order. In these approaches, one or more of the previous states are analyzed to predict the future state of the mobility model. However, in reality, each location point (i.e., model state) is heavily dependent on both the previous and next states. To this end, in this phase, we propose to deploy an approach that can thoroughly explore user movement history based on the both previous and next locations for each state. In the following, we extensively elaborate the main idea of this approach which has been submitted as an article entitled "A Bidirectional Trajectory Prediction Model for Users in Mobile Networks". This article is presented in chapter 5.

3.2.1 Mobility data preparation

In the previous phase (Section 3.1.1), we discussed about the importance of processing the raw data before model training. We employed an effective technique to modify user trajectory and convert it to a simpler trajectory with fewer number of data points and same accuracy. Here, we add a new step to the data preparation for reducing the noise level in the collected data. This noise comes from data measurement process which is inevitable. We apply a Kalman filter to remove noisy samples before train and inference phases.

3.2.2 Bidirectional mobility prediction model

In this step, we propose to deploy an approach that is able to carefully explore the dependencies in data sequence. In the previous works, for each state at each time step for a user, only the forward dependencies of information from past time steps until present is considered for predicting the next step. However, BTPM considers both the forward and backward dependencies of the user past movement history. Our approach is based on recurrent neural network concepts that bidirectionally investigate user movement history. When network can make a decision based on the whole input sequence (past and future), it would tremendously enhance the performance of the mobility model.

3.2.3 Experimental results

We conducted a comprehensive set of experiments for evaluating the performance of the proposed approach and its effectiveness. We performed our experiments on three main areas including (1)impact of trajectory simplification on the mobility model, (2)impact of noise reduction on the model performance and (3)how does the proposed bidirectional model perform specially in comparison with the previous methods. We carefully considered several metrics to show how each mobility model performs for different users. In our work, we utilized three different mobility datasets, and several users from each dataset. We believe a mobility predictor is robust and reliable when it can work well dealing with different users' mobility behaviors. Experimental results show that the proposed method has the lowest prediction error for all the users in the experiments. Our method has a remarkable improvement in time complexity reduction.

3.3 Phase 3: Mobility-aware handover management

User mobility and trajectory prediction can potentially be used in many research areas related to enhancing mobile networks quality of services from different aspects. In this phase, we want to fully exploit the potentials of information we gained from the previous phases and deploy to improve user mobility management. Our main objective is to reduce the unnecessary handover signaling messages using the knowledge of user predicted future trajectory. We propose a hybrid method based on statistical and learning techniques to extract user patterns, then, we investigate the impact of the proposed mobility model on the handover related cost. This approach is elaborated in the article entitled as "A Hybrid User Mobility Prediction Approach for Handover Management in Mobile Networks", presented in chapter 6.

3.3.1 Hybrid mobility learning

Having the movement history of a user for the previous time steps, we aim to predict the user's future trajectory for time steps ahead. First, we feed the user past trajectories into a vector autoregression model to obtain useful insights on the user mobility behaviour. Each variable is obtained by the lagged values of all of the variables. In another words, to predict the value of a variable at the current time step, we consider the impact of all of the other variables from the past time steps. For the next step, we feed the obtained results into a deep GRU-based neural network. For each time step, we consider the current location and the previous state to obtain the next state as the output. The GRU model has two main parts that deal with keeping or deleting user data from the past, namely the update gate and the reset gate. Therefore, we deploy a hybrid mobility learning method that can benefit from the merits of both approaches.

3.3.2 Handover related cost analysis

Having the user future trajectory, we want to reduce the number HO signaling messages needed when handover is triggered. We investigate the conventional (i.e., non-predictive) and predictive handover signaling procedures. More specifically, we analyze the impact of trajectory prediction on handover signaling flow and compare it with the non-predictive handover scenario.

3.3.3 Experimental results

For the experimental results, we carried out a series of simulations to investigate the impact of the proposed approach. We chose four mobile users to study how our model works in comparison with other predictors. GPS trajectory of these users were deployed as their movement history. Simulation results show the effectiveness of the proposed approach from three aspects: (1) significant reduction in mobility prediction error, (2) handover transmission cost improvement and (3) handover processing cost reduction.

CHAPTER 4 ARTICLE 1: RNN-BASED USER TRAJECTORY PREDICTION USING A PREPROCESSED DATASET

Authors: Nasrin Bahra and Samuel Pierre
Status: Accepted for publication in International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)

Abstract

Future mobile networks are rightly expected to face the prospect of limited available resources. Continuous technological advances and growing number of mobile devices highlight the importance of further improving the performance of mobile networks. User mobility poses technical problems in network management. It is essential to ensure a satisfactory level of quality of service for users. To achieve this goal, self organizing networks (SONs) are potential solutions to fulfill the requirements of users using learning algorithms. In this paper, we propose an intelligent mobility model to predict future trajectory of the mobile user in mobile networks. The proposed approach has two main parts, including mobility data preparation and user mobility prediction. Our primary focus is on providing a carefully tailored mobility data from raw mobility datasets using line simplification techniques. Next, we use the accurately prepared data for learning user mobility behaviour and predicting user future trajectory using recurrent neural networks and its variants. Simulation results show a substantial decrease in execution time from 4616s to 932s for the best case. The proposed learning approach obtains a loss value of 0.10 using a model based on long short term memory (LSTM).

IEEEkeywords

trajectory prediction, neural networks, mobile networks, data reduction

4.1 Introduction

Rapid technological advances in many areas, such as the wildly popular paradigm of internet of things (IoT), can provide plenty of golden opportunities for users in the mobile networks. However, user mobility can cause a range of problems such as increase in call dropping, increase in number of handovers, limited available bandwidth, connectivity issues and in general quality of service degradation. Mobile users are entitled to expect a reasonable level of quality of service (QoS) in the network. Therefore, it is crucial to further improve the network performance and provide high level services to meet the demands of the users in the mobile networks.

User mobility prediction as a part of intelligent mobility management is a promising solution to address the mentioned issues. This concept is defined in the context of self-organizing networks (SONs). These networks tend to eliminate the manual efforts in configuration, optimization and healing in the network management. Their focus is on developing fully automated functions for different components and tasks in the network. To achieve this objective, learning algorithms can be fully exploited to automatize the network management. Machine learning and deep learning algorithms can learn from the data and optimize the performance based on the past experiences [4].

From another stand point, there is a huge growth in the number of mobile devices in recent years. It is anticipated that there will be 5.7 billion mobile users by the year 2023 [46]. This rising trend in mobile users with mobile devices leads to an inevitable generation of mobility datasets. This huge amount of data can be used to broaden our understanding of users' needs and users' mobility behaviour. We can extract mobility patterns for predicting future trajectory of the user based on user's movement history. It is worth noting that the generated mobility data is a raw data that may contain noise and redundancy to some levels. It is highly recommended to apply some preprocessing techniques to the raw data and prepare the data before using it [47].

A large number of research areas can gain maximum benefit from predicting user mobility trajectory in the network. When network has the information of the user future locations, it can exploit this knowledge to better provide services in advance. It can be employed in passive resource allocation, handover management, passive bandwidth reservation, call admission control and optimized routing. As an example, for a mobility-aware handover management, network can complete handover preparation steps in advance. When handover is triggered, the procedure starts from the execution phase. This can significantly reduce the handover signalling overhead and latency [5].

In this paper, we propose an approach for user trajectory prediction. The proposed approach has two essential parts. First, we introduce an initial data preprocessing step to provide an appropriate dataset as an input for the model. This step is focusing on removing irrelevant data from the dataset and keeping only determining data points. We want to provide an analysis of different data reduction techniques. Second, we feed the prepared data to the mobility model to predict the future trajectory of the user. This model is based on recurrent neural network (RNN), long short term memory (LSTM) and gated recurrent unit (GRU). In the following, a part of recent related works are provided in Section II. Section III presents the proposed approach. Finally, simulation results and conclusion are provided in Sections IV and V, respectively.

4.2 Related works

A wide range of works in the literature have studied user mobility prediction in mobile networks from different aspects [4, 9, 13, 47, 48]. From one point of view, we can divide these works into two popular categories: Markov-based approaches and learning-based schemes.

A large part of existing works relies on Markov models. In [16], authors proposed a trajectory prediction method based on hidden Markov models. They considered the impact of changing speed and introduced a self-adaptive algorithm for selecting parameters. Also, in [27], authors proposed an approach to predict the destination and mobility path using a second-order Markov model. However, it is shown that Markov models fail to effectively capture the long-term data dependency in the user past trajectories [17].

Moreover, there are numerous works based on learning algorithms in the literature for user mobility prediction. In [41], authors used deep learning algorithms such as neural networks, principal component analysis (PCA) and GRU for providing an accurate prediction for mobile IoT devices. An LSTM-based prediction approach was proposed in [1] to predict user's mobility based on mobility patterns. They also introduced a new parameter to control data transfer during the training stage. Another LSTM-based model was proposed in [49] for vehicle trajectory prediction. This method can make prediction for one to five seconds ahead. In [50], authors proposed an RNN-based model to predict the next location of the user. They introduced time/space-specific transition matrices to provide a novel spatial temporal prediction. Authors in [48] discussed location prediction in vehicular ad hoc networks (VANETs). They mentioned neural networks and machine learning algorithms as possible techniques to predict the future location of the vehicle. They concluded that artificial neural networks (ANNs) do not perform very well in terms of time complexity. Furthermore, they fail to learn the long-term data correlation between past movement of the user and accordingly result in poor accuracy performance.

Therefore, we need to introduce a model that can take advantage of neural networks and address the time complexity issue of them. We need a mobility model that considers the data correlation in the user trajectory to improve the accuracy performance. Moreover, we can deploy new data preprocessing techniques to effectively reduce the time complexity of the mobility prediction model.

4.3 Proposed approach

In this section, we present our proposed approach. We propose to deploy line simplification methods to reduce the irrelevant data from a big dataset and keep only the appropriate data for the mobility learning model. Next, we use an RNN-based model to find the repetitive patterns in the data and predict the future trajectory of the user. Details regarding these two steps are provided in the following.

4.3.1 Data reduction

The primary step of the mobility learning is choosing an appropriate type of mobility data as movement history of the user. There are several types of mobility datasets such as GPS, Wi-Fi, cellular and location-based social networks datasets. We chose GPS data since it has the highest location accuracy. GPS dataset contains user location information in the form of geographical coordinates (i.e., longitude and latitude). However, raw GPS dataset is composed of user location information at almost each second with a high sample rate. All of this data is not necessarily needed for our purpose. Removing the irrelevant data results in almost the same performance accuracy but with much lower execution time. Fig. 1 depicts a mobile user GPS trajectory. The black line represents the actual path of the user and the raw information of user trajectory. Red points on the line are the points that we keep from the dataset and remove the rest. In this way, we have a similar trajectory with much fewer number of points.

There are several techniques to eliminate the unnecessary data. Here, we propose to apply line simplification methods to the raw trajectories of users. The core concept of line simplification approaches is to keep only the part of data that is essential to form the trajectory and delete the rest of the data points between them. Here, we provide a comprehensive analysis regarding some of the preprocessing techniques for data reduction and evaluate the impact of them on mobility learning.

Objective: Given the trajectory of user b with the length of n $[C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow ... \rightarrow C_n]$, we want to convert it to a simplified trajectory with the length of m $[C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow ... \rightarrow C_m]$, where $m < n$.

In the following, definitions of five line simplification methods are provided. It is worth mentioning that we hereafter refer to a geographical coordinate of GPS data as a location in this paper.

- *nth Point*: It is a naive algorithm that is relatively simple and fast. It does not involve forming any mathematical dependencies with the future or past locations in the user trajectory. Given a predefined number of consecutive locations (i.e., user trajectory), this algorithm keeps only the nth location point plus one random point in the set and removes the rest.

- $Reumann - Witkam$ $(R - W)$: Unlike nth point method, this algorithm considers the correlation of consecutive data points locally. It divides the trajectory into some sections and investigates neighbour locations based on a given threshold. In each section, it forms a line with the first two points in the trajectory. All the points in the section that are closer to the line more than a predefined threshold are deleted from the trajectory. This process is iteratively done for all the sections in the dataset.

- *Lang*: This method divides the trajectory into several segments including neighbour locations and considers a search region for each segment. For each point, if the perpendicular distance of its neighbour point is more than a predefined threshold, search region gets smaller excluding the last point of the segment. Segmentation is based on the first and last points in the current search region. This procedure is repeated for all the search regions in the entire trajectory.

- $Visvalingam\breve{}Whyatt$ $(V - W)$: This approach defines effective areas for each location in the trajectory according to the consecutive neighbours. Given the points in the original dataset, effective areas for them are calculated. At each iteration, any point in the dataset with the smallest effective area is removed from the trajectory.

- $Douglas - Peucker$ $(D - P)$: It is a popular approach that produces a highly accurate line as an output. However, it has a high time complexity. D-P has a global routine in which the whole user trajectory is considered during the data reduction procedure. It takes the first and last points of the trajectory and draws a line. The farthest point form the line is selected. Then, a line is defined based on the first and farthest points. In this range, any point lower than a predefined threshold from the line is removed. This procedure is repeated for all the locations in the trajectory.

All the mathematical details about these techniques are provided in [51–55]. We deploy these techniques to prepare the dataset (i.e., reduced version) for the next step which is user mobility behaviour.

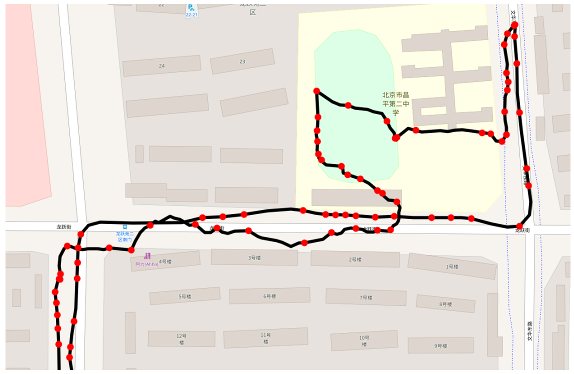


Figure 4.1 User raw GPS trajectory. Red points represents the necessary points that forms the path.

4.3.2 Learning user mobility behaviour

Objective: Given the modified trajectory of the user b $[C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow ... \rightarrow C_m]$, we plan to predict the future trajectory of the user for the next m time steps $[C_{m+1} \rightarrow C_{m+2} \rightarrow C_{m+3} \rightarrow ... \rightarrow C_{m+m}]$.

In this section, having the prepared movement history as prior knowledge, we feed this data to an RNN-based model to predict the future trajectory of user. The core concept is to take the user past locations and investigate the correlation between them in order to find repetitive patterns and predict the future trajectory.

For an RNN, given $C_b(t)$ as the past trajectory of the user b at time step t, hidden layer function $h(t)$ and output vector $y(t)$ are obtained by

$$h(t) = tanh(b + Wh(t-1) + UC_b(t)), \tag{4.1}$$

$$y(t) = softmax(b_1 + Vh(t)), \tag{4.2}$$

where W, U, V are weight matrices and b, b_1 are bias vectors [10].

The overall steps to process the prior information of user past location with an RNN are summarized as follows:

Step 1: We take the prepared data $[C_1 \rightarrow C_2 \rightarrow C_3 \rightarrow ... \rightarrow C_m]$, which is the past mobility trajectory of the user, as the input of the mobility model.

Step 2: We initialize the weights and biases in the neural network.

Step 3: For the forward direction, at each step, we complete the forward pass of the network for hidden layers.

Step 4: We complete the forward pass for the output layer.

Step 5: We evaluate the prediction error.

Step 6: The calculated error is backpropagated to updated the parameters of the network.

Step 7: We have the future trajectory of the user for the next m time steps ahead $[C_{m+1} \rightarrow C_{m+2} \rightarrow C_{m+3} \rightarrow ... \rightarrow C_{m+m}]$.

Unfortunately, RNNs face the major problem of vanishing or exploding gradients in which the model error increases dramatically after a number of iterations. To solve this issue, gated RNNs are introduced including GRUs and LSTMs [10].

The key difference is the fact that GRU adds two new gates to the network: update $u(t)$ and reset $r(t)$ gates. These gates help the model to only keep the useful information about user past locations. The activation layer function in this case is given by

$$\begin{aligned} h(t) &= u(t-1)h(t-1) + (1 - u(t-1)) \\ &\sigma(b + UC_b(t-1) + Wr(t-1)h(t-1)), \end{aligned} \tag{4.3}$$

where $u(t)$ and $r(t)$ are update and reset gates.

In LSTM, the main difference is adding a cell state unit $s(t)$ and a forget $f(t)$ gate to the model. In this case, the activation layer function is expressed as

$$h(t) = tanh(s(t))q(t), \tag{4.4}$$

where $s(t)$ is cell state unit and $q(t)$ is LSTM output.

Fig. 2 represents the overall procedure of the proposed approach. First, we take the raw data with the length n and convert it into a optimized data with the length of m using line simplification techniques. Next, we feed the prepared data to the RNN-based pattern learning step. Finally, we have the predicted future trajectory of the user in the output.

4.4 Simulation results

In this section, we conduct a series of experiments to analyze the impact of different line simplification methods on user mobility learning. We used a GPS dataset called "Geolife" for our experiments [56]. This dataset contains GPS trajectories (i.e., longitude and latitude) for 182 users. Geolife was collected over a period of almost five years (2007-2012). The length of trajectory (amount of available data) varies from user to user. We use these trajectories as

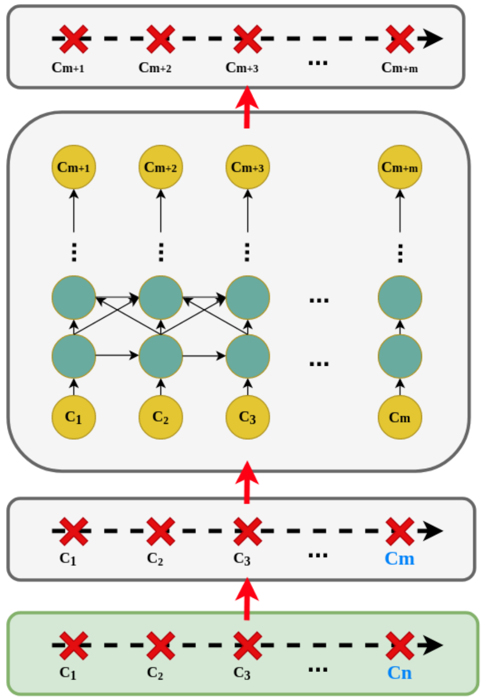


Figure 4.2 Overall procedure of the proposed method.

movement history (i.e., prior knowledge) to learn the mobility behaviour of the user in order to make future location prediction.

We employed Keras library (i.e., open source library in Python) to conduct our simulations, using an Intel core i7-6700k CPU, a 32 GB RAM and 4.00 GHz. In the following, first, we discuss the results regarding data reduction step. Then, we investigate the impact of data reduction on mobility learning.

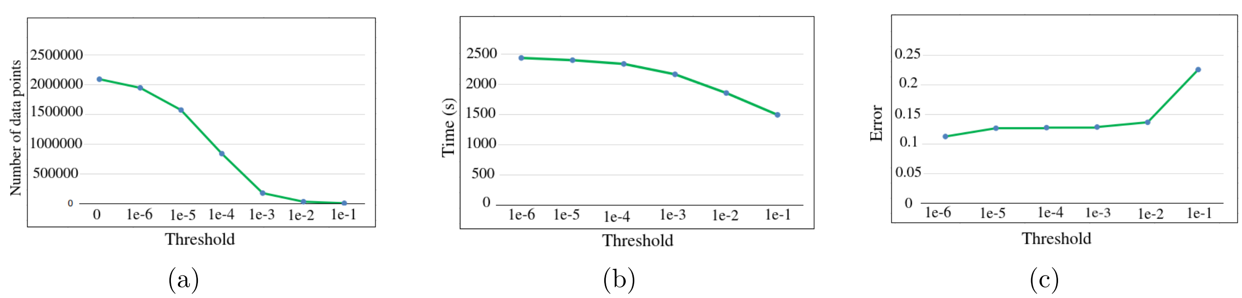


Figure 4.3 Impact of different thresholds on (a) data reduction, (b) execution time and (c) model performance.

4.4.1 Data reduction

Here, we chose the trajectory of the user number 153 for a period of approximately five years (from July 2007 to June 2012). Table 4.1 summarizes the results of applying simplification methods to the user trajectory. We considered $n = 3$ for the naive algorithm and the same threshold for the other methods. The original data contains $2,087,853$ data points. The threshold for each method, amount of the reduced data and execution time for each method are presented in Table 4.1. As is shown, all the methods significantly reduced the amount of original data. D-P has the highest time complexity in comparison with the other algorithms. For D-P algorithm, worst case scenario time complexity is $o(n^2)$.

Fig. 3 depicts the impact of different thresholds of data reduction algorithms on three parameters. Fig. 3(a) shows how data reduction procedure works as we increase the threshold value. It is shown that data reduces noticeably with higher thresholds. In Fig. 3(b), we can see the effect of different thresholds on execution time (i.e., data reduction and mobility learning steps). Logically, as we increase the threshold and obtain smaller dataset, the execution time reduces significantly as well. Lastly, Fig. 3(c) demonstrates that how data reduction affects model performance (i.e., loss value of the learning algorithm). It is evident that reducing data for some thresholds does not result in model performance degradation. However, for higher thresholds (e.g., 1e-2, 1e-1), data reduction has disruptive effect on the performance. It is reasonable due to the fact that too much data is removed and much lower data is fed to the learning algorithm.

Fig. 4 shows a trajectory based on geographical coordinates (i.e., latitude and longitude). It represents the result of applying data reduction technique to choose the necessary data points. Here, we applied D-P algorithm.

Table 4.1 Applying simplification methods to user trajectory.

Methods	Threshold	Reduction	Time (s)
V-W	$1e-4$	2087853 $\rightarrow$ 121294	4.70
R-W	$1e-4$	2087853 $\rightarrow$670009	10.53
nth Point	$n=3$	2087853 $\rightarrow$695952	0.03
Lang	$1e-4$	2087853 $\rightarrow$21420	5.72
D-P	$1e-4$	2087853 $\rightarrow$ 837544	2239.16

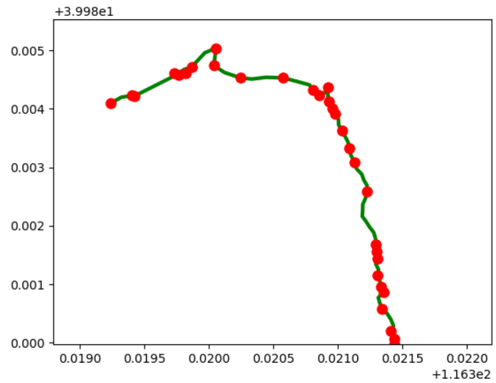


Figure 4.4 Applying line simplification technique to the user trajectory to choose the necessary data points.

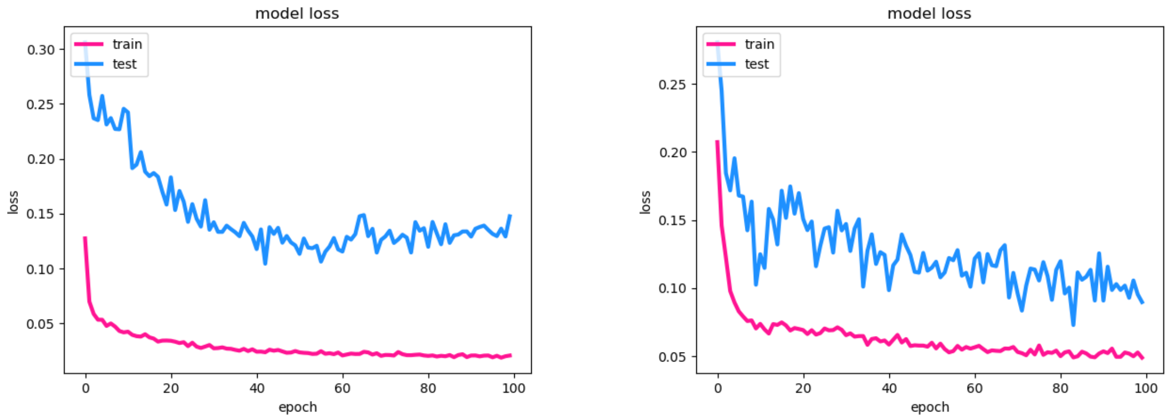
4.4.2 Mobility learning

Here, we chose a part of the trajectory of the user number 153 for a period of approximately 13 months. We deployed a network composed of 3 layers with 100 neurons in each layer. Also, we used a learning rate of 0.001 and deployed Adam optimizer. Table 4.2 presents the impact of each data reduction algorithm on the performance of user mobility learning. We set the value of the thresholds in a way that the reduced data has almost the same size for all the line simplification methods. This is due to the fact that learning performance is highly correlated to the amount of data. When we have approximately the same size of data (length of trajectory) as input for the learning algorithm, we can compare the effect of different data reduction approaches on the overall performance. Time refers to the execution time of the both data reduction and learning algorithm.

In Table 4.2, mean square error (MSE) for the RNN model is reported. We chose this metric since our work is a regression-based task. Considering x, $y(x)$ and $y\hat{(x)}$ as orderly input data, real output and estimated output, for m samples out of k sample, mean square error is calculated by

$$MSE_m(\hat{y}) = 1/k \sum_{i=1}^{m} (y(x) - y\hat{(x)})^2. \tag{4.5}$$

This metric gives us an idea of how well the predictor works. Obviously, lower values of MSE is better and in an ideal situation MSE is equal to zero. As is shown, for this user, mobility learning without any data reduction results in 0.16 error loss value in 2245 seconds. Applying data reduction methods, D-P and Lang obtain respectively the best and worst performances with 0.12 and 0.23. It is evident that exploiting a proper data reduction method, not only does not cause performance degradation but also it may effectively improve it. This implies



(a) Without data reduction. (b) With D-P.

Figure 4.5 Performance of LSTM-based model for train and test data.

that during data reduction phase, some of the noisy data is removed as well. However, time complexity of D-P is noticeably higher than other methods. It is worth noting, the results may be different for a another user with a different mobility behaviour.

Table 4.2 Impact of different preprocessing techniques on mobility model performance.

Methods	Threshold	Reduction	Time (s)	Error
No reduction	–	–	2245	0.16
V-W	$52e-6$	28163	466	0.19
R-W	$85e-6$	28185	432	0.22
nth Point	$n=5$	28111	448	0.17
Lang	$58e-6$	28150	471	0.23
D-P	$16e-5$	28115	582	0.12

Next, we want to investigate the impact of each line simplification approach as a prepossessing step for the three mobility learning models. Table 4.3 summarizes model loss values and execution times for RNN, GRU and LSTM. It is shown that LSTM model achieves the lowest error value among other mobility models. We can see that using data reduction techniques results in small performance degradation and in some cases even performance improvement but it can remarkably reduce execution time.

Fig. 5 shows train and test performances of the mobility learning model without (Fig. 5(a))

Table 4.3 Impact of different simplification techniques on three mobility models.

Method	RNN Error	Time (s)	GRU Error	Time (s)	LSTM Error	Time (s)
No reduction	0.16	2245	0.16	4616	0.13	3057
V-W	0.19	466	0.17	947	0.14	665
R-W	0.22	432	0.18	945	0.16	673
nth Point	0.17	448	0.13	932	0.11	666
Lang	0.23	471	0.20	961	0.19	680
D-P	0.12	582	0.11	1030	0.10	733

and with (Fig. 5(b)) data preprocessing. Here, D-P algorithm is deployed as reduction technique to remove the irrelevant data. As is shown, both train and test errors dramatically decrease after almost 40 epochs. It is noticeable that the average test error is much lower when using prepared data. Also, we can see that the model does not overfit and trains rapidly.

4.5 Conclusion

In this paper, we investigated several line simplification approaches as a potential preprocessing step for a mobility learning model in mobile networks. We proposed to deploy a line simplification method to reduce the irrelevant data from a big dataset and keep only the appropriate data for the mobility learning model. We analyzed the performance of the five data reduction techniques including nth point, R-W, Lang, V-W and D-P. Then, we used the reduced version of data for learning user mobility behaviour. We exploited RNN, LSTM and GRU models to find the repetitive patterns in user's movement history and predicted future trajectory of the user. Simulations results showed the effectiveness of the data reduction step. Using reduced data results in mobility model performance improvement.

Moreover, it can remarkably reduce the execution time of the train and inference phases since we are dealing with smaller dataset. D-P algorithm produced the lowest mobility model error performance and the highest time complexity with respectively 0.10 and 733s while the model loss and execution time using the original dataset results in orderly 0.13 and 3057s. Among learning algorithms, LSTM model results in the most accurate trajectory prediction.

CHAPTER 5 ARTICLE 2: A BIDIRECTIONAL TRAJECTORY PREDICTION MODEL FOR USERS IN MOBILE NETWORKS

Authors: Nasrin Bahra and Samuel Pierre
Status: Accepted for publication in IEEE ACCESS Journal

Abstract

Future mobile networks are envisioned to have critical limitations in terms of latency, energy usage, capacity and network resources since these networks are expected to become extremely dense and complex. The rapid enormous advances in recent technologies such as Internet of Things (IoT) highlights the urgent need for network performance enhancement as well. To this end, self-organizing networks are a promising solution to push the network performance to the next level. These scalable networks can dynamically adapt to possible changes in the network. Smart mobility management, in particular mobility prediction, is a subsection of self-organizing functions which are mainly based on the machine learning techniques. In this paper, we propose to estimate user's future trajectory using machine learning approaches for a better network management. We propose a novel bidirectional trajectory prediction model called BTPM to model the user mobility behavior. The proposed method exploits the potential benefits of bidirectional gated recurrent unit (GRU) for having an accurate prediction. Moreover, we introduce a data preprocessing phase to obtain better results with significantly lower execution time. The proposed approach takes full advantage of data analysis in both directions (backward and forward) in order to provide a long-term prediction and model user's mobility even with complex patterns. Experimental results show that the proposed bidirectional approach significantly improves the performance of the mobility predictor in terms of model accuracy, robustness and execution time. It achieves a model error of 0.014 and decreases the execution time up to 97%.

IEEEkeywords

Recurrent neural networks, Trajectory prediction, Mobile networks

5.1 Introduction

Future mobile networks have critically limited available resources. Although, mobility adds a wide range of great services for users in the network, it can cause several critical issues at the same time. Bandwidth restriction, high-speed packet transmission, frequent handovers, increase in call dropping probability and communication reliability pose serious challenges for future network management which all lead to quality of service (QoS) degradation for users [12]. A deep understanding of network traffic behavior is of crucial importance in today's rapidly growing mobile networks.

Mobility prediction is a promising key enabler for the intelligent mobility management in self-organizing networks (SONs). SON is an emerging paradigm that is based on adaptive and independent network management. It is able to learn from past experiences and to improve the network performance based on the user prior knowledge. Mobility management is a part of self-optimization function in self-organizing networks that can address the aforementioned issues and meet the demands of future mobile networks [4]. The main idea is to estimate user's movement trajectory using machine learning approaches for a better network management.

From another point of view, according to Cisco, there will be approximately 5.5 billion mobile users by the year 2020 [57]. The exponential growth in the number of mobile devices results in a huge mobility data generation by the mobile users every day. This data can be effectively exploited to extract valuable insight from them which later can be helpful in many ways.

Some of the potential research areas that can significantly benefit from location awareness are: resource allocation techniques, optimized handover decision, call admission control optimization, bandwidth reservation, routing mechanisms and provision of high QoS. In particular, when a mobile node moves between different access points with an active session, seamless connectivity without interruption is essential while user is moving. Location awareness can make handover process transparent to mobile users in dense networks and effectively eliminate the problem of QoS degradation [5]. Moreover, location information serves useful purposes for 5G networks in terms of energy consumption and latency reduction [58]. Massive increase in the number of mobile devices and connected objects stresses the needs for lower latency and high data rates.

Despite the efforts that have been made in the existing works on the mobility prediction techniques, a reliable robust mobility model with high accuracy that can cope well with scalability issues is still missing. There are still several research aspects regarding user trajectory prediction that can be further improved such as obtaining a long-term reliable prediction, achieving high prediction accuracy dealing with complex and irregular patterns, low time complexity, preventing excessive assumptions for the network or model, balancing the trade-off between the model complexity and accuracy. Most of the related works in this field are based on Markov models and neural networks. In Markov-based approaches, first order Markov models are simple with low accuracy, whereas Markov models with higher orders are complex with acceptable accuracy that rises the problem of accuracy-complexity trade-off. In traditional neural networks, input samples are assumed to be independent of each other which is a questionable assumption. As a result, they cannot learn long-term dependencies and process sequential data. This can have disruptive impact on prediction accuracy. We believe that a mobility model is needed that can take advantage of numerous merits of neural networks as well as eliminating the memory issue. More precisely, the model should be able to learn from the available prior knowledge and should be able to adopt to probable changes without manual effort. The main challenge is to introduce a novel trajectory prediction technique that emphasizes the need for a generic approach for all users with any degree of predictability.

In this paper, to address the shortcomings of the existing methods, we propose a mobility model that can effectively maximize the impact of data dependency of prior knowledge about user's past experiences on the mobility model. This paper is an extension to our previous work in [59]. Here, we propose a bidirectional trajectory prediction model called BTPM based on bidirectional gated recurrent unit (GRU) [60], [61]. In BTPM, first, we eliminate the unwanted raw data using Douglas-Peucker (D-P) algorithm [52]. Then, noisy data is discarded from user trajectory unsing Kalman Filter. Finally, we feed the data to the inference model which is based on bidirectional gated recurrent unit (GRU). The major contributions of our work are summarized in the following:

- Proposing a two-phase user mobility sequence preparation specifically introduced for mobility prediction purpose. To the best of our knowledge, this is the first time that this specific user trajectory preparation is introduced and deployed for trajectory prediction. The key feature of this procedure is huge time complexity reduction while improving the prediction accuracy and reducing the noise level. Kalman filter and line simplification algorithms are deployed to reduce the noise and obtain significantly lower execution time. One major problem of using deep learning methods is high execution time dealing with large datasets. Therefore, BTPM benefits from numerous advantages of deep learning while eliminating the complexity issue. This phase results in reducing the execution time up to 97%

- Proposing a bidirectional approach to extract mobility patterns by deeply considering the correlation in the user mobility trajectory. To the best of our knowledge, this is the first time that a bidirectional neural network is deployed to investigate the user trajectory prediction. The proposed bidirectional trajectory prediction model (BTPM) is based on bidirectional gated recurrent unit (BiGRU) that enables highly accurate predictions with analyzing both forward and backward correlations in the user past movement history. Another distinguishing feature of BTPM is achieving a robust performance dealing with different users with different degrees of predictability. It improves the model error up to 65%.

- Conducting a comprehensive series of experiments using three datasets and different users with different metrics. User mobility behavior and length of available data for each user are two factors that are inevitable for mobile users in reality. We considered the combination of these factors to evaluate different mobility models in terms of being practical in more close to real-world scenarios.

The rest of this paper is organized as follows. In Section II, we overview some of the related works in this field and provide a list of unsolved open problems. A preliminary to mobility prediction based on neural networks is provided in Section III. The proposed method for user trajectory prediction is presented in Section IV. Section V provides the experiment results. Section VI concludes this paper.

5.2 Related works

Trajectory data mining and mobility prediction have been extensively studied and there are a lot of wide-ranging works about them in literature [9, 12, 13, 47, 57, 62]. In general, almost all the methods in the mobility prediction concept can fall into two main categories [7]: (1) User future path/trajectory prediction: It is a regression task that predicts the geographical location of the user in the next time step. The prediction is the next geographical location point of the user (e.g., longitude and latitude). One popular example is user trajectory prediction (e.g., (lon1, lat1), (lon2, lat2), (lon3, lat3)) [63], [64]. (2) User next location prediction: It is a classification task that predicts the place that user will visit in future. In the dataset, each location has an specific location id. The prediction is the next location id of the user. One popular example of its application is point-of-interest recommendation methods (e.g., restaurant, park, gas station, ...) [1], [41], [48]. Our work is a part of the first category. In the following, we review some of the recent works of both groups.

There are several works on user trajectory analysis. In [65], authors proposed a semantic model that converts raw movement history to semantic trajectories. In [34], authors proposed a user movement prediction method considering user geographic, temporal and semantic intentions. In [35], authors proposed an approach for trajectory representation learning using road networks and mobile user trajectories. A trajectory prediction model was proposed in [66]. Using similarity metrics, it predicts user's trajectory based on finding similar informative segments in available initial user trajectory.

Learning-based mobility prediction approaches have attracted a tremendous attention in the recent years due to the popularity of machine learning techniques. A mobility prediction method was proposed in [14], it predicts users trajectory based on an online learning process using mobility data history. The main objective of this work is to decrease handover-related signaling load and to reduce the latency using a predictive handover scheme. In [24], a long-term mobility estimation scheme was introduced to extract the mobility patterns based on previous trajectories. Trajectories were clustered based on the trajectory similarities for the offline learning process. The main goal of this work was to estimate a group of possible future paths for each individual in the network. In [20], a supervised learning method was used for location prediction since no data labeling is required when using check-in databases. One major problem of methods based on the standard neural networks and machine learning techniques is that all of them rest on a questionable assumption of independency between data input samples. They do not consider the data correlation in the sequential data.

In recent years, some mobility models were proposed based on recurrent neural networks (RNNs). In [50], an RNN-based method is proposed: spatial temporal recurrent neural networks (ST-RNN). This model considers temporal and spatial dependencies in each layer using time-specific and distance-specific transition matrices. Furthermore, a linear interpolation method was used for training the transition matrices. Inspired by ST-RNN, authors in [67] and [29] proposed approaches for destination prediction based on long short-term memory (LSTM) models using spatial-temporal information in user's movement history. In [68], given the history of call detail records of each user, authors proposed an LSTM-based model that predicts the next location of the user based on the cell tower IDs. Mobility models based on RNNs and LSTMs consider the information regarding to the past time steps and their correlations only up to the current step. However, if the mobility model could make the prediction based on the whole input sequence (past and future), it can significantly improve the performance.

There are many works that were proposed different mobility models motivated by location-based social networks (LBSNs). A huge amount of check-in data is collected from LBSN (e.g., Foursquare) that can be exploited in mobility pattern learning and location-based services. In [44], authors proposed an RNN-based deep context-aware model that has three main parts: (1) The first part captures the long-term dependency in user mobility records, (2) The second part extracts user's location semantics and periodic pattern, and (3) The third part extracts social and temporal context. Using this deep model, they could predict the next point-of-interest (POI) of the user. The authors in [69] and [45] introduced two GRU-based models for the next POI prediction. Using LBSN datasets, a context-aware trajectory learning method was proposed in [70] to capture different characteristics of human mobility behavior.

A huge part of localization techniques and mobility predictors are based on different types of Markov models. The movement path of each user is modelled based on his/her record of visited cells or geographic information and Markov predictor is deployed in order to predict the future cells [11]. There is an important trade-off in Markov models that needs to be given a careful consideration. While higher orders of Markov models result in a better prediction accuracy, a huge number of model parameters are needed in these models. For an Mth-order Markov chain with d states, $d^{M-1}(d-1)$ parameters are needed. In [16], a Markov-based trajectory prediction technique called hidden Markov model-based Trajectory Prediction (HMTP) was proposed that can anticipate a complete path. Algorithm parameters are selected dynamically and a partitioning algorithm was introduced.

A location prediction system using GPS data was proposed in [31]. This approach contains three main parts: location extraction, location recognition and location prediction. Gaussian-means, k-nearest neighbor and HMM methods were deployed in these parts. Another Markov-based method was proposed in [27]. This approach consists of two main parts: destination prediction model (DPM) and path prediction model (PPM). DPM is responsible for estimating the final destination using second-order Markov chain. Then, the path toward the user's destination is anticipated by PPM.

In a traditional Markov model, states must be chosen from a large state space imposing restriction on the inference model. Hence, standard operations get less feasible in a hidden Markov model as hidden states increase. Therefore, each hidden state only depends on a few number of past states.

It seems that Markov models tend to fail when the number of locations increases and can be more accurate when dealing with short-term dependencies with few number of locations. Past researches have shown that Markov models fall short of modelling human mobility [17]. Furthermore, Markov models are too sensitive and can easily be affected by user mobility pattern changes. They struggle to cope well with drastic changes in mobility behavior of users.

5.3 Preliminary

In this section, we provide a preliminary to mobility prediction models based on neural networks. . Traditional neural networks are not able to store information while processing new inputs. Therefore, input samples are assumed to be independent of each other which is a questionable assumption for applications in which inputs are not independent of each other and this dependency has an impact on the learning process.

Recurrent neural networks are a special kind of neural networks that can capture the dependency between input samples and therefore are suitable for sequential data. RNNs solve the mentioned problem of traditional neural networks since they include loops in the network and have a memory to store the information about past locations.

For an RNN at each time step t with $L(t)$ (i.e., user location at time step t) as input, $y(t)$ as prediction output and $h(t)$ as hidden states, the forward propagation equations are given in the following. Hidden layer activation function is:

$$h(t) = tanh(b_1 + Wh(t-1) + UL(t)). \tag{5.1}$$

A softmax operation is used to obtain the output vector such that

$$y(t) = softmax(b_2 + Vh(t)), \tag{5.2}$$

where b_1 and b_2 are bias vectors. Also, U, V and W are weight matrices that decide to what extend each input should influence the location prediction [71].

However, RNN has the fundamental problem of vanishing or exploding gradients. The problem of exploding or vanishing gradients mostly happens when the network is large and has many layers. Simple RNNs are not able to remember very old information from the beginning of the sequence. In other words, RNN requires a very long time to learn long-term dependencies. Hence, this mobility model is not suitable for capturing users movement pattern from old states information.

The idea of gated RNN is a potential solution for the aforementioned problem that has been proved to be highly successful. The core idea is based on the fact that we do not necessarily need all the past information over a long period. In other words, for some states, it could be even more effective if network forget the previous states and eliminates the old information for making future decisions. When the network is able to learn which states to keep and which ones to forget, we have a gated RNN. Two popular forms of gated RNNs are long short-term memory and gated recurrent unit [61, 72].

For LSTM, the key difference is adding a *cell state* unit $s(t)$ to the RNN structure. Cell state keeps a record of current and previous user's locations information. This information can be regulated by a $forget\ gate$ unit $f(t)$ in each LSTM cell. Therefore, LSTMs can effectively capture the long-term dependencies and are an appropriate model for processing time series sequences.

The next gated RNN is GRU, a simplified version of LSTM network, that has two main gates: $updategate$ $u(t)$ and $resetgate$ $r(t)$. The update equations for GRU is expressed as follows:

$$\begin{aligned} h(t) &= u(t-1)h(t-1) + (1 - u(t-1)) \\ &\sigma(b + UL(t-1) + Wr(t-1)h(t-1)), \end{aligned} \tag{5.3}$$

where $u(t)$ and $r(t)$ are defined by

$$u(t) = \sigma(b_u + U_u L(t) + W_u h(t)), \tag{5.4}$$

$$r(t) = \sigma(b_r + U_r L(t) + W_r h(t)), \tag{5.5}$$

where σ denotes Sigmoid function that returns a value between 0 and 1. W_r and W_u are recurrent weight matrices. The update gate decides which part of the data to keep for future time steps. Sigmoid function of the reset gate determines what information to remove from the data. Mathematical notations for RNN and GRU are summarized in Table 6.1.

5.4 Bidirectional trajectory prediction model (BTPM)

In this section, we explain the proposed approach for predicting the user future trajectory.

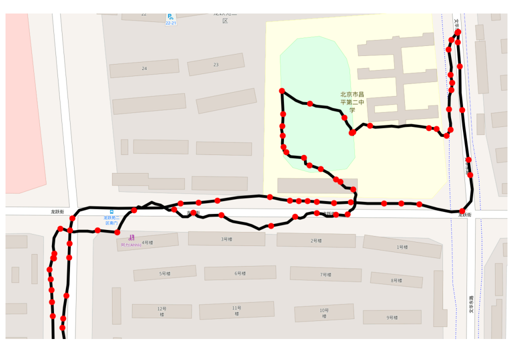


Figure 5.1 User raw GPS trajectory. Red points represents the necessary points that forms the path.

Table 5.1 Mathematical notations for RNN and GRU.

$L(t)$	User location	$y(t)$	Prediction output
b_1, b_2, b_u, b_r, b	Bias vectors	$h(t)$	Hidden states
U, U_u, U_r	Input weight	$u(t)$	Update gate
V, W, W_u, W_r	Recurrent weight	$r(t)$	Reset gate

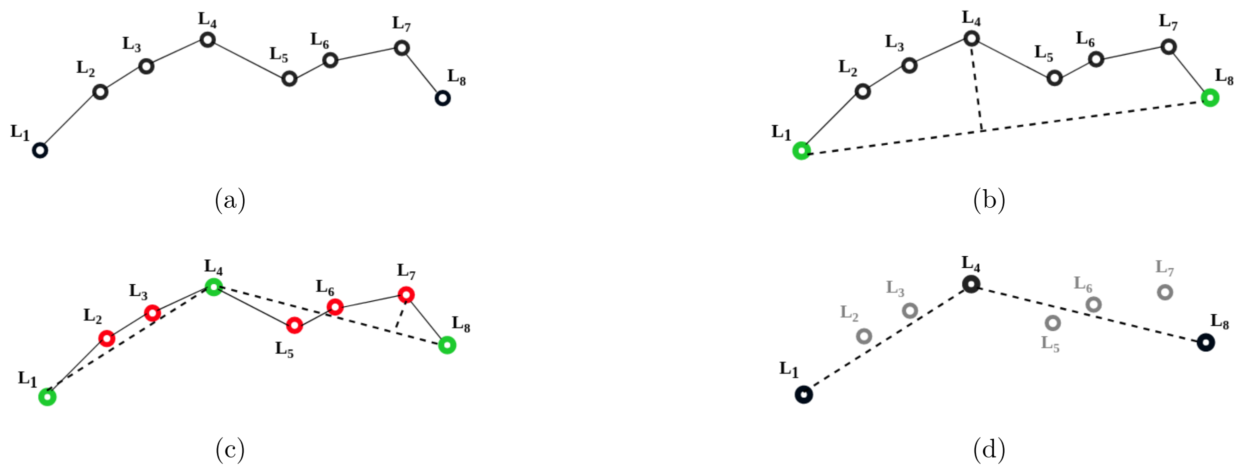


Figure 5.2 Trajectory simplification. (a) Original trajectory, (b) Keep L_4 in the trajectory, (c) Eliminate L_2, L_3, L_5, L_6 and L_7 and (d) Simplified trajectory.

Problem (Trajectory prediction): Given a trajectory of a user for the previuos n time steps (T_{user}), we want to predict the next n time steps ($\tilde{T}_{user}$). Where $T_{user} = (L_{t-n}, ..., L_{t-1}, L_t)$ and $\tilde{T}_{user} = (L_{t+1}, ..., L_{t+n})$.

The primary step of the mobility learning is choosing an appropriate type of mobility data as the movement history of a user. One challenging part of deploying user movement history is the fact that this information needs preprocessing before we can use it for our mobility model. It is crucial to convert the raw data (i.e., user movement history) into a clean dataset for the analysis. In fact, with the right data, even simple methods are able to provide valuable insights from the data. We choose GPS trajectories as the user movement history.

Trajectory Definition: A trajectory for the user i is represented as $T_{useri} = (L_{t-n}, ..., L_{t-2}, L_{t-1}, L_t)$ which is a sequence of time-stamped location points. We denote L_t as $L_t = (x_t, y_t)$ where x_t and y_t provide the geographical information of the user location at time t (i.e., longitude and latitude).

We carefully analyzed the input data of our mobility model and try to deal with the existing underlying issues related to the raw datasets that can have damaging impacts on the accuracy of the mobility learning. We realized there are two fundamental issues that need to be solved before training and inference phases: redundancy and noisiness. Raw GPS dataset is composed of user location information at almost each second with a high sample rate. All of this data is not necessarily needed for our purpose. We propose to deploy a line simplification method to reduce the irrelevant data from user raw trajectory and only keep the appropriate data for the mobility learning model from a big dataset. The core concept of line simplification approaches is to keep only the part of data that is essential to form the trajectory and delete the rest of the data points between them. This data reduction has the potential to substantially lower the processing time. Therefore, in BTPM, our first step is to remove the irrelevant data from the user trajectory. We deploy D-P algorithm to eliminate the unwanted data. Figure 5.1 depicts a mobile user GPS trajectory. The black line represents the actual path of the user and the raw information of user trajectory. Red points on the line are the points that we keep from the dataset and remove the rest. In this way, we have a similar trajectory with much fewer number of points.

Given the trajectory of a user with the length of m $[L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow ... \rightarrow L_m]$, we want to convert it to a simplified trajectory with the length of n $[L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow ... \rightarrow L_n]$, where $n < m$.

Data reduction procedure has a global routine in which the whole user trajectory is considered during the data reduction procedure. Figure 5.2 represents the main steps for choosing the determinant location points to keep from the original user trajectory. Figure 5.2(a) shows an illustration of a user original trajectory, where $L_i = (lon_i, lat_i), i \in \mathbb{N}$ denotes user locations in the trajectory in terms of longitude and latitude. This example considers 8 locations (i.e., $i = 8$). In Figure 5.2(b), we draw a line between the first and last points (i.e., L_1, L_8) and select the farthest location point from the line (i.e., L_4). We keep this location in the trace since its distance from the line is more than a predefined threshold. Similarly, in Figure 5.2(c), we eliminate L_2. L_3,L_5, L_6 and L_7 from the trajectory since their distances are lower than the predefined threshold. Figure 5.2(d) shows the final simplified trajectory. This procedure is repeated for all the locations in the trajectory. It will result in almost the same performance accuracy but with a remarkably lower execution time.

In our work, we deal with data produced by real users, so this data reflects the noisy reality as an inevitable consequence. It is vital to deal with this noise and remove it from data in the preprocessing stage. Therefore, as a subsequent step, we exploit Kalman filter to reduce the noise in the user data that commonly happens during GPS measurement process [73], [74]. This technique is composed of a prediction process followed by an update (or correction). Firstly, Kalman filter estimates the next state of the process. Then, it updates the estimation based on the noisy observations (i.e., measurements). These updates are performed in two distinct categories: time update and measurement update. Mathematical equations for updating time (also known as prediction stage) are used in order to update the process time step and calculating a prior estimation of the next step. Measurement updates (also called as correction stage) are in charge of providing feedback based on the observations that we already have and improve the estimation. Given some noisy information, Kalman filter tries to decrease the level of uncertainty or error in the data. The state equation of such process is

$$x_k = Bx_{k-1} + Cu_{k-1} + \alpha_{k-1}, \tag{5.6}$$

and the measurement equation is

$$M_k = Hx_k + \beta_k, \tag{5.7}$$

where x_k and M_k are state and measurements vectors at time step k. u_k is the control input vector. Matrix B shows the relationship between two consecutive time steps and Matrix C indicates the relationship between the control input and the states. Measurement and process noises are respectively shown by β and α. Measurement matrix is represented by H.

Therefore, we smooth the GPS trajectory using Kalman filter. This effectively leads to a noticeable decrease in the noise level. Kalman filter has the potentials to predict the state of a process with the minimum error. Here, we provided a high-level explanation for Kalman filter algorithm. The details and corresponding equations for prediction-correction process in Kalman filter algorithm are given in Appendix. It is worth mentioning that we directly apply these techniques to the raw GPS dataset. Then, the obtained data after preprocessing phase is used as the input for the mobility model.

Concurrently, we generated a tailored data sequence of user past locations suited to our specific framework. This can have a profound impact on the final results, since the model is learning from the initial data. We feed the data as an input for the mobility model.

In the previous works, for each state $L(t)$ at time step t for a user, only the forward dependencies of information from past time steps until present $\{L(t-n), ..., L(t-2), L(t-1)\}$ is considered for predicting the next step. However, BTPM considers both the forward and backward dependencies of the user past movement history. It has both forward and backward paths starting from beginning and end of the sequence using the concept of bidirectional neural networks. When network can make a decision based on the whole input sequence (past and future), it would tremendously enhance the performance of the mobility model.

In the forward path, we investigate user's movement history in chronological order and basically deal with the past steps' information regarding user's previous location points (i.e., $\{L(t-n) \rightarrow L(t-n+1) \rightarrow L(t-n+2) \rightarrow ... \rightarrow L(t-2) \rightarrow L(t-1)\}$). Hidden layer activation function for forward path is given by

$$\overrightarrow{h}(t) = tanh(\overrightarrow{b_1} + \overrightarrow{W}\overrightarrow{h}(t-1) + \overrightarrow{U}\overrightarrow{L}(t)). \tag{5.8}$$

We can add a backward path to the network and convert the unidirectional network into a bidirectional network. Therefore, we can take advantage of useful information from the future time steps in order to make an even more accurate decision. In the backward path, user's movement history is analyzed in reverse chronological order and user locations correlation is investigated in backward path (i.e., $\{L(t) \rightarrow L(t-1) \rightarrow L(t-2) \rightarrow ... \rightarrow L(t-n)\}$). Hidden layer activation function for backward path is given by

$$\overleftarrow{h}(t) = tanh(\overleftarrow{b_1} + \overleftarrow{W}\overleftarrow{h}(t-1) + \overleftarrow{U}\overleftarrow{L}(t)), \tag{5.9}$$

and output vector is expressed as

$$y(t) = f(b_2 + V[\overrightarrow{h}(t), \overleftarrow{h}(t)]). \tag{5.10}$$

This model is able to deeply investigate user past locations correlations and controls which locations to keep and which ones to forget using update and reset gates. Figure 5.3 and Algorithm 1 show the proposed architecture in detail. First, we have a sequence of spatial time series data as input for the model. Then, in the preprocessing stage, we apply some techniques to clean the data. The input data is desirably manipulated in order to reduce the number of unwanted observations ($n < m$). Next, the reduced version of the data passes through a Kalman filter to generate a clean data with the minimum level of noise. After data preparation, using bidirectional neural network, we train the model with BTPM and extract user mobility behavior.

Table 5.2 presents the substantial difference between the proposed approach and the popular methods towards trajectory prediction. BTPM gives us a distinct advantage of investigating user movement history not only in forward direction but also in backward direction. In other words, user's current location is highly dependent on the previous locations (forward direction) and also the next locations (backward direction) as well. The proposed model observes the correlation between locations in both chronological and reverse orders. This can have a profound impact on our analysis and gives us a deeper understanding of user mobility behavior with a significantly lower execution time.

Here, we want to provide an appropriate clarification on how exactly BTPM works to make a prediction. For example, imagine we have a sequence containing the previous 50 geographical location points of a user (i.e., $n = 50$) and we want to predict the trajectory of the next 50 future locations of the user. After data preparation procedure, the following steps should be completed for training and inference phases:

$L_1, L_2, L_3, ..., L_{50} \xrightarrow{?} L_{51}, L_{52}, L_{53}, ..., L_{100}$

Step 1: We feed the information of the first to fiftieth locations to the network and obtain the

Table 5.2 User past locations dependencies analysis using different approaches.

Approach	Procedure
Mobility models based on Linear regression	$L_1, L_2, L_3, ..., L_n \Longrightarrow$ Future locations prediction
Mobility models based on RNN/LSTM/GRU	$L_1 \rightarrow L_2 \rightarrow L_3 \rightarrow ... \rightarrow L_n \Longrightarrow$ Future locations prediction
BTPM (proposed)	$L_1 \rightleftarrows L_2 \rightleftarrows L_3 \rightleftarrows ... \rightleftarrows L_n \Longrightarrow$ Future locations prediction

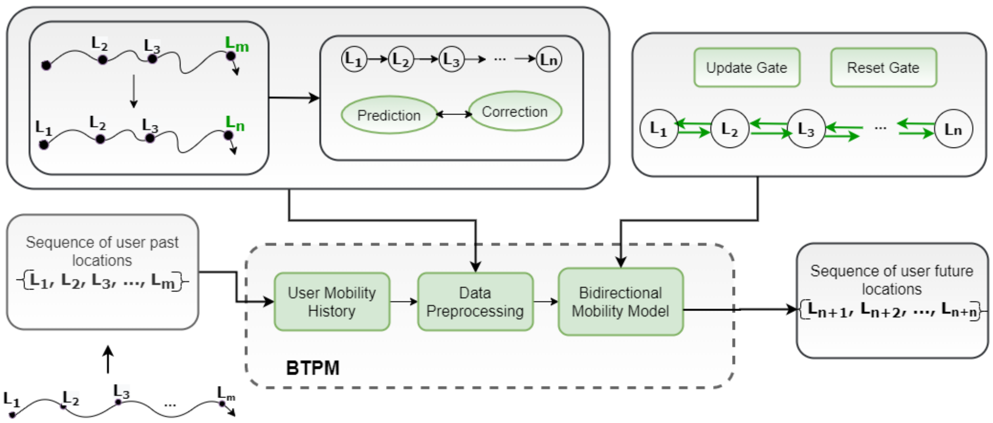


Figure 5.3 The proposed approach.

information for the second to fifty-first locations in output layer.
$[L_1 \rightleftarrows L_2 \rightleftarrows L_3 \rightleftarrows ... \rightleftarrows L_{50}] \Longrightarrow L_2, L_3, ..., L_{50}, L_{51}$

Step 2: Network takes the output of the last step as input (i.e., $L_2, L_3, ..., L_{50}, L_{51}$) and we have the third to fifty-second locations as results.
$[L_2 \rightleftarrows L_3 \rightleftarrows L_4 \rightleftarrows ... \rightleftarrows L_{51}] \Longrightarrow L_3, L_4, ..., L_{51}, L_{52}$

Step 3: Again, given the output of the last step, network predicts the forth to fifty-third locations.
$[L_3 \rightleftarrows L_4 \rightleftarrows ... \rightleftarrows L_{51} \rightleftarrows L_{52}] \Longrightarrow L_4, L_5, ..., L_{52}, L_{53}$

Step 4: We continue the prediction re-iteratively as previous steps.

Step 5: The final step is to predict the trajectory of the 51th to 100th locations.
$[L_{50} \rightleftarrows L_{51} \rightleftarrows ... \rightleftarrows L_{98} \rightleftarrows L_{99}] \Longrightarrow L_{51}, ..., L_{99}, L_{100}$

Therefore, for making a prediction, we investigate not only the forward locations correlations in the user movement history but also the backward consecutive locations dependencies as well.

Next, we want to perform a complexity analysis of the proposed approach. For data reduction phase, worst case scenario time complexity is $O(n^2)$. For noise removal phase, computational complexity is $O(n^{2.376})$ [75]. Finally, for the learning phase, time complexity is $O(2)$. Therefore, the overall complexity of BTPM is $O(n^{2.376})$. We have conducted simulations to evaluate the time complexity of the proposed approach even with large input data in Section 4.4 (see Figure 5.11).

5.5 Experimental results

In this section, we present results of a series of experiments that have been carried out to evaluate the performance of the proposed approach and to highlight the effectiveness of our model. To this end, first, we describe the data that has been used for the simulations in Section A. Then, we conduct our experiments in several phases including assessing the impact of data preprocessing, performance evaluation of trajectory prediction models, comparing the proposed approach with the existing techniques, robustness of the proposed method, and finally extending the results to the cellular networks.

For comparison, we implemented some of the most related previous methods including:

- Linear regression: A model based on linear regression is a basic method that take each user location independently [71].
- Support vector regression (SVR): A SVR-based model is based on machine learning techniques and considers each user location independently. It makes the prediction using kernels [76].
- Recurrent neural networks: RNN-based model considers user past locations to make a decsion [77].
- Long short term memory: LSTM-based methods analyze past locations using state unit and forget gate [72].
- Gated recurrent unit: GRU-based methods predict future trajectory using update and reset gate to control user movement history [61].

We considered the same framework and configuration for implementing all the methods in order to have a fair comparison. We deployed Keras library for our simulations that is a neural network library in Python (https://keras.io/) and all the simulations were executed using Intel core i7-6700k CPU with 4.00 GHz and 32 GB RAM. We used a neural network with 3 layers (i.e., one hidden layer) and 100 neurons in each layer for all the mentioned methods. We set the drop-out rate to 0.2 (i.e., 20 percent of the connections are randomly dropped in the training phase). Also, we set the learning rate to 0.001 and used Adam optimizer in our experiments.

Next, we choose the appropriate metrics for our work. Basically, our work is predicting user mobility trajectory. The proposed model for trajectory prediction is based on regression-based machine learning methods. Common metrics to evaluate regression-based machine learning algorithms are mean square error (MSE), mean absolute error (MAE) and root mean square error (RMSE) [78], [41].

We used these three metrics to evaluate and compare the performances of all the methods. These metrics help us describe how effective each technique is. Given k samples from the total of n samples, MAE is expressed as:

$$MAE_k(\bar{y}) = 1/n \sum_{i=1}^{k} |y(x) - \bar{y}(x)|, \tag{5.11}$$

where x, $y(x)$ and $\bar{y}(x)$ as respectively model input, actual output and predicted output. MSE is defined by

$$MSE_k(\bar{y}) = 1/n \sum_{i=1}^{k} (y(x) - \bar{y}(x))^2, \tag{5.12}$$

and RMSE is given by

$$RMSE_k(\bar{y}) = \sqrt{1/n \sum_{i=1}^{k} (y(x) - \bar{y}(x))^2}. \tag{5.13}$$

Lower values for MAE, MSE and RMSE are desirable and ideally the best value for a perfect model is zero for these errors. Root Mean squared error is very sensitive to high errors. In other words, RMSE value noticeably increases when error variance is high (i.e., there is a major difference between the lowest and highest error value) in comparison with MSE and MAE values.

5.5.1 Data Description

For conducting the experiments, three different GPS datasets were considered as user mobility data. Using different mobility data sources can guarantee the robustness and generality of a mobility model. An efficient approach should be able to appropriately predict user trajectory based on different mobility data with an acceptable accuracy. Information regarding the datasets are provided in the following:

- Dataset 1: Geolife [79]
  This GPS dataset was collected by 182 persons during five years (from 2007 to 2012). This

Table 5.3 Impact of changing RDP threshold.

RDP threshold	Reduction	Time(s)	MSE loss
$\varepsilon = 1e-6$	2087853 → 1942782	2432	0.112
$\varepsilon = 1e-5$	2087853→ 1570032	2395	0.126
$\varepsilon = 1e-4$	2087853→ 837544	2333	0.127
$\varepsilon = 1e-3$	2087853→174978	2161	0.128
$\varepsilon = 1e-2$	2087853→ 30668	1852	0.136
$\varepsilon = 1e-1$	2087853→ 4346	1491	0.225

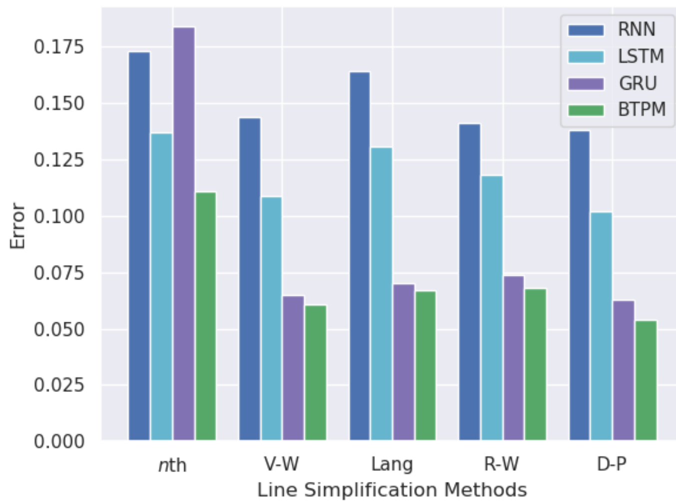
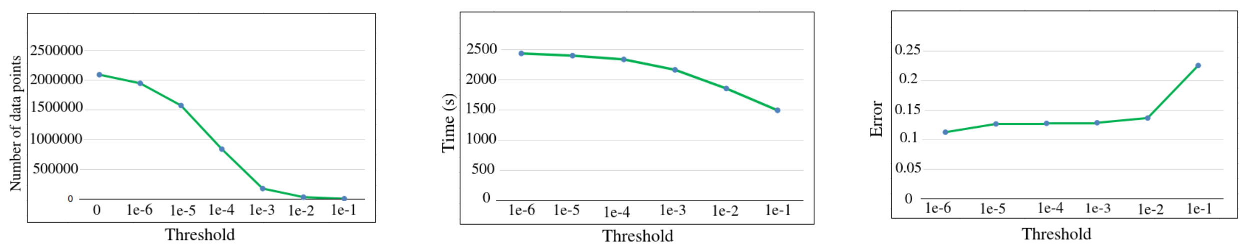


Figure 5.4 Trajectory simplification.



(a) (b) (c)

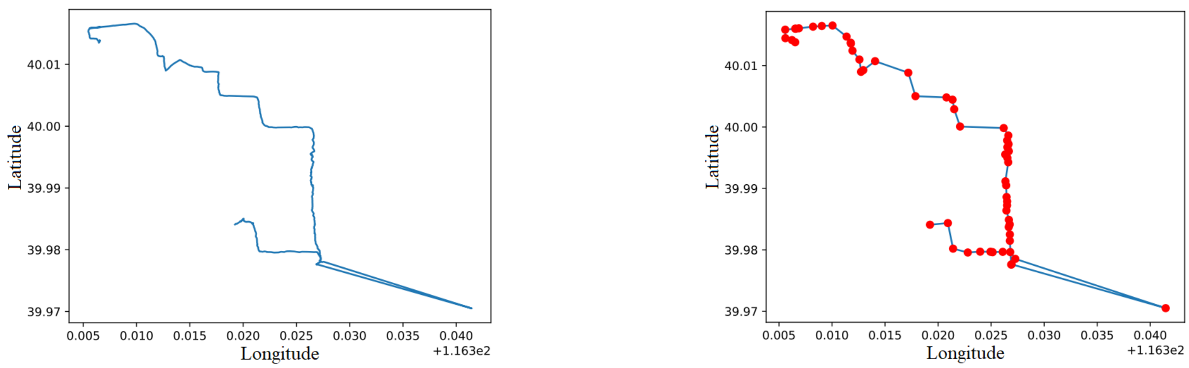
Figure 5.5 Impact of different thresholds on (a) data reduction, (b) execution time and (c) model performance.

dataset contains a set of points representing several values such as latitude, longitude and altitude. The total distance of this trajectory is 1,292,951 kilometres. A large amount of this data is generated in Beijing, China.

- Dataset 2: Open street map (OSM)
  OSM provide public access to all GPS tarcks ever uploaded by different users (https://www.openstreetmap.org/traces). It is one of the largest GPS traces dataset that is publicly available.

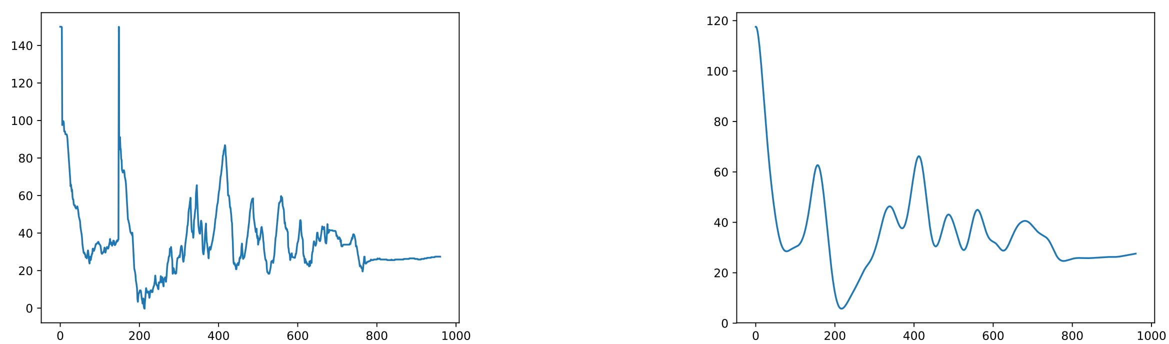
- Dataset 3: T-Drive trajectory [80, 81]
  This dataset provides GPS trajectories of 10,357 taxis over a period of one week for a total distance of almost 9 million kilometres.

For the experiments, 80% of the dataset was used for model training and the rest for the inference phase.



(a) (b)

Figure 5.6 Removing unnecessary observations in data cleaning phase. (a) Original user trajectory and (b) Simplified trajectory.



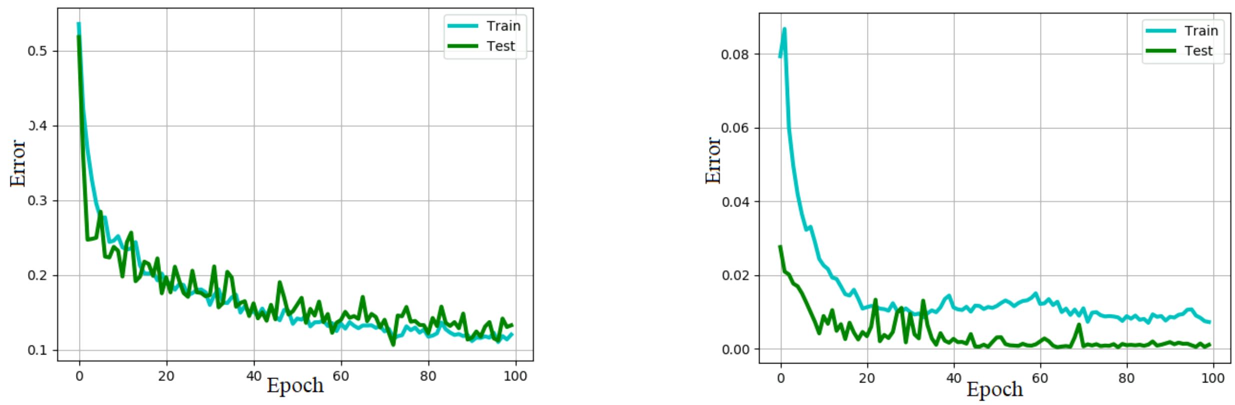
(a) (b)

Figure 5.7 Noise reduction in data cleaning phase. (a) Noisy movement trajectory and (b) Smoothed movement trajectory.

5.5.2 Data preprocessing

This section mainly analyzes the impact of applying preprocessing techniques to the raw GPS data. User trajectory data is loaded and then the preproccessing techniques are applied to the data frames. We deployed python panda library (http://pandas.pydata.org/) to store the data into several data frames.

Figure 5.4 represents the impact of different line simplification approaches on the mobility models' performances in terms of mean square error. We applied 5 line simplification methods on the dataset to evaluate the impact of data reduction on the prediction performance. These methods are: nth point, Reumann-Witkam (R-W), Lang, Visualingam-Whyatt (V-W) and D-P [59]. As shown, D-P technique outperforms other data reduction methods in terms of lower mobility model error. Therefore, this technique can be deployed to eliminate the unnecessary data without any



(a) (b)

Figure 5.8 Mobility model learning performance. (a) GRU-based model and (b) BTPM (proposed).

disruptive impact on prediction performance. Also, in comparison with the other mobility models, BTPM obtains the lowest error of almost 0.054.

Figure 5.3 depicts the impact of different thresholds of data reduction algorithm on three parameters. Figure 5.3(a) shows how data reduction procedure works as we increase the threshold value. It is shown that data reduces noticeably with higher thresholds. In Figure 5.3(b), we can see the effect of different thresholds on execution time (i.e., data reduction and mobility learning steps). Logically, as we increase the threshold and obtain smaller dataset, the execution time reduces significantly as well. Lastly, Figure 5.3(c) demonstrates that how data reduction affects model performance (i.e., loss value of the learning algorithm). It is evident that reducing data for some thresholds does not result in model performance degradation. However, for higher thresholds (e.g., 1e-2, 1e-1), data reduction has disruptive effect on the performance. When we increase the threshold, the model is learning from the fewer data points and it is reasonable to have a slight decrease in the model performance. Furthermore, the impact of different RDP thresholds on execution time and accuracy performance is summarized in Table 3. It is evident that a higher value of threshold results in fewer data points and slightly higher loss error in some cases. The threshold value (i.e., epsilon) is set to 1e-4. It is the optimum value that gives us the appropriate amount of data points to form an accurate trajectory without redundancy.

Figure 5.6(a) indicates a sample of a plain trajectory of a user with all the data points and Figure 5.6(b) shows the same trajectory after choosing necessary points from the trajectory. As shown, the simplified version of the user movement trajectory is very similar to the original track and keeps only the effective samples. This huge data reduction leads to a noticeable simplification for the next steps in terms of complexity and run time.

Next, in order to reduce the noise or uncertainty in the data, the trajectory data is fed into a Kalman filter. Given the noisy measurements and initial assumptions, it takes the imperfect information and provides only the useful parts at the end. In Figure 5.7(a), we observe that the trajectory is quite noisy and spiky. However, it is much more cleaner and smother after passing it through Kalman filter which is shown in Figure 5.7(b).

Finally, before applying mobility learning predictor to the user sequence, the dataset is standardized. This is a common step for many learning predictors which set the mean value of the data to zero and the standard deviation to 1.

5.5.3 Trajectory prediction performance evaluation

In this section, the main purpose is to assess the impact of the proposed model on the prediction accuracy. First, the influence of different number of layers and neurons in the deep neural network is investigated for the model.

Table 6.2 summarizes the results of different configurations for different numbers of layers and neurons in a multi layer bidirectional GRU-based network. Basically, a lower loss value is obtained as the number of neurons and layers increases. However, if the number of layers and neurons is increased too much, it may lead to model overfitting. For instance, in the proposed approach, a network with 5 layers and 100 neurons in each layer results in a higher error. It implies that the model is unnecessarily complex and it can effectively train the data with a simpler network; therefore, there is no need to add unnecessary layers to the model. Overfitting happens in the case that the network is not able to generalize the pattern from the training input samples to test data and performs poorly with unseen data. Therefore, a neural network with 3 layers and 100 neurons is chosen for the model.

BTPM performance in terms of MSE value is shown in Figure 5.8. Figure 5.8(a) and Figure 5.8(b) respectively show mobility model performance for GRU-based model and BTPM. Comparison of the proposed method with GRU is provided since it has the best performance in comparison with other methods. The main purpose is to show the overall trend of the model in training and test phases. For GRU-based model, test error is high at first and gradually decreases from almost 0.5 to 0.2 in 60 epochs. However, for BTPM, the loss value for the test data starts from 0.028 and reduces

Table 5.4 Model performance (Loss value) of BTPM based on different number of layers and neurons.

	BTPM			
	$n = 10$	$n = 50$	$n = 100$	$n = 150$
$L = 3$	0.029	0.028	**0.025**	0.035
$L = 4$	0.056	0.081	0.093	0.082
$L = 5$	0.123	0.093	0.084	0.091

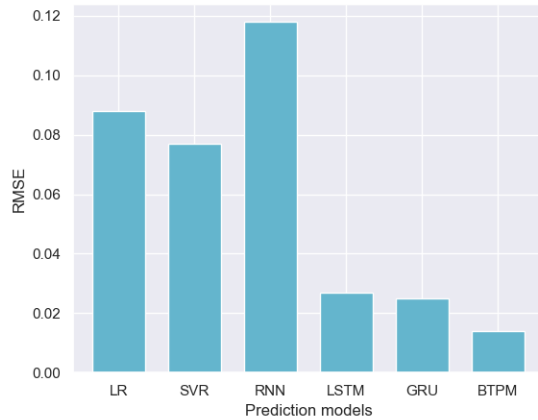


Figure 5.9 Mobility model prediction performances.

to 0.01 in 10 epochs which shows the excellent performance of the BTPM in terms of training fast and with high accuracy.

Figure 5.9 shows the result of prediction errors for all the techniques. As shown, there is a spike in error with RNN-based model due to the vanishing/exploding gradient issue of RNNs. The prediction error noticeably decreases for LSTM and GRU since their gated structure can learn user mobility behavior more effectively. We can observe that BTPM outperforms other approaches because of the fact that it extracts the correlations bidirectionally. Figure 5.10 shows the prediction errors for all the techniques in terms of MSE, MAE and RMSE applied on the three datasets. This comparison provides an intuitive idea of how the same mobility predictor can perform differently dealing with different user mobility data. For example, for dataset 3, BTPM slightly improves the prediction performance in comparison with LSTM/GRU-based models. However, for datasets 1 and 2, there is a noticeable improvement. As is shown, BTPM has the lowest error for all the three datasets compared to other methods. This can guarantee the effectiveness and generality of the proposed model.

Moreover, we have conducted simulations to evaluate the time complexity of the proposed mobility learning model with a large input (i.e., when we have a big dataset as user prior information). We considered user 153, from dataset 1, with almost 5 years movement trajectory from July 2007 to June 2012. Figure 5.11 presents processing times for all the methods. It is shown that GRU-based model is the most time-consuming method with 110,250 seconds. BTPM has the lowest time complexity with only 2,618 seconds. Using preprocessing techniques result in a huge reduction in execution time for the proposed method since a huge part of the unnecessary location points were removed before mobility learning. The proposed model takes advantage of numerous merits of bidirectional neural networks and addresses the inevitable problem of time complexity by applying appropriate preprocessing techniques.

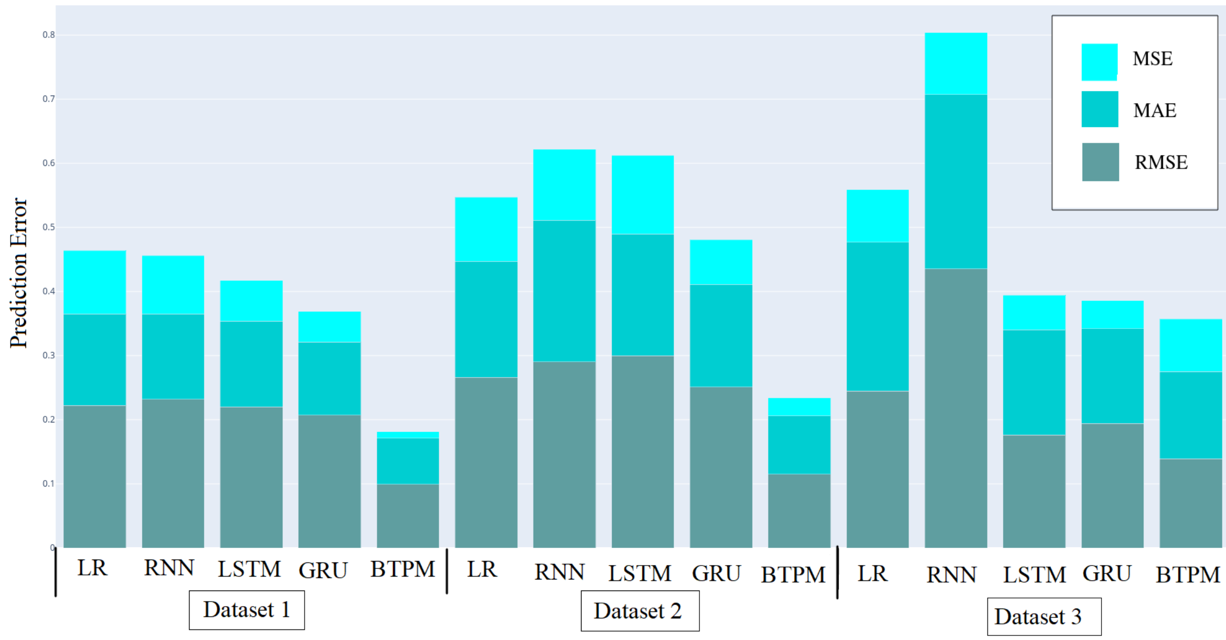


Figure 5.10 Mobility model performances using three different datasets.

5.5.4 Robustness of the proposed method

From another point of view, we conduct another experiment in order to evaluate the proposed model based on the robustness of the technique. We want to investigate the impact of two important factors on different mobility models:

1. *User mobility behavior*: Mobility behavior varies from one person to another and people have different degrees of predictability. Users that regularly travel hundreds of kilometres are most likely to have a low mobility predictability [6]. Therefore, a mobility predictor that can perform well for different users is a robust and reliable predictor.

2. *Length of available data for each user*: A determining factor in mobility learning is the amount of available data for each user. Undoubtedly, mobility learning accuracy for a user with longer available prior data is higher than having less available data for the same person.

These two factors are inevitable for mobile users in reality. We considered the combination of these factors to evaluate different mobility models in terms of being practical in more close to real-world scenarios. To this end, for the input data, we consider different users with different levels of predictability and also with different amounts of data available as movement history for each one them. In other words, we choose different users with their own different mobility behaviors to find out how well different techniques work dealing with different users.

Here, we selected three users from each of datasets. Table 5.5 presents the loss value of each method

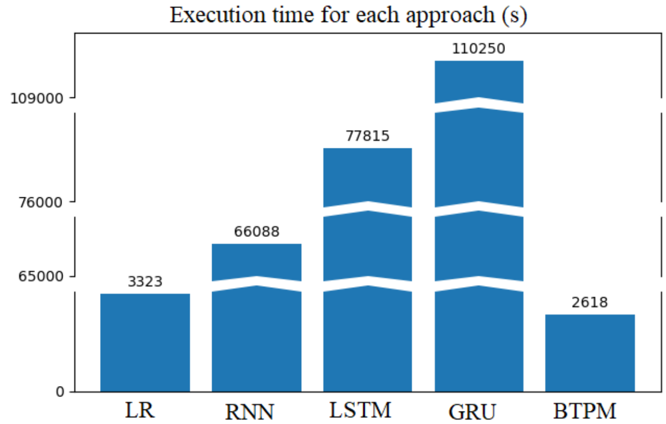


Figure 5.11 Execution time for each approach.

Table 5.5 Robustness of different approaches.

	Users	Duration	RNN	LSTM	GRU	BTPM (proposed)
Dataset 1	150	10 days	0.245	0.235	0.234	0.094
	153	3 months	0.118	0.027	0.025	0.014
	165	1 month	0.086	0.047	0.053	0.032
Dataset 2	1	hours	0.331	0.271	0.256	0.144
	2	hours	0.695	0.195	0.171	0.170
	3	hours	0.498	0.417	0.407	0.265
Dataset 3	t1	1 week	0.614	0.585	0.469	0.178
	t2	1 week	0.331	0.329	0.262	0.091
	t3	1 week	0.095	0.068	0.057	0.029
Range			0.609	0.558	0.444	**0.251**

for each user. In this table, the term "range" refers to the subtraction of the highest value and the lowest value of error for each method. When a method has a high value of range, it means that there is a noticeable difference between the highest and the lowest prediction errors. In other words, this mobility model can work well for a specific user with a specific mobility behavior (i.e., can learn the mobility pattern of the user) and can have a poor performance for another user; therefore, it is not a robust mobility model. The lowest range belongs to BTPM with 0.251 which is smaller than other methods and shows that it can perform very well for different mobility behaviors and it is not dependent on a specific mobility behavior. Table 5.6 represents a qualitative comparison between different approaches for user trajectory prediction. BTPM extracts mobility patterns by deeply

Table 5.6 Comparison between different approaches (more stars show a relatively better performance).

	RNN based model	LSTM based model	GRU based model	BTPM
Consider prior data correlation	Yes	Yes	Yes	Yes
Consider posterior data correlation	No	No	No	Yes
Fix the exploding/vanishing problem	No	Yes	Yes	Yes
Prediction accuracy	*	**	***	****
Time complexity	***	**	*	****
Example in the literature	[29]	[16]	[15]	Proposed

considering both the prior and posterior correlations in the user mobility trajectory. Consequently, it achieves a relatively better accuracy performance compared to the other methods. Moreover, the key feature of the two-phase user sequence preparation is huge time complexity reduction while improving the prediction accuracy.

Table 5.7 Cell tower ID and location area code (LAC) for the predicted trajectory.

Order	Cell ID	LAC	Order	Cell ID	LAC	Order	Cell ID	LAC	Order	Cell ID	LAC
1	2896615	41192	6	2907505	41192	11	58594	4526	16	58592	4526
2	2896616	41192	7	31137	4526	12	36422	4523	17	4331139	4147
3	19995649	4523	8	2918015	41192	13	2895107	41192	18	23535874	4523
4	59618	4309	9	2895117	41192	14	127883653	4523	19	20162561	4523
5	31085	4523	10	18181132	4423	15	2898306	41192	20	4926831	4147

5.5.5 Extension to cellular networks

Cellular networks mobility datasets (i.e.,call detail records (CDR) datasets) are mostly used to predict the next cell of the user in mobile networks. These methods to some extent suffer from a low location accuracy depending on the cell range (varying from meters to kilometers). On the other hand, GPS data have the best location accuracy performance and mostly are used for the next location prediction based on the geographical coordinates of the user. An ideal situation is when we can potentially benefit from both methods (i.e., to predict the next cell of the user with the exact geographical coordinates in the cell). In this work, we tried to fulfill this objective. A GPS dataset was deployed for the proposed approach to take advantage of having a precise and accurate record of user movement history which will result in a precise prediction as well. Then we can convert it to cellular networks information.

In this section, we want to take the predicted information of user's future trajectory in geographical coordinates and transfer it to the cellular networks information. Figure 5.12 shows a sample of

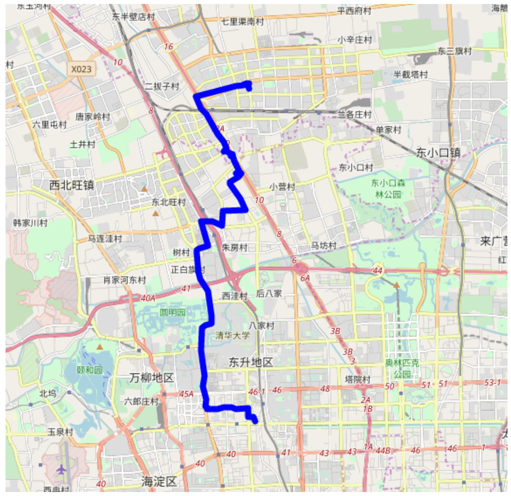


Figure 5.12 A sample of predicted trajectory for the user.

predicted future trajectory of a user. Next, using a real-world database of cellular towers with GPS positions (OpenCelliD [82]), we transferred the predicted information based on geographical coordinates (i.e., longitude and latitude) to the cellular network information (i.e., Cell ID and corresponding location area code). We map the predicted trajectory to the corresponding cell towers on that coordinates (based on closest distance). This database contains the information of cell towers, including their locations based on latitude and longitude, cell range, mobile country codes (MCC), mobile network code (MNC), location area code (LAC), cell tower ID (Cell ID) and radio type (i.e., LTE ,GSM). Table 5.7 summarizes an example of the corresponding cell towers information (Cell tower ID and location area code) for the predicted trajectory sample for the 20 time steps ahead.

5.6 Conclusion

In this paper, we highlighted the importance of mobility prediction as a promising key enabler for intelligent mobility management in self-organizing cellular networks. The proposed method mainly focuses on the prediction accuracy performance improvement to meet the demands of future mobile networks. A novel bidirectional trajectory prediction model is proposed that is mainly based on the bidirectional recurrent neural networks. Moreover, a preprocessing phase is introduced to prepare the mobility dataset specifically suitable for the framework. Line simplification techniques and kalman filter were deployed to respectively reduce the unnecessary data and remove the noise from the data before training the model. The proposed approach (BTPM) has two main distinguishing features in comparison with other methods: (i) having a two-phase user mobility sequence preparation specifically introduced for mobility prediction purpose which results in decreasing the noise level and significant time complexity reduction, (ii) having a bidirectional approach to extract mobility patterns by deeply considering the correlation in the user mobility trajectory which results in significant prediction accuracy improvement, obtaining a robust performance dealing with different users with different degrees of predictability and being practical in more close to real-world scenarios. This can have a profound impact on our analysis and gives us a deeper understanding of user mobility behavior with a significantly lower execution time. Simulation results show that BTPM has a high accuracy performance and outperforms other alternative approaches. It obtains a model error of 0.014 and effectively decreases the execution time up to 97%. The bidirectional network takes full advantage of data analysis in both directions (backward and forward) in order to provide a long-term prediction and to model user's mobility even with a complex pattern.

For future work, the proposed bidirectional mobility model can be applied to the other types of user mobility data (e.g., CDR datasets, check-in datasets). Moreover, it can be used as a potential tool for providing mobility-aware services in cellular networks such as mobility-aware call admission control techniques and resource allocation.

CHAPTER 6 ARTICLE 3: A HYBRID USER MOBILITY PREDICTION APPROACH FOR HANDOVER MANAGEMENT IN MOBILE NETWORKS

Authors: Nasrin Bahra and Samuel Pierre
Status: Accepted for publication in Telecom journal of communication and networks.

Abstract

Mobile networks are expected to face major problems such as low network capacity, high latency, and limited resources but are expected to provide seamless connectivity in the foreseeable future. It is crucial to deliver an adequate level of performance for network services and to ensure an acceptable quality of services for mobile users. Intelligent mobility management is a promising solution to deal with the aforementioned issues. In this context, modeling user mobility behaviour is of great importance in order to extract valuable information about user behaviours and to meet their demands. In this paper, we propose a hybrid user mobility prediction approach for handover management in mobile networks. First, we extract user mobility patterns using a mobility model based on statistical models and deep learning algorithms. We deploy a vector autoregression (VAR) model and a gated recurrent unit (GRU) to predict the future trajectory of a user. We then reduce the number of unnecessary handover signaling messages and optimize the handover procedure using the obtained prediction results. We deploy mobility data generated from real users to conduct our experiments. The simulation results show that the proposed VAR-GRU mobility model has the lowest prediction error in comparison with existing methods. Moreover, we investigate the handover processing and transmission costs for predictive and non-predictive scenarios. It is shown that the handover-related costs effectively decrease when we obtain a prediction in the network. For vertical handover, processing cost and transmission cost improve, respectively, by 57.14% and 28.01%.

IEEEkeywords

mobility prediction; machine learning; mobile networks

6.1 Introduction

Nowadays, there is a significant increase in the number of mobile users around the world. It is estimated that this number will dramatically increase up to 5.7 billions by 2023 based on the Cisco annual record [57]. Consequently, a considerable amount of mobility data is generated by these users while moving everyday. We can fully exploit the potential of these produced data in many applications in order to extract frequent patterns and to derive invaluable insights. This fact opens up plenty of golden opportunities in many prospective areas such as mobility-aware services. User mobility prediction techniques can be applied to these available data to obtain a deeper understanding of users' mobility behaviours and their demands by using data mining and learning algorithms.

Foreseeable future mobile communication networks are rightly expected to support high data rates and seamless connectivity for a vast number of devices. They are also expected to avoid lengthy delays when providing services to mobile users in the network. However, there are some inherent technical challenges in mobile networks that require special attention when providing a high quality of services (QoSs) for users [12]. User mobility can pose tremendous challenges and needs to be managed appropriately. Some of these problems are heterogeneity, more frequent handovers, increase in handover-related costs, and increase in call dropping probability [40]. Therefore, it is crucial to have proper mobility management in the network.

A self-organizing network (SON) is a potential long-term solution to cope with some of the mentioned underlying issues [4]. These networks are growing greatly in popularity due to the fact that they are able to dynamically adapt to changes and to learn from past experiences. A SON is composed of three main functions: self-configuration, self-optimization, and self-healing. In the context of intelligent mobility management, user mobility prediction is a subsection of self-optimization functions with the main objective of predicting the future location of the user using learning techniques. Mobility awareness can make potentially outstanding contributions to many research areas related to mobile network management including resource allocation, handover management, and location-based services and, in general, can increase the quality of services.

When a user moves between different cells, the procedure of reassigning user resources to a new base station is called handover (HO). Handover is triggered based on the received signal condition or network load balancing. In dense 5G cellular networks, user mobility results in more repeated handovers and service interruptions. This can cause high signaling overhead, especially for mobile users at high velocities. Hence, handover management has a leading role in cellular network performance in order to support seamless connectivity for users. Therefore, improving HO procedure performance can be highly beneficial in reducing link failures, latency, and costs in a cellular networks. Modeling user mobility can significantly help the network obtain prior knowledge about a user's future trajectory and handovers. Therefore, deploying predictive handover management, the network can prepare the required services for users in advance to reduce latency and costs [5]. Having predicted the future HO of the user, handover preparation steps can be performed beforehand, and when HO is needed, the process can start from the execution phase. This can reduce the number of signaling messages needed to be exchanged and consequently results in lower HO latency.

There are a large number of works related to user mobility prediction and mobility-aware services for cellular networks in the literature. Despite valuable efforts, a generic reliable mobility predictor with a low prediction error for short-term and long-term predictions with low complexity is still missing. We believe that the key factor for a successful mobility model is to properly discover the existing hidden correlations in a user's movement history. When the model captures the dependencies in the prior user movement trajectories, it can provide a better understanding of user mobility behaviour and can make a long-term prediction. In this paper, our main objective is to deploy user mobility prediction as an invaluable tool to optimize the handover signaling procedure. First, we propose a hybrid trajectory prediction method based on statistical and learning models. The proposed mobility model exploits the vector autoregression (VAR) model [83] and gated recurrent unit (GRU) [71] to further capture correlations in the user's past movement history as the input samples. Then, having predicted the future user trajectory, we optimize the handover signaling procedure in order to reduce the handover signaling costs. We observe that the predictor accuracy is of great important in obtaining an accurate and effective impact on handover signaling optimization. The major contributions of our work is as follows:

- Proposing a mobility prediction technique to model user mobility behaviour. The proposed VAR-GRU model predicts the future trajectory (i.e., path) of a user. The core concept is to fully analyze the existing dependencies in a user's past trajectories and to extract general patterns in the data.
- Investigating the impact of user mobility prediction on the conventional handover signaling procedure. Handover processing and transmission costs are evaluated to compare the predictive and non-predictive scenarios.
- Conducting experiments on the user mobility data generated from real users to provide an in-depth analysis of the effectiveness of the proposed approach.

The rest of this paper is organized as follows. Section 6.2 provides a short review of some of the recent related works in this field. Section 6.3 presents the proposed approach in two subsections: the proposed user mobility prediction technique and handover procedure analysis. Section 6.4 provides an analysis regarding HO signaling cost. In Section 6.5, we summarize the results of our experiments. Finally, the conclusion of our work is provided in Section 6.6.

6.2 Related Works

There are a vast number of works in the field of user mobility prediction and mobility-aware services in cellular networks. Generally, user mobility prediction can be categorized in two main parts: (1) mobility models that predict user future path or trajectory (i.e., regression task) and (2) mobility models that predict user next location (i.e., classification task). Our work belongs to the first category and we essentially predict the future trajectory of the user. In the following, we provide a short review of recent works from both categories.

A huge part of the related works are based on the concept of machine learning algorithms. In [41], the authors introduced a hybrid mobility prediction method using principal component analysis (PCA) and gated recurrent unit. This approach was developed to predict Internet of Things (IoT) mobile users. A destination prediction approach was introduced in [67] using a long short-term memory (LSTM) network. This method investigates user prior knowledge in a bidirectional structure. In [1], a method based on deep learning techniques was adopted for multi-user trajectory prediction. This approach uses LSTM cells to learn a user's pattern and then extends it to the general case. In [84], an LSTM-based model was proposed for destination prediction. This model copes well with data sparsity as well. An RNN-based method was proposed in [35] for trajectory representation learning with the focus of considering both spatial and temporal features in trajectory learning. In [85], the authors proposed a mobility model based on the recurrent neural network variations. They proposed to deploy line simplification techniques to simplify the user trajectory. The core idea is to eliminate irrelevant data to reduce the execution time while improving the prediction accuracy. Then, the preprocessed dataset is used to learn the user mobility behaviour. In [66], the authors proposed a trajectory prediction method. Their technique explores existing relations between a user's morning and afternoon trajectories using a similarity metric. In [86], an analytical model was presented to calculate the handover-related costs including HO latency, signaling overhead, and call dropping. A part of the existing works is based on the popular concept of location-based social networks (LSBNs) that mainly investigates check-in datasets to predict the next point-of-interest (POI). In [44], the authors proposed a context-aware scheme to discover regular patterns in the user's movement history based on RNNs. To solve the data sparsity problem, they analyzed the social relationships as well. In [45], POI estimation was provided by applying gated recurrent units on a check-in dataset.

Moreover, there are many mobility models based on Markov models and their variations. In [87], the authors proposed a mobility prediction method based on Markov chains, with the main objective of reducing handover-related costs. In [27], the authors proposed a Markov-based model that predicts both a user's path and destination. In [22], a mobility model was introduced to predict the next base station in an LTE network using a Markov model. In [18], a mobility-aware proactive multicast technique was presented. This approach deployed a Markov model to estimate user's next cells and staying durations. However, it has been proven that Markov models fail to deal with radical changes in user mobility behaviour and to make a long-term prediction [17]. They cannot predict when hidden states increase. They performed more accurately when there were a limited number of observations.

We believe that a mobility model should be able to fully analyze the past trajectory of the user to properly predict the future path of that user with the smallest possible error. A model is needed that can perform effectively for both short- and long-term predictions with an acceptable error.

6.3 The Proposed Predictive Handover Management Approach

In this section, we present information regarding the proposed predictive handover management approach. Figure 6.1 shows the overall procedure of the handover management approach. This approach is composed of two main parts: (1) user mobility prediction and (2) predictive handover procedure analysis. For the first step, the main idea is to predict the future trajectory of the user using a hybrid model based on the VAR and GRU models. For the second step, we exploited the obtained prediction information from the last step to optimize the handover signaling procedure for the both horizontal and vertical handovers. In the following, we provide the details regarding each step.

6.3.1 User Mobility Prediction

Objective: Given the movement history of a user for the previous n time steps (i.e., user past trajectory), we want to predict the user's future trajectory for n time steps ahead.

We define the trajectory for user i as a sequence of time-stamped points, $L_{useri} = (l_{t-n}, ..., l_{t-2}, l_{t-1})$, where $l_t = (x_t, y_t, z_t)$ contains information on the geographical location of the user (i.e., longitude, latitude, and elevation).

First, we feed the user past trajectories into a vector autoregression model to obtain useful insights on the user mobility behaviour. Given $L_t = (l_{t-n}, ..., l_{t-2}, l_{t-1})$ as a user trajectory, in which L_t is a $(v \times 1)$ vector with v variables, the VAR(n) model can be expressed as

$$l_t = a + C^1 l_{t-1} + C^2 l_{t-2} + ... + C^n l_{t-n} + \epsilon_t \tag{6.1}$$

where $C^1, C^2, ..., C^n$ denote coefficient matrices that are square matrices of order v and a is an intercept vector. ϵ_t is a white noise vector with zero mean, and it is assumed that there is no correlation between noises in the system across time. Each variable is obtained by the lagged values of all of the variables. In another words, to predict the value of a variable at the current time step, we consider the impact of all of the other variables from the past time steps. In our model, we considered three variables (i.e., $v = 3$) as defined earlier for the user trajectory $l_t = (x_t, y_t, z_t)$, and the model can be expressed as

$$\begin{pmatrix} x_t \\ y_t \\ z_t \end{pmatrix} = \begin{pmatrix} a_1 \\ a_2 \\ a_3 \end{pmatrix} + \begin{pmatrix} c^1_{11} & c^1_{12} & c^1_{13} \\ c^1_{21} & c^1_{22} & c^1_{23} \\ c^1_{31} & c^1_{32} & c^1_{33} \end{pmatrix} \begin{pmatrix} x_{t-1} \\ y_{t-1} \\ z_{t-1} \end{pmatrix} + ... + \begin{pmatrix} c^n_{11} & c^n_{12} & c^n_{13} \\ c^n_{21} & c^n_{22} & c^n_{23} \\ c^n_{31} & c^n_{32} & c^n_{33} \end{pmatrix} \begin{pmatrix} x_{t-n} \\ y_{t-n} \\ z_{t-n} \end{pmatrix} + \begin{pmatrix} \epsilon_{1t} \\ \epsilon_{2t} \\ \epsilon_{3t} \end{pmatrix} \tag{6.2}$$

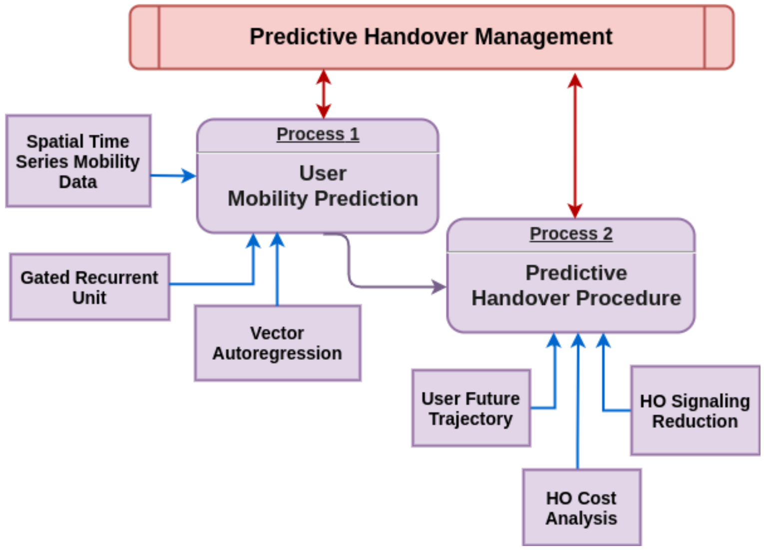


Figure 6.1 The proposed handover management approach.

Then, we have

$$
\begin{aligned}
x_t &= a_1 + c^1_{11}x_{t-1} + c^1_{12}y_{t-1} + c^1_{13}z_{t-1} + ... + c^n_{11}x_{t-n} + c^n_{12}y_{t-n} + c^n_{13}z_{t-n} + \epsilon_{1t} \\
y_t &= a_2 + c^1_{21}x_{t-1} + c^1_{22}y_{t-1} + c^1_{23}z_{t-1} + ... + c^n_{21}x_{t-n} + c^n_{22}y_{t-n} + c^n_{23}z_{t-n} + \epsilon_{1t} \\
z_t &= a_2 + c^1_{31}x_{t-1} + c^1_{32}y_{t-1} + c^1_{33}z_{t-1} + ... + c^n_{31}x_{t-n} + c^n_{32}y_{t-n} + c^n_{33}z_{t-n} + \epsilon_{1t}
\end{aligned} \tag{6.3}
$$

Since the VAR model is of order n, each variable is calculated by the n lagged values of all three variables in the model from the previous time steps. For example, c^1_{11} and c^1_{12} determine the impacts of x_{t-1} and y_{t-1}, respectively, when calculating x_t. This gives us a better understanding of input sample correlations.

For the next step, we feed the obtained results into a deep GRU-based neural network. For each time step, we consider the current location and the previous state to obtain the next state as the output. The GRU model has two main parts that deal with keeping or deleting user data from the past, namely the update gate and the reset gate. Using these two gates, the GRU model can solve the vanishing/exploding issue of RNNs and can make a long-term prediction [71]. Table 6.1 presents the model parameters.

The update gate $u(t)$ can be obtained by

$$u(t) = \sigma(b_u + U_u L(t) + W_u h(t-1)) \tag{6.4}$$

In (6.4), user trajectory $(L(t))$ and the information from the previous step $(h(t-1))$ are multiplied by their weights (the weights control the impact of each value on the final decision). Then, the sum of this multiplication with the update bias value goes through a sigmoid function (σ). The output is a value between 0 and 1. The update gate decides which part of the user's past movement history should be kept for analysis.

Next, the reset gate $r(t)$ can be given by

$$r(t) = \sigma(b_r + U_r L(t) + W_r h(t)) \tag{6.5}$$

It has the same procedure as that in Equation (6.4). The only difference is that they are multiplied by their own weights. The reset gate decides which part of the past information should be eliminated for future decisions. Lastly, we can compute the effect of these gates in the output as follows:

$$h(t) = u(t-1)h(t-1) + (1 - u(t-1))\sigma(b + UL(t-1) + Wr(t-1)h(t-1)) \tag{6.6}$$

where b, b_u, and b_r are bias vectors; U, U_u, and U_r are input weight matrices; and W, W_u, and W_r are recurrent weight matrices. Additionally, σ denotes a sigmoid function that has a value between 0 and 1 in the output.

Figure 6.2 depicts the VAR-GRU trajectory prediction model. As shown, we took a sequence of the user past trajectory as the input. First, the VAR unit models the dynamic behaviour of the variables based on the raw data. Then, the obtained results are fed into the GRU network for the training and inference steps and produce the future trajectory of the user in the output.

6.3.2 Predictive Handover Procedure

Objective: Given the user future trajectory, we want to reduce the number HO signaling messages needed when handover is triggered.

In this section, we investigate the conventional (i.e., non-predictive) and predictive handover signaling procedures. More specifically, we analyze the impact of trajectory prediction on handover signaling flow and compare it with the non-predictive handover scenario. A conventional handover procedure has three main steps including handover preparation, execution, and completion [88]. In the preparation step, target gNB is selected by the source gNB. The target gNB completes admission control. In the execution step, the user device attaches to the target gNB. In the last step, HO completion, the data path is switched to the new gNB and the handover steps are completed.

However, if the network has reliable information on the user's future locations (i.e., path), it can

Table 6.1 GRU model notations.

b, b_u, b_r	Bias vectors	U, U_u, U_r	Input weight matrices
W, W_u, W_r	Recurrent weight matrices	$h(t)$	Hidden states
$u(t)$	Update gate	$r(t)$	Reset gate

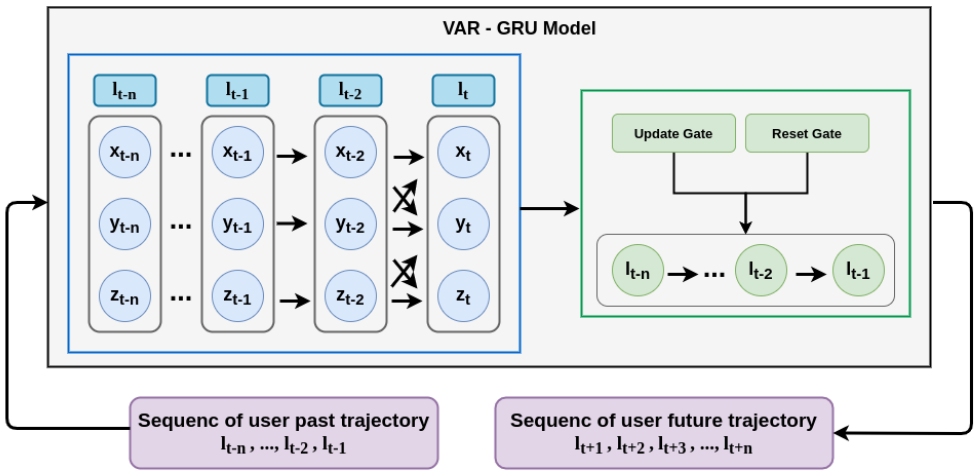


Figure 6.2 The proposed VAR-GRU mobility model.

effectively perform some of the HO procedure steps before handover triggers. With correct prediction of the user future trajectory, we can complete the HO preparation phase in advance and when user requires a handover, it will start the execution phase. However, with a wrong prediction, even more signaling messages are required compared to the conventional HO. In the following, we elaborate on the impact of prediction on HO signaling in both horizontal and vertical handovers in a 5G architecture.

Figure 6.3 shows a 5G intra-AMF-UPF handover signaling process where we predicted the target base station. AMF and UPF in 5G architecture, respectively, refer to access and mobility management function (AMF) and user plane function (UPF). We considered both the cases of right and wrong predictions regarding a target gNB. As we can see in the figure, when the target gNB is correctly predicted, it is equal to the actual next base station (highlighted in green). However, with a wrong prediction, the predicted target base station is not the same as the actual next base station (highlighted in red). In this case, the source base station should send a message to the wrongly predicted base station and cancel the HO request. Therefore, there are three cases including (1) non-predictive HO (i.e., conventional HO), (2) predictive HO with right prediction, and (3) predictive HO with wrong prediction. For the first case, we need to perform all of the HO preparation, execution, and completion steps. According to Figure 6.3, signaling messages from numbers 3 to 14 need to be exchanged (i.e., parts 2 and 3). For the second case, when we predict the next base station correctly, the preparation steps can be performed in advance. When HO is triggered, the procedure starts with the execution phase. Thus, we need signaling messages from numbers 8 to 14 (i.e., part 3). Lastly, for the third case with a wrong prediction, signaling messages from numbers 1 to 14 are needed to complete the HO. First, resources at the wrongly predicted gNB should be released (part 1). Next, the non-predictive HO is applied to communicate with the actual next base station (parts 2 and 3).

Similarly, Figure 6.4 shows an inter-radio access (RAT) handover signaling diagram from 5G next generation core (5G NGC) to evolved packet system (EPS). Similar to the procedure in the Figure 6.3, we investigated the HO signaling procedure for right and wrong predictions regarding the next target gNB. There are three cases here as well. We note that the prediction accuracy has a vital role in the success of reducing HO signaling costs. An accurate prediction can reduce the signaling noticeably. On the other hand, as is evident, wrong predictions lead to even more signaling messages. Therefore, it is important to give undivided attention to the mobility predictor accuracy performance.

6.4 Handover Signaling Cost Analysis

To investigate the impact of the proposed predictive handover procedure, we studied handover cost signaling for both the predictive and non-predictive (i.e., conventional handover procedure) cases. We assessed the handover signaling cost based on processing and transmission costs similar to that in [2, 87, 89].

6.4.1 Processing Cost

We define the handover processing cost C_P based on the required number of signaling messages during a handover. Therefore, handover processing cost reduction can be given by

$$C_{P-Reduction} = [\frac{1}{NM_{np}}(NM_{np} - NM_p)] \times 100\% \tag{6.7}$$

where NM_{np} and NM_p denote the number of messages needed for non-predictive and predictive scenarios, respectively.

6.4.2 Transmission Cost

Transmitting handover messages causes an inevitable delay in establishing a link. The total HO messages M needed to perform a handover is as follows:

$$M = m_{prep} + m_{exe} + m_{comp} \tag{6.8}$$

where m_{prep}, m_{exe}, and m_{comp} denote the messages that are needed to be exchanged during the HO preparation, execution, and completion steps, respectively. HO transmission cost C_T can be calculated based on the required link delay for handover signaling message transmission. This cost can be obtained by

$$C_T = \frac{1}{1ms}[D_M] \tag{6.9}$$

where D_M is the total required link delay for HO steps and can be defined as

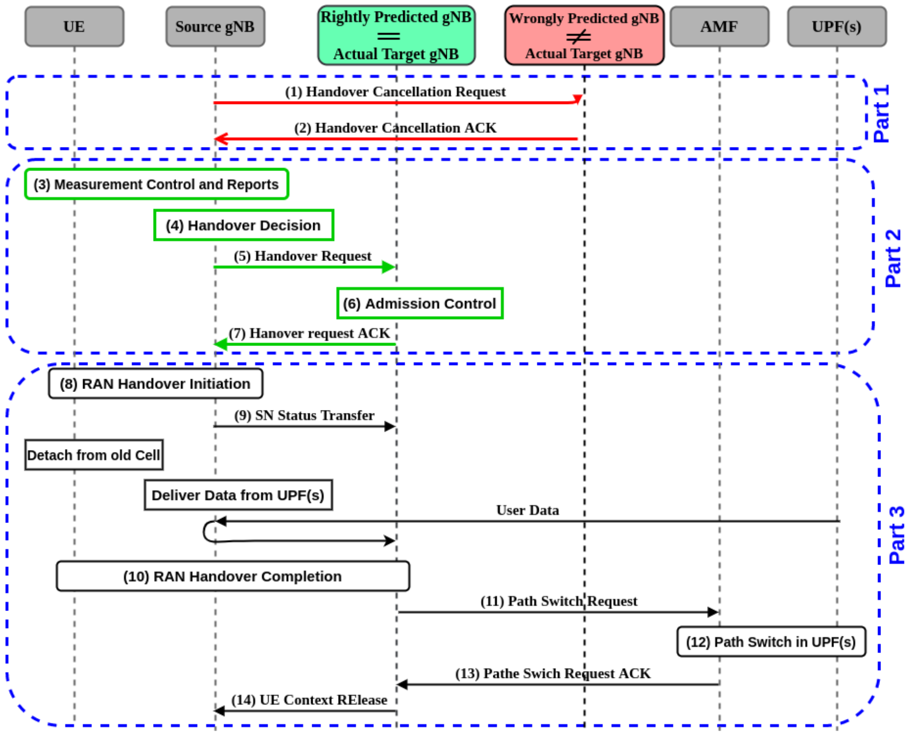


Figure 6.3 Intra-AMF-UPF handover signaling diagram. AMF and UPF in the 5G architecture refer to access and mobility management function (AMF) and user plane function (UPF).

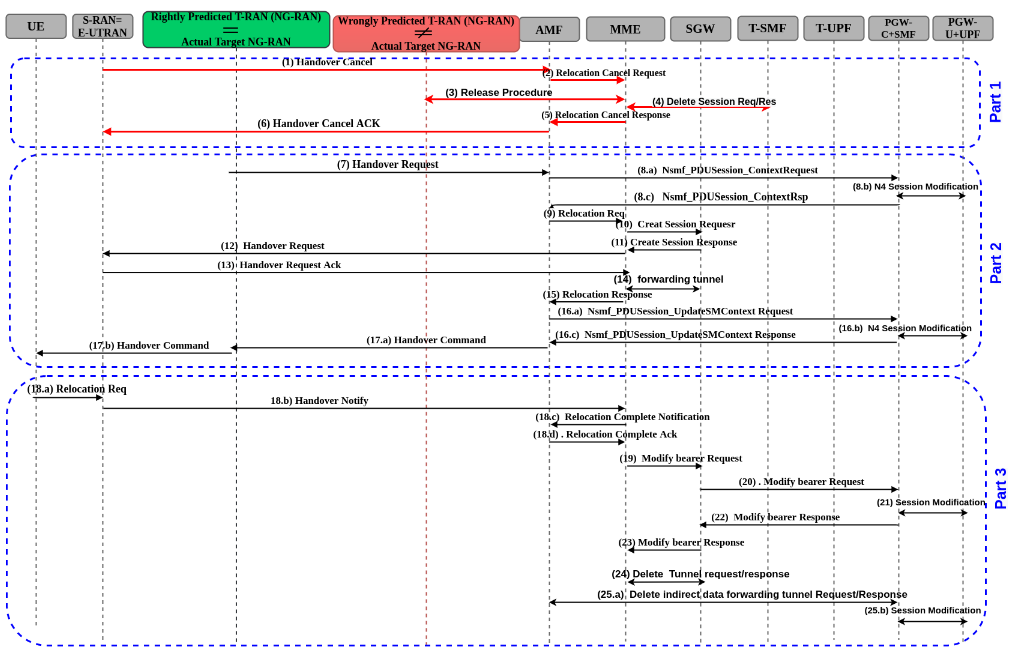


Figure 6.4 Inter-RAT handover signaling diagram (from 5G NGC to EPS).

$$D_M = \sum d_s, \quad s \in \mathbb{N} \tag{6.10}$$

where s refers to the number of signal (see Figures 6.3 and 6.4) and d_s denotes the link delay between two nodes regarding signal message s. In other words, HO transmission cost is calculated based on the total delay needed to establish links between nodes and ro complete the handover process.

Thus, for an intra AMF-UPF handover, the transmission cost for non-predictive or conventional HO can be expressed as

$$C_{T-np} = \frac{1}{1ms}[D_{M-np}], \quad D_{M-np} = \sum_{S=3}^{14} d_s \tag{6.11}$$

where we calculate the delay for sending signal message 3 to 14 (according to Figure 6.3).

For the predictive HO, it is crucial to consider the impact of not only the right prediction but also the wrong prediction. Thus, we have the following:

$$C_{T-p} = \frac{1}{1ms}[D_{M-p}], \quad D_{M-p} = P_{acc}(D_{M-right}) + (1 - P_{acc})(D_{M-wrong}) \tag{6.12}$$

where P_{acc} is the accuracy of the prediction and $D_{M-right}$ and $D_{M-wrong}$ are, respectively, the link delay regarding signaling messages when we predicted the target base station correctly or incorrectly. They can be calculated as

$$D_{M-right} = \sum_{S=8}^{14} d_s, \quad D_{M-wrong} = \sum_{S=1}^{14} d_s \tag{6.13}$$

where we calculated the delay for sending signal message 8 to 14 for right predictions and 1 to 14 for wrong predictions (according to Figure 6.3). Here, we considered a horizontal handover (Figure 6.3) for the range of s in Equations (6.11) and (6.13). For vertical HO, we used the same procedure for cost calculation, and the range of s is defined according to Figure 6.4.

6.5 Experimental Results

In this section, we carried out a series of simulations to investigate the impact of the proposed approach. Our main objective was to evaluate the performance of the proposed mobility model and to then analyze the impact of the prediction on HO-related costs. In the following, first, we describe the mobility data that we used for our experiments and the approaches with which we compared our method. Next, we provide the results regarding the mobility model performance and its effect on HO procedure optimization.

6.5.1 Mobility Data Description

There are several human mobility data types including mobility data generated in cellular networks, WiFi networks, social networks, and global positioning system (GPS). We deployed GPS data since it has the highest localization accuracy. To conduct our experiments and to analyze the impact of the proposed method on different users, we chose mobility data from open street map (OSM). We deployed GPS traces that were uploaded by different users and are publicly available (`https://www.openstreetmap.org/traces`, accessed on 12-10-2020). For each user i, a sequence of past GPS trajectories $L_{useri} = (l_{t-n}, ..., l_{t-2}, l_{t-1})$ was provided as the input of the mobility model. A mobility predictor uses this information as prior knowledge of user mobility behaviour and tries to learn the repetitive patterns within. These sequential data are time-stamped data that contain information regarding time, longitude, latitude, and elevation of each location point in the user trajectory. Hence, l_t in a user input sequence that contains information on longitude (x_t), latitude (y_t), and elevation (z_t) at time step t (i.e., $L_{useri} = ((x_{t_1}, y_{t_1}, z_{t_1}), (x_{t_2}, y_{t_2}, z_{t_2}), ...)$).

6.5.2 Comparison

In order to provide a better understanding of how well our method works, we provide a fair comparison between some related techniques:

- Recurrent neural network [77]: a RNN-based mobility model analyzes the user's past locations sequentially.
- Long short-term memory [72]: LSTM-based approaches deploy state units and a forget gate to learn the mobility pattern.
- Gated recurrent unit [61]: GRU-based techniques control the impact of the latest observations using update and reset gates.

6.5.3 Experimental Settings

We considered the same hardware and software configurations for our method and all of the other methods. All of the simulations were performed using an Intel core i7-6700k CPU with 4.00 GHz and 32 GB RAM. We used keras for the simulations, which is a popular Python library. In the neural network, we deployed four layers with 100 neurons in each layer. To avoid overfitting, we used two drop out layers with values of 0.2 that randomly dropped 20 percent of the connections in each layer. Stochastic gradient decent (SGD) was used as the model optimizer with a learning rate of 0.001. Table 6.2 shows the simulation parameters that we considered for our experiments.

Table 6.2 Simulation parameters for training the proposed mobility model.

Number of layers	4	Model optimizer	SGD
Number of neurons	100	Learning rate	0.001
Weight initializer	Glorot uniform	Loss function	RMSE, MAE
Training data percentage	75%	Sequence length	5–2000
Batch size	10	Dropout rate	0.2

6.5.4 Mobility Model Performance

To evaluate the error of the mobility model, we considered two metrics: root mean square error (RMSE) and mean absolute error (MAE). Given b input samples from n total number of samples, RMSE can be given by

$$RMSE_b(\bar{y}) = \sqrt{1/n \sum_{i=1}^{b} (y(x) - \bar{y}(x))^2} \tag{6.14}$$

where x, $y(x)$, and $\bar{y}(x)$ are the input sample, actual result, and predicted result. The ideal value for RMSE is zero to show that the predicted and actual values are the same. Additionally, the MAE value can be calculated as

$$MAE_b(\bar{y}) = 1/n \sum_{i=1}^{b} |y(x) - \bar{y}(x)| \tag{6.15}$$

with lower values of MAE and RMSE being better for the predictor.

Length of the user sequence plays a vital role in evaluating a mobility model performance. Generally, when the length of the sequence is small, it is easier for the model to investigate the correlations. There is a tradeoff between sequence length and mobility behaviour learning. It is hard for the model to extract the user's mobility pattern when it deals with either a too short trajectory or a too long trajectory. Figure 6.5 shows the impact of different sequence lengths (n) on the mobility model prediction error. Our model has a same-length input and same-length output structure. As shown, the proposed VAR-GRU mobility model has the lowest prediction error for all values of n.

One important point for robust and reliable mobility models is the fact that users have different degrees of predictability based on their movement regularities [6]. Some users have more repetitive behaviour and thus are more predictable. Some users tend to have constantly changing mobility behaviour, and therefore, they are more unpredictable. A reliable mobility predictor should be able to model different users' mobility with low error. Figure 6.6 shows how the proposed mobility model works when dealing with four different users. As shown, the VAR-GRU model outperforms other methods for the four different users.

Next, we make a comparison between the mobility models based on the MAE value. Table 6.3 summarizes the results regarding each method. We observe that the proposed VAR-GRU trajectory

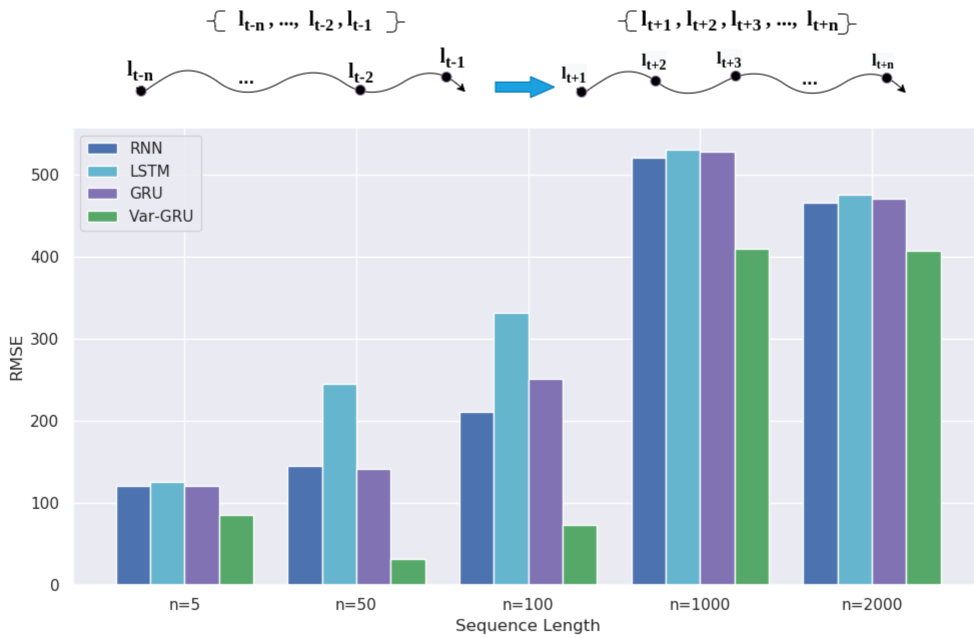


Figure 6.5 Impact of the sequence length on the prediction error.

predictor model outperforms the other approaches in terms of MAE value.

Next, having predicted the future trajectory of the user, we investigated the impact of this prediction on handover costs in terms of number of messages required and the transmission costs. After obtaining the predicted user future trajectory, we evaluated the accuracy of our predictions based on the actual future trajectory. We evenly distributed the base stations (BS) in the area of the user trajectory within the cell range of 1000 m. Therefore, each location point was within a cell area and we assigned it to the base station of that cell. We performed these steps for both the predicted and the actual user path. Then, we measured the accuracy of our predictions as a classification task and to observe whether the BS associated with the predicted trajectory points matched those of the actual trajectory points.

6.5.5 Impact of Prediction on HO Costs

Figure 6.7 represents the results of the vertical and horizontal costs regarding the non-predictive case and the predictive case using different mobility models. We can observe the predictor accuracy and the corresponding HO costs for each model. Table 6.4 summarizes the delays regarding each type of link between two nodes in the process of HO completion. As shown, we obtained the highest transmission cost at 138.5 for the vertical handover when there is no prediction, and it gradually decreased as the prediction accuracy increased. It has the lowest value for the VAR-GRU at 99.7 (i.e., 28.01% reduction). For horizontal HO, the transmission cost for the non-predictive case is 23. This value slightly increased when we used the RNN and LSTM models as the mobility model at,

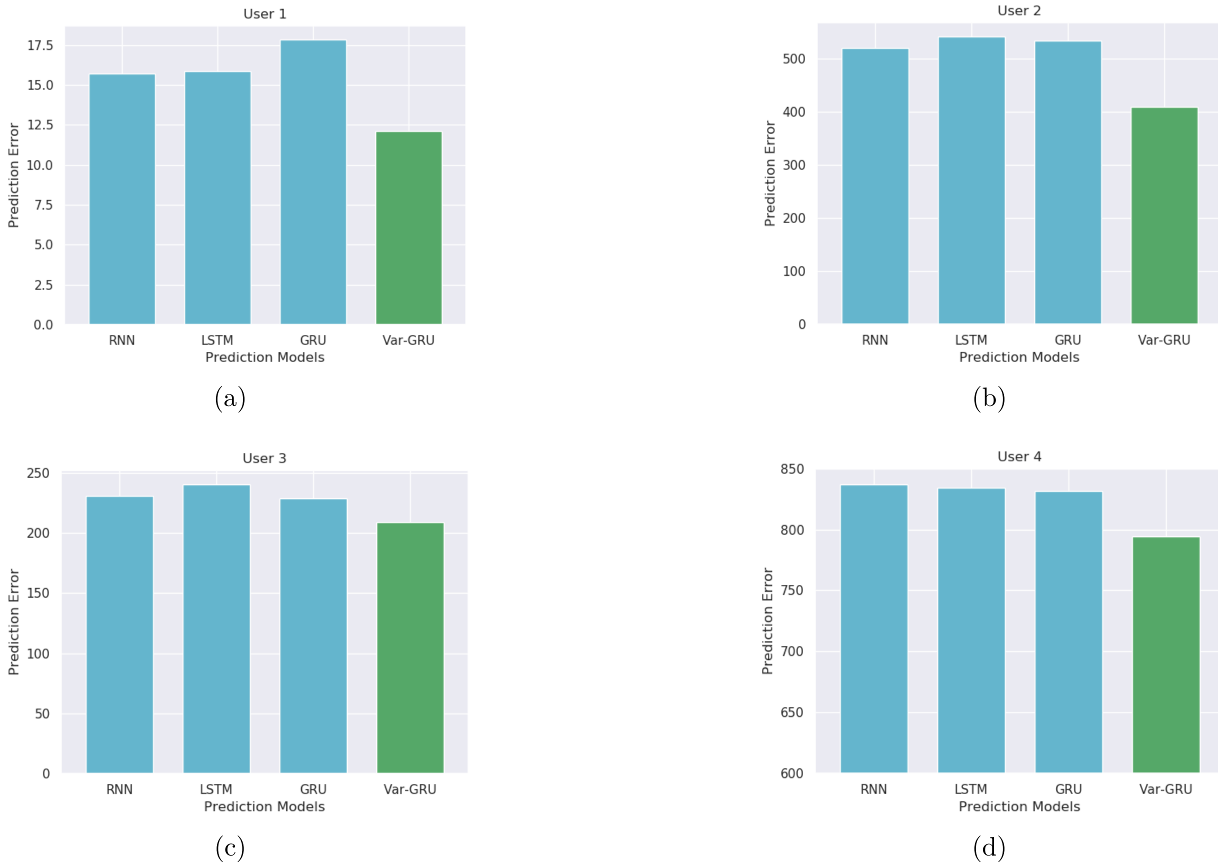


Figure 6.6 Impact of different mobility behaviours on the performance of the mobility models. (a) User 1, (b) User 2,(c) User 3 and (d) User 4.

respectively, 23.3 and 23.2. However, for GRU and the proposed VAR-GRU models, this cost was reduced to 22 and 21.67.

In Table 6.5, the handover processing costs for non-predictive (i.e., conventional) and predictive (proposed method) scenarios are presented. As mentioned earlier, this cost is defined based on the number of handover messages required for HO procedure completion. According to Table 6.5, for horizontal handover (see Figure 6.3), the proposed method effectively reduces the number of messages from 12 to 7 (i.e., 41.66% improvement). However, if the target gNB was predicted incorrectly, 14 messages are needed to complete the HO. Additionally, for a vertical handover (see Figure 6.4), the proposed method improves the processing cost by 57.14% (from 28 to 12) and it needs 34 messages for the case with a wrong prediction. Therefore, it is vital to consider the impact of wrong predictions in our evaluations as well. The obtained processing costs suggest that not having any prediction is more efficient than making decisions based on wrong predictions. Hence, predictor accuracy is of primary importance.

Table 6.3 Mean absolute error value (MAE) for all of the techniques.

Users	RNN	LSTM	GRU	VAR-GRU
User 1	12.55	11.75	12.10	5.35
User 2	308.46	317.09	306.99	240.11
User 3	121.74	124.01	117.41	121.50
User 4	497.07	502.05	497.01	474.87

Table 6.4 The delay regarding each link type in rgw HO procedure [1, 2].

Link Type	Delay
UE to NG-RAN	1 ms
NG-RAN to AMF	7.5 ms
AMF to SeMMu (PGW-C + SMF)	1 ms
SeMMu to S-GW	7.5 ms
SeMMu to PGW-U+UPF	7.5 ms
SeMMu to PCRF+PCF	7.5 ms
AMF to AMF	15 ms
SeMMu to PGW	7.5 ms
SeMMu to E-UTRAN	7.5 ms
E-UTRAN to UE	1 ms
PGW to PCRF	7.5 ms
S-GW to PGW	7.5 ms
SeMMu to SGSN	1 ms
SGSN to RNC	6 ms
SGSN to S-GW	7.5 ms
SeMMu to SeMMu	15 ms

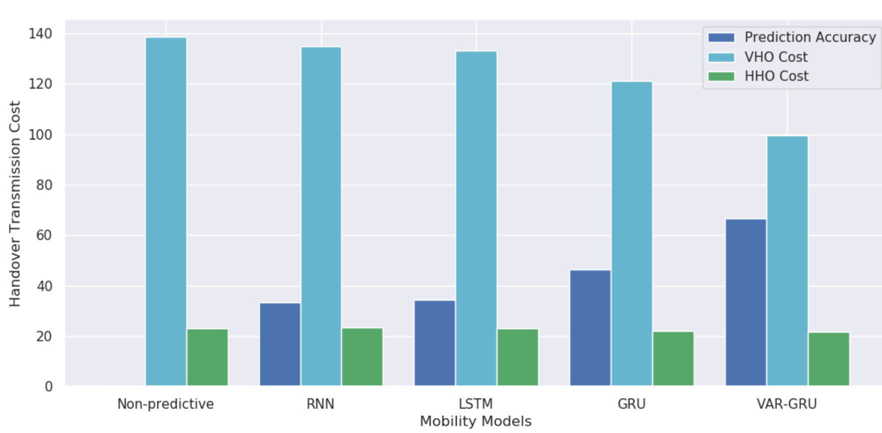


Figure 6.7 Handover transmission cost.

Table 6.5 Handover Processing cost for non-predictive and predictive scenarios when we use horizontal HO (HHO) and vertical HO (VHO).

Handover Type	**Approach**	**Processing Cost**
HHO	Conventional Handover Signaling	12 Messages
	Proposed Handover Signaling with right prediction	7 Messages
	Proposed Handover Signaling with wrong prediction	14 Messages
VHO	Conventional Handover Signaling	28 Messages
	Proposed Handover Signaling with right prediction	12 Messages
	Proposed Handover Signaling with wrong prediction	34 Messages

6.6 Conclusions

In this paper, we highlighted the significance of self-organizing networks for future network management using learning techniques. Our main objective was to deploy mobility prediction as a promising tool for mobile network management. We introduced a hybrid VAR-GRU mobility model to predict user future trajectory. The proposed mobility model was able to extract user mobility behaviour and repetitive patterns in their movement history. After predicting users' future trajectories, we effectively reduced the required number of handover signalings and optimized the HO signaling procedure based on our predictions.

Moreover, we investigated the handover signaling costs for the predictive and non-predictive scenarios to analyze the impact of mobility awareness on handover-related costs. The simulation results showed that the HO processing cost and transmission cost are reduced when we use mobility prediction. Our experiments indicate that the accuracy of prediction is of great importance and has a leading role in the effectiveness of the proposed method. It was shown that the proposed mobility model had the lowest error in comparison with the baseline methods.

For future work, the proposed hybrid VAR-GRU model can be exploited in many potential areas to further optimize cellular network services. Mobility-aware resource allocation, location-based services, and mobility-aware call admission control mechanisms are some of the promising applications for the proposed mobility prediction model.

CHAPTER 7 GENERAL DISCCUSSION

In this chapter, we provide a brief summary of the thesis outcomes with regard to the research objectives that we initially defined in Section 1.3. Subsequently, we summarize the methodological approach that have been deployed towards fulfilling those objectives in this work. Lastly, we conclude this chapter with an analysis of the main results that we have achieved and their practical significance.

7.1 Summary of thesis outcomes

The global objective of this thesis was to propose a mobility prediction models in self-organizing cellular networks that can best meet the demands of providing intelligent mobility management in terms of increasing accuracy performance with lower time complexity. In the previous chapters, we elaborated our approach towards fulfilling this main objective through three main phases. In the following, we briefly provide a high level explanation of these phases.

In the first phase, we proposed a RNN-based user trajectory prediction approach using a preprocessed dataset. In this work, not only we concentrate on learning user mobility behavior but also underline the importance of processing raw mobility data before training the model. This scheme has two main parts. In the first part, we focused on trajectory preparation before mobility learning using line simplification techniques. We attentively studied five different line simplification techniques and their impacts on generating a similar path but with fewer number of samples. In the second part, we applied RNN, GRU and LSTM models to the modified trajectory sequence to predict the future trajectory of the user.

In the second phase, we intended to push the performance of our model to the next level. Hence, we added a new step to the trajectory preparation part to clean the noisy data. We employed Kalman filter to carefully update and correct the noisy measurements. Moreover, we extended our mobility model to a bidirectional learning model. In this model, we highlighted the importance of the impact of the future locations of the user as well as its previous locations. The bidirectional model based on GRU analyzes the trajectory in both directions of the user mobility sequence.

In the third phase, we added a second training step to the previous model. this two-step training procedure is performed by vector autoregression and GRU models. This can have a profound impact on enhancing mobility learning performance. Then, having predicted future locations of the user, we modified handover signaling procedure for the both horizontal and vertical handovers.

7.2 Experimental configurations

To study the effectiveness of the proposed mobility models, we carried out a series of experiments. In the following, we sumjmarized the details regarding the hardware and software that had been used for the experiments.

- Hardware: all the simulations were executed using Intel core i7-6700k, CPU with 4.00 GHz and 32 GB RAM.
- Software: we deployed Keras library for our simulations that is a neural network library in Python (https://keras.io/). We used a neural network with 3 layers (i.e., one hidden layer) and 100 neurons in each layer for all the mentioned methods. We set the drop-out rate to 0.2 (i.e., 20 percent of the connections are randomly dropped in the training phase). Also, we set the learning rate to 0.001 and used Adam optimizer in our experiments.
- User mobility data: we deployed three mobility datasets from real-world users to examine the efficacy of the proposed mobility models. We included more than one dataset in our experiments to show that our models work robustly with different users from different environments. These datasets are: (1)Geolife GPS trajectory dataset, (2)Open street map GPS traces and (3)T-drive trajectory dataset (for more details see chapter 4).
- Measures of quantitative assessment: we investigateD the performance of the proposed methods to assess the potentials of them using several metrics including mean square error, mean absolute error, root mean square error, execution time, handover transmission cost and handover processing costs.

7.3 Obtained results

We present the obtained results for each proposed mobility model in three parts:

(1) RNN-Based User Trajectory Prediction Model: In the data preparation phase, the performance of five line simplification techniques applying on user movement history with the purpose of trajectory simplification is shown. Results showed that the predefined threshold plays a leading role in the accuracy of the modified trajectory. Too small or too large value of threshold results in respectively data redundancy and data sparsity. Given the same threshold nth point method is the fastest method. For the same sequence length, D-P method slightly has the highest execution time. Analyzing different line simplification techniques with different thresholds, we realized that selecting the right technique and threshold value results in not only huge time complexity reduction but also prediction error reduction as well. This is a highly important finding and can be a potential game changer since it can address the main drawback of applying deep learning techniques (i.e., high time complexity). Among these trajectory simplification techniques, D-P method gives the most accurate

modified trajectory close to the original user track with significant data reduction. In the mobility learning phase, RNN, LSTM and GRU models were used to extract mobility behavior features. Results show that D-P has the lowest error predicting the future trajectory using a LSTM-based model. Overall, lowest prediction error belongs to a LSTM-based model using D-P technique to compress the raw mobility data. The proposed mobility model improves error performance and time complexity up to respectively 31% and 79% in comparison with the case that there is no data reduction.

(2) Bidirectional Trajectory Prediction Model: we thoroughly studied the impact of five trajectory simplification methods along with kalman filter to clean the raw mobility data. We implemented alternative mobility models based on linear regression, support vector regression, RNN, LSTM and GRU models to compare the performance of them with the proposed model. Results show that BTPM achieves a model error of 0.014 and decreases the execution time up to 97%. The key distinguishing feature of our model is the fact that it is comparably robust dealing with different mobility behaviors in comparison with the alternative approaches. To evaluate the robustness of BTPM, we investigated the impact of user mobility behavior and the length of available data for each user. Mobility behavior and the amount of available data vary from one individual to another. We considered both of these factors in our experiments to compare different mobility models in terms of being practical in more close to real-world scenarios. Simulation results showed that BTPM has the lowest model error learning the mobility behavior of different users with different trajectory length. We defined a new performance indicator called "Range" which indicates the difference between the lowest and highest error. BTPM has the lowest range value and therefore lower error variance dealing with different individuals mobility behavior.

(3) Hybrid User Mobility Prediction Approach: The proposed hybrid mobility model is examined based on different users and performance metrics. The length of input sequence n affects the performance of the prediction. There is a trade-off between mobility learning and sequence length. Extracting mobility patterns analyzing either a too small or too large value of n results in higher prediction errors. The proposed VAR-GRU mobility model has the lowest prediction error for all values of n. Moreover, handover processing and transmission costs improve respectively by 57.14% and 28.01% for a vertical handover.

CHAPTER 8 CONCLUSION

In this chapter, we provide a summary of the work carried out within the framework of of this thesis by presenting our main contributions. Then, the inherent limitations regarding our work are presented. Subsequently, a list of recommendations that can be the subject of the future works are provided in Section 8.3.

8.1 Summary of Works

Foreseeable future communication networks should be able to cope well with rapid technological advances to provide high quality of services for mobile users. This thesis highlighted the undeniable need for designing intelligent mobility management for future advanced mobile networks. The core idea is to accurately predict the future trajectory of a mobile user based on the mobility behavior of the user in the past experiences. This mobility behavior can be extracted from the user frequent patterns since user mobility tends to be highly repetitive. To this end, we proposed three potential approaches to model user mobility and predict future trajectory using the powerful tools of deep neural networks and statistical models.

In the first approach, we proposed a unidirectional method based on the recurrent neural networks to model the user mobility behavior. In this method, we first fully exploit line simplification techniques to simplify user trajectory and then extract mobility pattern of the user. Simulation results showed a remarkable time complexity reduction from 4616s to 932s. The proposed method obtains a prediction error of 0.10. In the second approach, we proposed a novel bidirectional user trajectory prediction model. This technique can carefully observe and discover frequent patterns in user mobility behavior using both forward and backward user movement history analysis. A set of comprehensive experiments were conducted to study the performance of the proposed mobility model. We achieved a model loss value of 0.014 and reduced execution time up to 97%. Finally, the last approach proposed a hybrid mobility model compsed of two-step training procedure. Deploying this model provide a deeper understanding of the existing correlation of data samples using vector autoregression and deep learning. Experimental results demonstrated that this model is capable of accurately make predictions. We investigated the impact of this prediction on handover related costs. It is proven that it can improve handover transmission and processing costs significantly in comparison with the scenario that we have no prediction in the network. Main contributions of our work are summarized in the following.

- Proposing a preprocessing phase to carefully prepare the mobility dataset specifically suitable for the trajectory prediction model. The main concern is to deal with the existing underlying issues related to the raw datasets. This phase helps to address two fundamental issues of redundancy and noisiness before training and inference steps. Kalman filter and line simplification algorithms are deployed to reduce the noise and obtain significantly lower execution time. This process helps to yield more accurate results.

- Proposing an RNN-based approach for user future trajectory prediction in which user location sequence is investigated chronologically. Recurrent neural networks and its variants are carefully compared to prove their performance in modeling user mobility.

- Proposing a novel trajectory prediction approach based on bidirectional recurrent neural networks that enables highly accurate predictions. The proposed mobility model analyzes both forward and backward correlations in the user past movement history which gives us a deeper understanding of user mobility behavior. To the best of our knowledge, it is the first time that a model based on bidirectional gated recurrent unit (BiGRU) has been deployed for user trajectory prediction.

- Proposing a hybrid mobility prediction technique to model the user mobility behavior. The proposed VAR-GRU model predicts the future trajectory (i.e., path) of the user. The core concept is to fully analyze the existing dependencies in the user past trajectories and extract general pattern in the data.

- Investigating the impact of user mobility prediction on the conventional handover signaling procedure. Handover processing and transmission costs are evaluated to compare the predictive and non-predictive scenarios.

- Conducting comprehensive experiments on data generated from real-world users to show the effectiveness of the presented models and provide an in-depth analysis of the proposed approach. These experiments have evaluated the models based on the various number of metrics.

8.2 Limitations

This thesis made the aforementioned contributions towards user mobility prediction in mobile networks. However, there are some inherent limitations including:

- User-specific mobility model: we investigated user-specific mobility models in our work. First, these models learn from each individual movement history and predict the future trajectory corresponding to that person. Therefore, to have a proper mobility behavior analysis, mobility predictor needs to be trained for each user before making any predictions and for the next user we need to start from the training phase and re-train the model based on that user historic data.

- Data preparation limitation: we proposed a comprehensive preprocessing step to clean and prepare the data before pattern extraction. This process results in a tailor-made user data that is precisely suitable for the proposed mobility prediction approaches. However, the proposed data preparation phase is best compatible for time-stamped datasets such as GPS data.

- Mobilty data type limitation: our research underlined the importance of user trajectory prediction based on the user prior GPS trajectories. We enhanced the model accuracy and time complexity performance focusing on extracting patterns in user raw GPS tracks. However, as explained earlier, there are several types of mobility data types such as CDR datasets, check-in datasets, etc.

- Privacy concerns: our research is highly dependent on user mobility data record. Not all users can trust and share their data. Hence, the scale of examining users mobility behaviors is limited to the users that authorize the permission to conduct research on their movement history.

8.3 Future Research

For future work, there are several potential research areas that can benefit from this thesis contributions. In the following, we provide a list of future research directions using the findings of this thesis.

- The proposed preprocessing phase can be extended for other data types as well for the future work. In order to develop new data cleaning techniques for other mobility data types, issues related to that specific data should firstly be recognized and then carefully addressed to avoid accuracy degradation.

- In the scope of this thesis, we investigated time-stamped location data as an input for the mobility model. We deployed several user mobility datatsets to analyze mobile user mobility behaviors. One potential area as future work is to deploy and investigate the impact of using other types of mobility data. The proposed bidirectional mobility model (BTPM) and VAR-GRU model can be exploited to extract user patterns using other types of user mobility data (e.g., CDR datasets, check-in datasets).

- The proposed mobility models can be used in many potential areas as a potential tool for providing mobility-aware services in self-organized cellular networks. Mobility-aware resource allocation, location-based services, and mobility-aware call admission control mechanisms are some of the promising applications that can be exploited to further optimize cellular network services.

- This work investigates user-specific mobility models for each individual in the network. These user-specific mobility models can be extended to a generic mobility model. A mobility predictor that can be trained using a big dataset and then be utilized for different users to extract their movement patterns.
- In chapter 6, we proposed a hybrid VAR-GRU model for a two-step training procedure. For the vector autoregression model, we defined a VAR of order 3 with three variables. More number of variables can be added to the model. Increasing the number of variables can possibly result in a more accurate prediction since the model is learning from more features. However, there is a accuracy-complexity trade-off that needs to be considered carefully.

REFERENCES

[1] C. Wang, L. Ma, R. Li, T. Durrani, and H. Zhang, "Exploring trajectory prediction through machine learning methods," *IEEE Access*, vol. PP, pp. 1–1, 07 2019.

[2] Sangchul Oh, Byunghan Ryu, and Yeonseung Shin, "Epc signaling load impact over s1 and x2 handover on lte-advanced system," in *2013 Third World Congress on Information and Communication Technologies (WICT 2013)*, 2013, pp. 183–188.

[3] M. KAFSI, "Mining, modeling and predicting mobility," Ph.D. dissertation, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2015.

[4] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks," *IEEE Communications Surveys Tutorials*, vol. 19, no. 4, pp. 2392–2431, Fourthquarter 2017.

[5] R. Di Taranto, S. Muppirisetty, R. Raulefs, D. Slock, T. Svensson *et al.*, "Location-Aware Communications for 5G Networks: How location information can improve scalability, latency, and robustness of 5G," *IEEE Signal Processing Magazine*, vol. 31, no. 6, pp. 102–112, Nov 2014.

[6] C. Song, Z. Qu, N. Blumm, and A.-L. Barabási, "Limits of predictability in human mobility," *Science*, vol. 327, no. 5968, pp. 1018–1021, 2010. [Online]. Available: https://science.sciencemag.org/content/327/5968/1018

[7] Z. Zhao, H. N. Koutsopoulos, and J. Zhao, "Individual mobility prediction using transit smart card data," *Transportation research part C: emerging technologies*, vol. 89, pp. 19–34, 2018. [Online]. Available: http://dx.doi.org/10.1016/j.trc.2018.01.022

[8] H. Zhang and L. Dai, "Mobility prediction: A survey on state-of-the-art schemes and future applications," *IEEE Access*, vol. 7, pp. 802–822, 2019.

[9] G. Xu, S. Gao, M. Daneshmand, C. Wang, and Y. Liu, "A Survey for Mobility Big Data Analytics for Geolocation Prediction," *IEEE Wireless Communications*, vol. 24, no. 1, pp. 111–119, February 2017.

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

[11] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[12] P. Fazio, F. De Rango, and M. Tropea, "Prediction and QoS Enhancement in New Generation Cellular Networks With Mobile Hosts: A Survey on Different Protocols and Conventional/Unconventional Approaches," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1822–1841, thirdquarter 2017.

[13] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini *et al.*, "A Survey of Anticipatory Mobile Networking: Context-Based Classification, Prediction Methodologies, and Optimization Techniques," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1790–1821, thirdquarter 2017.

[14] A. Mohamed, O. Onireti, M. A. Imran, A. Imran, and R. Tafazolli, "Predictive and Core-Network Efficient RRC Signalling for Active State Handover in RANs With Control/Data Separation," *IEEE Transactions on Wireless Communications*, vol. 16, no. 3, pp. 1423–1436, March 2017.

[15] C. Yu, Y. Liu, D. Yao, L. T. Yang, H. Jin *et al.*, "Modeling user activity patterns for next-place prediction," *IEEE Systems Journal*, vol. 11, no. 2, pp. 1060–1071, 2017.

[16] S. Qiao, D. Shen, W. Xiaoteng, N. Han, and W. Zhu, "A Self-Adaptive Parameter Selection Trajectory Prediction Approach via Hidden Markov Models," *Intelligent Transportation Systems, IEEE Transactions on*, vol. 16, pp. 284–296, 02 2015.

[17] V. Kulkarni, A. Mahalunkar, B. Garbinato, and J. D. Kelleher, "On the Inability of Markov Models to Capture Criticality in Human Mobility," *CoRR*, vol. abs/1807.11386, 2018. [Online]. Available: http://arxiv.org/abs/1807.11386

[18] X. Vasilakos, M. Al-Khalidi, V. A. Siris, M. J. Reed, N. Thomos *et al.*, "Mobility-based Proactive Multicast for Seamless Mobility Support in Cellular Network Environments," in *Proceedings of the Workshop on Mobile Edge Communications*, ser. MECOMM '17. New York, NY, USA: ACM, 2017, pp. 25–30. [Online]. Available: http://doi.acm.org/10.1145/3098208.3098213

[19] S. Misra, S. Singh, and M. Khatua, "Miracle: Mobility prediction inside a coverage hole using stochastic learning weak estimator," *IEEE transactions on cybernetics*, vol. 46, 10 2015.

[20] D. Lian, X. Xie, V. W. Zheng, N. J. Yuan, F. Zhang *et al.*, "CEPR: A Collaborative Exploration and Periodically Returning Model for Location Prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 1, pp. 8:1–8:27, Apr. 2015. [Online]. Available: http://doi.acm.org/10.1145/2629557

[21] A. Chakraborty, L. E. Ortiz, and S. R. Das, "Network-side positioning of cellular-band devices with minimal effort," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, 2015, pp. 2767–2775.

[22] H. Farooq and A. Imran, “Spatiotemporal Mobility Prediction in Proactive Self-Organizing Cellular Networks,” *IEEE Communications Letters*, vol. 21, no. 2, pp. 370–373, Feb 2017.

[23] P. Fazio, M. Tropea, F. D. Rango, and M. Voznak, “Pattern Prediction and Passive Bandwidth Management for Hand-over Optimization in QoS Cellular Networks with Vehicular Mobility,” *IEEE Transactions on Mobile Computing*, vol. 15, no. 11, pp. 2809–2824, Nov 2016.

[24] H. Lee, M. Wicke, B. Kusy, O. Gnawali, and L. Guibas, “Predictive Data Delivery to Mobile Users Through Mobility Learning in Wireless Sensor Networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, pp. 1–1, 12 2015.

[25] Y. Chon, E. Talipov, H. Shin, and H. Cha, “Smartdc: Mobility prediction-based adaptive duty cycling for everyday location monitoring,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 3, pp. 512–525, 2014.

[26] Y. Chon, Y. Kim, H. Shin, and H. Cha, “Adaptive duty cycling for place-centric mobility monitoring using zero-cost information in smartphone,” *IEEE Transactions on Mobile Computing*, vol. 13, no. 8, pp. 1694–1706, 2014.

[27] A. Nadembega, A. Hafid, and T. Taleb, “A Destination and Mobility Path Prediction Scheme for Mobile Networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2577–2590, June 2015.

[28] T. Anagnostopoulos, C. Anagnostopoulos, and S. Hadjiefthymiades, “Efficient Location Prediction in Mobile Cellular Networks,” *IJWIN*, vol. 19, pp. 97–111, 2012.

[29] J. Xu, J. Zhao, R. Zhou, C. Liu, P. Zhao *et al.*, “Destination prediction a deep learning based approach,” *IEEE Transactions on Knowledge and Data Engineering*, vol. PP, pp. 1–1, 08 2019.

[30] L. Wang, Z. Yu, B. Guo, T. Ku, and F. Yi, “Moving destination prediction using sparse dataset: A mobility gradient descent approach,” *ACM Trans. Knowl. Discov. Data*, vol. 11, no. 3, apr 2017. [Online]. Available: https://doi.org/10.1145/3051128

[31] S.-B. Cho, “Exploiting Machine Learning Techniques for Location Recognition and Prediction with Smartphone Logs,” *Neurocomput.*, vol. 176, no. C, pp. 98–106, Feb. 2016. [Online]. Available: http://dx.doi.org/10.1016/j.neucom.2015.02.079

[32] A. Nadembega, A. Hafid, and T. Taleb, “Mobility-prediction-aware bandwidth reservation scheme for mobile networks,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 6, pp. 2561–2576, 2015.

[33] M. Lin and W.-J. Hsu, “Mining gps data for mobility patterns: A survey,” *Pervasive and Mobile Computing*, vol. 12, pp. 1–16, 2014. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1574119213000825

[34] J. J.-C. Ying, W.-C. Lee, and V. S. Tseng, "Mining geographic-temporal-semantic patterns in trajectories for location prediction," *ACM Trans. Intell. Syst. Technol.*, vol. 5, no. 1, Jan. 2014. [Online]. Available: https://doi.org/10.1145/2542182.2542184

[35] T.-Y. Fu and W.-C. Lee, "Trembr: Exploring road networks for trajectory representation learning," vol. 11, no. 1, Feb. 2020. [Online]. Available: https://doi.org/10.1145/3361741

[36] R. Bhandari, M. Chaudhary, B. Raman, K. Ramakrishnan, D. Chander *et al.*, "Crowdloc: Cellular fingerprinting for crowds by crowds," *ACM Transactions on Sensor Networks*, vol. 14, pp. 1–36, 01 2018.

[37] D. Zhang, D. Zhang, H. Xiong, L. T. Yang, and V. Gauthier, "Nextcell: Predicting location using social interplay from cell phone traces," *IEEE Transactions on Computers*, vol. 64, no. 02, pp. 452–463, feb 2015.

[38] H. Wu, W. Sun, B. Zheng, L. Yang, and W. Zhou, "Clsters: A general system for reducing errors of trajectories under challenging localization situations," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 1, no. 3, sep 2017. [Online]. Available: https://doi.org/10.1145/3130981

[39] R. Mohamed, H. Aly, and M. Youssef, "Accurate real-time map matching for challenging environments," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 4, pp. 847–857, 2017.

[40] D. Zhang, M. Chen, M. Guizani, H. Xiong, and D. Zhang, "Mobility prediction in telecom cloud using mobile calls," *IEEE Wireless Communications*, vol. 21, no. 1, pp. 26–32, 2014.

[41] A. B. Adege, H. Lin, and L. Wang, "Mobility predictions for iot devices using gated recurrent unit network," *IEEE Internet of Things Journal*, vol. 7, no. 1, pp. 505–517, Jan 2020.

[42] C. Yang, M. Sun, W. X. Zhao, Z. Liu, and E. Y. Chang, "A neural network approach to jointly modeling social networks and mobile trajectories," *ACM Trans. Inf. Syst.*, vol. 35, no. 4, aug 2017. [Online]. Available: https://doi.org/10.1145/3041658

[43] M. J. Chorley, R. M. Whitaker, and S. M. Allen, "Personality and location-based social networks," *Computers in Human Behavior*, vol. 46, pp. 45–56, 2015. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0747563214007559

[44] Y. Chen, C. Long, G. Cong, and C. Li, "Context-aware deep model for joint mobility and time prediction," in *Proceedings of the 13th International Conference on Web Search and Data Mining*, ser. WSDM '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 106–114. [Online]. Available: https://doi.org/10.1145/3336191.3371837

[45] Q. Gao, F. Zhou, G. Trajcevski, K. Zhang, T. Zhong *et al.*, “Predicting human mobility via variational attention,” in *The World Wide Web Conference*, ser. WWW ’19. New York, NY, USA: Association for Computing Machinery, 2019, p. 2750–2756. [Online]. Available: https://doi.org/10.1145/3308558.3313610

[46] Cisco. [Online]. Available: https://www.cisco.com/annual-internet-report/white-paper-c11-741490.html

[47] Y. Zheng, “Trajectory Data Mining: An Overview,” *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 29:1–29:41, May 2015.

[48] L. N. Balico, A. A. F. Loureiro, E. F. Nakamura, R. S. Barreto, R. W. Pazzi *et al.*, “Localization prediction in vehicular ad hoc networks,” *IEEE Communications Surveys Tutorials*, vol. 20, no. 4, pp. 2784–2803, Fourthquarter 2018.

[49] Y. Xing, C. Lv, and D. Cao, “Personalized vehicle trajectory prediction based on joint time-series modeling for connected vehicles,” *IEEE Transactions on Vehicular Technology*, vol. 69, no. 2, pp. 1341–1352, 2020.

[50] Q. Liu, S. Wu, L. Wang, and T. Tan, “Predicting the Next Location: A Recurrent Model with Spatial and Temporal Contexts,” in *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, ser. AAAI’16, 2016, pp. 194–200.

[51] K. Reumann and A. P. M. Witkam, “Optimizing curve segmentation in computer graphics,” 1974.

[52] D. H. Douglas and T. K. Peucker, *Algorithms for the Reduction of the Number of Points Required to Represent a Digitized Line or its Caricature.* John Wiley & Sons, Ltd, 2011, ch. 2, pp. 15–28.

[53] T. Lang, “Rules for the robot draughtsmen.” he Geographical Magazine, 42(1): 50–51, 1969.

[54] M. Visvalingam and J. D. Whyatt, “Line generalisation by repeated elimination of points,” *The Cartographic Journal*, vol. 30, no. 1, pp. 46–51, 1993.

[55] W. Shi and C. Cheung, “Performance evaluation of line simplification algorithms for vector generalization,” *The Cartographic Journal*, vol. 43, no. 1, pp. 27–44, 2006. [Online]. Available: https://doi.org/10.1179/000870406X93490

[56] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, “Understanding transportation modes based on gps data for web applications,” *ACM Transaction on the Web*, vol. 4, pp. 1–36, January 2010.

[57] Cisco, "Cisco Annual Internet Report (2018–2023) White Paper," 2020. [Online]. Available: https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html,urldate={2020-03-09}

[58] P. Fan, J. Zhao, and C. I, "5G high mobility wireless communications: Challenges and solutions," *China Communications*, vol. 13, no. Supplement2, pp. 1–13, N 2016.

[59] N. Bahra and S. Pierre, "Rnn-based user trajectory prediction using a preprocessed dataset," in *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2020, pp. 1–6.

[60] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov 1997.

[61] J. Chung, Ç. Gülçehre, K. Cho, and Y. Bengio, "Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling," *CoRR*, vol. abs/1412.3555, 2014. [Online]. Available: http://arxiv.org/abs/1412.3555

[62] R. Wu, G. Luo, J. Shao, L. Tian, and C. Peng, "Location prediction on trajectory data: A review," *Big Data Mining and Analytics*, vol. 1, no. 2, pp. 108–127, June 2018.

[63] D. Yao, C. Zhang, J. Huang, and J. Bi, "Serm: A recurrent model for next location prediction in semantic trajectories," in *Proceedings of the 2017 ACM on Conference on Information and Knowledge Management*, ser. CIKM '17. New York, NY, USA: Association for Computing Machinery, 2017, p. 2411–2414. [Online]. Available: https://doi.org/10.1145/3132847.3133056

[64] J. Feng, Y. Li, C. Zhang, F. Sun, F. Meng *et al.*, "Deepmove: Predicting human mobility with attentional recurrent networks," in *Proceedings of the 2018 World Wide Web Conference*, ser. WWW '18. Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2018, p. 1459–1468. [Online]. Available: https://doi.org/10.1145/3178876.3186058

[65] Z. Yan, D. Chakraborty, C. Parent, S. Spaccapietra, and K. Aberer, "Semantic trajectories: Mobility data computation and annotation," *ACM Trans. Intell. Syst. Technol.*, vol. 4, no. 3, Jul. 2013. [Online]. Available: https://doi.org/10.1145/2483669.2483682

[66] A. Sadri, F. Salim, Y. Ren, W. Shao, J. Krumm *et al.*, "What will you do for the rest of the day?: An approach to continuous trajectory prediction," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 2, pp. 1–26, 12 2018.

[67] Z. Lu, F. Zhao, H. Luo, Q. Wang, and W. Shao, "A destination prediction algorithm using spatial temporal bidirectional lstm networks," in *IPIN*, 2019.

[68] Y. Leng. (2016) Urban Computing using Call Detail Records: Mobility Pattern Mining, Next-location Prediction and Location Recommendation.

[69] B. Altaf, L. Yu, and X. Zhang, "Spatio-temporal attention based recurrent neural network for next location prediction," *2018 IEEE International Conference on Big Data (Big Data)*, pp. 937–942, 2018.

[70] F. Zhou, X. Yue, G. Trajcevski, T. Zhong, and K. Zhang, "Context-aware variational trajectory encoding and human mobility inference," in *The World Wide Web Conference*, ser. WWW '19. New York, NY, USA: Association for Computing Machinery, 2019, p. 3469–3475. [Online]. Available: https://doi.org/10.1145/3308558.3313608

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

[72] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, Nov. 1997. [Online]. Available: http://dx.doi.org/10.1162/neco.1997.9.8.1735

[73] G. Welch and G. Bishop, "An Introduction to the Kalman Filter," Chapel Hill, NC, USA, Tech. Rep., 1995.

[74] W. Florian. (2016) Handling GPS data with Python. [Online]. Available: https://doi.org/10.5446/21141

[75] C. Montella, "The kalman filter and related algorithms: A literature review," 05 2011.

[76] A. Smola and B. Schölkopf, "A tutorial on support vector regression," *Statistics and Computing*, vol. 14, pp. 199–222, 08 2004.

[77] D. E. Rumlehart, P. Smolensky, J. L. McClelland, and G. E. Hinton, *Schemata and Sequential Thought Processes in PDP Models*. Cambridge, MA, USA: MIT Press, 1986, p. 7–57.

[78] A. Gron, *Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*, 1st ed. O'Reilly Media, Inc., 2017.

[79] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W.-Y. Ma, "Understanding Transportation Modes Based on GPS Data for Web Applications," *ACM Trans. Web*, vol. 4, no. 1, pp. 1:1–1:36, Jan. 2010.

[80] J. Yuan, Y. Zheng, X. Xie, and G. Sun, "Driving with knowledge from the physical world," in *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD'11. New York, NY, USA: Association for Computing Machinery, 2011, p. 316–324. [Online]. Available: https://doi.org/10.1145/2020408.2020462

[81] J. Yuan, Y. Zheng, C. Zhang, W. Xie, X. Xie *et al.*, "T-drive: Driving directions based on taxi trajectories," in *Proceedings of the 18th SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS'10. New York,

NY, USA: Association for Computing Machinery, 2010, p. 99–108. [Online]. Available: https://doi.org/10.1145/1869790.1869807

[82] OpenCellID. The world's largest Open Database of Cell Towers. (https://opencellid.org/). [Online]. Available: https://opencellid.org/

[83] E. Zivot and J. Wang, *Vector Autoregressive Models for Multivariate Time Series*. New York, NY: Springer New York, 2003, pp. 369–413. [Online]. Available: https://doi.org/10.1007/978-0-387-21763-5_11

[84] J. Xu, J. Zhao, R. Zhou, C. Liu, P. Zhao *et al.*, "Predicting destinations by a deep learning based approach," *IEEE Transactions on Knowledge and Data Engineering*, vol. 33, no. 2, pp. 651–666, 2021.

[85] N. Bahra and S. Pierre, "Rnn-based user trajectory prediction using a preprocessed dataset," in *2020 16th International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, 2020, pp. 1–6.

[86] M. Ozturk, M. Gogate, O. Onireti, A. Adeel, A. Hussain *et al.*, "A novel deep learning driven, low-cost mobility prediction approach for 5g cellular networks: The case of the control/data separation architecture (cdsa)," *Neurocomputing*, vol. 358, pp. 479–489, 2019. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231219300438

[87] A. Mohamed, O. Onireti, S. A. Hoseinitabatabaei, M. Imran, A. Imran *et al.*, "Mobility prediction for handover management in cellular networks with control/data separation," in *2015 IEEE International Conference on Communications (ICC)*, 2015, pp. 3939–3944.

[88] 5G, "Procedures for the 5g system (5gs)," *(3GPP TS 23.502 version 15.4.1 Release 15)*, vol. 1, 2019.

[89] A. Jain, E. Lopez-Aguilera, and I. Demirkol, "Evolutionary 4g/5g network architecture assisted efficient handover signaling," *IEEE Access*, vol. 7, pp. 256–283, 2019.

APPENDIX A

The standard procedure of kalman filter algorithm has two main steps including time update and measurement update [73]. Equations for time update (prediction) are

$$\hat{x}_k^- = B\hat{x}_{k-1} + Cu_{k-1}, \tag{A.1}$$

$$P_k{}^- = Bp_{k-1}B^T + Q, \tag{A.2}$$

and the equations for measurement update (correction) are

$$K_k = P_k{}^- H^T (HP_k{}^- H^T + R)^{-1}, \tag{A.3}$$

$$\hat{x}_k = \hat{x}_k^- + K_k(M_k - H\hat{x}_k^-), \tag{A.4}$$

$$P_k = (I - K_k H)P_k{}^-. \tag{A.5}$$

The description for the variables are as follows:

$\hat{x}_k$: Posterior estimation of process states at time step k

$\hat{x}_k^-$: Prior estimation of process states at time step k

B: Matrix B Shows the relationship between two consecutive timesteps

C: Matrix C indicates the relationship between the control input and the states

u_k: Control input vector

P_k: Posterior estimation of error covariance

$P_k{}^-$: Prior estimation of error covariance

Q: Process noise covariance

R: Measurement noise covariance

K_k: Kalman filter gain

H: Measurement matrix

M_k: Measurement vector at time step k