Statistical models and algorithms for the analysis of human mobility data from smartphone applications

Author

Muhammad Haroon Shaukat

Matricola No. 1086469

Cycle XXXVIII (2022-2025)

Supervised By

Prof. Francesco Finazzi

PhD Coordinator

Prof. Alessandra Marini

Doctoral Program in Engineering and Applied Sciences

Department of Engineering and Applied Sciences

University of Bergamo, Italy

2025

Declaration of Authorship

I, Muhammad Haroon Shaukat, declare that this thesis titled, "Statistical models and algorithms for the analysis of human mobility data from smartphone applications" and the work presented in it is my own. I confirm that.

• This work was done wholly in candidature for a degree of Dottorato Di Ricerca at this University.

• Where I got help from the published work of others, this is always clearly stated.

• Where I have quoted from the work of others, the source is always mentioned. Except of such quotations, this thesis is entirely my own research work.

• Where the thesis is based on work done by myself jointly with my supervisor, I have made clear exactly what was done by others and what I have suggested.

Signed: .....

Date: .....

Acknowledgments

I would like to express my deepest gratitude to my supervisor, **Prof. Francesco Finazzi**, for their continuous guidance, valuable feedback, and encouragement throughout this research journey. Their support and knowledge have been a true source of inspiration for me. Also, I would like to express my heartiest gratitude to my other respected professors **Prof. Alessandro Fassò** and **Prof. Rodolfo Metulini**, who always guided and supported me.

My heartfelt thanks go to my parents, Mr. and Mrs. Shaukat Mehmood, and my wife, Hajra, whose love and moral support have always been my strongest pillars. I am forever indebted to them for believing in me. I am also thankful to my siblings for their support, and I pay my deepest respects to my grandfather (late), whose love for education has been a guiding light in my life.

Above all, I dedicate this thesis to my dearest son, Shahmeer Haroon, whose shining presence in my life gave me the courage to pass through every thick and thin. His smile has been my greatest motivation.

Lastly, I extend my gratitude to my colleagues, friends, and all those who contributed, directly or indirectly, to the successful completion of this thesis.

Muhammad Haroon Shaukat

Abstract

Urban mobility analysis is essential for understanding how individuals move within cities, as it reveals patterns shaped by personal behavior, social activities, and environmental factors. These insights are critical not only for improving transportation systems but also for guiding smart city development, enhancing location-based services, and ensuring that urban infrastructure effectively responds to the needs of its residents. However, the analysis of mobility datasets presents considerable challenges due to irregularity, sparsity, and the heterogeneous nature of individual movement patterns. In particular, incorporating both spatial and temporal information within a single model remains a significant challenge, and addressing this is essential for improving the accuracy and reliability of mobility predictions.

Therefore, the thesis aims to introduce techniques to analyze and predict smartphone users mobility pattern. The smartphone-based mobility datasets of four individuals are obtained from the earthquake network project, covering the period from March 1 to April 29, 2023. This captures the daily movement patterns of smartphone users whose movements took place within Istanbul, Turkey. The research is structured into two parts. Firstly, the spatiotemporal states are computed that contain the spatial and temporal information, and they represent where and when a smartphone user was present. Then, the next spatiotemporal states are predicted by both statistical and machine learning models. At the start, the first and second-order Markov Chain models are employed, and the outcomes highlight their limited effectiveness due to the high dimensionality of spatiotemporal states. Afterward, the Markov formulation was investigated in the Random Forest Classifier (RFC), Support Vector Classifier (SVC), and Multilayer Perceptron Classifier (MLPC), and assumed that the next spatiotemporal states conditionally depend on the current spatiotemporal state along with their transition probabilities. It is important to emphasize that this represents a modeling formulation rather than an assertion of a strict Markov property. The results

indicate that the RFC outperformed other models across all smartphone users, and it has achieved the highest accuracy of 90.56% for one smartphone user. While the SVC and MLPC exhibited competitive performance. Furthermore, the feature importance analysis indicates that the current spatiotemporal state is a highly influential predictor, while the newly introduced feature as spatiotemporal state transition probability also emerges as a key predictor, highlighting its contribution to capturing underlying mobility dynamics.

Secondly, the smartphone users' locations are predicted at fixed timestamps, and the location prediction focuses on estimating the geographic coordinates of users. To handle the irregular timestamps in the mobility dataset, linear interpolation and a coordinate replacement procedure were applied to regularize the trajectories at fifteen-minute intervals, when specific conditions were met. Subsequently, the Markov formulation was incorporated into the Random Forest Regression (RFR), Support Vector Regression (SVR), and Multilayer Perceptron Regression (MLPR) models to predict coordinates at fifteen-minute intervals. Therefore, it is considered that the coordinates observed fifteen minutes ahead conditionally depend on those recorded fifteen minutes earlier, together with other discrete spatial and temporal features. The evaluation results indicate that the RFR model consistently outperformed the other models across all smartphone users and spatial radii. In contrast, the SVR and MLPR models initially exhibited lower performance but showed improved accuracy at larger radius thresholds. The feature importance analysis confirmed the effectiveness of incorporating the Markov formulation, highlighting the robustness of the RFR model in predicting smartphone users' locations. It is important to note that the presented formulation adopts a Markovian approximation rather than assuming a strict Markov property, as human mobility patterns are known to exhibit temporal dependencies that extend beyond the immediately visited locations. This choice is empirically supported by the results, which demonstrate that incorporating additional historical coordinate information yields negligible performance improvements within fine spatial radii. Additionally, the localized feature explanations through Shapley Additive Explanations (SHAP) indicated how individual spatial and temporal features dynamically shape predictions, and capture personalized movement behaviors and directional trends.

Ultimately, the results suggest that the model's performance varies according to the diverse movement patterns observed in both spatiotemporal state prediction and location prediction. Even so, the Random Forest (RF) model consistently achieved superiority in smartphone users' mobility prediction, revealing its ability to effectively capture individual

mobility behaviors despite the irregularities and complexities present in the dataset. To the best of our knowledge, no prior research has employed models in this manner that utilize feature engineering based on a Markov formulation for spatiotemporal state prediction and location prediction. This represents a significant contribution to the field of smartphone-based individual mobility modeling. These findings offer useful insights for improving urban mobility planning, enhancing location-based services, and supporting smart city systems that rely on accurate mobility prediction.

Contents

List of Abbreviations

RF Random Forest

SVM Support Vector Machine

RFC Random Forest Classification

SVC Support Vector Classification

MLPC Multilayer Perceptron Classification

RFR Random Forest Regression

SVR Support Vector Regression

MLPR Multilayer Perceptron Regression

MC Markov Chain

MC-1 First-Order Markov Chain

MC-2 Second-Order Markov Chain

GPS Global Positioning System

CDRs Call Detail Records

SC Smart Card

IoT Internet of Things

LBS Location Based Services

HM Hidden Markov

WMCM Weighted Markov Chain Model

KVLMC Kernel Variable Length Markov Chain

POIs Points of Interest (POIs)

ROIs Regions of Interest (ROIs)

LSTM Long Short-Term Memory

MSD Mobile Signalling Data

SML Self-supervised Mobility Learning

GRUs Gated Recurrent Units

TUL Trajectory User Linking

CSLSL Causal and Spatial-constrained Long and Short-term Learner

NYC New York City

TKY Tokyo

DL Dallas

ML Machine Learning

HWN Heterogeneous Wireless Networks

Multi-SVMMP Multi-SVM-based Mobility Prediction

SLAW Self-similar Least Action Walk

DNN Deep Neural Networks

BN Bayes Network

MRF Multi-class Random Forest

PHTI Possibilities of Historical Travel Intentions

RMState real-time moving states

STCorrelation spatiotemporal correlations of road intersections

MAE Mean Absolute Error

RMSE Mean Square Error

GMM Gaussian Mixture Model

EEI equal volume, equal shape, and axes aligned with the coordinate system

MDG Mean Decrease in Gini

R-squared Coefficient of Determination

AWR Accuracy Within Radius

MSE Mean Square Error

SHAP Shapley Additive Explanations

ARIMA Autoregressive Integrated Moving Average

ST-GCN Spatial-Temporal Graph Convolutional Networks

DWSTTN Deep Wide Spatial-Temporal Based Transformer Network

ST-LSTM Spatial-Temporal LSTM

DeepAGS Deep learning model with Activity, Geographic, and Sequential information

Chapter 1

Urban Human Mobility

1.1 Introduction to Mobility

Urbanization is happening rapidly all over the world. In 2018, the global population was around 55% about 4.2 billion people were already living in cities. By 2050, it's estimated that nearly 68% of the world's population will be living in urban areas (Dobbs et al., 2011; Nations, 2018). Therefore, it is crucial to understand how and why people travel within cities, suburbs, and surrounding regions. This insight helps city planners and policymakers better predict and manage the demand for infrastructure, such as roads, public transport, and energy systems (Zheng et al., 2014).

A major step toward building smarter and more sustainable cities is understanding how people move within them. By tracking movement in real-time, city planners and authorities can make better and faster decisions. This kind of real-time insight is known as situational awareness and this refers to being aware of what's happening in the environment by using live data, understanding the meaning behind that data, and predicting what might happen next. This type of insight is not only useful for managing daily life in cities but also essential for protecting cities from disasters like forest fires and earthquakes (Mokbel et al., 2024).

Understanding human movements is not only important for planning cities, but it is also critical for public health. The way people travel and interact in urban spaces has a direct impact on how infectious diseases spread (Hou et al., 2021). Researchers have increasingly utilized mobility datasets to forecast outbreaks and better understand how diseases spread, and this approach is often referred to as data-driven epidemic forecasting (Rodríguez et al., 2022). One practical example of using mobility data in public health is contact tracing.

This involves identifying people who have been in close contact with someone infected and then gathering more information to stop further spread (Mokbel et al., 2020).

Urban human mobility refers to the way people move around within cities and it's all about understanding daily movement patterns in urban spaces like commuting to work, going to school, running errands, or visiting friends. Importantly, this concept is specifically focused on mobility inside city boundaries. So, it doesn't usually include long-distance travel like going from one city to another or traveling to rural areas. This means that things like train rides between cities or flights are generally outside the scope of urban mobility research Kapp et al. (2023). As cities grow larger and more complex, the study of urban mobility has become a key focus for city planners and researchers to improve public transportation, reduce traffic congestion, and make cities more liveable.

Urban mobility can be studied in two main ways such as collective and individual mobility Ma and Zhang (2022). When researchers focus on collective mobility, they're studying how groups of people move through a city like spotting rush hour trends, figuring out when buses or trains are busiest, or seeing where taxis and shared bikes are most in demand. This kind of insight is helpful for running city transportation smoothly. For instance, if officials know where traffic usually gets jams, they can adjust traffic signals or add more public transit options to keep things moving (Antoniou et al., 2018). On the other hand, individual mobility focuses on the movement of a single person. It tries to predict where someone is likely to go, when they'll go there, and how they'll get there. This kind of data is especially useful in designing personalized services and recommendations. For example, your navigation app might suggest a route based on your usual habits, or a shopping app might recommend nearby stores based on where you often go. Researchers have explored individual mobility in various fields, including recommendation systems (Zhao et al., 2016), behavioral choice modeling (Ben-Akiva and Lerman, 2021), and even marketing strategies (Danaf et al., 2019). These insights allow businesses and services to better align with real human behavior.

Even though predicting individual human mobility is incredibly valuable, however, it has not received as much attention as it deserves (Feng et al., 2018). One major challenge is that individual behavior is naturally hard to predict as people's routines are not always consistent. Another challenge is limited access to personal movement data, which makes it hard to build accurate models. A major breakthrough came with the rise of big data technologies, which allow huge amounts of location data from individuals to be collected,

stored, and analyzed. This information forms the foundation for studying and predicting personal mobility. Without the ability to handle this kind of data at a large scale, the predictions simply would not be possible.

To understand human movement, researchers often rely on something called trajectories (Kapp et al., 2023). A trajectory is a series of data points that track where a person was and when. Each point in the dataset typically includes a location, represented by latitude and longitude, as well as a timestamp. When these points are ordered by time, they trace the path of a person's movement, much like a trail of breadcrumbs showing where they went. Sometimes, even just knowing the order of locations without exact timestamps can provide meaningful insights. So, researchers often use a broader definition: any dataset that tracks people's movement through space in a city can be considered an urban mobility dataset. Within this mobility dataset, location points are categorized into different types, each helping to reveal patterns in how people move through a city. For instance, stay points are locations where someone spends a significant amount of time like their home, workplace, grocery store, or school. These represent important places in a person's daily routine. On the other hand, waypoints are locations that people simply pass through while traveling between stay points. They don't stop there for long, but these points still help trace their route. Imagine a single taxi ride tracked by Global Positioning System (GPS) and most of the recorded points would be waypoints marking the journey from one stay point like home to another stay point like airport. On the other hand, a travel survey might list only the key places visited in a day that are just stay points. Researchers use the term trajectory semantics to describe the different categories or meanings within a person's movement data like distinguishing between places where someone stops and places they just move through. By analyzing these different types of trajectories, scientists and planners can gain deeper insights into human behavior, improve transportation systems, and design better services. Then, individual mobility prediction is becoming more reliable and valuable than ever before with the help of more accurate data and smarter technology.

1.2 Mobility Data Source

When studying individual-level human mobility in cities, the researchers use different types of datasets depending on the specific problem they're trying to solve. For example, one study might aim to predict the next location a person will visit using GPS data (Calabrese et al., 2010), while another might use smart card data from public transport systems to

predict someone's next trip (Zhao et al., 2018). The choice of dataset shapes how mobility is analyzed and understood. Broadly, individual mobility data can be divided into two main types such as trip data and trajectory data. Trip data only captures moments when actual movement happens. The automatic fare collection system is installed in public transportation that only records when a person taps in or out of a bus or meter station, meaning the data directly reflects an action related to travel. If there's no travel, there's no record. Another type of trip dataset comes from electronic toll collection systems, which record when a vehicle enters and exits a toll station, along with the exact times, vehicle identity, type, and weight (Yang et al., 2019). In contrast, trajectory data captures information continuously, regardless of whether a person is actively traveling or standing still. For instance, GPS trajectory data is often collected at fixed intervals, so even if someone stops to grab a coffee, the system still records their location. This kind of data helps form a much more detailed picture of a person's movement patterns. According to Ma and Zhang (2022), there are several specific sources of trajectory data as described as follows.

- **Call Detail Records:** The Call Detail Records (CDRs) data is collected by telecom companies mostly for billing purposes that log the time and location usually based on the nearest cell tower when someone makes a phone call or sends a text. While it is not originally intended for mobility tracking, it can be used to trace broad movement patterns across a city especially since nearly everyone uses mobile phones regularly.

- **Global Positioning System Traces:** The GPS uses satellites to determine the exact location of devices like smartphones, vehicles, or bikes. This type of data is extremely precise and is now commonly collected by apps, cars, and shared mobility services. It offers high accuracy in tracking how people or vehicles move in real-time.

- **Social Media:** Many social media platforms allow users to share their location when posting photos, updates, or events. These location-tagged posts can be collected and analyzed to understand mobility patterns. Even though the data might be less consistent than GPS or CDRs. However, it still provides a real-time user-driven view of how people move around and engage with urban spaces.

- **Smart Card:** Smart Card (SC) data is collected on public transportation by transport operators and local government bodies for commercial and planning purposes

when the passenger taps their card to board boards or exit from a bus or metro. However, the data is usually limited to predefined locations like bus stops or train stations and records are in the form of check-in and check-out.

- **Synthetic Data:** Synthetic data is artificially generated using models trained on real data to mimic its key patterns and structure. When these models accurately reflect the original data's statistical properties, they can produce an arbitrary number of new trajectories. The main goal is to protect individual privacy while still allowing researchers to access valuable insights without exposing sensitive personal information. These datasets also help evaluate mobility models without privacy concerns and are useful for creating or refining urban planning tools.

Each dataset comes with its own strengths and weaknesses. Some are rich and detailed like GPS, while others offer broader but less granular views like CDR or CS datasets. The right dataset depends on the question being asked whether it's about predicting where someone might go next, understanding how public transport is used, or modeling citywide traffic flow.

1.3 Challenges in Mobility Data Collection

Privacy is a major concern when working with human mobility data. These datasets often contain detailed records of where and when people travel, which can unintentionally reveal personal habits, lifestyle choices, and sensitive information (Chen et al., 2013). To address these concerns, researchers have developed privacy-preserving methods. However, such techniques can reduce the usefulness of the data by removing important mobility details, making meaningful analysis more difficult. Researchers are exploring how to create synthetic movement data that mimics real-world travel patterns. This approach aims to protect individual privacy while still allowing useful insights for things like mobility prediction and urban planning. However, there's still a challenge as it's not yet clear how to measure how realistic the synthetic data should be. If it becomes too close to actual human behavior especially when trained on real data, it might still reveal sensitive details, such as where a household regularly goes, raising new privacy concerns (Mokbel et al., 2024). Another key challenge in working with mobility data is bias. Data collection methods often don't capture everyone equally. For instance, mobile app or phone network data tends to miss people who don't use smartphones or rely on prepaid plans. Similarly, most traffic

sensors are designed to count cars, but they often ignore pedestrians, cyclists, wheelchair users, and other non-motorized forms of travel. Even mobile phone network data can be uneven as cell tower coverage varies in size with rural areas having larger cells that offer less precise location data compared to urban areas. These biases can lead to gaps in understanding how different groups move through space. Moreover, the volunteered tracking data such as it is collected through fitness apps or location-sharing platforms often biased toward people who are more comfortable using digital tools. Similarly, sports tracking data tends to overrepresent health-conscious individuals often from middle to upper socioeconomic groups. As a result, large portions of the populations such as the elderly, low-income communities, or those without access to technology may be underrepresented or entirely excluded from these datasets. This kind of bias can lead to skewed insights and may result in transportation policies or urban planning decisions that serve only a subset of the population if not addressed. That's why it's essential to recognize, measure, and actively work to reduce these biases. It ensures that mobility insights and decisions benefit everyone not just the majority by taking into account the needs of underrepresented and vulnerable groups (Shaham et al., 2022).

1.4 Applications of Mobility Modelling

In the fast-moving digital world, understanding human movement has become incredibly important. This growing interest in human mobility isn't just an academic topic; it's playing a significant role in shaping smarter, more efficient cities. One major reason for this is the rise of Location Based Services (LBS), which are applications that rely on knowing a person's location to provide useful, real-time information or services (Cheng et al., 2016). Consider the everyday conveniences people rely on such as receiving live traffic updates, finding the nearest automated teller machine, getting alerts about sales when near a store, or even receiving personalized news updates based on location. All these are examples of LBS. They've become so effective due to the widespread use of GPS, smartphones, Internet of Things (IoT) devices, and wireless networks. These technologies work together to track people's movements in a seamless and often invisible way, which in turn allows for the delivery of all these smart services. Location prediction is a key area of LBS, leveraging historical movement data to forecast a user's next destination. Accurate prediction enables service providers to deliver timely, context-aware offerings, thereby enhancing both user experience and overall service quality. In marketing, LBS can target advertising more ef-

fectively by sending promotions when someone is near a store or likely to visit. For public services and city management, it helps with better planning of resources like traffic control, public transit routes, and emergency services. Like, the prediction of tomorrow's traffic congestion based on past trends can transform urban planning and service delivery. However, it all depends heavily on the accuracy and reliability of location prediction models. Poor predictions can lead to missed opportunities or irrelevant suggestions, while accurate ones can create a much more connected and responsive urban environment (Chekol and Fufa, 2022). In short, studying human mobility is more than just tracking where people go. It's about unlocking the potential to build smarter, more responsive, and more liveable cities.

1.5 Composition of Thesis

The thesis is organized into six chapters. The Chapter 2 presents a comprehensive review of relevant literature on models and techniques that were used for the analysis of mobility and prediction. The Chapter 3 provides an overview of data structures and preliminary analysis. The Chapters 4 and 5 are dedicated to exploring different approaches for predicting smartphone user mobility. The Chapter 4 focuses on the prediction of spatiotemporal states by analyzing regions of interest over fixed time intervals. In contrast, the Chapter 5 introduces location prediction models, detailing the methodology, findings, and associated limitations. Finally, the Chapter 6 offers concluding remarks and general recommendations based on the study's insights. Also, it includes supplementary material related to the analyses presented in Chapters 4 and 5.

Chapter 2

Literature Review on Models and Techniques for Mobility Modeling

In recent years, human mobility has become a major research focus across multiple disciplines due to its profound impact on social, economic, and environmental dynamics at both global and local scales (Du et al., 2025). Consequently, researchers have proposed numerous models to predict human movement patterns, including approaches based on statistical methods, Machine Learning (ML), and Deep Learning (DL) (Toch et al., 2019; Luca et al., 2021). The existing literature includes diverse methods and algorithms developed for mobility prediction using various types of datasets. Although some studies are based on network-level, synthetic, or vehicle-based mobility data rather than smartphone-based trajectories, they share the common objective of predicting future locations or movement trajectories at the individual level. Thus, these studies remain highly relevant to the broader field of human mobility modeling.

Urban mobility can be broadly classified into collective mobility and individual mobility, depending on the perspective of analysis. Collective mobility focuses on aggregated movement patterns of groups or populations, whereas individual mobility examines the travel behavior of a single person over time. In individual mobility prediction, research mainly addresses two types of problems as next-location prediction, which aims to forecast the immediate future location of an individual, and next-trip prediction, which seeks to predict a trip based on past travel behavior. The study of individual mobility examines patterns of human movement within networks or systems (Keyfitz, 1973). This area has attracted substantial attention across multiple disciplines, including recommendation systems (Zhao

et al., 2016) and marketing science (Danaf et al., 2019). Another key aspect of individual mobility prediction is the prediction horizon, which defines how far into the future the prediction is made. Based on the prediction horizon, prediction tasks are commonly divided into one-step and multi-step predictions. One-step prediction focuses on predicting the next mobility record, while multi-step prediction extends this to forecasting a sequence of future movements, which is more challenging due to increasing uncertainty over time (Ma and Zhang, 2022).

Statistical models have been widely used for predicting individual mobility, and they model the mobility sequence dependencies using probabilistic models such as Markov Chain (MC) and its variants (Ma and Zhang, 2022). However, they are limited in handling long-term patterns, such as periodic routines. The Hidden Markov (HM) and MC models are both popular for modeling sequence data. However, they differ fundamentally in handling observable and hidden states (Loo et al., 2021). MC models assume that states are directly observable, and transitions depend only on the current state, while the HM models introduce hidden layers to model the more complex and unobservable behaviors. This review focuses specifically on next-location and next-trip prediction, highlighting key earlier studies in the area of individual mobility prediction.

Early research in modeling individual mobility often assumed the Markovian property for location visits, where the prediction of the next location depends solely on the current one. As the study of Ashbrook and Starner (2002) utilized this assumption to model human location patterns. Building on this concept, Gambs et al. (2010) introduced the Mobility Markov Chain (MMC) as a probabilistic model designed to capture individual movement sequences. This approach was further applied by (Gambs et al., 2012) in their work on next-place prediction, demonstrating the practical utility of MMCs in forecasting mobility. A study by (Wang et al., 2020a) that enhanced the predictive performance of the MC model and tackled the limitations of traditional mobility prediction methods when applied to app-collected location data, which are typically sparse, unevenly distributed, and partially missing. The study utilized app-collected location records from one million anonymous users in Beijing, spanning from September 17 to October 31, 2016. A context-aware high-order MC model combined with Gibbs sampling was proposed, where context refers to the semantic meaning of a location derived from application usage (e.g., transportation, food, or service-related activities). Based on this contextual information, locations were clustered into street, district, and region-level divisions. In contrast, non-context

approaches rely solely on uniform geographic grids without considering functional semantics. The results showed that the contextual model achieved higher prediction accuracy up to 67%, 78%, and 87% at the street, district, and region levels, respectively, and represented an average improvement of about 10% over non-contextual methods. Despite these gains, the model remains limited in capturing long-range spatiotemporal dependencies, as mobility behavior was found to depend primarily on the most recent location. However, (Qiao et al., 2018) incorporated spatiotemporal characteristics in the proposed hybrid MC model to improve the prediction accuracy, particularly for the mobility data that follows the non-gaussian distribution. Their study utilized a real mobility dataset of 3474 individuals over 21 days from October 10, 2013, to October 31, 2013, with a small granularity of every 12 minutes, and it was collected from long-term evolution in the Chinese city. The proposed model consists of three stages such as pattern discovery, a variable-order Markov predictor that accounts for temporal factors, and a user similarity calculation based on mobility patterns. They also incorporated the temporal characteristics of mobility patterns to improve the predictability of the next location, and results suggest that the proposed algorithm is beneficial for non-gaussian data with more than accuracy of 56.39% while the classical MC model has 44.02% of prediction accuracy for the same dataset. It was also suggested that prediction accuracy varies depending on the datasets, whether crowd or individual, and even across different time periods due to distinctive mobility patterns. Cheng et al. (2016) evaluated the performance of MC-based mobility prediction using a 27-day real-world traffic dataset collected from a 2G/3G network in northern China, covering the trajectories of 4914 individuals between July 25 and August 20, 2015. The study aimed to examine the impact of trajectory regularity and to incorporate spatiotemporal information for next-location prediction. Experimental results showed that the classical MC model achieved lower prediction accuracy compared to the LeZi Update and Active LeZi models. To address this limitation, the authors proposed an improved MC model that integrates temporal features and pattern weighting without constructing complex prediction trees. The improved model achieved an accuracy improvement of approximately 6%, reaching performance levels comparable to the LeZi-based approaches while maintaining lower computational complexity. However, the study also revealed that when trajectory regularity is low, the performance gap between the improved and classical MC models narrows, and the temporal effect of the trajectory has little importance when regularity is low. Wang et al. (2018a) assessed the efficiency of models with regard to lower and higher

orders and presented a Kernel Variable Length Markov Chain (KVLMC) model to predict trajectories. Their study includes two datasets, such as Geolife and floating cars. The Geolife is a GPS pedestrian travel data, and it was collected from 182 users from April 2007 to August 2012 in the Microsoft Asian Research Institute project, and the second dataset is floating cars covering Fuzhou city and also surrounding areas in China. It was suggested that the lower-order models are not very accurate because they only rely on recent data, while higher-order models, though more precise, require much more time and suffer from space complexity. The outcomes indicate that the KVLMC model obtained good performance by effectively balancing accuracy and computational efficiency. The research conducted by (Yan et al., 2021) that focused on the heterogeneity of individual mobility behaviour, while existing solutions are based on the historical data of all users at an aggregated level. Therefore, the Weighted Markov Chain Model (WMCM) based on the user's classification was proposed, and their study utilized a real encrypted dataset of 5000 users of China Mobile, which includes variables such as mobile user identity number, recording time, location area code, service type, uplink, and downlink traffic. The classification criteria are separately developed for users considering the distinctive mobility pattern using the ML model. The findings suggest that the WMCM with an optimized weighting coefficient approach can improve the model's performance. Since MC and hybrid MC models assume fully observable mobility states, while HM models relax this assumption by incorporating hidden states. As highlighted by Sadeghian et al. (2024), HM models effectively capture sequential and cyclic mobility patterns, and their study explored the feasibility of predicting both the next location and the associated time of arrival. A detailed record of over four million GPS points was collected from 91 volunteer users in Berlanga, Sweden, between September 2019 and October 2020. The dataset includes variables such as coordinates, date, time, altitude, and speed. Also, it includes the recorded observations at regular intervals approximately every five seconds, which provides a high temporal resolution for subsequent analysis. Two schemes were employed in the HM model to predict human mobility. The first scheme treats Points of Interest (POIs) as latent variables and time slots as observed variables, while the second scheme considers the time slots as a latent variable and the POIs as an observed variable. POIs refer to clusters of locations, and they were identified using hierarchical density-based spatial clustering of application with noise. The outcome shows that the HM model achieves an overall highest accuracy of 82% as compared to the study of (Lv et al., 2016; Khoroshevsky and Lerner, 2016; Qiao

et al., 2018; Wang et al., 2020b). Besides, it was suggested that future research should incorporate contextual information such as socioeconomic factors and real-time traffic situations and also develop strategies to handle missing data to ensure high-quality data to improve accuracy further. The study of (Mo et al., 2021) incorporated underlying activity patterns, which are often overlooked in traditional models, to enhance the performance. An activity pattern refers to the habitual sequence and characteristics of activities that an individual performs over a period of time. It includes when, where, and how long a person engages in different activities such as working, staying at home, shopping, dining, or leisure. An activity-based Input-Output Hidden Markov model (IOHMM) was developed to simultaneously predict the time and location of the next trip, providing better behavioral interpretability. The model was validated using real transit smart card data from Hong Kong's mass transit railway system, spanning from July 2014 to March 2017. The results showed that the IOHMM achieved prediction performance comparable to advanced Long Short-Term Memory (LSTM) neural networks, while offering interpretable behavioral insights. However, the model's limitations include reliance on transit data only, which excludes trips made by other modes, and challenges in activity recognition when data is sparse. Ayumi and Nurhaida (2020) analyzed one individual trajectory data that was obtained from the Geolife project. The dataset contains 71 trajectories with 109607 location sample points from October 23, 2008, to December 15, 2008. In the beginning, the stay points were determined using time and distance thresholds, and then the Regions of Interest (ROIs) were identified by using the density-based clustering approach. ROIs and POIs are essentially similar, as both represent clusters of locations. So, the ROIs were considered as the states of interest in the HM model to predict the next location and reported that the performance of the model is reasonably good, with an accuracy of 76.9%. HM models are well-suited for time series analysis when the primary objective is to infer or classify hidden states of a system evolving over time and when the exact duration spent in each state is not of primary importance. In the HM model, state durations are implicitly assumed to follow a geometric distribution, reflecting a memoryless property in which the probability of transitioning to another state remains constant over time. This assumption simplifies modeling but can limit realism when state persistence varies. In contrast, Hidden Semi-Markov (HSM) models are more appropriate when the duration of time spent in each state is a key characteristic of the underlying process. HSM models explicitly model state sojourn times using flexible, non-geometric distributions, allowing them to capture

behaviors with variable or prolonged durations more accurately. As a result, they provide a more realistic representation of temporal dynamics in systems where state durations are structured or heterogeneous (Ruiz-Suarez et al., 2022). Baratchi et al. (2014) developed a robust approach for predicting mobility patterns from high-resolution location data, and their aim was to address the limitations of existing models in capturing temporal variability and complex movement behaviors. To this end, they proposed a Hierarchical Hidden Semi-Markov (HHSM) model, which extends the traditional HM model and HSM model by incorporating a hierarchical state structure and explicit duration modeling to better represent stay-points and transitions. The model was evaluated using two real-world datasets: the Geolife human mobility dataset, which includes two individual trajectories spanning 76 to 254 days, and the Capricorn animal movement dataset, covering 332 days. The results showed that the HHSM model outperformed other models in prediction accuracy, robustness to noise, and identification of stay-points and movement transitions. However, the proposed model is computationally more complex, requires careful parameter tuning, and may perform less effectively for highly irregular or weakly structured mobility patterns. The work of (Menz et al., 2018) based on GPS trajectory data collected over a weekly cycle, focusing not only on predicting future locations but also the likely departure and arrival times. The primary motivation was to address the complexity and limited accuracy of existing methods that fail to effectively integrate spatial and temporal data for accurate location prediction. To tackle this, the researchers developed a model combining a Markov model for location prediction and a probability density function for time prediction. Overall, the developed model performed well in predicting regular weekly patterns, but it has faced challenges with sporadic behaviors occurring outside weekly cycles. It was indicated that the quality issues of GPS data have affected accuracy, with the missing trajectory in the dataset causing incorrect departure and arrival times.

ML techniques have been widely adopted for next-location prediction because of their ability to learn patterns and relationships from mobility data. However, these models often require large and diverse datasets for effective training, and their predictions may still involve uncertainty despite the high data and computational costs (Garola et al., 2024). ML approaches used in mobility prediction are generally divided into supervised and unsupervised learning. In supervised learning, models are trained on labeled data to learn the relationship between mobility features and known outcomes, enabling the prediction of future locations or trips. In contrast, unsupervised learning does not rely on

labeled data and is mainly used to identify similar mobility behaviors by grouping users or locations with comparable movement patterns. Supervised learning methods typically involve classification and regression tasks. Classification is used to assign users or locations to predefined categories, supporting applications such as location-based services and user profiling, while regression models aim to predict continuous values, such as geographic coordinates or travel distance (Toch et al., 2019). Although some of studies rely on network-level, synthetic, or vehicle-based datasets rather than smartphone mobility data, and they share the common objective of predicting future locations or trajectories and are therefore relevant to mobility modeling. Therefore, the commonly ML algorithms applied to next-location prediction include decision trees, random forests, Support Vector Machine (SVM), and neural networks. Unsupervised methods, such as clustering, are widely used to discover mobility patterns by grouping individuals or locations with similar characteristics (Garola et al., 2024). Within this context, trajectory clustering has emerged as a prominent approach for analyzing GPS trajectory data. Recent research on trajectory clustering has explored a variety of approaches for analyzing GPS trajectory data. For example, clustering-based methods have been applied to taxi GPS trajectories to identify pick-up and drop-off locations (Saputra et al., 2023). Trajectory clustering has also been used to detect anomalous trajectories and abnormal movement patterns (Wang et al., 2018b). Moreover, prior studies have highlighted the importance of trajectory clustering for extracting trajectory similarity, identifying anomalies, and discovering meaningful patterns from large-scale trajectory data (Liu et al., 2020). In addition, a Collaborative Map Matching (CMM) approach has been proposed for low-sampling-rate GPS trajectories, where similar trajectories are grouped into clusters to improve matching accuracy. The CMM removes the outliers and then Density-Based Spatial Clustering of Applications with Noise (DBSCAN) algorithm is extended to cluster GPS trajectories into different groups, based on path similarity instead of geometric similarity (Bian et al., 2020). While unsupervised methods such as clustering are effective in uncovering latent mobility patterns without labeled data, supervised learning approaches leverage labeled mobility information to directly predict future user locations. Early supervised ML efforts focused on mobility prediction within communication networks, as (Chen et al., 2015) utilized a user's mobility dataset within Heterogeneous Wireless Networks (HWN) to improve mobile-aware applications by developing a method to predict user locations across network subnets. The proposed Multi-SVM-based Mobility Prediction (Multi-SVMMP) model uses differ-

ent multi-class Support Vector Machine (SVM) approaches to handle regular and random user movement patterns separately, resulting in more realistic and accurate predictions for HWN environments. To enhance energy efficiency, the model includes a target region feature, where predictions are triggered only when users enter specific areas, reducing energy consumption while maintaining high accuracy. This selective prediction conserves resources by avoiding unnecessary location tracking, ultimately lowering energy use without compromising performance. The Multi-SVMMP model outperformed other approaches in accuracy and adaptability to multiple users, showing promise in proactive service allocation. Future research could focus on predicting user temporal patterns, such as time spent in a location, potentially reducing location-sensing energy needs by up to 50%. To support next-generation cellular networks, Gebrie et al. (2019) generated a synthetic mobility dataset using the Self-similar Least Action Walk (SLAW) model, which simulates realistic human movement patterns. The dataset consisted of one-week trajectories for 84 mobile users with one-minute granularity. The research aimed to enhance 5G network performance by predicting user mobility patterns and supporting proactive network management to reduce mobile traffic and energy consumption in future cellular networks. Four predictors such as XGBoost, SVM, Deep Neural Networks (DNN), and Semi-Markov were evaluated using the cross-validation approach for accuracy and efficiency. XGBoost emerged as the most effective model, achieving a high prediction accuracy of 90.22% with minimal execution time, making it ideal for energy-saving applications in 5G. SVM and DNN also performed well, and their accuracies are 88.07% and 83.89% respectively, though DNN's training time was longer, and its output consistency depended on feature count. The Semi-Markov model, while lower in accuracy at 81.46%, provided fast execution suitable for scenarios needing rapid response times. Using real-world smartphone mobility data, (Zhao et al., 2020) applied numerous ML models to predict the future location and trajectory of users between the locations using separate and hybrid features. The boosting and bagging algorithms are used to improve the accuracy of individual models, while the stacking is applied to integrate multiple models. Their study utilized the Nokia mobile dataset, which incorporates diverse mobility traces collected from the smartphone Nokia N95 over a period from October 2009 to March 2011 in Switzerland. It was stated that overall, 180 volunteers participated and provided varied quality in data collection due to differences in device usage. The purpose was to evaluate how different features such as individually and in combination can influence the level of accuracy while accounting for

both homogeneous and heterogeneous patterns of user movement. Results indicated that the presented slacking algorithm consists of three individual models such as J48 decision trees, Bayes Network (BN), and multilayered perception with hybrid features achieved 83.37% accuracy for the location prediction, while the adoptive MC model obtained nearly 80% accuracy for homogeneous movements and 70% for heterogeneous movements for one user. Comparatively, the BN model with hybrid features obtained the highest accuracy of 84.76%, while it obtained 55.47% accuracy with separate spatial and temporal features. While this model outperforms others in predicting both homogeneous and heterogeneous movement. It was suggested that refining feature selection and integrating additional contextual data could further enhance its prediction accuracy. According to (Garola et al., 2024), the BN model captures variable dependencies over time, enabling the representation of dynamic systems and the modeling of complex temporal relationships. Hybrid supervised approaches as combining ML models with probabilistic mobility assumptions, have also been explored. Araújo et al. (2020) proposed the Ensemble Random Forest-Markov (ERFM) model to predict mobility using real user trajectories. The global check-in dataset is obtained from the United States region and includes 400 thousand users, consisting of about two million recorded locations from April 2012 to January 2014. It was stated that the next place that is going to be visited is only dependent on the current location becomes unsuitable, because it may not be enough to extract the patterns. Therefore, the trajectories were built based on different orders to extract different patterns. In their study, the two aspects were applied as individual and general. The individual aspect assumes that the user's mobility is only influenced by their behavior, while the general aspect considers that the behavior of users can be somehow correlated to each other. The general category was further divided into two types such as collective and hybrid. All the individual trajectories are merged into a collective trajectory set, and considered that all trajectories are the same for all users. Whereas in the hybrid sense, the user's mobility could be correlated with some users but not at all. In this way, similar routines were found for mobility prediction. Afterward, the individual random forests models with Markov property are trained considering that the next location only depends on the last visited location using the different types of trajectories, and their predictions were integrated into the outer layer. Initially, the author randomly selected 50% of the users and the ERFM model was verified using three types of trajectories, and suggested that the model with hybrid trajectories attained the highest accuracy of 51% and 58% for orders 2 and 3 as compared to other trajectory

types. Similarly, using the trajectories of all users with hybrid features, the model still achieved the highest accuracy of 71% and 83% for orders 2 and 3, respectively, as compared to models such as Adaboost, TEMMUS, random forest, SVM, and gradient boosting. Xue et al. (2021) suggested that by integrating geographical data with contextual and social behavior features, both the prediction accuracy and insights into visitor behavior can be enhanced. The research work by (Bieler et al., 2022) concluded that the incorporation of additional contextual information can improve the accuracy of the model in destination predictions. They aimed to enhance the accuracy of destination prediction using a public transport user dataset from February 23, 2021, to August 18, 2021. Therefore, the Multi-class Random Forest (MRF) model was proposed to predict user destinations and provided an initial prediction of a user's likely destination using contextual features such as time of travel, mode of transport, and historical travel distance to common destinations. Following the destination prediction by the MRF model, a MC model is applied to analyze the probability of user's next location based on the most common travel paths derived from historical data. It assessed transitions between predicted and observed destinations. It was highlighted that the proposed model achieved on average higher F1 score of approximately 81% using the parameter optimization technique and including additional contextual information. Also, they discovered the most contributing features as transport model and hour of departure for destination prediction. While the baseline model comparatively obtained approximately 40% of the F1 score without incorporating contextual information. Although the study by (Wang et al., 2021) focuses on traffic mobility, it shares similarities with human mobility prediction, as both use ML models to analyze and predict movement patterns. The highway dataset was generated using a traffic simulator to predict vehicle route choices, utilizing variables like vehicle identity, latitude, longitude, and route selection. The study employed an SVM model to improve traffic management and resource allocation in 5G networks. Similarly, ML models are increasingly being applied to human mobility, offering valuable insights for urban planning, resource distribution, and predicting individual movement patterns. Future work in both domains could explore more complex models and environments to further enhance prediction accuracy. A recent study by (Zhang et al., 2025) aimed to enhance the accuracy of mobility prediction by leveraging both current traveling features and historical activity chains to improve the understanding of spatiotemporal contexts. The study utilized a vehicle GPS trajectory dataset collected from 1916 travelers from January 1, 2019, to December 31, 2019, who commuted within

Shenzhen, China. Also, the road network dataset of the year 2018 of urban areas was employed for intersection extraction to represent Possibilities of Historical Travel Intentions (PHTI) and it included eight categories such as dining, medical facilities, residential areas, etc. The dataset is structured into current movement modes and historical activities, capturing variables such as travel distance, timestamps, and PHTI. The proposed methodology consists of three main parts: (1) current traveling feature extraction using spatiotemporal correlations of road intersections (STCorrelation) and real-time moving states (RMState) to capture real-time movement; (2) historical activity chain construction that concatenates spatiotemporal features from completed trajectories to learn long-term dependencies; and (3) model training and mobility prediction, which employs a destination prediction model that integrates both current and historical data. The model utilized latent Dirichlet allocation to represent travel intentions and a spatiotemporal scoring mechanism to enhance prediction accuracy. Results indicate that the proposed model achieved the highest prediction accuracy compared to four baseline models with a Mean Absolute Error (MAE) of 760.362 meters, Root Mean Square Error (RMSE) of 1438.317 meters and mean relative error of 20.86%. The study highlighted the importance of integrating both current and historical data to capture long-term spatiotemporal dependencies, which is crucial for improving mobility predictions. Although ML models are frequently criticized for their limited interpretability. This limitation primarily arises from their inherent black-box nature, which obscures the understanding of how individual input variables influence the target response (Lv et al., 2023). In response to this challenge, (Bénard et al., 2021b) proposed the Stable and Interpretable Rule Set (SIRUS) algorithm to address the lack of interpretability and stability in Random Forest models. SIRUS extracts a short and interpretable set of decision rules while remaining robust to data perturbations and achieving predictive accuracy comparable to RF. Later, (Bénard et al., 2021a) extended SIRUS to regression settings, indicating that it maintains a simple structure, high stability, and competitive performance, supported by both theoretical and empirical analyses. While SIRUS improves interpretability, it does not account for spatial dependence, which is common in geospatial applications. To address this limitation, (Saha et al., 2023) introduced RF-GLS, a Random Forest framework that incorporates spatial correlation through a Generalized Least Squares (GLS) loss function. By explicitly modeling spatial dependence, RF-GLS enables accurate estimation of nonlinear covariate effects and significantly outperforms traditional Random Forests in estimation and prediction, particularly under

strong spatial correlation. However, extending RF-GLS to binary outcomes presents challenges due to the reliance on the Gini impurity measure in classification trees. Saha et al. (2023) resolved this issue by showing that the GLS loss extends the Gini impurity measure via its equivalence to the ordinary least squares loss, thereby justifying the use of RF-GLS for binary dependent data. Building on generalized mixed-effects models with Gaussian process spatial effects, they proposed RF-GP, which integrates RF-GLS using a novel link-inversion strategy. The method is theoretically supported by consistency results and empirically shown to outperform competing approaches on both simulated and real-world datasets. Despite these advances, existing methods either focus on interpretability or spatial dependence, but rarely address both simultaneously. To bridge this gap, (Patelli et al., 2024) proposed S-SIRUS, a spatial extension of SIRUS designed to explain regression RF in the presence of spatial dependence. By extracting a compact and stable set of rules from spatial RF models, S-SIRUS improves predictive accuracy and yields shorter, more interpretable rule sets when spatial correlation is present.

ML approaches have achieved strong performance in mobility prediction tasks such as next-location prediction. However, researchers have increasingly turned to DL techniques. This shift is mainly driven by DL's ability to automatically learn complex patterns from heterogeneous and unstructured data (Luca et al., 2021). As a result, DL methods have gained considerable attention in individual mobility prediction. These models can effectively capture complex spatiotemporal relationships by employing advanced neural network architectures, including long-term and periodic movement patterns, often outperforming traditional statistical approaches (Ma and Zhang, 2022). Moreover, their capacity to integrate diverse data sources makes them well-suited for modeling the dynamic nature of human mobility (Sadeghian et al., 2024). However, the limited interpretability of models remains a key challenge, particularly for decision-making applications. The interpretability refers to the ability of humans to understand how a model reaches its conclusions. In DL-based mobility prediction, the decision-making process is obscured by complex layers of connection weights, making it difficult to comprehend. As a result, authorities and planners may hesitate to rely on these models for critical tasks, where understanding the "why" behind a decision is just as important as the prediction itself. In contrast, statistical models may offer lower prediction accuracy, but they are highly interpretable and flexible, allowing the integration of domain knowledge. For instance, the MC model is easily understood, as its predictions are based on transition probabilities, and it can incorporate external fac-

tors. Therefore, combining the strengths of both statistical and DL models could provide a balanced approach, where statistical models enhance interpretability without sacrificing predictive power (Ma and Zhang, 2022). The investigation of (Xu et al., 2022) is based on a DL model as LSTM and MC model to understand the movement predictability of urban tourists in South Korea. The study utilizes two nationwide mobile phone datasets such as CDRs and Mobile Signaling Data (MSD) recorded at regular intervals and consisting of anonymized location footprints from 192302 international tourists who visited South Korea between August 1, 2018, to August 15, 2018. The aim was to understand the relationship between movement predictability and other factors such as the length of stay of travelers and the activeness of travel patterns. The model's performance was evaluated using accuracy at ranks 1, 3, and 5, which indicates how often the correct individual's next location prediction appeared among the top 1, top 3, or top 5 predictions obtained by the model. The results showed that the MC model achieved accuracies of 39.7% at rank 1, 62.8% at rank 3, and 72.1% at rank 5. In comparison, the LSTM model outperformed it, reaching 46.3%, 68.5%, and 76.7% at ranks 1, 3, and 5, respectively. The findings highlight geographic variation in the two model's prediction accuracy and the ability to predict travelers' movements tend to differ across cities or destinations. As a result, a single model may not work well everywhere. Instead, building local models that consider the urban and behavioral contexts of tourists may further improve the quality of tourism services and experiences. Additionally, the study suggests that models based on overall travel patterns can still deliver good results in many urban settings. This may often share similar plans by tourists or making decisions in predictable ways. In such situation, simpler models like the MC, which don't rely heavily on complex or detailed data, can still be effective tools for planning and managing tourism services. Although, a limitation of these models is that they tend to reflect the average behavior of the majority, which can lead to overlook the needs of specific traveler groups. For instance, the accuracy of the MC model varied depending on the travelers' different backgrounds and behaviors. Feng et al. (2018) conducted a study to address complex sequential transitions, multi-level periodicity, and data sparsity challenges. They developed the DeepMove model, which is an attentional recurrent neural network that jointly models transition regularities and periodic patterns to enhance prediction accuracy and interpretability. They aimed to predict an individual next visiting location, and used three real-world datasets, such as Foursquare of New York from February 2010 to January 2011, mobile application data, and cellular network data

of China from November 2016 to January 2017. The results showed that the DeepMove model outperformed state-of-the-art models by over 10%, demonstrating better accuracy and interpretability. However, limitations include its focus on next-location prediction with fixed time intervals and the lack of semantic contextual information, which could be addressed in future work. Kong and Wu (2018) investigated the problem of next-location prediction and focused on a weak real-time prediction setting, where future locations are predicted minutes or hours ahead. The study identifies a key gap in existing work, where long-term visit prediction is well studied, but weak real-time mobility prediction has received limited attention. In addition, large-scale trajectory datasets are inherently sparse and irregular, which significantly limits the effectiveness of conventional deep learning models. Another challenge lies in the inability of the standard LSTM model to jointly capture spatiotemporal dependencies and users' historical visiting context, both of which are crucial for accurate mobility prediction. To address these challenges, the authors proposed a Spatial-Temporal LSTM (ST-LSTM) model that integrates spatial and temporal influences directly into the LSTM, thereby mitigating the impact of data sparsity and improving sequential modeling. Furthermore, a hierarchical extension as HST-LSTM, was introduced to incorporate contextual historical visit information, enabling the model to better learn long-term mobility patterns. The proposed models were evaluated using a large-scale real-world trajectory dataset derived from Baidu map logs collected in Beijing over one week from December 2 to 8, 2015. The dataset contained 311310 point-level trajectories, which were aggregated into 144320 area of interest paths by grouping consecutive location points within the same functional zones. The experimental results demonstrated that HST-LSTM consistently outperformed baseline models, including standard LSTM and ST-LSTM, confirming its effectiveness in addressing data sparsity and enhancing prediction accuracy in weak real-time mobility scenarios. Overall, this study highlights the importance of integrating spatiotemporal dynamics and historical contextual information to improve deep learning based mobility prediction models. Unlike approaches that rely on POIs information, Abideen et al. (2020) addresses next-destination prediction by explicitly modeling geographical factors, aiming to predict the exact longitude and latitude coordinates of a taxi's next destination. To overcome the limitations of traditional sequence-based models in capturing long-range spatial-temporal dependencies, the authors proposed a Deep Wide Spatial-Temporal Based Transformer Network (DWSTTN) based on an encoder-decoder transformer architecture. The model was evaluated on two large-scale real-world taxi tra-

jectory datasets from Porto and Manhattan. The Porto dataset spans from July 1, 2013, and June 30, 2014, while the Manhattan dataset spans from January 3, 2013, to January 3, 2014. The results indicated that DWSTTN significantly outperformed state-of-the-art baselines, including Autoregressive Integrated Moving Average (ARIMA), LSTM-based models, and Spatial-Temporal Graph Convolutional Networks (ST-GCN), by more effectively capturing long-range temporal dependencies and complex spatial relationships. Despite its effectiveness, the study acknowledges limitations such as the lack of integration of multimodal transportation data and real-world factors like traffic congestion and environmental conditions. Zhou et al. (2021) presented a Self-supervised Mobility Learning (SML) framework aimed at improving predictions of individuals' next locations and their trajectory classifications using a sparse and noisy human mobility dataset. By leveraging vast digital traces from sources like GPS trajectories, social media check-ins, and CDRs, the study focuses on modeling human mobility patterns and predicting future movements. The study utilizes Foursquare, Brightkite, and Gowalla datasets that include various POIs, user trajectories, and check-in records, and the observations were recorded irregularly as check-ins do not occur uniformly due to sporadic user engagement with mobile applications. The proposed framework incorporates several methods, including DL techniques like Gated Recurrent Units (GRUs), along with contrastive learning to enhance the representation of mobility patterns. The first step involves encoding user trajectory data into a compact hidden state using GRUs, followed by predicting future visits through contrastive mobility learning, which distinguishes real user check-ins from negative samples. The method maximizes the mutual information between historical mobility contexts and future check-in occurrences, providing richer representations. A comparative analysis shows that the SML model not only enhances the accuracy of next-location predictions but also demonstrates flexibility across Trajectory User Linking (TUL) tasks, indicating robust performance in differing contextual settings. TUL refers to the task of associating anonymous mobility trajectories with their corresponding users based on movement patterns and behaviors. The model's performance was evaluated based on how accurately it could predict the next location within the top recommended locations and measured by accuracy with ranks 1 and 5. On the Gowalla dataset, the SML-TUL model achieved 45.71% accuracy for the top 1 recommended location and 63.98% for the top 5. For the Brightkite dataset, it also led the way with 43.88% in the top 1 and 61.45% in the top 5. The best results came from the Foursquare dataset, where SML-TUL reached its highest accuracy of 57.23% in

the top 1 and 66.07% in the top 5 as compared to other models. Huang et al. (2024) presented the Causal and Spatial-constrained Long and Short-term Learner (CSLSL) model to predict the location's category and location. This model captured the decision logic of human travel specifically the causal relationships among time, activities, and destinations while ensuring spatial consistency in the predicted locations. The model's performance was compared with the baseline model using three different datasets such as New York City (NYC), Tokyo (TKY), and Dallas (DL). The NYC and TKY datasets were used from April 3, 2012, to February 16, 2013, while the DL was used from February 4, 2009, to October 22, 2010. The model's performance was evaluated using the recall metric at ranks 1, 5, and 10, similar to the approach used by (Zhou et al., 2021; Xu et al., 2022), with the difference being in the selected ranks. For predicting location categories, the CSLSL model achieved the highest accuracy of 66.1% at rank 5 on the NYC dataset, outperforming other models. On the TKY dataset, it reached 48.8%, 80.1%, and 87.5% accuracy at recall ranks 1, 5, and 10, respectively. Since the DL dataset doesn't include location category information, only location prediction was performed for this dataset. Similarly, for location prediction, the CSLSL model achieved the highest accuracies of 26.8% and 56.8% at recall ranks 1 and 5, respectively, on the NYC dataset. On the TKY dataset, it reached 24.0% of recall at rank 1 and 48.8% at rank 5. For the DL dataset, the model obtained an accuracy of 12.6% at rank 1. Additionally, the proposed model was also tested on the check-in dataset, but the improvement was limited due to the sparse nature of the dataset. Qin et al. (2024) focused on the issue of existing trip-based destination prediction models, which often miss crucial activity semantic and geographical association information, making it difficult to capture these characteristics due to the invisibility of activities. Therefore, they proposed a DeepAGS (Deep learning model with Activity, Geographic, and Sequential information) to dynamically and simultaneously capture activity, geographical, and temporal mobility patterns for predicting an individual's next trip destination. DeepAGS was validated using real-world smart card data from urban railways, as well as a synthetic dataset to explore its geographical semantic feature capture. DeepAGS demonstrated superior performance, outperforming the best benchmark, LSTM-Hi-Attn, by an average of 2.1% in Accuracy and 1.5% in Recall. However, traditional attention mechanisms used in the model are limited in capturing spatial mobility patterns compared to regular temporal patterns, particularly when predicting highly irregular travel in synthetic data scenarios. Overall, the DeepAGS model achieved an average actual accuracy of 70.99% and a recall of 71.85%.

It has been observed that the performance of mobility prediction models, whether statistical, machine learning, or deep learning, varies substantially depending on the characteristics of the underlying mobility data, such as GPS trajectories, check-in records, or smartphone data. Existing studies have identified several challenges that limit the applicability of these models to individual-level smartphone mobility data. One major limitation is the strong heterogeneity in individual mobility behavior. Each individual exhibits a distinct movement pattern, which can significantly affect predictive performance. However, many existing models are trained on aggregated historical data across all users, and they implicitly assume homogeneous mobility patterns. This aggregation masks individual-level differences and reduces the model's ability to capture personalized mobility behaviors present in smartphone data.

Another challenge is the low regularity and high variability of individual mobility patterns. Unlike aggregated flows, individual smartphone-based trajectories often exhibit irregular travel schedules, which makes it difficult for existing models to learn stable and repeatable patterns, thereby reducing prediction accuracy. In addition, incorporating both spatial and temporal dependencies remains challenging due to the sparsity of smartphone users' mobility data and the frequent presence of missing observations. Many studies rely on synthetic datasets, which do not fully reflect real-world smartphone mobility dynamics. As a result, models developed on such data may not generalize well to real urban environments. Despite the rapid growth of location-based services, smartphone mobility data remains unexplored in previous studies, partly due to privacy concerns and data accessibility issues. Therefore, existing models and methods are not fully adequate for the data considered in this thesis. This thesis addresses the aforementioned limitations by focusing on individual-level smartphone mobility data, explicitly accounting for behavioral heterogeneity and incorporating fine-grained spatiotemporal features to develop a more accurate and adaptable mobility prediction framework for diverse urban populations.

Chapter 3

Overview of Smartphone-Based Urban Mobility Dataset

3.1 Dataset Description and Structure

This thesis investigates the mobility patterns of four smartphone users over two months, from March 1 to April 29, 2023, whose movements were recorded entirely within Istanbul, Turkey. The individual-level urban mobility datasets are obtained from the Earthquake Network Project[1], and they include three primary variables such as latitude, longitude, and timestamp, capturing each user's location coordinates at irregular time intervals. The latitude and longitude and measured in degrees.

Although the aforementioned project provided a large number of smartphone users' trajectories, and this thesis focuses on four smartphone users' movements who traveled the highest total distance within the available data span. These users exhibit more heterogeneous movement patterns compared to other users in the dataset, reflecting greater variability in their spatial behavior. The smartphone users' trajectories are characterized as sparse. Specifically, the geographic coordinates are recorded at a relatively low sampling rate with respect to time, and the recording frequency varies from day to day. Furthermore, the smartphone user's coordinates are captured at irregular timestamps, resulting in nonuniform temporal gaps between consecutive timestamps. As a consequence of this temporal sparsity, the spatial distances between successive locations also vary considerably.

The Earthquake Network Project operates a global, and it is volunteer-based earthquake

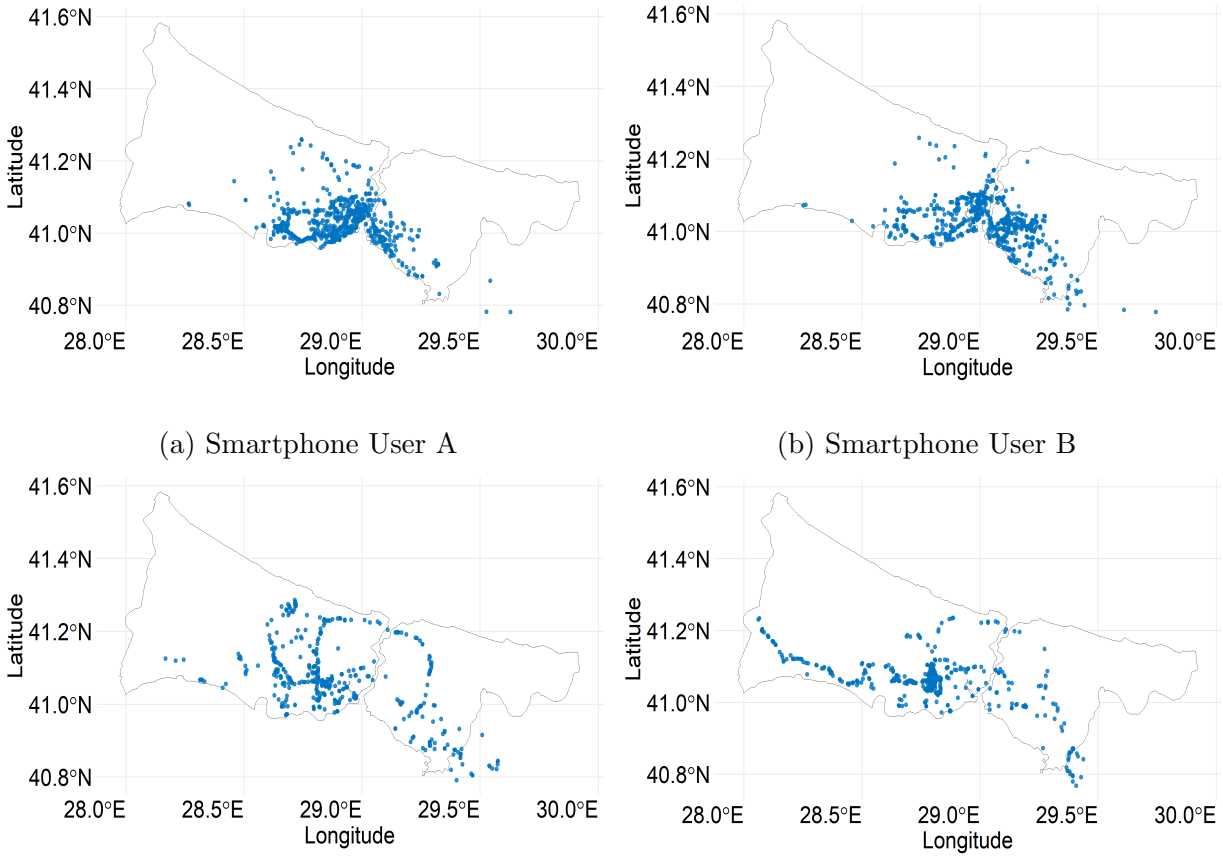
[1] www.sismo.app

early warning system that functions through networks of smartphones (Finazzi and Fassò, 2014). The system relies on the participation of individuals whose smartphones serve as sensing devices. Modern smartphones are equipped with geolocation capabilities, constant internet connectivity, and built-in accelerometric sensors that can detect ground vibrations. These sensors are capable of capturing the seismic waves generated during an earthquake. During an earthquake, two main kinds of seismic waves are produced. The first are primary waves, which move very quickly but usually cause little damage. The second are secondary waves, which arrive later, move more slowly, and are responsible for most of the destruction. Once primary waves are identified in real time, the system issues alerts to warn people of the imminent arrival of secondary waves, giving them valuable seconds to seek safety (Finazzi, 2016). Recent studies on the advancement of early earthquake detection within this project have been reported by (Finazzi et al., 2024; Finazzi and Massoda Tchoussi, 2024; Aiello et al., 2025).

3.2 Exploratory Data Analysis

The explanatory analysis focuses on the urban mobility dataset of four smartphone users over sixty days, which contains latitude, longitude, and timestamp for each location. It is important to note that the dataset does not contain contextual information about specific movement patterns, such as trips from home to work, work to shopping, or other purpose-driven routes. Instead, it simply records the coordinates and timing of each location, giving a basic view of the movement of users without the context of their destinations or intentions. As depicted in Figure 3.1, the mobility patterns of the four users cover the geographical area of Istanbul over sixty days. Each user exhibits distinctive and varying movement behaviors within the city. User A primarily travels within the city center, and it suggests a periodic pattern in these central areas, but also occasionally visits farther out from the center. Overall, it is creating a somewhat sparse movement pattern. Whereas user B moves widely across the city, showing a broader and more dispersed travel pattern than user A. User C typically follows a defined route but also makes frequent visits to distant locations, adding a scattered aspect to their movement. Finally, user D tends to stay less in concentrated areas and follows longer routes, indicating a less periodic travel pattern.

To assess the consistency of user's mobility dataset, the total number of recorded timestamps is counted for each date and visualized in Figure 3.2. It is crucial to identify



(a) Smartphone User A

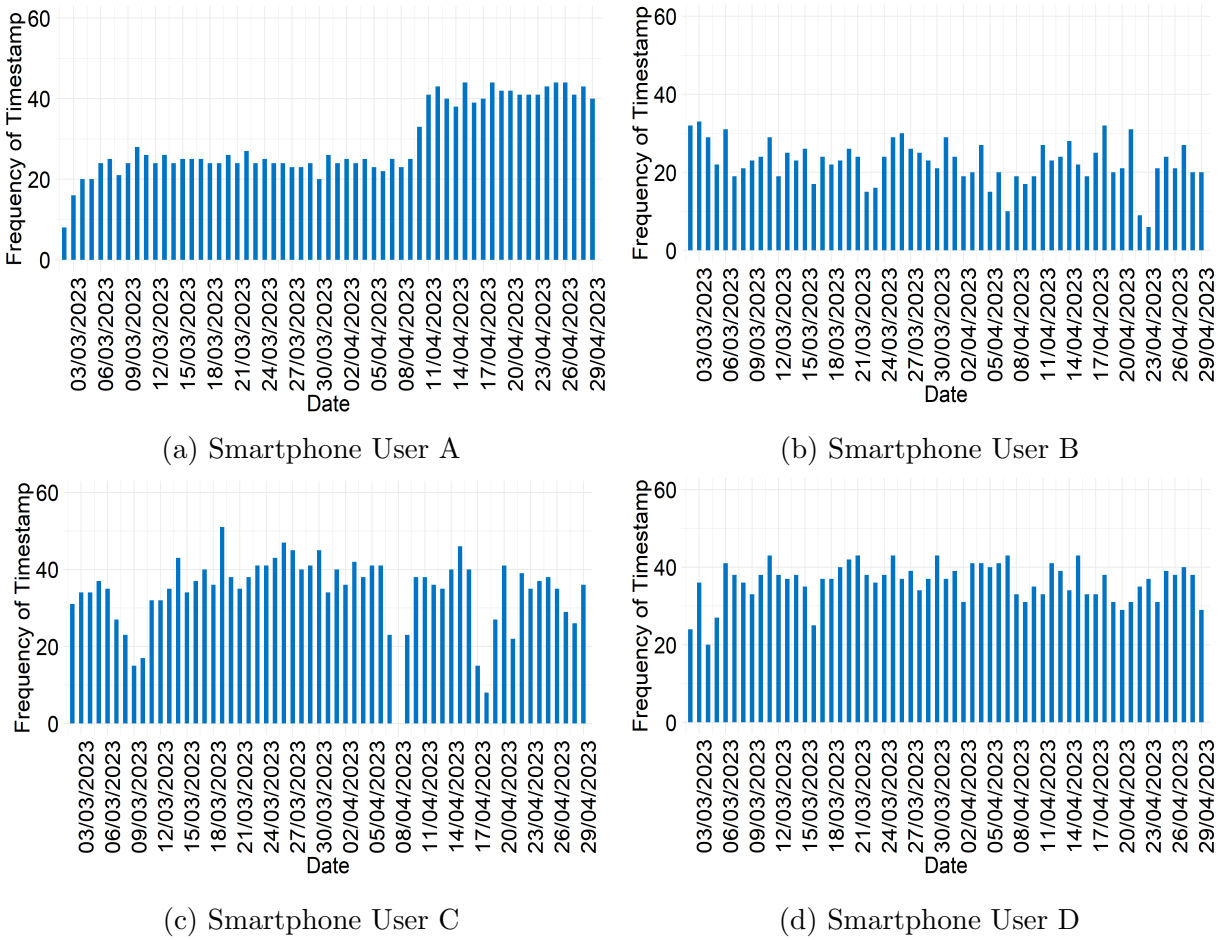
(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 3.1: The movement patterns of smartphone users in Istanbul, Turkey, from March 1 to April 29, 2023. The figure reflects spatial mobility trends and highlights areas of concentrated user activity. The latitude and longitude are described in degree units.

potential data gaps, and irregular recording patterns that could impact the reliability of the dataset. Understanding these variations helps ensure data quality and supports accurate interpretations of user activity and mobility trends. As shown in Figure 3.2, the distribution of recorded timestamps over the sixty days. The total number of recorded timestamps fluctuates from date to date, suggesting inconsistencies in data recording. For user A, the recorded timestamps range between 8 and 28 from March 3, 2023, to April 9, 2023. However, this number increases to between 32 and 44 from April 10, 2023, to April 29, 2023, indicating a potential change in data collection frequency. In the case of users B, C, and D, the recorded timestamps show no clear distribution, and it emphasizes possible data inconsistencies. Noticeably, there is a missing dataset on April 8, 2023, as no timestamps were recorded on that day for user C, which could be due to data recording issues or external factors such as the device being switched off or disconnected from the internet. It is confirmed by Figure 3.2 that the distribution of recorded timestamps varies significantly, suggesting that the consecutive timestamp intervals between users' locations



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

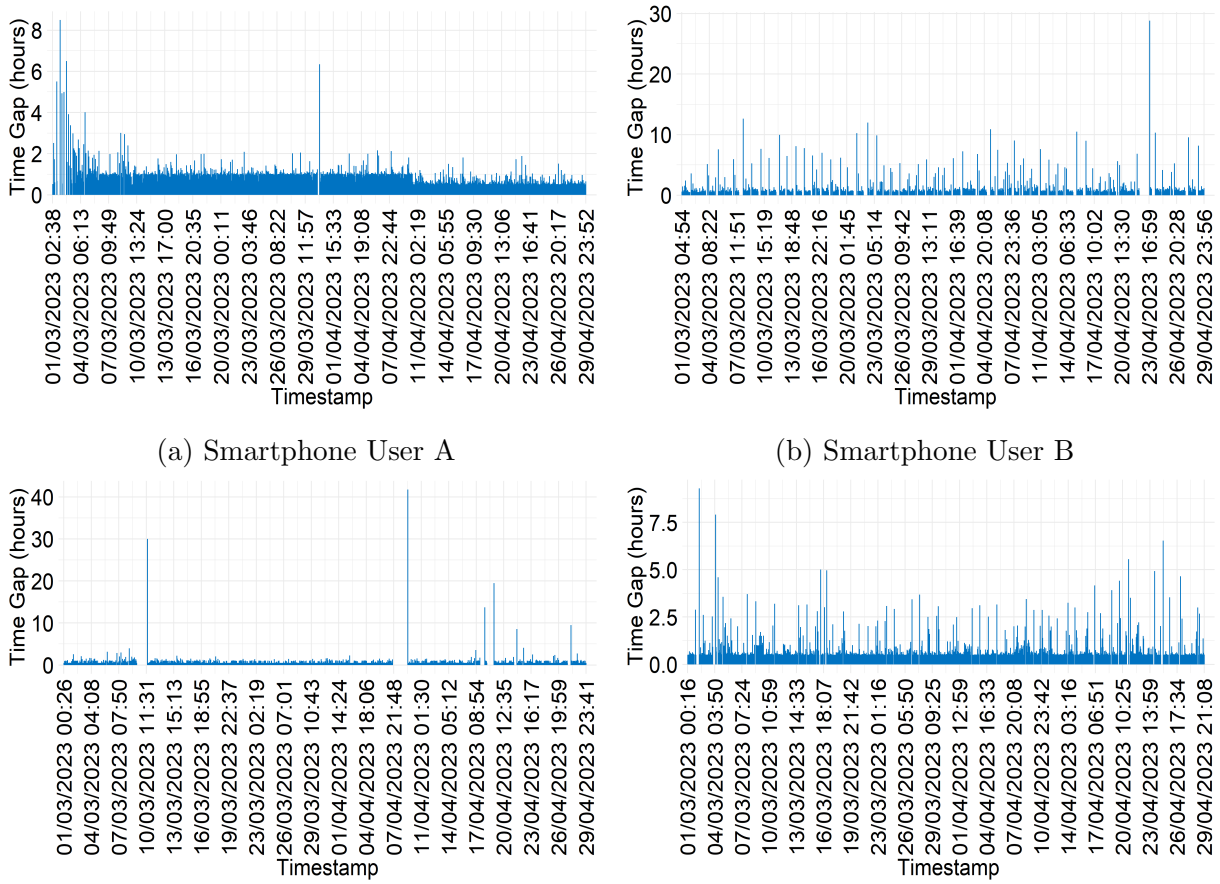
Figure 3.2: The frequency of recorded timestamps per day from March 1 to April 29, 2023. The figure illustrates fluctuations in mobility data collection across dates and users.

can vary. Therefore, the consecutive difference between the timestamps is computed in minutes, and then the median value is calculated to know the central time difference for all users. Additionally, it can be observed in Figure 3.1 that there are diverse movement patterns among users, making it essential to analyze both the total distance traveled over the study period and the distance between consecutive recorded locations. Understanding these metrics provides valuable insights into user mobility behavior and potential variations in travel patterns. Since the mobility datasets include coordinates in degrees, the distances between consecutive locations were computed in kilometers using the Haversine distance formula. The resulting step-wise distances were summarized using the median to characterize typical movement behavior, while the total travel distance was obtained by summing all consecutive distances over the full sequence of locations. The further details are described in the Table 3.1. The Table 3.1 summarizes the mobility statistics, including the median time gap between consecutive timestamps in minutes, the median distance gap between consecutive locations in kilometers, and the total traveled distance by users.

Table 3.1: The summary statistics is presented for smartphone users, including the median time gap between consecutive timestamps in minutes, the median distance gap between consecutive locations in kilometers, and the total distance traveled in kilometers from March 1 to April 29, 2023.

Smartphone User	Median Time Gap (minutes)	Median Distance Gap (kilometers)	Total Traveled Distance (kilometers)
A	42	0.3493	9659
B	39	1.2155	7035
C	30	0.0084	6635
D	31	0.0336	8406

User A has a median time gap of 42 minutes, indicating a longer time interval between their consecutive timestamps, and their distance gap is 0.3493 kilometers, indicating the consecutive coordinates are recorded on a median distance of 349 meters and traveled the farthest, covering 9659 kilometers over the sixty days. User B has a slightly shorter median time gap of 39 minutes, but a larger median distance between consecutive locations at 1.2155 km, and covering a total distance of 7035 kilometers. User C has the shortest median time gap at 30 minutes and the smallest median distance gap of just 0.0084 kilometers as approximately eight meters, with the lowest total distance traveled at 6635 kilometers. User D has a median minute gap of 31 minutes, and their median distance gap is 0.0336 kilometers, and total traveled distance of 8406 kilometers. Although the median provides insight into the central value of the observations, and it does not capture the complete picture, especially in the presence of irregular gaps. To better understand potential disruptions or missing data, the hourly gaps between consecutive timestamps were also calculated and are visually represented in Figure 3.3. As shown in Figure 3.3, the hourly gaps between consecutive timestamps vary noticeably for each user. User A exhibits several spikes where the gap exceeds 1 hour, with some surpassing 2 hours and a maximum gap of 8 hours. In comparison, user B shows a majority of spikes above 5 hours, including one spike reaching nearly 30 hours. User C has the highest consecutive hourly gaps, with one instance exceeding 40 hours and several others above 8 hours. User D shows multiple spikes over 2 hours, with three spikes exceeding 6 hours. Overall, these large gaps indicate periods of missing data in the user's mobility dataset. Hence, the Figure 3.3 shows that there are hourly gaps between consecutive timestamps. However, it does not provide sufficient detail about gaps measured in minutes. To address this, Figure 3.4 presents a histogram that is based on a logarithmic scale with limiting the x-axis



(a) Smartphone User A

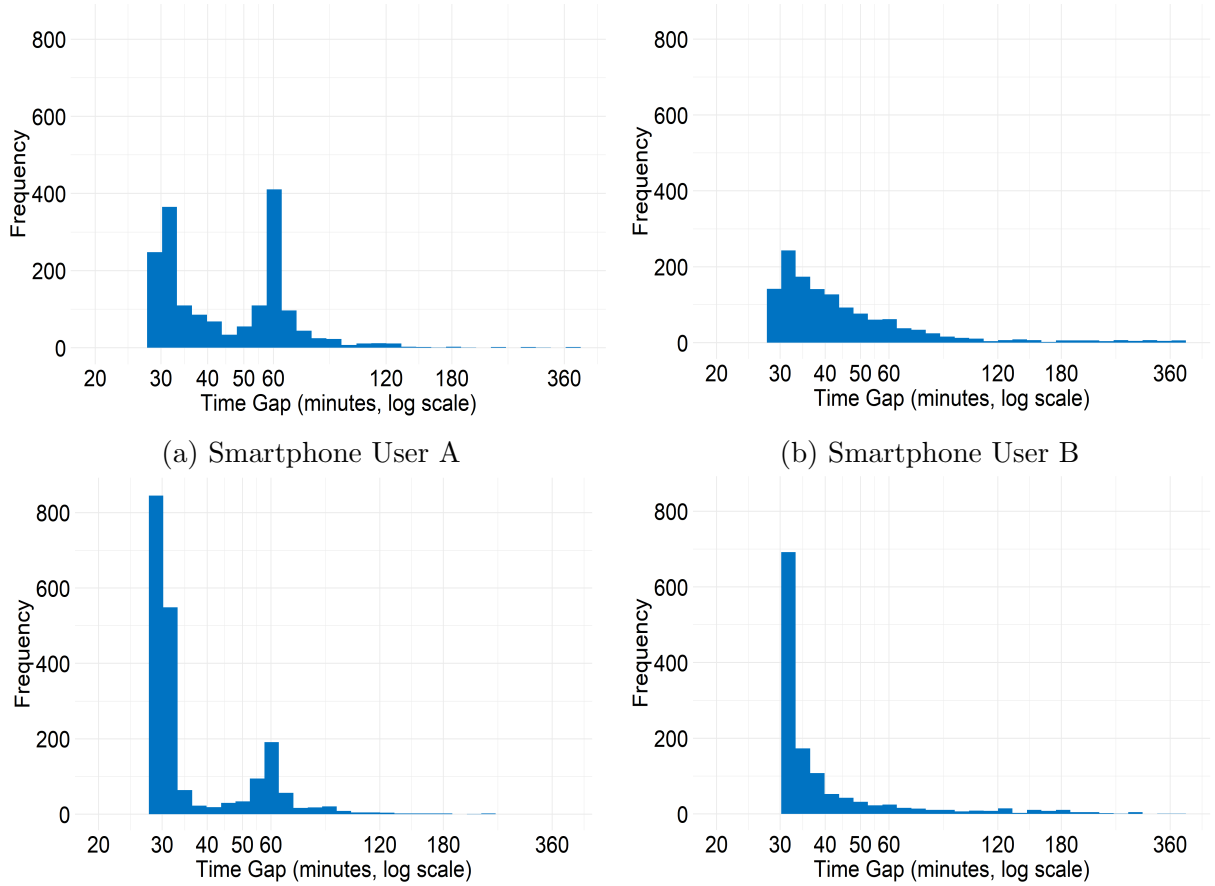
(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 3.3: The calculated hourly gaps between consecutive timestamps. The y-axis represents the time gap in hours, while the x-axis displays irregularly spaced timestamps from March 1 to April 29, 2023.

to 4 hours to focus on minute-level variations. The Figure 3.4 illustrations that the minimum gap between consecutive timestamps is consistently 30 minutes across all users. For user A, the most frequent gaps occur between 32 and 34 minutes, slightly more common than exactly 30 minutes. Then the frequency gradually decreases until around 34 minutes, rises again near 55 minutes, and shows another peak at 60 minutes. It indicates that the coordinates were often recorded at intervals of 30–34 minutes and exactly 60 minutes. For user B, the frequent gaps occur between 32 and 36 minutes, and then decrease gradually up to 120 minutes. Also, there are additional lower-frequency peaks between 120 and 360 minutes. For user C, the most common gap occurs at exactly 30 minutes, with an additional peak between 32 and 34 minutes, and another noticeable frequency at 60 minutes. For user D, the highest frequency is observed between 32 and 34 minutes, followed by a secondary peak at 30 minutes, and then the frequency decreases. Overall, these patterns highlight irregularities in how coordinates were recorded with respect to timestamps. To better understand the variations in user mobility patterns, the distance between consec-



(a) Smartphone User A

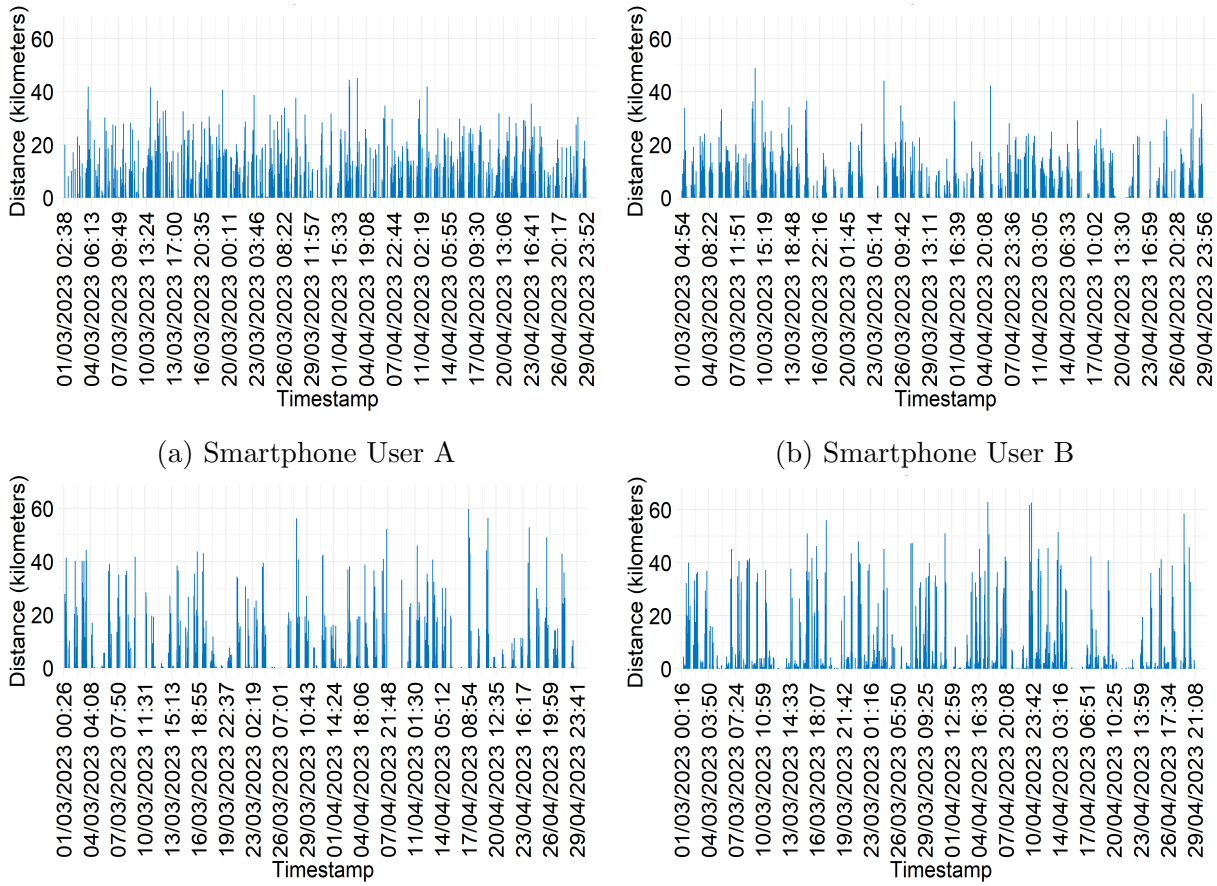
(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 3.4: The histogram of time gaps in minutes between consecutive timestamps for all smartphone users. The x-axis is limited to 4 hours to emphasize minute-level variations on a logarithmic scale and capture the distribution of short-duration gaps in the mobility dataset from March 1 to April 29, 2023.

utive locations is calculated and visually represented in Figure 3.5. This provides a clear illustration of how far each user travels between recorded locations and provides insights into movement behavior. From Figure 3.5, it is observed that user A generally travels distances greater than 5 kilometers, with some trips extending beyond 30 kilometers. This indicates that user A often undertakes relatively long journeys. User B follows a similar trend, with majority of trips exceeding 8 kilometers and a few also reaching beyond 30 kilometers. However, the occasional shorter movements are also present. Overall, it suggests that long-distance travel dominates, along with some variability in their travel behavior. User C displays greater variation, as the mostly trips exceed 10 kilometers but several extend up to 50 kilometers. However, the noticeable short-distance trips are also recorded, and they indicate that the user alternates between very long journeys and much shorter visits to nearby locations. Similarly, user D's travel patterns are dominated by trips over 10 kilometers, with some extending beyond 40 kilometers, while the shorter



(a) Smartphone User A

(b) Smartphone User B

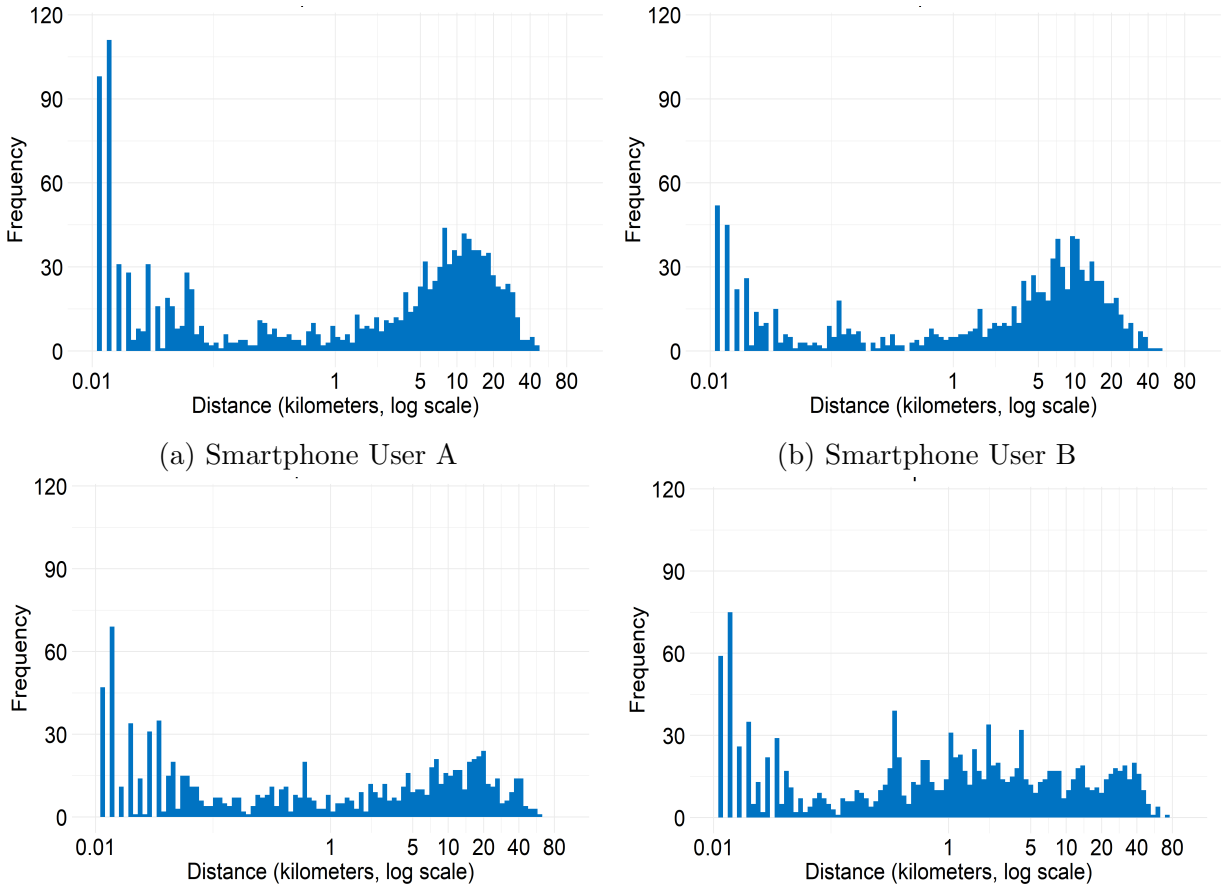
(c) Smartphone User C

(d) Smartphone User D

Figure 3.5: The measured distance gaps between consecutive locations are measured in kilometers. The x-axis shows irregular spaced timestamps from March 1 to April 29, 2023, while the y-axis represents the distance gap in kilometers between each recorded consecutive location.

movements below 10 kilometers are also evident. Overall, the variation in travel distances among users highlights the diverse movement patterns over timestamps. While some users covered relatively long distances in a single trip, others took a mix of both short and long journeys depending on their activities or destinations. This diversity reflects the inherent complexity of urban mobility.

A logarithmic scale is used to better visualize the wide range of distance gaps in the histogram. On a linear scale, the short distances are compressed near the axis in Figure 3.5, making it difficult to identify patterns in local movements. Therefore, the histogram of distance-gap with log scale is presented in Figure 3.6. As presented in Figure 3.6 that the distance gaps highlights distinct mobility patterns across the users. The distance gaps ranging from 0.01 to 1 kilometers represent periods where users are mostly stationary or engaged in very local movements. For user A, distance gaps of 1 to 5 kilometers occur less frequently, whereas movements in the range of 5 to 10 kilometers and especially 10



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 3.6: The histogram of distance gaps is shown for smartphone users. The x-axis represents the distance gap between consecutive locations in kilometers on a log scale, and the y-axis shows the frequency of occurrence. Very short gaps between 0.01 and 1 kilometers reflect stationary or local movements, while medium to long gaps highlight patterns of regular and occasional long-distance travel.

to 20 kilometers appear more often. Beyond 30 kilometers, the frequency drops sharply, indicating only occasional long-distance trips. A similar pattern is observed for user B, where medium-range distances of 5 to 20 kilometers dominate, while very short and very long trips are less common. User C also follows this general pattern, with fewer trips in the range of 1 to 5 kilometers, more frequent trips between 5 and 20 kilometers, and noticeable fluctuations in this range. Interestingly, this user shows an increase in frequency again around the 30 to 40 kilometer range, suggesting occasional longer-distance travel. For user D, the trend is somewhat different. The trips between 1 to 5 kilometers are the most frequent, while movements between 5 and 40 kilometers remain relatively steady with some fluctuations. However, the frequency drops noticeably after 40 kilometers, and it suggests that very long trips are uncommon.

Chapter 4

Spatiotemporal States Prediction of Smartphone Users

4.1 Contribution

This chapter presents a technique that incorporates both spatial and temporal information within a single model and introduces a feature engineering approach based on a Markov formulation. This approach is integrated into models such as SVC, RFC, and MLPC to predict the next spatiotemporal state. The spatiotemporal state is the combination of regions of interest as clusters and temporal fixed intervals of hours. Our work is motivated by Mohammed and Gündüç (2022), which integrates the transition probability matrix as a feature vector in the ML approach to capture local connectivity patterns of nodes for node classification and link prediction. The study by Bray and Han (2004) employed historical rainfall and runoff as inputs, with subsequent flow values as target variables to predict future discharge. Similarly, (Yu et al., 2006) applied lagged hydrological variables within a SVM framework for real-time flood forecasting. Additionally, Zhang et al. (2023) integrates the CA-Markov spatial-temporal modeling framework with a Random Forest algorithm to predict land-use transition probabilities. Their approach uses the Markov chain to estimate how land-use transitions occur over time, while Cellular Automata (CA) manage the spatial distribution of these changes. The aim was to predict the likelihood of transitions based on environmental and socioeconomic factors, resulting in more accurate and dynamic land-use change simulations.

Inspired by these studies, we adapt the idea of using transition probabilities as features

for sequential data, modeling the next spatiotemporal state as conditionally dependent on the current spatiotemporal state according to a Markov formulation, along with additional contextual features. However, we explicitly acknowledge that this is a modeling formulation rather than a strict property of human mobility patterns, which often exhibit dependencies extending beyond the immediately preceding state. In our method, we compute empirical transition probabilities between spatiotemporal states using historical information and incorporate them as features in the models to capture spatial and temporal dependencies within the sequence. Furthermore, the day of week is incorporated as a temporal feature to capture longer-term temporal dependencies. This feature has been widely utilized in previous studies within the mobility modeling literature. This strategy is considered to address the challenges posed by high irregularity and sparsity in the mobility dataset while effectively integrating spatial and temporal information in the model. Unlike conventional methods, the prediction of the next spatiotemporal state requires not only knowledge of the observed spatiotemporal states but also an understanding of the transition dynamics, leading to a more comprehensive and accurate representation of the mobility behavior of smartphone users. To the best of our knowledge, no prior research has employed models in this manner for spatiotemporal state prediction, making this a significant contribution to the field of smartphone-based individual mobility modeling.

4.2 Methodology

4.2.1 Random Forest Classification Model

The RFC model is intrinsically capable of dealing with multiclass problems, as originally discussed by (Díaz-Uriarte and Alvarez de Andrés, 2006). In this setting, let $Y \in R^p$ represents the input vector and $Z \in \{1, 2, ..., K\}$ denotes the class label, where k denotes the total number of possible classes. The aim is to build a classification function $h : R^p \to \{1, 2, ..., K\}$ that accurately maps input vectors to class labels. This function is assumed to be Borel measurable, ensuring that it is mathematically well-defined and can interact properly with probability distributions. Given the training sample $S_n = \{(y_1, z_1), ..., (y_n, z_n)\}$, the goal is to use the data to estimate a classifier h_n that approximately the true conditional distribution. According to Biau and Scornet (2016), a classifier h_n is said to be consistent if its misclassification probability converges to the

Bayes error as the number of training points tends to infinity. i.e.,

$$L(h_n) = P[h_n(Y) \neq Z] \underset{n\to\infty}{\to} L^*, \tag{4.1}$$

where the notation L^* is the error of the Bayes classifier, and it is optimal in the sense that it minimizes the probability of misclassification, and it assigns the most probable class to a given input vector Y based on the posterior probabilities.

$$h^*(y) = \arg \max_{k\in\{1,..,K\}} P[Z = k|Y = y]. \tag{4.2}$$

The RFC model consists of M decision trees $\{\nu_1, \nu_2, ..., \nu_M\}$, each trained on a different bootstrap sample of the training dataset. For a given input vector Y, each tree ν provides a class prediction. The final prediction is determined by taking a majority vote across multiple classification trees. Therefore, the multiclass RFC is described as follows,

$$h_{M,n}(y;\, \nu_1, ..., \nu_M, S_n) = \arg \max_{k\in\{1,...,K\}} \frac{1}{M} \sum_{j=1}^{M} 1[h_n(y;\, \nu_j, S_n) = k], \tag{4.3}$$

which means that the class most frequently predicted by the trees is selected as the final class. If a leaf node represents a region A, the trees predict the class that has the highest observed frequency among the resampled training points that fell into that region. Mathematically, this is expressed as:

$$h_n(y;\, \nu_j, S_n) = \arg \max_{k\in\{1,...,K\}} \sum_{i\in S_n^*(\nu)}^{M} 1[Y_i \in A,\, Z_i = k], \tag{4.4}$$

this equation determines which class has the most training samples inside the region A, and assigns that class label to all new inputs that also fall into this region. It's essentially a majority voting rule localized to each node. The splitting process of a region in a tree is handled by the Classification and Regression Trees (CART) algorithm. The CART algorithm searches for a split that best separates the classes by minimizing a node impurity measure. According to (Breiman et al., 2017), the Gini impurity can extend naturally to multiclass classification. It provides an intuitive way to assess the impurity of a region when making splits. To classify a data point that falls within a region A, so, there is a rule that assigns a randomly selected point from the set $\{Y_i \in A : (Y_i,\, Z_i) \in S_n\}$ to class k with probability $p_{k,n}(A)$, where $k \in \{1, 2, ..., K\}$ denotes one of the k possible classes.

The calculated probability of selected point belongs to the class k is $p_{k,n}(A)$. For a region A, the Gini impurity is calculated as follows,

$$G(A) = 1 - \sum_{k=1}^{K} p_{k,n}(A)^2. \tag{4.5}$$

Gini impurity measures the likelihood of misclassification if a class is randomly assigned according to the observed class distribution in a region. A lower Gini impurity means a purer region, while a higher value reflects a more mixed distribution of classes. When a region A is split into left and right subregions, and the decrease in impurity is given by:

$$L_{class,n(j,z)} = p_{class,n}(A) - \frac{N_n(A_L)}{N_n(A)} \times p_{class,n}(A_L) - \frac{N_n(A_R)}{N_n(A)} \times p_{class,n}(A_R), \tag{4.6}$$

where the notation $p_{class,n}(A)$ represents the empirical class distribution and $N_n(A)$ is the total number of data points in the region A. After the split, the notations $p_{class,n}(A_L)$ and $p_{class,n}(A_R)$ represent the empirical class distributions for the left and right subregions, respectively, while $N_n(A_L)$ and $N_n(A_R)$ are denoting the total number of data points or node size in each region. This generalization enables random forests to efficiently handle multiclass classification, making them well-suited for high-dimensional and complex datasets. RF models also offer convenient mechanisms for parameter tuning. One critical parameter is the number of trees M. According to Biau and Scornet (2016), the variance of a RF decreases as trees increases, making the predictions more stable. Breiman (2001) also observed that increasing the number of trees improves stability without causing overfitting, though the computational cost grows linearly with M. A practical balance is needed where M is large enough for stability but not so large that training becomes inefficient. A key advantage of the RF model is its ability to assess performance using out-of-bag error. Since each tree is trained on a bootstrap sample, about one-third of the data is left out and serves as a testing observation for that specific tree. By averaging the prediction errors across all trees, the model provides a reliable performance estimate. This makes it easy to tune parameters like the number of trees without needing a separate validation set (Kruppa et al., 2013).

4.2.2 Support Vector Classification Model

According to (Suykens, 2001), the given training set $\{(y_1, z_1), (y_2, z_2), ..., (y_n, z_n)\}$ with input data $Y \in R^p$ and it is corresponds to binary class labels $Z \in \{-1, +1\}$. The SVC

model formulation starts from the following assumption:

$$\begin{cases} \omega^T \varphi(y_k) + \beta \geq 1, & \text{if } z_i = +1, \\ \omega^T \varphi(y_k) + \beta \leq -1, & \text{if } z_i = -1. \end{cases} \tag{4.7}$$

where the notation ω is a weight vector and it defines the orientation of the decision boundary, $\varphi(y_k)$ indicates the transformation of input y_k, and β represents the bias, and it shifts the decision boundary. There are two boundary lines described in equation (4.7) that separate the two classes, and it's used to define the decision boundary. These two boundary lines are often known as margin boundaries, and they ensure that all correctly classified points remain on the appropriate side of the decision boundary. Any data point belonging to class +1 must be on or beyond the positive margin boundary, while any data point belonging to class -1 must be on or beyond the negative margin boundary. These conditions create a well-defined separation between classes, helping the model distinguish between them with minimal classification error. The goal of SVC is to maximize this margin, as a larger margin generally leads to better generalization and improved classification performance on unseen data. Finally, the equation (4.7) is equivalent to the following:

$$z_i[\omega^T \varphi(y_k) + \beta] \geq 1. \tag{4.8}$$

The nonlinear function $\varphi(.) : R^p \rightarrow R^{n_h}$ maps the input space to a higher dimensional feature space. The dimension n_h of this space is only defined in an implicit way and in some cases, it can even be infinite dimensional. The classifier function in the primal form is written as:

$$z(y) = sign[\omega^T \varphi(y) + \beta], \tag{4.9}$$

where $z(y)$ denotes the classification function evaluated at the input vector y and the sign function determine the class label between the -1 and +1. The notation $\omega^T \varphi(y)$ is the weighted sum of the transformed features and the bias term is represented by β. However, the decision function in Equation (4.8) is not directly evaluated because the high-dimensional transformation can be massive, potentially infinite, and computationally expensive. Therefore, the optimization problem can be defined as,

$$\underset{\omega,\beta,\xi}{Minimize}\, \varsigma(\omega, \xi) = \frac{1}{2}\omega^T \omega + C \sum_{i=1}^{n} \xi_i, \tag{4.10}$$

$$Subject\,to \begin{cases} z_i[\omega^T\varphi(y)+\beta] \geq 1-\xi_i, & i=1,...,n \\ \xi_i \geq 0, & i=1,...,n \end{cases} \tag{4.11}$$

The first term $\frac{1}{2}\omega^T\omega$ ensures that the margin between the classes is as large as possible for better generalization. While the notation C is the regularization parameter, which balances margin maximization and misclassification tolerance. A higher C forces fewer misclassifications and a lower one allow more misclassification. Furthermore, slack variables ξ_i provide flexibility by allowing some data points to fall inside the margin or even be classified. This helps the SVC handle real-world noise and overlapping classes, making it more robust when perfect separation is not possible. Therefore, the Lagrangian function can be used to transform the original primal problem which includes the complex constraints into a dual problem that is easy to solve. The Lagrangian formulation is defined as follows,

$$L(\omega,\beta,\xi;a,v) = \varsigma(\omega,\xi) - \sum_{i=1}^{n} a_i\{z_i[\omega^T\varphi(y)+\beta]-1+\xi_i\} - \sum_{i=1}^{n} v_i\xi_i, \tag{4.12}$$

$$\underset{a,v}{Maximize}\ \ \underset{\omega,\beta,\xi}{Minimize}\ L(\omega,\beta,\xi;a,v). \tag{4.13}$$

The Lagrangian function combines the objective function and constraints using the Lagrange multiplier a_i and v_i. The goal is to satisfy the saddle point property in the Lagrangian optimization by maximizing the function with respect to the Lagrange multipliers a_i and v_i. While minimizing it with respect to ω, β and ξ_i. To find the saddle point, take partial derivatives of L with respect to ω, β and ξ_i, and set them to zero as described as follows,

$$\begin{cases} \frac{\partial L}{\partial \omega} = 0 & \rightarrow \ \omega = \sum_{i=1}^{n} a_i z_i \varphi(y_i) \\ \frac{\partial L}{\partial \beta} = 0 & \rightarrow \ \sum_{i=1}^{n} a_i z_i = 0 \\ \frac{\partial L}{\partial \xi} = 0 & \rightarrow \ 0 \leq a_i \leq C,\ i=1,...,n \end{cases} \tag{4.14}$$

From equation (4.14), by substituting w into the Lagrangian, one obtains the dual problem, which is the quadric programming problem. Instead of working with the weight vector ω and transformed features $\varphi(y)$ directly, the optimization is rewritten in the Lagrange multiplier a_i. The function depends on only the kernel function $K(y_{i,}y_j)$ which computes inner products in a higher-dimensional space without explicitly transforming the data and

this makes the SVC model efficient, even for complex nonlinear problems.

$$\max_{a} \vartheta(a) = -\frac{1}{2}\sum_{i,j=1}^{n} z_i z_j K(y_i,y_j) a_i a_j + \sum_{i=1}^{n} a_j \tag{4.15}$$

such that,

$$\begin{cases} \sum_{i=1}^{n} a_i z_i = 0 \\ 0 \leq a_i \leq c, \ i = 1, ..., n \end{cases} \tag{4.16}$$

The kernel function satisfies the Mercer condition, and it ensures that a function can be used as a valid kernel in SVC. It states that the function is a valid kernel if and only if the kernel matrix is also called the Gram matrix and positive semi-definitive for any possible input data. The Mercer theorem guarantees that if $K(y_i, y_j)$ satisfies this condition, there exists some feature transformation $\varphi(y)$ such that,

$$K(y_i, y_j) = \varphi(y_i)^T \varphi(y_j). \tag{4.17}$$

Finally, in dual space the nonlinear SVC model becomes,

$$z(y) = sign[\sum_{i=1}^{n} a_i z_i K(y, y_i) + \beta]. \tag{4.18}$$

The non-zero Language multipliers a_i are called support values, and their corresponding data points are known as support vectors, and these are located at the decision boundary. Also, these data points contribute to the classifier model.

4.2.3 Multilayer Perceptron Classification Model

The MLPC model for classification consists of three primary layers such as the input layer, the hidden layer, and the output layer. Each layer is composed of interconnected neurons, where each neuron in one layer is connected to neurons in the next through weighted connections. The input layer introduces a feature vector $Y \in R^p$, which is passed to the hidden layer after being multiplied with respective weights. The neurons in the hidden layers sum up the weighted inputs to neurons and include the bias. In the hidden layer, the weighted sum is passed through an activation function, also known as the transfer function, which shifts the space in the nonlinearity of input data. For clarity and consistency, the term activation function is used throughout. According to (Agirre-Basurko et al., 2006),

the general output layer can be expressed as follows,

$$z_k^0 = f_k^o(\beta_k^o + \sum_{i=1}^{s} \omega_{ik}^o z_i^h), \tag{4.19}$$

where the superscript h represents the elements of the hidden layer, while the superscript o indicates the elements of the output layer. The output layer is defined by z_k^o, where f_k^o is a transfer function of neuron k, β_k^o signifies the bias of neuron k, ω_{ik}^o is the weight connecting the neuron i of the hidden layer with the neuron k of the output layer. Additionally, the notation z_i^h is the output of the neuron i of the hidden layer. The general structure of the MLP model with a hidden layer is outlined as follows,

$$z_k^0 = f_k^o(\beta_k^o + \sum_{i=1}^{s} \omega_{ik}^o f_i^h(\beta_i^h + \sum_{j=1}^{n} \omega_{ji}^h y_j)), \quad k = 1, ..., L \tag{4.20}$$

where, f_i^h represents the transfer function of neuron i in the hidden layer, while β_i^h is its bias. The weight ω_{ji}^h connects the neuron j of the input layer with the neuron i of the hidden layer and y_j is the input of neuron j of the output layer. The activation function may either be linear or nonlinear (Marchitan et al. 2010; Khayet et al. 2011). It plays an important role in introducing nonlinearity into the model. However, the sum of weighted information still is in its linear model. The nonlinearity of information in the model occurs when it passes through the activation function. One of the most widely used nonlinear transfer functions is the logarithmic sigmoid function defined as follows,

$$\log sigmoid(y) = \frac{1}{1 + e^{-y}}. \tag{4.21}$$

This activation function is bounded and differentiable, and it maps input values to a range between 0 and 1. The choice of the number of neurons in the hidden layer is critical, as it has a direct impact on the model's performance. According to (Sun et al. 2008), there is no standard rule to determine the minimum or maximum number of neurons in the hidden layer. Too few neurons can lead to underfitting, while too many can cause overfitting. The hidden layer output is then passed to the output layer, which consists of K neurons corresponding to K possible classes, and each neuron indicates the probability that the input belongs to its corresponding class. To ensure that the outputs of the final layer sum to one and can be interpreted as probabilities, the SoftMax function is typically used.

It transforms raw scores (logits) into a normalized probability distribution as defined as follows.

$$p_k = \frac{e^{z_k}}{\sum_{c=1}^{K} e^{z_c}}, \tag{4.22}$$

where, z_k is the raw output as logits, the term p_k represents the predicted probability for class k, and K indicates the total number of classes. To train the network, the categorical cross-entropy loss, also referred to as the negative log-likelihood is minimized. This loss function is mathematically expressed as follows:

$$E = -\sum_{p}\sum_{k} t_{pk} \log p_{pk}, \tag{4.23}$$

where the term t_{pk} denotes the true label, and the p_{pk} is the predicted probability for class k for the p^{th} sample. Since each training sample is associated with a single correct class, only the probability assigned to that class contributes to the cross-entropy loss calculation. In other words, the loss function focuses solely on the predicted probability for the true class, disregarding the probabilities for the other classes. This approach, known as SoftMax fitting, ensures that the network generates valid probability distributions across all possible classes. To ensure that the function f remains smooth, and avoids overfitting, one way is to limit the complexity of the model known as regularization. Where the objective function is modified by adding a penalty term to discourage over-complex solutions. The regularization term is expressed as follows,

$$E + \lambda C(f), \tag{4.24}$$

where E is the original loss function, $C(f)$ represents the penalty term and it measures the complexity of function f. Furthermore, the term λ is a parameter controlling the strength of regularization. A specific form of regularizing is known as weight decay, and it penalizes large weight values by incorporating the sum of squared weights into the penalty term. This approach helps in smoothing and learning process and it prevents overfitting. For the entropy-based models, the range of λ between 0.01 and 0.1 is often recommended (Venables and Ripley, 2013).

4.2.4 Markov Chain Model

Consider a discrete-time stochastic or random process with finite state space as $\{Z_n : n = 0, 1, 2, ...\}$, then $Z_n = i$ indicates that the object is in state of system i at time n. According to Elfeki and Dekking (2001); Liu (2010), if for any positive integer n, the following equality is true,

$$P(Z_{n+1} = z|Z_1 = z_1, ..., Z_n = z_n) = P(Z_{n+1} = z|Z_n = z_n). \tag{4.25}$$

It states that the process's future state depends only on the present state and not on the sequence of past states, and this property is known as the memoryless property. Therefore, this stochastic process is called a Markov Chain (MC). The foundation of the MC model is based on the construction of a transition probability matrix. Therefore, the transition probability from the state $Z_n = i$ to state $Z_{n+1} = j$ in one step is defined as,

$$P_{i,j} = P(Z_{n+1} = z_j|Z_n = z_i). \tag{4.26}$$

For the t multiple steps, the probability of moving from state $Z_n = i$ to state $Z_{n+t} = j$ is given by,

$$P_{i,j}^{(t)} = P(Z_{n+t} = z_j|Z_n = z_i). \tag{4.27}$$

Let $Z_{ij}^{(t)}$ is the total number of transitions from state$Z_n = i$ to state $Z_{n+1} = j$ through the step t. As a result, the transition probabilities for various time steps can be calculated using the following approach,

$$P_{ij}^{(t)} = \frac{Z_{ij}^{(t)}}{Z_i}, \quad i, j = 1, 2, ..., n, \tag{4.28}$$

where Z_i signifies the total number of objects in the state i, and n is the total number of states. A matrix composed of transition probability that represents the probabilities of transitioning among states is known as a transition probability matrix. The transition probability matrix can be categorized into a one-step transition probability matrix or a t-step transition probability matrix. The one-step transition matrix represents the probabilities of moving from one state to another in a single step and is arranged in a square

matrix format as shown below,

$$P^{(1)} = \begin{pmatrix} P_{1,1} & \cdots & P_{1,j} & \cdots & P_{1,n} \\ \vdots & \ddots & \vdots & & \vdots \\ P_{i,1} & \cdots & P_{i,j} & \cdots & P_{i,n} \\ \vdots & & \vdots & \ddots & \vdots \\ P_{n,1} & \cdots & P_{n,j} & \cdots & P_{n,n} \end{pmatrix}, \tag{4.29}$$

the one-step transition probability matrix is an $n \times n$, where each entry $P_{i,j}$ signifies the probability of transitioning from state i to state j. Thus, the probability of transitioning from state z_1 to $z_1, z_2, ..., z_n$ is given by P_{1n}, $j = 1, 2, ..., n$ in the first row and so on. The transition probabilities must meet two important conditions: first, all probabilities must be positive values (greater than or equal to zero), and second, the sum of all transition probabilities in each row must be equal to one, i.e,

$$\begin{cases} P_{ij} \geq 0, \\ \sum\limits_{j=1}^{n} P_{ij} = 1. \end{cases} \tag{4.30}$$

One can consider t-step transition, which means that the transitions from state i to state j take place in t-steps. The t-step transition probabilities are arranged in square matrix format, which is expressed as follows,

$$P^{(t)} = \begin{pmatrix} P_{1,1}^{(t)} & \cdots & P_{1,j}^{(t)} & \cdots & P_{1,n}^{(t)} \\ \vdots & & \vdots & & \vdots \\ P_{i,1}^{(t)} & \cdots & P_{i,j}^{(t)} & \cdots & P_{i,n}^{(t)} \\ \vdots & & \vdots & & \vdots \\ P_{n,1}^{(t)} & \cdots & P_{n,j}^{(t)} & \cdots & P_{n,n}^{(t)} \end{pmatrix}. \tag{4.31}$$

The t-step transition probability matrix can be derived from the one-step transition probability matrix by applying matrix multiplication, as expressed below,

$$P^{(t)} = P^t. \tag{4.32}$$

Under some conditions such as aperiodicity and irreducibility, the successive multiplication leads to identical rows $(\omega_1, \omega_2, ..., \omega_n)$. Therefore, the identical rows $\{\omega_j;\ j = 1, 2, ..., n\}$

are given by,

$$\lim_{N\to\infty} P_{i,j}^{(N)} = \omega_j, \tag{4.33}$$

and, it is called stationary probabilities. The ω_j, $j = 1, ..., n$ are no longer dependent on the initial state z_n. The stationary probabilities can be obtained by solving the following equations,

$$\sum_{j=1}^{n} \omega_i P_{i,j} = \omega_j, \quad j = 1, ..., n, \tag{4.34}$$

subject to the conditions,

$$\omega_j \geq 0 \quad and \quad \sum_{j=1}^{n} \omega_j = 1. \tag{4.35}$$

4.2.5 Gaussian Mixture Model

The Gaussian Mixture Model (GMM) is a probabilistic framework that represents the distribution of dataset as a weighted linear combination of multiple Gaussian probability density functions. According to (Scrucca et al., 2016), the probability density function of a GMM with K components is expressed as,

$$p(Z) = \sum_{k=1}^{K} \pi_k \, N\big(Z \mid \mu_k, \Sigma_k\big), \tag{4.36}$$

where K denotes the number of components or clusters in the mixture, π_k represents the mixing coefficient associated with the k^{th} component, and $N(Z|\mu_k, \Sigma_k)$ corresponds to a multivariate Gaussian distribution with mean vector μ_k and covariance matrix Σ_k. The mixing coefficients satisfy the conditions $\pi_k \geq 0$ and $\sum_{k=1}^{K} \pi_k = 1$. Each Gaussian component defines an ellipsoidal cluster, where the mean vector determines the centroid of the cluster, the covariance matrix governs its shape, orientation, and spread. The mixing coefficient reflects the relative contribution of the component to the overall distribution. The GMM uses the Expectation Maximization (EM) to estimate the parameters, which provides maximum likelihood estimates of the parameters (μ_k, Σ_k, π_k). The algorithm operates iteratively in two main steps. In the expectation step, the posterior probability of each observation belonging to each mixture component is computed using the current parameter estimates. Subsequently, in the maximization step, the model parameters are updated by maximizing the expected complete-data log-likelihood based on the posterior probabilities obtained in the expectation step. This iterative process continues until conver-

gence, typically when changes in the likelihood function fall below a predefined threshold. The resulting model provides a flexible and robust approach for clustering and density estimation, as it can represent complex, multimodal distributions that cannot be captured by simpler and other approaches, which assume spherical clusters of equal size. The GMM allows components to assume elliptical shapes with varying volumes, orientations, and densities, and thereby offers a more general and realistic representation of the underlying dataset structure. Further details can be found in (Scrucca et al., 2016).

4.2.6 Evaluation Metrics

The performance of classification models is typically assessed using metrics such as precision, recall, and F1 score, which provide a comprehensive measure of their effectiveness. The details of these metrics are described below.

Precision

Precision is the proportion of predicted classes that are actually correct. According to Sarker and Salah (2019), the formula of precision is described as follows,

$$Precision = \frac{True\,Positive}{True\,Positive + False\,Positive} \tag{4.37}$$

In classification tasks, the positive class refers to the category of interest that the model aims to identify, while the negative class includes all other categories. The true positives occur when the model correctly predicts an instance as positive, and the false positives happen when the model incorrectly classifies a negative instance as positive. The high precision value indicates that most of the positive predictions made by the classifier are correct. The low precision means the classifier makes many false positive predictions.

Recall

Recall captures all the true positives, ensuring that as few as possible are missed. Recall is the number of true positives divided by the total number of true positives, which is the sum of true positives and False Negatives (FN). The FN are those instances where the model fails to identify a positive case and incorrectly predicts it as negative. A high recall means the model is excellent at detecting positive cases, but it does not necessarily mean all its predictions are accurate. According to Sarker and Salah (2019), the formula of recall

is as follows,

$$\text{Re}call = \frac{True\,Positive}{True\,Positive + False\,Negative} \tag{4.38}$$

F1 Score

The F1 score is the harmonic mean of precision and recall, and it is designed to provide a single balanced metric when both are important. It considers both how accurate the positive predictions are as precision and how well the model captures all actual positives as recall. By using the harmonic mean, the f1 score ensures that if either precision or recall is low, the score reflects that imbalance. A high f1 score means the model performs well in both scenarios. According to (Sarker and Salah, 2019; Araújo et al., 2020), the f1 score is described as follows,

$$F_1\,Score = 2 \times \frac{\text{Re}call \times Precision}{\text{Re}call + Precision} \tag{4.39}$$

Accuracy

The accuracy is commonly used metric that measures how often a model's predictions are correct. It is computed as the ratio of correctly predicted observations to the total number of predicted observations. According to (Araújo et al., 2020), the formula of accuracy is described as follows,

$$Accuracy = \frac{Number\,of\,Correct\,Predicted}{Total\ \ Number\,of\,Predicted} \tag{4.40}$$

4.3 Results

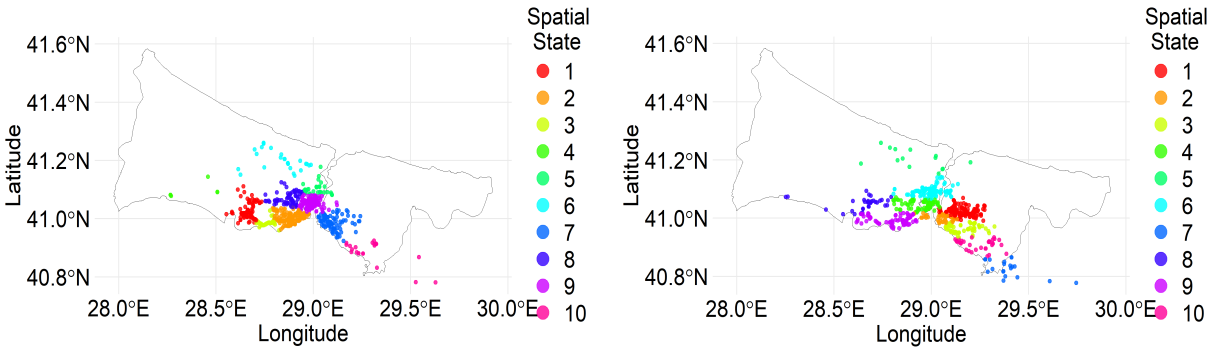
The chapter aims to develop a novel approach that integrates spatial and temporal information into the models to predict the spatiotemporal states of smartphone users. A spatiotemporal state represents a user's presence in a specific geographic region at a given time interval. This region can be referred to as a "cluster" or a "state" but for consistency, the term "state" throughout will be used in this chapter. Similarly, the word "user" will be used instead of smartphone user across the chapter. As shown in Figure 3.1, each user exhibits unique movement patterns across Istanbul, reflecting distinct individual behaviors. Consequently, it is important to identify the spatial regions that users visited or where they stayed during the sixty days. To achieve this, a model-based clustering approach as a GMM is implemented based solely on geographic coordinates. The GMM clusters locations according to their spatial proximity, thereby partitioning the city into distinct spatial states that correspond to areas most frequently visited by users. The total number of spatial states may vary among users, depending on the number of recorded coordinates and the distinctive nature of their movement patterns. The Figure 3.1 highlights the sparsity of their movements, as some visited locations are far from the city center. To ensure that those distant points are not treated as noise, therefore, the GMM is applied with the EEI covariance structure. The EEI structure assumes equal volume, equal shape, and axes aligned with the coordinate system. This means that all spatial states are modeled with a comparable spread and orientation, independent of how many points they contain. In this structure, latitude and longitude are treated as independent dimensions, so correlations between them are not explicitly captured. This configuration was chosen because it provides a balanced approach for identifying spatial states, as it allows for covering all visited locations uniformly, including those distant from the city center, are systematically groups them into meaningful spatial states.

The optimal number of spatial states is determined using the Bayesian Information Criterion (BIC), which provides a trade-off between model complexity and goodness of fit. The parameters of the GMM are then estimated using the EM algorithm, enabling the identification of meaningful spatial states within each user's movement patterns. The detailed results of the GMM are presented in Table 4.1. From Table 4.1, it can be noticed that user A has 10 optimal spatial states with 1748 sample points in the mobility dataset, and an average probability uncertainty of 0.1396, indicating confidence in the assignment of points

Table 4.1: The GMM's outcomes are described for smartphone users. Each sample point represents the total number of recorded locations with timestamps, and the average uncertainty measured in probability reflects the confidence in assigning the sample point to its corresponding optimal spatial state.

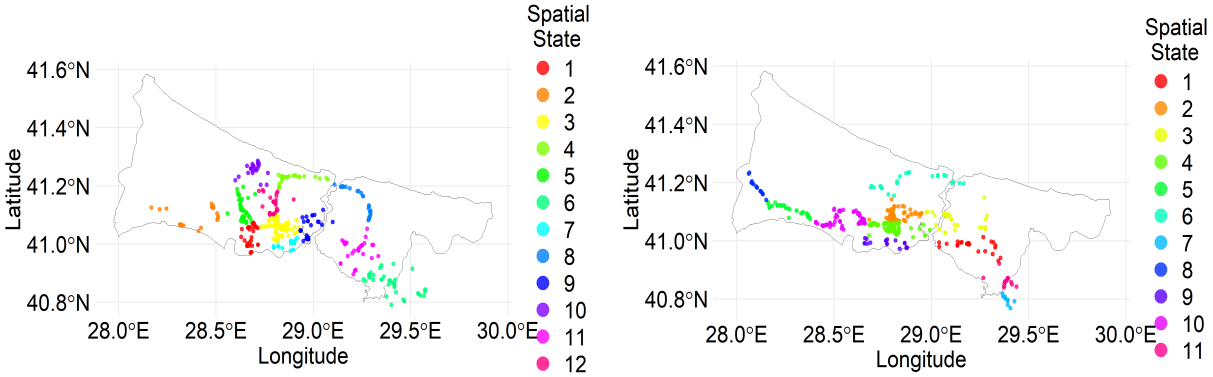
Smartphone User	Sample points	Spatial States	Average Uncertainty (Probability)
A	1748	10	0.1396
B	1368	10	0.1767
C	2067	12	0.0425
D	2178	11	0.0895

to the identified spatial states. Similarly, user B also has 10 spatial states, with the lowest number of observations of 1368 and an average uncertainty of 0.1767, suggesting reliable spatial state assignments. User C has 12 spatial states with the highest number of sample points as 2067 and the lowest average uncertainty of 0.0425, reflecting highly precise spatial state assignments. User D has 11 spatial states with 2178 sample points, and an average uncertainty of 0.0895, indicating a high level of confidence in the spatial state assignments. Overall, each user's movement is effectively segmented into spatial states with low uncertainty, confirming that the GMM accurately captures unique movement patterns. User C stands out with one additional spatial state and the lowest uncertainty, indicating particularly precise clustering. While the other users show slightly higher uncertainty, their movement patterns are still well-defined within the identified spatial states. The GMM-based spatial state segmentation for all users is illustrated in Figure 4.1. From Figure 4.1, each color represents a distinct spatial state, capturing different movement patterns of each user across Istanbul. Within each spatial state, the distance between coordinates is relatively small, meaning that the user's locations grouped in the same spatial state are close to each other. This state segmentation effectively groups the areas where users traveled or stayed, enabling clearer visualization of mobility patterns across different parts of the city. The Figure 4.2 illustrates the temporal closeness within each spatial state for all users. The dark navy blue indicates high counts, representing locations where users spent significant time or made repeated visits on specific timestamps, suggesting these spatial states are frequently revisited or habitual. The very light blue indicates no counts, meaning there is no temporal closeness at that timestamp; however, it still reflects that the spatial state was visited on those dates. The gaps in the heatmap correspond to periods with no movement between timestamps for certain spatial states. A clear pattern is observed in the first two spatial states of user A, the first state of user B, the third state of user C, and the first



(a) Smartphone User A

(b) Smartphone User B

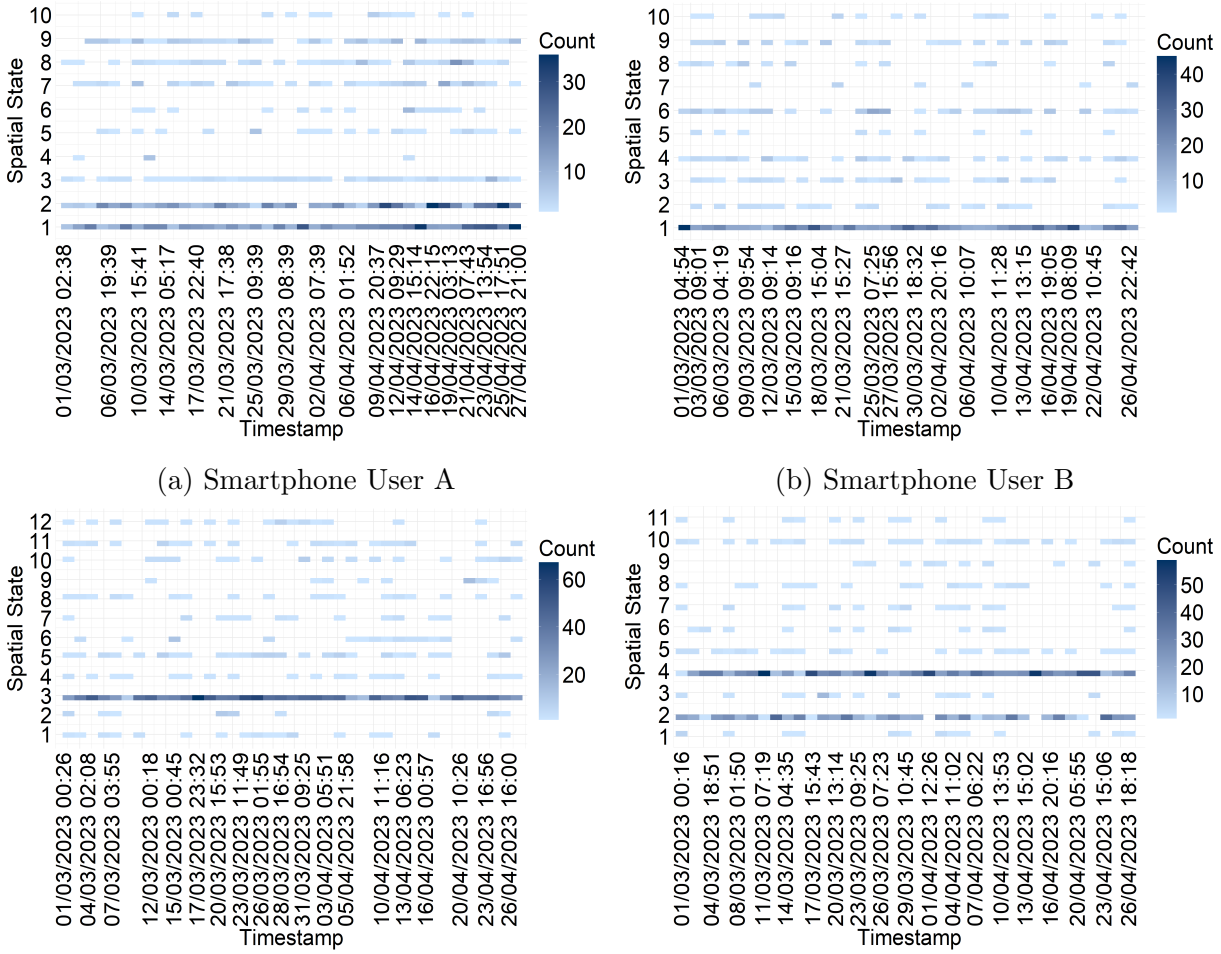


(c) Smartphone User C

(d) Smartphone User D

Figure 4.1: The spatial state segmentation based on the GMM. Each color represents a distinct spatial state, illustrating the user's movement across different areas of Istanbul from March 1 to April 29, 2023.

and fourth states of user D, indicating these are routine or regularly visited locations. The remaining spatial states shown in lighter shades of navy reflect less frequent visits or lower temporal closeness at specific timestamps. Eventually, Istanbul city is partitioned into spatial states based on the unique movement patterns for each user. Since each spatial state is a place where a user either visits or stays. To determine the temporal states, each day is divided into five time segments: night-time (00:00–06:00), morning commute and arrival (06:00–09:00), daytime (09:00–15:00), afternoon commute and office departure (15:00–18:00), and evening (18:00–24:00). Next, each user's timestamps are aligned with the defined temporal states and then combined with the spatial states, providing a comprehensive view of both where users were and when they were present at each location. For example, user A has 10 spatial states. When combined with the temporal states, this results in 47 spatiotemporal states, allowing a detailed analysis of where user A is likely to



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

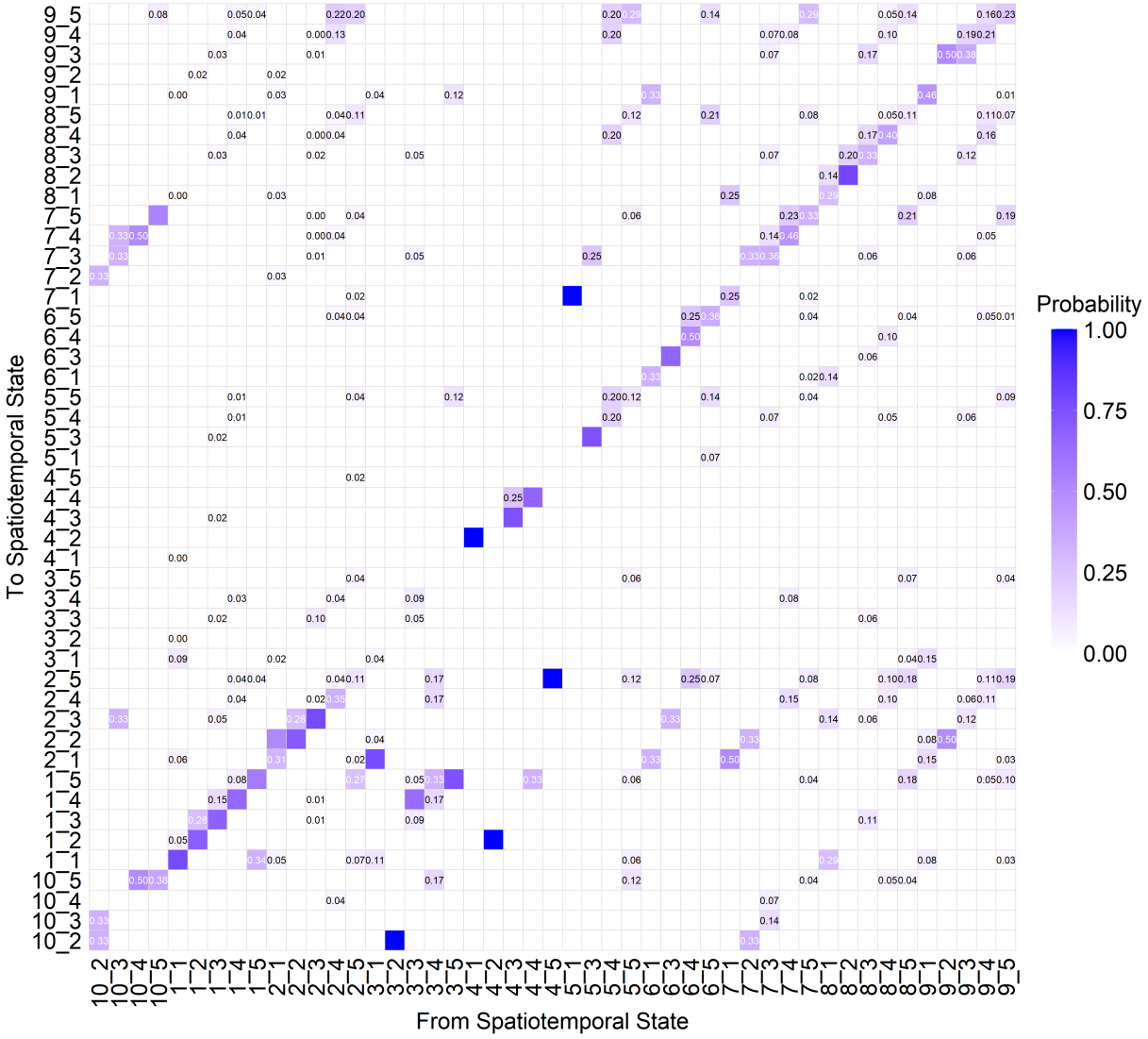
(d) Smartphone User D

Figure 4.2: The temporal distribution of spatial states for all smartphone users. The figure highlights the intensity of visits or stay within each spatial state over irregularly spaced timestamps from March 1 to April 29, 2023.

be during different parts of the day and providing insights into their movement patterns across these location-time pairings. The same process is applied to the other users and the user B has 43 spatiotemporal states, user C has 46, and user D has 47. This framework offers a comprehensive representation of each user's mobility, capturing not only the locations they visit but also the timing and duration of their presence. It provides a clear and systematic way to explore patterns of movement across both space and time. A brief overview of the construction and description of spatiotemporal states is provided in Table A.1 (Appendix A).

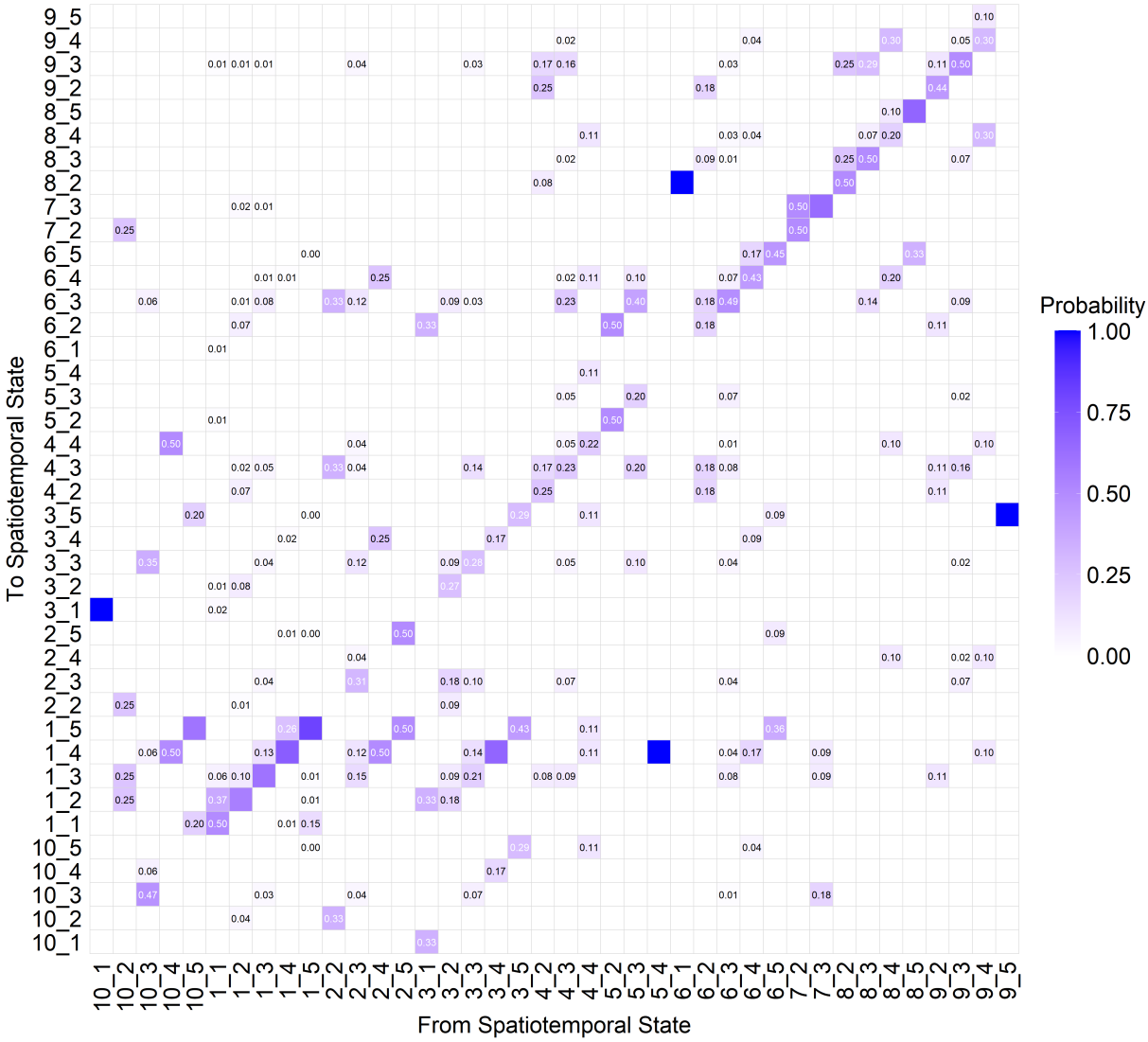
To predict the next spatiotemporal state of users, the mobility dataset is first divided into training and testing sets while preserving its chronological order. The training set consists of the first 80% of the spatiotemporal states, while the remaining 20% is reserved for testing. This setup ensures that the transition dynamics are learned from earlier se-

quences, and predictions are evaluated on future unseen behavior. From the training set, a count matrix is constructed to record how often transitions occur between different spatiotemporal states. For example, it captures the frequency of movements from one spatial location during a given time segment to another state in the next step. The count matrix is then normalized to form the transition probability matrix, which expresses the likelihood of moving from one spatiotemporal state to another. This normalization corresponds to maximum likelihood estimation, as the probabilities are directly inferred from the observed frequencies. The spatiotemporal transition matrices computed from the First-Order Markov Chain (MC-1) model are shown in the Figure 4.3 to understand overall transitions among different spatiotemporal states for each user.



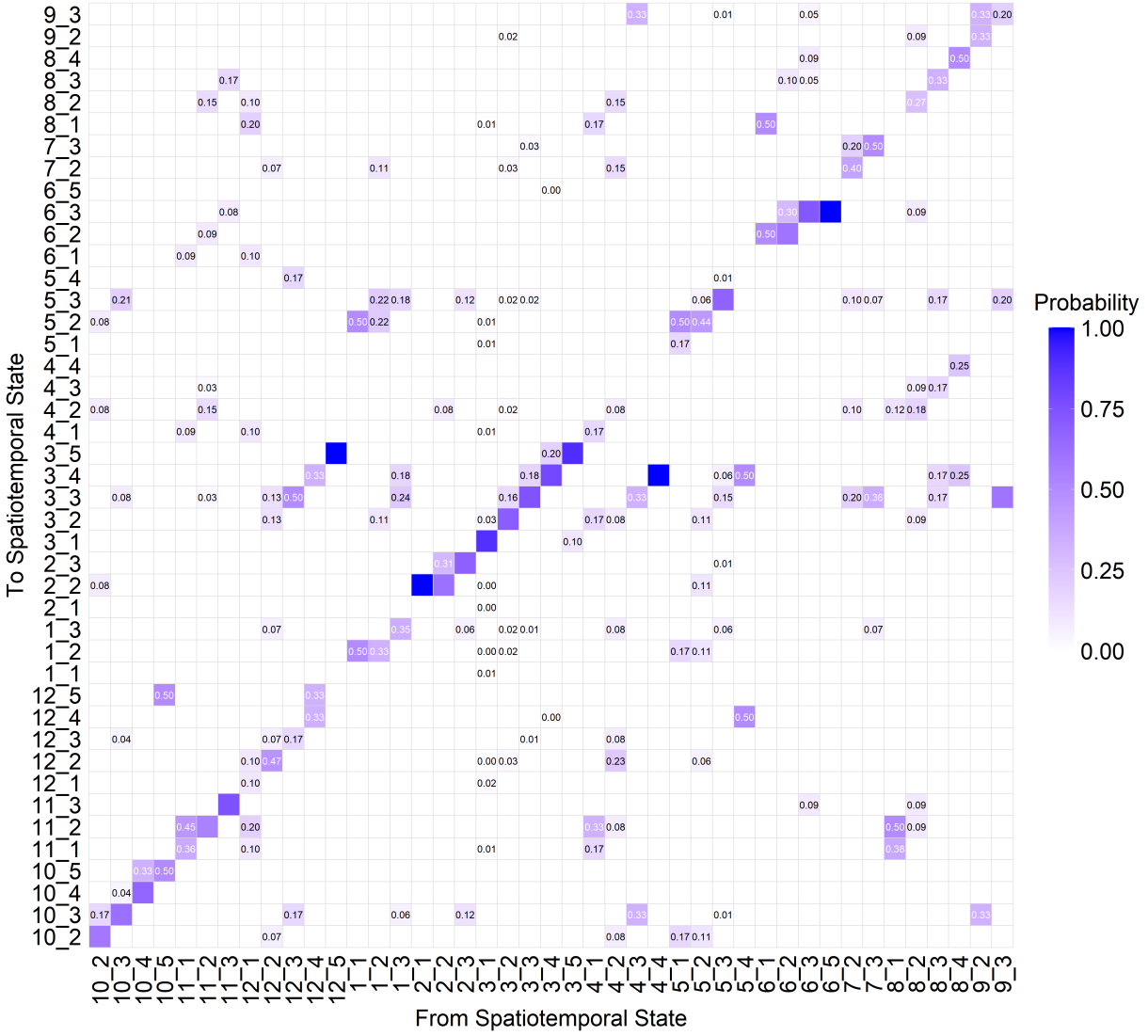
4.3 (a) Smartphone User A

In Figure 4.3, it can be discovered that some diagonal transitions, where the user remains



4.3 (b) Smartphone User B

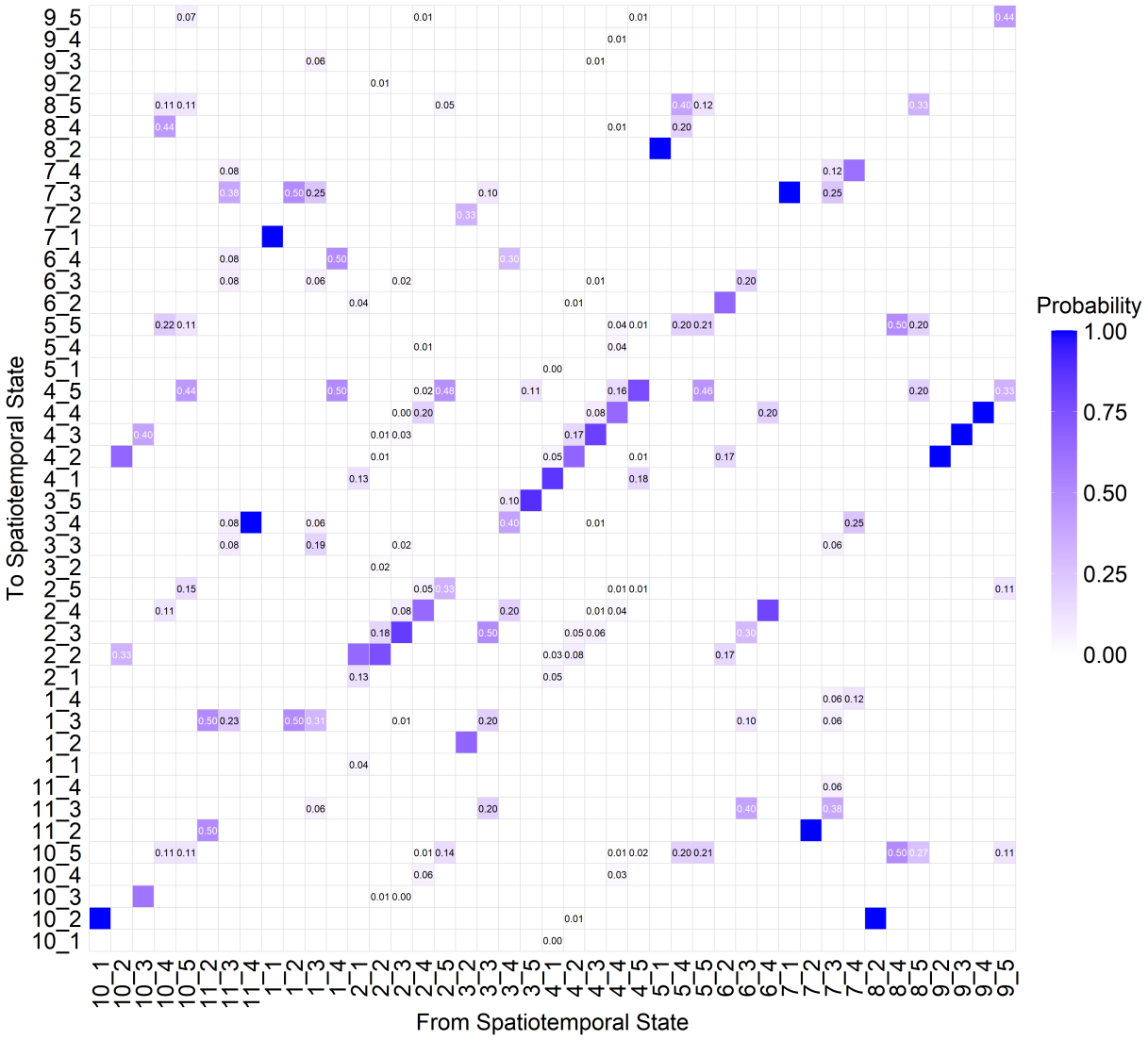
in the same spatiotemporal state and have the highest probability for all users. This indicates that users are likely to stay within specific locations of spatial state during particular time intervals, and it is reflecting strong preferences for routine behaviors. As shown in Figure 4.3(a) of user A, the probability of having 0.25 from spatiotemporal state 1_2 to 1_3 indicates that the user frequently remains in spatial state 1 from the early morning (06:00–09:00) to the daytime period (09:00-15:00) and suggesting stationary behavior or routine daytime activities such as being at the workplace. Similarly, the 0.25 probability from spatiotemporal state 1_5 to 1_1 shows that the user sometimes remains in spatial state 1 overnight (18:00–24:00 to 00:00–06:00), reflecting a consistent evening and night-time location, likely at home. The highly deterministic transitions further illustrate predictable behaviors as the probability of having 1 from spatiotemporal state 4_2 to 1_2 shows that whenever the user is in spatial state 4 during the morning, they always move to spatial



4.3 (c) Smartphone User C

state 1 and indicating a repeated commuting pattern or routine movement. Likewise, the probability of having 1 spatiotemporal state from 4_5 to 2_5 shows that in the evening interval (18:00–24:00), the user consistently moves from spatial state 4 to spatial state 2 and highlighting a regular pattern of visiting a particular location of spatial state 2 during this time period. The probability of having 1 from 3_2 to 10_2 also indicates a predictable morning movement, reinforcing the presence of routine mobility behaviors. Moreover, it is noted that there are several transitions with zero probability, indicating no movement between certain spatiotemporal states. This absence of transition indicates that some spatial states and time intervals are not part of the routine of user A, highlighting a behavior where the user has not visited or stayed within the specific time intervals. For other users, the interpretation criteria are the same. The diagonal transitions for each user, where they remain in the same spatiotemporal state, consistently show high probabilities,

confirming that users frequently stay in certain locations during specific time intervals. The high-probability transitions between different states reflect predictable routines, such as commuting, office arrival, or leaving patterns, while the lower-probability transitions capture variability or less frequent movements. For example, users may occasionally shift to new spatial states during office hours or evenings, indicating deviations from routine behavior. Overall, the heatmap provides a comprehensive view of where and when users are likely to be throughout the city.



4.3 (d) Smartphone User D

Figure 4.3: The heatmaps of spatiotemporal state transition probabilities were computed using the MC-1 (first-order Markov chain) model. The training set (which includes the first 80% of each smartphone user's spatiotemporal states of mobility dataset) is utilized to compute these probabilities. Each cell shows the likelihood of transitioning from one spatiotemporal state to another, with darker colors indicating higher probabilities.

Afterward, the two MC models are implemented, such as the First-Order Markov Chain

(MC-1) and the Second-Order Markov Chain (MC-2) models. In the MC-1 model, predictions of the next state rely only on the current spatiotemporal state. In contrast, the MC-2 incorporates the two most recent states, providing additional historical context to capture more complex patterns such as commuting routines or repeated sequences of visits. During the prediction phase, the procedure differs slightly between the MC-1 and MC-2 models. For the MC-1 model, the prediction process begins with a single initial spatiotemporal state taken from the testing set. This initial state is then used with the spatiotemporal transition probability matrix to predict the next spatiotemporal state. Each newly predicted spatiotemporal state is subsequently fed back into the model to predict the following state, and this process continues sequentially until the entire testing sequence is covered. In contrast, the MC-2 model requires two consecutive initial spatiotemporal states from the testing set to begin the prediction process. These two spatiotemporal states provide the context for estimating the spatiotemporal transition probability of the next spatiotemporal state. After the first prediction is made, the model shifts forward by discarding the oldest spatiotemporal state and using the most recent pair of spatiotemporal states to predict the next. This iterative process continues for the full length of the testing sequence. For fairness in evaluation, the first test state in MC-1 and the first two test states in MC-2 are excluded from the accuracy measure step, since they are used only as starting points for the prediction process. The predicted sequences are compared against the observed spatiotemporal states in the test set. The models' performances are assessed using standard metrics, including accuracy, precision, recall, and F1 score. The results of both MC-1 and MC-2 models for all users are summarized in Table 4.2. It can be observed

Table 4.2: The evaluation results of the first-order (MC-1) and second-order (MC-2) Markov chain models for each smartphone user. Both models are trained on the first 80% of the spatiotemporal states, and predictions are generated for the remaining 20% of the testing set. The predicted spatiotemporal states are compared against the observed spatiotemporal states in the testing set to assess model performance.

Model	Smartphone User	Accuracy	F1 Score	Precision	Recall
MC-1	A	0.0343	0.0611	0.0315	1.0000
	B	0.0803	0.1429	0.0769	1.0000
	C	0.1256	0.2198	0.1235	1.0000
	D	0.1812	0.3041	0.1793	1.0000
MC-2	A	0.1829	0.3024	0.1782	1.0000
	B	0.2737	0.4232	0.2684	1.0000
	C	0.2464	0.3906	0.2427	1.0000
	D	0.1743	0.2913	0.1705	1.0000

from the Table 4.2 that the prediction performance of MC-1 is relatively low across all

users. The accuracy ranges from 3.4% for user A to 18.1% for user D, and it is indicating that the model correctly predicts the observed spatiotemporal state in the testing set only a small fraction of the time. The F1 scores are also low, and they reflect the combined effect of low precision and the model's ability to match predicted spatiotemporal states to the observed spatiotemporal states. The precision values are low, which shows that many predicted spatiotemporal states do not correspond exactly to the observed spatiotemporal states in the testing set. In contrast, the recall values are higher for all users, meaning that every spatiotemporal state in the testing set appears at least once in the predictions. This high recall indicates that the model covers all possible spatiotemporal states, but it does not always predict the correct spatiotemporal state at the correct position in the sequence, which explains the lower values of precision and F1 scores. Comparatively, the MC-2 model shows clear improvement for user A, B, and C. By considering the previous two observed spatiotemporal states, the model identifies sequential patterns more effectively. The accuracy improves substantially from 3.4% to 18.3% for user A and from 8.0% to 27.4% for user B and it is indicating a higher proportion of predicted spatiotemporal states match the observed spatiotemporal states in the testing set. The F1 scores and precision values also increase and showing that the model more reliably predicts observed transitions. The recall remains high as 1.0 for all users and reflecting that all observed spatiotemporal states are included in the predictions. Interestingly, the user D shows a slight decrease in accuracy in MC-2 compared to MC-1 and suggesting that movement depends primarily on the previous spatiotemporal state rather than a longer sequence, so including an additional previous state does not necessarily improve predictive performance.

The chapter introduces a feature engineering approach based on a Markov formulation for spatiotemporal state prediction by integrating it into ML models such as Support Vector Classifier (SVC), Random Forest Classifier (RFC), and Multilayer Perceptron Classifier (MLPC). According to the Markov formulation, the next spatiotemporal state depends only on the most recently observed spatiotemporal state and the transition probabilities between states.

The initial step is to create the next spatiotemporal states with the help of observed spatiotemporal states. As described earlier, spatial states are identified using the GMM, while temporal states are defined as fixed hourly time intervals within a day. By combining these two components, each recorded observation of user mobility is represented as a spatiotemporal state, which captures both the location (spatial state) and the time of day (temporal

state) associated with that visit. The user's mobility dataset provides user locations with their corresponding timestamps in chronological order. Based on this ordering, the dataset is transformed into sequences of observed spatiotemporal states that reflect a user's actual movement behavior. The next spatiotemporal states are derived directly from the chronological order of the observed sequences. Each observed spatiotemporal state represents a user's presence in a specific spatial state during a defined temporal interval, and the subsequent entry in the sequence naturally defines the next spatiotemporal state. Next, the spatiotemporal state dataset is divided into training and testing sets while strictly preserving the temporal order of user movements. The first 80% of the sequences representing the earlier portion of the dataset is considered as the training set. The remaining 20%, corresponding to the later portion of the dataset, is reserved as the testing set for the evaluation. Using the training set, two key features were derived, such as the spatiotemporal state transition probabilities and the corresponding day of the week associated with each observed spatiotemporal state. The transition probabilities were derived empirically using only the first 80% of the data in chronological order, ensuring that the model is trained solely on past observations. A count matrix was first constructed to record how often transitions occur from one spatiotemporal state to another within the training set. This matrix was then normalized so that each row sums to one, producing a transition probability matrix that reflects the likelihood of moving from one spatiotemporal state to another. The general model formulation and feature construction are detailed in Appendix A.1.

First, the RFC model was trained on a training set, where the model parameters were carefully tuned to achieve the best performance. The RFC model operates by constructing an ensemble of decision trees, with each tree trained on a unique subset of the training set. These subsets are generated through a process called bootstrap sampling, where instances are randomly selected from the training set with replacement, meaning some samples can appear multiple times in a subset. This process increases model robustness by reducing variance and preventing from overfitting. To identify the optimal combination of parameters, a grid search approach was implemented in conjunction with rolling-window head cross-validation. In this procedure, the original training set is split based on the temporal sequence, ensuring that the chronological order of observations is preserved. For each fold, the first 80% of the data is used to train the model, and the subsequent 4% of the sequence is used for validation. After evaluation, the window is shifted forward by 4%, leaving out the first 4%, and the next 80% of the remaining data is used as the training set, with

the following 4% used for validation. This process is repeated across five folds, allowing each part of the dataset to serve as a validation segment while maintaining the temporal order. By systematically evaluating all parameter combinations within this rolling-window framework, the approach ensures that the selected parameters are robust and optimized for predicting future spatiotemporal states. The RFC model has three parameters such as "ntrees" is the number of trees, "mtry" is the number of features considered at each split, and the node size is the minimum number of samples required to form a terminal node. Specifically, the number of trees, such as 200, 350, and 500, are randomly selected to determine how the performance of the model varies. Generally, the highest number of trees leads to improved stability in the model. However, it increases computation time, so finding the optimal number is trees essential for performance improvement. For the parameter "mtry", the different values such as 1, 2, and 3 are tested, which control the number of features randomly chosen to evaluate at each split in a decision tree. A lower value introduces more randomness and enhances generalization, while a higher value makes each tree more similar and can improve performance when features are highly relevant. The parameter node size is also tested with different values of 1, 5, and 10. A smaller node size allows trees to grow deeper and capture finer details, but it can also increase the risk of overfitting. Whereas the larger node size produces simpler trees and might yield better generalization. The grid search approach presenting the optimal parameter values for each tree for all users are summarized in Table 4.3. The Table 4.3 shows that unique parameter combinations for the trees yield varying levels of accuracy across users. To select the

Table 4.3: The results of grid search approach implemented with rolling window cross-validation to identify the optimal parameters of the RFC models are presented for all smartphone users. The unique parameter combinations with respect to the number of trees are described, along with their corresponding accuracy values.

Smartphone User	Trees	Mtry	Node Size	Accuracy
A	200	3	1	0.7457
	350	3	1	0.7429
	500	3	10	0.7857
B	200	3	15	0.7920
	350	3	1	0.8029
	500	3	1	0.8029
C	200	3	5	0.8450
	350	3	10	0.8378
	500	3	15	0.8305
D	200	3	1	0.9060
	350	3	5	0.9060
	500	3	1	0.9037

optimal model, each unique parameter combination was considered individually. For each user, a model was first trained on the training set using a given parameter combination and then validated on the testing set. This process was repeated for all possible parameter combinations with respect to the trees. After evaluating all models, the optimal model was chosen based on the highest value of accuracy, F1 score, precision, and recall. In this way, the final chosen models ensured a balance between predictive accuracy and robustness. The detailed results of these optimal models are summarized in Table 4.4.

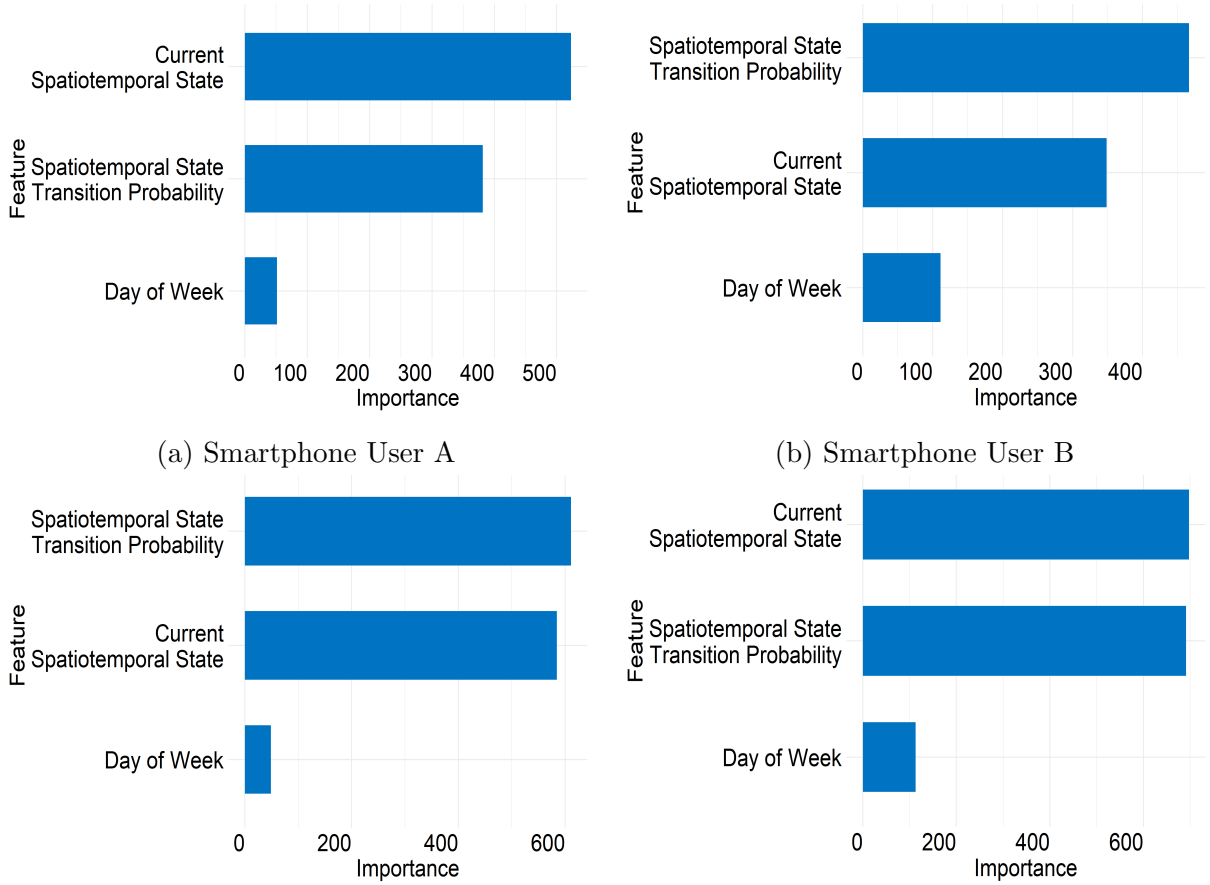
Table 4.4: The selected RFC models with optimal parameters identified through the grid search with a rolling-window cross-validation approach are presented for all smartphone users. Each model was trained on the first 80% of the spatiotemporal sequence and then validated by predicting the next spatiotemporal states, which were compared against the observed next spatiotemporal states in the remaining 20% of the testing set.

Smartphone User	**Trees**	**Mtry**	**Node Size**	**Accuracy**	**F1 Score**	**Precision**	**Recall**
A	500	3	5	0.7857	0.7038	0.7374	0.7198
B	350	3	1	0.8029	0.6881	0.7045	0.7039
C	200	3	5	0.8450	0.7477	0.7564	0.7807
D	200	3	1	0.9060	0.8576	0.8570	0.8940

From Table 4.4, the results signify that the RFC model effectively predicted the next spatiotemporal states for all users. For user A, the model achieves an accuracy of 78.57%, a F1 score of 70.38%, a precision of 73.74%, and a recall of 71.98%. The precision indicates that all the next spatiotemporal states predicted by the model are correct approximately 74%, reflecting the model's ability to minimize false positive predictions. Recall measures the proportion of observed next spatiotemporal states in the testing set that the model successfully predicts. For instance, a recall of 71.98% indicates that the model accurately predicted approximately 72% of the actual next spatiotemporal states observed in the testing set. The F1 score indicates how well the model achieves a trade-off between correctly predicting next spatiotemporal states and capturing the majority of actual next spatiotemporal states observed in the testing set. For user B, the model attained an accuracy of 80.29%, precision of 70.45%, recall of 70.39%, and a F1 score of 68.81%. This suggests that the model prioritizes making accurate predictions of the next spatiotemporal states, even though it may not capture all of the true next spatiotemporal states observed in the testing set. For user C, the model improved results with an accuracy of 84.50%, precision of 75.64%, recall of 78.07%, and a F1 score of 74.77%, indicating that the model reliably predicts the next spatiotemporal states while minimizing both false positives and false negatives. The high precision and recall suggest that the model consistently identifies the

correct next spatiotemporal states while also capturing most of the true next spatiotemporal states observed in the testing set. For user D, the highest predictive performance is obtained with an accuracy of 90.60%, a precision of 85.70%, a recall of 89.40%, and a F1 score of 85.76%. The slightly higher recall compared to precision shows that the model captures nearly all true next spatiotemporal states observed in the testing set while maintaining strong correctness in its predictions.

Overall, these results confirm that the RFC model effectively learns user-specific mobility behaviors and accurately predicts the next spatiotemporal states. The high precision values indicate that the predicted next spatiotemporal states are highly correct, while high recall values show that the model captures the majority of true next spatiotemporal states in the testing set. The F1 score maintains a balance between precision and recall and reinforces the model's ability to provide confident and reliable predictions of users' next spatiotemporal states. To assess the significance of each feature within the fitted RFC models, the Mean Decrease in Gini (MDG) score is used to evaluate feature importance. The key features analyzed include the current spatiotemporal state, spatiotemporal state transition probabilities, and day of week. The contribution of each feature to the model's decision-making process is visually represented in Figure 4.4. From Figure 4.4, it is observed that the current spatiotemporal state has the highest MDG score of 552.8, and indicates its substantial role in predicting the next spatiotemporal state for user A. The strong influence of the current spatiotemporal state aligns with the Markov formulation, which models the prediction of the next spatiotemporal state as conditioned on the current state. The spatiotemporal state transition probability is also a strong contributing feature, with a MDG score of 381.4. The third feature as the day of week has a MDG score of 51.7, and it suggests that it exerts relatively less influence on predicting the user's next spatiotemporal states. For user B, the spatiotemporal state transition probability emerges as the most influential feature, with the MDG score of around 466.2. The current spatiotemporal state also shows a significant contribution, with a MDG score of 348.4, while the day of week feature contributes less, with a MDG score of approximately 111. For user C, the spatiotemporal state transition probability exhibits the strongest influence, attaining a high MDG score of 610.9, and confirms it as the most critical feature in predicting the next spatiotemporal state. The current spatiotemporal state also displays considerable importance, with a MDG score of around 584.3, whereas the day of week feature remains the least impactful, with a MDG score of 48.9. For user D, the current



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 4.4: The feature importance in spatiotemporal state prediction through the RFC models is presented for all smartphone users. The x-axis represents feature importance based on the mean decrease in Gini index, which reflects how much each feature reduces impurity when splitting nodes in the decision trees. The y-axis shows each feature's overall contribution to the model's predictive performance.

spatiotemporal state shows the highest contribution as the MDG score is 697.4, followed closely by the spatiotemporal state transition probability, which ranks second with a score of approximately 691.3. The day of week feature continues to be the least influential, with a MDG score of 112.8. Overall, both the spatiotemporal state transition probability and the current spatiotemporal state stand out as the most significant factors influencing the prediction of the next spatiotemporal states across all users. Although the day of week feature has a comparatively lower impact, its consistent presence indicates a minor but steady influence on users' mobility behavior.

Second, the SVC model with a Radial Basis Function (RBF) kernel is employed to predict the next spatiotemporal states of users. Similar to the RFC model, a Markov-based feature engineering approach is applied to derive features from the observed spatiotemporal sequences. To optimize the SVC model's performance, a grid search over various combi-

nations of the hyperparameters C and sigma is conducted. The C parameter is the cost parameter that controls the trade-off between maximizing the margin and minimizing the misclassification error, and it was evaluated using a range of values such as 0.1, 1, and 10. A smaller C allows for a wider margin, leading to the risk of misclassification and promoting generalization. However, there is the possibility of increasing bias. Conversely, a larger C places more focus on correctly classifying all spatiotemporal states of the training set, which can make the model more complex with a narrower margin. This can lead to overfitting, where the model becomes too specialized for the training set. The sigma parameter determines the width of the RBF kernel, which affects how the model captures patterns in the dataset. This parameter was explored with several values, including 0.001, 0.01, 0.1, and 1, to analyze its impact on classification performance. A smaller sigma value results in a broader influence of individual sample points, leading to a smoother decision boundary that may generalize well; however, there is a risk of underfitting the dataset. Conversely, a larger sigma value makes the model more sensitive to individual data points and creating a more complex decision boundary that fits the training set closely. However, the excessively high sigma values can lead to overfitting, where the model captures noise rather than meaningful patterns. Thus, selecting an optimal sigma value is essential for balancing model complexity and generalization.

The evaluation is performed using rolling-window ahead cross-validation, which preserves the temporal order of the mobility data. In this approach, sequential segments of the training set are used for model fitting, and the immediately following segment is used for validation. The window then moves forward, leaving out the previously validated segment, and the process repeats for five folds. This ensures that the model is evaluated on an unseen test set while respecting the time-dependent structure of the spatiotemporal sequences, providing a robust assessment of predictive performance across different parameter combinations. The grid search with rolling-window cross-validation thus identifies the optimal combination of C and sigma values to enhance the SVC model's predictive performance. The results of this systematic evaluation for all users are summarized in Table 4.5. From Table 4.5, it is noticed that different combinations of parameters led to varying levels of accuracy, while the performance of some parameter combinations is noticeable. To enhance the model's performance, the parameter combinations were selected for each user by prioritizing the highest value of accuracy. The chosen parameters were then applied to train the optimal SVC models on the training set and followed by valida-

Table 4.5: The results of the grid search approach is applied with rolling window ahead cross-validation to identify the optimal parameters, are presented for all smartphone users. For each user, the unique parameter combinations for the SVC model are reported along with their corresponding accuracies.

Parameters		Smartphone User A	Smartphone User B	Smartphone User C	Smartphone User D
C	Sigma	Accuracy	Accuracy	Accuracy	Accuracy
0.1	0.001	0.1357	0.2465	0.1939	0.1391
0.1	0.01	0.1357	0.2465	0.3000	0.1391
0.1	0.1	0.3000	0.3767	0.3758	0.4696
0.1	1	0.3821	0.4465	0.3424	0.5130
1	0.001	0.1357	0.2465	0.3000	0.1391
1	0.01	0.3000	0.3907	0.3758	0.5217
1	0.1	0.4357	0.4977	0.4000	0.5478
1	1	0.4964	0.6186	0.4273	0.5942
10	0.001	0.3000	0.3907	0.3758	0.5217
10	0.01	0.4429	0.5163	0.4212	0.5478
10	0.1	0.5000	0.6093	0.4333	0.5913
10	1	0.5143	0.6512	0.4273	0.6145
100	0.001	0.4429	0.5163	0.4212	0.5565
100	0.01	0.4964	0.5953	0.4333	0.6000
100	0.1	0.5357	0.6512	0.4333	0.6029
100	1	0.5143	0.6465	0.4333	0.6203

tion using the testing set. The optimal model was selected based on the highest value of accuracy, F1 score, precision, and recall. This process was adopted for all users and the results of optimal SVC models are described in Table 4.6.

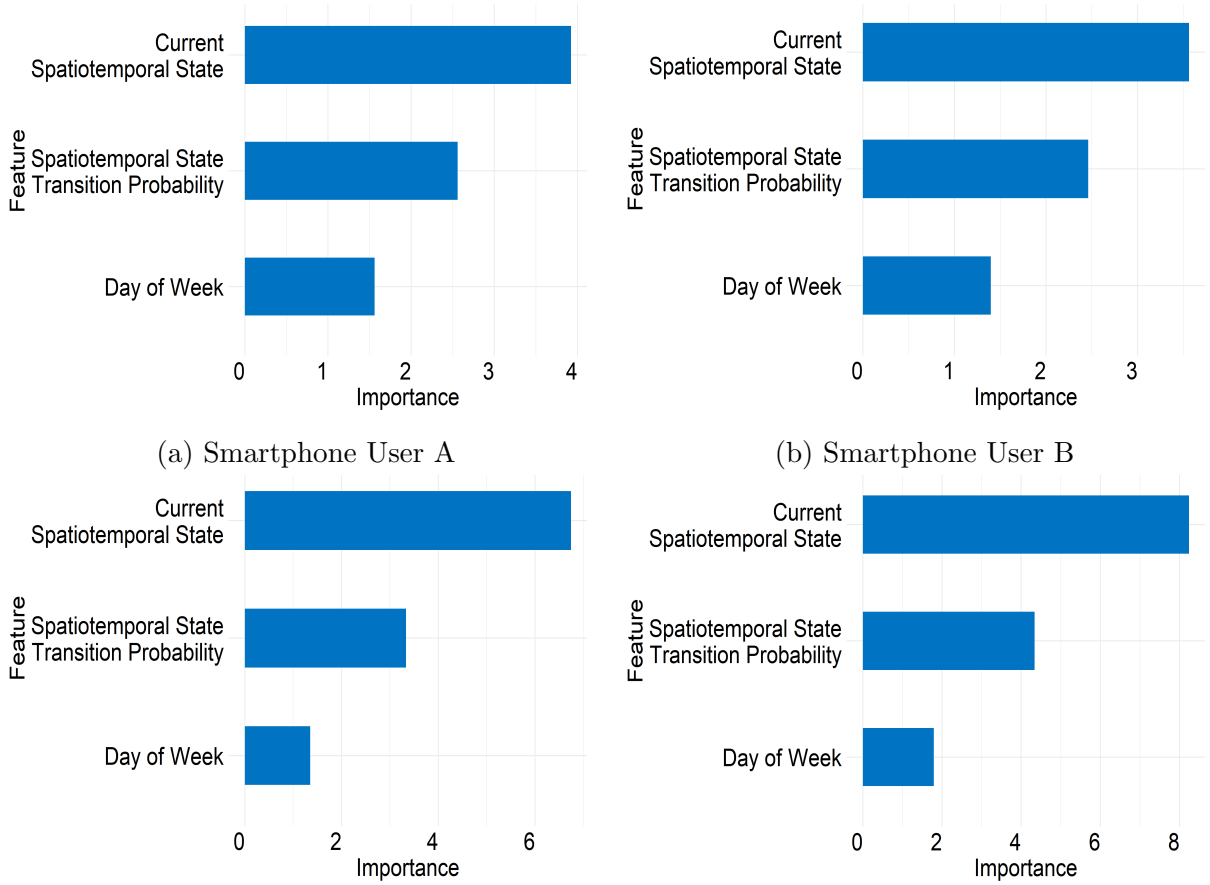
From Table 4.6, the outcomes exhibit that the SVC models effectively predict the next spatiotemporal states for all users, though with slightly lower overall performance compared to the RFC model. For user A, the SVC model achieves an accuracy of 71.14%, a F1 score of 69.44%, with a precision of 67.38% and a recall of 76.00%. The precision indicates that approximately 67% of the next spatiotemporal states predicted by the model are correct, and it reflects the model's ability to limit false positive predictions. A false positive refers to a predicted next spatiotemporal state that does not actually occur in the observed sequence of the testing set. The recall of 76.00% indicates that the model successfully captures around 76% of the observed next spatiotemporal states in the testing set. The F1 score indicates a balanced trade-off between precision and recall, and it suggests that the model is reasonably effective in predicting the next spatiotemporal states while capturing the majority of true next spatiotemporal states. For user B, the model obtained an accuracy of 71.90%, a precision of 67.58%, a recall of 67.83%, and an F1 score of 65.17%. These results suggest that the model performs consistently in identifying

Table 4.6: The selected SVC models with optimal parameters identified through the grid search with a rolling-window cross-validation approach are presented for all smartphone users. Each model was trained on the first 80% of the spatiotemporal sequence and then validated by predicting the next spatiotemporal states, which were compared against the observed next spatiotemporal states in the remaining 20% of the testing set.

Smartphone User	C	Sigma	Accuracy	F1 Score	Precision	Recall
A	100	0.1	0.7114	0.6944	0.6738	0.7600
B	100	0.1	0.7190	0.6517	0.6758	0.6783
C	10	0.1	0.8160	0.7649	0.8210	0.7425
D	100	1	0.8578	0.7571	0.7581	0.7791

true next spatiotemporal states, with the recall being slightly higher than precision. This indicates that the model captured a marginally greater proportion of the actual next spatiotemporal states in the testing set, even if a few of its predictions are less precise. For user C, the model shows improved performance with an accuracy of 81.60%, precision of 82.10%, recall of 74.25%, and a F1 score of 76.49%. The high precision indicates that most predicted next spatiotemporal states are correct, while the slightly lower recall indicates that not all true next spatiotemporal states are captured. Overall, the F1 score confirms a good trade-off and shows that the model can reliably predict next spatiotemporal states while minimizing false positives. For user D, the highest performance is achieved with an accuracy of 85.78%, precision of 75.81%, recall of 77.91%, and an F1 score of 75.71%. The higher recall compared to precision suggests that the model captures most of the true next spatiotemporal states in the testing set, although some predictions are less precise. This reveals that the model is particularly effective at identifying the majority of actual next spatiotemporal states while maintaining reasonable correctness.

In the SVC model, the feature importance is evaluated using permutation-based cross-entropy loss. This approach measures how well the predicted probabilities from the RBF kernel align with the observed next spatiotemporal states in the testing set. Each feature is shuffled individually, and the resulting increase in cross-entropy loss indicates how much the model relies on that feature. The features causing a larger increase in cross-entropy loss are considered more informative, as they help the model to assign higher probabilities to the correct next spatiotemporal states. The detailed feature importance scores for each user are depicted in Figure 4.5. From Figure 4.5, it is observed that the current spatiotemporal state exhibits the highest importance value of approximately 3.90 for user A, and it indicates a dominant role in predicting the next spatiotemporal state. The spatiotemporal state transition probability is the second most influential feature, with an importance



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 4.5: The feature importance for next spatiotemporal state prediction through the SVC model is presented for all smartphone users. The x-axis shows the increase in cross-entropy loss, represented as importance, when each feature is permuted. This reflects the feature's impact on the model's predictive performance. The y-axis lists the features, ranked by their relative contribution to prediction accuracy.

score of 2.60, while the day of week feature shows comparatively lower importance, with a score close to 1.57. For user B, the current spatiotemporal state again dominates, with an importance score of 3.57, followed by the spatiotemporal state transition probability at around 2.56, and the day of week feature showing a moderate contribution with a score of 1.40. For user C, the current spatiotemporal state has the highest importance, with a score of 6.73, followed by the spatiotemporal state transition probability at 3.26, and the day of week feature contributing modestly with an importance score of approximately 1.34. For user D, the current spatiotemporal state maintains a strong influence, with an importance value of 8.11, while the spatiotemporal state transition probability provides a moderate contribution of about 4.35. The day of week feature remains the least influential, with an importance score of around 1.80. Overall, these results suggest that the current spatiotemporal state is consistently the most critical factor for predicting the next

spatiotemporal state, followed by the spatiotemporal state transition probability, while the feature as day of week contributes relatively less. The cross-entropy-based computation ensures that these importance scores reflect how much each feature reduces uncertainty in the model's predictions, making it a rigorous measure of predictive influence.

Third, the MLPC model is also employed to predict the next spatiotemporal states of all users. To enhance its performance, a grid search procedure was conducted with varying parameters for the model's architecture and regularization. Two key parameters were considered, such as size and decay. The size parameter defines the number of neurons in the hidden layer, and values of 5, 10, and 15 were tested. A smaller hidden layer size produces a simpler model that may generalize well to unseen test set but might fail to capture complex mobility patterns. In contrast, a larger hidden layer size enables the model to learn richer and more detailed representations of spatiotemporal transitions, but at the risk of overfitting. The decay parameter introduces weight regularization and penalizes large parameter values to prevent overfitting. This parameter was explored with values of 0, 0.01, and 0.1. The lower decay values allow the model to fit more closely to the training set, which can improve training accuracy, but it increases overfitting risk. However, the higher decay values constrain the weights, encouraging the model to rely less on individual patterns and thus improving generalization. To ensure robust evaluation of these parameter combinations, the rolling-window ahead cross-validation was applied while preserving the temporal order of the spatiotemporal state sequences as they were implemented in the RFC and SVC models. The grid search with rolling-window cross-validation thus identifies the optimal combination of size and decay values to enhance the MLPC model's predictive performance. The results of this systematic evaluation for all users are summarized in Table 4.7. From Table 4.7, it is observed that different combinations of parameters led to varying levels of accuracy, while the performance of some parameter combinations is noticeable. To boost the model's performance, the parameter combinations were selected for each user by prioritizing the highest value of accuracy. The chosen parameters were then applied to train the MLPC models on the training set and followed by validation using the testing set. The optimal model was selected based on the highest value of accuracy, F1 score, precision, and recall. This process was adopted for all users and the results of optimal MLPC models are described in Table 4.8.

From Table 4.8, the results indicate that the MLPC model is also capable of predicting the next spatiotemporal states for all users, although its performance varies across users.

Table 4.7: The results of the grid search approach is applied with rolling window ahead cross-validation to identify the optimal parameters, are presented for all smartphone users. For each user, the unique parameter combinations for the MLPC models are reported along with their corresponding accuracies.

Parameters		Smartphone User A	Smartphone User B	Smartphone User C	Smartphone User D
Size	**Decay**	**Accuracy**	**Accuracy**	**Accuracy**	**Accuracy**
5	0	0.4143	0.5023	0.4061	0.4551
5	0.01	0.4536	0.5953	0.3848	0.4957
5	0.1	0.3857	0.5814	0.4242	0.4783
10	0	0.4321	0.6047	0.4333	0.5246
10	0.01	0.5000	0.6093	0.4333	0.5507
10	0.1	0.4714	0.5907	0.4152	0.5275
15	0	0.4393	0.5767	0.4091	0.5565
15	0.01	0.5429	0.6186	0.4152	0.5797
15	0.1	0.5143	0.5581	0.3879	0.5565

Table 4.8: The selected MLPC models with optimal parameters identified through the grid search with a rolling-window ahead cross-validation approach are presented for all smartphone users. Each model was trained on the first 80% of the spatiotemporal sequence and then validated by predicting the next spatiotemporal states, which were compared against the observed next spatiotemporal states in the remaining 20% of the testing set.

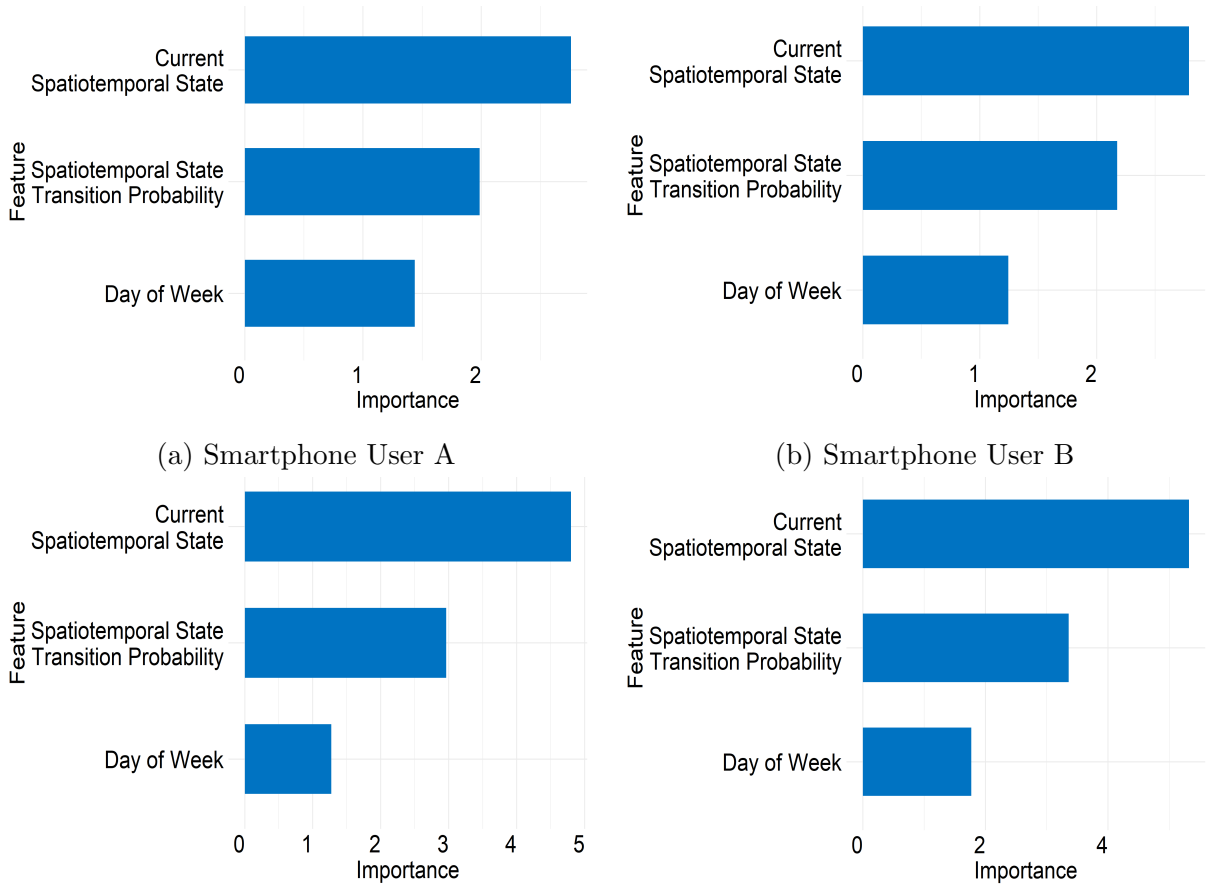
Smartphone User	Size	Decay	Accuracy	F1 Score	Precision	Recall
A	15	0.01	0.6629	0.7717	0.7443	0.8324
B	15	0.01	0.6971	0.7058	0.6897	0.7589
C	10	0.01	0.7990	0.7310	0.8045	0.7621
D	15	0.01	0.8234	0.7754	0.7637	0.8672

For user A, the model achieves an accuracy of 66.29%, a precision of 74.43%, a recall of 83.24%, and a F1 score of 77.17%. The higher recall compared to precision indicates that the model effectively captures the majority of the actual next spatiotemporal states observed in the testing set, although some predictions may not be entirely correct. This shows that the model has a strong ability to detect true next states but allows a few false positives. For user B, the model obtained an accuracy of 69.71%, a precision of 68.97%, a recall of 75.89%, and a F1 score of 70.58%. These results suggest that the model maintains a balanced trade-off between correctly predicting next spatiotemporal states and capturing most of the actual next states in the testing set. The slightly higher recall indicates that the model is marginally better at identifying true next spatiotemporal states than maintaining the absolute prediction precision. For user C, the model provides improved results with an accuracy of 79.90%, a precision of 80.45%, a recall of 76.21%, and a F1 score of 73.10%. The close alignment of precision and recall suggests that the model reliably

predicts next spatiotemporal states while effectively minimizing both false positives and missed true states. For user D, the model achieves its best performance with an accuracy of 82.34%, a precision of 76.37%, a recall of 86.72%, and a F1 score of 77.54%. The higher recall compared to precision suggests that the model successfully captures most of the true next spatiotemporal states observed in the testing set while maintaining a solid level of correctness in its predictions. These results indicate that the MLPC model effectively learns user-specific mobility patterns and achieves consistent predictive performance across all users. The slightly higher recall values reveal that the model tends to favor capturing the majority of true next spatiotemporal states.

Similarly to the SVC model, the feature importance is evaluated using a permutation-based approach with cross-entropy loss for the MLPC models. Each feature is shuffled individually, and the resulting increase in cross-entropy loss indicates how much the model relies on that feature. The calculated feature importance scores for each user are shown in Figure 4.6. From Figure 4.6, it can be observed that the current spatiotemporal state is the most influential feature with an importance of around 1.73 for user A, followed by the spatiotemporal state transition probability with a value slightly above 2, while the day of week feature contributes the least, around 1.44. A similar pattern is noticed for user B, as the current spatiotemporal state has an importance of about 2.79, the spatiotemporal state transition probability has an importance of 2.22, and the day of week is around 1.21. In the case of user C, the current spatiotemporal state again dominates, with an importance score of 4.84, followed by the spatiotemporal state transition probability at around 2.92. Whereas the day of week feature remains relatively minor, approximately 1.29. Similarly, for user D, the current spatiotemporal state emerges as the most significant predictor, with an importance value of 5.34, followed by the spatiotemporal state transition probability at 3.41. The day of the week feature continues to contribute the least, with an importance value of approximately 1.80. Overall, these outcomes indicate that the current spatiotemporal state and the spatiotemporal state transition probability are the dominant factors in predicting the next spatiotemporal state across all users, while the day of week feature contributes minimally.

Since the RFC, SVC, and MLPC, were trained using the same configuration and feature set, it is essential to identify the model that performs best across all users for the next spatiotemporal state prediction. Whereas the MC-1 and MC-2 models are based solely on spatiotemporal state transition probabilities, and they do not incorporate feature infor-



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 4.6: The feature importance for next spatiotemporal state prediction using the MLPC model is presented for all smartphone users. The x-axis shows the increase in cross-entropy loss, represented as importance, when each feature is permuted. This reflects the feature's impact on the model's predictive performance. The y-axis lists the features, ranked by their relative contribution to prediction accuracy.

mation by default, which makes them not fully comparable to the feature-based models. Nevertheless, Figures 4.7 and 4.8 illustrate the comparison of models based on accuracy and F1 score. From Figure 4.7, it is observed that the RFC model performs well with an accuracy of 78.57%, followed by SVM with 71.14% for user A. The MLPC model shows a decent performance with 66.29%, while both the MC-1 and MC-2 models are far behind, and they are achieving an accuracy of 3.43% and 18.29%, respectively. For user B, the RFC model achieves the accuracy of 80.29%, clearly outperforming the other models. The SVC model follows with an accuracy of 71.90% while the MLPC model performs at a comparable level with 69.71% as well. In contrast, the MC-1 and MC-2 models exhibit considerably lower accuracies of only 8.03% and 27.37%, respectively, highlighting their limited predictive capability for this user. For User C, the RFC model once again achieves the highest accuracy at 84.50%, with the SVC model following closely at 81.60%. The

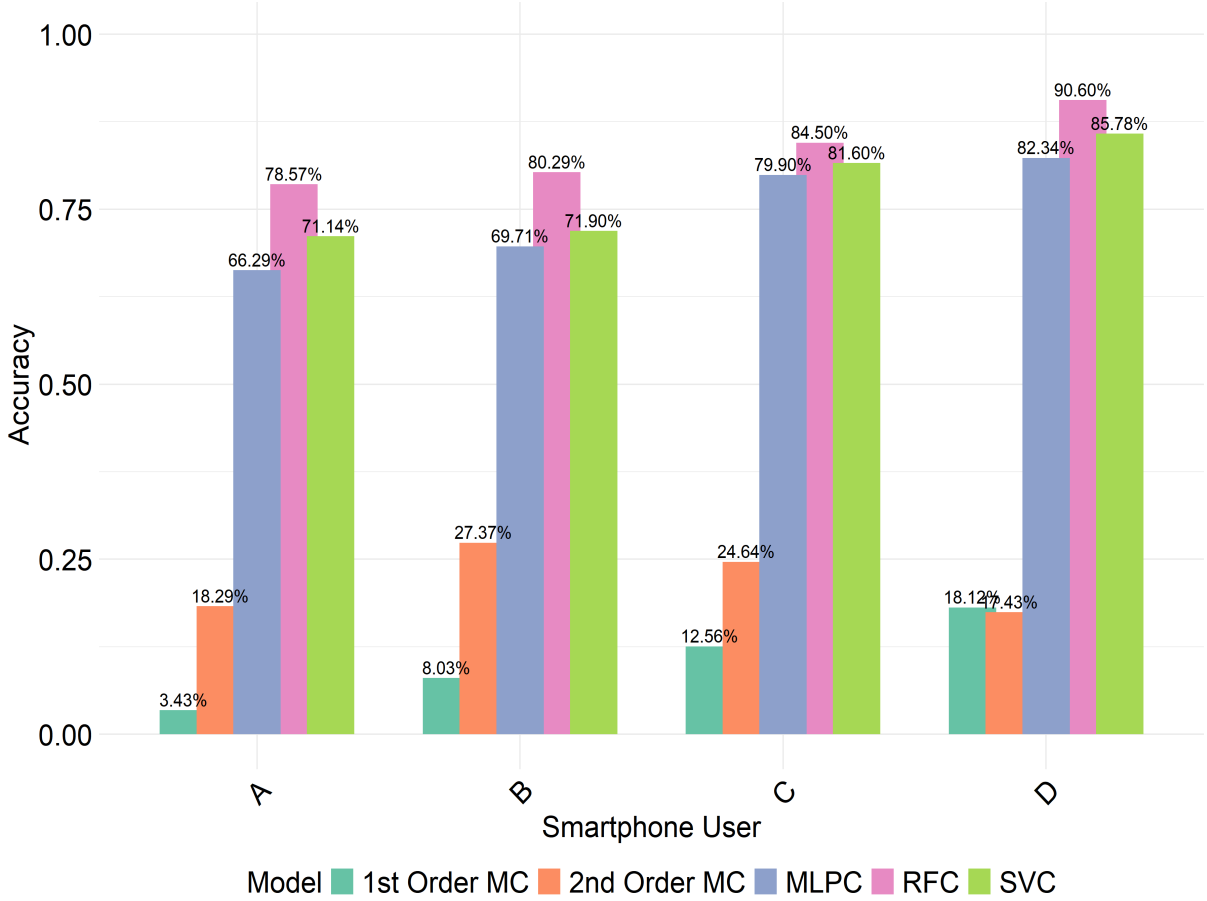


Figure 4.7: The comparison of models is based on their accuracy. The y-axis represents the accuracy scaled between 0 and 1, while the x-axis corresponds to individual smartphone users. Distinct colors denote different models, and the percentage labels on the bars indicate the percentage accuracy, reflecting each model's overall correctness in identifying the next spatiotemporal states.

MLPC model attains a moderate accuracy of 79.90%, while the MC-1 and MC-2 models remain much lower with accuracies of only 12.56% and 24.64%, respectively. For user D, the RFC model shines with the highest accuracy at 90.60%, followed by the SVC model at 85.78%. The MLPC model also delivers a strong performance at 82.34%. In contrast, the MC-1 and MC-2 models perform the weakest, with accuracies of only 18.12% and 17.43%, respectively. Generally, the RFC model consistently outperforms the others by a significant margin across users. The SVC and MLPC models achieve moderate to reliable accuracy, whereas both the MC-1 and MC-2 struggle with much lower performance across all users. From Figure 4.8, it can be observed that the MLPC model achieves the highest F1 score of 77.17%, indicating strong overall predictive performance and a well-balanced trade-off between precision and recall for user A. The RFC model follows with a F1 score of 70.38%, showing slightly lower performance, however, still effective in predicting the next spatiotemporal states. The SVC model obtains the F1 score of 69.44%, reflecting compa-

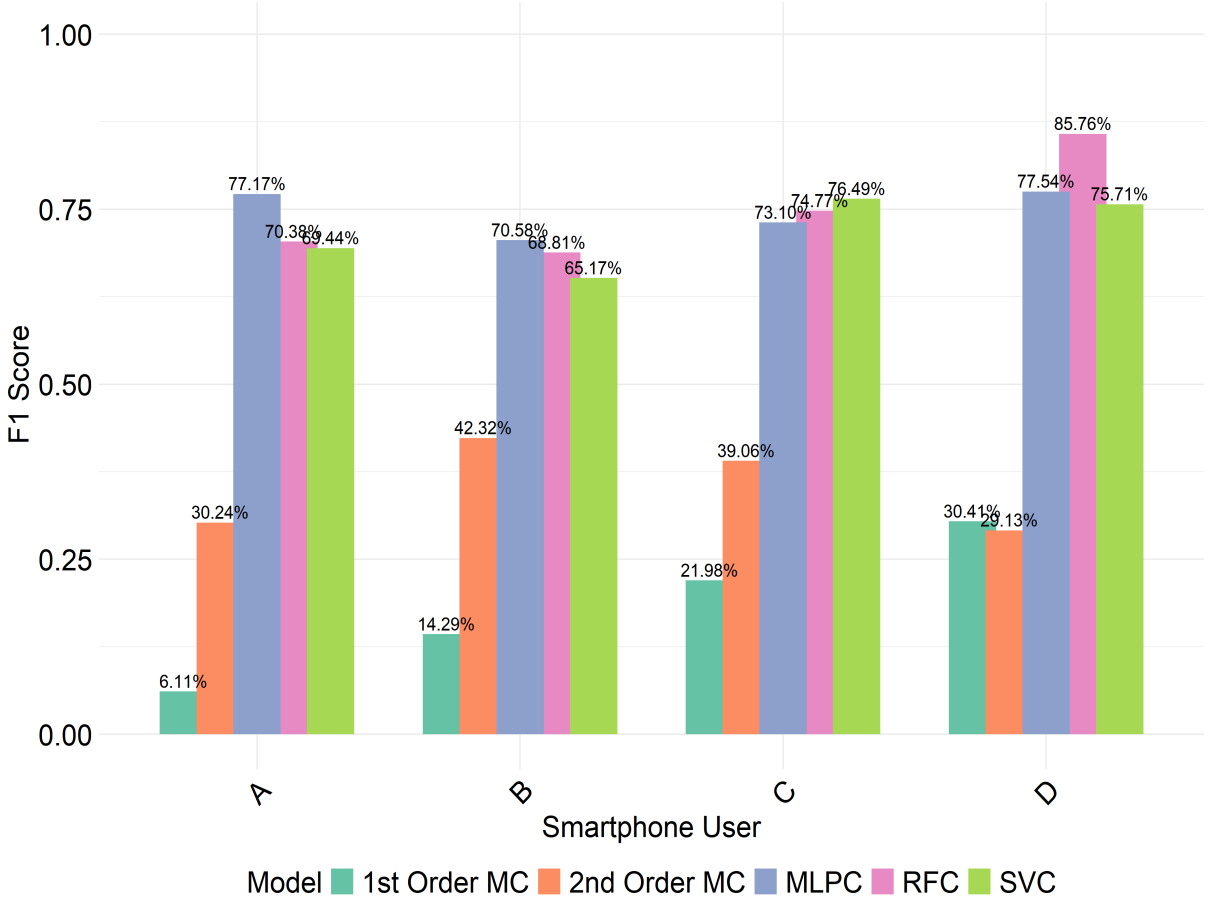


Figure 4.8: The comparison of models is based on their F1 scores. The y-axis represents the F1 scores scaled between 0 and 1, while the x-axis corresponds to individual smartphone users. Distinct colors denote different models, and the percentage labels on the bars indicate the F1 scores in percentage, reflecting each model's ability to correctly identify spatiotemporal states while minimizing errors.

rable results. In contrast, the first-order MC and second-order MC models perform poorly, with F1 scores of only 6.11% and 30.24%, respectively, and these indicate the limited ability to capture the user's movement patterns. For user B, the MLPC model again obtains the highest F1 score of 70.58%, followed by the RFC model at 68.81%. The SVC model achieves a F1 score of 65.17%, showing moderate performance. However, the first-order MC and second-order MC models attain very low F1 scores of 14.29% and 42.32%, respectively. These results suggest that the simpler MC models are unable to capture user B's complex spatiotemporal behavior, while the ML-based models perform more consistently. For user C, the SVC model achieves the highest F1 score of 76.49%, closely followed by the RFC model at 74.77% and the MLPC model at 73.10%. This indicates that the models effectively capture the spatiotemporal transitions and maintain a good balance between precision and recall. In contrast, the first-order MC and second-order MC models show much lower F1 scores of 21.98% and 39.06%, respectively, and this reflects weak predictive

capability and limited generalization. For user D, the RFC model achieves the highest F1 score of 85.76%, exhibiting strong predictive accuracy and excellent balance between precision and recall. The MLPC model follows with a F1 score of 77.54%, while the SVC model performs slightly lower at 75.71%. Conversely, the first-order MC and second-order MC models perform poorly, with F1 scores of 30.41% and 29.13%, respectively, and indicate that they fail to capture the dynamic and complex nature of user D's mobility spatiotemporal patterns. These results confirm that the RFC, SVC, and MLPC models significantly outperform both the first-order and second-order MC models across all users. Among these models, the MLPC shows a slightly better balance between precision and recall for users A and B, while the RFC model indicates strong predictive capabilities for users A, B, and D, highlighting its superior reliability in capturing user mobility patterns compared to the SVC. Although the SVC model still exhibits robust generalization in identifying spatiotemporal transitions. In contrast, the MC-1 and MC-2 models consistently underperform with significantly lower F1 scores among models.

4.4 Discussion

The main focus of this chapter lies in predicting the spatiotemporal states of smartphone users by introducing a Markov-based feature engineering, which assumes that a user's next spatiotemporal state conditionally depends primarily on their current spatiotemporal state along with other features. The study relies on mobility data collected from four smartphone users, where each record contains only latitude, longitude, and timestamp, without any additional contextual information such as user intent, transportation mode, or surrounding environment. However, working with this type of dataset presents several challenges. First, the data suffer from irregular time intervals between recorded coordinates, making it difficult to consistently capture user movements. As shown in Figure 3.2, the frequency of timestamp records varies significantly across dates. Some days contain dense clusters of observations, while others have only a few, leaving gaps that disrupt continuity in the mobility patterns. In addition, missing values also occur likely due to technical issues such as internet disconnection or devices being switched off. These irregularities not only introduce uncertainty but also complicate the learning of reliable mobility transitions. This chapter emphasizes careful handling of data quality issues and focuses on the Markov-based modeling approach within machine learning models through the development of transition-based features, specifically the spatiotemporal state transition probabilities. These improvements enable the models to remain reliable and effective even when the data is incomplete or irregularly collected. Another key challenge arises from the diversity in mobility patterns among users. Some users primarily move within the city center, exhibiting frequent but localized movement, while others occasionally take longer trips beyond the city limits, creating a sparser mobility pattern Figure 3.1. These variations make movement prediction challenging, as some users follow relatively consistent routines, whereas others display unpredictable behaviors. Capturing these differences in a predictive model requires careful consideration of both spatial and temporal variations. A key challenge and the primary objective of this chapter was the effective incorporation of spatiotemporal information into the predictive models. To address this, spatiotemporal states were constructed by combining spatial states with temporal states that were manually defined based on hourly intervals. As the evaluation of spatial states constructing techniques was not the focus of this study, a GMM with equal volume and shape was adopted as a practical solution to ensure consistency. This selection ensured that all

recorded locations, including those situated far from the city center, were retained in the analysis rather than being treated as noise. Excluding such locations could result in the loss of potentially valuable information, which might negatively affect the accuracy and reliability of the mobility modeling process. For spatiotemporal state prediction, the MC-1 and MC-2 models were trained on a training set comprising the first 80% of spatiotemporal states from the mobility dataset. Maintaining the temporal order of the dataset is crucial in mobility prediction, as mobility data inherently represent a spatiotemporal sequence where the order of observations directly impacts the accuracy of predictions. The derived spatiotemporal state transition probabilities are entirely obtained from the training set. Class imbalance can arise when certain spatiotemporal states are rarely visited by the user or are absent from the training set, meaning that the model has no information about these states during learning. Consequently, the model tends to predict the more frequently observed spatiotemporal states, which explains why the MC-1 and MC-2 model achieves a perfect recall of 1 for all users. The high recall indicates that the model effectively identifies previously observed spatiotemporal states. But it also reflects a bias toward dominant states and a limited ability to generalize to less frequent transitions. The performance of the MC models declines as the number of spatiotemporal states increases, leading to spatiotemporal state imbalance. The spatiotemporal transition matrix becomes sparse, with many transitions having very low or zero probabilities, which makes it difficult for the model to predict movements to rare states (Figure 4.3: a, b, c, d). As the number of spatiotemporal states grows, the model is also prone to overfitting, memorizing frequently occurring state sequences rather than learning generalizable transition patterns. Overfitting is further worsened by the limited training set, which spans only forty-eight out of sixty days and contains irregularly recorded coordinates with unequal numbers of timestamps across different dates. This irregularity results in a shortage of samples for certain spatiotemporal transitions, reducing the model's ability to learn rare or less frequent patterns. Overall, these factors highlight the challenges of class imbalance, sparsity, and limited data in accurately modeling user mobility using MC-based approaches. In the second phase of the chapter, the Markov-based feature engineering is introduced in RFC, SVC, and MLPC models. In these models, the next spatiotemporal state is predicted by considering the current spatiotemporal state along with the spatiotemporal state transition probabilities and the day of the week. The general model framework for spatiotemporal state prediction and feature construction, including the estimation of spatiotemporal state

transition probabilities, is described in this Appendix A.1. Overall, the results indicate that the RFC outperformed both the SVC and MLPC in terms of accuracy. The SVC model generally achieved higher accuracy than the MLPC model. It is also noteworthy that the accuracy of the RFC model varied across users' mobility patterns, with the highest accuracy of 90.60% observed for user D, followed by 84.50% for user C, 80.29% for user B, and 78.57% for user A (Figure 4.7). However, the SVC model exhibited lower F1 scores compared to the MLPC model. In terms of F1 scores, the predictive performance of the models varies across users. These findings align and contrast with results from previous studies in meaningful ways. For instance, (Araújo et al., 2020) explored ensemble RF models incorporating Markov properties and highlighted that next-location prediction should consider more than just the immediately previous location, as mobility patterns often rely on sequences of past behavior. The model based on hybrid trajectories achieved accuracies of 71% and 83% for trajectory orders 2 and 3, respectively. Whereas the individual-level trajectories were based on 50% of selected users, the model obtained an accuracy of 39% and 46% for the same trajectory order. In comparison, traditional models like RF and SVM scored between 48% and 66% for similar trajectory orders, depending on the approach. Similarly, (Bieler et al., 2022) proposed a MRF model integrated with a MC model and highlighted the importance of parameter tuning and the inclusion of contextual features. Their approach achieved an average F1 score of approximately 81% on public transport user data, significantly outperforming baseline models, which reached around 41%. In our study, the RFC achieved F1 scores below 80% for users A, B, and C, while obtaining 85.76% for user D. It is important to note that our results are based on an individual-level smartphone mobility dataset, which differs substantially from the public transport dataset used in (Bieler et al., 2022). Huang et al. (2024) presented the proposed CSLSL model to predict the location category. The model's performance was tested on different datasets by using recall metrics, indicating how often the actual location appears among the top predicted locations. The model achieved 48.8% recall with the top 1 predicted locations on the TKY dataset as compared to higher than baseline models. On the NYC dataset, it still performed strongly, ranking second overall with a recall of 32.7%. A study by (Zhao et al., 2020) focused on the Nokia N95 smartphone data and highlighted the benefits of combining spatial and temporal features. Their results suggest that the MLP model achieved 83.43% accuracy using hybrid features as compared to a model with 48.22% of accuracy when spatial and temporal features were used independently. In our

case, the MLPC obtained the highest accuracy of 77.54% for user D (Figures 4.7). Unlike those studies, which relied on high-quality datasets such as public transportation records and user check-ins with extended time spans and rigorous preprocessing, our results show that accuracy can vary considerably within the same model across different users. This variability reflects the inherent challenges of working with smartphone mobility data, which is often irregular and limited compared to the datasets used in prior research. Compared to some earlier studies, our approach takes a more personalized angle by leveraging an individual-level smartphone mobility dataset. We extended traditional models by directly integrating the Markov formulation and introduced a novel feature, such as spatiotemporal state transition probabilities using the historical movement that have not been previously applied in mobility prediction research. Although the feature as the day of week have been commonly utilized in prior studies. This innovative feature engineering significantly boosted model performance. In particular, the RFC model consistently delivered high accuracy, even across diverse user behavior. The strong performance highlights the value of incorporating a richer trajectory context for achieving more robust and reliable mobility predictions. These comparisons not only validate the effectiveness of our model but also offer deeper insights into current trends and ongoing challenges in mobility prediction research.

Despite the strong performance of models, there are some limitations, such as spatiotemporal class imbalance and a limited training set, as previously discussed in the context of the MC models, that affect the overall effectiveness of these models. Overall, the chapter provides valuable insights into incorporating spatiotemporal information into mobility prediction models and highlights the importance of spatiotemporal state transition probabilities in linking between movements in spatiotemporal states (Figures 4.3, 4.4, 4.5, and 4.6). The results suggest that RFC, SVC, and MLPC models perform well despite the presence of spatiotemporal state imbalance, while the MC models struggle as the number of spatiotemporal states increases. Future work should prioritize high-frequency mobility datasets with coordinates recorded at equal intervals and adopt spatiotemporal state-balancing techniques to enhance the reliability of mobility prediction models. The statistical analysis was conducted using R version 4.1.

Chapter 5

Location Prediction of Smartphone Users with Respect to Fixed Timestamps

5.1 Contribution

The chapter presents an application for location prediction with respect to the fixed timestamps by incorporating Markov-based feature engineering into RFR, SVR, and MLPR models. This chapter addresses the challenge of predicting user locations at fixed timestamps, despite the mobility dataset being recorded at irregular intervals and characterized by long gaps, sparse movements, and heterogeneity among users. To achieve this, users' trajectories are regularized at fifteen-minute intervals using linear interpolation or by retaining the previous coordinates when a predefined condition is met. The presented formulation adopts a Markov approximation in which the user's coordinates at timestamp $t + 15$ (i.e., fifteen minutes later) are assumed to conditionally depend on their coordinates at timestamp t, which corresponds to fifteen minutes earlier, along with other features. However, we explicitly recognize that this assumption is an approximation rather than a strict property of human mobility patterns, which often exhibit dependencies extending beyond the immediately preceding state. In addition, the discrete spatial and temporal features are also considered during the training phase of models. Unlike traditional approaches that rely solely on absolute coordinates, our method incorporates both spatial and temporal features. The discrete spatial features are included as the grid ID visited at timestamp t and

the transition probabilities between grids visited at timestamp t and $t+15$. The temporal features, such as hour of day and day of week are commonly utilized in the literature. The feature importance analysis shows that the next location at timestamp $t+15$ is depend on the location at timestamp t, which aligns with the Markov approximation. To the best of our knowledge, prior studies have not integrated Markov-based feature engineering with discretized spatial features for predicting smartphone users' coordinates. This integration represents a significant contribution to the field of smartphone-based individual mobility modeling.

5.2 Methodology

5.2.1 Random Forest Regression Model

The RFR model is based on nonparametric regression estimation, which includes the observed input vector $y \in Y \subset R^p$, and the aim is to predict a square integrable random response $Z \in R$, which has finite variance, by estimating the regression function $h(x) = E[Z\,|Y = y]$. Consider a training sample $S_n = \{(y_1, z_1), (y_2, z_2), ..., (y_n, z_n)\}$ is independent random variables distributed as the independent prototype pair (Y, Z) by using the dataset S_n to compute an estimate $h_n : Y \to R$ of the function h. Therefore, the estimated regression function h_n is consistent if $E[h_n(y) - h(y)]^2 \to 0$, $n \to \infty$. The expectation is evaluating over $Y$and the training sample S_n. It represents that if the sample size increases infinitely, the expected squared difference between the estimated function $h_n(x)$ and the true function $h(x)$ gets closer to zero. A RFR is a predictor model that is made up of M randomized regression trees. Each tree in the forest makes its own prediction for a given input y. The prediction from the j^{th} tree is represented by $h_n(y; \nu_j, S_n)$, where $\nu_1, \nu_2, ..., \nu_M$ are independent random variables. These variables follow the same distribution as a general random variable ν and are independent of the training sample S_n. Simply, ν helps create variety among the trees by resampling the training sample before growing each tree and determining how splits are made within the tree. The more detail will be explained later. Mathematically, the prediction can be made by the j^{th} trees following a specific formula is described by Biau and Scornet (2016) as follows,

$$h_n(y; \nu_j, S_n) = \sum_{i \in S_n^*(\nu_j)} \frac{1_{Y_i \in A_n(y; \nu_j, S_n)} Z_i}{N_n(y; \nu_j, S_n)}, \tag{5.1}$$

where the notation $S_n^*(\nu_j)$ denotes the bootstrap sample used to train the j^{th} tree ν_j and it is a subset of training sample points drawn with replacement from $S_n(\nu_j)$. Further, $A_n(y;\, \nu_j, S_n)$ is the cell including y, and $N_n(y;\, \nu_j, S_n)$ is the number of preselected sample points that fall into the cell $A_n(y;\, \nu_j, S_n)$. Therefore, the trees can be combined into the shape of forests, and can be computed as follows,

$$m_{M,n}(y; \nu_1, \nu_2, ..., \nu_M, S_n) = \frac{1}{M} \sum_{j=1}^{M} h_n(y; \nu_j, S_n). \tag{5.2}$$

Since the notation M represents the total number of trees, and it can be selected as large as computing resources allow, and it makes sense from a modeling perspective to consider what happens when it is approaching infinity. Instead of working with the fixed number of trees as described in the equation (5.2), the infinite forest estimate as follows,

$$h_{\infty,n}(y;\, S_n) = E_\nu[h_n(y;\, \nu, S_n)]. \tag{5.3}$$

From equation (5.3), it can be noticed that E_ν indicates the expectation taken over the random parameter ν, while keeping the training sample S_n fixed. Further, if $M \to \infty$ is supported by the law of large numbers, which states that as increasing the number of trees leads to almost surely converges to the finite forest estimate, while keeping the training sample S_n fixed as follows,

$$\lim_{M \to \infty} h_{M,n}(y;\, \nu_1, ..., \nu_M,\, S_n) = h_{\infty,n}(y;\, S_n). \tag{5.4}$$

The randomness comes into play in the construction of individual trees. According to the RF model of (Breiman, 2001), each node in a tree corresponds to a hyperrectangular cell which is a specific region in the input space. The process starts with the entire input space Y, as the root of the tree. At each step, a node or its corresponding cell splits into two regions, continuing until the tree reaches its final structure. The terminal nodes or leaves together create a partition of the input space. To build a RF, the algorithm constructs M different randomized trees, before growing each tree, a subset of the original dataset is randomly selected, either with or without replacement, and it can be possible that some data points may appear multiple times while others may not be included at all. Only this randomly selected subset is used in building that specific tree. This randomness ensures that each tree is different, leading to a more diverse and robust model when combined into

a forest.

At each step in building a tree, a split is made within each cell by selecting the best division based on the maximization of the CART criterion (which will be explained later). However, instead of considering all possible directions for splitting, the algorithm randomly selects features out of the p original features and chooses the best split from this subset. This selected group of features is known as M_{try}. The tree continues growing until each cell contains fewer than node size points, at this point no further splits can be made. When predicting a new input y, the tree defines which final cell y falls into and average the target values (Z_i) of all training points in that cell y. It's important to note that both building the tree and making predictions depend only on the randomly selected subset of training points a_n chosen before constructing the tree. This randomness helps ensure that each tree is unique, contributing to the overall strength of the random forest model. Now, let's explain how the CART split criterion works. Let's first consider a tree that does not use subsampling, meaning that it uses the entire original dataset S_n for its construction. Each cell A in the tree includes a certain number of training points and it is denoted by $N_n(A)$ in cell A. The split in a cell A is defined by a pair (j, l), where j represents the p features or dimensions used for splitting. While l is training points position that split along with j^{th} features within the limits of A. Let's C_A denote the set of all possible splits in a cell A. With this setting, the CART split criterion helps to determine the best way to split each cell A by evaluating all possible pairs (j, l). To put it simply, it finds the feature and split point that best separates the data, ensuring that the resulting subgroups are as homogeneous as possible. Thus, with the notation $Y_i = (Y_i^{(1)}, ..., Y_i^{(p)})$ for any pair (j, l) $\in C_A$, the CART-split criterion takes the form as follows,

$$\begin{aligned} L_{reg,n(j,l)} = \tfrac{1}{N_n(A)} \sum_{i=1}^{n} (Z_i - \bar{Z}_A)^2 1_{Y_i \in A} \\ - \tfrac{1}{N_n(A)} \sum_{i=1}^{n} (Z_i - \bar{Z}_{A_L} 1_{Y_i^j < l} - \bar{Z}_{A_R} 1_{Y_i^j \geq l})^2 1_{Y_i \in A} \end{aligned}, \tag{5.5}$$

where, $A_L = \{y \in A : y^{(j)} < l\}$ is the left subregion after splitting the cell A, including the training sample points where the value of j^{th} feature is less than l. While $A_R = \{y \in A : y^{(j)} \geq l\}$ is the right subregion contains training sample points where the values of j^{th} greater than or equal to l. $\bar{Z}_A$ includes subregions $\bar{Z}_{A_L}$ and $\bar{Z}_{A_R}$, and it is the average of Z_i belongs to A including both subregions A_L and A_R) with convention that the average is equal to 0 when no point Y_i belongs to A including both subregions A_L and A_R. The

first term in equation (5) indicates the total square error of prediction in the region A, assuming $\bar{Z}_A$ is the average target value in region A. While the second term expresses the sum of square error for the two subregions as A_L and A_R, where prediction in the subregion A_L based on $\bar{Z}_{A_L}$, and the prediction in the subregion A_R based on $\bar{Z}_{A_R}$. For each cell A, the algorithm evaluates all possible splits $(j,\, l)$ from the set of C_A feasible splits in cell A, and it selects the best split (j_n^*, l_n^*) that maximizes the equation (5.5) as follows,

$$(j_n^*, l_n^*) \in \underset{j \in Mtry}{\arg} \max L_{reg,n}(j, l), \qquad and \quad (j, l) \in \zeta_A. \tag{5.6}$$

To break any ties when selecting the best split, the algorithm always splits at the midpoint between two consecutive data points. This ensures consistency in the splitting process. In RF algorithm of (Breiman, 2001), the splitting criterion is evaluated only on a randomly selected subset of features, rather than considering all features at each split as in the CART technique of (Breiman et al., 2017). However, the technique checks all possible split points to find the best one within the selected subset. Further, the individual trees in a random forest are not pruned. They continue growing until each final cell contains fewer than node size observations unless all data points in a cell share the same feature value. Instead of using the entire dataset S_n, each tree is built using only a subset of observations, which are randomly selected from the original sample. Then, the predictions are based only on these selected observations. If $a_n = n$ the sampling is done with replacement, and the algorithm operates in bootstrap mode. Otherwise, $a_n < n$ the method follows subsampling, which can be done with or without replacement. This design introduces randomness at multilevel in both feature selection and data sampling, making the random forest more robust and resistant to overfitting.

5.2.2 Support Vector Regression Model

Given the training dataset $\{(y_1, z_1), (y_2, z_2), ..., (y_n, z_n)\} \subset R^p \times R$, where $Y \in R^p$ represents the space of the input feature and $Z \in R$ represents the corresponding response variable. In the Support Vector Regression Model (SVRM), the goal is to find a function $f(y)$ that predicts target values with a minimum deviation of ε from the actual values in the training dataset while keeping the model as simple and smooth as possible. This helps avoid unnecessary complexity and improves generalization. According to Smola and

Schölkopf (2004), the function f is represented in a linear form as follows,

$$f(y) = \langle \omega, y \rangle + \beta, \omega \in N, \quad \beta \in R \tag{5.7}$$

where $\langle ., . \rangle$ is the dot product in the N, and it represents the multiplication of two vectors, which helps to measure the similarity between them. The weight vector ω determines the importance of each feature in the model. To keep the model simple and generalizable, we minimize the Euclidean norm $\|\omega\|^2$, which helps control complexity. A smaller norm results in a flatter function, reducing the risk of overfitting and making the model perform better on new data. This optimization process is formulated as a convex problem, and it aims to find the most efficient and optimal solution.

$$minimize \frac{1}{2}\|\omega\|^2, \tag{5.8}$$

$$subject\,to \begin{cases} z_i - \langle \omega, y_i \rangle - \beta \leq \varepsilon \\ \langle \omega, y_i \rangle + \beta - z_i \leq \varepsilon \end{cases}$$

the convex optimization problem works well when there exists a function that can predict all target values with an error of at most ε. In some cases, achieving perfect precision is often impossible. To handle this issue, it is appropriate to introduce the slack variables which allow for some flexibility by tolerating small errors. These variables help the model when the exact predictions are not possible, and make the optimization problem more practical and feasible, while still maintaining a good balance between accuracy and simplicity.

$$Minimize \frac{1}{2}\|\omega\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*), \tag{5.9}$$

and,

$$subject\,to \begin{cases} z_i - \langle \omega, y_i \rangle - \beta \leq \varepsilon + \xi_i \\ \langle \omega, y_i \rangle + \beta - z_i \leq \varepsilon + \xi_i^* \\ \xi_i,\, \xi_i^* \geq 0 \end{cases}$$

where ω is the weight vector that determines the orientation the the regression hyperplane, β is the bias term, and ξ_i, ξ_i^* are slack variables that allow deviation larger than ε. The parameter $C > 0$ controls the trade-off between the flatness of the function and the tolerance of deviations exceeding ε. Specifically, a larger C puts more emphasis on minimizing the training error, while a smaller C promotes smooth and simple models.

The ε intensive loss function used in SVRM is defined as,

$$|\xi|_{\varepsilon} = \begin{cases} 0 \ \ if \ |\xi| < \varepsilon \\ |\xi| - \varepsilon \ \ otherwise, \end{cases} \tag{5.10}$$

meaning that deviations within the margin are ignored, and only deviations outside this margin contribute to the loss. This loss function allows for robustness to small errors and noise in the data, resulting in better generalization.

Instead of solving the problem in its primal form, the dual formulation uses the Lagrange multiplier which expresses the solution using the support vectors as important training points and it also helps the support vectors to work in higher dimensional space without transforming the data and reducing the computational cost and making it possible to use kernel functions for better learning. In the dual form, the solution is represented by the support vectors rather than relying on all data points. This makes the model sparser and more efficient, as only a subset of points contributes to the final prediction. The standard dual formulation using the Lagrange multiplier is described as follows,

$$\begin{aligned} L = \frac{1}{2}\left\|\omega\right\|^2 + C\sum_{i=1}^{n}(\xi_i + \xi_i^*) - \sum_{i=1}^{n} a_i\left(\varepsilon + \xi_i - z_i + \langle\omega, y_i\rangle + \beta\right) \\ - \sum_{i=1}^{n} a_i^*\left(\varepsilon + \xi_i^* + z_i - \langle\omega, y_i\rangle - \beta\right) \\ - \sum_{i=1}^{n}(\eta_i\xi_i + \eta_i\xi_i^*). \end{aligned} \tag{5.11}$$

The constraints such as a_i, a_i^*, η_i, $\eta_i^* \geq 0$ must have to satisfy the positivity condition. Whereas the notations a_i and a_i^* help to define the model by selecting the most important training point as a support vector to construct the function. While the notations η_i and η_i^* allow some flexibility in the model to handle cases where perfect predictions are not possible. The purpose is to satisfy the saddle point property in the Lagrangian optimization as maximize the function with respect to primal variables such as ω, β, ξ_i, and ξ_i^* have to vanish for optimality. Therefore, taking the partial derivatives of L with respect to primal variables ω, β, ξ_i, and ξ_i^*, and equating them to zero as follows,

$$\frac{\partial L}{\partial \beta} = \sum_{i=1}^{n}(a_i^* - a_i) = 0, \tag{5.12}$$

$$\frac{\partial L}{\partial \omega} = \omega - \sum_{i=1}^{n} (a_i^* - a_i) y_i = 0, \tag{5.13}$$

$$\frac{\partial L}{\partial \xi_i^{(*)}} = C - a_i^{(*)} - \eta_i^{(*)} = 0, \tag{5.14}$$

substituting equations (5.12), (5.13), and (5.14) into (5.11) and gives the dual optimization problem as follows,

$$Maximize \left\{ -\frac{1}{2} \sum_{i,j=1} (a_i - a_i^*)(a_j - a_j^*) \langle y_i, y_j \rangle - \varepsilon \sum_{i=1}^{n} (a_i + a_i^*) + \sum_{i=1}^{n} z_i (a_i - a_i^*) \right\}, \tag{5.15}$$

Subject to;

$$\sum_{i=1}^{n} (a_i - a_i^*) = 0 \quad \text{and} \quad a_i,\, a_i^* \in [0, C].$$

The dual variables such as η_i and η_i^* are eliminated through condition (5.14), which allows some training points to violate the margin. Therefore, the equation (5.13) can be rewritten as follows,

$$\omega = \sum_{i=1}^{n} (a_i - a_i^*) y_i \quad and \quad therefore \quad f(y) = \sum_{i=1}^{n} (a_i - a_i^*) \langle y_i, y \rangle + \beta. \tag{5.16}$$

Therefore, this is called a support vector expansion, which w can be completely described as a linear combination of the training point y_i by dual coefficients a_i and a_i^*. Even for evaluating $f(y)$, it is not needed to compute ω explicitly (although this may be computationally more efficient in the linear setting). The notation β is a bias, and the Karush Kuhn Tucker (KKT) condition helps to compute the bias, which states that the product between the dual variables and constraints must vanish. In this case, it is described as follows,

$$a_i(\varepsilon + \xi_i - z_i + \langle \omega, y_i \rangle + \beta) = 0, \tag{5.17}$$

$$a_i^*(\varepsilon + \xi_i^* - z_i + \langle \omega, y_i \rangle + \beta) = 0, \tag{5.18}$$

$$(C - a_i)\xi_i = 0, \tag{5.19}$$

and,

$$(C - a_i^*)\xi_i^* = 0. \tag{5.20}$$

The KKT condition ensures that only certain training points lie on the margin boundaries

and actively can determine the bias β and these training points contribute to the final prediction function. The following conclusion can be made as the training points with the corresponding $a_i^* = c$ lie outside of the insensitive margin and means that they have significant prediction errors. The dual variables a_i and a_i^* cannot be nonzero at the same time, meaning that the training points can exceed the margin either on the lower side or upper side, but not both sides simultaneously, and this would require nonzero slacks in both directions. Ultimately, consider the training points as a_i, $a_i^* \in (0, C)$ means that they lie on the margin boundaries, and it is known as the support vectors, and the slack variables such as $\xi_i$and ξ_i^* become zero. Simplifying the equation (5.17) and (5.18) and making the final formulation as follows,

$$\beta = z_i - \langle \omega, y_i \rangle - \varepsilon \quad for\ a_i \in (0, C), \tag{5.21}$$

and,

$$\beta = z_i - \langle \omega, y_i \rangle + \varepsilon \quad for\ a_i^* \in (0, C). \tag{5.22}$$

The support vector algorithm can be made to solve nonlinear problems by simply preprocessing as by transforming the input space to high dimensional space using the mapping function $\phi(x)$. This transformation permits to capture highly complex patterns, even if the training points are not separable in their original form. After transformation, the SVR algorithm is applied just like in the linear case. Therefore, the expression in equation (5.16) becomes,

$$\omega = \sum_{i=1}^{n} (a_i - a_i^*)\phi(y_i), \tag{5.23}$$

and,

$$f(y) = \sum_{i=1}^{n} (a_i - a_i^*)k(y_i, y) + \beta, \tag{5.24}$$

The important difference between the linear and nonlinear cases is that the weights vector w is not explicitly known. Instead of finding the simplest (flattest) function in the original input space, the optimization process seeks the flattest function in the higher-dimensional feature space. Therefore, the SVR model is expressed with kernel function is expressed as follows,

$$f(y) = \sum_{i=1}^{n} (a_i^* - a_i)k(y_i, y) + \beta, \tag{5.25}$$

where the notations a_i and a_i^* are the Lagrange multipliers, and they play an essential

role in determining how much influence each training point has on the final prediction. Furthermore, the notation $k(x_i, x)$ represents the kernel function. The kernel function is defined as a linear dot product of the nonlinear mapping as follows,

$$k(y_i, y) = \varphi(y_i)\varphi(y), \tag{5.26}$$

it measures the similarity between two training points in the transformed space, making it possible to capture the relationship in the training point.

5.2.3 Multilayer Perceptron Regression Model

The MLP model can be effectively configured to handle regression problems by modifying the output layer and choosing an appropriate loss function. Like the MLPC model, the MLPR model consists of three fundamental components such as an input layer, one or more hidden layers with nonlinear activation functions, and a final output layer. The input layer receives a scaled input vector, which is passed forward through the network via weighted connections. Each neuron in the hidden layer computes a weighted sum of its inputs, adds a bias term, and processes the result through a nonlinear transfer function. The log-sigmoid activation function is commonly employed in hidden layers, and it is defined in Equation (4.22) of Section 4.2.3 Unlike classification, the output layer consists of a single neuron with a linear activation function to generate a continuous-valued prediction. As outlined by (Agirre-Basurko et al., 2006), the general output formulation of the MLPR model is presented in Equation (4.21) of Section 4.2.3. Feed-forward neural networks serve as powerful tools for modeling complex, non-linear relationships in datasets. Nevertheless, they are highly flexible and can approximate any continuous function when given a sufficiently large hidden layer. This property has been mathematically supported by (Cybenko, 1989), and (Funahashi, 1989), who demonstrated that neural networks with linear output layers can uniformly approximate any function on compact sets. In practice, the network's weights must be optimized to minimize an appropriate loss function. One common approach is to use the least squares error function described as follows,

$$E = \frac{1}{L}\sum_{k=1}^{L}(t_k - z_k^o)^2, \tag{5.27}$$

where t_k represents the target values and z_k^o is the output for the k layer. To enhance the generalization capability of the MLPR model and reduce the risk of overfitting, a

regularization term is typically added to the loss function. This term penalizes large weight values, helping to smooth the learning process and limit model complexity. For regression tasks, choosing an appropriate regularization strength is crucial. As suggested by Venables and Ripley (2013), the effective values usually range between 10^{-4} and 10^{-2}, striking a balance between underfitting and overfitting while maintaining predictive accuracy.

Linear Interpolation

The linear interpolation is a numerical technique used to compute unknown values of a function between two known data points. The approach assumes that the variation between consecutive data points can be approximated by a straight line. According to (Chapra et al., 2011), the given two points (y_0, z_0) and (y_1, z_1) with $y_0 < y < y_1$, the interpolated value z corresponding to y can be obtained using the equation as follows,

$$z = z_0 + \frac{(y - y_0)(z_1 - z_0)}{(y_1 - y_0)}. \tag{5.28}$$

The nominator term represents the proportional change in the dependent variable, while the denominator normalizes the step along the independent variable axis. It is an efficient and straightforward way to approximate values under the assumption of linearity.

5.2.4 Evaluation Metrics

Mean Absolute Error

Mean Absolute Error (MAE) is the average of absolute values of errors, and the error is computed between the observed and predicted values. The range of MAE from zero to infinity, with lower values indicating better model performance. According to (Shaukat et al., 2020; Mani et al., 2022), the mathematical formula is described as follows;

$$MAE = \frac{1}{N} \sum_{i=1}^{N} |O_i - P_i|, \tag{5.29}$$

where the notation O_i is the observed value, P_i denotes the predicted value, and N is the total number of observations.

Root Mean Square Error

Root Mean Square Error (RMSE) is the square root of the mean of the squared errors. It indicates how close the observed values are to the predicted values. RMSE ranges from

zero to infinity and the value of RMSE close to zero signifies that the model performance is good. According to (Shaukat et al., 2020; Mani et al., 2022), the mathematical formula is described as follows,

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(O_i - P_i)^2}, \tag{5.30}$$

where the notation O_i indicates the observed value, P_i represents the predicted value, and N is the total number of observations.

Haversine Distance Formula

The Haversine formula is a widely used technique for computing the distance between two geographical points using their latitudes and longitudes (Azdy and Darnis, 2020). Haversine distance computation approach takes the Earth's curvature into account, making it particularly useful for navigation, GPS positioning, and mapping applications. It provides the great-circle distance, which is the shortest path between two locations along the surface of a sphere and it ignores any elevation changes such as mountains or valleys (Maria et al., 2020). To compute distance, the formula considers the differences or magnitudes in changes in coordinates between two locations and converts them into radians for accurate trigonometric calculations. According to (Azdy and Darnis, 2020; Maria et al., 2020), the mathematical representation of the Haversine formula is described as follows,

$$\Delta\phi = \phi_2 - \phi_1, \tag{5.31}$$

$$\Delta\lambda = \lambda_2 - \lambda_1, \tag{5.32}$$

the equation (5.31) represents the difference between the latitudes of two points in radians, while the difference between the longitudes of two points in radians is represented by equation (5.32).

$$a = \sin^2(\frac{\Delta\phi}{2}) + \cos(\phi_1).\cos(\phi_2).\sin^2(\frac{\Delta\lambda}{2}), \tag{5.33}$$

$$distnace = 2R * a\tan 2(\sqrt{a}, \sqrt{1-a}), \tag{5.34}$$

where, a helps to compute the great circle distance on a sphere, while $\Delta\phi$ and $\Delta\lambda$ are differences in latitudes and longitudes respectively. The R is the radius of the earth. For each threshold radius T, classify the predictions as correct if distance is within in T as

defined as follows,

$$Correct\,Prediction = \begin{cases} 1, & if\ distance \leq T \\ 0, & otherwise \end{cases} . \tag{5.35}$$

Thus, we get,

$$Accuracy(T) = \frac{\sum_{i=1}^{n} 1(distance_i \leq T)}{n}, \tag{5.36}$$

where 1(.) is the indicator function returns 1 if the predictions fall within the threshold and 0 otherwise. The n represents the total number of predictions. Therefore, the equation 5.36 measures the accuracy within radius threshold to determine whether the predicted coordinates lie within the given radius.

Coefficient of Determination

The coefficient of determination R^2 measures the proportion of the variance in the response variable that can be explained by the explanatory variables in the model. According to Renaud and Victoria-Feser (2010), the formula of R^2 is described as follows,

$$R^2 = 1 - \frac{\sum_{i=1}^{n} (O_i - P_i)^2}{\sum_{i=1}^{n} (O_i - \bar{O})^2}. \tag{5.37}$$

The numerator measures the total squared error between the observed and predicted values, while the denominator captures the total variance in the observed dataset. A higher R^2 value indicates a better fit, as it shows that the model can explain a greater portion of the variation observed in the dataset.

5.3 Results

The chapter presents a Markov-based feature engineering approach derived from a Markov formulation, which is integrated into machine learning models to predict smartphone users' locations. According to this formulation, the next location conditionally depends on the current location along with other discretized spatial and temporal features, making it especially suitable for modeling mobility patterns. For this purpose, an individual-level mobility dataset of four smartphone users is utilized, and the dataset details and preliminary analysis are provided in Chapter 3. The users' dataset contains only latitude, longitude, and timestamp, and is characterized by irregularly spaced timestamps, missing gaps, and the absence of contextual features. These irregularities present significant challenges, particularly for coordinate prediction at the fixed timestamps. The objective of this chapter is to handle irregularly spaced timestamps associated with geographical coordinates and predict user locations at fixed timestamps using Markov-based feature engineering, treated as a practical approximation rather than a strict Markov assumption.

In this chapter, the users' mobility trajectories were regularized so that their locations are consistently represented at fixed fifteen-minute time intervals. According to the Earthquake Network project, the minimum expected temporal gap between two observed locations was approximately thirty minutes. As shown in Figure 3.3 and 3.4, some trajectories exhibit irregular and sometimes lengthy temporal gaps. To maintain analytical consistency while avoiding excessive interpolation, a fifteen-minute interval was selected as a balanced resolution. To populate the missing positions at these regular intervals, two interpolation strategies were applied based on the user's movement behavior. First, if the distance between two consecutive observed locations was less than or equal to 500 meters, the user was assumed to be stationary during that period, as there is a minimum gap of 30 minutes between consecutive timestamps in the dataset. In such cases, the previous location was replicated at each fifteen-minute interval until the next observed point. This approach prevents unnecessary displacement in scenarios where the user likely remains in the same area. Conversely, if the distance between two consecutive observations exceeded 500 meters, it was inferred that the user was in motion. For these segments, linear interpolation was applied to compute intermediate positions proportionally over time. As a result, the trajectory was gradually filled with plausible locations that reflect continuous movement. This combined approach ensures that the final mobility datasets are both temporally uni-

form and behaviorally realistic, allowing for more reliable downstream mobility modeling and analysis. Afterward, the users' mobility datasets were regularized at a fixed 15-minute interval. For consistency in notation throughout this chapter, "t" denotes an observed timestamp corresponding to a location recorded directly from the reconstructed mobility datasets. The notation "t+15" refers to the subsequent point in the timestamp that occurs 15 minutes after "t" within the regularized timeline. Since the aim is to apply the Markov approximation, the next location at "t+15" minutes is assumed to be conditionally dependent only on the previous location at t, along with other relevant features. Accordingly, a sequence of future coordinates at fifteen-minute intervals was generated, such that each determined position at "t+15" follows directly from the observed position at "t". It is essential to preserve the temporal order in mobility prediction, as the datasets are inherently spatiotemporal, and the coordinates are recorded relative to their timestamps. To maintain this structure, each user's mobility dataset was divided into training and testing sets, where the first 80% of observations formed the training set and the remaining 20% constituted the testing set. For fair and consistent modeling, only the training set was used to construct the features and train the ML models. The study area was divided into grid cells of size 2×2 kilometers to cover the entire Istanbul. Then, the Haversine distance was computed between the user's observed coordinates at timestamp t and each grid cell using the training set. Each user's coordinate at timestamp t is assigned the ID of the nearest grid cell. Similarly, this procedure was applied to assign the grid ID to coordinates at timestamp t+15, and it is represented by the grid ID at timestamp t+15. In this way, the movements are represented as a transition from the grid ID visited at timestamp t to the grid ID visited at timestamp t+15. From these transitions, the empirical transition probabilities were then computed between the grid ID visited at timestamp t and timestamp t+15. These probabilities represent the likelihood of a user moving from one grid cell to another within the fifteen-minute interval. Additionally, the temporal features, such as the hour of the day and the day of week were derived with respect to timestamp t to account for the influence of daily and weekly routines on user mobility. The mathematical formulation employed to derive the feature is presented in Appendix B.1.

Firstly, the RFR models were separately trained on the training set for latitude and longitude prediction. The grid search approach with rolling window ahead cross-validation is employed to identify the optimal combination of parameters. In this approach, the original training set is split based on the temporal sequence, ensuring that the chronological order

of observations is preserved. The five folds were considered, and each fold includes the first 80% of the observations, and was used to train the models. The subsequent 4% of the testing set is used for validation. After evaluation, the window is shifted forward by 4%, leaving out the first 4%, and the next 80% of the remaining observations are considered as the training set, with the following 4% used for validation. This process is repeated across five folds, allowing each part of the dataset to serve as a validation segment while maintaining the temporal order. By systematically evaluating all parameter combinations

Table 5.1: The optimal parameter combinations with respect to the number of trees were identified by the grid search with a rolling-window ahead cross-validation for all smartphone users. The reported RMSE values represent the overall RFR model's performance under the selected parameters.

Smartphone User	Coordinate	Trees	Mtry	Node Size	RMSE
A	Latitude	200	4	5	0.0168
	Latitude	350	4	5	0.0168
	Latitude	500	4	5	0.0168
	Longitude	200	6	10	0.0353
	Longitude	350	6	10	0.0353
	Longitude	500	6	10	0.0353
B	Latitude	200	2	10	0.0122
	Latitude	350	2	10	0.0122
	Latitude	500	2	10	0.0122
	Longitude	200	4	10	0.0227
	Longitude	350	4	10	0.0227
	Longitude	500	4	10	0.0227
C	Latitude	200	5	10	0.0164
	Latitude	350	5	10	0.0164
	Latitude	500	5	10	0.0164
	Longitude	200	5	10	0.0269
	Longitude	350	5	10	0.0268
	Longitude	500	5	10	0.0268
D	Latitude	200	2	5	0.0126
	Latitude	350	2	5	0.0126
	Latitude	500	2	5	0.0126
	Longitude	200	3	10	0.0352
	Longitude	350	3	10	0.0353
	Longitude	500	3	10	0.0353

within this rolling-window framework, the approach ensures that the selected parameter combinations are robust and optimized for predicting users' locations ten minutes ahead of the observed timestamp t. The RFR model has three parameters such as mtry, node size, and the total number of trees. The total number of trees was selected randomly as 200, 350, and 500 to determine how the performance of the RFR model varies. For the parameter "mtry", the different values between 1 to 6 are selected to control the number of features, and these are randomly chosen to evaluate at each split in a decision tree. The

parameter node size is also verified with numerous values as 1, 5, and 10. The smaller node size allows trees to grow deeper and capture finer details, but it can also increase the risk of overfitting. In contrast, a larger node size produces simpler trees and might yield better generalization. The grid search approach presents the optimal parameter combinations w.r.t trees separately for latitude and longitude for the RFR models are summarized in Tables 5.1. It is shown in Tables 5.1 that the optimal parameter combinations with respect to the number of trees resulted in varying RMSE values for both latitude and longitude. To further improve model performance, all identified parameter combinations were utilized to train the models on the training set and validated on the testing set. The final optimal model was selected based on the lowest MAE and RMSE values and the highest value of R-square. This procedure was repeated for all users, and the detailed results are presented in Table 5.2.

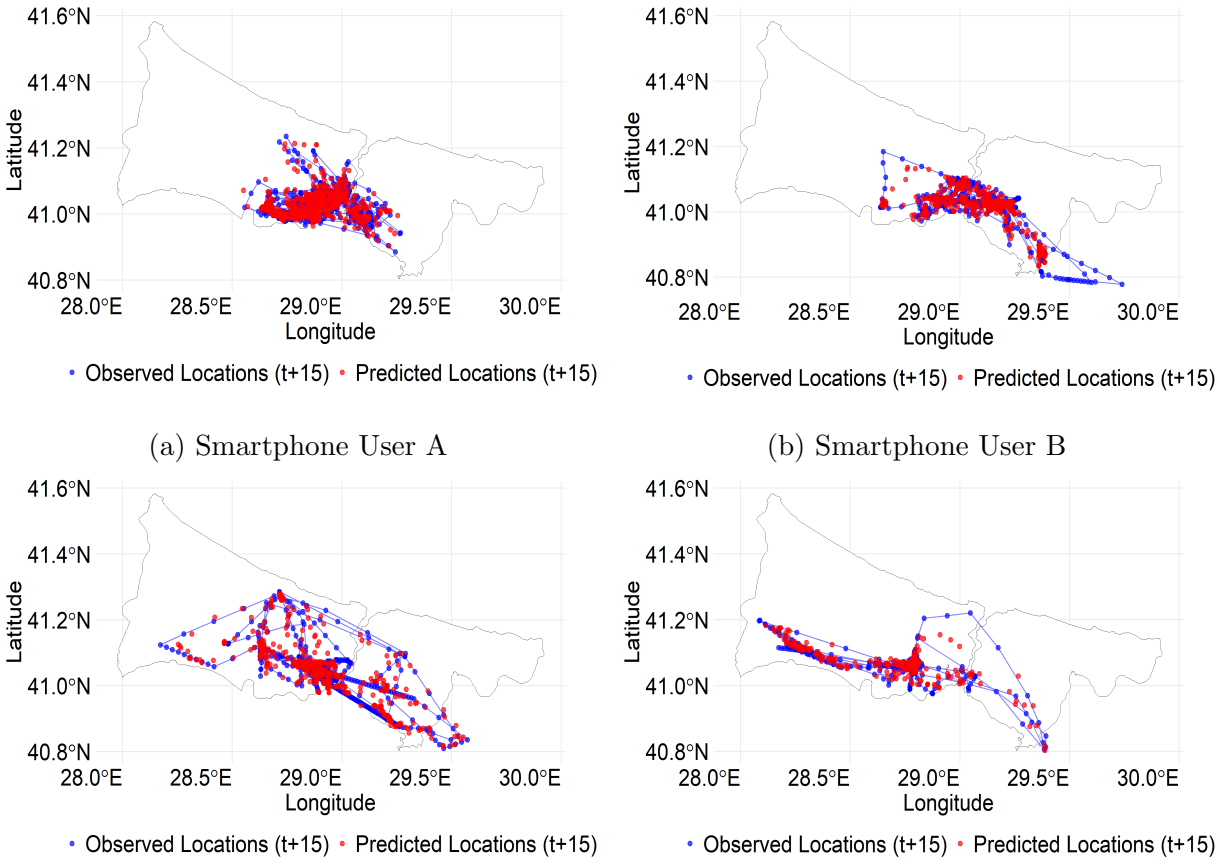
As presented in Table 5.2, the accuracy of the selected RFR models that predicted the

Table 5.2: The selected RFR models with optimal parameters identified through the grid search with a rolling-window cross-validation are presented for all smartphone users. Each model was trained on the first 80% of the observations and then validated by predicting the coordinates at timestamp t+15, which were compared against the observed coordinates at timestamp t+15 in the remaining 20% of the testing set.

Smartphone User	**Coordinate**	**Trees**	**Mtry**	**Node Size**	**MAE**	**RMSE**	**R-squared**
A	Latitude	500	4	5	0.0067	0.0140	0.8103
	Longitude	500	6	10	0.0166	0.0332	0.9481
B	Latitude	500	2	10	0.0066	0.0148	0.9443
	Longitude	500	4	10	0.0127	0.0316	0.9548
C	Latitude	350	5	10	0.0053	0.0126	0.9576
	Longitude	350	5	10	0.0124	0.0272	0.9689
D	Latitude	200	2	5	0.0037	0.0113	0.8690
	Longitude	200	3	10	0.0089	0.0297	0.9537

geographical coordinates as latitude and longitude at timestamp t+15 for the four smartphone users. For user A, the model explains 81.03% of the variation in latitude prediction with a MAE of 0.0067 and RMSE of 0.0140. This indicates that most predicted values are close to the observed latitudes at timestamp t+15. The longitude prediction achieved even higher accuracy, and it explains 94.81% of the variation with a MAE of 0.0166 and RMSE of 0.0332. Although the longitude prediction has a slightly higher error than the latitude prediction. However, the model still showed strong predictive capability. For user B, the model performs well, explaining 94.43% of the variation in latitude prediction with a MAE of 0.0066 and RMSE of 0.0148. Although these values are slightly higher except MAE than

those of user A, however, the predictions remain reliable. For the longitude prediction, the R-squared is 95.48% with a MAE of 0.0127 and RMSE of 0.0316, and it indicates a slightly larger deviation in predictions compared to the latitude prediction. For user C, the model still achieved good performance as it explained 95.76% of the variation in latitude prediction with a MAE of 0.0053 and RMSE of 0.0126. While the longitude prediction is even more precise, as the R-squared is 96.89% with a MAE of 0.0124 and RMSE of 0.0272, highlighting the model's strong ability to predict mobility patterns. For user D, the model explains 86.90% of the variation in latitude prediction with a MAE of 0.0037 and RMSE of 0.0113. The longitude prediction explains 95.37% of the variation with a MAE of 0.0089 and RMSE of 0.0297. Overall, the R-squared values remain strong across all users and confirming that the model successfully captures the movement patterns of smartphone users. While latitude predictions tend to have slightly lower errors compared to longitude. The model's performance is particularly strong for users B and C, where it achieved near-perfect predictive capability.



(a) Smartphone User A

(b) Smartphone User B

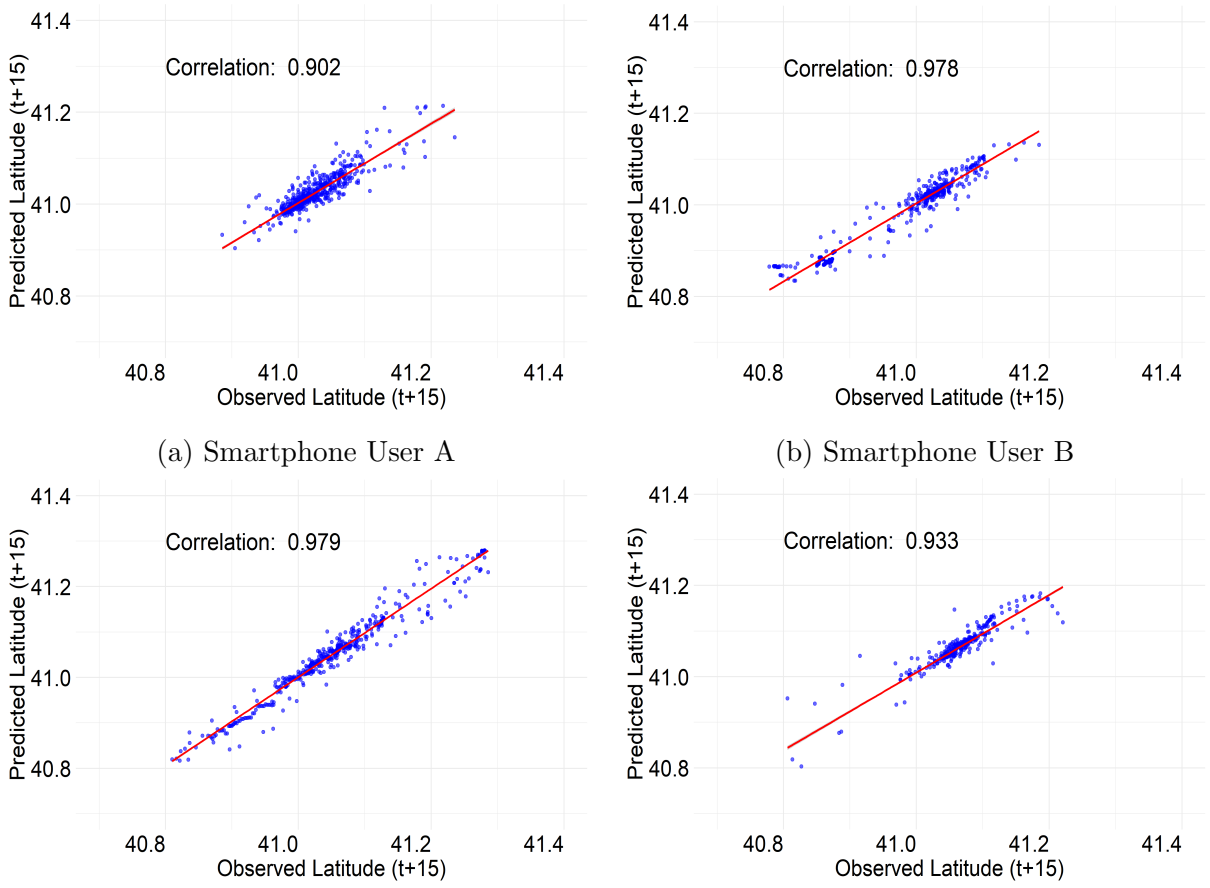
(c) Smartphone User C

(d) Smartphone User D

Figure 5.1: The general overview of observed and predicted coordinates at timestamp t+15 through the RFR models for all smartphone users. The x-axis and y-axis represent geographic coordinates expressed in degrees. The blue points denote the observed trajectories at successive fifteen-minute intervals from the testing set, and the connecting line illustrates the true movement path. The red points represent the corresponding coordinates predicted by the model.

Afterward, the predicted longitude and latitude are combined to determine the locations of smartphone users at timestamp t+15. A visual comparison of the observed and predicted locations is shown in Figure 5.1. From Figure 5.1, it is noticed that the model generally predicts locations close to the observed ones for user A. However, some locations are away from the rest of all, and these were underestimated. This is likely due to most of the users' movements are localized, and the distant location is an outlier that affects the overall spatial distribution. For user B, some northern locations are underestimated, while southern locations, particularly those outside the borders of Istanbul, are largely ignored by the model. The effect of linear interpolation is evident, as interpolated coordinates exhibit unrealistically straight transitions between sparse observed points. Most of this user's movements occur within a localized area, but the presence of a few distant observed

locations increases the overall spatial spread. For user C, the movements are sparse overall, and the model attempts to capture them closely.



(a) Smartphone User A

(b) Smartphone User B

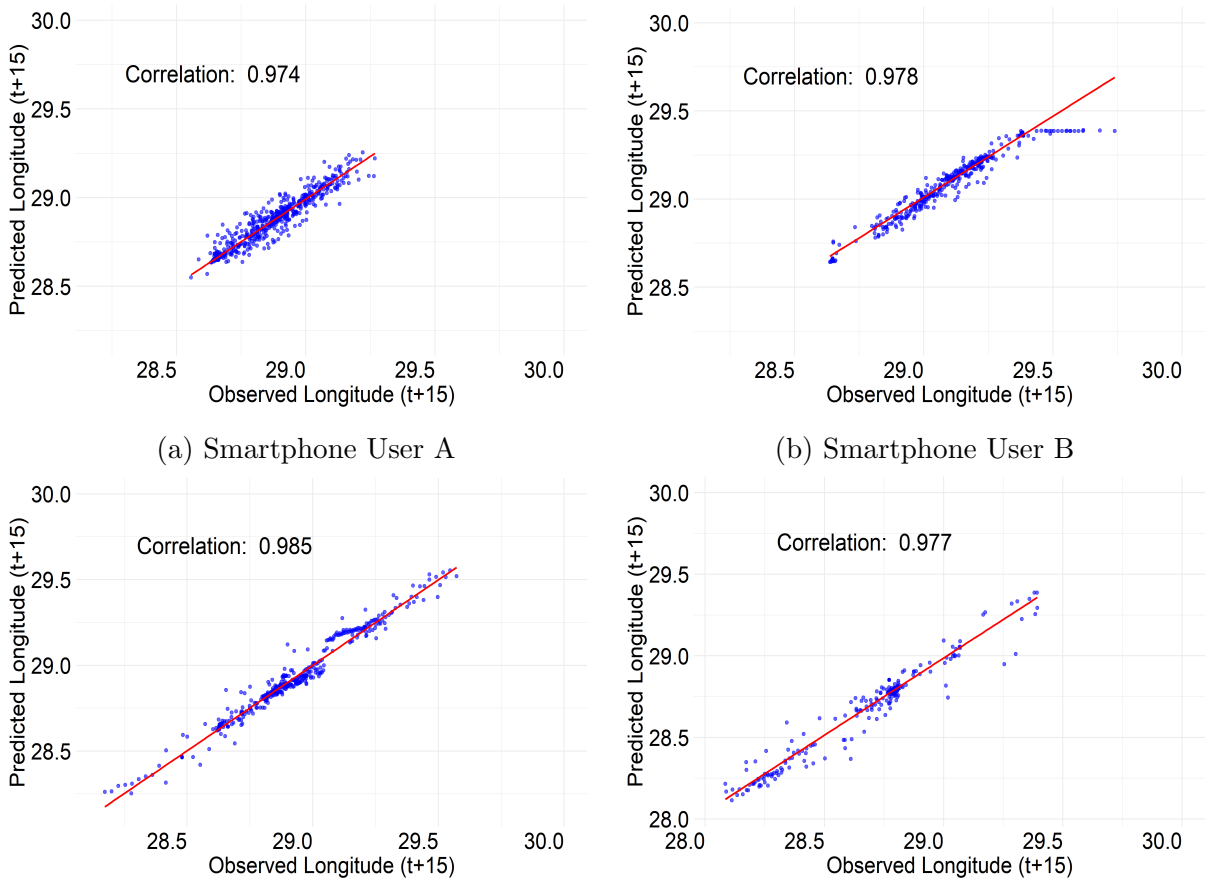
(c) Smartphone User C

(d) Smartphone User D

Figure 5.2: The relationship between the observed latitude in the testing set and the predicted latitude by the RFR models is presented, along with the correlation values for all smartphone users

However, the model may have difficulty accurately predicting locations when the movement pattern is widely dispersed. In contrast, the majority of predicted locations are close to the observed ones for user D, although a few locations in the western part of the city are distant from the rest of the locations.

To understand the relationship between the observed and predicted latitude at timestamp t+15, a scatter plot with a regression line is illustrated, as shown in Figure 5.2. From Figure 5.2, it is evident that some points deviate noticeably from the regression line in the plots for all users, indicating that the model struggled to accurately predict latitude for these specific locations. These larger deviations may be due to differences in users' movement patterns. Nonetheless, the majority of points are closely aligned with the regression line, and it indicates that the model generally predicts latitude values with high



(a) Smartphone User A

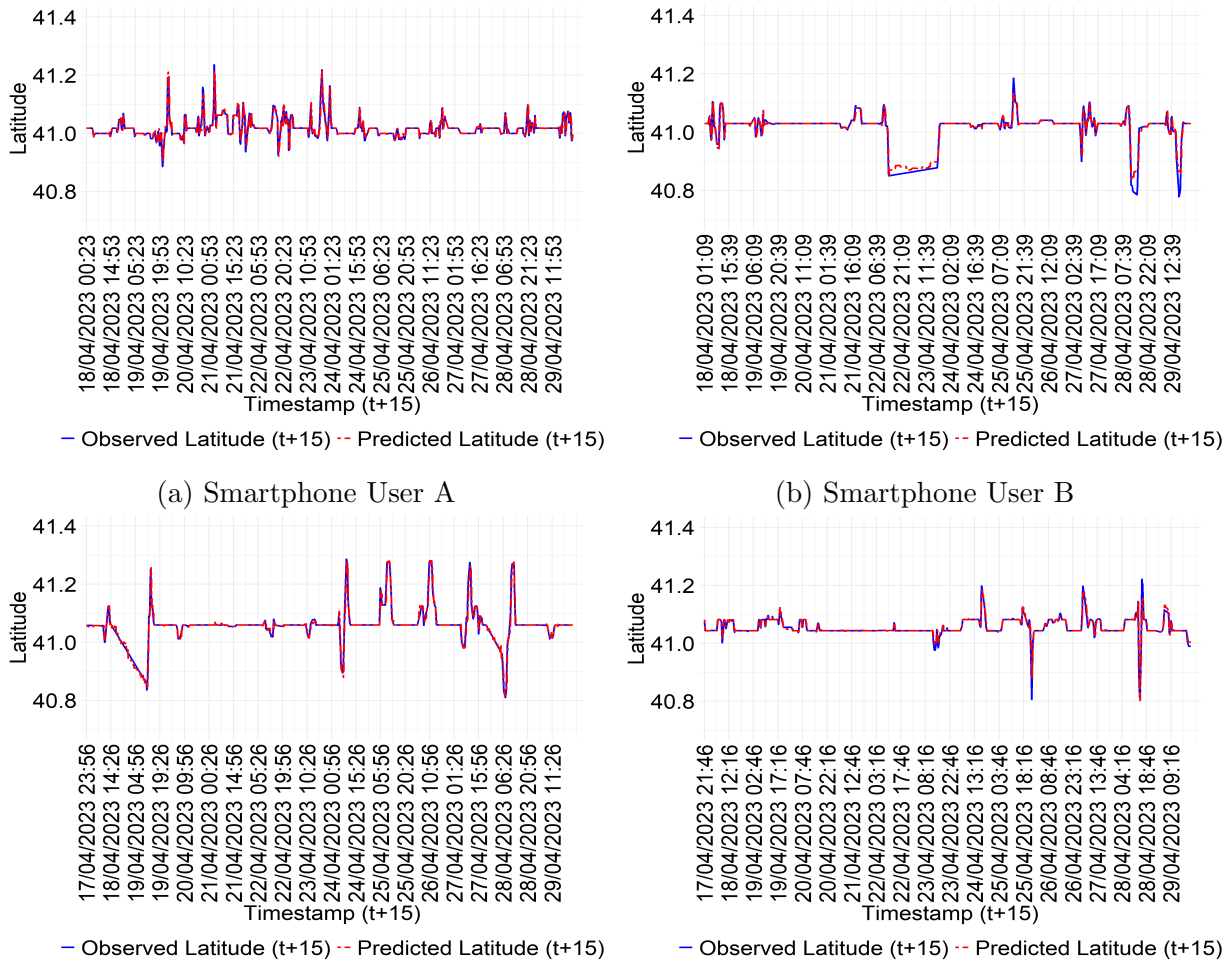
(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 5.3: The relationship between the observed longitude in the testing set and the predicted longitude by the RFR models is shown, along with the correlation values for all smartphone users.

accuracy across users. This is further supported by strong correlations between observed and predicted latitude, with values of 0.90, 0.97, 0.97, and 0.93 for users A, B, C, and D, respectively. Similarly, the scatter plot with a regression line is presented to understand the relationship between the observed and predicted longitude at timestamp t+15 as illustrated in Figure 5.3. The Figure 5.3 shows the relationship between observed and predicted longitude at timestamp t+15, along with the correlation values for the users. For users A and D, the majority of points are clustered around the regression line, although a few show noticeable deviations. Similarly trend follows by user B, but some points are completely away from the line, which might be due to the linear interpolation effect. In the case of user C, nearly all points lie very close to the line compared to other users. Also, the correlation between the observed and predicted longitude at timestamp t+15 is found to be high, with values of 0.97, 0.97, 0.98, and 0.97 for users A, B, C, and D, respectively.



(a) Smartphone User A

(b) Smartphone User B

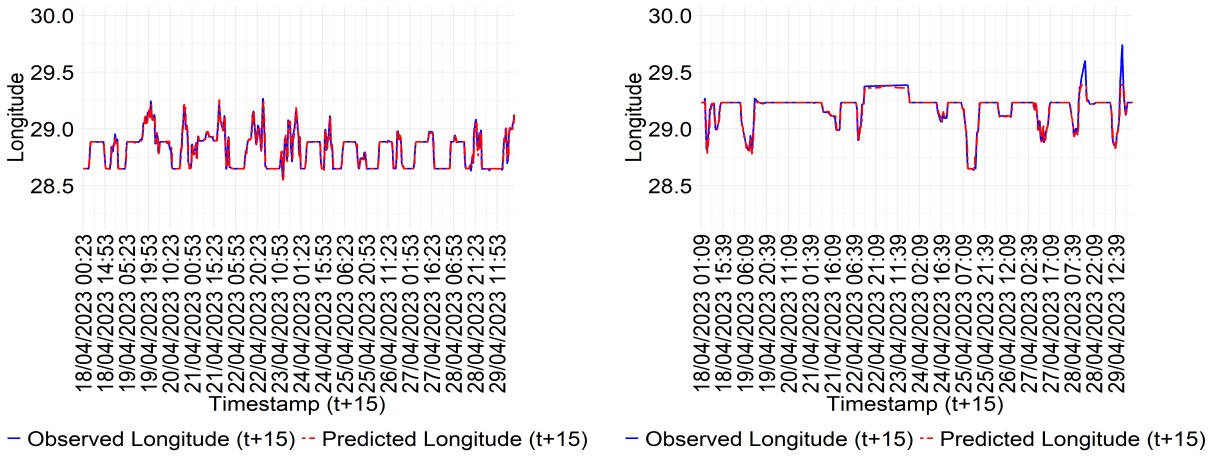
(c) Smartphone User C

(d) Smartphone User D

Figure 5.4: A visual comparison of the observed and predicted latitude by the RFR models is presented over the timestamps t+15 for all smartphone users. The observed latitudes at timestamp t+15 are interpolated and belong to the testing set.

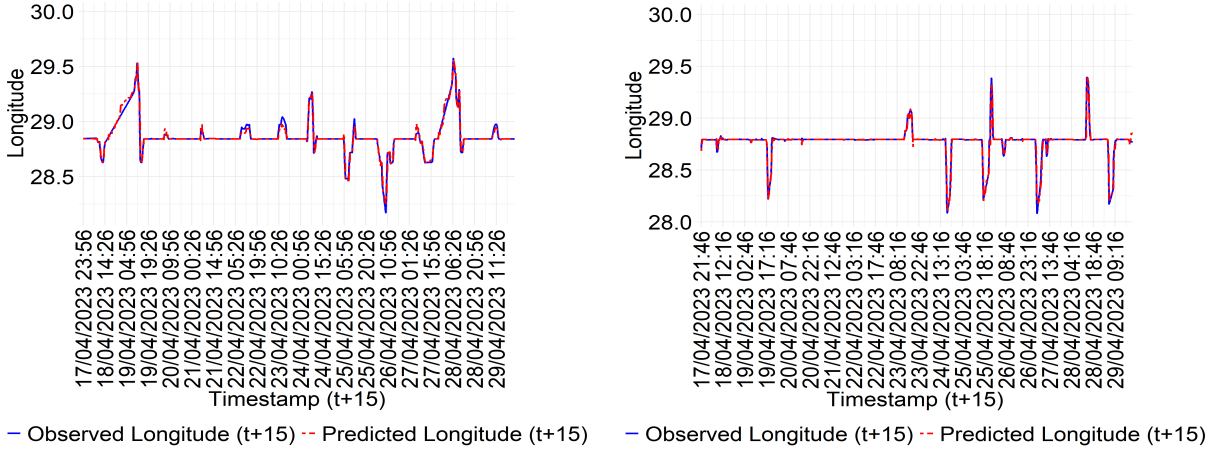
Afterward, the observed and predicted coordinates at t+15 are plotted against timestamps at t+15 to visually assess the model's performance. These plots help to identify moments when the predictions deviate significantly from the observed ones, and they highlight both strengths and weaknesses of the model. Therefore, the two separate plots are created as Figure 5.4 illustrates the latitude predictions, while Figure 5.5 shows longitude predictions w.r.t timestamps t+15. These visualizations are crucial for understanding how well the model tracks coordinate changes over time. From Figure 5.4, it is evident that the predicted latitude generally aligns closely with the observed latitude for all users, and it indicates strong model performance overall. However, small deviations are observed for users A, B, and D, where the predicted latitude does not fully capture the peaks of the observed values at certain timestamps (t+15). These spikes suggest that the model experienced difficulty in accurately reflecting sudden changes in latitude during those moments, although the

overall difference remained minimal. In contrast, the predicted latitude for user C aligns almost perfectly with the observed latitude, showing the model's effectiveness.



(a) Smartphone User A

(b) Smartphone User B



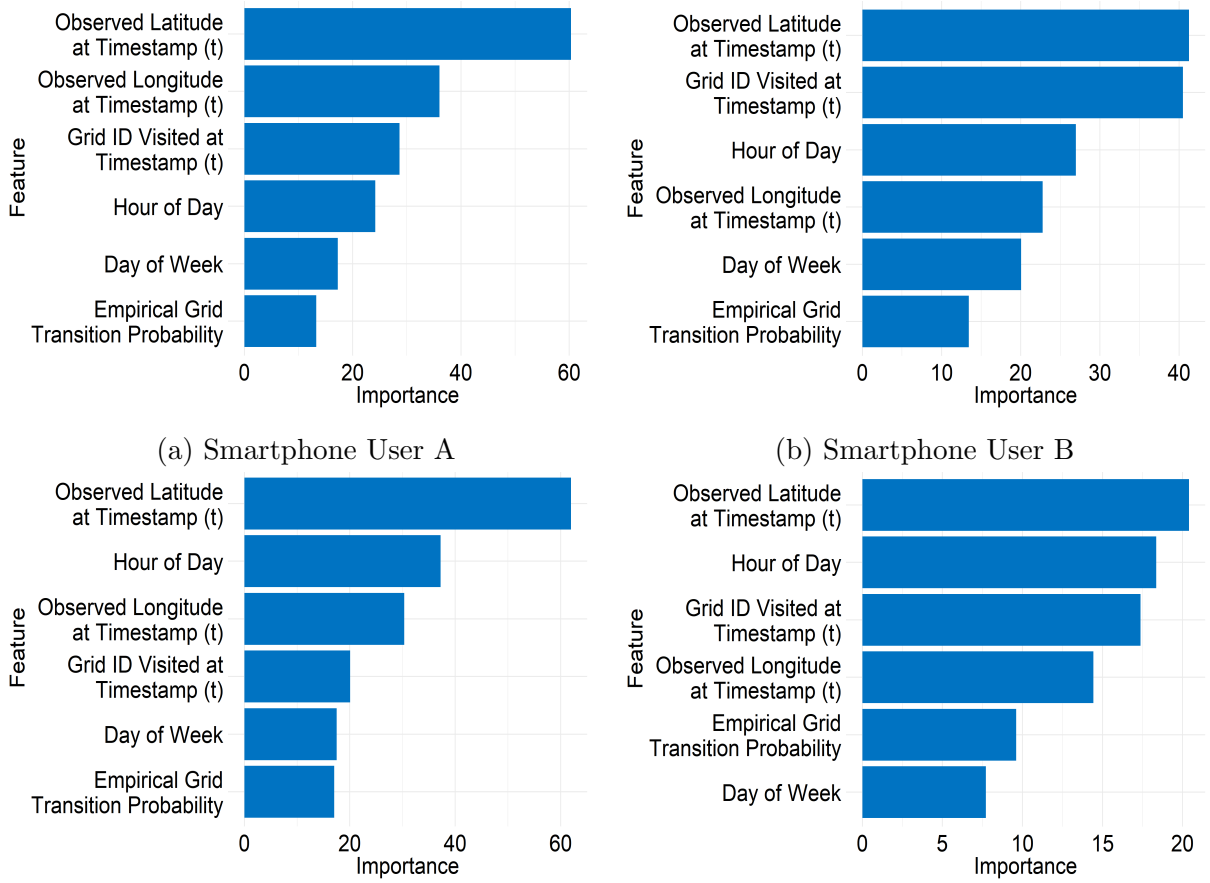
(c) Smartphone User C

(d) Smartphone User D

Figure 5.5: A visual comparison of the observed and predicted longitude by the RFR models is presented over the timestamps t+15 for all smartphone users. The observed longitudes at timestamp t+15 are interpolated and belong to the testing set.

From Figure 5.5, it is noticed that the predicted longitude closely aligns with the observed longitude for users A and D, and it indicates that the model performs well overall. However, there are noticeable spikes in the plots of user B and C, where the predicted longitude did not capture the peaks of the observed longitude at certain timestamps t+15. These deviations are more pronounced for user B than for user C. As a result, these spikes suggest moments where the model struggled to maintain accuracy, possibly due to occasional trips. To evaluate the contribution of each feature in predicting latitude and longitude by the RFR models, the permutation method is applied to the Out of Bag (OOB) samples. The OOB samples are the subset of the training set that is not used to build a tree and provides an unbiased estimate of prediction error. For each tree, the Mean Square Error (MSE) is

calculated on its OOB samples before and after permuting a feature. The average increase in MSE across all trees reflects the importance of that feature. The larger increases in MSE indicate greater significance. Therefore, the feature importance for the latitude prediction through the RFR models is illustrated in Figure 5.6. From Figure 5.6, it is seen that the



(a) Smartphone User A

(b) Smartphone User B

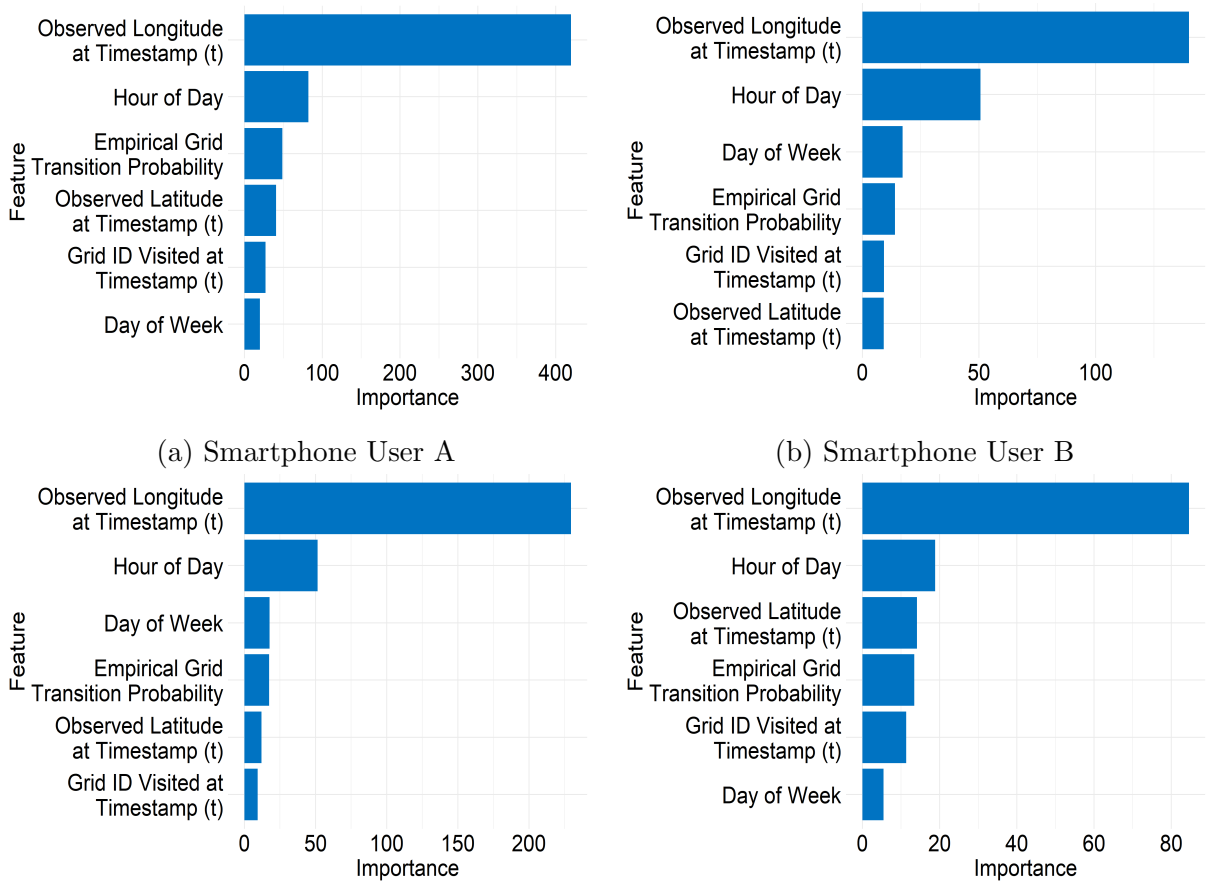
(c) Smartphone User C

(d) Smartphone User D

Figure 5.6: The feature importance for latitude prediction through the RFR models is presented for all smartphone users. The x-axis represents the importance of each feature, measured by the increase in MSE when the feature's values are permuted. The y-axis lists the features, reflecting their overall contribution to the model's predictive performance.

observed latitude at timestamp t is the most influential feature in predicting the latitude at timestamp t+15 for all users. This aligns with the Markov approximation, which states that the next location is primarily dependent on the current location. Therefore, it is reasonable that the observed latitude at time t has the strongest impact on the model's predictions. For user A, the second most important feature is the observed longitude at timestamp t, followed by the grid ID visited at the same timestamp. For user B, the grid ID at timestamp t ranks second, the hour of the day is third, and the observed longitude at timestamp t is fourth. For user C, the hour of the day is the second most significant feature, followed by the observed longitude at timestamp t in third place, and the grid ID at times-

tamp t in fourth place. For user D, the hour of day is the second most influential feature, the grid ID at timestamp t is third, and the observed longitude at timestamp t is fourth. Additionally, the day of week feature contributes more to the model than the empirical grid transition probability for all users except user D. Similarly, the key features contributing to longitude prediction at timestamp t+15 are shown in Figure 5.7. Figure 5.7 presents



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

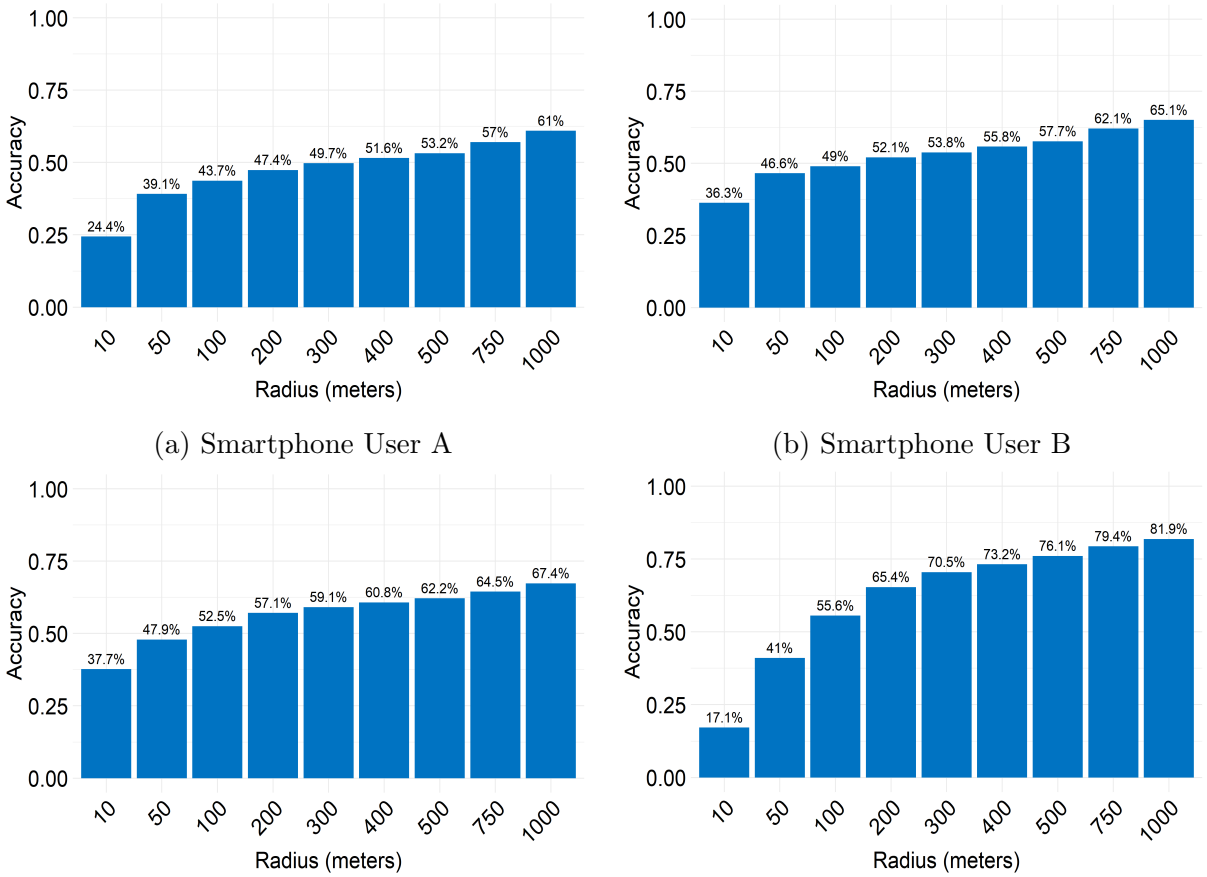
(d) Smartphone User D

Figure 5.7: The feature importance for longitude prediction through the RFR models is presented for all smartphone users. The x-axis represents the importance of each feature, measured by the increase in MSE when the feature's values are permuted. The y-axis lists the features, reflecting their overall contribution to the model's predictive performance.

the relative importance of each feature in predicting longitude at timestamp t+15. For all users, the observed longitude at timestamp t is the most influential predictor, which aligns with the Markov approximation, as the next location is strongly dependent on the current one. The hour of day is the second most important feature across all users. For users B and C, the day of week and the empirical grid transition probability are ranked third and fourth, respectively. In the case of user A, the empirical grid transition probability ranks third, followed by the observed latitude at timestamp t. Conversely, the observed latitude at timestamp t is ranked third for user D, while the empirical grid transition probability is

ranked fourth. Although the contribution of the grid ID visited at timestamp t is relatively lower, its importance varies among users. Overall, the observed coordinates at timestamp t are the most significant features for predicting future coordinates at t+15, and are consistent with the Markov approximation. The spatial context is captured by grid ID visited at timestamp t, and empirical grid transition probabilities, and the temporal features such as hour of day and day of week, provide additional predictive power and further refine the prediction by capturing regular movement patterns and routine behaviors. Further details on the local variability of features are provided in Appendix B.2.

The accuracy of the RFR models was initially assessed using standard metrics such as MAE and RMSE, which provide a general sense of how close the predicted coordinates are to the observed ones. However, these measures alone are insufficient to fully evaluate model performance in the context of mobility modeling. To address this limitation, an approach based on the Haversine distance formula is introduced and referred to as Accuracy Within Radius (AWR) in this chapter. The AWR method evaluates prediction accuracy across different radius thresholds and indicates how frequently the predicted locations fall within a specified distance of the observed locations, and provides a more intuitive and spatially relevant measure of how well the model predicts locations. To assess the model's ability to predict locations, multiple radius thresholds such as 10, 50, 100, 200, 300, 400, 500, 750, and 1000 meters were considered, and the results for all users are shown in Figure 5.8. The results in Figure 5.8 illustrate the RFR model's prediction accuracy with respect to different radius thresholds. At the fine scale between 10 and 100 meters, the prediction accuracy is relatively low for most users. At 10 meters, user C achieves the highest accuracy of 37.7%, followed by user B at 36.3%, user A at 24.4%, and user D at 17.1%. As the radius expands to 50 meters, the accuracy improves with user C reaching 47.9%, user B at 46.6%, user A at 39.1%, and user D at 41.0%. At 100 meters, user C maintains the highest accuracy at 52.5%, followed by user B at 49.0%, user D at 55.6%, and user A at 43.7%. These results indicate that user C's movements remain relatively predictable at fine-grained scales, while the other users exhibit greater variability, making short-distance predictions more challenging. At the medium scale, between 200 and 400 meters, the model shows marked improvements for all users. At 200 meters, user C achieves 57.1%, user D at 65.4%, user B at 52.1%, and user A at 47.4%. At 300 meters, the accuracies increase further as user C reaches 59.1%, user D at 70.5%, user B at 53.8%, and user A at 49.7%. By 400 meters, user C achieves 60.8%, user D 73.2%, user B 55.8%, and user A at



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

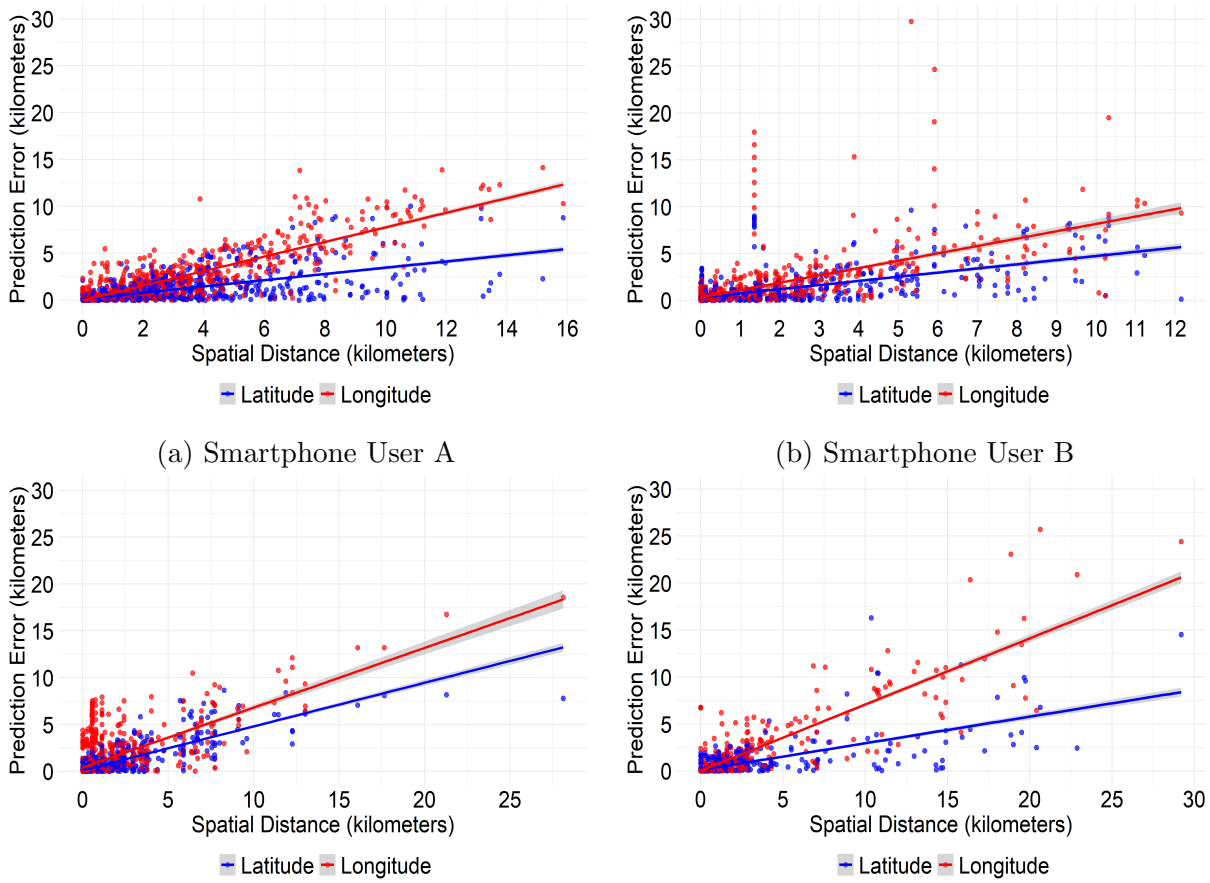
(d) Smartphone User D

Figure 5.8: The comparison of prediction accuracies across radius thresholds for all smartphone users. The y-axis represents accuracy, defined as the proportion of predicted coordinates at timestamp t+15 through RFR models that fall within the specified radius of the observed coordinates at the same timestamp.

51.6%. These results highlight that the model effectively captures mobility patterns at a neighborhood or regional scale, particularly for users C and D. At the large scale between 500 and 1000 meters, the model shows strong performance across all users. At 500 meters, user C reaches 62.2%, user D at 76.1%, user B at 57.7%, and user A at 53.2%. At 750 meters, the accuracies increase to 64.5% for user C, 79.4% for user D, 62.1% for user B, and 57.0% for user A. At 1000 meters, the model achieves its highest performance, with user C at 67.4%, user D at 81.9%, user B at 65.1%, and user A at 60.9%. These findings highlight that the model effectively captures broad mobility trends at the city scale, where predicted locations generally fall within one kilometer of the observed locations.

The results confirm that the observed coordinates at timestamp t play a significant role in predicting the coordinates at timestamp t+15. To better evaluate the model's performance, it is useful to compare distances in kilometers between two pairs of points; (1) the observed coordinates at timestamp t and the observed coordinates at timestamp t+15,

reflecting the Markov approximation that the coordinates at t+15 depend on those at timestamp t; and (2) the observed coordinates at timestamp t+15 and the predicted coordinates at t+15 by the RFR models for the same timestamp. This comparison provides a quantitative measure of how accurately the model predicts the location fifteen minutes after timestamp t relative to the actual movement. The comparison between the prediction error and spatial distance is separately illustrated for latitude and longitude, and it is depicted in Figure 5.9. As shown in Figure 5.9, the prediction error increases as the



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 5.9: The prediction errors versus spatial distances are shown for all smartphone users. Each point represents a single prediction, and the x-axis shows the spatial distance traveled between the timestamp t and the timestamp t+15 in kilometers, and the y-axis shows the prediction error in kilometers. The prediction errors are calculated between the observed coordinates and the predicted coordinates at timestamp t+15 by the RFR models. The smoothed lines represent linear trends for each error type.

spatial distance between the observed coordinates at timestamp t and t+15 raises for all users. However, the magnitude of this error varies across individuals. Notably, the error in latitude remains consistently lower than that in longitude by a substantial margin for all users, although both latitude and longitude exhibit a general trend of increasing prediction error with greater spatial separation. This difference can be explained by the fact

that longitude measures east-west position, while latitude measures north-south position. In Istanbul, the users' movements tend to be more dispersed along the east-west direction than along the north-south direction, resulting in larger prediction errors for longitude compared to latitude.

Secondly, the SVR is also employed with the RBF kernel to predict the locations of users. A Markov-based feature engineering strategy is similarly implemented using the same set of features in the SVR model as it was applied in the RFR model. The SVR models were separately trained on the training set for the latitude and longitude predictions. A grid search with rolling window ahead cross-validation is employed to identify the optimal values of parameters C and sigma. Similar to the RFR model, the original training set is split based on the temporal sequence, ensuring that the temporal order of observations is preserved. A five-fold strategy was applied, and each fold includes the first 80% of the observations, and the subsequent 4% of the testing set is used for validation. The C is the cost parameter that controls the trade-off between maximizing error and model complexity, and it was evaluated using a range of values such as 1, 10, and 100. A smaller C value allows for a more generalized model by placing less emphasis on minimizing errors, which can lead to a higher bias, but it is better for generalization to unseen data. Conversely, a larger C focuses more on reducing errors in the training set and making the model more complex. It potentially leads to overfitting, although it performs well on the training set, but it struggles with unseen data. The sigma parameter determines the width of the RBF kernel, which affects how the model captures patterns in the dataset. The parameter was investigated with numerous values such as 0.001, 0.01, 0.1, and 1, and this parameter plays an essential role in defining how the model perceives the distribution of sample points in the feature space. A smaller sigma value leads to more localized influence, and it allows the model to capture finer details and fit the training set more closely. However, this can increase the risk of overfitting, where the model learns noise instead of general patterns. Conversely, a larger sigma value results in a smoother function, and it reduces sensitivity to noise and improves generalization. Therefore, the grid search approach identified the optimal set of parameters separately for latitude and longitude, and the results of this systematic evaluation are detailed in Table 5.3. It is shown in Table 5.3 that the unique combination of parameters produces different levels of RMSE values for longitude and latitude. To optimize the model's performance, the parameter combinations were selected by prioritizing those that yielded the lowest RMSE for each user. Then, the SVR model

Table 5.3: The optimal parameter combinations were identified by the grid search approach with a rolling window ahead cross-validation for all smartphone users. The reported RMSE values represent the overall SVR model's performance under the selected parameters.

	Parameters		**Latitude**	**Longitude**
Smartphone User	**C**	**Sigma**	**RMSE**	**RMSE**
A	1	0.001	0.0165	0.0373
	1	0.01	0.0171	0.0400
	1	0.1	0.0210	0.0480
	1	1	0.0313	0.0649
	10	0.001	0.0160	0.0360
	10	0.01	0.0161	0.0378
	10	0.1	0.0192	0.0442
	10	1	0.0301	0.0690
	100	0.001	0.0157	0.0360
	100	0.01	0.0156	0.0372
	100	0.1	0.0195	0.0493
	100	1	0.0317	0.0782
B	1	0.001	0.0121	0.0249
	1	0.01	0.0120	0.0230
	1	0.1	0.0126	0.0229
	1	1	0.0162	0.0486
	10	0.001	0.0121	0.0232
	10	0.01	0.0121	0.0221
	10	0.1	0.0128	0.0238
	10	1	0.0172	0.0493
	100	0.001	0.0120	0.0228
	100	0.01	0.0122	0.0219
	100	0.1	0.0132	0.0268
	100	1	0.0203	0.0521
C	1	0.001	0.0164	0.0291
	1	0.01	0.0164	0.0287
	1	0.1	0.0167	0.0296
	1	1	0.0370	0.0757
	10	0.001	0.0160	0.0282
	10	0.01	0.0158	0.0277
	10	0.1	0.0180	0.0319
	10	1	0.0372	0.0760
	100	0.001	0.0159	0.0284
	100	0.01	0.0159	0.0272
	100	0.1	0.0216	0.0343
	100	1	0.0424	0.0788
D	1	0.001	0.0135	0.0429
	1	0.01	0.0131	0.0414
	1	0.1	0.0134	0.0421
	1	1	0.0184	0.0714
	10	0.001	0.0133	0.0406
	10	0.01	0.0133	0.0405
	10	0.1	0.0137	0.0430
	10	1	0.0195	0.0707
	100	0.001	0.0130	0.0397
	100	0.01	0.0138	0.0418
	100	0.1	0.0151	0.0558
	100	1	0.0231	0.0771

was trained using an optimal set of parameters on the training set and followed by validation using the testing set. The same procedure was adopted across all users, and the detailed findings are described in Table 5.4. The Table 5.4 provides the accuracies of the

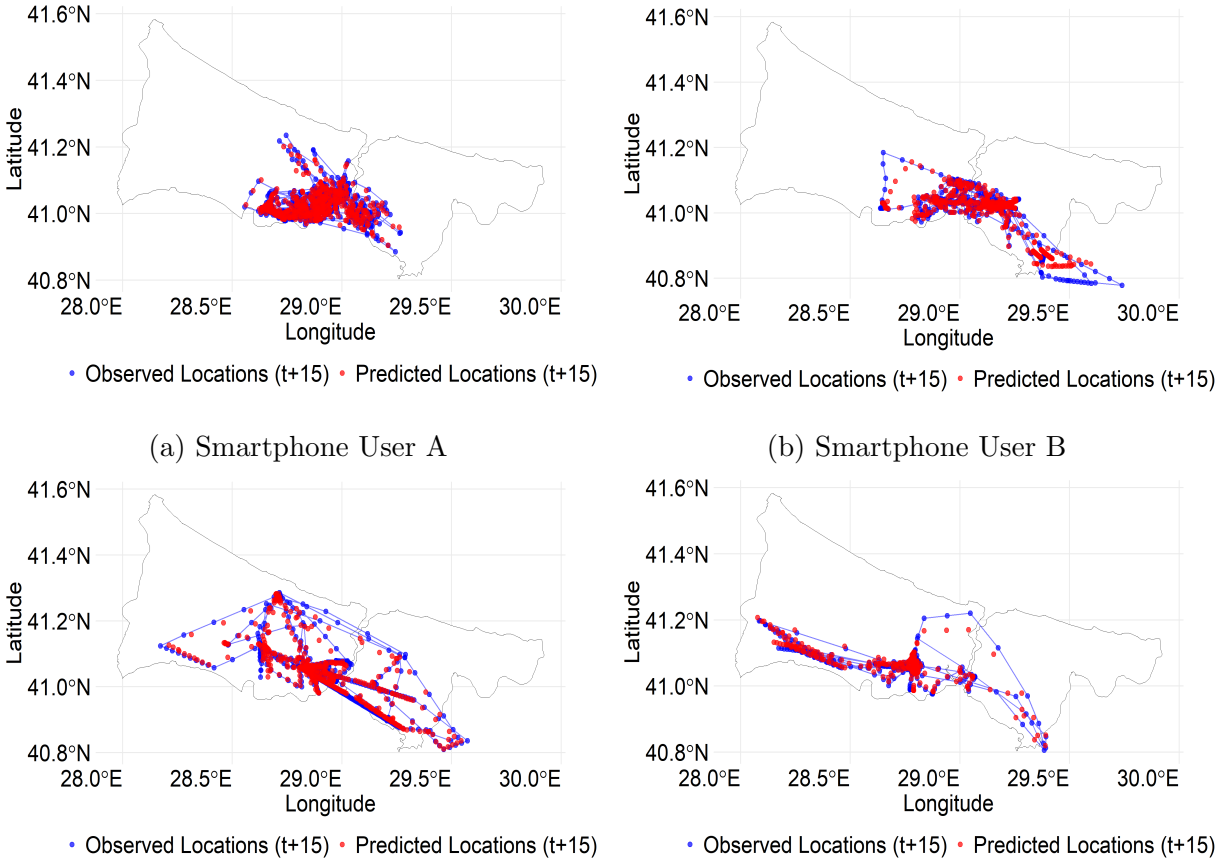
Table 5.4: The selected SVR models with optimal parameters identified through the grid search with a rolling-window ahead cross-validation are described for smartphone users. Each model was trained on the first 80% of the observations and then validated by predicting the coordinates at timestamp t+15, which were compared against the observed coordinates at timestamp t+15 in the remaining 20% of the testing set.

Smartphone User	**Coordinate**	**C**	**Sigma**	**MAE**	**RMSE**	**R-squared**
A	Latitude	100	0.01	0.0068	0.0136	0.8227
	Longitude	10	0.001	0.0200	0.0344	0.9444
B	Latitude	1	0.01	0.0055	0.0125	0.9599
	Longitude	100	0.01	0.0137	0.0242	0.9733
C	Latitude	10	0.01	0.0061	0.0125	0.9580
	Longitude	100	0.01	0.0135	0.0235	0.9769
D	Latitude	100	0.001	0.0054	0.0122	0.8477
	Longitude	100	0.001	0.0134	0.0340	0.9392

SVR models for predicting the latitude and longitude at timestamp t+15 for users. For user A, the model explains 82.27% of the variation in latitude prediction with a MAE of 0.0068 and an RMSE of 0.0136. This indicates that the model predicted with a moderate margin of error. The model accounts for 94.44% of the variation with a MAE of 0.0200 and RMSE of 0.0344 for longitude prediction and indicates strong predictive accuracy, although it has a higher error compared to latitude. For user B, the model continues to perform consistently, and it explains 95.99% of the variation in latitude with a MAE of 0.0055 and RMSE of 0.0125. The longitude prediction for this user has an R-squared of 97.33% with a MAE of 0.0137 and RMSE of 0.0242, and it indicates a smaller deviation in predictions compared to latitude. For user C, the model achieves good performance, and it explains 95.80% of the variation in latitude prediction with a MAE of 0.0061 and RMSE of 0.0125. The longitude prediction is also more precise with an R-squared of 97.69%, MAE of 0.0135, and RMSE of 0.0235, and it highlights the model's high capability to predict longitude accurately for this user. For user D, the model maintains reasonable accuracy, and it explains 84.77% of the variation in latitude prediction with a MAE of 0.0054 and RMSE of 0.0122. While the longitude prediction is slightly less precise than latitude, it explains a variation of 93.92% with a MAE of 0.0134 and RMSE of 0.0340. Overall, the R-squared values remain above 82% for latitude predictions and above 93% for longitude predictions for all users. This confirms that the models explain a substantial proportion

of the variability in latitude and longitude predictions, effectively capturing the movement patterns of users.

Afterward, the predicted longitude and latitude are combined to determine the locations of users at timestamp t+15. A visual comparison of the observed and predicted locations is shown in Figure 5.10. From Figure 5.10, it can be observed that the model generally



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

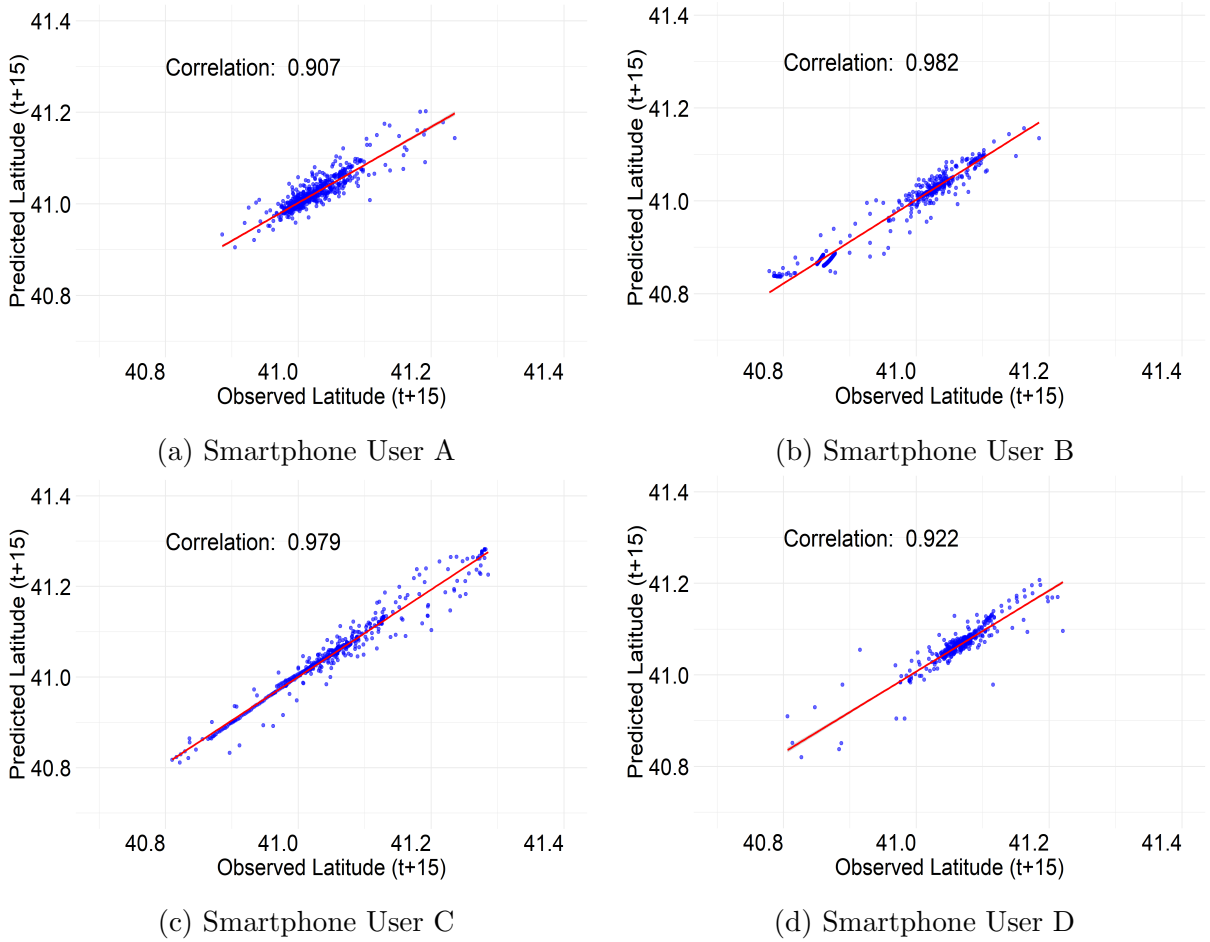
(d) Smartphone User D

Figure 5.10: The general overview of observed and predicted coordinates at timestamp t+15 through the SVR models for all smartphone users. The x-axis and y-axis represent geographic coordinates expressed in degrees. The blue points denote the observed trajectories at successive fifteen-minute intervals from the testing set, and the connecting line illustrates the true movement path. The red points represent the corresponding coordinates predicted by the model.

predicts locations close to the observed locations for user A. However, a few instances appear in the western and southeastern regions of the city where the predicted locations deviate noticeably from the actual ones. Overall, this user's movements remain relatively localized at timestamp t+15. The movement of user B also appears localized within the city. However, some distant observed locations contribute to an increased spatial spread, particularly beyond the southeastern border of Istanbul. In these regions, the model attempts to predict locations slightly outside the city boundary, but it was unsuccessful in

accurately predicting, resulting in noticeable deviations from the true positions. Additionally, there are locations in the northern part of the city where the model underestimates. For user C, the movements are generally more dispersed, and the model attempts to predict these locations with reasonable accuracy. In the western part of Istanbul, the model encounters difficulty when the movement pattern becomes highly scattered, leading to reduced prediction precision. A similar pattern is noticed for user D, where certain locations in the western region are challenging for the model to predict accurately, as these points are spatially separated from the main cluster of movement trajectories.

The relationship between the observed and predicted latitude at timestamp t+15 is illustrated in a scatter plot with a regression line, as shown in Figure 5.11. Figure 5.11 illus-



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

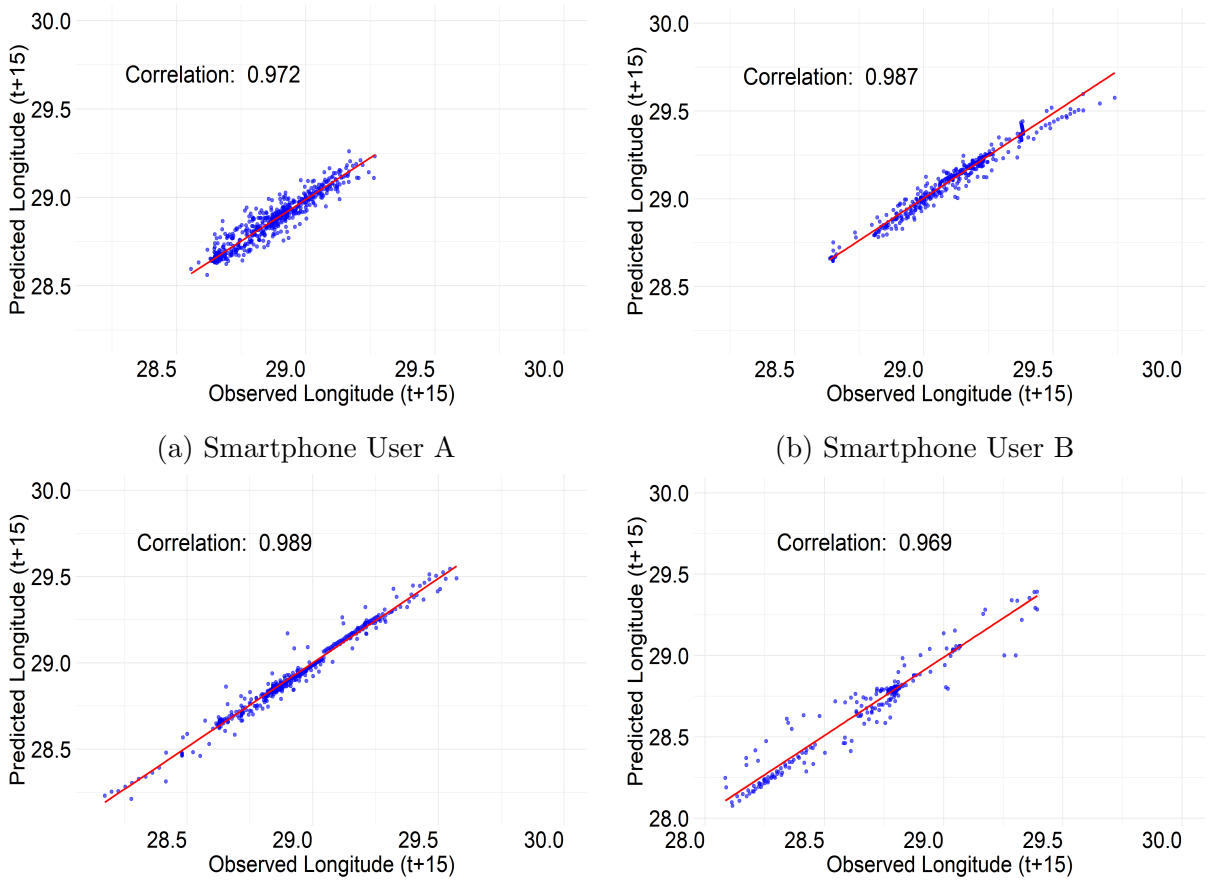
(d) Smartphone User D

Figure 5.11: The relationship between the observed latitude at timestamp t+15 from the testing set and the predicted latitude at timestamp t+15 by the SVR models is presented, along with the correlation values for all smartphone users.

trates the relationship between the observed and predicted latitude at timestamp t+15, along with the corresponding correlation values for the users. Overall, the points are generally clustered around the regression line, indicating strong predictive performance; however, some points deviate from the line, creating a noticeable spread across all users.

Among them, user C exhibits the least dispersion, suggesting more consistent predictions. The deviations are particularly evident within the latitude range of approximately 41.1 to 41.2 for all users, though they are more pronounced for users A, C, and D. In the case of user D, the spread is greater compared to other users, reflecting higher variability in the model's prediction accuracy. Furthermore, the correlation between the observed and predicted latitude is found to be high, with values of 0.90, 0.98, 0.97, and 0.92 for users A, B, C, and D, respectively.

Similarly, the relationship between the observed and predicted longitude at timestamp t+15 is represented in a scatter plot with a regression line, as depicted in Figure 5.12. It can be seen in Figure 5.12 that the relationship between the observed and predicted



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

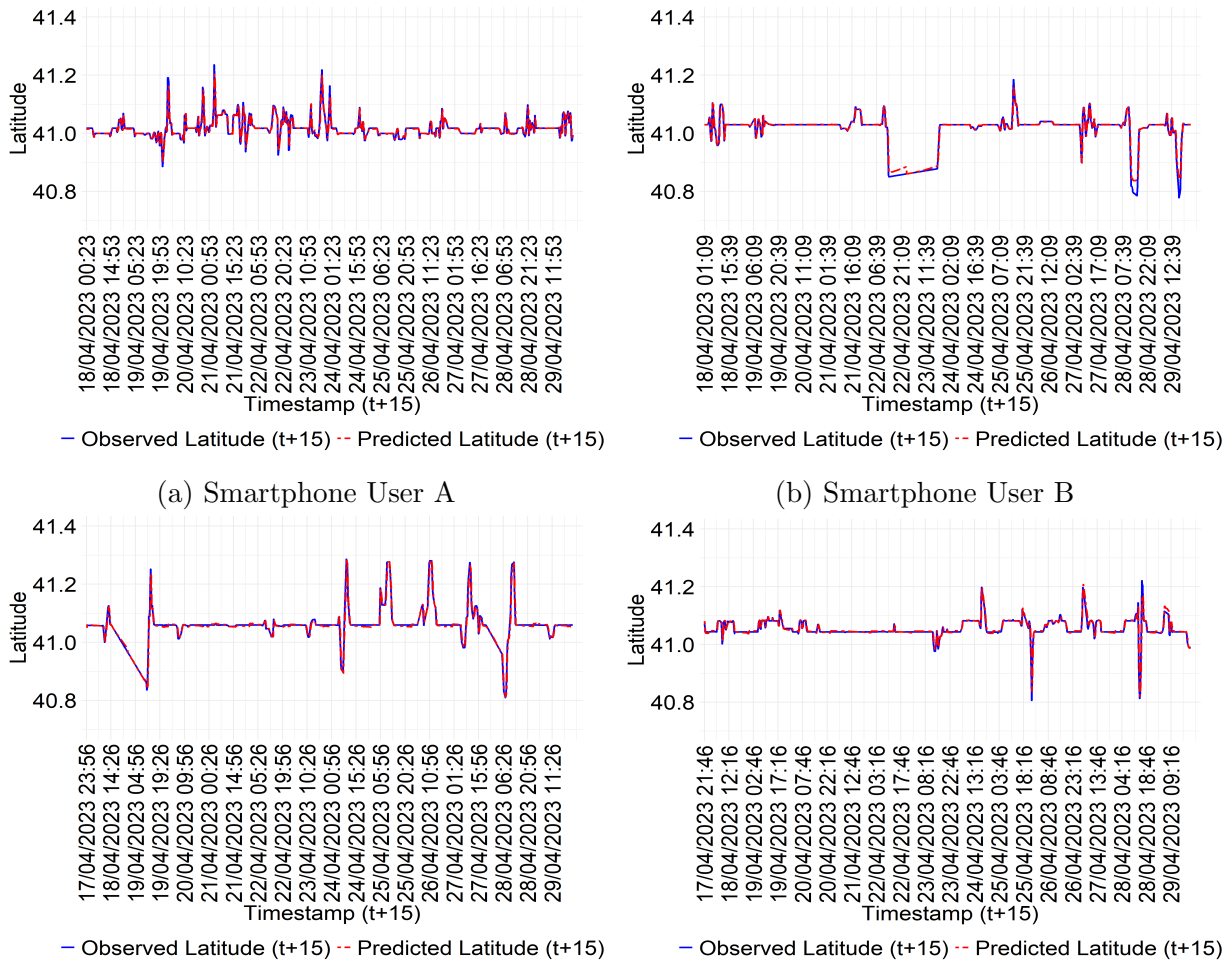
(d) Smartphone User D

Figure 5.12: The relationship between the observed longitude at timestamp t+15 from the testing set and the predicted longitude at timestamp t+15 by the SVR models is shown, along with the correlation values for all smartphone users.

longitude at timestamp t+15, along with the corresponding correlation values for each user. For users A and D, most of the points are generally clustered around the regression line. However, several points deviate from the line, which increases the overall spread of the data. The deviations appear more pronounced in the case of user D compared to user

A, which may be attributed to more heterogeneous movement patterns. For user C, the points are tightly clustered around the line, which indicates a strong agreement between observed and predicted values. In the case of user B, the points are also concentrated near the line but with a slightly wider spread, and a few points deviate more noticeably from the linear trend. Furthermore, the correlation between the observed and predicted longitude is found to be high, with values of 0.97, 0.98, 0.98, and 0.96 for users A, B, C, and D, respectively.

Furthermore, the predicted coordinates at timestamp t+15 are plotted against timestamps t+15 to visually assess the performance of models for each user. These plots help to identify moments when the predictions deviate significantly from the observed coordinates, and they also highlight both the strengths and weaknesses of the models. The Figure 5.13 illustrates the observed and predicted latitude values plotted against timestamp t+15, while Figure 5.14 shows the observed and predicted longitude values for the same timestamp. The Figure 5.13 illustrates that the predicted latitude closely aligns with the observed latitude for all users, and it indicates that the model performs well. However, there are a few spikes that appear in all users' plots except user C, where the predicted latitude fails to accurately capture the peak of the observed latitude at the specific timestamp t+15. In the case of user B, the spikes are highly visible compared to the other users. This spike suggests a moment where the model struggled to maintain accuracy, possibly due to sudden movement. In general, the models effectively capture the broad trend of latitude changes, although they slightly underestimate the exact peak values. The Figure 5.14 provides a comparison between the observed and predicted longitude over timestamps t+15 for all users. The predicted longitudes closely align with the observed longitudes, and this indicates strong model performance for all users. Also, there are noticeable spikes in the plots of user A and B, where the predicted longitude deviates slightly from the observed longitude. This pattern suggests that the models struggled to identify the mobility pattern during sudden changes in movement, and overall, the trend is well captured. To better understand the contribution of different predictors to the latitude and longitude prediction, the permutation-based feature importance is derived. Specifically, the RMSE loss was used to quantify the decrease in model performance when the values of a given feature were permuted, breaking its relationship with the target variable. A larger increase in RMSE indicates a higher importance of that feature. The feature importance analysis for latitude and longitude prediction through SVR models is presented in Figures 5.15.



(a) Smartphone User A

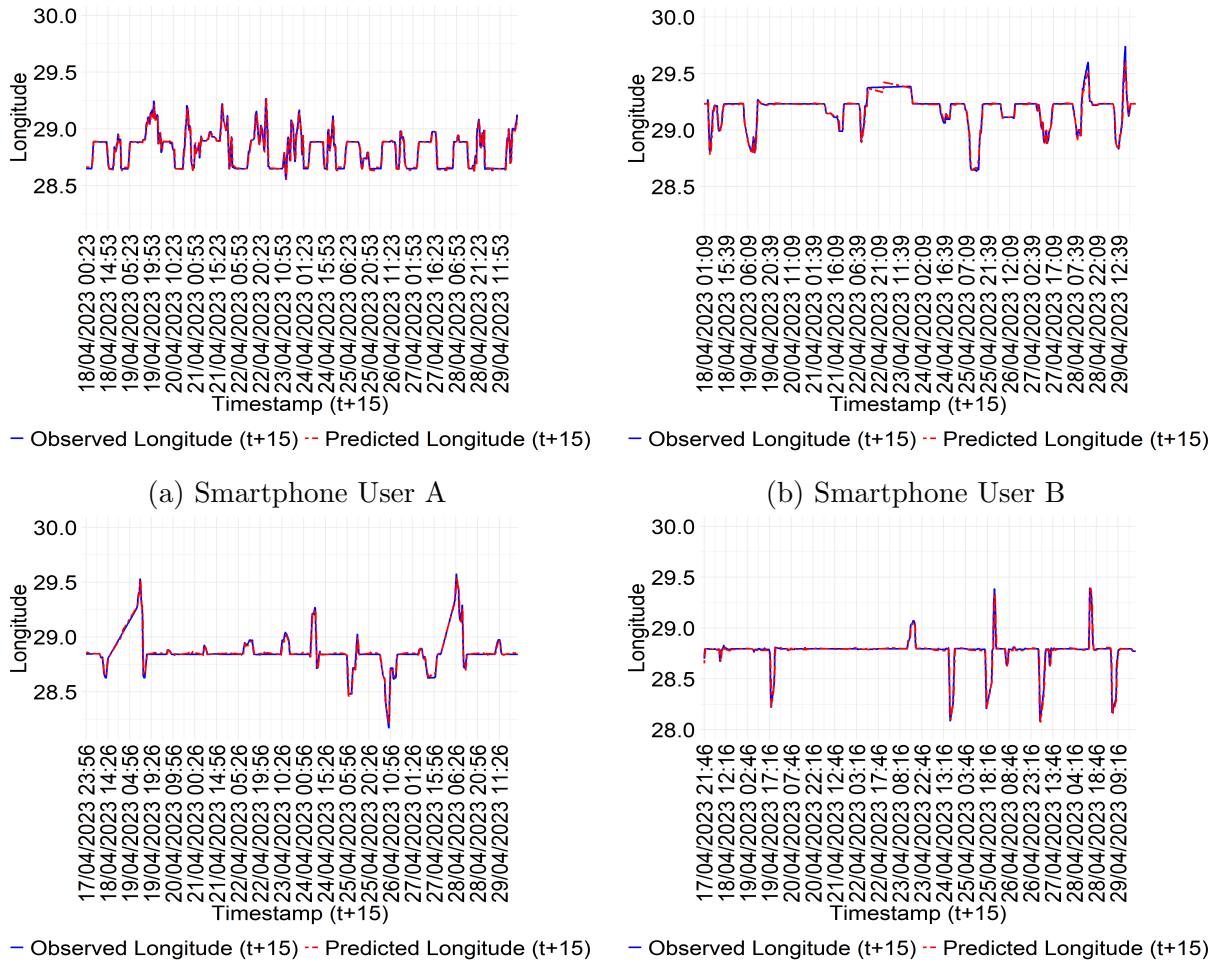
(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 5.13: A visual comparison of the observed and predicted latitude by the SVR models is presented over the timestamps t+15 for all smartphone users. The observed latitudes at timestamp t+15 are obtained from the testing set.

and 5.16. It can be observed in Figure 5.15 that the observed latitude at timestamp t is the most influential feature for predicting latitude at timestamp t+15, consistent with the Markov approximation across all users. The grid ID visited at timestamp t emerges as the second most important feature for users A, B, and C, while it is the third important predictor for user D. The observed longitude at timestamp t is the second most influential feature for user D, ranks third for users A and C, and is positioned fourth for user B. Temporal features such as hour of day and day of week, as well as the empirical grid transition probability, also contribute meaningfully across all users, although their relative importance varies among individuals. The Figure 5.16 shows that the observed longitude at timestamp t is the most dominant feature for predicting longitude at timestamp t+15, which is consistent with the Markov approximation across all users. For users A, B, and D, the hour of the day and the grid ID visited at timestamp t are ranked as the second



(a) Smartphone User A

(b) Smartphone User B

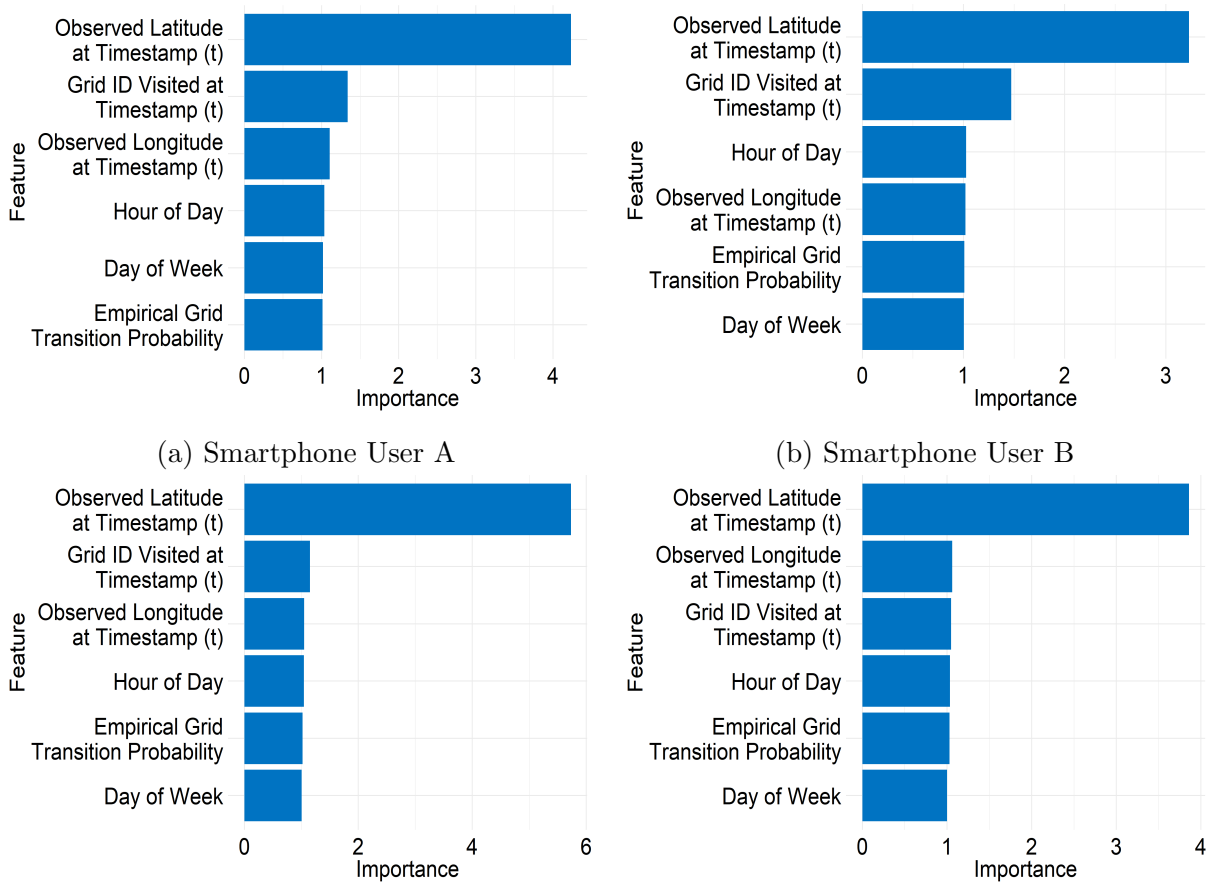
(c) Smartphone User C

(d) Smartphone User D

Figure 5.14: A visual comparison of the observed and predicted longitude by the SVR models is presented over the timestamps t+15 for all smartphone users. The observed longitudes at timestamp t+15 are obtained from the testing set.

and third most influential features, respectively. In the case of user C, the grid ID visited at timestamp t and the observed latitude at timestamp t are ranked second and third. The remaining features also contribute to longitude prediction, although their relative importance and ranking differ among users. In general, the integration of both spatial and temporal features enhances the model's ability to understand user mobility, as previous coordinates with respect to timestamp t provide the foundation and temporal rhythms, helping to refine the predictions. Further details on the local variability of features are provided in Appendix B.2.

Next, the AWR was computed between the observed and predicted coordinates at timestamp t+15 using the same set of radius thresholds employed to evaluate the RFR model's accuracy. The resulting prediction accuracies across different radius thresholds are presented in Figure 5.17. The Figure 5.17 depicts the SVR model's prediction accuracy with



(a) Smartphone User A

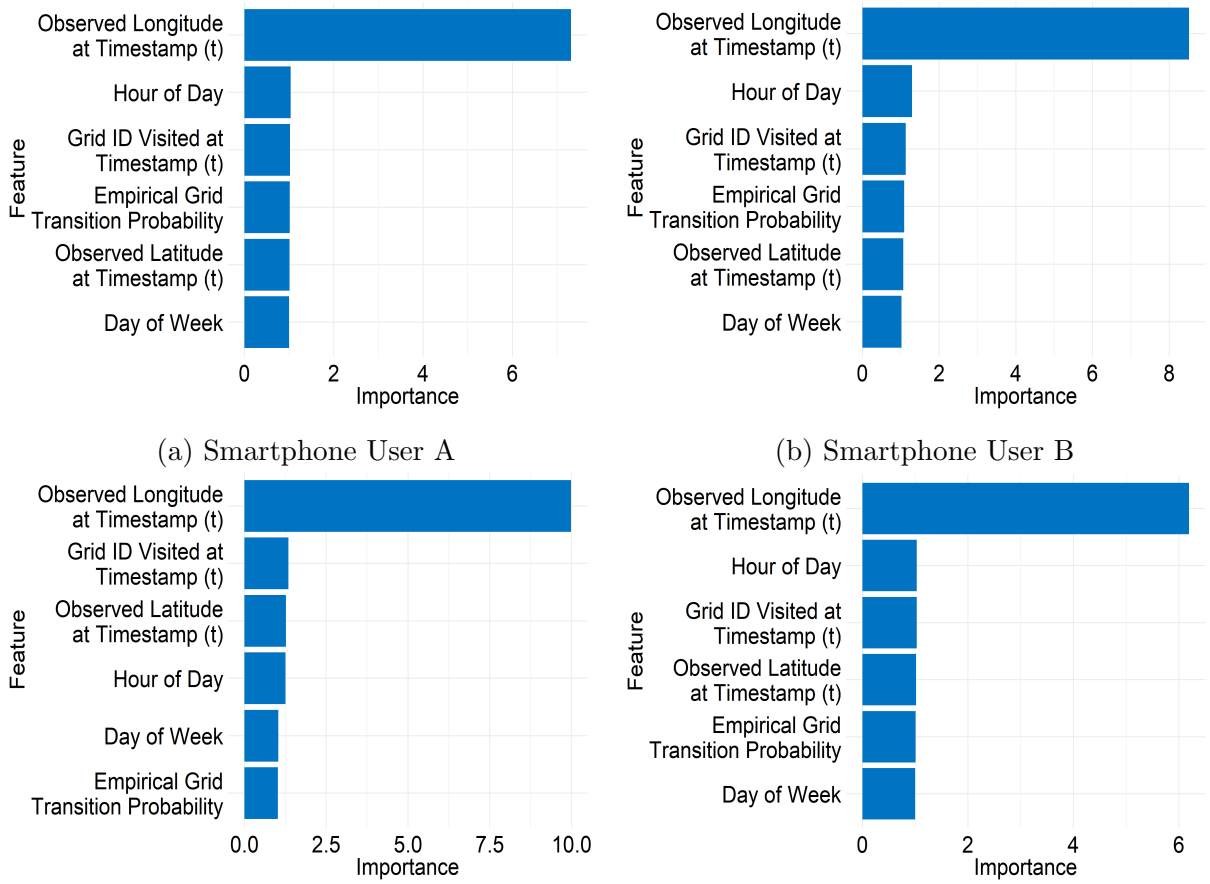
(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 5.15: The feature importance in latitude prediction through the SVR models is presented for all smartphone users. The x-axis represents feature importance based on the RMSE, which reflects how much the model's performance drops when a feature's values are shuffled. The y-axis shows the overall contribution of each feature to the model's predictive performance.

respect to the radius threshold. At the fine scale between 10 and 100 meters, the performance remains very low for all users. At 10 meters, user A achieves an accuracy of only 0.1%, while users B, C, and D remain near 0.00%. At 50 meters, user B reaches 2.2%, followed by user A at 1.22%, user D at 1%, and user C at 0.9%. By 100 meters, user B increases to 5.6%, user A reaches 5%, user C achieves 1.6%, and user D records 1.6%. These values indicate that the model struggles to provide meaningful predictive performance at very localized scales. At the medium scale between 200 and 400 meters, the performance improves notably for all users. At 200 meters, user B achieves 15.4%, followed by user A at 13.83%, user C at 6.86%, and user D at 5.83%. At 300 meters, user B reaches 26.66%, user A at 23.91%, user D at 15.04%, and user C at 13.54%. By 400 meters, user B improves to 38.68%, user A to 34.00%, user D to 27.13%, and user C to 20.40%. These results suggest that the model is considerably more effective at neighborhood-level spatial scales, partic-



(a) Smartphone User A

(b) Smartphone User B

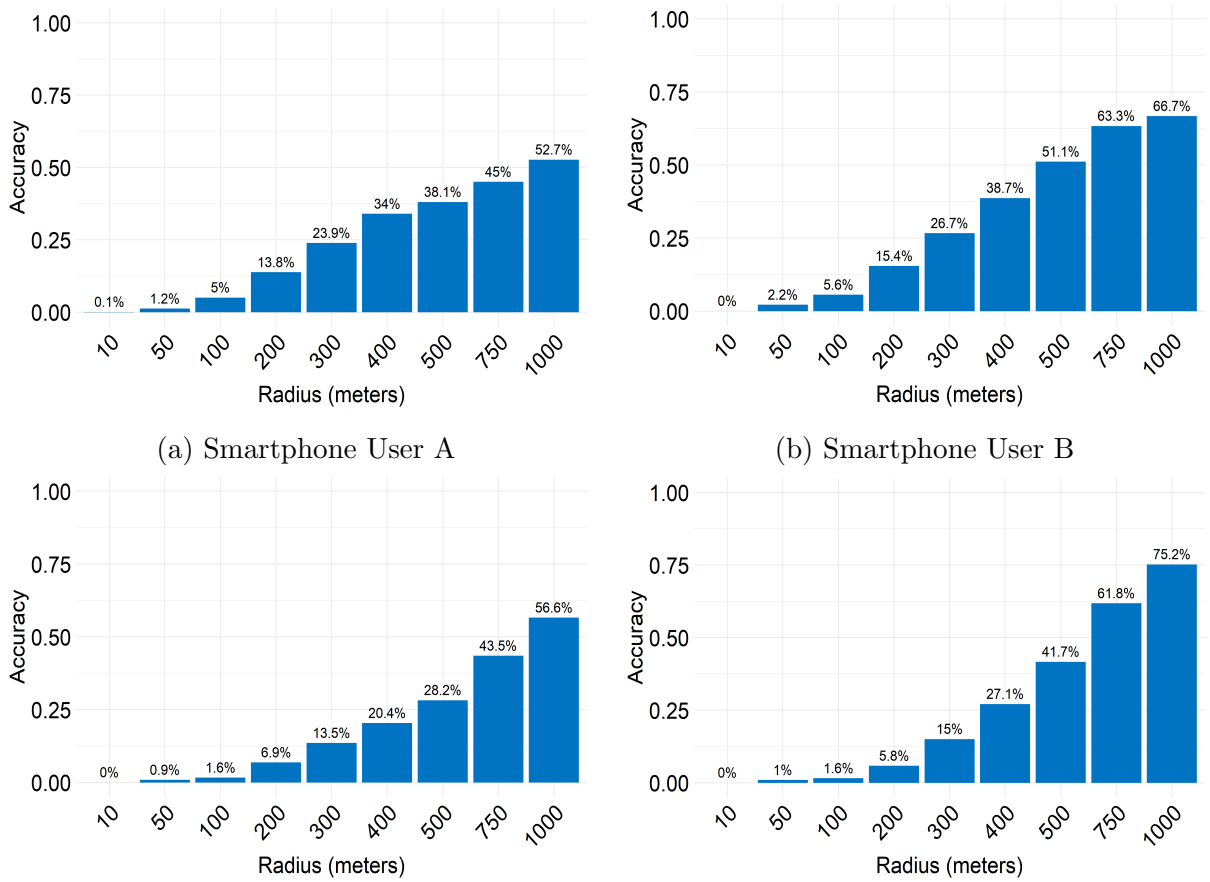
(c) Smartphone User C

(d) Smartphone User D

Figure 5.16: The feature importance in longitude prediction by the SVR models is presented for all smartphone users. The x-axis represents feature importance based on the RMSE, which reflects how much the model's performance drops when a feature's values are shuffled. The y-axis shows the overall contribution of each feature to the model's predictive performance.

ularly for user B. At the larger scale, between 500 and 1000 meters, the model exhibits a strong and consistent upward trend. At 500 meters, user B achieves 51.1%, followed by user D at 41.7%, user A at 38.1%, and user C at 28.2%. At 750 meters, the performance continues to rise, with user B at 63.3%, user D at 61.8%, user A at 45.5%, and user C at 43.5%. Finally, at 1000 meters, the model reaches its highest performance levels, with user D achieving 75.2%, user B at 66.7%, user C at 56.6%, and user A at 52.7%. These findings indicate that the model is most effective in capturing broader mobility trends at city-scale distances.

Similar to the RFR model, the comparison between the prediction error and spatial distance is separately illustrated for latitude and longitude prediction, and it is depicted in Figure 5.18. This comparison provides a quantitative measure of how accurately the SVR models predict the location ten minutes after timestamp t relative to the actual movement.



(a) Smartphone User A

(b) Smartphone User B

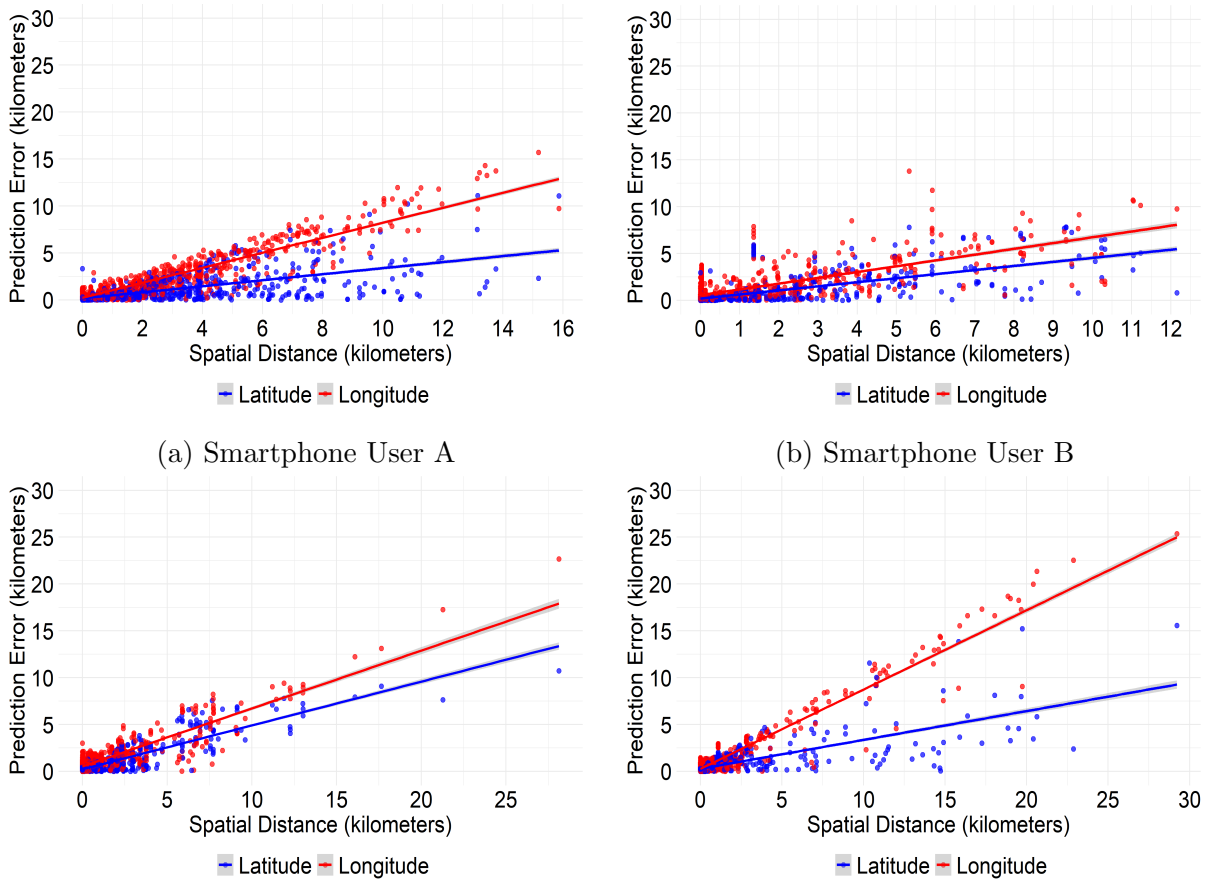
(c) Smartphone User C

(d) Smartphone User D

Figure 5.17: The comparison of prediction accuracies across radius thresholds for all smartphone users. The y-axis represents accuracy, defined as the proportion of predicted coordinates at timestamp t+15 through SVR models that fall within the specified radius of the observed coordinates at the same timestamp.

As shown in Figure 5.18, the prediction error increases as the spatial distance between the observed coordinates at timestamps t and t+15 grows, across all users. However, the magnitude of these errors varies between individuals. Notably, the prediction error for latitude is consistently lower than that for longitude, with a substantial margin. For users B and C, the error trends are closely aligned with minor differences, whereas a larger separation is evident for the other two users. This difference can be attributed to the directional nature of movement as the longitude reflects east-west positioning, while the latitude reflects north-south positioning. Since users' movements are generally more dispersed in the east-west direction. Therefore, the longitude errors tend to be consistently higher than the latitude errors.

Third, the MLPR is also implemented to predict the locations of users. A Markov-based feature engineering strategy is similarly employed using the same set of features in the MLPR model as it was applied in the RFR and SVR models. For the coordinates predic-



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 5.18: The prediction errors versus spatial distances are shown for all smartphone users. Each point represents a single prediction, and the x-axis shows the spatial distance traveled between the timestamp t and the timestamp t+15 in kilometers, and the y-axis shows the prediction error in kilometers. The prediction errors are calculated between the observed coordinates and the predicted coordinates at timestamp t+15 by the SVR models. The smoothed lines represent linear trends for each error type.

tion, the MLPR models were separately trained on the training set, which includes the first 80% of observations of the users' mobility dataset. To enhance the model's performance, A grid search with rolling window ahead cross-validation is similarly applied with the five folds to identify the optimal values of parameter size and decay. Similar to the RFR and SVR models, the original training set is split based on the temporal sequence, ensuring that the temporal order of observations is preserved. A five-fold strategy was applied, where each fold consisted of the first 80% of the observations, and the subsequent 4% of the testing set is used for validation. In the next fold, the window shifts forward, using the subsequent 4% of the set for validation. This process is repeated for the remaining folds until all portions of the training set are used.

The size parameter defines the number of neurons in the hidden layer, and the values of 5, 10, and 20 were selected randomly to train the model. A smaller hidden layer leads to a

simpler model that may generalize well to a new dataset, but it could struggle to capture complex patterns. Alternatively, increasing the size of the hidden layer enables the model to learn more intricate relationships within the dataset, potentially improving accuracy but also increasing the risk of overfitting. While the decay parameter introduces regularization by penalizing large weights and it helps to control model complexity. The values of this parameter are selected randomly as 0.001, 0.01, and 0.1. A smaller decay value permits the model to freely adjust its weights and enables better learning from the training set, but it also increases the risk of overfitting if the model becomes too specialized in training patterns. In contrast, the higher decay values impose greater constraints on weight updates, and they encourage better generalization by preventing the model from relying excessively on specific patterns in the training set. Therefore, the grid search approach discovered the optimal set of parameters separately for latitude and longitude for MLPR models, and the results of this systematic evaluation across all users are detailed in Table 5.5. It can be observed in Table 5.5 that the unique combination of parameters produces varying levels of RMSE for longitude and latitude. To enhance the performance of the model, the parameter combinations were selected for each user by prioritizing those that yielded the lowest RMSE. Afterwards, the MLPR model was trained using an optimal set of parameters on the training set and followed by validation using the testing set. The same procedure is implemented across all users, and the detailed outcomes are characterized in Table 5.6. The Table 5.6 provides the accuracies of the MLPR models along with optimal parameter combinations for predicting the latitude and longitude separately for users. It is discovered that the model explains 80.64% of the variation in latitude prediction with a MAE of 0.0074 and an RMSE of 0.0142 for user A, and it indicates good predictive performance. The longitude prediction achieves a R-squared of 94.05% with a MAE of 0.0191 and an RMSE of 0.0356, and it shows a high accuracy. For user B, the model explains 85.08% of the variation with a MAE of 0.0097 and RMSE of 0.0242 for latitude prediction. The longitude prediction achieves a 92.86% R-squared with a MAE of 0.0190 and a RMSE of 0.0397, and the model signifies better predictive power for longitude than latitude for this user. For user C, the model explains 94.24% of the variation in latitude prediction with a MAE of 0.0088 and RMSE of 0.0147. Comparatively, the longitude prediction is more precise, with an R-squared of 97.47%, MAE of 0.0112, and RMSE of 0.0245. For user D, the model explains 68.84% of the variation with a MAE of 0.0066 and RMSE of 0.0175, and it ensures reliable predictions. For the longitude prediction, the R-squared is 94.69% with

Table 5.5: The optimal parameter combinations were identified by the grid search approach with a rolling window ahead cross-validation across all smartphone users. The reported RMSE values represent the overall performance of the MLPR models under the selected parameter range.

Smartphone User	Parameters		Latitude	Longitude
	Size	Decay	RMSE	RMSE
A	5	0.001	0.0167	0.0347
	5	0.01	0.0171	0.0355
	5	0.1	0.0192	0.0372
	10	0.001	0.0179	0.0528
	10	0.01	0.0165	0.0345
	10	0.1	0.0181	0.0357
	20	0.001	0.0295	0.0341
	20	0.01	0.0166	0.0343
	20	0.1	0.0172	0.0359
	30	0.001	0.0179	0.0513
	30	0.01	0.0171	0.0347
	30	0.1	0.0173	0.0356
B	5	0.001	0.0124	0.0220
	5	0.01	0.0126	0.0223
	5	0.1	0.0131	0.0608
	10	0.001	0.0122	0.0221
	10	0.01	0.0123	0.0222
	10	0.1	0.0130	0.0230
	20	0.001	0.0123	0.0220
	20	0.01	0.0123	0.0218
	20	0.1	0.0126	0.0223
	30	0.001	0.0125	0.0222
	30	0.01	0.0124	0.0222
	30	0.1	0.0124	0.0222
C	5	0.001	0.0199	0.0277
	5	0.01	0.0221	0.1031
	5	0.1	0.0209	0.0276
	10	0.001	0.0194	0.0329
	10	0.01	0.0202	0.0295
	10	0.1	0.0202	0.0278
	20	0.001	0.0195	0.0386
	20	0.01	0.0202	0.0298
	20	0.1	0.0197	0.0273
	30	0.001	0.0223	0.0369
	30	0.01	0.0215	0.0281
	30	0.1	0.0196	0.0261
D	5	0.001	0.0143	0.0378
	5	0.01	0.0140	0.0423
	5	0.1	0.0150	0.0426
	10	0.001	0.0141	0.0421
	10	0.01	0.0140	0.0403
	10	0.1	0.0144	0.0392
	20	0.001	0.0152	0.0682
	20	0.01	0.0136	0.0385
	20	0.1	0.0150	0.0394
	30	0.001	0.0143	0.0383
	30	0.01	0.0143	0.0387
	30	0.1	0.0135	0.0366

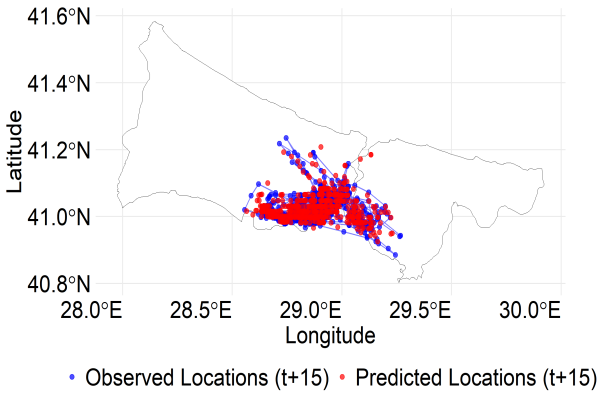
Table 5.6: The selected MLPR models with optimal parameters identified through the grid search with a rolling-window ahead cross-validation are described for all smartphone users. Each model was trained on the first 80% of the observations and then validated by predicting the coordinates at timestamp t+15, which were compared against the observed coordinates at timestamp t+15 in the remaining 20% of the testing set.

Smartphone User	Coordinate	Size	Decay	MAE	RMSE	R-squared
A	Latitude	10	0.01	0.0074	0.0142	0.8064
	Longitude	20	0.001	0.0191	0.0356	0.9405
B	Latitude	10	0.001	0.0097	0.0242	0.8508
	Longitude	20	0.01	0.0190	0.0397	0.9286
C	Latitude	10	0.001	0.0088	0.0147	0.9424
	Longitude	30	0.1	0.0112	0.0245	0.9747
D	Latitude	30	0.1	0.0066	0.0175	0.6884
	Longitude	30	0.1	0.0125	0.0318	0.9469

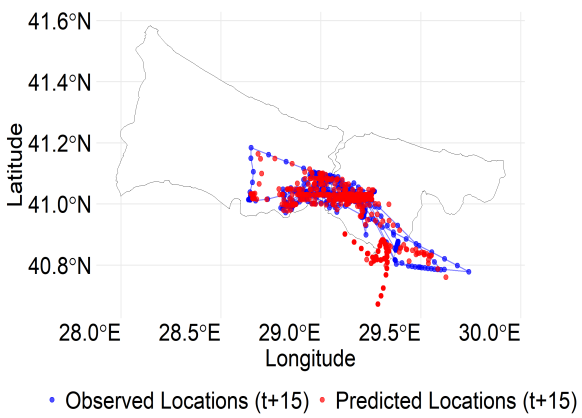
a MAE of 0.0125 and RMSE of 0.0318, which highlights a greater deviation in longitude predictions compared to latitude predictions. Overall, the MLPR models effectively capture the movement patterns of all users. The longitude predictions generally show slightly higher errors than the latitude predictions. The R-squared values remain above 80% for latitude predictions except for user D case and above 92% for longitude predictions across all users.

Next, the predicted latitude and longitude are combined to determine the locations at timestamp t+15. A visual comparison of the observed and predicted locations is visualized in Figure 5.19. It can be observed from Figure 5.19 that, for user A, the model generally predicts locations close to the observed ones. However, its performance declines for distant movements, particularly toward the northern and southeastern parts of Istanbul, where predictions become less precise. For user B, the model attempts to predict locations near the observed points in the northern area, although some discrepancies are evident. Moreover, it tends to underestimate movements in the southern and southeastern regions, resulting in predictions that fall outside the city boundary. Nevertheless, the geographically localized movements within the city center are captured relatively well. In the case of user C, the movement trajectories are more dispersed, and the model attempts to predict these locations, particularly on the western side of the city. For user D, locations in the western region present challenges for the model due to their separation from the primary movement in central areas. Also, the model produces several predictions that extend beyond the study area, especially toward the eastern side.

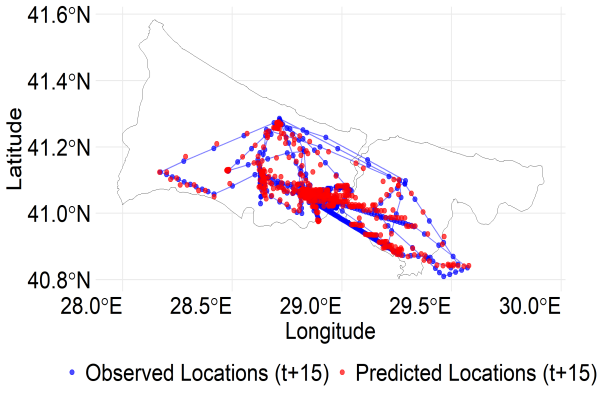
Subsequently, the relationship between the observed and predicted latitude at timestamp



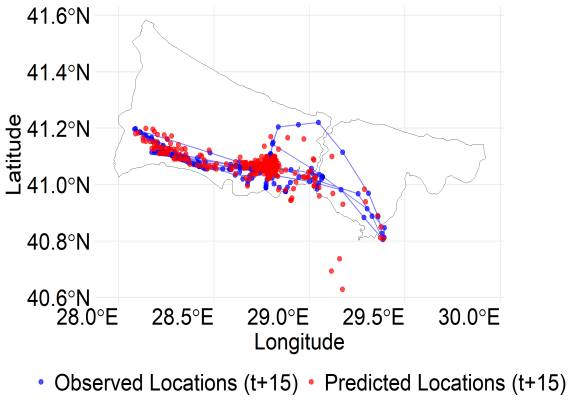
(a) Smartphone User A



(b) Smartphone User B



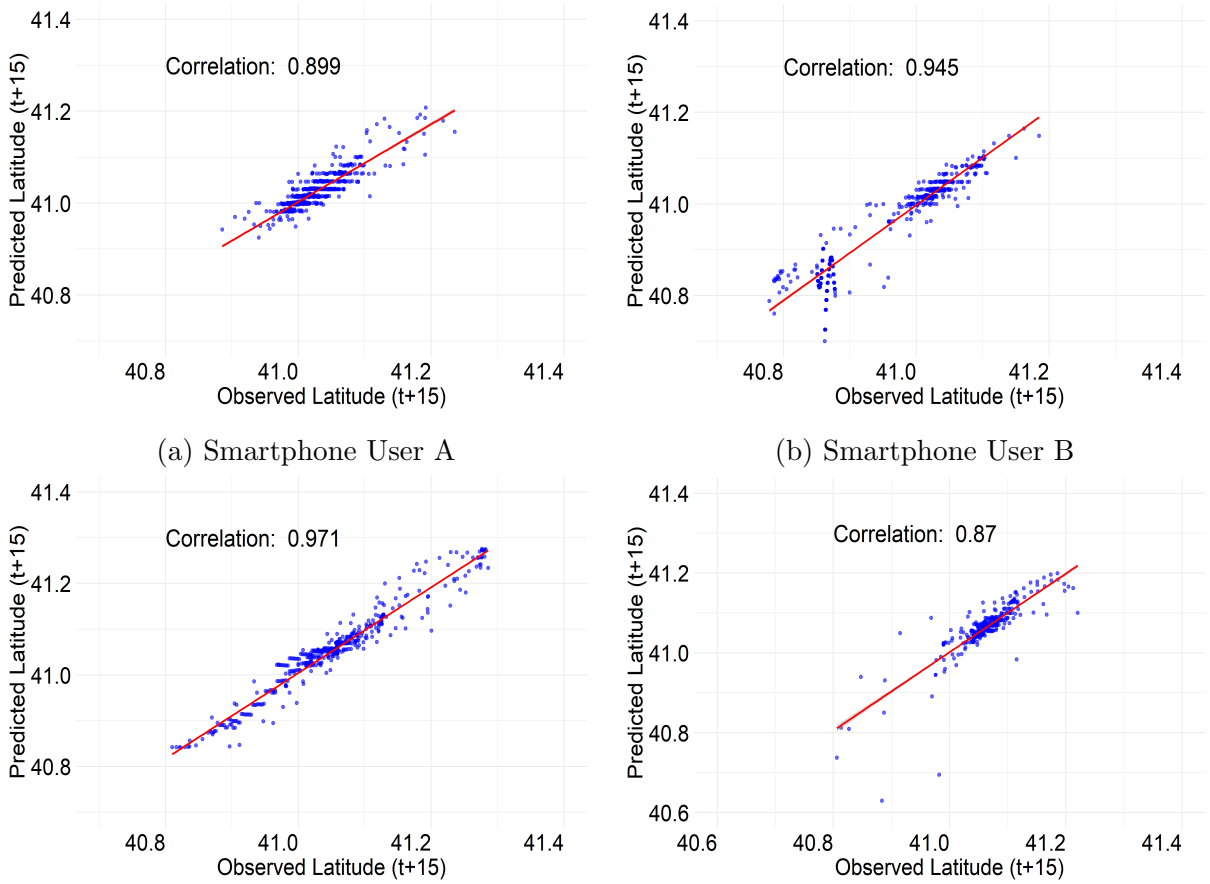
(c) Smartphone User C



(d) Smartphone User D

Figure 5.19: The general overview of observed and predicted coordinates at timestamp t+15 through the MLPR models for all smartphone users. The x-axis and y-axis represent geographic coordinates expressed in degrees. The blue points denote the observed trajectories at successive fifteen-minute intervals from the testing set, and the connecting line illustrates the true movement path. The red points represent the corresponding coordinates predicted by the model.

t+15 is illustrated through a scatter plot with a regression line, as shown in Figure 5.20. As illustrated in Figure 5.20, the majority of the predicted points are distributed around the regression line, and this indicates a generally consistent relationship between observed and predicted values, although not all points are tightly concentrated. The noticeable deviations are evident across all users, as users B and D exhibit a wider spread of points, suggesting greater variability in prediction accuracy. For user C, the points are more compactly distributed around the line, suggesting higher precision in the model's predictions. Overall, the observed and predicted values are strongly correlated, with correlation coefficients of 0.89, 0.94, 0.97, and 0.87 for users A, B, C, and D, respectively. Similarly, the relationship between the observed and predicted longitude at timestamp t+15 is displayed by the scatter plot with a regression line, as depicted in Figure 5.21. Figure 5.21 shows that most points are distributed around the regression line for all users, although some



(a) Smartphone User A

(b) Smartphone User B

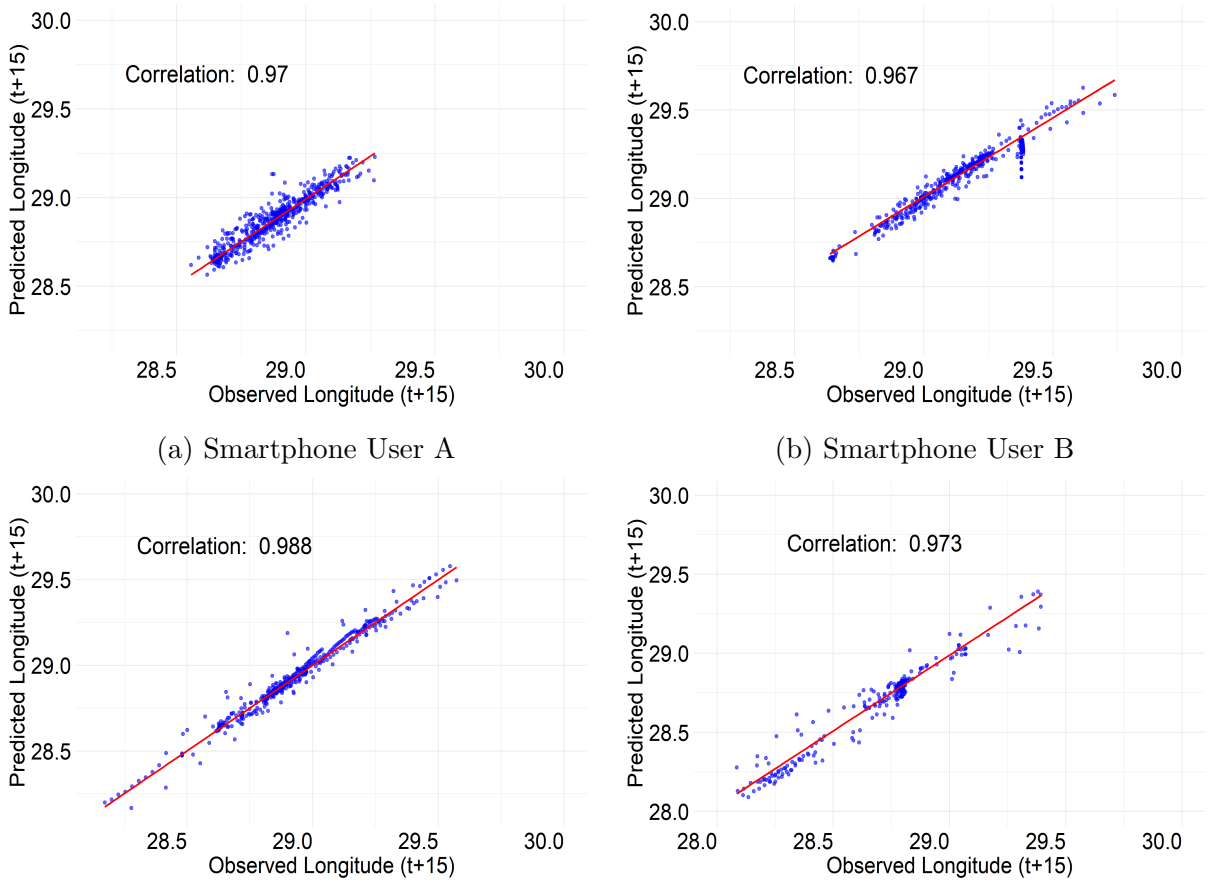
(c) Smartphone User C

(d) Smartphone User D

Figure 5.20: The relationship between the observed latitude at timestamp t+15 from the testing set and the predicted latitude at timestamp t+15 by the MLPR models is portrayed, along with the correlation values for all smartphone users.

deviations are observable. For users A and D, several points are more widely scattered, but the majority still cluster near the regression line. For user C, the points are tightly clustered around the line, which indicates high consistency. For user B, the points are less tightly clustered, with some lying significantly farther from the regression line, suggesting lower prediction precision. Moreover, the correlation values between observed and predicted locations at timestamp t+15 are 0.97, 0.96, 0.98, and 0.97 for users A, B, C, and D, respectively, and these indicate a strong association across all users.

Next, the deviations between the observed and predicted coordinates at timestamp t+15 through MLPR models are analyzed. The Figure 5.22 presents the predicted latitude values plotted against timestamp t+15, while Figure 5.23 shows the corresponding longitude values for the same timestamp. From Figure 5.22, it is noticed that the predicted latitude closely aligns with the observed latitude for all users, and this indicates the good predictive capability of the models. However, there are a few spikes that appear in all users' plots, where the predicted latitude fails to accurately capture the peak of the observed



(a) Smartphone User A

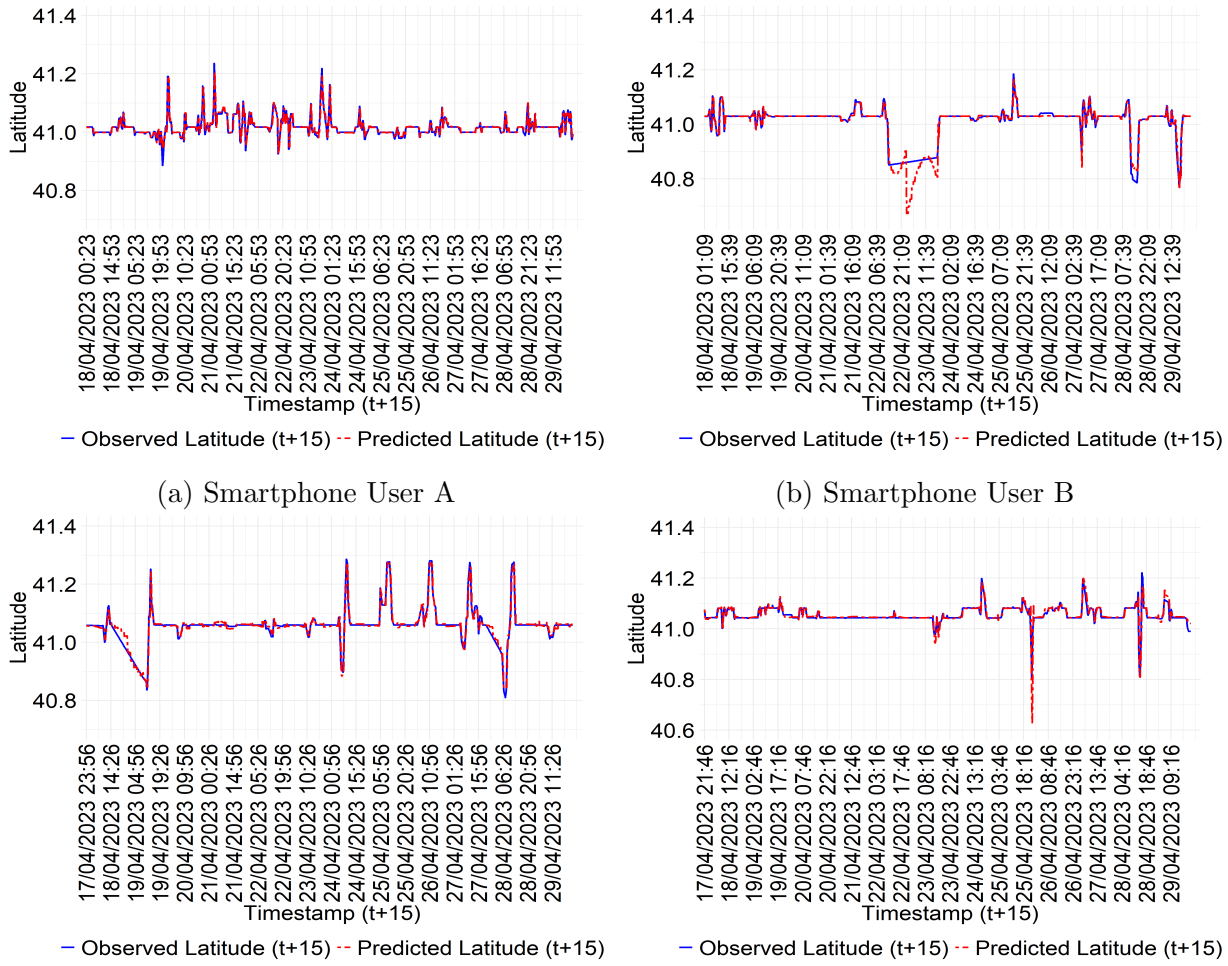
(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 5.21: The relationship between the observed longitude at timestamp t+15 from the testing set and the predicted longitude at timestamp t+15 by the MLPR models is shown, along with the correlation values for all smartphone users.

latitude at certain timestamps t+15, possibly due to sudden movement. For user B, these peaks are particularly pronounced, indicating that the model fails to capture the latitude trend effectively. For user D, the model overestimates the values at two spikes occurring at timestamp t+15. The Figure 5.23 illustrates the observed and predicted longitude over the timestamp t+15. For user C, the predicted longitudes closely follow the observed pattern, and this suggests the model's strong predictive capability. For users A and C, the model overestimates the longitude at certain timestamps, resulting in deviations from the observed values. Additionally, a few spikes are evident for users A, B, and D, where the model fails to accurately capture the peaks of the observed longitude at specific timestamps. To assess the contribution of predictors to latitude and longitude prediction, the permutation-based feature importance was utilized, which relies on the increase in RMSE when a feature is permuted. The larger RMSE increases indicate greater feature importance. The contribution of each feature in the MLPR models is depicted in Figures 5.24 and 5.25. Figure 5.24 indicates that the grid ID visited at timestamp t is the most influential feature for predict-



(a) Smartphone User A

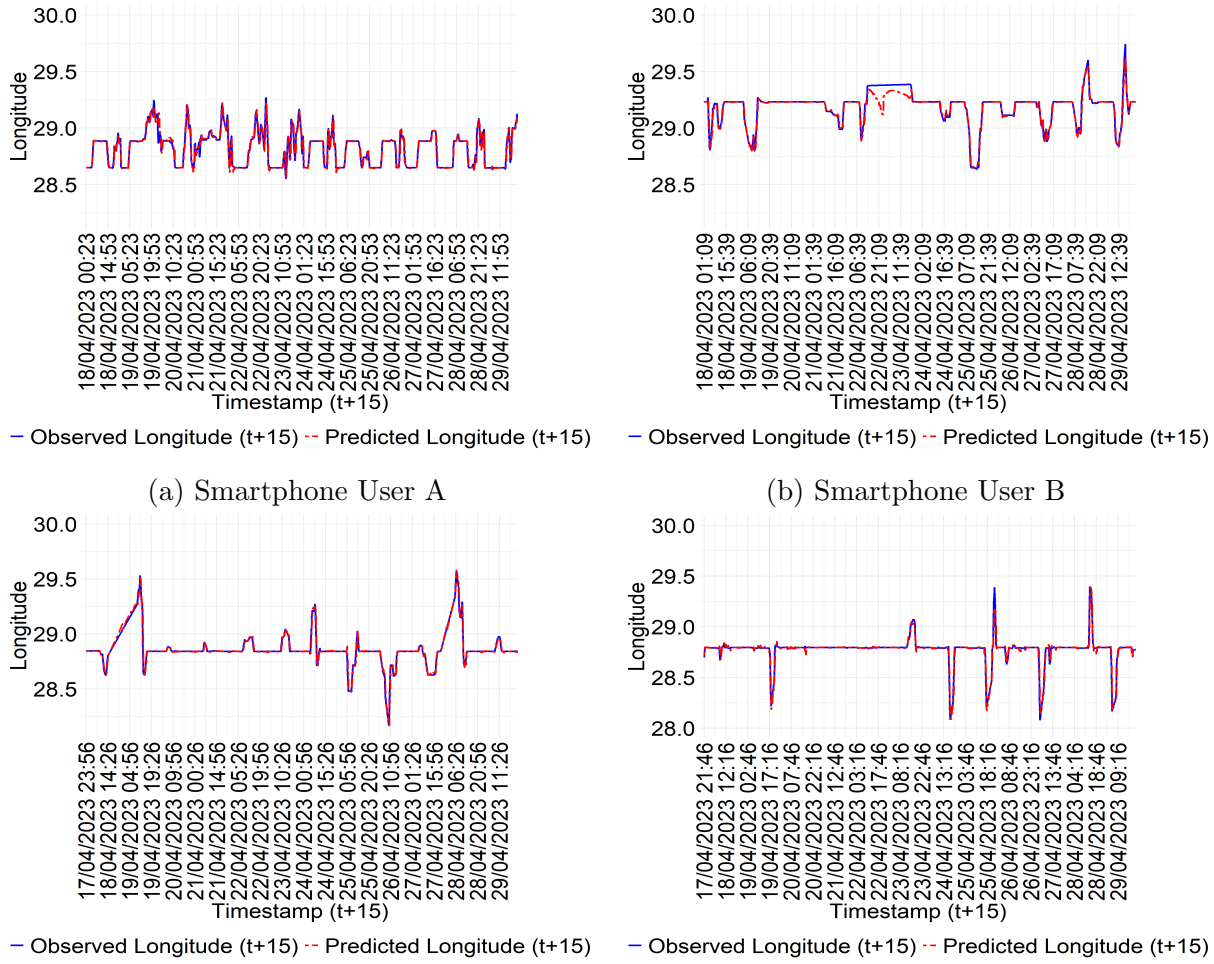
(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 5.22: A visual comparison of the observed and predicted latitude by the MLPR models is shown over the timestamps t+15 for all smartphone users. The observed latitudes at timestamp t+15 belong to the testing set.

ing latitude across all users, even though this feature does not strictly satisfy the Markov approximation. For users A, B, and D, the hour of day is the second most important predictor, whereas for user C, it ranks fourth. The observed longitude at timestamp t also appears among the top predictors, and it ranks third for users B, C, and D, and fourth for user. The remaining features have smaller and more similar contributions to latitude prediction, and their relative importance varies more noticeably across users compared to the consistently dominant influence of the grid ID visited at timestamp t. From Figure 5.25, it is evident that the observed longitude at timestamp t is the most influential predictor for predicting longitude at timestamp t+15 across all users, and it reflects the Markov approximation. The grid ID visited at timestamp t ranks as the second most important feature for users A, B, and D, and third for user C. The observed latitude at timestamp t ranks third for users A, B, and D and second for user C. Other features such as including



(a) Smartphone User A

(b) Smartphone User B

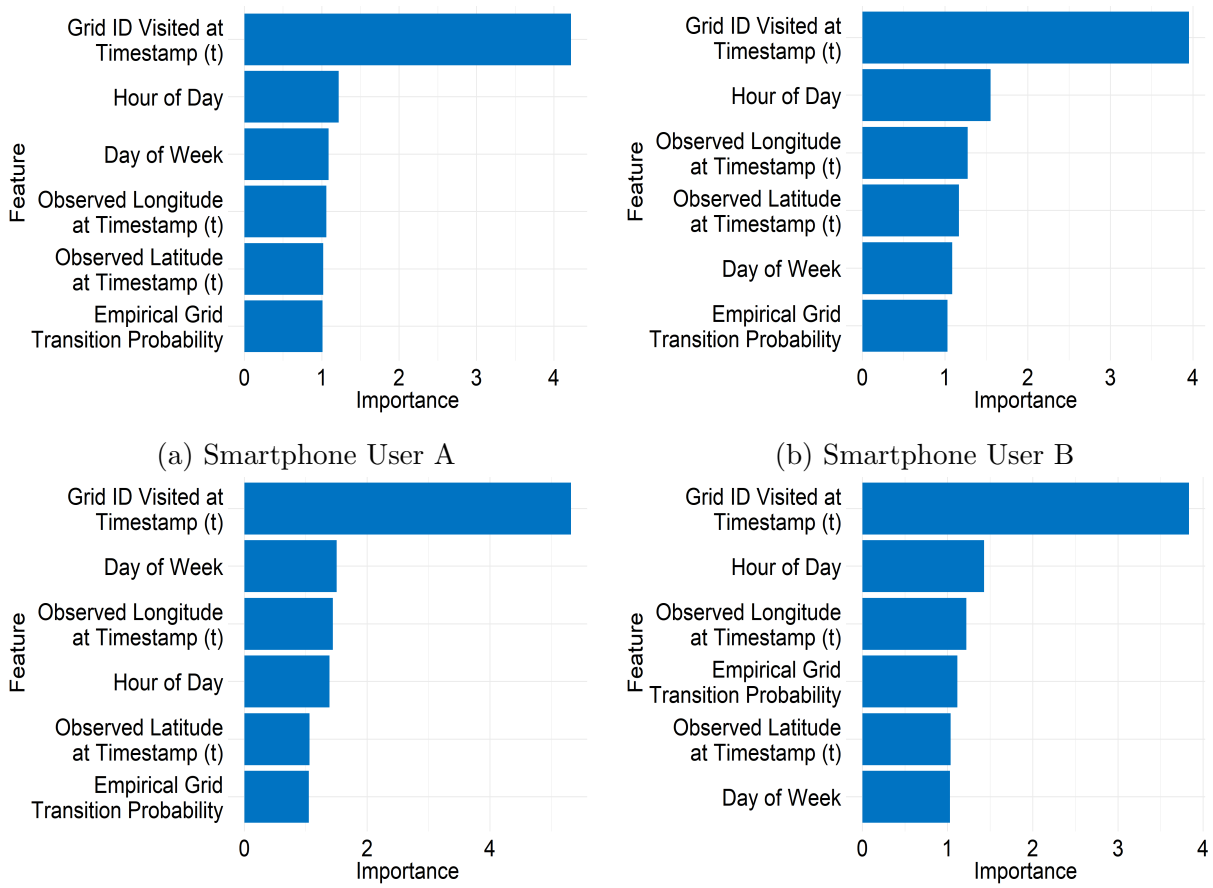
(c) Smartphone User C

(d) Smartphone User D

Figure 5.23: A visual comparison of the observed and predicted longitude by the MLPR models is shown over the timestamps t+15 for all smartphone users. The observed longitudes at timestamp t+15 are obtained from the testing set.

the hour of day, day of week, and the empirical grid transition probability have lower contributions, and their relative importance varies across users. Further details on the local variability of features are provided in Appendix B.2.

Subsequently, the AWR was computed between the observed and predicted coordinates at timestamp t+15. The evaluation employed the same radius thresholds as those used for the SVR and RFR models. The prediction accuracies across the different thresholds are shown in Figure 5.26. The Figure 5.26 illustrates that the MLPR model's prediction accuracy with respect to the radius threshold. At the smaller scale radius between 10 to 100 meters, the accuracy remains very low for all users. At 10 meters, all users achieve 0% accuracy, indicating that predictions at extremely fine granularity are unsuccessful. The accuracy slightly improves at 50 meters, with user B achieving the highest accuracy of 6.9%, followed by user A at 1%, user D at 0.8%, while user C remains at 0%. At 100 meters, user B



(a) Smartphone User A

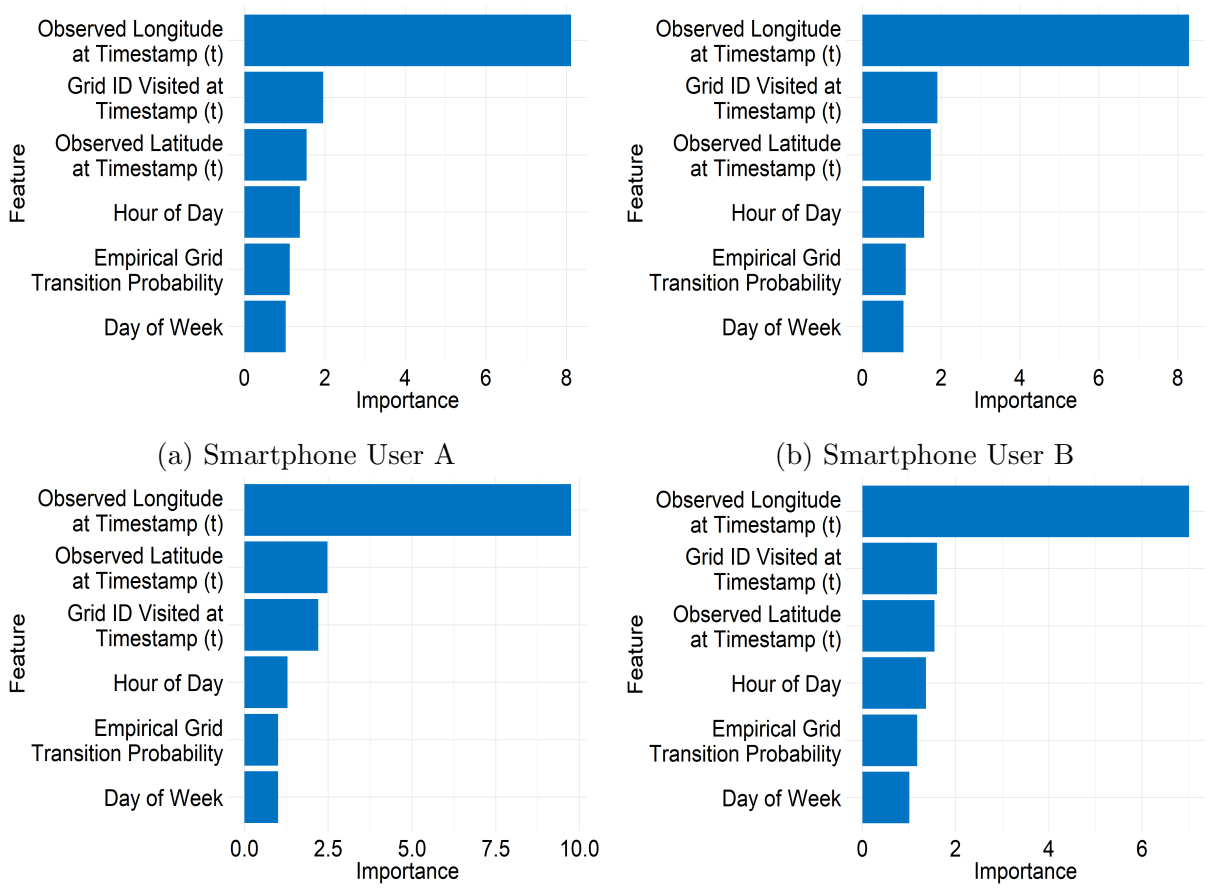
(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 5.24: The feature importance in latitude prediction through the MLPR models is shown for all smartphone users. The x-axis represents feature importance based on the RMSE, which reflects how much the model's performance drops when a feature's values are shuffled. The y-axis shows the overall contribution of each feature to the model's performance.

leads with 18.5%, followed by user A at 7.7%, user D at 2.6%, and user C at 0.2%. These findings suggest that the users exhibit substantial spatial uncertainty in highly localized distances, leading to weak predictability. At the medium scale between 200 to 400 meters, the model shows noticeable improvements across all users. At 200 meters, user B achieves 40.7%, while user A reaches 28.2%, followed by user D at 13.5% and user C at 4.3%. This upward trend continues at 300 meters, where user B reaches 47.9%, followed by user A at 40%, user D at 27.2%, and user C at 12.2%. By 400 meters, performance increases further, with user B at 51.5%, user D at 42.8%, user A at 45%, and user C at 27.5%. These results indicate that the model becomes more effective in recognizing broader spatial tendencies at neighborhood-level movement scales. At the large scale radius between 500 to 1000 meters, the model shows strong performance for all users. At 500 meters, user D achieves 56.4%, followed by user B at 52.3%, user A at 46.9%, and user C at 40.5%. The accuracies



(a) Smartphone User A

(b) Smartphone User B

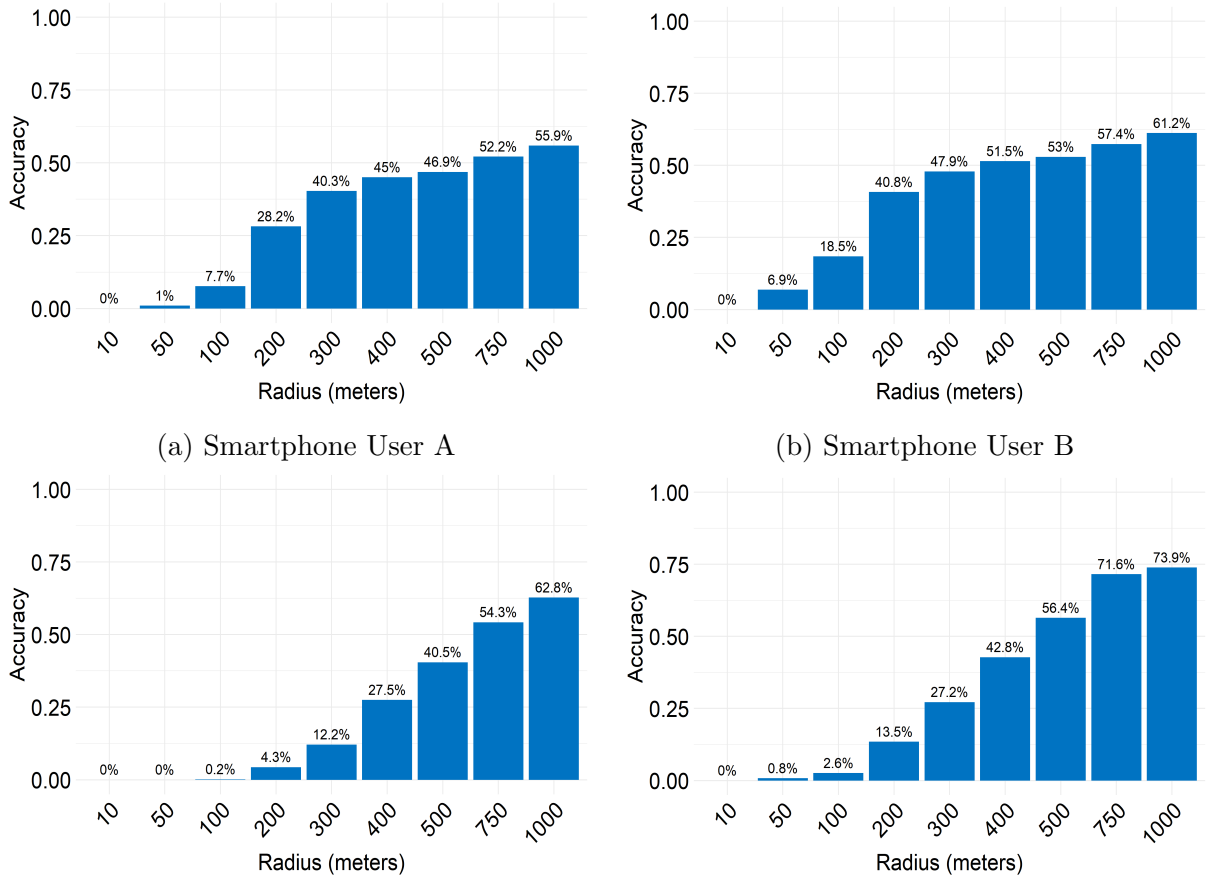
(c) Smartphone User C

(d) Smartphone User D

Figure 5.25: The feature importance in longitude prediction using the MLPR model is presented for all smartphone users. The x-axis represents feature importance based on the RMSE, which reflects how much the model's performance drops when a feature's values are shuffled. The y-axis shows the contribution of each feature to the model's performance.

increase further at 750 meters, as user D leads with 71.6%, followed by user B at 57.4%, user C at 54.3%, and user A at 52.2%. Finally, the model achieves its highest performance at 1000 meters, as user D reaches 73.9%, user C 62.8%, user B 61.2%, and user A 55.9%. These findings highlight that the model effectively captures broader mobility behaviors at larger spatial scales, where predicted locations increasingly fall within one kilometer of the observed positions.

Similar to the RFR and SVR models, the comparison between the prediction error and spatial distance is separately illustrated for latitude and longitude prediction, and it is shown in Figure 5.27. This comparison provides a quantitative measure of how accurately the MLPR models predict the location ten minutes after timestamp t relative to the actual movement. It can be seen in Figure 5.27 that the prediction error rises as the spatial distance between the observed coordinates at timestamp t and t+15 increases for all users. However, the magnitude of the errors varies across users. The prediction errors of latitude



(a) Smartphone User A

(b) Smartphone User B

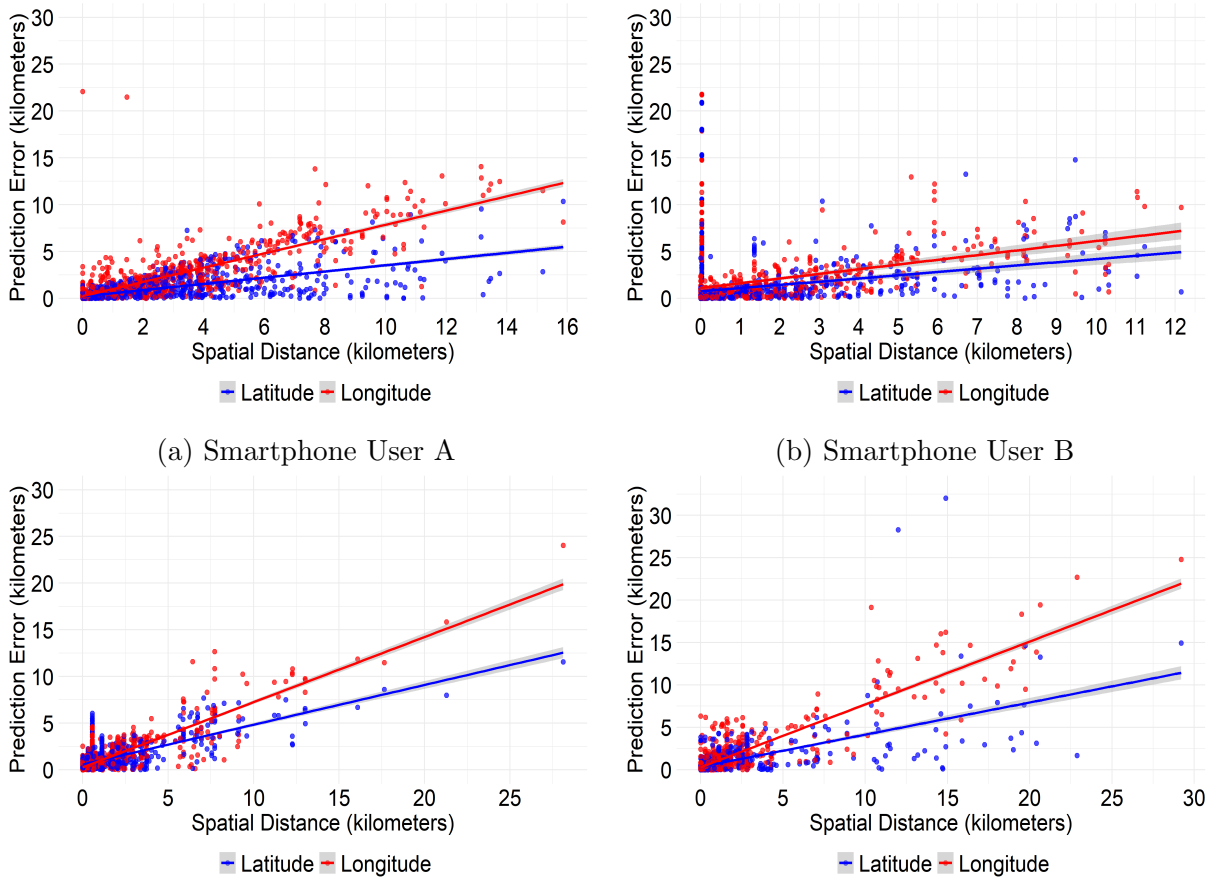
(c) Smartphone User C

(d) Smartphone User D

Figure 5.26: The comparison of prediction accuracies across radius thresholds for all smartphone users. The y-axis represents accuracy, defined as the proportion of predicted coordinates at timestamp t+15 through MLPR models that fall within the specified radius of the observed coordinates at the same timestamp.

are lower than those of longitude for all individuals, with a substantial margin. However, the error trends are closely aligned, with minor differences for users B and C. Since the users' movements are typically more dispersed in the east–west direction. Consequently, the longitude errors are consistently greater than the latitude errors.

Finally, it is essential to determine the most effective model for the location prediction. The Figure 5.8, 5.17, and 5.26indicate that all three models showed different levels of accuracy w.r.t different radii, particularly when considering heterogeneous movement behaviors. To evaluate performance, the accuracies with respect to small, medium, and large radii are illustrated in Figure 5.28 and 5.29. This categorization enables a more targeted analysis of which model is best suited for predicting mobility at different scales. The Figure 5.28 indicates that the RFR model consistently achieved the strongest performance compared to the SVR and MLPR models from small to medium-scale radii. In particular, it produced the highest accuracy values as the radius increased for users C and D. The MLPR



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure 5.27: The prediction errors versus spatial distances are shown for all smartphone users. Each point represents a single prediction, and the x-axis shows the spatial distance traveled between the timestamp t and the timestamp t+15 in kilometers, and the y-axis shows the prediction error in kilometers. The prediction errors are computed between the observed coordinates and the predicted coordinates at timestamp t+15 by the MLPR models. The smoothed lines represent linear trends for each error type.

models achieved moderate improvements as the radius increased, although still lagging behind RFR in most cases. Conversely, the SVR models underperformed across all settings, with their values remaining very close to zero at lower radii and only showing marginal improvements as the radius increased. Overall, the RFR model emerged as the most reliable model, while MLPR showed gradual improvement, and the SVR model consistently lagged behind.

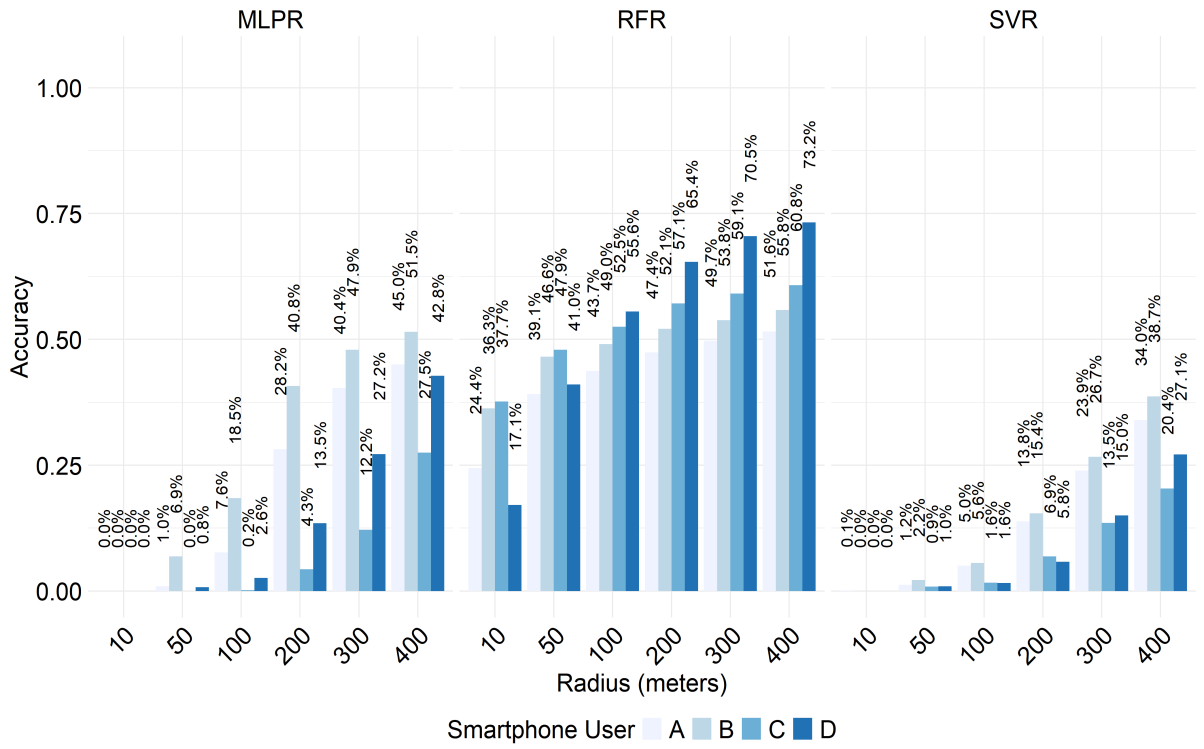


Figure 5.28: The comparison of models highlights the best-performing ones across small to medium-scale radii.

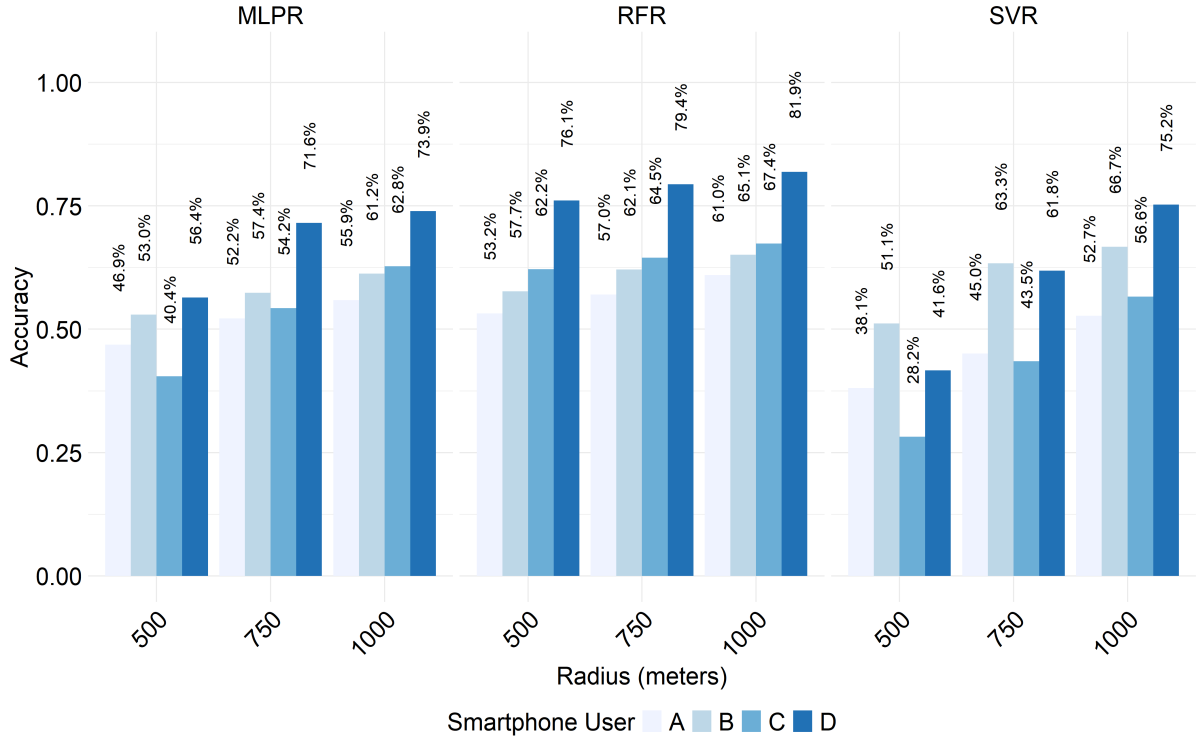


Figure 5.29: The comparison of models highlights the best-performing ones over large-scale radii.

From Figure 5.29, it is evident that all models generally improved in performance as

the spatial radius increased, and it reflects the fact that the broader spatial thresholds reduce prediction difficulty. Among the models, the RFR consistently accomplished the highest accuracies across users, and indicates its strong ability to capture the heterogeneity of movement patterns while still outperforming the other approaches. Both SVR and MLPR models showed notable improvements at larger scale radii. For instance, the SVR model obtained accuracies of 63.3% and 66.7% at radii of 750 and 1000 meters for user B, respectively. Similarly, the SVR model achieved an accuracy of 75.25% at 1000 meters for user D, surpassing the 73.9% accuracy achieved by the MLPR model for the same user. This suggests that the SVR models become more competitive when the prediction space is less restrictive, while the MLPR models also benefit from larger scale radii. However, both SVR and MLPR models remained inferior to the performance of RFR models.

5.4 Discussion

The chapter presents a strategy for predicting user coordinates by integrating the Markov formulation into ML models. However, working with mobility datasets poses several challenges during data processing and modeling. The study utilized an individual-level mobility dataset of four smartphone users over a period of sixty days, including latitude, longitude, and timestamp information. A primary challenge is that coordinates are recorded at irregular timestamps, with occasional gaps of several hours (Figure 3.3). Additionally, the frequency of recorded coordinates varies across dates and users, making user mobility patterns unpredictable (Figure 3.2). Another significant challenge arises from the diversity of mobility behaviors among users. Some users exhibit frequent and localized movements within the city, while occasionally taking long trips, resulting in sparsity in the movement (Figure 3.1). This combination of irregular timestamps and movement sparsity makes the accurate prediction of user mobility highly unpredictable. Considering these challenges, the following techniques were adopted for trajectory regularization. First, if the distance between consecutive locations is less than or equal to a specified threshold, the previous coordinates are maintained for the subsequent 15-minute intervals. Second, if the distance exceeds this threshold, a linear interpolation technique is applied to generate regular 15-minute trajectories. This approach has been widely used in vessel trajectory prediction (Shin and Yang, 2024) and urban human mobility prediction (Imai et al., 2024) to enhance data consistency and improve the performance of predictive models. The primary motivation for applying this technique is the presence of large temporal gaps between consecutive location records, which in some cases extend to several hours. Under such conditions, interpolation based on road-map matching between two coordinates can easily introduce bias. The study of (Mokbel et al., 2024) highlighted that most previous research relies on shortest-path search within road networks. However, this technique is limited when the road network is unknown, unreliable, or inaccurate, and it is practical only when trajectories are captured at very high sampling rates. In our case, the user's mobility dataset is recorded at low frequency and lacks contextual information to support reliable route inference. For example, it does not indicate whether the user stopped at a shopping mall, gas station, or other points of interest, nor does it provide details about the mode of transportation used to cover longer distances. In an urban environment, where certain roads are restricted to specific types of vehicles, such missing contextual information becomes

particularly critical. As a result, a uniform linear interpolation approach is adopted rather than relying on road-specific assumptions. It is important to clarify that the training set was constructed by preserving the temporal order of the data, consisting of the first 80% of the user mobility observations. All feature engineering was performed exclusively on this training set. The remaining 20% of the observations were reserved for validation purposes. In this chapter, the latitude and longitude are predicted separately by incorporating the Markov formulation into RFR, SVR, and MLPR models. The mathematical formulation for feature generation, along with the model equations, is provided in Appendix B.1. The feature importance analysis from RFR and SVR revealed that the coordinates at timestamp t+15 depend directly on the coordinates at timestamp t, which satisfy the Markov approximation (Figure 5.6 and 5.7).

In this chapter, a Markovian formulation is adopted in which a user's next location at timestamp t+15 is modeled as conditionally dependent on the location at timestamp t, together with other discretized contextual features. This formulation does not imply that smartphone-based human mobility strictly follows a first-order Markov process; rather, it provides a practical approximation, acknowledging that human movement may exhibit higher-order temporal dependencies. To empirically evaluate the validity of this Markovian approximation, history-augmented models incorporating coordinate information up to timestamp t-15 were additionally trained, while keeping the same model architectures, parameter estimation procedures, and evaluation criteria as in this chapter. The prediction target is the user's location at timestamp t+15 minutes remains unchanged. The prediction accuracy within multiple radius thresholds was then computed and compared against models using only information up to timestamp t. The results reveal negligible differences in prediction accuracy at smaller spatial radius thresholds, indicating that short-term mobility dynamics are effectively captured by the first-order Markov assumption. However, noticeable improvements at larger radius thresholds suggest the influence of long range temporal dependencies in human mobility patterns. Further implementation details are provided in Appendix B.3.

Although the ML models are frequently criticized for their limited interpretability, primarily due to their inherent black-box nature, which obscures the understanding of how individual factors influence the target variable (Lv et al., 2023). Therefore, the SHAP technique was used to measure the local effects of discretized spatial and temporal features on model predictions (Lundberg and Lee, 2017). In Appendix B.2, it can be observed that the

contribution of each feature varies across different models and users, reflecting both model-specific and user-specific patterns in how spatial and temporal features influence mobility predictions. In contrast, the standard feature importance analysis provides the aggregated effect of predictors on model performance. As outlined earlier, the Markov formulation was assumed as the next coordinates at timestamp t+15 (after fifteen minutes) depend on the coordinates at timestamp t (fifteen minutes earlier). Therefore, these spatial features are directly associated with short-term transition dynamics, while other spatial and temporal explanatory predictors exhibit comparatively lower contributions. The SHAP analysis for discrete features goes beyond aggregate feature importance by illustrating how individual feature values influence the model's predicted coordinates. The results confirm that both spatial and temporal features play a significant role in the model's predictions, capturing directional shifts and variations in movement patterns across space and time. Importantly, the effects of these features vary across users and models, reflecting user-specific behaviors and model-specific dynamics. This approach provides a localized and directional interpretation of user mobility, offering a more comprehensive understanding of how spatial and temporal dependencies shape movement behavior.

According to the results presented in Table 5.2, 5.4, and 5.6, the model performance varies across different users and modeling approaches. Nevertheless, all models show strong predictive capabilities, as evidenced by relatively high R-squared and low MAE and RMSE values for all users. These findings are also compared with previous studies in a meaningful way. As the study of (Zhang et al., 2025) obtained the highest prediction accuracy for their proposed model in terms of lower MAE of 730.362 meters and RMSE of 1438 meters for the coordinate's prediction, while compared to the baseline RF model of their study achieved the MAE of 8399.702 meters and RMSE of 11053.48 meters, which was higher. In our study, the coordinates were predicted separately for each user. After converting the degree-based errors into meters, the RFR model achieved MAE values of approximately 945, 690, 746, and 590 meters for latitude predictions of users A, B, C, and D, respectively. For longitude predictions, the corresponding MAE values were approximately 1495, 873, 858, and 1164 meters for the same users (as shown in Tables 5.5). Araújo et al. (2020) proposed ensemble RF models incorporating Markov properties and highlighted that next-location prediction should consider more than just the immediately previous location, as mobility patterns often rely on sequences of past behavior. Their model is based on hybrid trajectories and achieved accuracies of 71% and 83% for trajectory orders 2 and 3, respectively. Whereas

the individual-level trajectories were based on 50% of selected users, and the model obtained an accuracy of 39% and 46% for the same trajectory order. In comparison with traditional models like RF and SVM scored between 48% and 66% for similar trajectory orders. This proposed ensemble learning approach was designed for classification tasks and used features like bearing angle, last visited location, and the distance between previous and next locations. Our methodology relies on regression-based tasks, and it differs from existing approaches in several key aspects. Instead of adopting a one-size-fits-all model, this thesis adopts a personalized strategy by utilizing individual-level smartphone mobility datasets. To enhance predictive performance, new features were introduced, including the grid ID visited at timestamp t and the transition probability between the grids visited at timestamps t and t+15. Whereas the previous studies have incorporated temporal features such as the hour of the day, day of the week, and spatial coordinates. Our approach explicitly links the coordinates at timestamp t+15 to those at timestamp t through the Markov approximation. This relationship is further validated through feature importance analysis. These distinctions not only highlight the robustness and accuracy of our models but also contribute to a deeper understanding of emerging trends and enduring challenges in the domain of mobility prediction. Unlike previous studies that relied solely on MAE and RMSE to assess model accuracy. This thesis also employed the AWR metric to verify the precision of location predictions, as the traditional error metrics do not always reflect the practical accuracy of spatial predictions. The results based on the AWR indicate that the RFR model consistently achieved strong performance across small to large-scale radius for all users (Figure 5.28 and 5.29). Specifically, it maintained high accuracy and stability starting from a 10-meter radius, highlighting its ability to reliably predict precise locations. In comparison, the MLPR model showed only minor improvements at a 100-meter radius, while the prediction accuracy gradually increased as the radius expanded. The SVR model outperformed MLPR for users B and D at radii of 750 meters and 1000 meters, respectively. Overall, the RFR model proved to be the best performer and remained consistent across all radii, and it showed its ability to handle variations in mobility patterns. However, the model may produce less accurate predictions for locations that are not encountered during training, as it primarily interpolates within the range of the training set and cannot reliably extrapolate to entirely new or distant locations. The SVR and MLPR model tend to struggle with small-scale location predictions, which limit their ability to capture fine-grained mobility patterns and generally show improved performance only at larger spatial

scales. The ability to predict accurate locations has several advantages, particularly in applications such as urban planning, traffic management, and personalized mobility services. The precise location prediction can help optimize public transportation systems, improve ride-sharing algorithms, and enhance location-based services. Additionally, it can assist in emergency response planning by predicting where users are likely to be at a given time. However, the future study will focus on the high-frequency individual-level mobility datasets recorded at equal timestamps between 5 to 10 minutes to improve predictive performance. Additionally, the mobility interpolation techniques will be explored with contextual information to fill data gaps and improve the robustness of predictive models. By addressing these limitations, future research can further enhance mobility prediction models, making them more reliable and applicable in real-world scenarios. The statistical analysis was conducted using R version 4.1.

Chapter 6

Conclusion and Recommendations

Predicting human mobility remains a challenging problem due to the highly individualized nature of movement behavior and the inherent sparsity and inconsistency of spatiotemporal mobility data. The data sparsity arises from the low and variable frequency at which geographical coordinates are recorded, and sampling rates can differ substantially from date to date. Inconsistency further stems from missing observations and irregular timestamps. From a modeling perspective, the mobility data consist of both spatial coordinates and temporal information, and effectively integrating these two dimensions within a single predictive framework poses an additional challenge.

To address these issues, this thesis presents two complementary modeling strategies, both grounded in a Markov-based formulation integrated into ML models. The first strategy focuses on predicting spatiotemporal states, where a user's location is represented within fixed hourly intervals. Initially, classical statistical models such as the MC-1 and MC-2 models were evaluated. However, these models struggled to deliver reliable predictions due to the large number of distinct spatiotemporal states, which led to data sparsity and limited their predictive capacity. In contrast, ML models augmented with a Markov formulation and Markov-based feature engineering demonstrated substantially stronger performance by leveraging richer spatial and temporal information.

It is important to emphasize that the Markov formulation adopted in this thesis does not assume that the mobility of smartphone users strictly follows the Markov property. However, it serves as a modeling approximation that future spatiotemporal states conditionally depend on the most recently observed state, while allowing ML models to capture more complex dependencies through additional features. Among the evaluated ML approaches,

the RFC consistently achieved the highest accuracy across all users, outperforming SVC and MLPC models. The SVC showed only marginal improvements over the MLPC for most users and was competitive in a limited number of cases. The inferior performance of MC-1 and MC-2 can be attributed to their reliance on simplified assumptions and their inability to cope with the high dimensionality and imbalance of spatiotemporal states. Overall, the RFC is recommended for spatiotemporal state prediction, as it effectively captures both where a user will be and when they will be there within fixed hourly intervals. An important challenge identified in this phase is the imbalanced distribution of spatiotemporal states, which arises when certain locations are visited infrequently or only during specific times, leading to low-frequency states. This imbalance negatively affects model learning and highlights the need for more uniformly sampled, high-frequency mobility datasets in future studies. Since the spatiotemporal state formulation addresses irregular sampling by construction, and the prediction of precise geographic coordinates at fixed timestamps remains particularly difficult. To overcome this, the second strategy in this thesis focuses on coordinate prediction, where all user trajectories are first regularized to uniform fifteen-minute intervals. According to the Earthquake Network project, mobility data are generally recorded at approximately thirty-minute intervals. However, inconsistencies in recording frequency remain, thereby making temporal regularization a necessary preprocessing step.

Using this regularized data, the thesis adopts a Markov-based feature engineering approach within ML models to predict future locations at a fifteen-minute horizon. The results reveal that the RFR consistently outperformed other models across both small and large spatial radius thresholds, and indicating its strong ability to capture heterogeneous mobility patterns. In contrast, SVR and MLPR models exhibited poor performance at fine spatial radii, limiting their effectiveness for localized prediction, and showed improvements only at broader radii. The feature importance analysis reveals that the most recent location is the dominant predictor in both RFR and SVR models, which is consistent with the Markov formulation. Although human mobility may exhibit higher-order temporal dependencies. Therefore, the validity of the Markovian approximation was empirically evaluated by incorporating additional historical coordinate information. The results show negligible differences in prediction accuracy at fine spatial radii, indicating that short-term mobility dynamics are effectively captured using information up to the current timestamp. This supports the use of a first-order Markov approximation for short-horizon, localized

coordinate prediction, while acknowledging its limitations at broader spatial radii.

In summary, the RFR emerges as the most suitable model for coordinate prediction due to its robustness, adaptability, and balanced performance across users and spatial radii. Future work will focus on leveraging high-frequency individual-level mobility datasets recorded at uniform intervals of 5 to 10 minutes, as well as exploring ensemble modeling approaches that jointly capture localized and global spatiotemporal dependencies. These advancements are expected to contribute to the development of more accurate and context-aware urban mobility forecasting systems, with practical applications in transportation planning, smart city infrastructure, and location-based services.

Appendix A

Supplementary Material of Chapter 4

A.1 Model Formulation

To characterize the dynamic evolution of movement patterns, the study discretizes both the spatial and temporal domains. The spatial discretization is achieved using a GMM clustering approach. Let each observation at timestamp t be represented by a continuous spatial coordinate vector $z_t = (x_t, y_t)$. The GMM partitions the space into K spatial components, each corresponding to a latent region represented by a "spatial state". The assignment of z_t to a spatial state is obtained as,

$$S_t^s = \underset{k \in \{1,\ldots,K\}}{\arg\max}\, p_k(z_t) \tag{A.1}$$

where $p_k(z_t)$ is the posterior probability that the observation z_t belongs to cluster k under the fitted GMM. Temporal discretization is implemented considering the fixed hourly intervals to capture diurnal variations in movement. The day is divided into five non-overlapping intervals, and each interval is denoted by an index $S_t^{(t)} \in \{1, ..., 5\}$. A current spatiotemporal state S_t is then defined as the joint combination of spatial and temporal components as follows,

$$S_t = (S_t^{(s)}, S_t^{(t)}). \tag{A.2}$$

Therefore, the evolution of system states is expressed as a discrete sequence $\{S_1, S_2, ..., S_T\}$.

A.1.1 Estimation of Empirical Spatiotemporal State Transition Probabilities

To capture historical movement tendencies between spatiotemporal states, we derive empirical transition probabilities from an ordered dataset. Let the dataset be chronologically ordered by timestamp, ensuring temporal causality. The number of observed transitions from state i to state j up to timestamp $t-1$, but excluding timestamp t is denoted by $N_{i\to j}(t-1)$. The total number of transitions departing from state i up to timestamp $t-1$ is defined as,

$$N_i(t-1) = \sum_j N_{i\to j}(t-1) \tag{A.3}$$

The empirical transition probability, based solely on historical observations before the timestamp t, is then

$$\hat{P}_t(S_{t+1} = j | S_t = i) = \frac{N_{i\to j}(t-1)}{N_i(t-1)}, \quad for\, N_i(t-1) > 0 \tag{A.4}$$

This formulation ensures that only past information is used to compute the probability for the current observation, preserving the temporal integrity of the estimation.

A.1.2 Unseen Transitions and First-Occurrence Behavior

In practice, some transitions between states may not have been observed in the training data. For such unseen transitions, the empirical counts satisfy:

$$N_{i\to j}(t-1) = 0, \tag{A.5}$$

which directly leads to an computed transition probability of zero according to (A.4):

$$\hat{P}_t(S_{t+1} = j | S_t = i) = 0 \tag{A.6}$$

Thus, the first occurrence of a transition $i \to j$ in the chronological dataset will always have a zero probability prior to being observed. After the first instance is recorded, the cumulative counts are updated as follows,

$$N_{i\to j}(t) = N_{i\to j}(t-1) + 1, \tag{A.7}$$

and,

$$N_i(t) = N_{i\to j}(t-1) + 1, \tag{A.8}$$

and the probability becomes positive for subsequent instances:

$$\hat{P}_{t+1}(S_{t+2} = j | S_{t+1} = i) = \frac{N_{i\to j}(t)}{N_i(t)}. \tag{A.9}$$

Hence, the transition probability for a pair (i, j) evolves dynamically with the accumulation of observed transitions. There was no smoothing or prior adjustment applied. For the unseen transitions, the probability of zero was assigned in the test dataset. This design directly reflects the empirical frequency of transitions as observed in the historical data, with zero indicating an unobserved movement.

A.1.3 Dynamics of Self-Transitions and Departures

The empirical formulation captures realistic movement dynamics through the changing balance of self and cross-state transitions. If the observed transition at timestamp t is a self-transition ($i \to i$), both numerator and denominator in (A.9) increase, maintaining or slightly raising the persistence probability. Conversely, when the observed transition is a departure ($i \to j,\ j \neq i$), the denominator $N_t(t)$ increases while $N_{i\to i}$ remains constant, thereby reducing the self-transition probability is described as follows,

$$\hat{P}_t(S_{t+1} = i | S_t = i) = \frac{N_{i\to i}(t-1)}{N_t(t-1)} \to \frac{N_{i\to i}(t-1)}{N_i(t-1)+1}. \tag{A.10}$$

This behavior reflects the empirical effect of mobility as remaining in the same state sustains high persistence, while departures dilute that likelihood. Consequently, the frequent transitions between distinct states naturally lead to increased cross-state probabilities and decreased self-transition dominance, representing realistic movement patterns observed in spatial systems.

A.1.4 Feature Construction and Model Integration

The empirically computed transition probability was incorporated as an explanatory variable in the supervised learning framework. For each observation at timestamp t, the input feature vector is defined as,

$$y_t = [S_t, d_t, \hat{P}_t(S_{t+1}|S_t)], \tag{A.11}$$

where S_t is the current spatiotemporal state, d_t is the categorical day of week feature, and $\hat{P}_t(S_{t+1}|S_t)$ is the historical spatiotemporal transition probability derived from training data. The target variable is the next spatiotemporal state S_{t+1}, leading to a standard supervised learning formulation. Formally, the relationship between input and output can be written as follows,

$$S_{t+1} = f(S_t, d_t, \hat{P}_t(S_{t+1}|S_t)) + \varepsilon_t. \tag{A.12}$$

Where $f(.)$ is a nonlinear mapping learned from data, and ε_t denotes model error. The presented mobility prediction framework is formulated as a supervised classification task, where the target variable is the next spatiotemporal state and input variables are lag-based features derived from past observations. This formulation ensures that only past information contributes to predicting future states, maintaining temporal causality. The study of Bray and Han (2004) used rainfall and flow series, where rainfall and previous flow observations served as input features and the subsequent flow value was treated as the target variable. Similarly, (Yu et al., 2006) applied lagged hydrological variables within a support vector regression framework for real-time flood forecasting. The incorporation of transition probabilities as predictive features was introduced by Mohammed and Gündüç (2022), who constructed feature vectors for graph nodes to capture their local connectivity structures. In their work, these transition-based representations were utilized for learning learning tasks such as node classification and link prediction. By following the aforementioned studies, the chapter adopts a lag-based structure for smartphone user mobility prediction, in which the future spatiotemporal state is inferred from current and historical features under a first-order Markov assumption using ML classifiers such as RFC, SVC, and MLPC to predict the most probable next spatiotemporal state based on empirically derived spatiotemporal transition information together with other contextual features.

A.2 Supplementary Details

The Table A.1 provides an overview of the spatiotemporal states, including an explanation of their construction and characterization. As shown in Figure 4.1 in Chapter 4, the users A, B, C, and D have 10, 10, 12, and 11 spatial states, respectively. When combined with temporal states as hourly intervals of the day, this results in a large number of possible spatiotemporal states. To simplify interpretation, the four spatial states are considered generally, then each is combined with all temporal states, and provides their interpretation.

Table A.1: Overview of spatiotemporal states. Only the four spatial states are considered, and each in combination with temporal states to explain their construction and description.

Spatial State	Temporal State	Hourly Interval in Temporal State	Notation of Spatiotemporal States	Description
1	1	00:00 to 6:00	1_1	The user visited or remained in spatial state 1 during the night-time
1	2	6:00 to 9:00	1_2	The user visited or remained in spatial state 1 during the morning commute or arrival period at the workplace.
1	3	9:00 to15:00	1_3	The user visited or remained in spatial state 1 during office or work time.
1	4	15:00 to 18:00	1_4	The user visited or remained in spatial state 1 during the afternoon commute or departure from the workplace.
1	5	18:00 to 24:00	1_5	The user visited or remained in spatial state 1 during the evening.
2	1	00:00 to 6:00	2_1	The user visited or remained in spatial state 2 during the night-time.
2	2	6:00 to 9:00	2_2	The user visited or remained in spatial state 2 during the morning commute or arrival period at the workplace.
2	3	9:00 to15:00	2_3	The user visited or remained in spatial state 2 during office or work time.
2	4	15:00 to 18:00	2_4	The user visited or remained in spatial state 2 during the afternoon commute or departure from the workplace.
2	5	18:00 to 24:00	2_5	The user visited or remained in spatial state 2 during the evening.
3	1	00:00 to 6:00	3_1	The user visited or remained in spatial state 3 during the night-time.
3	2	6:00 to 9:00	3_2	The user visited or remained in spatial state 3 during the morning commute or arrival period at the workplace.
3	3	9:00 to15:00	3_3	The user visited or remained in spatial state 3 during office or work time.
3	4	15:00 to 18:00	3_4	The user visited or remained in spatial state 3 during the afternoon commute or departure from the workplace.
3	5	18:00 to 24:00	3_5	The user visited or remained in spatial state 3 during the evening.

The rest of all combinations that share the same interpretation but differ only in spatial states.

Appendix B

Supplementary Material of Chapter 5

B.1 Model Formulation

Let $T \in N$ denote the number of observed timestamps, and let $\Delta t = 15$ minutes represent the fixed sampling interval. For each time index $t \in \{1, 2, 3, ..., T\}$, the observed spatial position is given by latitude and longitude, which are described in vector form as follows,

$$z_t = \begin{bmatrix} lat_t \\ lon_t \end{bmatrix} \in R^2, \tag{B.1}$$

where z_t denotes the geographical coordinates observed at timestamp t. Let the spatial domain be partitioned into a finite set $G = \{1, 2, 3, ..., K\}$ of uniform grid cells ($2\,km \times 2\,km$) and K denoting the total number of distinct grid cells. For each grid $i \in G$ and let $c_i = (lat_i, lon_i) \in R^2$ denote a representative point of cell i (centroid). Let $d(.\,,.)$ denote the geodesic distance function between two geographic pairs (implemented using the Haversine formula in practice). Let each grid cell $g \in G$ be associated with a predefined centroid c_g. The grid assignment function $M : R^2 \to G$ then maps a coordinate pair (lat_t, lon_t) to the grid cell whose centroid is closest under the distance measure $d(.\,,.)$, i.e.,

$$g_t = M(z_t) = \arg\min_{g \in G} d((lat_t, lon_t), c_g), \tag{B.2}$$

where g_t indicates the grid cell assigned at the timestamp t. Temporal contextual variables are derived from the timestamp t and take values in discrete finite sets. Specifically, the

hour of day is represented as follows,

$$h_t \in \{0, 1,\ 2,\ 3, ..., 23\}, \tag{B.3}$$

and the day of week is encoded as follows,

$$d_t \in \{1,\ 2,\ 3, ..., 7\}. \tag{B.4}$$

Mobility regularity is partially characterized through empirical transitions between the grid cells. Let $N_t(i \to j)$ represents the cumulative number of transitions from grid cell i to grid cell j observed prior to timestamp t. This is formally expressed as follows,

$$N_t(i \to j) = \sum_{\tau=1}^{t-1} 1\{g_\tau = i,\ g_{\tau+\Delta t} = j\}, \tag{B.5}$$

where $1\{.\}$ is the indicator function equal to 1 if the condition holds and 0 otherwise. The total number of historical departures from the grid cell i before the timestamp t is thus defined as

$$N_t(i) = \sum_{j \in G} N_t(i \to j) = \sum_{\tau=1}^{t-1} \{g_t = i\}. \tag{B.6}$$

Using equation B.5 and B.6, the empirical transition probability of moving from grid i at timestamp t to grid j at timestamp $t + \Delta t$ based solely on observed historical data available before timestamp t, given by:

$$\hat{P}_t(g_{t+\Delta t} | g_t) = \hat{P}_t(j|i) = \left\{ \begin{array}{ll} \frac{N_t(i \to j)}{N_t(i)}, & if\ N_t(i) > 0 \\ 0, & if\ N_t(i) = 0 \end{array} \right\}. \tag{B.7}$$

The concept follows the same formulation as that adopted for the estimation of spatiotemporal state transition probabilities. However, in this case, it specifically considers transitions among grids, including unseen and self-transitions. The corresponding mathematical details are provided in Appendices A.1.2 and A.1.3.

Based on spatial, temporal, and transition-based dependencies, the complete feature vector at timestamp t is constructed as follows,

$$y_t = \{lat_t,\ lon_t,\ h_t,\ d_t,\ g_t,\ \hat{P}_t(g_{t+\Delta t} | g_t)\}. \tag{B.8}$$

Under the Markov assumption, the $z_{t+\Delta t}$ is conditionally dependent only on the current spatial-temporal context y_t. Therefore,

$$P(z_{t+\Delta t} \,|\, y_t,\, y_{t-\Delta t}, ...) = P(z_{t+\Delta t} \,|y_t). \tag{B.9}$$

Therefore, the mobility prediction problem is formulated as a supervised regression task, where the goal is to predict the future spatial coordinates based on the current spatial coordinates along with discretized spatial and temporal contextual features. Let the target output at timestamp $t + \Delta t$ as follows,

$$z_{t+\Delta t} = \begin{bmatrix} lat_{t+\Delta t,} \\ lon_{t+\Delta t} \end{bmatrix}. \tag{B.10}$$

We seek a mapping t and parameterized by θ, such that,

$$z_{t+\Delta t} = f_\theta(y_t) + \varepsilon_t, \tag{B.11}$$

where y_t is the feature vector at timestamp t, $f_\theta(.)$ denote a nonlinear regression function, and ε_t is a stochastic error term. The function $f_\theta(.)$ is approximated using three machine learning regression models such as RFR, SVR, and MLPR. Consequently, the latitude and longitude predictions are expressed as separate regression functions as follows,

$$lat_{t+\Delta t} = f_{lat}(lat_t,\ lon_t,\, h_t,\, d_t,\, g_t,\, \hat{P}_t(g_{t+\Delta t} \,|g_t)), \tag{B.12}$$

and

$$lon_{t+\Delta t} = f_{lat}(lat_t,\ lon_t,\, h_t,\, d_t,\, g_t,\, \hat{P}_t(g_{t+\Delta t} \,|g_t)). \tag{B.13}$$

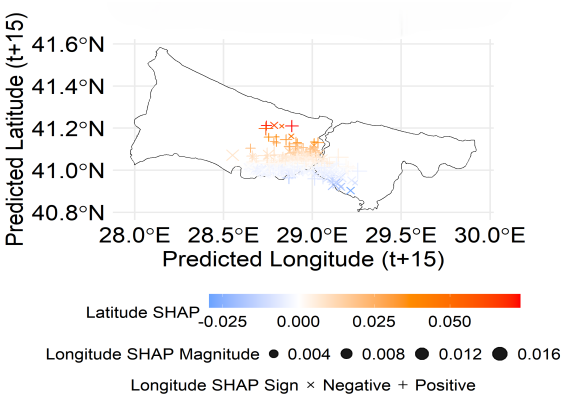
The presented mobility prediction framework is formulated as a supervised regression task that infers future spatial coordinates from current and historical spatiotemporal features. This lag-based feature construction is consistent with established time-series modeling practices, as a study by Bray and Han (2004), who employed historical rainfall and runoff inputs to predict future discharge using SVM. Similarly, a related formulation was adopted by (Yu et al., 2006), who utilized lagged hydrological variables within the SVR framework for real-time flood stage forecasting. Inspired by these formulations, the mobility prediction problem is modeled under a Markov assumption, in which the future location is represented as a functional mapping of the current feature vector, thereby justifying the use of nonlinear

regression models such as RFR, SVR, and MLPR within the human mobility forecasting context.

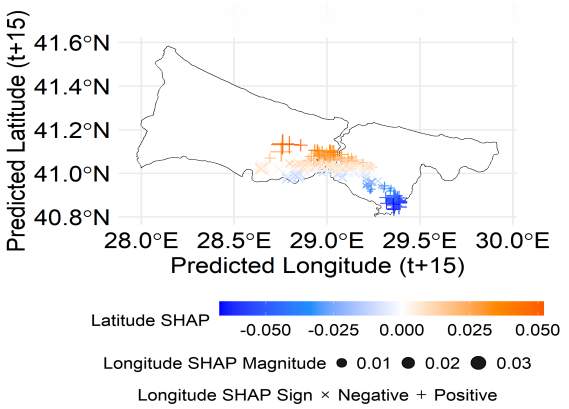
B.2 Analysis of Local Feature Effects Using the Shapley Additive Explanations (SHAP)

The Figure 5.6 and 5.7 illustrate the contribution of each feature to the RFR model's prediction. Although the standard feature importance analysis only quantifies the magnitude of each feature's contribution, and does not provide information about the direction of the effect in terms of smartphone user movement.

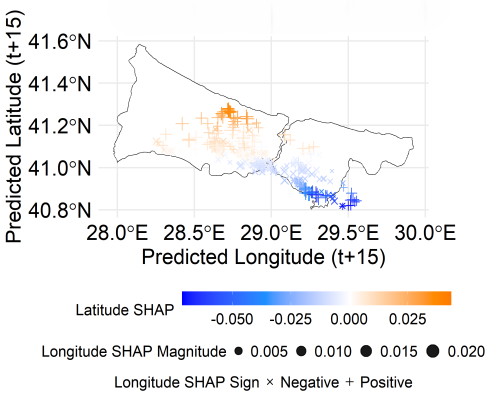
To address this limitation, the Shapley Additive Explanations (SHAP) technique was applied (Lundberg and Lee, 2017), and it is based on cooperative game theory and computes Shapley values, which fairly attribute the prediction output to individual features by considering all possible combinations of features. For each observation, the SHAP value of a feature quantifies its average marginal contribution to the prediction, computed over all possible subsets of features. Therefore, the SHAP inherently produces observation-level explanations, and it assigns a unique value to each sample point of the feature. The SHAP values are expressed on a signed scale, ranging from negative to positive. A positive value means that the feature contributes to an increase in the predicted target, for example, pushing the predicted latitude northward or the predicted longitude eastward in terms of coordinate prediction. Conversely, a negative value indicates that the feature contributes to a decrease in the predicted target, corresponding to a southward shift in latitude or a westward shift in longitude. This signed scale provides not only the strength of a feature's influence but also its directional effect, offering a clear and interpretable view of how each feature affects the model's predictions. The spatial distribution of SHAP values for grid IDs visited at timestamp t is illustrated in Figure B.1.



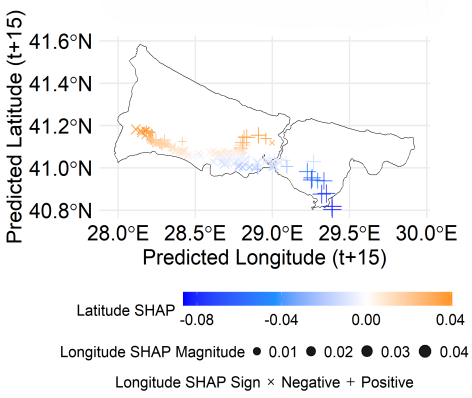
(a) Smartphone User A



(b) Smartphone User B



(c) Smartphone User C



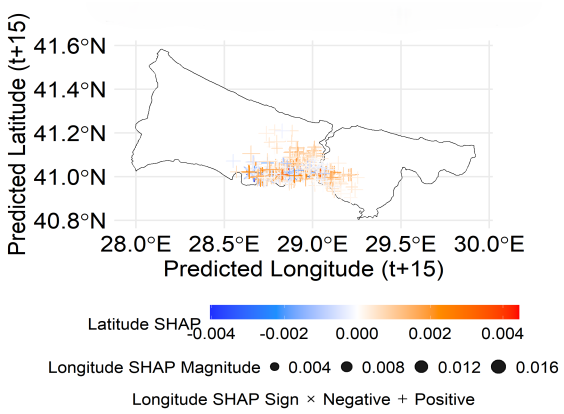
(d) Smartphone User D

Figure B.1: The spatial distribution of SHAP values for grid IDs visited at timestamp t is presented. The figure illustrates how grid-to-grid transitions with respect to timestamps influence the RFR model's predicted coordinates. The color intensity represents the magnitude and direction of the latitude SHAP values, where orange to red shades indicate positive contributions associated with northward movement, while light to dark blue shades indicate negative contributions corresponding to southward movement. The positive and negative SHAP values of longitude are represented by plus and cross symbols, respectively, while the magnitude of these values is indicated by the size of the markers. The cross symbol represents westward movement, while the plus symbol denotes eastward movement.

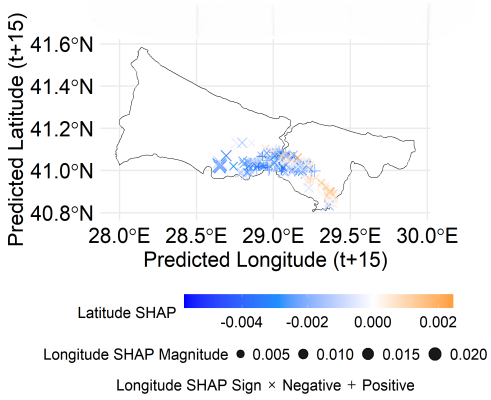
The Figure B.1 shows the SHAP analysis of how the grid ID visited by smartphone users influences the predicted coordinates, and therefore the user's movement. For user A, the northern part of the city displays prominent plus and cross symbols with strong red color intensity, and it indicates high positive latitude SHAP values. This suggests that the grid IDs visited at timestamp t in this region contribute strongly to northward movement predictions. Toward the central area, the color intensity decreases to orange, and the longitude marker sizes become smaller, reflecting a lower contribution of these visited grids to both latitude and longitude predictions. In the eastern region, the latitude SHAP values are slightly negative with light blue coloration, implying a weak southward effect, while the

longitude symbols remain of moderate size. For users B and C, the northern part of the city displays large plus symbols representing positive longitude SHAP values, accompanied by high-intensity orange coloration indicating strong positive latitude SHAP values. This suggests that grid IDs visited in this region contribute notably to both eastward and northward movement. Moving toward the central area, the color intensity decreases to a milder orange, and the longitude symbols transition to crosses, indicating a shift toward negative SHAP values and reduced eastward influence. In the eastern region, the latitude SHAP values become slightly negative with light blue tones, turning into darker blue in the southeastern area, and it reflects a stronger southward contribution. The longitude SHAP symbols in these regions vary in size, signifying differences in their impact on east-west movement. For user D, the western part of the city displays varying longitude symbols as both plus and cross markers, indicating positive and negative SHAP values. In this region, the latitude SHAP values appear in shades of orange, suggesting a positive contribution toward northward movement. Moving further toward the east, the color intensity of the latitude SHAP values gradually decreases, and the sizes of the longitude symbols become smaller and more variable, and it indicate a reduced influence on east–west movement. A similar pattern is observed in the northern areas. In contrast, the southern region exhibits negative latitude SHAP values shown in light blue, which intensify to darker blue toward the southeast, and it exhibits a stronger southward contribution. In this southeastern area, the positive longitude SHAP symbols are also larger, indicating a stronger eastward influence. In general, these spatial patterns indicate that the grid IDs visited at timestamp t play a key role in shaping the predicted direction of users' movement, with the color intensity and symbol type effectively representing the magnitude and direction of their influence.

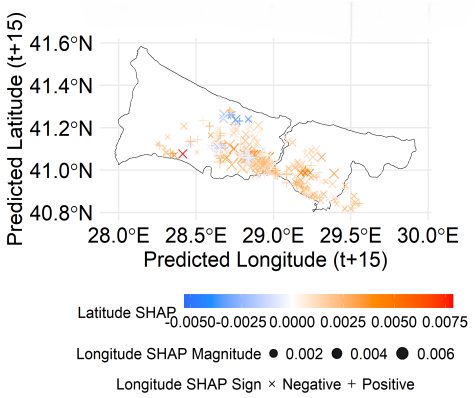
The empirical grid transition probabilities represent the likelihood of a user moving from one geographic area to another based on historical movement patterns. The spatial variability of the empirical grid transition probabilities is shown in Figure B.2, showing how their influence differs across regions of Istanbul.



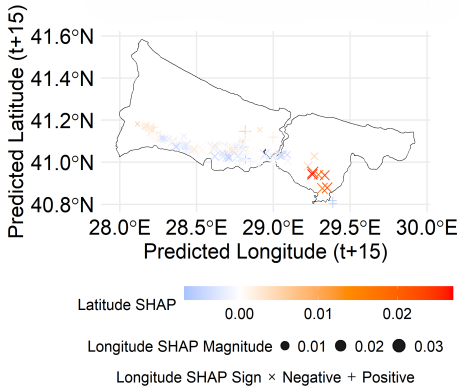
(a) Smartphone User A



(b) Smartphone User B



(c) Smartphone User C



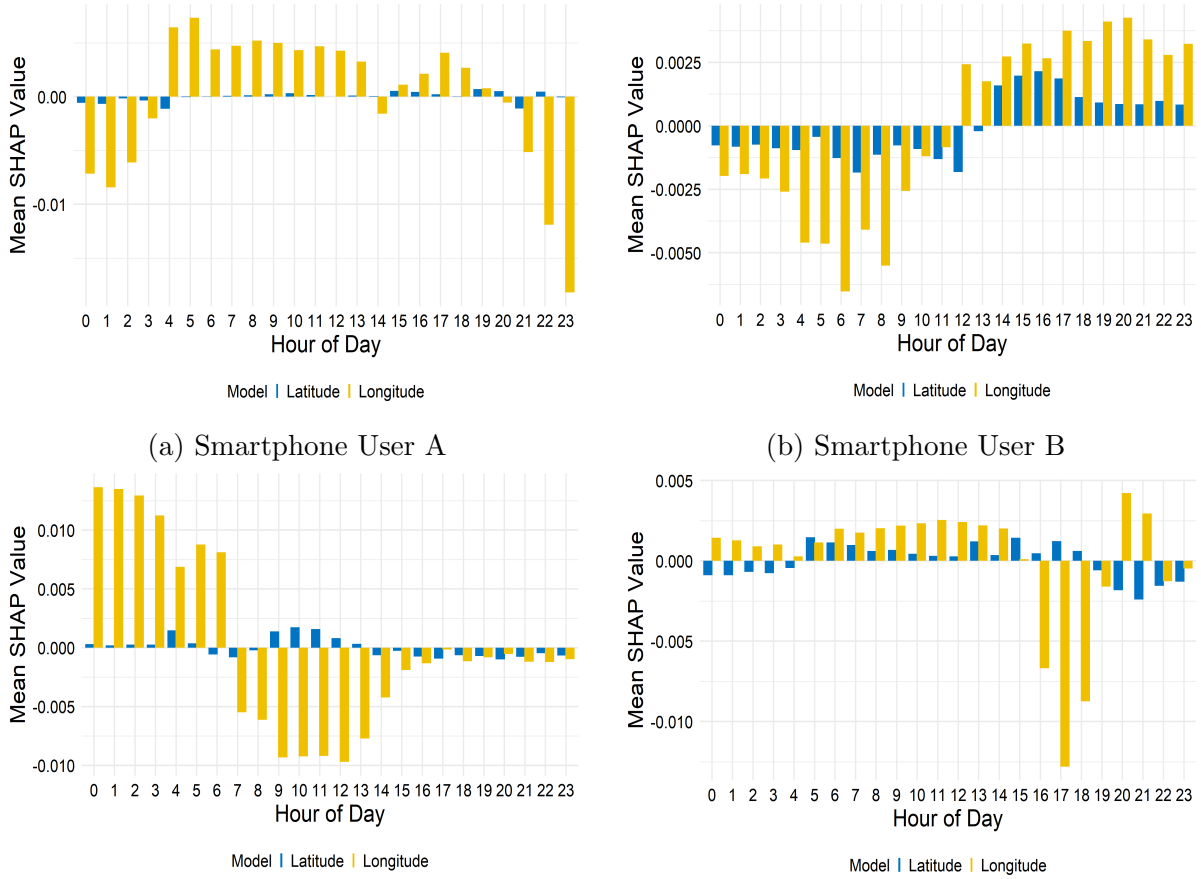
(d) Smartphone User D

Figure B.2: The spatial distribution of SHAP values for the empirical grid transition probability feature is shown. The figure displays how historical grid-to-grid transition probabilities influence the RFR model's predicted coordinates. The color intensity reflects the magnitude and direction of latitude SHAP values, where orange to red shades correspond to positive contributions indicating northward movement, and light to dark blue shades represent negative contributions associated with southward movement. The positive and negative SHAP values for longitude are depicted by plus and cross symbols, respectively. While the magnitude of values is depicted by the size of the markers. The cross symbol represents westward movement, while the plus symbol denotes eastward movement.

It can be seen in Figure B.2 that the latitude SHAP values show spatial variation across different parts of Istanbul for user A, as the positive and negative values reveal northward and southward movement tendencies, respectively. Similarly, the longitude SHAP values are portrayed by plus and cross symbols, representing positive and negative effects, respectively. These vary regionally, reflecting eastward and westward directional influences. These patterns are driven by the empirical historical transition probabilities between visited grids at successive timestamps, which serve as a key feature in the model to capture the likelihood of movement from one geographical area to another. For user B, the intense blue shades in the central, western, and southern areas highlight strongly negative latitude

SHAP values, indicating that historical grid transitions in these regions are linked to southward movement, which lowers the predicted latitude. Meanwhile, the larger cross symbols denote negative longitude SHAP values, signaling a westward influence that weakens near the central area. In the southeastern region, the mild orange colors for latitude SHAP combined with cross symbols for longitude suggest a pattern of northward movement accompanied by westward shifts. For user C, the positive latitude SHAP values are found throughout Istanbul, indicating a general tendency for movement toward the north. In some areas, these positive values are stronger, which means that those locations have a greater influence on predicting northward movement. The negative latitude SHAP values are shown by the light blue shade, and these appear mainly in the northern and western-central parts of the city. These suggest localized tendencies for movement toward the south in those regions. The longitude SHAP values show both positive and negative effects, and their strength varies depending on the location. This reflects complex patterns of eastward and westward movement. Overall, these differences in SHAP values indicate how the historical probabilities of moving between grids affect the directional predictions of the model differently across the city. In the southeastern area, the latitude SHAP values are strongly positive, and these are highlighted by deep red tones. These indicate a notable northward movement tendency for user D. At the same time, the longitude SHAP values are negative in this region, and depicted by prominent cross symbols. These suggest a shift toward the west. In contrast, the western and southern parts exhibit negative latitude SHAP values with lighter blue shades, reflecting tendencies for southward movement. The longitude SHAP values in these regions vary, showing both positive and negative effects, which correspond to mixed influences in east-west directions. Collectively, the historical transition probabilities play a key role in shaping the model's understanding of directional movements. Although their effects vary due to differences in user behavior.

The hour of day is treated as a discrete temporal feature in the model to capture its overall effect on predictions. Understanding this influence is important because it reflects how behavior varies periodically throughout the day. Therefore, the average SHAP values for the hour of day feature are shown in Figure B.3.



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

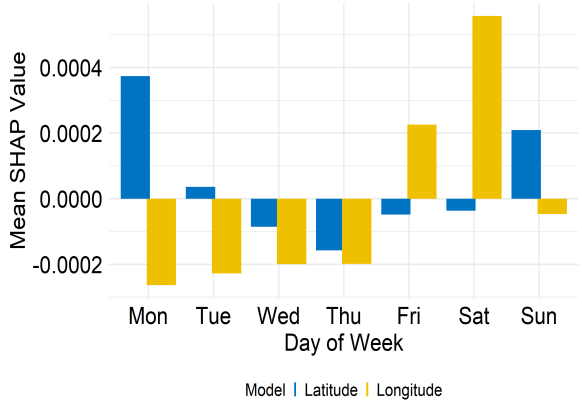
Figure B.3: The mean SHAP values for the hour of day feature are presented, illustrating its influence on the coordinates predicted by the RFR model. The blue color represents the effect on latitude, where positive values indicate a northward shift and negative values represent a southward shift. The yellow color represents the effect on longitude as positive values indicating eastward movement and negative values indicating westward movement. This visualization highlights how the hour of day impacts directional movement tendencies.

From Figure B.3, the mean SHAP values of longitude are negative between 00:00 and 03:00 hours for user A, and these indicate westward directional influence during night hours. Between 05:00 and 13:00 hours, the SHAP values become positive, reflecting an eastward movement tendency that persists again between 15:00 and 20:00 hours. The values turn strongly negative from 21:00 to 23:00 hours, and these suggest a pronounced westward shift during late evening. In contrast, the latitude mean SHAP values remain slightly negative between 00:00 and 03:00 hours, and only exhibit weak positive effects between 15:00 and 20:00 hours, denoting limited north-to-south variation and overall low sensitivity to hourly change. For user B, both the latitude and longitude SHAP values are negative between 00:00 and 12:00 hours, and these indicate a predominant westward and southward influence during that period. The longitude SHAP magnitudes are consistently higher than

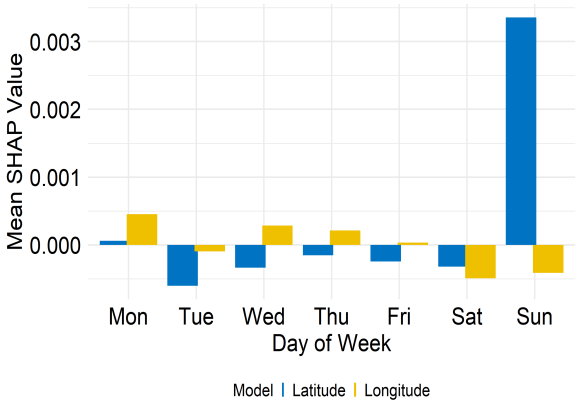
those of latitude, signifying a stronger east-west sensitivity in model predictions. During the afternoon and evening hours from 13:00 to 23:00, both the latitude and longitude SHAP values become positive, reflecting a directional shift toward northward and eastward predictions, while the longitude SHAP again exerts the dominant influence. For user C, the longitude SHAP values are strongly positive from 00:00 to 06:00 hours, reflecting a consistent eastward influence on the model's predicted longitude, while the latitude SHAP values remain slightly positive but with comparatively low magnitude. Following this, from 06:00 to 23:00 hours, the longitude SHAP values become negative, denoting a westward influence, while the latitude SHAP values remain weakly positive between 09:00 and 13:00 hours, and this suggests a subtle northward contribution during mid-day. The pronounced negative longitude SHAP values in this period suggest dominant westward shifts in predicted positions. For user D, the average longitude SHAP values show a slight positive influence from 00:00 to 14:00, indicating a mild tendency toward eastward movement during this period. However, the values shift to strongly negative between 16:00 and 18:00 hours, reflecting a pronounced westward direction. This is followed by a return to positive values with moderate strength from 20:00 to 21:00, suggesting a short eastward movement again. Regarding latitude, the average SHAP values remain positive from 05:00 to 19:00 hours, indicating a general northward tendency, while the remaining hours show negative values, and these suggest southward movement. Overall, these latitude values are relatively small, indicating only a subtle influence on the predicted direction. Across all users, the SHAP-based temporal interpretation reveals distinct and individualized diurnal mobility patterns. The average SHAP values of longitude generally exhibit stronger temporal variability and have higher magnitudes than latitude SHAP values. This indicates that east-west movements are more time-dependent than north-south shifts.

Similarly, the average SHAP values for each day of the week are shown in Figure B.4. It is noticed in Figure B.4 that the longitude mean SHAP values are negative on Monday, Tuesday, Wednesday, Thursday, and Sunday for user A, and this indicates that these days are associated with westward tendencies in the predicted longitude. Conversely, the SHAP values are positive on Friday and Saturday, suggesting a shift toward eastward movement during the weekend. The latitude SHAP values are positive only on Monday and Sunday, suggesting a limited northward influence on those days, while the SHAP values are negative or near-zero for the rest of the week. Although the longitude SHAP value on Sunday is

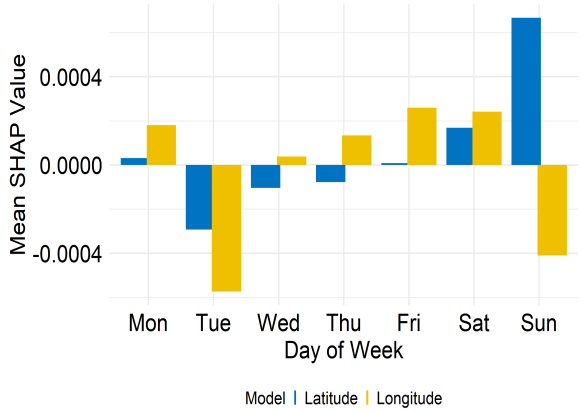
negative, its magnitude is relatively low, and it reflects a weaker directional effect.



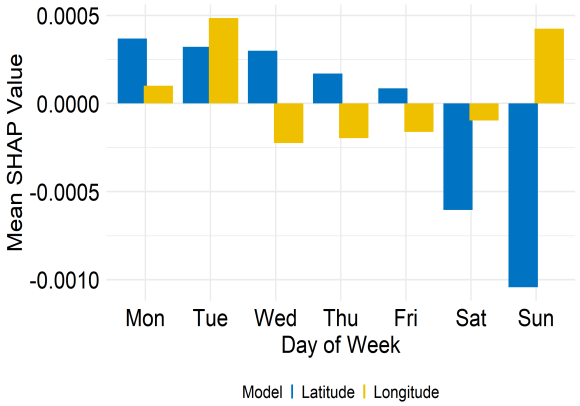
(a) Smartphone User A



(b) Smartphone User B



(c) Smartphone User C



(d) Smartphone User D

Figure B.4: The mean SHAP values for the day of week feature are presented, illustrating its influence on the coordinates predicted by the RFR model. The blue color represents the effect on latitude, where positive values indicate a northward shift and negative values represent a southward shift. The yellow color represents the effect on longitude as positive values indicating eastward movement and negative values indicating westward movement. This visualization highlights how the day of the week impacts directional movement tendencies and contributes to the model's coordinate predictions.

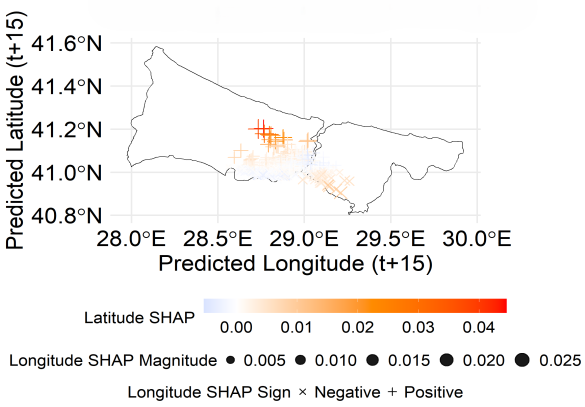
For user B, the mean latitude SHAP values are strongly positive on Sunday, indicating a northward influence on that day, while they remain negative with low magnitudes throughout the rest of the week. These suggest minimal directional change. The longitude SHAP values fluctuate slightly between positive and negative across the week; however, their magnitudes remain small, reflecting minimal variation in longitudinal behavior across different days of the week. For user C, the longitude SHAP values are negative on Tuesday and Sunday with high magnitudes, signifying pronounced westward effects on these days, whereas on Friday and Saturday, they become strongly positive, and these indicate a substantial eastward influence. The latitude SHAP values are positive for both Saturday and

Sunday, but the peak magnitude is observable on Sunday, which signifies intensified northward directional influence across the weekend. During the rest of the week, the latitude SHAP values are negative and vary slightly from day to day, reflecting subtle but consistent southward effects. For user D, the latitude SHAP values are negative on Saturday and Sunday. The effect is stronger on Sunday than on Saturday. During the rest of the week, the values remain positive, indicating a general tendency toward northward movement on weekdays. The longitude SHAP values are positive with low magnitudes on Monday, Tuesday, and Sunday, but these are negative for the remaining days. Therefore, these indicate a tendency toward eastward predictions early in the week and westward tendencies in mid-week. Across all users, the SHAP analysis of the day of week feature reveals distinct spatial patterns that vary throughout the week, indicating a clear dependence on weekly routines that significantly influence the predicted coordinates.

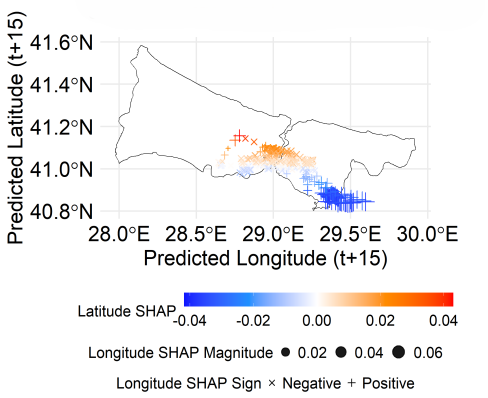
Ultimately, the local effects of spatial and temporal features on coordinate predictions in the RFR model reveal important dependency patterns. Although standard feature importance methods indicated a relatively low contribution from these features, SHAP analysis provides a clearer and more detailed explanation. This approach captures how spatial and temporal dependencies influence predictions at the individual level, with their impact varying among users depending on individual mobility patterns.

SHAP Analysis for SVR and MLPR Models

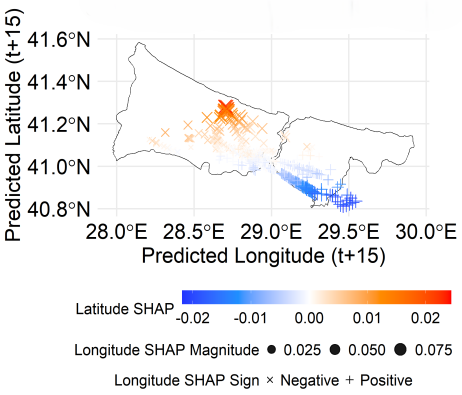
This sections provides comprehensive visualizations of SHAP values that quantify the contribution of features to the prediction of smartphone users' coordinates. These analyses are presented for two models such as SVR and MLPR. By comparing SHAP values across models and individual users, the Figure B.5, B.6, B.7, B.8, B.9, B.10, B.11, and B.12 highlight the divergence in how each model interprets mobility dynamics. The SVR and MLPR models exhibit variations in the magnitude and direction of feature contributions, revealing nuanced insights into user behavior and model decision-making processes.



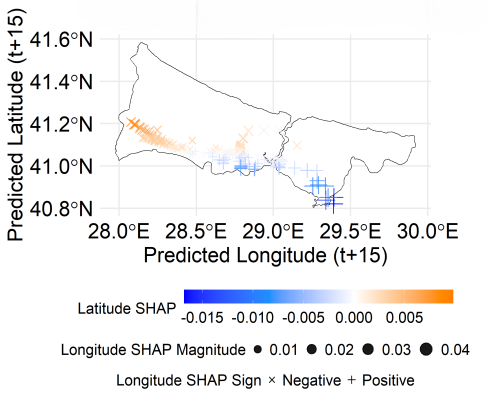
(a) Smartphone User A



(b) Smartphone User B

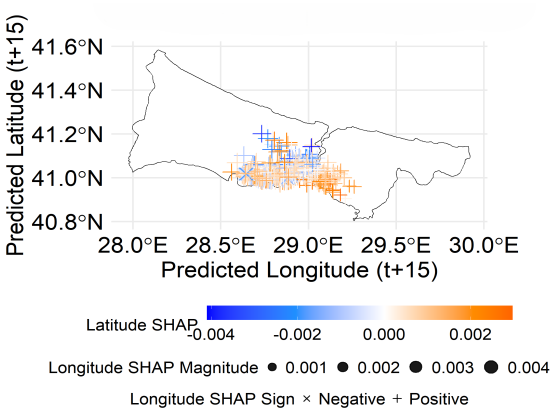


(c) Smartphone User C

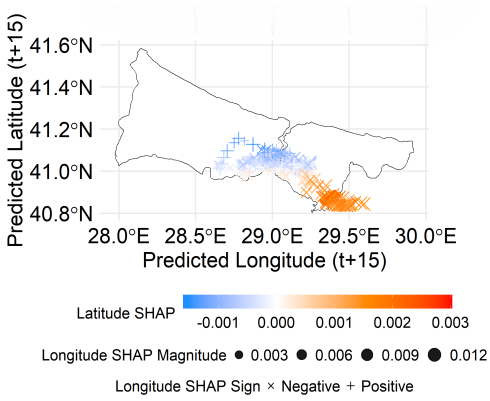


(d) Smartphone User D

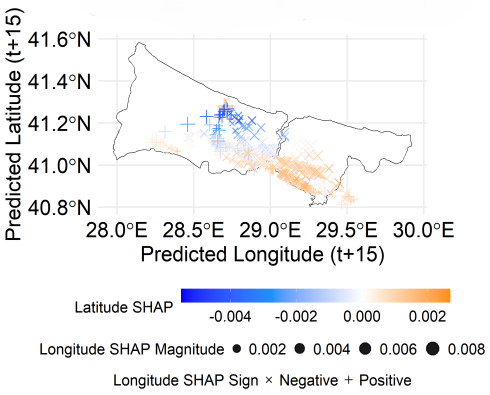
Figure B.5: The spatial distribution of SHAP values for grid IDs visited at timestamp t is presented. The figure shows how grid-to-grid transitions with respect to timestamps influence the SVR model's predicted coordinates. The color intensity represents the magnitude and direction of the latitude SHAP values, where orange to red shades indicate positive contributions associated with northward movement, while light to dark blue shades indicate negative contributions corresponding to southward movement. The positive and negative SHAP values of longitude are represented by plus and cross symbols, respectively, while the magnitude of these values is indicated by the size of the markers. The cross symbol represents westward movement, while the plus symbol denotes eastward movement.



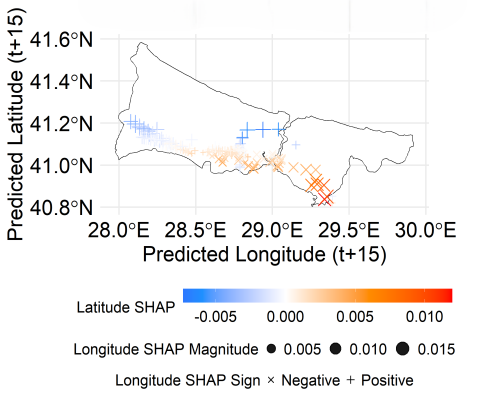
(a) Smartphone User A



(b) Smartphone User B

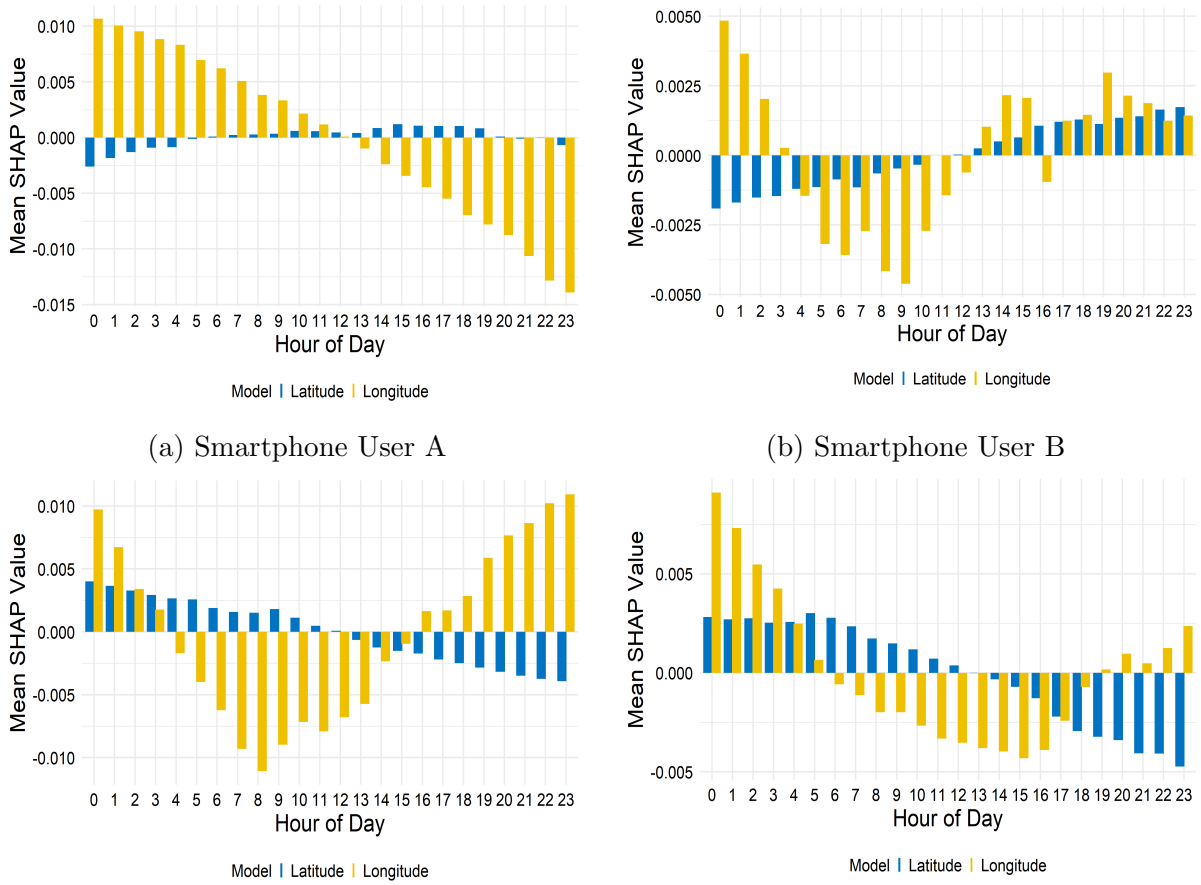


(c) Smartphone User C



(d) Smartphone User D

Figure B.6: The spatial distribution of SHAP values for the empirical grid transition probability feature is shown. The figure illustrate how historical grid-to-grid transition probabilities influence the SVR model's predicted coordinates. The color intensity reflects the magnitude and direction of latitude SHAP values, where orange to red shades correspond to positive contributions indicating northward movement, and light to dark blue shades represent negative contributions associated with southward movement. The positive and negative SHAP values for longitude are depicted by plus and cross symbols, respectively. While the magnitude of values is depicted by the size of the markers. The cross symbol represents westward movement, while the plus symbol denotes eastward movement.



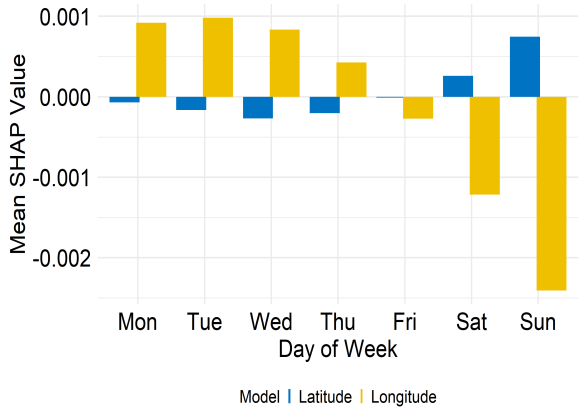
(a) Smartphone User A

(b) Smartphone User B

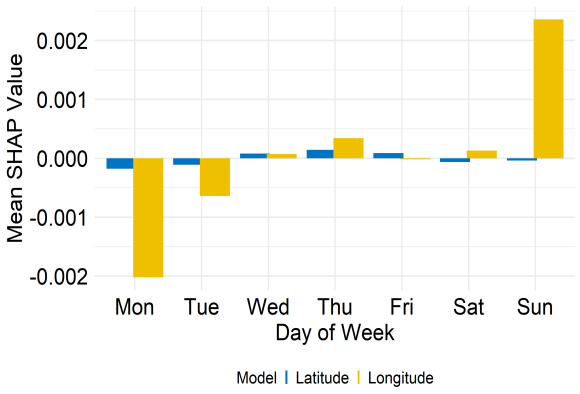
(c) Smartphone User C

(d) Smartphone User D

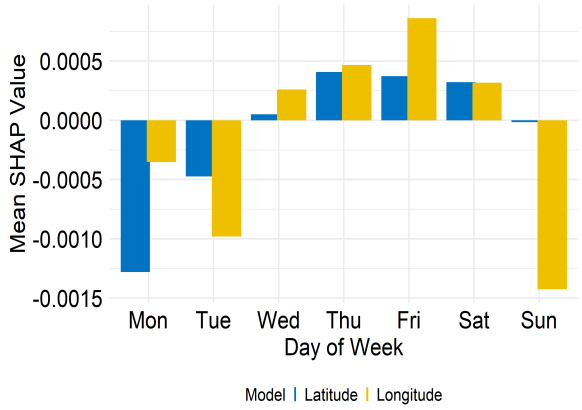
Figure B.7: The mean SHAP values for the hour of day feature are presented, illustrating its influence on the coordinates predicted by the SVR model. The blue color represents the effect on latitude, where positive values indicate a northward shift and negative values represent a southward shift. The yellow color represents the effect on longitude as positive values indicating eastward movement and negative values indicating westward movement. This visualization highlights how the hour of day impacts directional movement tendencies.



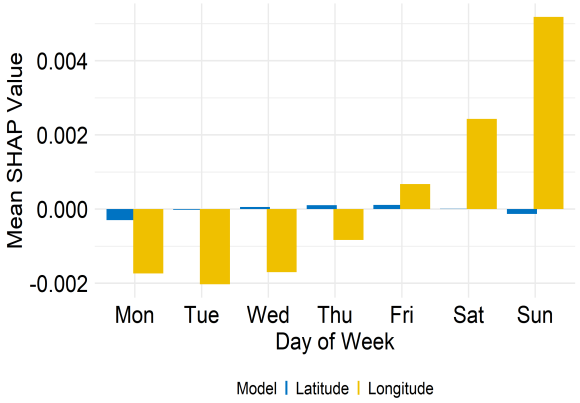
(a) Smartphone User A



(b) Smartphone User B

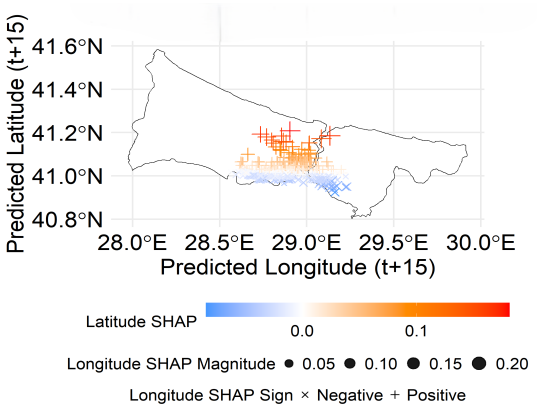


(c) Smartphone User C

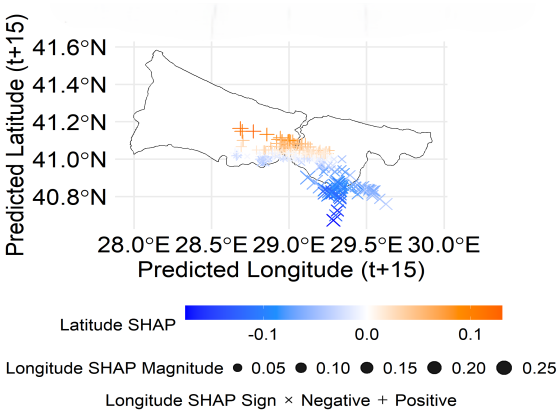


(d) Smartphone User D

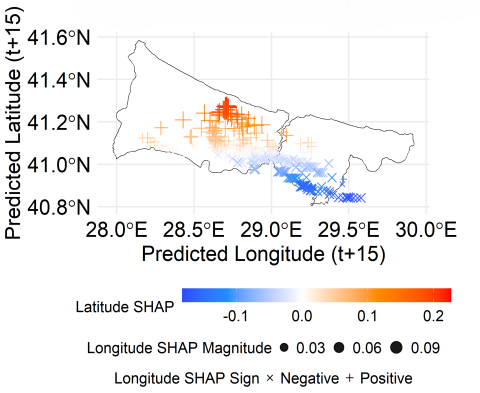
Figure B.8: The mean SHAP values for the day of week feature are presented, illustrating its influence on the coordinates predicted by the SVR model. The blue color represents the effect on latitude, where positive values indicate a northward shift and negative values represent a southward shift. The yellow color represents the effect on longitude as positive values indicating eastward movement and negative values indicating westward movement. This visualization highlights how the day of the week impacts directional movement tendencies and contributes to the model's coordinate predictions.



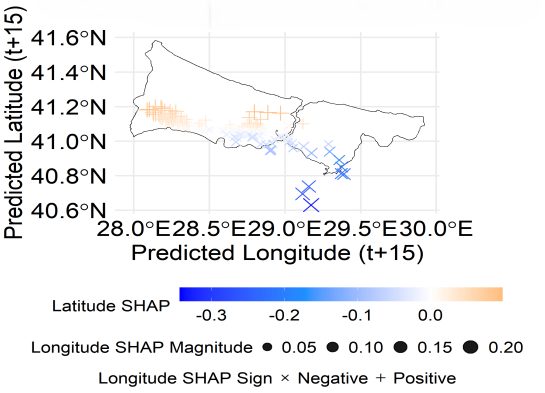
(a) Smartphone User A



(b) Smartphone User B

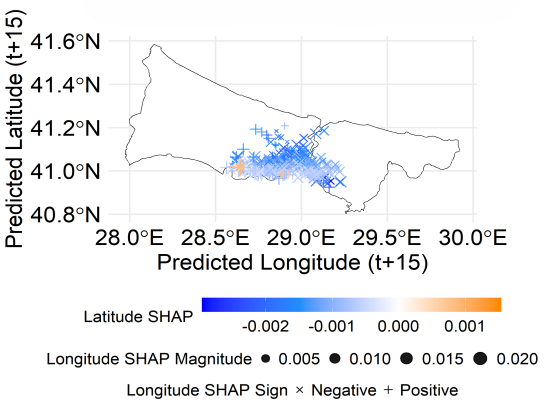


(c) Smartphone User C

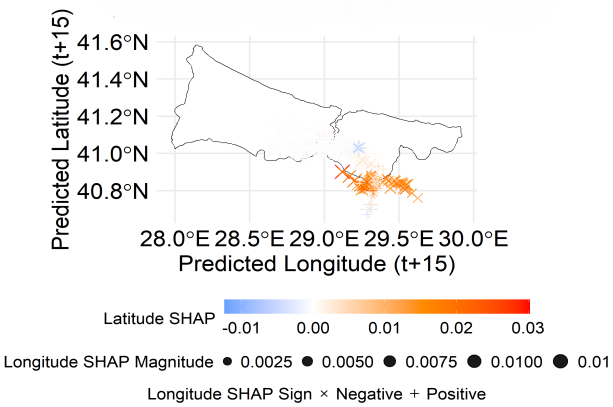


(d) Smartphone User D

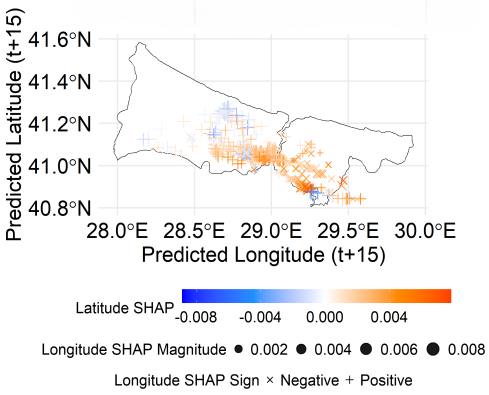
Figure B.9: The spatial distribution of SHAP values for grid IDs visited at timestamp t is presented. The figure displays how grid-to-grid transitions with respect to timestamps influence the MLPR model's predicted coordinates. The color intensity represents the magnitude and direction of the latitude SHAP values, where orange to red shades indicate positive contributions associated with northward movement, while light to dark blue shades indicate negative contributions corresponding to southward movement. The positive and negative SHAP values of longitude are represented by plus and cross symbols, respectively, while the magnitude of these values is indicated by the size of the markers. The cross symbol represents westward movement, while the plus symbol denotes eastward movement.



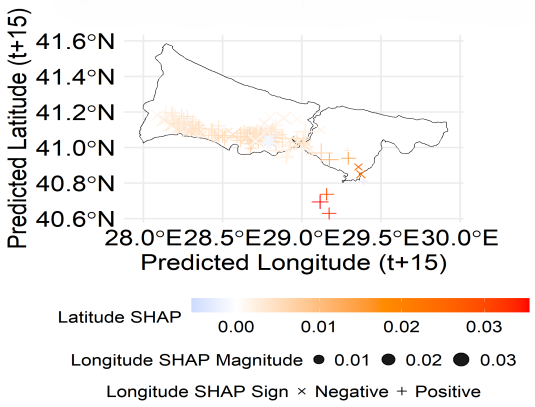
(a) Smartphone User A



(b) Smartphone User B

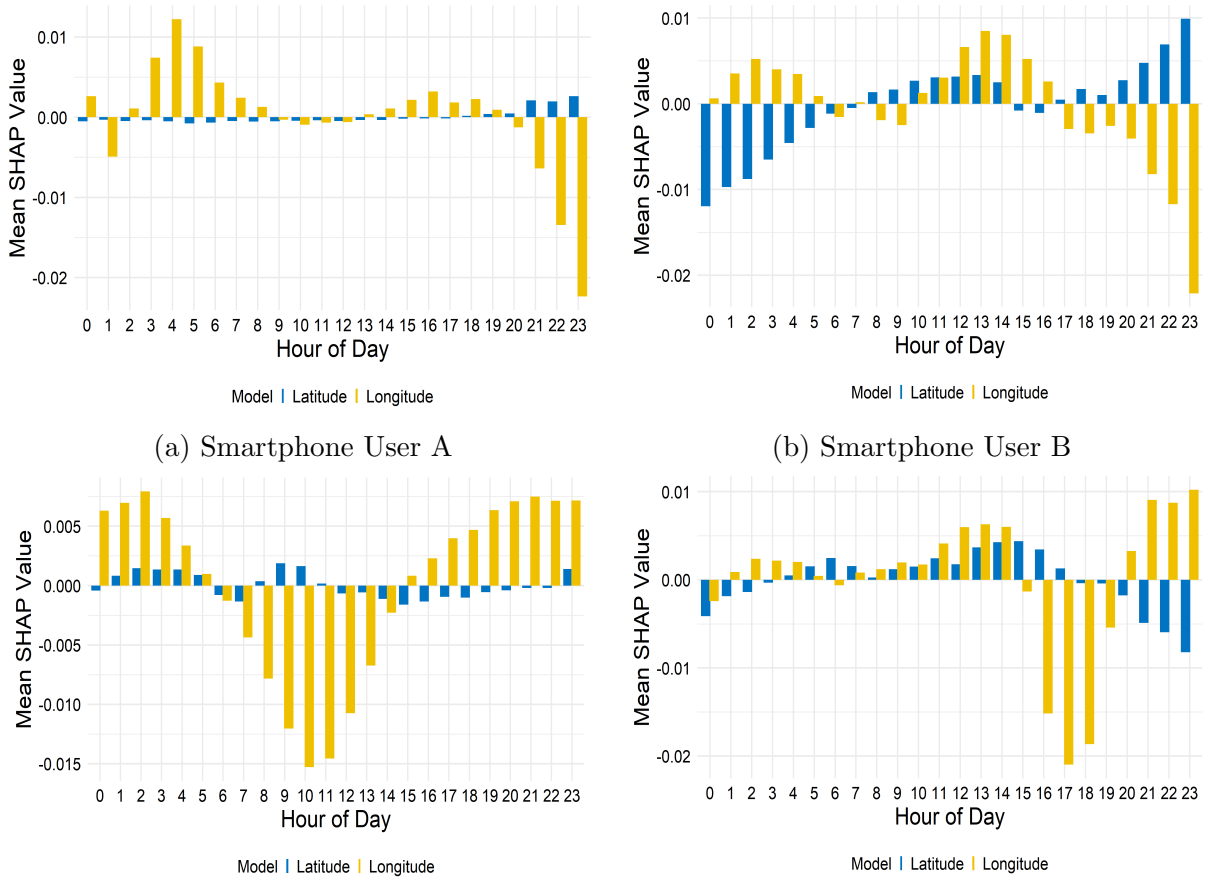


(c) Smartphone User C



(d) Smartphone User D

Figure B.10: The spatial distribution of SHAP values for the empirical grid transition probability feature is shown. The figure illustrates how historical grid-to-grid transition probabilities influence the MLPR model's predicted coordinates. The color intensity reflects the magnitude and direction of latitude SHAP values, where orange to red shades correspond to positive contributions indicating northward movement, and light to dark blue shades represent negative contributions associated with southward movement. The positive and negative SHAP values for longitude are depicted by plus and cross symbols, respectively. While the magnitude of values is depicted by the size of the markers. The cross symbol represents westward movement, while the plus symbol denotes eastward movement.



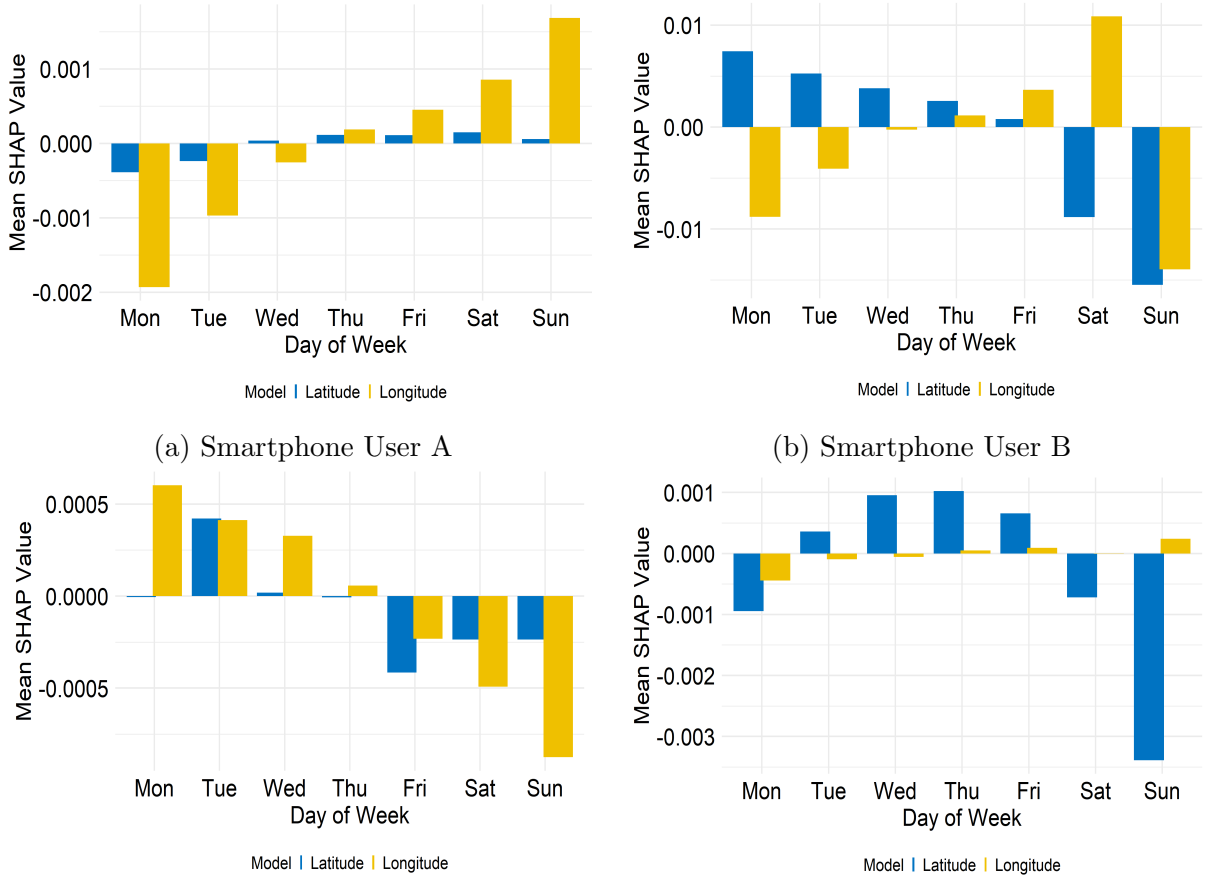
(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure B.11: The mean SHAP values for the hour of day feature are presented, illustrating its influence on the coordinates predicted by the MLPR model. The blue color represents the effect on latitude, where positive values indicate a northward shift and negative values represent a southward shift. The yellow color represents the effect on longitude as positive values indicating eastward movement and negative values indicating westward movement. This visualization highlights how the hour of day impacts directional movement tendencies.



(a) Smartphone User A

(b) Smartphone User B

(c) Smartphone User C

(d) Smartphone User D

Figure B.12: The mean SHAP values for the day of week feature are presented, illustrating its influence on the coordinates predicted by the MLPR model. The blue color represents the effect on latitude, where positive values indicate a northward shift and negative values represent a southward shift. The yellow color represents the effect on longitude as positive values indicating eastward movement and negative values indicating westward movement. This visualization highlights how the day of the week impacts directional movement tendencies and contributes to the model's coordinate predictions.

B.3 Effect of Historical Information Inclusion on Location Prediction Models

This supplementary analysis evaluates the impact of incorporating additional historical information into coordinate prediction models. Two model configurations are considered, and the details are as follows;

Location Prediction Models

In Chapter 5, the coordinate prediction models use information available up to timestamp t, including the user's latitude and longitude at timestamp t, together with discretized spatial features, temporal attributes, and transition-related features. This formulation adopts a Markov approximation, in which the user's location at timestamp $t+15$ minutes is modeled as a function of the location at timestamp t and other contextual features. We emphasize that this formulation represents a practical modeling approximation rather than a strict assumption about the underlying mobility process.

History Augmented Location Prediction Models

The coordinate prediction models are further extended by incorporating additional historical information in the form of the user's latitude and longitude observed at timestamp $t-15$ minutes. No other changes are made to the feature set.

For both configurations, the prediction target is the user's location at timestamp $t+15$ minutes. Importantly, the same training and evaluation procedure was adopted for all models and configurations, including identical data partitions, hyperparameter tuning strategies, and evaluation metrics. This ensures that observed performance differences are attributable solely to the inclusion of additional historical information rather than differences in model training.

B.3.1 Δ Metric Definition

To quantify the effect of historical information, we define the difference metric Δ as,

$$\Delta(r) = Accuacy(Y_t, Y_{t-15})^{(r)} - Accuracy(Y_t)^{(r)}, \tag{B.14}$$

where $Accuacy(.)^{(r)}$ represents the accuracy within a threshold radius r, defined as the proportion of predictions falling within a radius of the true location. The formal definition

of this metric is given in Section 5.2.4. A positive Δ value indicates that incorporating historical coordinates observed at timestamp $t-15$ improves prediction accuracy within a radius, whereas a negative value indicates superior performance of the model, which includes coordinates information up to timestamp t.

B.3.2 Results and Interpretation

The Δ analysis was performed using the RFR, SVR, and MLPR models, with prediction accuracy evaluated across spatial radii ranging from 10 to 1000 meters. Results are reported separately for four smartphone users. The objective of this analysis is to empirically assess whether incorporating immediate historical information (i.e., coordinates at timestamp $t-15$) yields meaningful performance improvements compared to models that rely solely on information available up to timestamp t. The accuracy values within each radius threshold for both model configurations are summarized in Table B.1. Additionally, the Δ analysis, which represents the difference in performance with respect to the model and radius threshold, is described in Table B.2. From Table B.2, the Δ values observed for

Table B.1: The table presents accuracy values with respect to radii in meters. M1 represents models that use coordinate information up to timestamp t, while M2 refers to models augmented with historical coordinates up to timestamp $t-15$.

Smartphone User		**M2**				**M1**			
Model	**Radius**	**A**	**B**	**C**	**D**	**A**	**B**	**C**	**D**
RFR	10	0.2452	0.3728	0.3637	0.1339	0.2443	0.3632	0.3767	0.1713
	50	0.4017	0.4704	0.4627	0.3591	0.3913	0.4660	0.4792	0.4104
	100	0.4470	0.5078	0.5035	0.5461	0.4374	0.4904	0.5252	0.5557
	200	0.4861	0.5409	0.5816	0.6583	0.4739	0.5209	0.5712	0.6539
	300	0.5139	0.5662	0.6024	0.7183	0.4974	0.5383	0.5911	0.7052
	400	0.5417	0.5976	0.6250	0.7522	0.5157	0.5584	0.6076	0.7322
	500	0.5687	0.6228	0.6424	0.7843	0.5322	0.5767	0.6215	0.7609
	750	0.6417	0.6803	0.6858	0.8270	0.5704	0.6211	0.6450	0.7939
	1000	0.6861	0.7256	0.7196	0.8548	0.6096	0.6507	0.6736	0.8191
SVR	10	0.0009	0.0000	0.0000	0.0000	0.0009	0.0000	0.0000	0.0000
	50	0.0252	0.0148	0.0061	0.0000	0.0122	0.0218	0.0087	0.0096
	100	0.0626	0.0427	0.0061	0.0070	0.0504	0.0557	0.0165	0.0157
	200	0.1165	0.1211	0.0547	0.1461	0.1383	0.1542	0.0686	0.0583
	300	0.2539	0.1951	0.1554	0.2722	0.2391	0.2666	0.1354	0.1504
	400	0.3600	0.2605	0.2413	0.4035	0.3400	0.3868	0.2040	0.2713
	500	0.4417	0.3423	0.3550	0.4974	0.3809	0.5113	0.2821	0.4165
	750	0.5574	0.5793	0.5764	0.6330	0.4504	0.6333	0.4349	0.6183
	1000	0.6443	0.7474	0.6884	0.7374	0.5270	0.6672	0.5660	0.7522
MLPR	10	0.0035	0.0061	0.0000	0.0078	0.0000	0.0000	0.0000	0.0000
	50	0.1043	0.1611	0.0009	0.0496	0.0096	0.0688	0.0000	0.0078
	100	0.3157	0.3929	0.0009	0.1217	0.0765	0.1847	0.0017	0.0261
	200	0.4409	0.4704	0.0122	0.3365	0.2817	0.4077	0.0434	0.1348
	300	0.4704	0.5105	0.0625	0.4913	0.4035	0.4791	0.1215	0.2722
	400	0.4974	0.5444	0.1745	0.5843	0.4504	0.5148	0.2752	0.4278
	500	0.5304	0.5653	0.4479	0.6322	0.4687	0.5296	0.4045	0.5643
	750	0.5878	0.6098	0.6632	0.7165	0.5217	0.5740	0.5425	0.7157
	1000	0.6548	0.6568	0.7378	0.7713	0.5591	0.6124	0.6276	0.7391

Table B.2: This table shows the difference in accuracy (Δ) across various radii between models augmented with historical coordinates up to timestamp $t-15$ and models using coordinates only up to timestamp t. The comparison is conducted for RFR, SVR, and MLPR models evaluated individually for smartphone users. Positive Δ values indicate improved performance when incorporating historical information, while negative Δ values reflect better accuracy of models relying solely on coordinates observed up to timestamp t.

Smartphone User		Δ			
Model	**Radius**	**A**	**B**	**C**	**D**
RFR	10	0.001	0.010	-0.013	-0.037
	50	0.010	0.004	-0.016	-0.051
	100	0.010	0.017	-0.022	-0.010
	200	0.012	0.020	0.010	0.004
	300	0.017	0.028	0.011	0.013
	400	0.026	0.039	0.017	0.020
	500	0.037	0.046	0.021	0.023
	750	0.071	0.059	0.041	0.033
	1000	0.077	0.075	0.046	0.036
SVR	10	0.000	0.000	0.000	0.000
	50	0.013	-0.007	-0.003	-0.010
	100	0.012	-0.013	-0.010	-0.009
	200	-0.022	-0.033	-0.014	0.088
	300	0.015	-0.071	0.020	0.122
	400	0.020	-0.126	0.037	0.132
	500	0.061	-0.169	0.073	0.081
	750	0.107	-0.054	0.141	0.015
	1000	0.117	0.080	0.122	-0.015
MLPR	10	0.003	0.006	0.000	0.008
	50	0.095	0.092	0.001	0.042
	100	0.239	0.208	-0.001	0.096
	200	0.159	0.063	-0.031	0.202
	300	0.067	0.031	-0.059	0.219
	400	0.047	0.030	-0.101	0.157
	500	0.062	0.036	0.043	0.068
	750	0.066	0.036	0.121	0.001
	1000	0.096	0.044	0.110	0.032

both the RFR and SVR models are close to zero and occasionally negative at fine radii ranging from 10 to 100 meters. This indicates that incorporating location information up to the timestamp $t-15$ does not significantly enhance fine-grained localization accuracy. These findings suggest that short-term mobility of smartphone users is largely determined by their current coordinates at timestamp t, thereby supporting the applicability of a first-order Markov approximation for short-range predictions. The MLPR model shows slightly greater sensitivity to historical information at these small radii, likely due to its higher representational capacity. However, the improvements observed with MLPR remain inconsistent and vary between users. At intermediate radii between 200 and 400 meters, Δ values exhibit greater variability across both users and models. While some users ex-

perience performance gains from including historical location information, others do not benefit noticeably. This heterogeneous behavior suggests the existence of weak higher-order temporal dependencies that depend on individual mobility patterns and the specific capabilities of the predictive models employed. At broader radii, equal to or exceeding 500 meters, the Δ values are predominantly positive across all examined models. This trend indicates that integrating historical coordinates up to the timestamp $t - 15$ effectively reduces large spatial errors. Such improvements are indicative of enhanced robustness in coarse-grained localization rather than better precision in point-level prediction accuracy. Overall, the results indicate that smartphone user mobility does not strictly adhere to the first-order Markov approximation across all spatial scales. Nonetheless, the negligible performance differences observed at fine spatial resolutions justify the use of a first-order Markov approximation for short-term mobility prediction. As the radius increases, the incorporation of additional historical information becomes increasingly beneficial, thereby underscoring the limitations of Markovian models at broader spatial radii.

List of Figures

List of Tables

References

Abideen, Z. U., Sun, H., Yang, Z., Ahmad, R. Z., Iftekhar, A., and Ali, A. (2020). Deep wide spatial-temporal based transformer networks modeling for the next destination according to the taxi driver behavior prediction. *Applied Sciences*, 11(1):1–24.

Agirre-Basurko, E., Ibarra-Berastegi, G., and Madariaga, I. (2006). Regression and multilayer perceptron-based models to forecast hourly O3 and NO2 levels in the Bilbao area. *Environmental Modelling & Software*, 21(4):430–446.

Aiello, L., Argiento, R., Finazzi, F., and Paci, L. (2025). Survival modelling of smartphone trigger data in crowdsourced seismic monitoring: with applications to the 2023 pazarcik and 2019 ridgecrest earthquakes. *Journal of the Royal Statistical Society Series A: Statistics in Society*, pages 1–16.

Antoniou, C., Dimitriou, L., and Pereira, F. (2018). *Mobility patterns, big data and transport analytics: tools and applications for modeling*. Elsevier.

Araújo, F., Araújo, F., Machado, K., Rosário, D., Cerqueira, E., and Villas, L. A. (2020). Ensemble mobility predictor based on random forest and markovian property using lbsn data. *Journal of Internet Services and Applications*, 11:1–11.

Ashbrook, D. and Starner, T. (2002). Learning significant locations and predicting user movement with gps. In *Proceedings. Sixth International Symposium on Wearable Computers,*, pages 101–108. IEEE.

Ayumi, V. and Nurhaida, I. (2020). Prediction using Markov for determining location of human mobility. *International Journal of Information Science and Technology*, 4(1):1–6.

Azdy, R. A. and Darnis, F. (2020). Use of haversine formula in finding distance between temporary shelter and waste end processing sites. In *Journal of Physics: Conference Series*, volume 1500, pages 1–6. IOP Publishing.

Baratchi, M., Meratnia, N., Havinga, P. J., Skidmore, A. K., and Toxopeus, B. A. (2014). A hierarchical hidden semi-markov model for modeling mobility data. In *Proceedings of the 2014 ACM international joint conference on pervasive and ubiquitous computing*, pages 401–412.

Ben-Akiva, M. and Lerman, S. R. (2021). Disaggregate travel and mobility-choice models and measures of accessibility. In *Behavioural travel modelling*, pages 654–679. Routledge.

Bénard, C., Biau, G., Da Veiga, S., and Scornet, E. (2021a). Interpretable random forests via rule extraction. In *International conference on artificial intelligence and statistics*, pages 937–945. PMLR.

Bénard, C., Biau, G., Da Veiga, S., and Scornet, E. (2021b). Sirus: Stable and interpretable rule set for classification.

Bian, W., Cui, G., and Wang, X. (2020). A trajectory collaboration based map matching approach for low-sampling-rate gps trajectories. *Sensors*, 20(1–22):2057.

Biau, G. and Scornet, E. (2016). A random forest guided tour. *Test*, 25(2):197–227.

Bieler, M., Mukkamala, R. R., and Grønli, T.-M. (2022). A context-and trajectory-based destination prediction of public transportation users. *IEEE Intelligent Transportation Systems Magazine*, 15(1):300–317.

Bray, M. and Han, D. (2004). Identification of support vector machines for runoff modelling. *Journal of Hydroinformatics*, 6(4):265–280.

Breiman, L. (2001). Random forests. *Machine learning*, 45:5–32.

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (2017). *Classification and regression trees*. Routledge.

Calabrese, F., Di Lorenzo, G., and Ratti, C. (2010). Human mobility prediction based on individual and collective geographical preferences. In *13th international IEEE conference on intelligent transportation systems*, pages 312–317. IEEE.

Chapra, S. C., Canale, R. P., et al. (2011). *Numerical methods for engineers*, volume 1221. Mcgraw-hill New York.

Chekol, A. G. and Fufa, M. S. (2022). A survey on next location prediction techniques, applications, and challenges. *EURASIP Journal on Wireless Communications and Networking*, 2022(1):1–24.

Chen, J., Ma, L., and Xu, Y. (2015). Support vector machine based mobility prediction scheme in heterogeneous wireless networks. *Mathematical Problems in Engineering*, 2015(1):1–10.

Chen, R., Fung, B. C., Mohammed, N., Desai, B. C., and Wang, K. (2013). Privacy-preserving trajectory data publishing by local suppression. *Information Sciences*, 231:83–97.

Cheng, Y., Qiao, Y., and Yang, J. (2016). An improved Markov method for prediction of user mobility. In *2016 12th International Conference on Network and Service Management (CNSM)*, pages 394–399. IEEE.

Cybenko, G. (1989). Approximation by superpositions of a sigmoidal function. *Mathematics of control, signals and systems*, 2(4):303–314.

Danaf, M., Becker, F., Song, X., Atasoy, B., and Ben-Akiva, M. (2019). Online discrete choice models: Applications in personalized recommendations. *Decision Support Systems*, 119:35–45.

Díaz-Uriarte, R. and Alvarez de Andrés, S. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7:1–13.

Dobbs, R., Smit, S., Remes, J., Manyika, J., Roxburgh, C., and Restrepo, A. (2011). Urban world: Mapping the economic power of cities. Technical report, McKinsey Global Institute.

Du, Y., Aoki, T., and Fujiwara, N. (2025). A review of human mobility: Linking data, models, and real-world applications. *Journal of Computational Social Science*, 8(4):1–90.

Elfeki, A. and Dekking, M. (2001). A Markov chain model for subsurface characterization: theory and applications. *Mathematical geology*, 33:569–589.

Feng, J., Li, Y., Zhang, C., Sun, F., Meng, F., Guo, A., and Jin, D. (2018). Deepmove: Predicting human mobility with attentional recurrent networks. In *Proceedings of the 2018 world wide web conference*, pages 1459–1468.

Finazzi, F. (2016). How a smartphone network detects earthquakes in real time. *Significance*, 13(6):6–7.

Finazzi, F., Bossu, R., and Cotton, F. (2024). Smartphones enabled up to 58 s strong-shaking warning in the m7. 8 türkiye earthquake. *Scientific Reports*, 14(1):1–11.

Finazzi, F. and Fassò, A. (2014). Earthquake monitoring using volunteer smartphone-based sensor networks. In *Proceedings of the METMA VII and GRASPA14 Conference. Torino (IT)*.

Finazzi, F. and Massoda Tchoussi, F. Y. (2024). Assessing the alerting capabilities of the earthquake network early warning system in haiti with monte carlo simulations. *Stochastic Environmental Research and Risk Assessment*, 38(1):147–156.

Funahashi, K.-I. (1989). On the approximate realization of continuous mappings by neural networks. *Neural networks*, 2(3):183–192.

Gambs, S., Killijian, M.-O., and del Prado Cortez, M. N. (2010). Show me how you move and i will tell you who you are. In *Proceedings of the 3rd ACM SIGSPATIAL International Workshop on Security and Privacy in GIS and LBS*, pages 34–41.

Gambs, S., Killijian, M.-O., and del Prado Cortez, M. N. (2012). Next place prediction using mobility markov chains. In *Proceedings of the first workshop on measurement, privacy, and mobility*, pages 1–6.

Garola, G., Siragusa, C., Seghezzi, A., and Mangiaracina, R. (2024). Next place prediction model: A literature review. *Emerging Cutting-Edge Developments in Intelligent Traffic and Transportation Systems*, pages 21–30.

Gebrie, H., Farooq, H., and Imran, A. (2019). What machine learning predictor performs best for mobility prediction in cellular networks? In *2019 IEEE International Conference on Communications Workshops (ICC Workshops)*, pages 1–6. IEEE.

Hou, X., Gao, S., Li, Q., Kang, Y., Chen, N., Chen, K., Rao, J., Ellenberg, J. S., and Patz, J. A. (2021). Intracounty modeling of covid-19 infection with human mobility: Assessing spatial heterogeneity with business traffic, age, and race. *Proceedings of the National Academy of Sciences*, 118(24):1–9.

Huang, Z., Xu, S., Wang, M., Wu, H., Xu, Y., and Jin, Y. (2024). Human mobility prediction with causal and spatial-constrained multi-task network. *EPJ Data Science*, 13(1):1–20.

Imai, Y., Tokumoto, T., Koyama, K., Ochi, T., Imai, S., Mori, T., Nakao, T., and Maruyama, K. (2024). Urban human mobility prediction using support vector regression: A classical data-driven approach. In *Proceedings of the 2nd ACM SIGSPATIAL International Workshop on Human Mobility Prediction Challenge*, pages 37–41.

Kapp, A., Hansmeyer, J., and Mihaljević, H. (2023). Generative models for synthetic urban mobility data: A systematic literature review. *ACM Computing Surveys*, 56(4):1–37.

Keyfitz, N. (1973). Individual mobility in a stationary population. *Population studies*, 27(2):335–352.

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

Kong, D. and Wu, F. (2018). Hst-lstm: A hierarchical spatial-temporal long-short term memory network for location prediction. In *Ijcai*, volume 18, pages 2341–2347.

Liu, C., Qin, K., Chen, K., and Ma, R. (2020). Uncovering the aggregation pattern of gps trajectory based on spatiotemporal clustering and 3d visualization. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 42:255–260.

Liu, T. (2010). Application of Markov chains to analyze and predict the time series. *Modern Applied Science*, 4(5):162–166.

Loo, B. P., Zhang, F., Hsiao, J. H., Chan, A. B., and Lan, H. (2021). Applying the hidden Markov model to analyze urban mobility patterns: an interdisciplinary approach. *Chinese Geographical Science*, 31:1–13.

Luca, M., Barlacchi, G., Lepri, B., and Pappalardo, L. (2021). A survey on deep learning for human mobility. *ACM Computing Surveys (CSUR)*, 55(1):1–44.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. *Advances in neural information processing systems*, 30:1–10.

Lv, H., Li, H., Chen, Y., and Feng, T. (2023). An origin-destination level analysis on the competitiveness of bike-sharing to underground using explainable machine learning. *Journal of Transport Geography*, 113:1–18.

Lv, Q., Qiao, Y., Ansari, N., Liu, J., and Yang, J. (2016). Big data driven hidden Markov model based individual mobility prediction at points of interest. *IEEE Transactions on Vehicular Technology*, 66(6):5204–5216.

Ma, Z. and Zhang, P. (2022). Individual mobility prediction review: Data, problem, method and application. *Multimodal transportation*, 1(1):1–11.

Mani, G., Viswanadhapalli, J. K., and Stonier, A. A. (2022). Prediction and forecasting of air quality index in chennai using regression and arima time series models. *Journal of Engineering Research*, 10(2A):179–194.

Maria, E., Budiman, E., and Taruk, M. (2020). Measure distance locating nearest public facilities using haversine and euclidean methods. In *Journal of Physics: Conference Series*, volume 1450, pages 1–7. IOP Publishing.

Menz, L., Herberth, R., Luo, C., Gauterin, F., Gerlicher, A., and Wang, Q. (2018). An improved method for mobility prediction using a Markov model and density estimation. In *2018 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 1–6. IEEE.

Mo, B., Zhao, Z., Koutsopoulos, H. N., and Zhao, J. (2021). Individual mobility prediction: an interpretable activity-based hidden markov approach. *arXiv preprint arXiv:2101.03996*, pages 1–11.

Mohammed, S. N. and Gündüç, S. (2022). Tpm: Transition probability matrix–graph structural feature based embedding. *arXiv preprint arXiv:2208.03712*, pages 234–253.

Mokbel, M., Abbar, S., and Stanojevic, R. (2020). Contact tracing: Beyond the apps. *SIGSPATIAL Special*, 12(2):15–24.

Mokbel, M., Sakr, M., Xiong, L., Züfle, A., Almeida, J., Anderson, T., Aref, W., Andrienko, G., Andrienko, N., Cao, Y., et al. (2024). Mobility data science: Perspectives and challenges. *ACM Transactions on Spatial Algorithms and Systems*, 10(2):1–35.

Nations, U. (2018). Revision of world urbanization prospects. *United Nations: New York, NY, USA*, 799.

Patelli, L., Golini, N., Ignaccolo, R., and Cameletti, M. (2024). S-sirus: an explainability algorithm for spatial regression random forest. *arXiv preprint arXiv:2408.05537*.

Qiao, Y., Si, Z., Zhang, Y., Abdesslem, F. B., Zhang, X., and Yang, J. (2018). A hybrid Markov-based model for human mobility prediction. *Neurocomputing*, 278:99–109.

Qin, Z., Zhang, P., and Ma, Z. (2024). Deepags: Deep learning with activity, geography and sequential information in predicting an individual's next trip destination. *IET Intelligent Transport Systems*, 18(10):1895–1909.

Renaud, O. and Victoria-Feser, M.-P. (2010). A robust coefficient of determination for regression. *Journal of Statistical Planning and Inference*, 140(7):1852–1862.

Rodríguez, A., Kamarthi, H., Agarwal, P., Ho, J., Patel, M., Sapre, S., and Prakash, B. A. (2022). Data-centric epidemic forecasting: A survey. *arXiv preprint arXiv:2207.09370*.

Ruiz-Suarez, S., Leos-Barajas, V., and Morales, J. M. (2022). Hidden markov and semi-markov models when and why are these models useful for classifying states in time series data? *Journal of Agricultural, Biological and Environmental Statistics*, 27(2):339–363.

Sadeghian, P., Han, M., Håkansson, J., and Zhao, M. X. (2024). Testing feasibility of using a hidden Markov model on predicting human mobility based on gps tracking data. *Transportmetrica B: Transport Dynamics*, 12(1):1–20.

Saha, A., Basu, S., and Datta, A. (2023). Random forests for spatially dependent data. *Journal of the American Statistical Association*, 118(541):665–683.

Saputra, R., Sihabudin, A., et al. (2023). Hotspot identification through pick-up and drop-off analysis of ride-hailing transport service. *International Journal of Advanced Computer Science & Applications*, 14(11).

Sarker, I. H. and Salah, K. (2019). Appspred: predicting context-aware smartphone apps using random forest learning. *Internet of Things*, 8:1–11.

Scrucca, L., Fop, M., Murphy, T. B., and Raftery, A. E. (2016). mclust 5: clustering, classification and density estimation using gaussian finite mixture models. *The R journal*, 8(1):289–317.

Shaham, S., Ghinita, G., and Shahabi, C. (2022). Models and mechanisms for spatial data fairness. In *Proceedings of the VLDB Endowment. International Conference on Very Large Data Bases*, volume 16, pages 167–179.

Shaukat, M. H., Hussain, I., Faisal, M., Al-Dousari, A., Ismail, M., Shoukry, A. M., Elashkar, E. E., and Gani, S. (2020). Monthly drought prediction based on ensemble models. *PeerJ*, 8:1–25.

Shin, G.-H. and Yang, H. (2024). Vessel trajectory prediction at inner harbor based on deep learning using ais data. *Journal of Marine Science and Engineering*, 12(10):1–25.

Smola, A. J. and Schölkopf, B. (2004). A tutorial on support vector regression. *Statistics and computing*, 14:199–222.

Suykens, J. A. (2001). Nonlinear modelling and support vector machines. In *IMTC 2001. proceedings of the 18th IEEE instrumentation and measurement technology conference. Rediscovering measurement in the age of informatics (Cat. No. 01CH 37188)*, volume 1, pages 287–294. IEEE.

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

Venables, W. N. and Ripley, B. D. (2013). *Modern applied statistics with S-PLUS*. Springer Science & Business Media.

Wang, D., Zhou, Q., Partani, S., Qiu, A., and Schotten, H. D. (2021). Mobility prediction based on machine learning algorithms. In *Mobile Communication-Technologies and Applications; 25th ITG-Symposium*, pages 1–5. VDE.

Wang, H., Zeng, S., Li, Y., and Jin, D. (2020a). Predictability and prediction of human mobility based on application-collected location data. *IEEE Transactions on Mobile Computing*, 20(7):2457–2472.

Wang, H., Zeng, S., Li, Y., Zhang, P., and Jin, D. (2020b). Human mobility prediction using sparse trajectory data. *IEEE Transactions on Vehicular Technology*, 69(9):10155–10166.

Wang, X., Jiang, X., Chen, L., and Wu, Y. (2018a). Kvlmm: A trajectory prediction method based on a variable-order Markov model with kernel smoothing. *IEEE Access*, 6:25200–25208.

Wang, Y., Qin, K., Chen, Y., and Zhao, P. (2018b). Detecting anomalous trajectories and behavior patterns using hierarchical clustering from taxi gps data. *ISPRS International Journal of Geo-Information*, 7(1–20):25.

Xu, Y., Zou, D., Park, S., Li, Q., Zhou, S., and Li, X. (2022). Understanding the movement predictability of international travelers using a nationwide mobile phone dataset collected in south korea. *Computers, Environment and Urban Systems*, 92:1–13.

Xue, H., Salim, F., Ren, Y., and Oliver, N. (2021). Mobtcast: Leveraging auxiliary trajectory forecasting for human mobility prediction. *Advances in Neural Information Processing Systems*, 34:30380–30391.

Yan, M., Li, S., Chan, C. A., Shen, Y., and Yu, Y. (2021). Mobility prediction using a weighted Markov model based on mobile user classification. *Sensors*, 21(5):1–19.

Yang, Y., Xie, X., Fang, Z., Zhang, F., Wang, Y., and Zhang, D. (2019). Vemo: Enabling transparent vehicular mobility modeling at individual levels with full penetration. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16.

Yu, P.-S., Chen, S.-T., and Chang, I.-F. (2006). Support vector regression for real-time flood stage forecasting. *Journal of hydrology*, 328(3-4):704–716.

Zhang, X., Gui, Z., Liu, Y., Peng, D., Lan, Q., Shen, Z., Chen, H., Zuo, Y., Yao, Y., Wu, H., et al. (2025). Individual mobility prediction by considering current traveling features and historical activity chain. *Geo-spatial Information Science*, pages 1–28.

Zhang, Z., Hörmann, G., Huang, J., and Fohrer, N. (2023). A random forest-based ca-markov model to examine the dynamics of land use/cover change aided with remote sensing and gis. *Remote Sensing*, 15(8):1–17.

Zhao, S., King, I., and Lyu, M. R. (2016). A survey of point-of-interest recommendation in location-based social networks. *arXiv preprint arXiv:1607.00647*, pages 1–30.

Zhao, Z., Karimzadeh, M., Gerber, F., and Braun, T. (2020). Mobile crowd location prediction with hybrid features using ensemble learning. *Future Generation Computer Systems*, 110:556–571.

Zhao, Z., Koutsopoulos, H. N., and Zhao, J. (2018). Individual mobility prediction using transit smart card data. *Transportation research part C: emerging technologies*, 89:19–34.

Zheng, Y., Capra, L., Wolfson, O., and Yang, H. (2014). Urban computing: concepts, methodologies, and applications. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 5(3):1–55.

Zhou, F., Dai, Y., Gao, Q., Wang, P., and Zhong, T. (2021). Self-supervised human mobility learning for next location prediction and trajectory classification. *Knowledge-Based Systems*, 228:1–15.