Capturing Complex Behavior of Mobile Users for Predicting Next Trajectories

Bachelorarbeit
der Philosophisch-naturwissenschaftlichen
Fakultät der Universität Bern

vorgelegt von

Florian Gerber
HS18

Leiter der Arbeit:
Professor Dr. Torsten Braun
Institut für Informatik und angewandte Mathematik

Abstract

The wide spread adoption of cellular phones equipped with global positioning system (*GPS*) sensors makes the exploration of pedestrian mobility patterns possible. Discovering relevant places can be achieved through accumulating *GPS* coordinates. This thesis demonstrates a spatio-temporal analysis on collected geo-location points to discover Zones of Interest (*ZOIs*) of pedestrians to understand underlying movement patterns. This analysis involves the discovery of Points of Interest (*POIs*) based on selected criteria and then an aggregation of these *POIs* to obtain the *ZOIs*. Furthermore, a new Markov based model to predict long distance trajectories of pedestrians is introduced. The model is capable of choosing between the first and second order Markov chain in order to accommodate to the different movement behaviours of individual pedesterians and the available trace data. To quantify the movement behaviour of users an existing Periodicity Detection algorithm is modified to achieve a better task-specific, computational performance. In addition, the adaptive Markov model is evaluated and compared to other current trajectory prediction methods using the real-life Mobile Data Challenge (*MDC*) dataset. The proposed model achieves a precision of up to 86% and a recall of up to 84% for predicting future trajectories of users in the *MDC* dataset. The thesis further presents a mechanism to predict the number of pedestrians in urban areas traveling on the same trajectory at a future point in time. This mechanism combines the adaptive Markov model with an existing next Place Prediction algorithm and a means of storing and aggregating predicted trajectories for multiple users.

Mobile Crowd Location Prediction with Hybrid Features using Ensemble Learning

Zhongliang Zhao*, Mostafa Karimzadeh, Florian Gerber, Torsten Braun

Institute of Computer Science, University of Bern
Neubrückstrasse 10, 3012 Bern, Switzerland

Abstract

With the explosive growth of location-based service on mobile devices, predicting users' future locations and trajectories is of increasing importance to support proactive information services. In this paper, we model this problem as a supervised learning task and propose to use ensemble learning methods with hybrid features to solve it. We characterize the properties of users' visited locations and movement patterns and then extract feature types (temporal, spatial, and system) to quantify the correlation between locations and features. Finally, we apply ensemble methods to predict users' future locations with extracted features. Moreover, we design an adaptive Markov Chain model to predict users' trajectories between two locations. To evaluate the system performance, we use a real-life dataset from the Nokia Mobile Data Challenge. Experiment results unveil interesting findings: (1) For individual predictors, Bayes Networks outperform all others when data quality is good, while J48 delivers the best results when data quality is bad; (2) Ensemble predictors outperform individual predictors in general under all conditions; and (3) Ensemble predictor performance depends on the user movement patterns.

Keywords: Hybrid feature, Supervised learning, Ensemble learning, Location and trajectory prediction.

*Corresponding author
Email address: zhao@inf.unibe.ch (Zhongliang Zhao)

1. Introduction

Smart-phones are becoming part of people's daily life. Increasing pervasive usage of location-based services and smart-phones around the world contributed to vast and rapid growth of mobility data volume. The large size of heterogeneous mobility data gives rise to new opportunities for discovering characteristics and movement patterns of human mobility behaviors. Mobile data normally consists of historical information of users' visiting sequence, which includes the detailed context of the visited locations and corresponding time-stamps.

Future location prediction is a specific topic in mobile data analysis. The knowledge of mobile user positions fosters applications that need to know this information to operate efficiently. Examples of such services are traffic control, location-based advertising, mobile network management, etc. Many location-based services depend on the current or future locations of users. In addition to location prediction, predicting trajectories between two locations is also of great importance, which helps to optimize travel paths between two locations.

The type of dataset plays an important role in accurate location prediction as the prediction models learn user movement patterns from collected data. The Nokia Mobile Data Challenge (MDC) dataset [1] holds great potential for providing fine-quality information for predicting users' next places. It includes the mobility profiles of nearly 180 users for almost 2 years. From the study of the MDC dataset and the ground truth, we could find out that the visits of certain places follow some regular patterns. Moreover, people behaviors at specific locations also provide useful information for certain predictions.

In this work, we formulate the location prediction problem as a standard supervised machine learning task, where an user-place pair is represented by a set of features and the future places are considered as targets. Our goal is to extract and properly select as many useful features as possible, and build accurate classifiers (both individual and ensemble ones) with those features. We prefer to extract features that have discriminative information among different locations, such that locations can be identified from the observed features. Machine learn-

ing techniques have been widely used to discover behaviors and patterns based on large-scale empirical data. Machine learning algorithms can take advantages of training data to capture characteristics of the unknown probability distribution among different locations. They could automatically learn to recognize complex patterns and make intelligent decisions based on the learned knowledge. In this work, we use WEKA [2], which is a comprehensive open source tool for machine learning and data mining. WEKA provides implementations of multiple machine learning algorithms, and we propose to apply ensemble methods to combine multiple individual predictors to achieve the best performance.

Machine learning can only make accurate classification, if high discriminative features are constructed and useful patterns can be observed from the defined features. However, traditional location prediction methods often separately consider spatial or temporal context information [3] [4]. Although there have been some efforts to integrate spatial and temporal features for location prediction, most of them suffer from over-fitting problems due to the large number of spatial-temporal trajectory patterns. Some existing works model next place prediction as a classification problem [5] [6]. However, issues such as the consideration of other rich contextual data, such as accelerometer, Bluetooth/WiFi connectivity, call/SMS logs, information about running applications have not been investigated systematically. In order to accurately predict the future place of a user, it is fundamental to identify and extract a number of descriptive features for each place visited by the user.

Therefore, this work focuses on extracting discriminative features among different locations, such as temporal, spatial, and smart-phone system features. With these features, we apply ensemble learning techniques to improve the location prediction accuracy. The main contributions of this work are as follows.

- First, we systematically characterize the properties of users' visited places and movement patterns from a real-life dataset and then extract various types of features (temporal, spatial, and smartphone system features) to quantify the correlations between places and features.

- Second, with the extracted features, we propose to apply ensemble learning techniques to improve the crowd location prediction performance by integrating multiple individual predictors. We conducted detailed experiments for users with different movement types and trace qualities to show the superiority of ensemble predictors over individual predictors. Moreover, we also measure the algorithm execution time to show that the superior-performance of ensemble predictors comes at a price of higher computation overheads. This detailed analysis enables us to understand which algorithms could achieve the best performance under what conditions.

- Third, we analyze the performance of different individual and ensemble learning predictors from a mathematical perspective and conduct the time complexity analysis of each algorithm to theoretically understand why there are significant performance differences.

- Fourth, we propose an adaptive Markov Chain-based trajectory prediction approach, which adaptively selects the first-order or the second-order Markov Chain model to predict the future trajectory of mobile users based on dataset conditions.

- Fifth, from the experimental and theoretical analysis, we analyze how the prediction performance is affected by various factors such as mobility trace qualities, extracted features, user movement patterns, predictor models, etc. This knowledge enables us to further design an adaptive prediction system, which dynamically selects predictors based on dataset and smartphone conditions, to guarantee the required system performance.

The structure of this paper is as follows. Section 2 discusses existing efforts on location prediction from mobile data. Section 3 describes the dataset that has been used in this work. Section 4 details how we define the features and which features are used in our prediction system. Section 5 explains the individual and ensemble predictors that are used in this study. Section 6 discusses the performance evaluation, and the paper concludes in Section 7.

2. Related Work

With a large number of built-in sensors, smartphones are able to record rich types of quality data without the need of any additional devices. Compared to the check-in data collected from the location-based social networks such as Foursquare [7], which only records the discrete checked-in data at different locations, smartphones have the unique advantage to record data in a continuous way. Therefore, human mobility analysis has become an active research topic thanks to the fast development of continuous location tracking techniques. Song et al. [8] presented a study on predictability of human mobility by analyzing the entropy of location traces. Several prediction methods have been proposed for human mobility in different contexts. Ashbrook et al. [9] introduced to extract significant places and represent location traces as strings and then use Markov models to predict the next place that a user will visit. NextPlace [10] proposed a location prediction solution based on nonlinear time series analysis of the arrival and staying duration of users in relevant places. However, the work is only focusing on GPS coordinates-based prediction. Zhao et al. [11] designed a Dynamic Bayesian Network-based model to predict the future cells of mobile users to optimize telecommunication network operations. He et al. [12] described a time-based Markov predictor for the location prediction of stationary and mobile users. However, their works are limited to specific methods, which can only produce a prediction accuracy of nearly 60%. Moreover, the transition matrix-based approaches have clear drawbacks, since they take only the visit logs as model inputs, but completely ignore the rich context information.

In the next place prediction task of Nokia Mobile Data Challenge 2012, the best methods relied only on spatial-temporal information to predict future locations [13], [14], [15], [16]. For instance, Lu et al. [16] focused on using the transitions between places for each individual user, as well as the time context, to make predictions. They also tried to explore other context information such as call-logs and accelerometer data in the current place. However, they only applied a support vector machine (SVM) for each user to predict their future lo-

cations. Tran et al. [17] applied an user-specific decision tree, which was learned from each user's movement history, to predict their future locations. However, their works were limited to the decision tree-based predictor. [18] proposed to learn the time distribution for each place as well as the transition patterns between places by using the kernel density estimation to capture spatial-temporal context features. Zhu et al. [19] introduced a feature engineering mechanism to predict semantic meaning of places. However, their works were also limited to very few individual classifiers. As we can see, most of the existing works focused on applying only individual machine learning algorithms to improve prediction accuracy. However, ensemble learning has been proven to obtain better performance than could be obtained from any of the constituent algorithms alone [20] [21]. Therefore, we focus on applying different ensemble learning methods to optimize location prediction accuracy. Trajectory prediction estimates the path between two locations. In [22], authors proposed a solution considering users' movement patterns among different zones of interest. However, it's a pure statistical approach, which does not include any future location predictions.

3. MDC Dataset

Our experiment data is from the Nokia Mobile Data Challenge (MDC) [1], a dataset that was collected using Nokia N95 smartphones on a 24/7 basis in Switzerland from October 2009 to March 2011. About 180 volunteers participated in the campaign, where they were asked to carry the smartphones during their daily life with recording software running in the background. Even though volunteers agreed to carry the smartphones during the campaign, their different behaviors lead to different trace qualities. Moreover, users also had different movement patterns, and some users traveled regularly while others did not. Based on these observations, we divided the users into multiple categories, depending on the number of available data points, so called instances, which have been recorded and the movement patterns of the mobile users.

3.1. User Classification

3.1.1. User Trace Quality

Different behaviors of users lead to different trace qualities. Some users carry the smartphones all the time. Therefore, the recorded data is complete and useful for making prediction. However, some others forgot to carry the devices or to charge them in time, such that data recordings are non-continuous and useless for prediction. In the MDC dataset, whenever a user stayed in a place for more than 10 minutes, an entry will be created in the table. The instance includes: `User_ID`, `Place_ID`, `Starting_Time`, `Ending_Time`. `Samp_Dist_Corr`, which means a user with `User_ID` has arrived at a place (with `Place_ID`) from `Starting_Time` and left the place at `Ending_Time`. Therefore, we define 5 categories of quality, depending on the number of instances recorded in a user's movement traces.

- Very good: $\geq$ 1500instances
- Good: 1200-1500 instances
- OK: 1000-1200 instances
- Bad: 800-1000 instances
- Very bad: $\leq$ 800 instances

3.1.2. User Movement Patterns

In addition to the trace quality, user movement patterns also have significant impact on location prediction. Users had different mobility patterns. Some users moved regularly, they traveled between home and office during working days with a homogeneous movement pattern, and, thus, it is easy to find out patterns. However, some other users traveled randomly and visited many different places for very few times during the data collection period. Their movements are heterogeneous and it is hard to predict their future locations even though the recorded number of data entries is high. Based on this, we defined two types of user movements: homogeneous and heterogeneous. Homogeneous movement means that the user's mobility pattern is quite regular and repeatable, and the user visits some places quite frequently. In contrast, heterogeneous movement means that the movement traces are rather random and non-repeatable. In the experiments we retrieved the visited places of each user, and classify users'

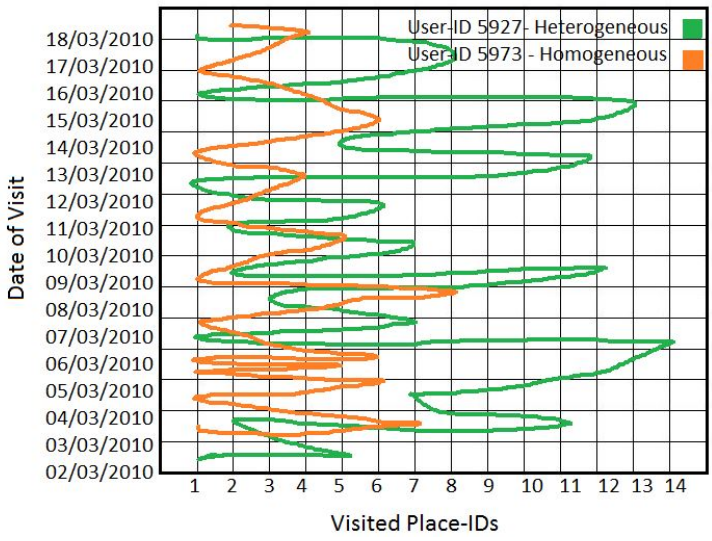


Figure 1: Homogeneous and heterogeneous movements.

movements types based on the number of places a user has visited and the number of the visit. Figure 1 shows an example of homogeneous and heterogeneous movement types, where the user visits very few places frequently homogeneous movement pattern and visited many different places occasionally for heterogeneous movement types.

3.2. Place Category

The raw location data from the MDC dataset were recorded as sequences of GPS coordinates. In our work, we defined places as circular areas that around GPS coordinate points. As most works on MDC-based location prediction, we defined ten categories of places, which are shown in Table 1.

Table 1: Visited Place Categories

Label	**Place**	**Label**	**Place**
1	Home	6	Outdoor sports
2	Friend home	7	Indoor sports
3	Office	8	Restaurant
4	Transportation	9	Shop
5	Friend office	10	Holiday

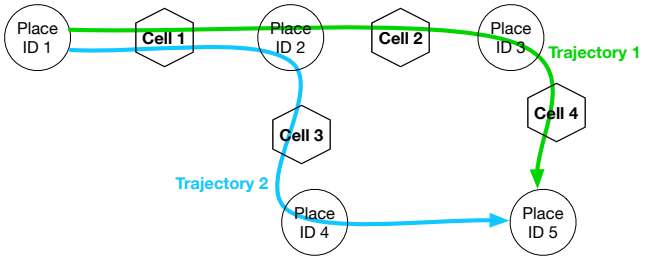


Figure 2: Mobile user trajectories.

3.3. User Trajectory

A mobile user can take different paths to move from one place to another. In Fig.2, the user has two possible trajectories to go from place id 1 to 5, via different other places while being connected to different cells. Thanks to the availability of connected cell ID in the MDC dataset, we are able to extract correlations between users' trajectories with their movement behaviors. We formally define a trajectory t between two places as a sequence t = $\{cell_1, cell_2, ..., cell_n\}$, which contains all the GSM cells that the user connected to while moving from one place to another one. Furthermore, we define $T_{i,j} = \{t_1, t_2, ..., t_n\}$ as the set of all trajectories between places i and j.

4. Features

As stated before, a proper feature construction is fundamental to apply supervised machine learning algorithms to make accurate prediction. Therefore, we need to construct features from a tremendous amount of raw data and assign a set of features (feature vector) to each user-place pair. Feature selection is a process of selecting a subset of relevant features (attributes) for their use in prediction model construction. It is the process of choosing a subset of original features such that the feature space is optimally adapted and the appropriate features are selected for classification. The collected MDC raw data is of huge size. Therefore, it is important to select a subset of data by creating feature sets, and identify redundant and irrelevant information. Table 3 shows the association between all the features and places that are used in this work.

4.1. Feature Construction

Most of the MDC-based prediction works use only temporal or spatial features. We combine both and additionally consider the smartphone system-related features, which include context like battery level, charging frequency, detected WiFi network, etc. Below we describe the three categories of features that are used in our system.

4.1.1. Temporal Features

Temporal features include context information relevant to the staying time of a visit. Our visits to certain places tend to have some temporal characteristics that are relevant to the places. For instance, we stay at offices normally between 8:00 to 12:00 and 14:00 to 18:00, and we are at restaurants for lunch between 12:00 to 14:00. Below we detail the extracted temporal features and the feature-place association. We used a time granularity of 1 hour to divide a day of 24 hours. An example of a day time decomposition is shown in Figure 3.

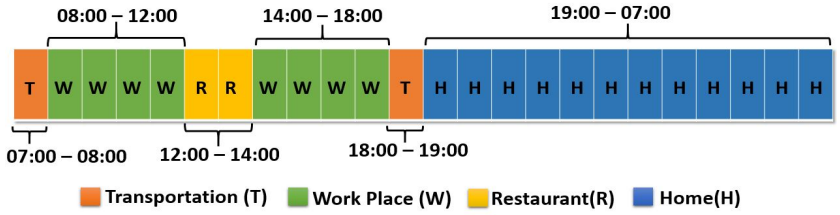


Figure 3: Day time decomposition.

- **Weekday**: to indicate which weekday is the visit.
- **Leaving time**: the ending time of the visit. We defined 6 time intervals, and each time period could be mapped to a specific place. For instance, if the visit is between 07:00 and 08:00, then the place is a transportation hub of a certain probability.
- **Duration**: time duration of the visit at a place.

4.1.2. Spatial Features

Spatial features include context relevant to the geographical information of the visits. We have selected the following feature.

- **Visiting frequency**: how often to re-visit a place.

4.1.3. System Features

Smartphone system features also have discriminative characteristics in different places, and include context information relevant to the smartphone's system information. We suppose that this information is also helpful when predicting users' future locations. For instance, places like restaurants or homes tend to have more WiFi networks visible than other places, and people tend to have different types of applications running on their phones when they are working in the office or enjoying holidays in a resort.

- **WiFi connection**: the number of visible WiFi networks.
- **Acceleration variation**: movement speed variation, which can be derived from the smartphones' motion sensors. It can be used to detect changes of movement types, for instance a change from slow speed to fast speed probably means the user is at the transportation places.
- **Running application**: the type of running application. This feature is mainly used to detect that whether the users are in indoor or outdoor environments. For instance, map applications are mostly used outdoors, while a connected WiFi network indicates the user is more probably in an indoor environment. These information could further help us to improve the location prediction accuracy.
- **Smartphone profile statement**: profile of the phone, for instance normal or silent mode. Silent mode is more used during office time or concerts, which helps us to predict those places.
- **Charging frequency**: how often the smartphones are charged during the whole period of data collection. People tend to charge their phones in offices and home, which helps us to detect home and office areas.

Table 2: Feature coefficients

Feature	**Coefficient**
Detected WLAN (1-4)	97.2
Charging frequency (90-100)	85.35
Acceleration variation	32.06
Staying duration (48-120)	30.19
Leaving time (12:30-14:00)	20.5
Frequency of visit (20-60)	29.89
Weekday (Thursday)	21.44
Is_weekend	7.41

4.2. Feature Importance

Given the extracted features, the next step is to select those features that influence the prediction output more than others. WEKA has many algorithms to do this automatically, and we choose the *Logistic Regression* algorithm [23]. The *Logistic Regression* algorithm is very efficient for the MDC data set, since it has both nominal and numerical features. Table 2 represents the feature coefficients, which are generated automatically by *Logistic Regression* from WEKA. It shows that *Detected WLAN* has the best contribution for the prediction result. The *Charging frequency*, *Acceleration variation* and *Duration* of staying at a place are ranked on second level, third level features include the *Visiting frequency* and *Leaving Time* and the *Week day* is the feature with lowest impact on prediction output.

5. Predictors

In this section, we describe the predictors we used to evaluate our prediction system. We focus on the individual predictors as well as on ensemble predictors.

5.1. Individual Predictor

Three categories of individual predictors are mostly used in machine learning: Decision Tree predictors, Bayes predictors, and Neural Networks predictors/Multilayer perceptron.

Table 3: Place-Feature Correlation

Feature / **Place**	Leaving Time	Duration (Minutes)	Weekday	Visit Freq.	# Visible WiFi	Acce. Var. (M/s^2)	Running APP	Phone Profile	Charge Freq.
Home	20:00∼07:00	[480, 2880)	MON to SUN	[300, 450]	[1, 4)	[10, 100)	Indoor	Normal	[250, 300)
Work	08:00∼12:30 13:30∼18:30	[120, 480)	MON to FRI	[200, 300)	[4, 6)	[10, 100)	Indoor	Silent	[90, 250)
Restau.	07:00∼09:00	[40, 120)	MON to SAT	[60, 250)	[6, 12)	–	–	Normal	–
Transp.	07:00∼08:30 18:00∼19:30	[0, 40)	MON to SUN	[20, 100)	[4, 6)	[100,)	–	Normal	–
Outdoor Sports	12:00∼14:00	[0, 60)	SAT to SUN	[15, 70)	–	[50, 100)	Outdoor	Normal	–
Indoor Sports	18:00∼20:00	[0, 60)	SAT to SUN	[15,80)	[1, 3)	[50, 100)	–	Normal	–
Shopping Center	–	[40, 120)	FRI to SAT	[30, 130)	[6, 12)	[10, 100)	Outdoor	Normal	–
Holiday Resorts	–	–	–	[5, 30)	–	–	Outdoor	Normal	–
Friend Home	19:00∼22:00	[60, 180)	FRI to SUN	[5, 10)	[1, 4)	[10, 100)	–	Normal	[20, 90)
Friend Office	–	–	–	–	[4, 6)	[10, 100)	–	Normal	–

5.1.1. Decision Tree

A decision tree is a hierarchical structure for classifying objects, composed of nodes that correspond to primitive classification decisions. At the top of the tree is the root node that specifies the first dividing criterion. The root, and every non-leaf node, has multiple child nodes, which can be classified further by checking other criteria. The root node contains all the visits of the training data, while child nodes contain those visits that match the dividing criteria along the path from root to that node. In our experiments, we used the *J48* and the *Random Forest* algorithms. *J48* is one of the mostly used statistical classifier, and *Random Forest* is a combination of tree predictors such that each tree depends on the values of a random vector sampled independently. Figure 4 shows a J48 tree, in which the first dividing feature is the number of detected WLAN networks, and the features along the path towards the leaf are: duration of a visit in a place, acceleration variation, charging frequency, leaving time from a place, visit frequency of a place, whether the visit is on a weekday or not. The feature ranking is consistent with the feature coefficient shown in Table 2.

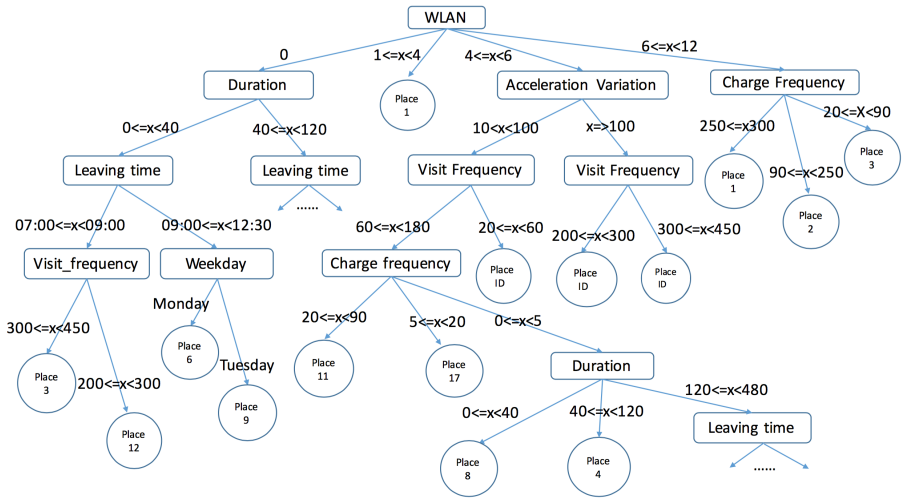


Figure 4: A J48 decision tree.

5.1.2. Bayesian Networks

Bayesian Networks are a class of statistical models to define conditional dependencies between attributes and parent node, represented by a graph. To do so, the *Bayesian Network* uses a *Directed Acyclic Graph* (DAG), to create connections between a set of attributes A=$\{attribute_1, attribute_2, ..., attribute_n\}$ and the parent node. In our case the parent node is visited Place-IDs, because we believed that the current place has a strong connection with the user's next place. Figure 5 shows an example of the *Directed Acyclic Graph* with the extracted features and parent node.

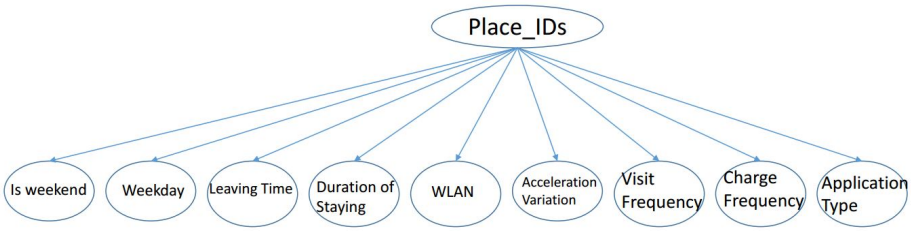


Figure 5: A Directed Acyclic Graph (DAG) of Bayesian networks.

5.1.3. Neural Networks

Artificial Neural Networks are a mathematical model to solve a variety of problems in pattern recognition and classification. ANNs can be viewed as weighted directed graphs in which defined attributes are input layer, classes (Place-IDs) are output layer and directed edges with weights are connections between input and output. In this work, we used the WEKA implementation of ANNs called Multilayer Perceptron (MLP). Figure 6 shows the MLP with extracted features in our case. In this model, connections are organized into layers that have unidirectional connections between them. Weights are determined to allow the network to produce answers as close as possible to the known correct answers. The network usually must learn the connection weight from available training patterns. Performance is improved over time by iteratively updating the weights in the network.

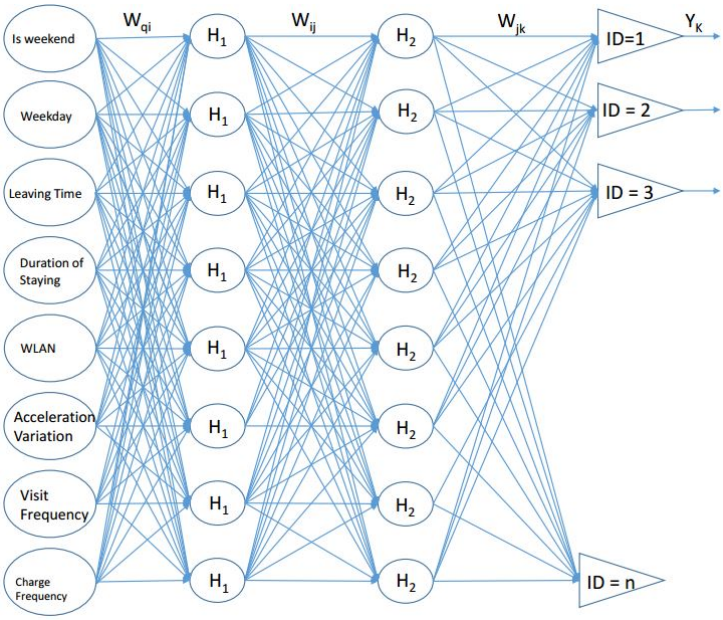


Figure 6: A typical two-layer Multilayer Perceptron Architecture.

5.2. Ensemble Predictors

Ensemble learning is an approach to combine individual predictors to achieve better performance. As different users have different mobility patterns, there is no single predictor that could outperform the others for all users. Therefore, we focus on finding suitable models for different mobility pattern and combine the models to deliver the optimized performance. The task of constructing an ensemble classifier can be broken into two sub-tasks: (1) selecting diverse set of base classifiers with acceptable performance; and (2) appropriate combinations of their predictions with appropriate weights. In this work, three types of ensemble predictors are applied: Boosting, Bagging, and Stacking.

5.2.1. Boosting

Boosting is an ensemble method that begins with a base classifier, which is selected from a first experiment results performed on the training data. A second classifier is then created behind it to focus on the instances in the training data that the first classifier got wrong. The process continues to add classifiers, until an accurate threshold is reached. The *AdaBoost* algorithm was the first practical boosting algorithm that is widely used and studied in numerous applications and research fields [24]. We use it to integrate *J48*, *Random Forest*, *Bayes Networks*, *Naive Bayes* and *MLP*.

5.2.2. Bagging

Bagging is an ensemble method that divides the training data set into several subsets with the same sizes. Then, it creates a classifier for each subset. Afterwards, the final decisions are calculated by getting average values from the results obtained using the individual data sets. In this work, we used *Bagging* to integrate *J48*, *Random Forest*, *Bayes Networks*, *Naive Bayes* and *MLP*.

5.2.3. Stacking

Stacking focuses on a function to combine the outputs of the base learners using a meta-learner, which called *Simple Logistic*. In this work, we integrated*J48*, *Bayes Networks*, and *MLP* with *Stacking*.

6. Performance Evaluation

This section presents the experimentation parameters and detailed performance evaluation of the discussed prediction methods. The evaluation metrics we used are prediction accuracy and prediction execution time, which indicate how accurate the algorithm is and how long it takes to generate the prediction results. From these evaluation results, we further analyze the potential influencing factors on the prediction accuracy performance. We highlight the impacts of temporal and hybrid features, as well as trace quality. Finally, the paper also includes the theoretical analysis about the performance of different algorithms under different conditions.

All experiments were run on a laptop running Windows 8.1 Enterprise with Intel vPro (64-bit-X68 architecture) core i7 CPU 3.2 GHz and 16 GB memory.

6.1. Machine Learning Approaches and Parameters

6.1.1. Location Prediction

In this work we use WEKA [2] to discover the behaviors and mobility patterns of the mobile users by learning from their historical trajectories. WEKA includes several types of machine learning algorithms, such as *Tree-based*, *Bayesian Networks-based* and *Neural Network-based*. Moreover, it also provides ensemble learning methods, such as *Bagging*, *Boosting* and *Stacking*. We study the performance of *J48*, *Random Forest*, *Bayes Networks*, *Naive Bayes* and *Multilayer Perceptron (MLP)* algorithms. In order to improve the accuracy of individual algorithms, we apply *Boosting* and *Bagging* to individual algorithms and apply *Stacking* to integrate multiple individual predictors. We carry out all experiments using temporal+spatial features and hybrid (temporal+spatial+system) features. The experiments are performed using traced data sets of fifteen users, which are randomly selected from different quality categories, and results are averaged over those users. For each user, we divide available trace data into ten subsets using *10-fold cross-validation*, in which one of the 10 subsets is used as the testing set and the other 9 subsets are put together to form a training set. Table 4 shows some of the experiment parameters.

Table 4: Experiments parameters.

Parameter	**Definition**	**Value**
Confidence factor	Reduce the size of the decision tree by removing insignificant nodes	0.25
Number of objects	Minimum number of instances per leaf in the decision tree	2
Hidden layers	Hidden layers of the neural network	45-55
Validation	Number of iterations to run after observing lower prediction accuracy in *Boosting*	2
Maximum depth	Maximum depth of a tree in *J48* and *Random Forest*	1000 *level*
Training time	Duration of training for individual algorithms per iteration in *Boosting*	300 *sec*
h	Number of neurons at each hidden layer *Stacking*	3
o	Number of outputs in MLP *Stacking*	100-160
i	Number of iterations in MLP *Stacking*	5
T	Number of trees to generate in *Random Forest*	20
L	Number of possible iterations for individual algorithms in *Boosting*	5
N	Number of new generated training sets in *Bagging*	10
J	Number of new generated training sets in *Stacking*	10

6.1.2. Trajectory Prediction

For trajectory prediction, we have developed a novel adaptive Markov Chain-based model. As defined in Section 3.3, $T_{i,j}$ is a set of trajectories $T_{i,j} = \{t_1, t_2, t_3, ..., t_n\}$, where each trajectory t_n is a set of m connected cells such that $t_n \in T_{i,j} : \{cell_1, cell_2, cell_3, ..., cell_m\}$. For each subset t_n the first cell is located in *Place-ID* $= i$ and the last cell is located in *Place-ID* $= j$. In addition, connected cells on trajectories between two places i and j do not appear on other trajectories starting from place i towards other places. As shown in Fig. 7, the model compares the detected periodicity (P) with a predefined threshold value (P_{th}) to decide either the first order or the second order Markov Chain model should be applied. The First Order Markov Chain is applied if the user's mobility pattern between two places is regular (homogeneous movements). For place pairs where the user's mobility pattern is irregular (heterogeneous movements), the Second Order Markov Chain is used. We use the periodicity detection approach proposed in [25] to identify the user movement types and detect the

changes of user movement patterns such that the corresponding Markov Chain model is applied. With this model, the probability of the next cell in a trajectory is given by:

$$Pr(cell_{i+1}) = \begin{cases} Pr(cell_{i+1} \mid cell_i) & \text{if } P \leq P_{th} \\ Pr(cell_{i+1} \mid cell_i, cell_{i-1}) & \text{if } P > P_{th} \end{cases}$$

In the experiments, for a given $T_{i,j} = \{t_1, t_2, ..., t_n\}$ we use the threshold value $P_{th} = \sum_{i=1}^{n} \frac{|t_i|}{n}$, which denotes the mean length of a trajectory in $T_{i,j}$.

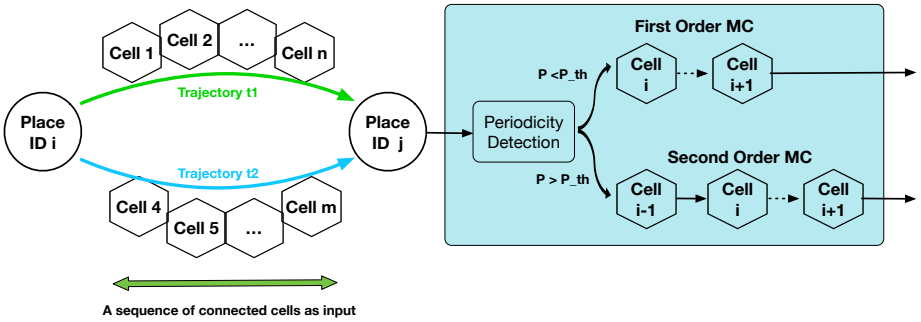


Figure 7: Adaptive Markov Chain-based Trajectory Prediction.

6.2. Evaluation Results

In this subsection, we present the evaluation results of different predictors. We focus on two metrics: prediction accuracy and prediction time. Location prediction accuracy refers to the percentages of correct location prediction, and prediction time refers to the execution time of performing the prediction task. For trajectory prediction accuracy, we use the metrics of precision and recall, as defined in the work of [22].

6.2.1. Location Prediction Accuracy of Individual Algorithms

This subsection details the prediction accuracy results of individual algorithms. We first present the average prediction accuracy of all the users with different trace qualities. Then, we discuss more details about the prediction accuracy of users with homogeneous and heterogeneous movement patterns.

Fig. 8 and Fig. 9 show the average prediction accuracy of all the users for different individual algorithms using temporal, spatial, and hybrid features. The results clearly show that the *Decision Trees* family (specially *J48*) outperform others, when using the trace data with lower quality, and *Bayes Networks* provides better performance ($> 84\%$ accuracy) when the data is with higher quality. Moreover, it can be observed that the estimated accuracy is improved significantly if the hybrid features are used instead of using only temporal+spatial features. For instance, *Bayes Networks* delivers an accuracy of 84.76% with hybrid features, while only 55.47% can be reached with temporal+spatial features.

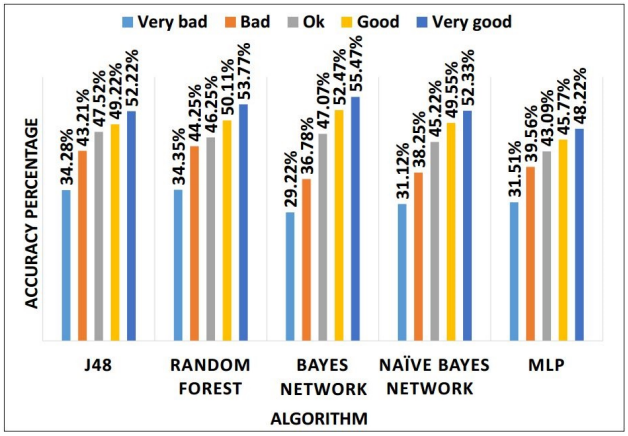


Figure 8: Prediction accuracy of individual algorithms using Temporal+Spatial features.

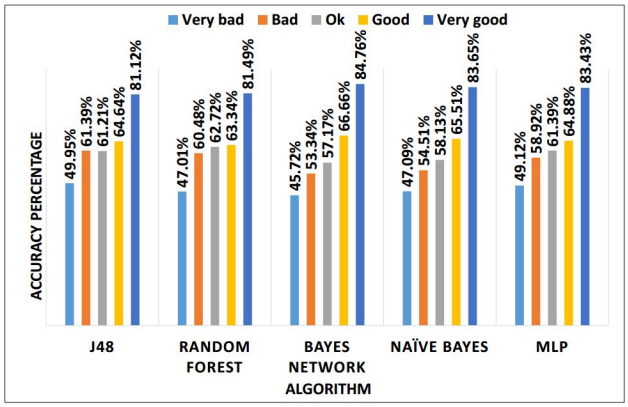
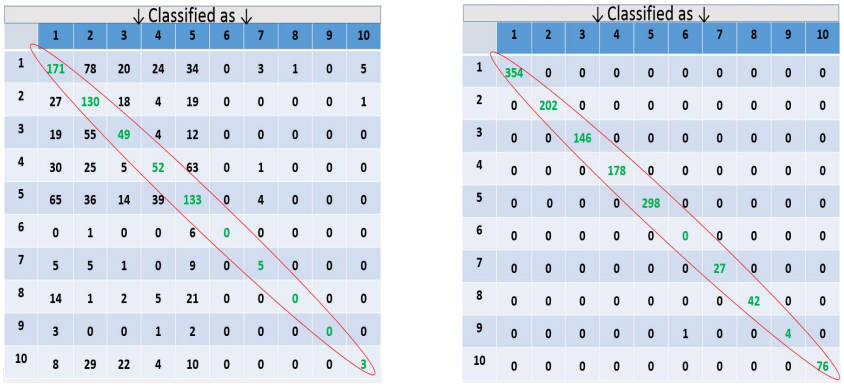


Figure 9: Prediction accuracy of individual algorithms using Hybrid features.

Fig. 10 shows two confusion matrices that help to explain the nature of the errors made by the classifier with different features[26]. A confusion matrix is a table that is often used to describe the performance of a classifier on a set of test data for which the true values are known. For instance, row 1 of the table shows that 78 places with real class type = 1 were wrongly predicted as class 2, and 171 places with real class type = 1 were correctly predicted. These matrices are generated by the *J48* algorithm over the 10 most visited places (indicated by IDs). For instance, Fig. 10a shows that when the predictor uses only the temporal+spatial features, prediction accuracy is lower and several incorrect predictions are observed. Fig. 10b shows that the number of correct predictions are significantly improved when the hybrid features are used.



↓ Classified as ↓

	1	2	3	4	5	6	7	8	9	10
1	171	78	20	24	34	0	3	1	0	5
2	27	130	18	4	19	0	0	0	0	1
3	19	55	49	4	12	0	0	0	0	0
4	30	25	5	52	63	0	1	0	0	0
5	65	36	14	39	133	0	4	0	0	0
6	0	1	0	0	6	0	0	0	0	0
7	5	5	1	0	9	0	5	0	0	0
8	14	1	2	5	21	0	0	0	0	0
9	3	0	0	1	2	0	0	0	0	0
10	8	29	22	4	10	0	0	0	0	3

(a) *Temporal.*

↓ Classified as ↓

	1	2	3	4	5	6	7	8	9	10
1	354	0	0	0	0	0	0	0	0	0
2	0	202	0	0	0	0	0	0	0	0
3	0	0	146	0	0	0	0	0	0	0
4	0	0	0	178	0	0	0	0	0	0
5	0	0	0	0	298	0	0	0	0	0
6	0	0	0	0	0	0	0	0	0	0
7	0	0	0	0	0	0	27	0	0	0
8	0	0	0	0	0	0	0	42	0	0
9	0	0	0	0	0	1	0	0	4	0
10	0	0	0	0	0	0	0	0	0	76

(b) *Hybrid.*

Figure 10: Confusion matrices using different features.

Next, we present the prediction accuracies of individual predictors for users with homogeneous and heterogeneous movement patterns. As shown in Fig. 11, the *Bayes Networks* scheme delivers the best performance for both movement patterns, which is consistent with its superior performance presented in Fig. 9.

6.2.2. Location Prediction Accuracy of Ensemble Methods

In this subsection, we present the prediction accuracy of different ensemble learning algorithms. Same as for the individual algorithms, we first present the average prediction accuracy of ensemble learning algorithms for all users. Then,

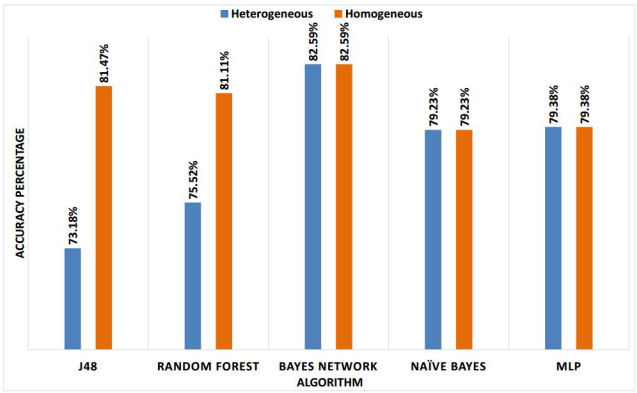


Figure 11: Prediction accuracy of individual predictors with hybrid features for homogeneous and heterogeneous movements

we discuss more details about the prediction accuracy of users with homogeneous and heterogeneous movement patterns.

Fig. 12 and Fig. 13 present the prediction results of *Boosting* and *Bagging* using hybrid features. The graphs show that using *Boosting*, prediction accuracy is improved by around 10% compared to when individual algorithms are applied. It can also be observed that *Boosting* outperforms *Bagging*. Different algorithms provide different prediction performance values. For instance, *J48* using *Boosting* performs better when the traced data is of low quality. However, using traced data with higher quality, the integration of the *Bayes Networks* and *Boosting* outperforms the others.

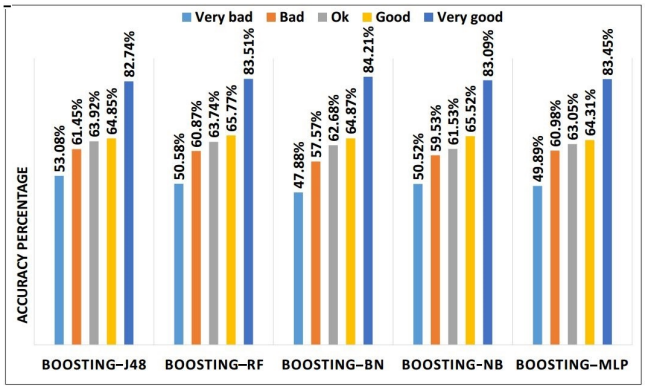


Figure 12: Prediction accuracy of *Boosting*.

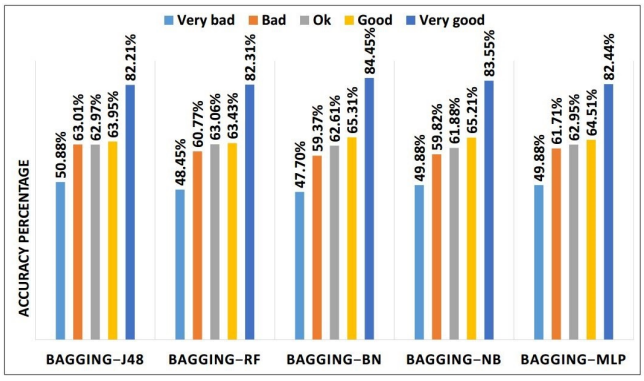


Figure 13: Prediction accuracy of *Bagging*.

Fig. 14 shows the evaluation results of the *Stacking* learning method built by *Simple logistics* as a meta-learner for the hybrid features. Due to generating higher accuracy results by *J48*, *Bayes Networks*, and *MLP*, we decided to integrate them using *Stacking*. *Random Forest* and *Naive Bayes* are ignored as they do not improve prediction accuracy. The graph shows that by integrating *J48* and *MLP*, prediction performance is improved by 10% to 14% compared to the individual algorithms even for trace data with low quality. Another significant improvement can be observed when *J48* is integrated with *Bayes Networks* and *MLP* mechanisms, particularly when the trace data is of high quality.

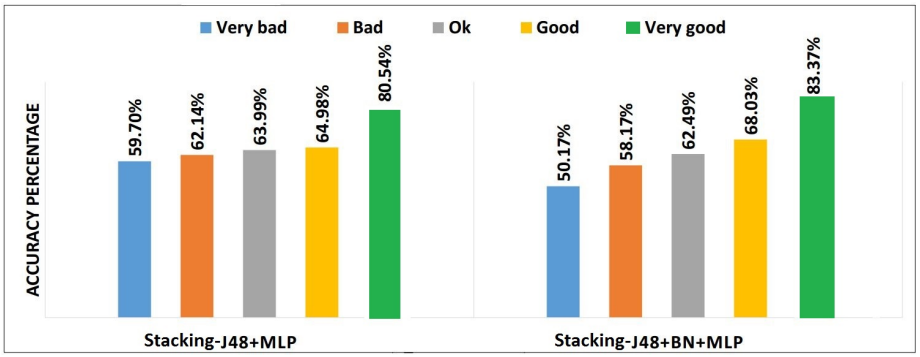


Figure 14: Prediction accuracy of *Stacking*

Next, we discuss the prediction accuracies of ensemble predictors for users with different movement patterns. We take user 5927 as an example, and as shown in Fig. 15, *Boosting* delivers better results than *Bagging* for both movement patterns, which is also consistent with the results presented in Fig. 12 - 13. Therefore, from Fig. 11 and Fig. 15 we see that *Boosting* significantly outperforms individual predictors for homogeneous movements, while for heterogeneous movements, their performance are similar to each others. Therefore, an adaptive model selection mechanism should be developed based on the detected movement patterns such that the appropriate predictors can be applied to guarantee optimal prediction performance.

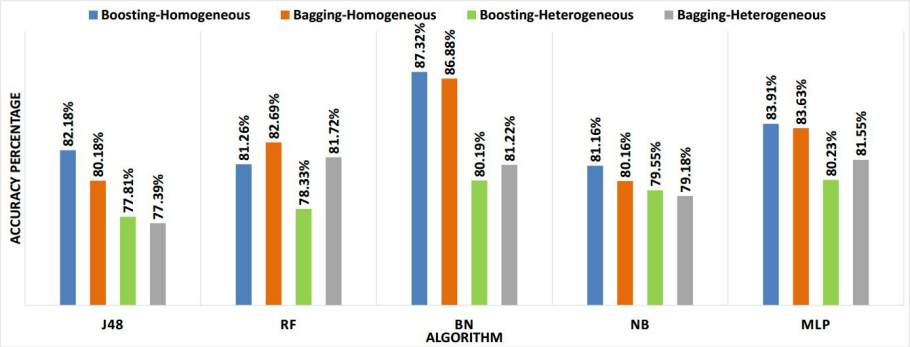


Figure 15: Prediction accuracy of *Boosting* and *Bagging* with hybrid features for homogeneous and heterogeneous movements

6.2.3. Location Prediction Execution Time of Individual Algorithms

In addition to prediction accuracy, we also measure the prediction execution time of each individual algorithm using temporal+spatial features and hybrid features. The obtained results, as shown in Fig. 16 and Fig.17, indicate that the *Decision Tree* and *Bayes* families could generate the prediction faster. *MLP* is the one requiring more execution time compared to the others.

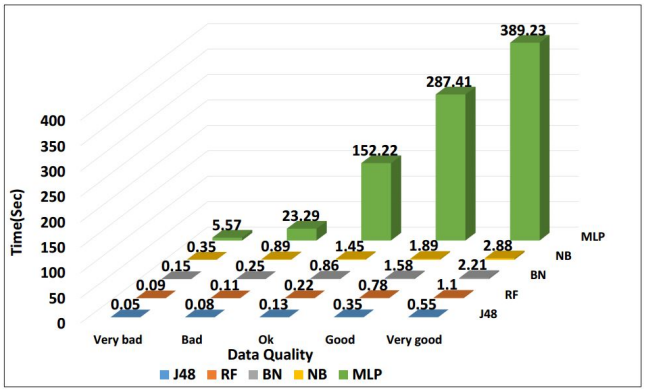


Figure 16: *Average execution time of individual algorithms using Temporal+Spatial features.*

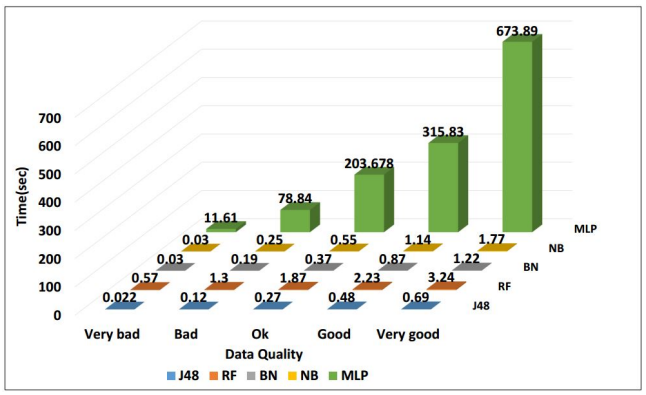


Figure 17: *Average execution time of individual algorithms using Hybrid features.*

6.2.4. Location Prediction Execution Time of Ensemble Methods

Fig. 18 - 23 present the average execution time of *Boosting*, *Bagging* and *Stacking* learning methods, using temporal+spatial and hybrid features. The results show that *Boosting* outperforms *Bagging* for different algorithms. When *J48* and *MLP* are combined using *Stacking*, the execution time is 12'012 seconds for very good quality traces and 109 seconds for very bad quality traces. When *J48*, *Bayes Networks* and *MLP* are combined with *Stacking*, the execution time is 15'078 seconds for very good quality traces and 187 seconds for very bad quality traces.

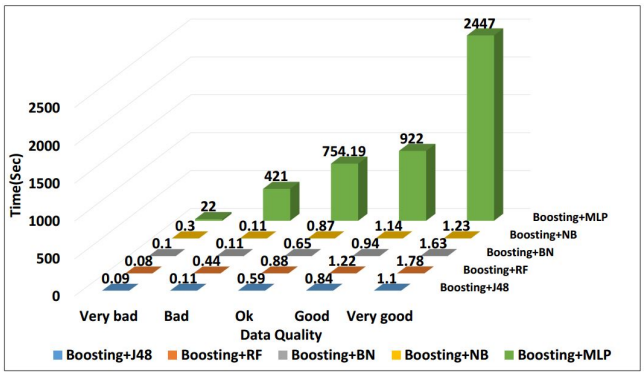


Figure 18: Average execution time of *Boosting with Temporal+Spatial features.*

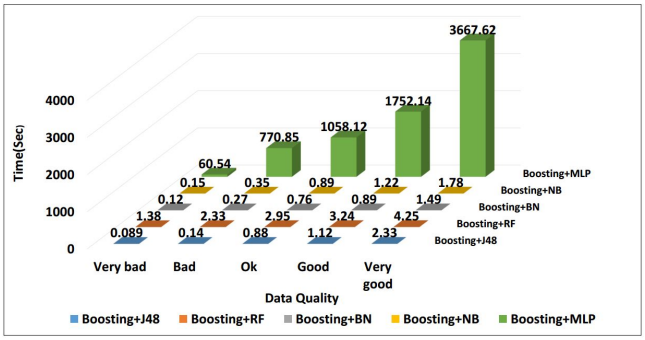


Figure 19: Average execution time of *Boosting with Hybrid features.*

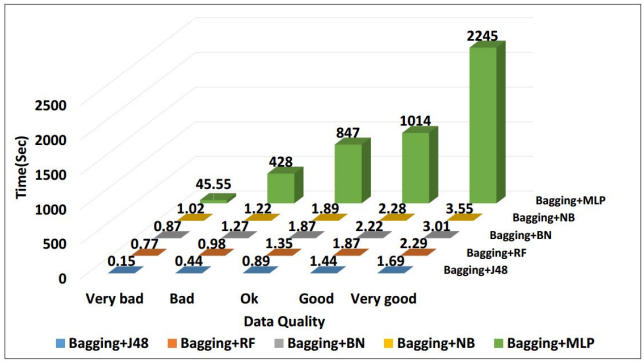


Figure 20: Average execution time of *Bagging with Temporal+Spatial features.*

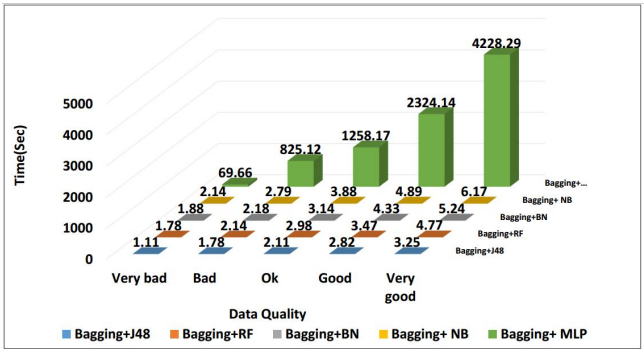


Figure 21: Average execution time of *Bagging with Hybrid features.*

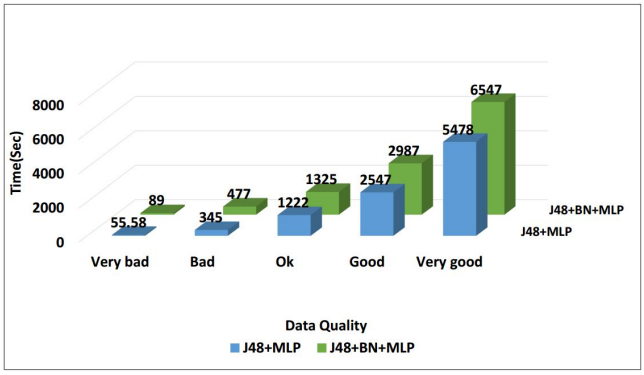


Figure 22: Average execution time of *Stacking with Temporal+Spatial features.*

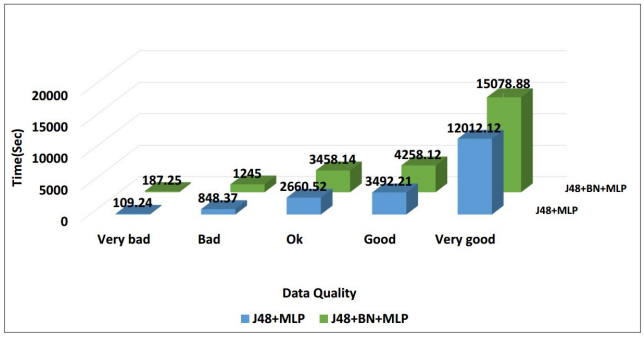


Figure 23: Average execution time of *Stacking with Hybrid features.*

6.2.5. Trajectory Prediction Accuracy

This subsection discusses the results of our trajectory prediction algorithm. We take user 5927 as an example, whose mobility trace includes both homogeneous and heterogeneous movement patterns. Fig. 24 - 25 show the predicted trajectories for one transition with connected cells between location ID 2 to 3 and 4 to 5 for user 5927, in which black dots are GPS coordinates, red circles indicate the frequently visited places, and the yellow circles are the sequence of cells that the user will be connected between the places. Since the exact coverage areas of the GSM cells are not known, we estimated their position by calculating the mean position of the user within a time window of one minute when a GSM entry was registered. As shown in Fig. 26, our proposed adaptive Markov Chain model could achieve a trajectory prediction accuracy of nearly 80% for homogeneous movements and 70% for heterogeneous movements for user 5927.

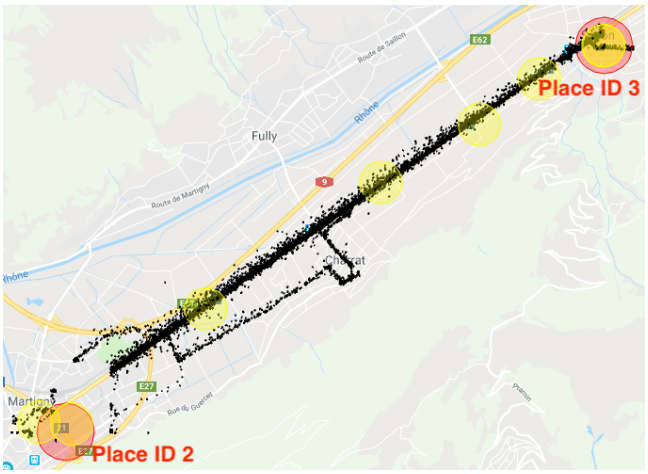


Figure 24: Trajectory prediction of user 5927 between location ID 2 and 3.

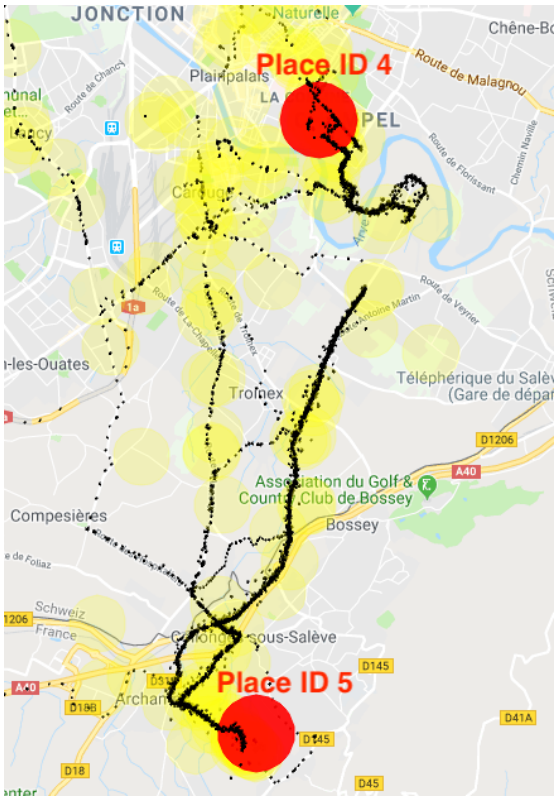


Figure 25: Trajectory prediction of user 5927 between location ID 4 and 5.

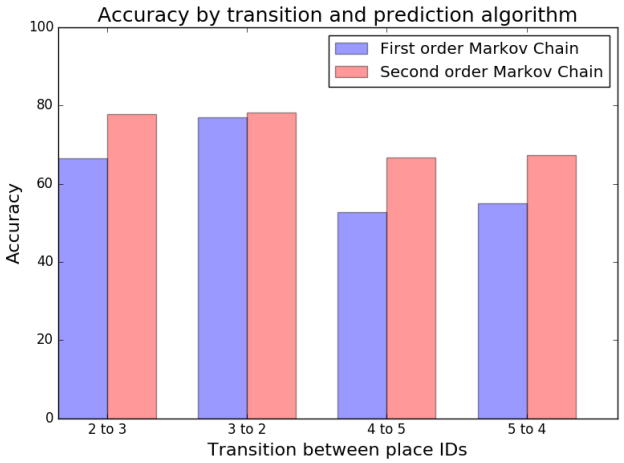


Figure 26: Trajectory prediction accuracy of user 5927 between place IDs

6.3. Location and Trajectory Prediction Accuracy Comparison with Past Studies

In this section, we present the location and trajectory prediction performance comparison with past correlated studies to show the superiority of our solutions. For location prediction, we take relevant location prediction accuracy results from [16] [6], which are the winners of the mobility prediction task in the Nokia Mobile Data Challenge. The results are shown in Table 5.

Table 5: Accuracy comparison of location prediction approaches.

Work	**Algorithms**	**Features**	**Best Accuracy (%)**
Our work	Stacking	Hybrid features	83.37
HKUST [16]	Gradient Boosting Trees	Limited hybrid features	76.32
EPFL [6]	Blending	Temporal features	56.22

As we can see from Table 5, our solutions significantly outperform the others. This is because in [6], authors applied the *Blending* technique, which is an ensemble learning approach similar to *Stacking*, to deliver the best accuracy using only temporal features. They considered information such as starting/ending time of a visit, the visit is on weekday or weekend. In [16], authors explored temporal and smartphone system features with the *Gradient Boosting Trees* approach. However, they did not consider a wide range of features as we did. For instance, they only used the mean and variance of visit duration at a place for the temporal features. Therefore, by applying ensemble learning using a wide range of hybrid features, our solutions provide the best performance.

For trajectory prediction, we compare our work to the trajectory estimation using the adaptive, mean and F-score optimization threshold [22]. All methods were implemented using the GSM cell representation of the trajectories as input data. We use the performance metrics of precision and recall, as defined in the work of [22]. The results are shown in Table 6.

Table 6: Precision and recall comparison of trajectory prediction approaches.

Work	**Method**	**Precision (%)**	**Recall (%)**
Our work	Markov Chain	81.92	68.00
Chapuis et al. [22]	Mean threshold	60.30	92.81
	F_1 optimization threshold	70.42	89.51
	Adaptive threshold	70.42	89.51

As shown in the Table 6, our methods outperform all trajectory estimation methods proposed in work of Chapuis et al. [22] significantly in terms of precision but it is outperformed in terms of recall. The lower recall number can be explained by looking at how the proposed adaptive Markov Chain predicts full trajectories. When dealing with predicting the full trajectory starting from one place, the adaptive Markov Chain sequentially adds the next most probable cell to the predicted trajectory. Considering the case that starting from some cells the transition probabilities to two different cells is high, one of them will be left out from the prediction, since only the cell with the highest transition probability is added. Therefore, for evaluating the performance of the adaptive Markov Chain-based trajectory prediction mechanism, it is better to use prediction accuracy as a metric. This is because as opposed to other methods [22], the cells in the trajectory are predicted in order. Using the Markov Chain as a predictor also has the advantage that after the mis-prediction of a cell, a new trajectory starting from the actual cell can be generated.

6.4. Algorithm Complexity Analysis

In this subsection, we present computational complexity of individual and ensemble algorithms. In machine learning, model complexity often depends on the number of extracted features and samples in the training set. *Decision trees* are the fastest known algorithms, the run time cost to construct a *decision tree* is $O(n_{samples} m_{features} log(n_{samples}))$. In general, the *Bayes Networks* are powerful algorithms and efficient in terms of execution time. Their run time is $O(2^{m_{features}-2}(m_{samples} n_{features}))$[27]. $n_{samples}$, $m_{features}$ represent number of records in training set and number of features, respectively. *MLP* has a high

Table 7: Time complexity comparison

Learning algorithm	**Complexity**
Decision tree (DT)	O(28,800)
Bayes network (BN)	O(614,400)
MLP	O(500×10^7)
Boosting + DT	O(144×10^3)
Bagging + DT	O(288×10^3)
Stacking + (DT + MLP + BN)	O(501×10^8)

time complexity. Suppose that there are $n_{samples}$ training samples, $m_{features}$ features, k hidden layers, each containing h neurons and o output neurons. The time complexity of *MLP* is $O(n \times m \times k^h \times o \times i)$, where i is the number of iterations. Since *MLP* has a high execution time, it is advisable to start with a smaller number of hidden layers for training [28]. In ensemble learning, execution time of *meta-learners* is negligible and they have not much impact on running time of base classifiers. Running time of *Boosting* is $O(L \times f)$, where f is the runtime of the base classifier and T is number of iterations. Time complexity for *Bagging* is $O(N \times f)$, where N is number of new generated training sets and f is run time of individual algorithm [29]. *Stacking* applies several individual learner to training data and then combines output of them using a meta-learner. The overall complexity of stacking is $O(f_1 + f_2 + f_3, ..., f_n)$ n=1,...,N, where f_n denotes time complexity of each individual learner. Table 7 presents time complexity comparison for individual and ensemble learning algorithms. For this experiment we choose user 5925 with 1200 records in the training set and 8 extracted features. As we can observe, the time complexity follows the same ordering of execution time as shown in Fig.17, 19, 21, and 23.

6.5. Theoretical Analysis

In this section, we analyze the performance of different predictors from a mathematical perspective. We aim to find out the impacting factors of difference predictors, and understand theoretically why they have different performance under different conditions.

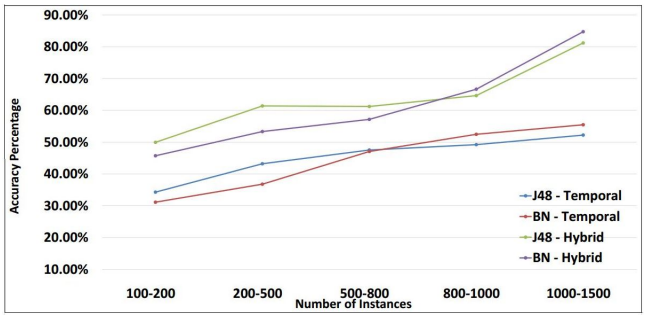


Figure 27: Prediction accuracy of *J48* and *Bayes Networks*

6.5.1. Analysis of Individual Algorithms Performance

Section 6.2.1 presents the prediction accuracies of individual predictors. As we can see from the results, Decision Tree-based approaches (especially the *J48* algorithm) outperform others when the trace data is of lower quality, while the *Bayes Networks* scheme provides better performance ($> 84\%$ accuracy) for trace data with higher quality. To better understand these behaviors, we highlight the performance comparison of *J48* and *Bayes Networks* by decomposing the mathematical components of each model to explain why different predictors have different performance. Fig. 27 shows the average prediction accuracy of the *J48* and *Bayes Networks* algorithms using temporal or hybrid features as a function of trace qualities, which are summarized in Fig. 8. It is interesting to observe that for both cases, *J48* outperforms *Bayes Networks* when the quality of traced data is low (e.g., with 100-500 instances). This is due to the fact that the algorithms relying on the decision tree use the *surrogate splits* approach, which is a method to estimate missing data, to overcome the deficit of missing data on the trace files [30]. However, *Bayes Networks* do not have a future action in presence of a trace file with a lot of missing data, and its prediction is based only on available data.

When making a prediction, J48 estimates the missing instances based on the present ones, resulting in higher accuracy of the prediction outcomes. The missing instances can be either numerical attributes (e.g., leaving time, duration

of staying in each place, place id, etc), or nominal attributes (e.g., application type, etc), whose values could be missing randomly. The missing attribute parameters with nominal value can be estimated based on available instances with the same attribute. Assuming that the day of visiting a particular place (e.g., Place-ID = 1) for a user is missing, the surrogate split approach [31] can estimate the missing value (e.g., day of a visit), knowing that (using users previous trajectories) on which day the user often visits the location with the same Place-ID. Our problem can be modelled by Eq. 1.

$$\widehat{V}_{i,j} \approxeq argmax_{v_{i,j} \in (a_i)} | \sigma_{a_i} = v_{i,j} \quad and \quad y = y_p^i \quad D| \tag{1}$$

$\widehat{V}_{i,j}$ defines the estimated parameter, $v_{i,j}$ represents the missing parameter of attribute a_i with index j, σ_{a_i} includes the subset of missing parameters for attribute a_i, y_p^i shows the value of the target attribute (e.g., *duration_time*, *application_type*) and D is the provided data set. If the missing parameter of attribute a_i has a numerical value, the estimation is performed by calculating the *mean* (*average*) of the existing data instances with the same attribute. The outcome of the estimation of the *decision tree-based* algorithms is more similar to the original data if there is no continuously missing data on the trace files. As shown in Fig. 27, the *J48* and *Bayes Networks* algorithms generate similar results if the trace data is of low quality (e.g., with 100-200 instances). *J48* performs better for improved quality of trace data (e.g., with 200-500 instances). However, it is interesting to observe that *Bayes Networks* overtake *J48*, if the quality is better (e.g., with 700-1500 instances). This is due to the fact that *Bayes Networks* follows a graphical model, making possible relations between the parameters with particular probabilities [32]. When the number of existing instances raises, the generated graph used in the model requires more computation overhead, but resulting in more accurate prediction. The graph is integrated with a set of local probability distributions to define the joint probability distribution [33]. The joint probability distribution is defined in Eq. 2.

$$Pr(X|m,\theta) = \Pi_{i=1}^{n} Pr(X^i|\Pi(X^i),\theta) \quad (2)$$

X^i, denotes attributes in DAG, $\Pi(X^i)$ shows the set of parents (e.g., Place-ID = 1, Place-ID = 2,...), θ is a vector of conditional probabilities, m represents the DAG model and local probability distributions are the distributions corresponding to the terms in the product of Eq. 2.

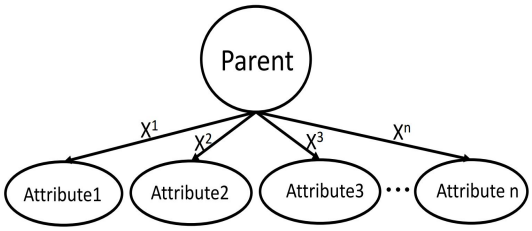


Figure 28: Directed Acyclic Graph (DAG) of Bayes Networks.

6.5.2. Analysis of Ensemble Learning Algorithm Performance

As presented in Fig. 13 and 14, the experiment results show that the integration of the individual algorithms (e.g., *J48*, *Bayes Networks* and *MLP*) using *ensemble learning* methods can efficiently improve prediction accuracy. This is because for machine learning algorithms, the bias error and variance error, as explained in Eq. 3, are the main components of the prediction errors. However, all ensemble learning methods are able to mitigate these errors such that the prediction performance could be enhanced. The bias error defines the difference between values of the expected prediction (*average of estimated predictions*) and the real one. The variance error determines the variability of the prediction accuracy due to small modifications in the training set.

$$\begin{aligned} Err(X) &= bias\ error^2 + variance\ error + noise\ error \\ &= (E[g(x)] - f(x))^2 + E[(g(x) - E[g(x)])^2] + \epsilon_e^2 \end{aligned} \quad (3)$$

$f(x)$, $g(x)$, $E[g(x)]$ and ϵ_e^2 denote the correct value to predict ($Place\_ID$), estimated prediction calculated by the algorithm, expected prediction, and noise

error, respectively. Ensemble predictors can be applied to enhance the prediction performance of individual algorithms by mitigating the variance error.

Even though ensemble learning could deliver better prediction accuracy than individual algorithms, they also perform differently according to how they address the variance error. *Bagging* does this by creating N new subsets of training data with the same size, as shown in Table 4. The new data sets are generated from the original data, randomly sampled and replaced [34]. Therefore, the total variance (Z) will be decreased as it is divided among the newly generated training data sets. Variance of each new subset can be calculated using Eq. 4.

$$Variance_j = \frac{1}{N} Var(Z) \quad j = 1, ..., N \tag{4}$$

For *Bagging*, the training phase is performed independently over all the new data sets. Later, as shown in Eq. 5, the final prediction accuracy ($Pr_{Bagging}$) is obtained by getting a *simple-averaging* over the outcomes computed in each new data set (e_j). This implies that there is no mechanism in *Bagging* to specify whether the parameters are classified correctly or not. This means that all the parameters appear with the same probability in newly generated data sets [35].

$$Pr_{Bagging} = \frac{1}{N} \Sigma_{j=1}^{N} e_j \quad j = 1, ..., N \tag{5}$$

Boosting applies a sequential model in the learning phases [36]. After each iteration, the weights of parameters are determined based on the current prediction error, as shown in Eq. 6. Next, the weights are assigned to uncorrected classified parameters. Therefore, the wrongly-classified parameters will appear in the new training set with bigger weights than the correctly classified ones. This repetition decreases the diversity of the parameters in the training sets, which results in a reduction of the variance and consequently a better prediction performance. The parameters used in this equation 6 are listed in Table 8.

$$w_t^{h+1} = \frac{w_t^h \beta_h^{(1-l_h^t)}}{\Sigma_{i=1}^{N} w_i^h \beta_h^{(1-l_h^t)}}, \quad w_t^1 \in [0, 1], \quad \Sigma_{t=1}^{N} w_t^1 = 1 \tag{6}$$

Table 8: The notations and definition of parameters

Parameter Name	**Parameter Definition**
$w_t^1 = [1, ..., w_N]$	Set of possible weights for the first step of iterations, usually $w_t^1 = \frac{1}{N}$
$h = 1, ..., L$	Number of iterations in *Boosting*
$l_t^h = 0, 1$	Prediction in iteration h is incorrect (=0) / correct (=1)
$\beta_h^{(1-l_h^t)}$	Current prediction error of algorithm in iteration h
w_t^h	Current weight at iteration h
w_t^{h+1}	Calculated weight for iteration h+1

For *Stacking*, different kinds of individual algorithms can be integrated to improve performance. *Stacking* achieves this through two steps. Firstly, the given data set of $D = \{(y_n, x_n), n = 1, ..., N\}$ is randomly split into J smaller data sets (parameters defined in Table 4). The generated sets have almost equal sizes, denoted by the $d_1, ..., d_J$. Thereafter, the individual algorithms (*level-0 algorithms*) carry out prediction on the generated data sets independently [37]. The outcomes of each prediction algorithm (e.g., visited place in our scenario) can be defined using Eq. 7:

$$z_{kn} = \{(P_k^{(d_1)}(x_n), ..., P_k^{(d_j)}(x_n)), k = 1, ..., K, \quad n = 1, ..., N\} \tag{7}$$

$P_k^{(d_j)}(x_n)$ denotes the prediction of individual algorithms for each instance x in the newly generated data sets (d_j). Later, a new data set is created using the IDs of the visited places (y_n) and the output of the K individual algorithms (z_{kn}). Formally, the new data set is represented as:

$$L_{Level-1} = \{(y_n, z_{1,n}, ..., z_{k,n}), n = 1, ..., N\} \tag{8}$$

$L_{Level-1}$ defines the input data for the second step, including the predicted values for each visited place. This input is different from the one for the first step. The input of the first step includes the Place-ID and extracted features from the trace data. Next, the *meta-learner* (*Level-1 algorithm*) uses the *Weighted Majority* method [38][39] to further improve prediction accuracy. *Weighted Majority* is an approach to decide weights of each algorithm based on their individual prediction performances.

Based on the aforementioned description, we could imagine that a particular algorithm could only provide a low prediction accuracy, due to the high variance of the data set used in the learning phase. Afterwards, the *Weighted Majority* method can be applied to enhance the accuracy of the final prediction by getting benefits of other algorithms, which provides more accurate results.

7. Conclusions

In this paper, we model the future place prediction problem as a standard supervised learning task and ensemble learning methods with hybrid types of features. Our approach characterizes the properties of users' movement patterns and visited places, then extracts rich types of features (temporal, spatial, and system features) to quantify the correlation between places and features. Finally, we propose to use ensemble learning approaches to predict users' future locations. Additionally, we also propose an adaptive Markov Chain-based model for trajectory prediction. Our system is extensively evaluated using real-world datasets, and experiment results unveil interesting findings: (1) For individual predictors, Bayes Networks outperform all others when data quality is good, while J48 delivers the best results when data quality is bad; (2) Ensemble predictors outperform individual predictors in general under all conditions; and (3) Ensemble predictor performance depends on user movement patterns.

Acknowledgments

This work has been supported by the Swiss National Science Foundation via the SwissSenseSynergy project (154458).

References

[1] J. K. Laurila, D. Gatica-Perez, I. Aad, B. J., O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, M. Miettinen, The mobile data challenge: Big data for mobile computing research, in: Pervasive Computing, 2012.

[2] Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/index.html (November 2016).

[3] A. Kirmse, T. Udeshi, P. Bellver, J. Shuma, Extracting patterns from location history, in: ACM SIGSPATIAL GIS 2011, http://www.sigspatial.org/, 2011, pp. 397–400.

[4] A. Monreale, F. Pinelli, R. Trasarti, F. Giannotti, Wherenext: A location predictor on trajectory pattern mining, in: Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '09, ACM, 2009, pp. 637–646.

[5] Z. Ying, S. Yong, W. Yu, Nokia mobile data challenge: Predicting semantic place and next place via mobile data, in: Proceedings of Mobile Data Challenge by Nokia, 2012.

[6] V. Etter, M. Kafsi, E. Kazemi, M. Grossglauser, P. Thiran, Where to go from here? mobility prediction from instantaneous information, Pervasive and Mobile Computing 9 (6) (2013) 784 – 797.

[7] R. Lambiotte, A. Noulas, M. Pontil, S. Scellato, C. Mascolo, A tale of many cities: Universal patterns in human urban mobility.

[8] C. Song, Z. Qu, N. Blumm, A.-L. Barabasi, Limits of predictability in human mobility 327 (5968) (2010) 1018–1021.

[9] D. Ashbrook, T. Starner, Fusing gps to learn significant locations and predict movement across multiple users, Personal Ubiquitous Computing.

[10] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, A. T. Campbell, Nextplace: A spatio-temporal prediction framework for pervasive systems, in: Proceedings of the 9th International Conference on Pervasive Computing, 2011.

[11] M. Karimzadeh, Z. Zhao, L. Hendriks, R. d. O. Schmidt, S. la Fleur, H. van den Berg, A. Pras, T. Braun, M. J. Corici, Mobility and bandwidth

prediction as a service in virtualized lte systems, in: Cloud Networking (CloudNet), 2015 IEEE 4th International Conference on, IEEE.

[12] H. He, Y. Qiao, S. Gao, J. Yang, J. Guo, Prediction of user mobility pattern on a network traffic analysis platform, in: Proceedings of the 10th International Workshop on Mobility in the Evolving Internet Architecture, ACM, 2015, pp. 39–44.

[13] V. Etter, M. Kafsi, E. Kazemi, Been there, done that: What your mobility traces reveal about your behavior, in: Proceedings of Mobile Data Challenge by Nokia Workshop, 2012.

[14] W. Jingjing, P. Bhaskar, Periodicity based next place prediction, in: Proceedings of Mobile Data Challenge by Nokia Workshop, 2012.

[15] G. Huiji, T. Jiliang, L. Huan, Mobile location prediction in spatio-temporal context, in: Proceedings of Mobile Data Challenge by Nokia, 2012.

[16] L. Zhongqi, Z. Yin, Z. Vincent, Y. Qiang, Next place prediction by learning with multiple models, in: Proceedings of Mobile Data Challenge by Nokia, 2012.

[17] T. Le-Hung, C. Michele, M. Luke, A. Karl, Next place prediction using mobile data, in: Proceedings of Mobile Data Challenge by Nokia, 2012.

[18] T. M. T. Do, O. Dousse, M. Miettinen, D. Gatica-Perez, A probabilistic kernel method for human mobility prediction with smartphones, Pervasive and Mobile Computing 20 (C) (2015) 13–28.

[19] Y. Zhu, E. Zhong, Z. Lu, Q. Yang, Feature engineering for semantic place prediction, Pervasive and mobile computing 9 (6) (2013) 772–783.

[20] D. Opitz, R. Maclin, Popular ensemble methods: An empirical study, Journal of Artificial Intelligence Research 11 (1999) 169–198.

[21] L. Rokach, Ensemble-based classifiers, Artificial Intelligence Review 33 (1-2) (2010) 1–39.

[22] B. Chapuis, A. Moro, V. Kulkarni, B. Garbinato, Capturing complex behaviour for predicting distant future trajectories, in: Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems, MobiGIS '16, ACM, 2016, pp. 64–73.

[23] E. Tuv, A. Borisov, G. Runger, K. Torkkola, Feature selection with ensembles, artificial variables, and redundancy elimination, Journal of Machine Learning Research 10 (Jul) (2009) 1341–1366.

[24] R. E. Schapire, Explaining adaboost (2015).

[25] M. G. Elfeky, W. G. Aref, A. K. Elmagarmid, Periodicity detection in time series databases, IEEE Transactions on Knowledge and Data Engineering 17 (7) (2005) 875–887.

[26] S. Koço, C. Capponi, On multi-class learning through the minimization of the confusion matrix norm, CoRR abs/1303.4015.

[27] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: Machine learning in Python, Journal of Machine Learning Research 12 (2011) 2825–2830.

[28] L. Buitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. VanderPlas, A. Joly, B. Holt, G. Varoquaux, API design for machine learning software: experiences from the scikit-learn project, in: ECML PKDD Workshop: Languages for Data Mining and Machine Learning, 2013, pp. 108–122.

[29] T. G. Dietterich, Ensemble methods in machine learning, in: Proceedings of the First International Workshop on Multiple Classifier Systems, MCS '00, Springer-Verlag, London, UK, UK, 2000, pp. 1–15.

[30] L. Rokach, O. Maimon, Data Mining with Decision Trees: Theroy and Applications, World Scientific Publishing Co., Inc., River Edge, NJ, USA, 2008.

[31] R. J. A. Little, D. B. Rubin, Statistical Analysis with Missing Data, John Wiley & Sons, Inc., New York, NY, USA, 1986.

[32] D. Heckerman, Learning in graphical models, MIT Press, Cambridge, MA, USA, 1999, Ch. A Tutorial on Learning with Bayesian Networks, pp. 301–354.

[33] C. Fiot, G. A. P. Saptawati, A. Laurent, M. Teisseire, Learning Bayesian Network Structure from Incomplete Data without Any Assumption, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008, pp. 408–423.

[34] B. Efron, Bootstrap methods: Another look at the jackknife, Ann. Statist. 7 (1) (1979) 1–26.

[35] B. Efron, S. for Industrial, A. Mathematics, The jackknife, the bootstrap, and other resampling plans, Philadelphia, Pa. : Society for Industrial and Applied Mathematics, 1982, notes from ten lectures given at Bowling Green State Univ., June 1980. Bibliography pp 91-92.

[36] R. Meir, G. Rätsch, Advanced lectures on machine learning, Springer-Verlag New York, Inc., New York, NY, USA, 2003, Ch. An Introduction to Boosting and Leveraging, pp. 118–183.

[37] K. M. Ting, I. H. Witten, Stacked generalization: when does it work?, in: in Procs. International Joint Conference on Artificial Intelligence, Morgan Kaufmann, 1997, pp. 866–871.

[38] L. Breiman, Stacked regressions, Machine Learning 24 (1) (1996) 49–64.

[39] S. Nagi, D. K. Bhattacharyya, Classification of microarray cancer data using ensemble approach, Network Modeling Analysis in Health Informatics and Bioinformatics 2 (3) (2013) 159–173.

Mobile Users Location Prediction with Complex Behavior Understanding

Mostafa Karimzadeh, Zhongliang Zhao, Florian Gerber, Torsten Braun
Institute of Computer Science, University of Bern, Switzerland
Email : {karimzadeh, zhao, braun}@inf.unibe.ch, florian.gerber@students.unibe.ch

***Abstract*—The growing ubiquity of smart-phones equipped with built-in sensors and global positioning system (GPS) has resulted in the collection of large volumes of mobility data without the need of any additional devices. The large size of heterogeneous mobility data gives rise to rapid development of location-based services (LBSs). The predictability of mobile users' behavior is essential to enhance LBSs. To predict human mobility, many techniques have been proposed. However, existing techniques require good data quality to guarantee optimal performance. In this paper, we proposed a hybrid Markov chain to predict mobile users' future locations. Our model constantly adapts to available user trace quality to select either the first order or the second order Markov chain. Compared to existing solutions, our model is adaptive to discrete gaps in data trace. In addition, we implemented a proper mechanism to predict congestion in city areas. To help us understanding complex user behaviors, we have also proposed a technique benefiting both temporal and spatial parameters to extract Zone of Interests (ZOIs). To evaluate the algorithms performance, we use a real-life dataset from the Nokia Mobile Data Challenge (MDC) collected around Lake Geneva region from 180 users. We found a satisfactory user future location prediction accuracy of $70-84\%$ and area congestion prediction accuracy of $65-73\%$ for the users.**

***Index Terms*—Mobile analysis, Mobility and Congestion Prediction, Mobility Behavior, Location based Services.**

I. Introduction

Extracting meaningful information from collected trace data of users to determine their movement pattern is an important part of location-based services (*LBSs*). For example, to predict future behavior of a mobile user, mobility predictors rely on clustering techniques to capture a user's Individual Zone of Interests (*I-ZOIs*) from collected trace data. Intuitively, an I-ZOI is a city area that is frequently visited by a user and the user spends considerable time in this region. Typically, LBSs are using mobility prediction as a means to improve quality of service by providing context-aware information to users beforehand.

In recent years, we have seen a rapid proliferation in the number of services such as *Google Now*, which proactively collects rich contextual data, such as Bluetooth/WiFi connectivity, call/SMS log, information about running applications, to deliver information to users that they may need in daily life activities. Another popular service, *Moves* enables automatically recording of any walking, cycling and running of users and displaying pertinent information, such as traveled distance, duration and calories burned for each activity. Similarly, *Google Maps* is a web mapping service to predict future location of users based on the movement history. It is apparent from the above examples that location based services are thriving, providing a remarkable opportunity to collect fine grained data about visited places of users. This source of user mobility offers new possibilities to tackle established research problems on human mobility. In addition to mobility prediction, area congestion prediction in large cities is also of great importance. The past decades have witnessed a rapid development of modern cities accompanied with an increasing demand for mobility [1], accounting for the conflict between the limited resource capacities and the increment of traffic demand reflected by severe user congestion in hot spot regions. Induced by such a problem, several negative impacts arise for citizens, e.g., economic losses, reduction of travel efficiency and accessing to resources. Fortunately, smart-phone and LBS data have been employed to explore and predict congested areas in cities.

The type of dataset plays an important role in accurate location prediction as the prediction algorithms learn user movement patterns from collected data. To evaluate performance of implemented algorithms, we use a real-life dataset from the Nokia Mobile Data Challenge, collected around Lake Geneva from 180 users.

In this work, the fundamental goal is to formulate the location prediction problem using a Mobility Markov Chain model (*MMC*). We propose a dynamic Markov chain-based model, which adaptively selects the first-order or the second-order Markov chain model based on the available trace quality. We propose a clustering algorithm to discover frequently visited places by the user and integrate it with a mobility predictor to predict a user's future behavior. Moreover, in order to estimate number of users in frequently visited places we propose a congestion prediction algorithm. The system overview is depicted in Figure 1. The contributions of the paper can be summarized as follows:

- We design a mobility prediction algorithm that benefits from both the first-order Markov chain and the second-order Markov chain to forecast users' future location.
- We propose the Zone of Interest discovery scheme, which helps us to model the mobility behavior of users.
- We introduce a proper mechanism to predict congestion in frequently visited places by users.
- We evaluate our methodology using a real-life dataset, obtaining consistently satisfactory results.

The remainder of the paper is organized as follows. Section

II discusses research efforts. Section III presents our system model and introduces some formal definitions and notations used in the paper. Sections IV and V discuss the methodology to evaluate the mobility model and demonstrate obtained results respectively. Finally, Section VI concludes the paper by sketching future research directions.

II. Related Work

Understanding human mobility by mining raw GPS logs has been a long-standing subject in academic research [2], [3]. These approaches rely on clustering user visits to extract hot spots. Clustering is a form of unsupervised learning that groups objects that are similar into the same cluster while putting objects that are dissimilar into different clusters. The premier research in adapting the data clustering algorithms for modeling mobility behavior [4] proposes to iteratively extract hot spots of users. Montoliu et al. proposed a clustering algorithm [5], where GPS coordinates (*latitude and longitude*) are clustered in the temporal domain to detect the stay points that are used to derive frequently visited regions using a grid-based clustering approach. Several other clustering algorithms such as Density-Time (DT) clustering [6], Density-Join able (DJ) [7] and Time-Density (TD) clustering [8] have also been proposed to detect clusters, which are then considered as frequently visited places.

The above techniques use several temporal bounds to classify a particular region as a cluster. Some parameters include maximum distance between the collected locations, maximum and minimum time bound, cluster shapes. However, if we only take into account the temporal bounds, this leads to some inaccuracies in estimating the total number of clusters belonging to a user. As opposed to using only temporal metrics to cluster the individual regions, we form the clustering algorithm by benefiting both temporal and spatial metrics (*instantaneous velocity, average velocity*) to quantify the correlation between visited regions.

Regarding the prediction techniques, a majority of existing works first formulate a mobility model and consequently use it to make prediction [9],[10]. Numerous map-matching algorithms have been proposed to predict user mobility. However, most existing map-matching techniques are not always practicable and need additional services such as network connectivity [11]. Such schemes completely ignore the trace data with poor quality and just consider users with qood quality of trace data. Our work consists of estimating the frequently visited locations, in which a user spends considerable amount of time, and then attaining the hybrid Markov chain model, which adaptively selects from the first-order or the second order Markov chain, depending on the quality of user traces to predict future behavior of the users.

Forecasting of urban congestion has become an active research topic thanks to the fast development of continuous location tracking techniques. A wide spread techniques derives from pure time-series models like Autoregressive Integrated Moving Average (*ARIMA*) [12], [13], a fine tuned version of random walk algorithm. In [14], Olszewski uses Markov chains to obtain the probability distribution of overflow queue. The algorithm estimates the mean queue and its variance under different conditions such as stationary and non-stationary arrival processes. However, the existing techniques are not well suited to predict time of congestion. Therefore, we utilize the area congestion predictor with a tunable time threshold, which means the temporal granularity of the algorithm is tunable and based on application requirements it could be adjusted in seconds or minutes scales.

III. System Model

The system overview is depicted in Figure 1. The proposed system model involves three main layers: Common Zone of Interest (*C-ZOI*) Discovery, Individual Zone of Interest (*I-ZOI*) Prediction and Common Zone of Interest (*C-ZOI*) Congestion Prediction. The relevant notations and system component definitions are explained in following subsections.

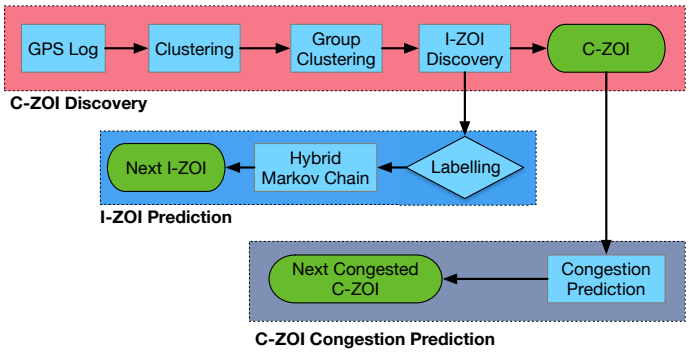


Fig. 1: Overview of the System Model.

A. Locations and Places

Mobile devices have the potential to track users' visited locations using the Global Positioning System (*GPS*). The device regularly collects a user's raw location logs as a list $L = [loc_1, loc_2, loc_3, ..., loc_n]$, where $loc_i = (\alpha, \beta, t, v)$ is a tuple representing a location point in the format $(latitude, longitude, timestamp, velocity)$. The rest of the paper uses the notations $loc.\alpha$, $loc.\beta$, $loc.t$ and $loc.v$ for the location point elements. In addition to GPS coordinates, users' daily activities consist of some places that they find useful or where they spend a considerable amount of time. This work focuses on places, which we refer to Zone of Interest (*ZOI*) hereafter.

B. Common Zone of Interest (C-ZOI) Discovery

Intuitively, an Individual Zone of Interest (*I-ZOI*) is a cluster group that is frequently visited by a user and a Common Zone of Interest (*C-ZOI*) depicts a region covering several of I-ZOIs. In order to extract the C-ZOIs of users based on the location history L, we must first introduce the notations of cluster and cluster group. A cluster represents a hot spot in an encapsulated area. It includes a subset of locations with similar temporal (*e.g., visiting time, visitation repeatability rate*) and spatial (*e.g., speed, acceleration*) features. A cluster group is an aggregation of intersected clusters.

1) Cluster Discovery: Location points with common temporal and spatial characteristics are considered as a cluster. Δd_{max}, e_{max}, e_{min} and $v \in \mathbb{R}$, represents the distance expressed in meters, maximum velocity, minimum velocity and instantaneous velocity expressed in meters per second, respectively. In addition, $\Delta t_{min} \in \mathbb{N}$ is a time duration expressed in minutes. We introduce two functions: $ClusterCentroid\left([loc_1, loc_2, loc_3, ..., loc_n]\right)$, to compute the centroid of visited location points, and $Distance\left([loc_i, loc_j]\right)$, which measures the Euclidean distance between the two locations loc_i and loc_j. A subset $l \subseteq L$ becomes a cluster if the following conditions in Equation 1, 2, 3 and 4 are met:
$\forall loc_i, loc_{i+1} \in l$:

$$Distance\left(centroid\left(loc_1, ..., loc_i\right), loc_{i+1}\right) \leq \Delta d_{max} \quad (1)$$

$$loc_n.t - loc_1.t \geq \Delta t_{min} \quad (2)$$

$$e_{min} \leq \sum_{i=1}^{n} \frac{v_i}{n} \leq e_{max} \quad (3)$$

$$\nexists l' \neq \quad : \quad l \subset l' \quad (4)$$

A cluster is a 4-item tuple $c = (\alpha_c, \beta_c, \Delta r, l)$, where α_c and $\beta_c \in \mathbb{R}$ are the latitude and longitude coordinates of the centroid, $\Delta r \in \mathbb{R}$ is it's radius in meters, and $l \in L$ is the subset of locations belonging to c. The average of all $loc.\alpha$ and $loc.\beta$ of the locations contained in the subset l is the centroid (α_c, β_c) of the cluster, which is designated as $c.centroid$. We introduce C, the set of clusters extracted from the location log of a user as $C = \{c_1, c_2, c_3, ..., c_n\}$. Disjointness of discovered clusters can not be guaranteed by equations 1, 2, 3 and 4 that is in turn to explain cluster group construction in next step.

2) Cluster Group Discovery: A cluster group includes a set of overlapped clusters. Thus, we define equation 5 to check whether two clusters c_i, $c_j \in C$ are intersected or not.

$$\begin{aligned} Distance\left(c_i.centroid, c_j.centroid\right) \\ - \left(c_i.\Delta r + c_j.\Delta r\right) < 0 \end{aligned} \quad (5)$$

A cluster group is a 4-item tuple $cg = (\alpha_{cg}, \beta_{cg}, \Delta r, \{c_1, c_2, c_3, ...\})$, where α_{cg}, β_{cg} and $\Delta r \in \mathbb{R}$, $\{c_1, c_2, c_3, ...\} \in C$ are latitude, longitude, radius and array of clusters constituting g respectively. $(\alpha_{cg}, \beta_{cg})$ represents the centroid of the cluster group, which is the mean of all the clusters formed g, and Δr must be compared to enclose all the individual clusters present in g. G contains the n cluster groups belonging to a user as $G = \{cg_1, cg_2, cg_3, ..., cg_n\}$

3) Individual Zone of Interest (I-ZOI): An I-ZOI refers to a visited region by a user frequently and during daily activities. We define two constants $minCountThreshold$ and $maxTimeDifference \in \mathbb{N}$ representing the minimum threshold of visits and the maximum time difference threshold between two consecutive visits, respectively. Then, $CountVisits(cg)$ is a function that counts the number of clusters included in cluster group cg, and $timeDuration(G)$ is a function that returns the duration between two consecutive visited dates of cluster group cg in G. A cluster group $cg \in G$ is transformed into an I-ZOI z if the conditions of Equation 6 and 7 are met:

$$CountVisits(cg) \geq minCountThreshold \quad (6)$$

$$timeDifference(G) \leq maxTimeDifference \quad (7)$$

An I-ZOI z is a 6-item tuple $z = (\alpha_z, \beta_z, \Delta r, ID_{zone}, \{g_1, g_2, g_3, ..., g_n\}, T_{ID})$, where α_z, β_z and $\Delta r \in \mathbb{R}$, $ID_{zone} \in \mathbb{N}$ and $\{g_1, g_2, g_3, ..., g_n\} \in G$ are the latitude, longitude, radius, Zone-ID and group clusters becoming an I-ZOI. T_{ID} represents visiting dates of the I-ZOI by each user. The set Z is finally the set of I-ZOIs of the user such that $Z = \{z_1, z_2, z_3, ..., z_n\}$. As shown in Figure 2, the set of discovered cluster groups are depicted by intersected yellow circles. Finally, the cluster groups that could satisfy above conditions are considered as I-ZOIs.

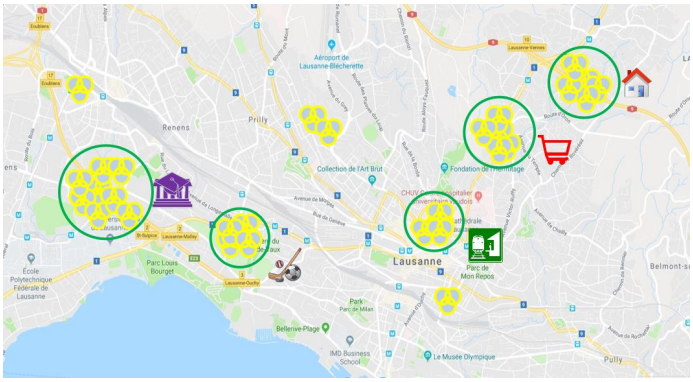


Fig. 2: I-ZOI construction from cluster groups of location points.

4) Common Zone of Interest (C-ZOI): A C-ZOI is an aggregation of adjacent I-ZOIs. We introduce a constant $distanceThreshold \in \mathbb{N}$ to represent the maximum threshold of distance between each I-ZOIs. $Distance\left(I - ZOI_i.centroid, I - ZOI_j.centroid\right)$ is a function to compute Euclidean distance between the $I - ZOI_i$ and $I - ZOI_j$. I-ZOIs are grouped whenever the following condition in Equation 8 is satisfied:

$$\begin{aligned} Distance\left(I - ZOI_i.centroid, I - ZOI_j.centroid\right) \\ \leq distanceThresold \end{aligned} \quad (8)$$

A C-ZOI is a 6-item tuple $C - ZOI = (\alpha_{cz}, \beta_{cz}, \Delta r, ID_{cz}, \{ZOI_1, ZOI_2, ZOI_3, ..., ZOI_n\}, T_{ID})$, where α_{cz}, β_{cz} and $\Delta r \in \mathbb{R}$, $ID_{cz} \in \mathbb{N}$ and $\{ZOI_1, ZOI_2, ZOI_3, ..., ZOI_n\} \in Z$ are the latitude, longitude, radius, C-ZOI-ID and group of I-ZOIs, respectively. The last item of the tuple indicates visiting dates of the C-ZOI by each user. Finally, we introduce CZ, which contains the n C-ZOIs belonging to users as $CA = \{C - ZOI_1, C - ZOI_2, C - ZOI_3, ..., C - ZOI_n\}$. Figure 3 shows an example of extracted C-ZOIs for a group of users. Each C-ZOI encapsulates a set of nearby I-ZOIs.

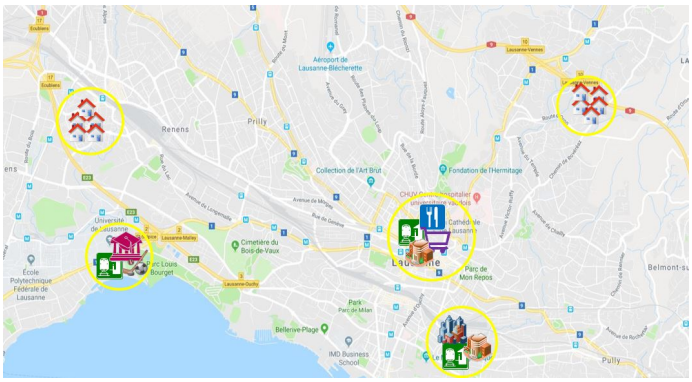


Fig. 3: C-ZOIs construction from I-ZOIs.

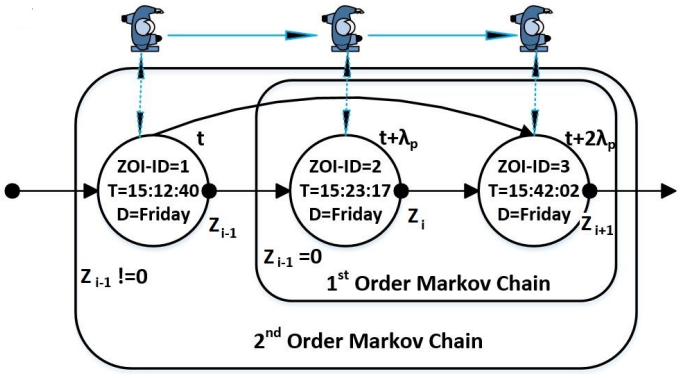


Fig. 4: Hybrid Markov Chain.

C. Individual Zone of Interest (I-ZOI) Prediction

This module predicts the users' future locations (*I-ZOI*). The mobility model represents the movement of mobile users and how their locations change over time. One of the intuitive methods to determine a mobile node's movement pattern, is the attempt to trace and capture some sort of regularity in the user mobility. Such behavior regularity could be considered as a user's profile and utilized to estimate places a user may visit in future. The proposed mobility prediction scheme in this paper is based on a hybrid Markov chain model, which adaptively selects from the first-order or the second-order Markov chain, depending on the availability and quality of user traces. The proposed hybrid Markov model benefits from both the first-order Markov chain [15] and the second-order Markov chain. The rationale behind using a hybrid predictor is that the standard first-order Markov chain algorithms are memoryless models, which means that the input for the next-place prediction task includes current visited location, current time and the day of week that the user is in the movement. The second-order Markov chain model is slightly different. In addition to current state, it benefits from previous state to predict future location. Indeed, such information is very useful: if a user is currently at a city center, e.g., a restaurant, knowing whether he/she was at work or at home just before greatly helps in predicting his/her next move. However, we experience for some users periods (*ranging from a few seconds to a few minutes*) with no information about their behavior, when trace data includes discrete gaps, the 2-order state information conditions will not met, which led to poor performance for the second-order Markov predictor.

The proposed hybrid model is illustrated in Figure 4, in which a Markov chain state consists of a time step and an I-ZOI ID. Equation 9 defines the calculation of future location probability, in which Z_i represents an I-ZOI with ID *i*, D indicates the day of the week (*e.g., Saturday*), T_i defines the time of the day D (*e.g., 13:22:43 h*), and λ_p determines the future time interval.

$$Pr(Z_{i+1}(t+2\lambda_p)) = \begin{cases} Pr(Z_{i+1}|Z(t+\lambda_p)=Z_i, & Z_{i-1}=0 \\ \quad D, T(t+\lambda_p)=T_i) & \\ Pr(Z_{i+1}|Z(t)=Z_{i-1}, Z(t+\lambda_p)=Z_i, & Z_{i-1}\neq 0 \\ \quad T(t)=T_{i-1}, T(t+\lambda_p)=T_i, D) & \end{cases} \tag{9}$$

Equation 9 can be considered as a location-dependent distribution and a time-dependent distribution (as expressed in Equations 10 and 13). As shown in Equation 13, both the time-dependency and location-dependency distributions in the second-order Markov chain model benefiting from current and previous state information (*e.g., time, day and location*). The location dependent distribution can be modeled as a Mobility Markov Chain (*MMC*). The MMC is described by a state-transition matrix including the user's I-ZOIs, which are the states and all the transitions among them. These transitions are collected during training period of the predictor by considering a tunable time threshold (*e.g., each minutes*) on the given days of the week (*e.g., all Wednesdays, all Thursdays and etc.*). For the case of the second-order Markov chain, the counting of transition frequency happens only when the user's movement is continuously following the sequence of two states.

$$Pr(Z_{i+1}|Z(t+\lambda_p)=Z_i, T(t+\lambda_p)=T_i, D) \tag{10}$$
$$=Pr(Z_{i+1}|Z(t+\lambda_p)=Z_i) \tag{11}$$
$$+Pr(T(t+\lambda_p)=T_i, D) \tag{12}$$

$$Pr(Z_{i+1}|Z(t)=Z_{i-1}, Z(t+\lambda_p)=Z_i, T(t)=T_{i-1}, T(t+\lambda_p)=T_i, D) \tag{13}$$
$$=Pr(Z_{i+1}|Z(t)=Z_{i-1}, Z(t+\lambda_p)=Z_i) \tag{14}$$
$$+Pr(T(t)=T_{i-1}, T(t+\lambda_p)=T_i, D) \tag{15}$$

D. Common Zone of Interest (C-ZOI) Congestion Prediction

From the probability distribution of the users' future visited I-ZOIs, we can further estimate the number of users that may visit and stay in a C-ZOI together at a future moment. Therefore, next, we target at predicting the probability distribution of the number of users that may visit together a

specific C-ZOI within a given time. As explained in Section III, nearby I-ZOIs are covered by C-ZOIs. In these regions users either do not move or they move very slowly and users are spending a considerable amount of time together in each C-ZOI. Each of these hot spot regions are candidates to host a significant number of users (*pedestrians*). Then, we define the *AreaCongestionThreshold (M)*, which refers to the number of predicted users in each C-ZOI. The congestion prediction model counts the number of predicted users in each C-ZOI. If the number of users exceed the defined threshold, we assume that this region will experience congestion within the next λ_c minutes. Predicting the number of users will help us to facilitate tasks such as resource management, logistic administration, and urban planning. For instance, if we know how many users will be in a specific C-ZOI between time t and $t+\lambda_c$ we could optimize placement of resources in the city or dynamically adapt those resources, while taking into account the number of users. Equation 16 defines the probability of having M users visiting $C-ZOI_i$ at time $t+\lambda_c$, which is derived from the estimated number of upcoming and outgoing of users in $C-ZOI_i$ at time $t+\lambda_c$. The parameters used in this equation are listed in Table I.

$$
\begin{aligned}
&P\left\{N_{C-ZOI_i}(t+\lambda_c)=M\right\}= \\
&\quad \sum_{m} P\left\{N_{C-ZOI_i}(t+\lambda_c)=M \mid N_{C-ZOI_i}(t)=m\right\} \times \\
&\quad\quad P\left\{N_{C-ZOI_i}(t)=m\right\}= \\
&\sum_{m} \langle \sum_{n_1,n_2,n_1-n_2=\triangle m} P\left\{N_{in,C-ZOI_i}(t+\lambda_c)=n_1\right\} \times \\
&P\left\{N_{out,C-ZOI_i}(t+\lambda_c)=n_2\right\}\rangle \times P\left\{N_{C-ZOI_i}(t)=m\right\} \qquad (16)
\end{aligned}
$$

In Equation 16, $P\left\{N_{in,C-ZOI_i}(t+\lambda_c)\right\}$ describes the probability of having n_1 users that may move into $C-ZOI_i$ at time $t+\lambda_c$ and $P\left\{N_{out,C-ZOI_i}(t+\lambda_c)\right\}$ indicates the probability of having n_2 users may moving out from $C-ZOI_i$ at time $t+\lambda_c$. These probabilities can be calculated using Equation 17.

$$
\begin{aligned}
P\left\{N_{in,C_i}(t+\lambda_c)=n_1\right\} &= \sum_{A_1 \in F_{C_{i'}(t)}} \prod_{j_1 \in A_1} P_{j_1} \prod_{j_1' \in A_1^c} \left(1-P_{j_1'}\right) \times \\
&\quad P\left\{N_{out,C_i}(t+\lambda_c)=n_2\right\} \\
&= \sum_{A_2 \in F_{C_i(t)}} \prod_{j_2 \in A_2} P_{j_2} \prod_{j_2' \in A_2^c} \left(1-P_{j_2'}\right) \qquad (17)
\end{aligned}
$$

IV. Evaluation

In this section, we present an evaluation methodology to validate the proposed user mobility and area congestion prediction models.

1) Dataset: In order to make mobility and congestion prediction, historical user traces are required. We used a mobility data trace collected during the Nokia Mobile Data Challenge (*MDC*) [16], which is a large-scale research initiative aimed at generating innovations around smart phone-based research. This dataset includes rich context information from the mobile phones for around 180 users around the lake Geneva region in Switzerland from October 2009 to March 2011. The mean duration of the participants, which mainly consisted of professionals and university students, was about 14 months. It includes Global Positioning System (*GPS*) information, running applications, chat records, calling records, etc. However, for mobility and congestion predictions, we are only interested in GPS location information, which are more than 10 million location points. For each user, we separated available data into two parts: the first part is the learning data set (L) and the rest is the testing data set (T). The learning data set includes the first 70% of user data. It is used to drive the states for both algorithms and to determine their transition probability matrix. The testing data set T contains the last 30% of the traces, which is used to test and evaluate the accuracy of the proposed prediction algorithms.

TABLE I: Area congestion prediction algorithm parameters.

Parameter Name	Parameter Definition
Z_i	State i in the Markov chain
$Pr\left\{Z_{i+1}(t)\right\}$	Probability of at State $(i+1)$ at t
$C-ZOI_i$, U_j	C-ZOI ID i, User ID j
D, T_i	Weekday and time of being at State i
λ_c, t	Future time interval and current time
$C=\{C-ZOI_1,..,i\}, \lvert C\rvert = I$	Set of C-ZOIs, I is the total numbers
$U=\{User_1,..,j\}, \lvert U\rvert = J$	Set of Users, J is the total numbers
$N_{C-ZOI_i}(t)$, $N_{C-ZOI_i}(t+\lambda_c)$	Number of Users in C-ZOI i at time t and t+λ_c (*e.g.,* m and M)
$N_{in,C-ZOI_i}(t+\lambda_c)$	Number of Users that may move to C-ZOI i at time t+λ_c (*e.g.,* n1)
$N_{out,C-ZOI_i}(t+\lambda_c)$	Number of Users that may move from C-ZOI i at time t+λ_c (*e.g.,* n2)
$F_{C-ZOI_i(t)}$	Subset of all users in $C-ZOI_i$ at time t
$F_{C-ZOI_{i'}(t)}$	Subset of all users out of $C-ZOI_i$ at time t
P_{j_1} , $User_{j_1} \in F_{C-ZOI_{i'}(t)}$	$P\{User_{j_1}$ *is in* $F_{C-ZOI_{i'}(t)}$ *at time* $t\}$ $\times$ $P\{User_{j_1}$ *moves to* $C-ZOI_i$ *at time* t+$\lambda_c\}$
P_{j_2} , $User_{j_2} \in F_{C-ZOI_i(t)}$	$P\{User_{j_2}$ *is in* $F_{C-ZOI_i(t)}$ *at time* $t\}$ $\times$ $P\{User_{j_2}$ *moves from* $C-ZOI_i$ *at time* t+$\lambda_c\}$

2) User Trace Quality: Different behaviors of users lead to different trace qualities. Some users carry the smart-phone all the time. However, some others forgot to carry the devices or had to charge them, such that data recordings are non-continuous. As learned from our previous experiences [17] [18], the number of valid states (*with a time stamp and I-ZOI ID*) in the drived hybrid Markov chain for each user depend on the quality of data trace in each day. Therefore, we first classify the dataset into two groups (*good or poor quality*) based on the number of recorded instances during the whole data collection period. We choose five users with good quality of trace data (*e.g., 500000-400000 records*) and five users with poor quality of trace data (*e.g., 250000-350000 records*).

3) Evaluation Metrics: Prediction accuracy measures the accuracy of the location prediction algorithm. We randomly select 10% of the states out of all the Markov chain states (*e.g., states from* 9 *AM to* 11 *AM*) derived for each particular weekday from the training data set L for each user. Afterwards, the prediction algorithm is performed for each of the selected states to estimate the possible future visited I-ZOI(s) for mobility prediction in the next λ_p minutes. These states have been chosen as random testing points. We check the transition

probability for states during the same period of time in the testing data set T as well. Afterwards, the Mean Absolute Error (*MAE*) of the possible transitions of the corresponding testing points is calculated according to Equation 18. To evaluate performance of the area congestion predictor we define two metrics: (*i*) density of users, which counts the number of users that may move to each C-ZOI; (*ii*) area congestion prediction accuracy, which represents probability of moving users to a C-ZOI in a specific day of week and is calculated by average of future location prediction accuracies of users in each C-ZOI drived from Equation 18.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \mid Pr_iL - Pr_iT \mid, Accuracy = (1 - MAE) \times 100 \quad (18)$$

4) Experimental Settings: We describe the experimentation parameters of the discussed clustering, mobility prediction, and congestion prediction algorithms. In order to determine the parameters we analyze traced data for users with at least 10 months duration of collected data. Then, we read the data-points sequentially according to the recorded time stamps. Table II shows the experiment parameters and the associated values in our assessment.

TABLE II: Experiments parameters.

Parameter	Definition	Value
Δd_{max}	Maximum Euclidean distance between centroid of location points and next location	60 m
Δt_{min}	Minimum time threshold of staying in each location point	15 min
e_{max}	Maximum instantaneous velocity threshold	50 *m/s*
e_{min}	Minimum instantaneous velocity threshold	0
M	Number of predicted users in each C-ZOI	6
minCountTh	Minimum number of visits of each cluster group	60
maxTimeDiff	Maximum time difference between two consecutive visits of a cluster group by a user	24 *h*
DistanceTh	Maximum Euclidean distance between two I-ZOIs	500 *m*
λ_p	Time threshold for hybrid-MC algorithm	1 *min*
λ_c	Time threshold for area congestion prediction algorithm	15 *min*

V. Evaluation Results

1) Mobility Prediction Accuracy Results: This subsection details the prediction accuracy results of the proposed hybrid predictor, the first order and the second order Markov chain. We first present the average prediction accuracy of all the users with different trace qualities. Then, we discuss more details about the prediction accuracy per day, for users with poor and good trace qualities.

Figures 5 and 6 show the prediction accuracy of different MMC predictors for users with good and poor quality of mobility traces. We define two categories of quality depending on the number of instances recorded in a user's movement traces. We randomly choose 5 User IDs (5973, 5928, 5993, 5977, 5925) from the group of good quality trace data, and 5 User IDs (6177, 5927, 5969, 6037, 5961) from the group of poor quality trace data. As we can see from Figure 5, the hybrid predictor can deliver a average prediction accuracy over all weekdays and weekends of nearly 83% for User-ID 5928. Moreover, it can be observed from Figure 6 that the estimated accuracy is improved significantly when the hybrid predictor

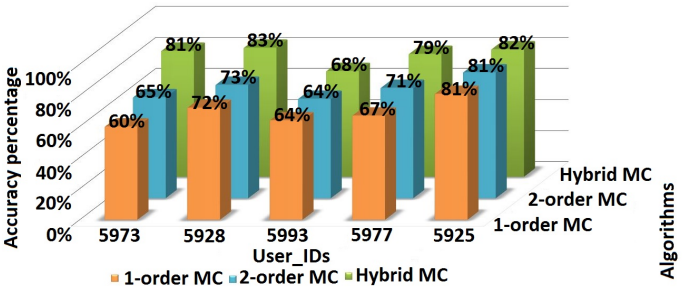


Fig. 5: Prediction accuracy for users with good quality.

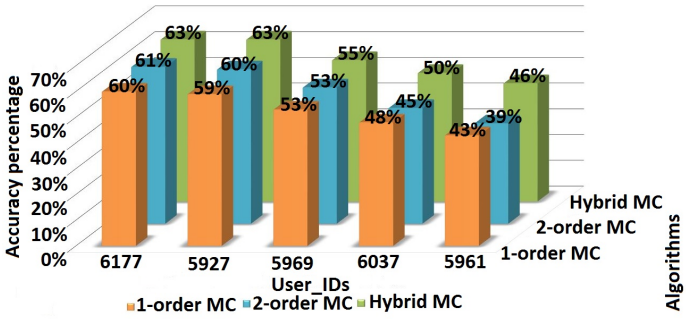


Fig. 6: Prediction accuracy for users with poor quality.

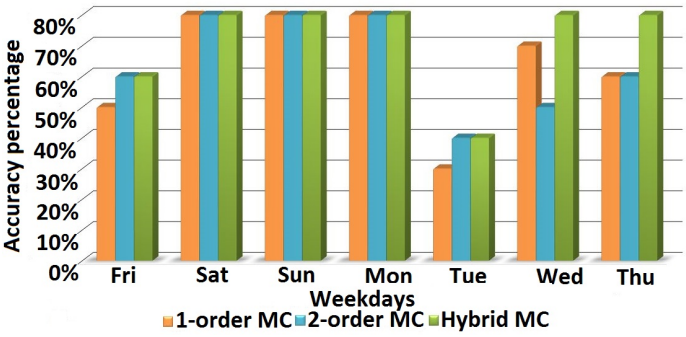


Fig. 7: Prediction accuracy per day for User-5928.

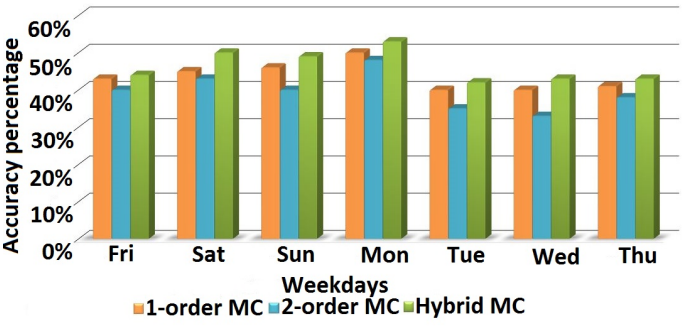


Fig. 8: Prediction accuracy per day for User-6037.

used for users with poor quality trace data. For instance, it delivers an average accuracy of 63% for User-IDs 6177 and 5927. The results clearly demonstrate that the hybrid predictor outperform others, while using the traced data with either poor or good quality. Figures 7 and 8 show the prediction accuracy of three different predictors per each day. This helps us to explain the advantages of the hybrid predictor compared to the

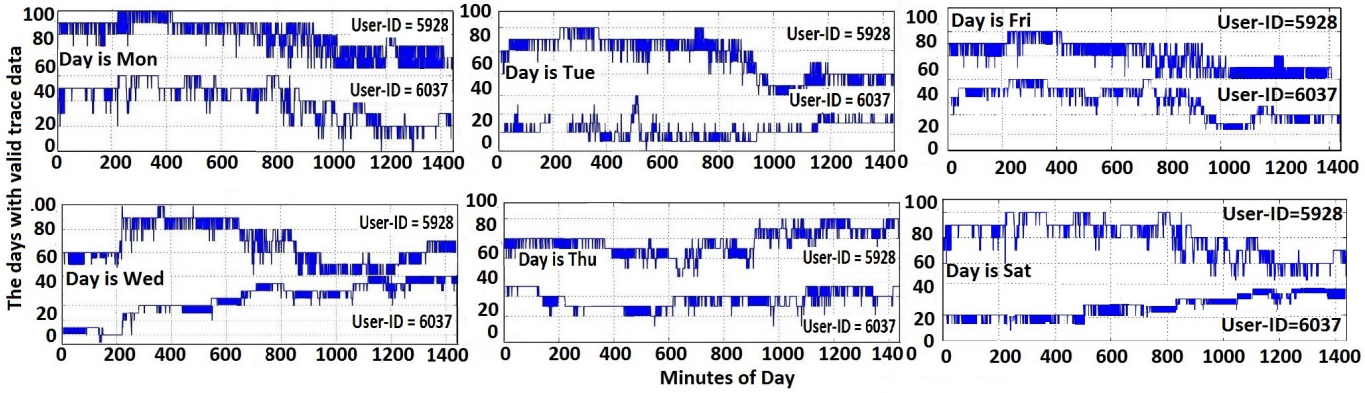


Fig. 9: Trace qualities of User_IDs 5928/6037 on weekdays.

first-order and second-order Markov chains. From defined user categories we randomly select User-ID 5928 and User-ID 6037 as the representatives of users with good and poor qualities. The graphs show that the hybrid predictor performs better than the first-order and the second-order MC predictors for both categories. To explain the performance difference of mobility predictors for these two users, we next discuss the data quality of User-ID 5928 and 6037. Figure 9 depicts mobility traces of users over a year, shown as a matrices, where each column is a minute of the day and each line indicates the number of days with valid trace data (*with time stamp and GPS coordinates*). We map each interval of valid records to continuous pulses, and leave blank intervals time during which we have no information about users' locations. To count the number of days with valid trace records, we introduce a *threshold*, which counts the days with more than 1500 records as valid days for prediction. If a user has less than 1500 records in one day. Then data of that specific day will not be included in the prediction. This is because such a low number of records happen most probably due to imperfect geolocation sensors or network unavailability. Therefore, collected traces for these users are not valid and should not be included in the prediction procedure. Figure 6 illustrates that for User-ID 6037, the hybrid predictor can only deliver an accuracy of around 52% for Monday. This situation arises typically because location data are partly available. For User-ID 5928, due to having continuous intervals of collected GPS records at a high number of days with valid trace data between 80 to 100, the hybrid predictor has improved performance (81% *to* 83%) for all weekdays.

2) Congestion Prediction Accuracy Results: In addition to estimating future locations of mobile users, we are also interested in area congestion prediction. In this subsection, we present the prediction accuracy of the congestion prediction algorithm. Then we discuss more details about the number of predicted users in each Common-ZOI.

We explore the predictability of congestion by using GPS records. We focus on the extracted Common-ZOIs in Lausanne by predicting the number of users that may move and stay together in each common hot spot. Figure 10 depicts the results of the congestion prediction algorithm for time-of-days (*08:00 h, 12:00 h, 18:00 h and 22:00 h*). The graph shows that the congestion prediction algorithm achieves accuracies exceeding 70% for C-ZOIs. In addition to congestion prediction accuracy, we also count the number of users in each Common-ZOI for time-of-days (*08:00 h, 12:00 h, 18:00 h and 22:00 h*). Figure 11 shows the variation of population during the corresponding hour. We observe that in the evening the population tends to move towards the city suburbias (*C-ZOI 1, C-ZOI 5*), going back home for dinner. An opposite trend is observed at 08:00 AM and 12:00 PM, when there is a significant inflow towards the city center or universities (*C-ZOI 2, C-ZOI 3 and C-ZOI 4*). Although we can not compare these population densities against a proper ground truth, we remark that the model represents very reasonable results that match well to the movements of inhabitants in the city of Lausanne.

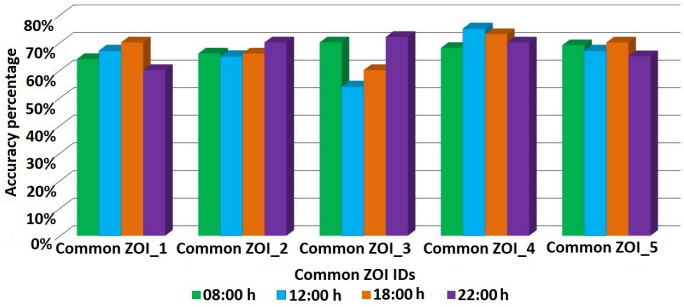


Fig. 10: Area congestion prediction accuracy.

VI. CONCLUSIONS

With the explosive growth of location-based service on mobile devices, predicting users' future locations is of increasing importance to support proactive information services. In this paper, we introduce a hybrid predictor to estimate future locations of a user. Further, we propose a technique to discover hot spot regions for users by relying on spatial and temporal constraints. The achieved results over real world mobility traces validates our proposed algorithms, which achieve more than 81% correct predictions for users. More important, we present a novel approach to predict congestion in hot spot regions using GPS coordinates. This achieves accuracy exceeding 70% for discovered Common-ZOIs from the available dataset.

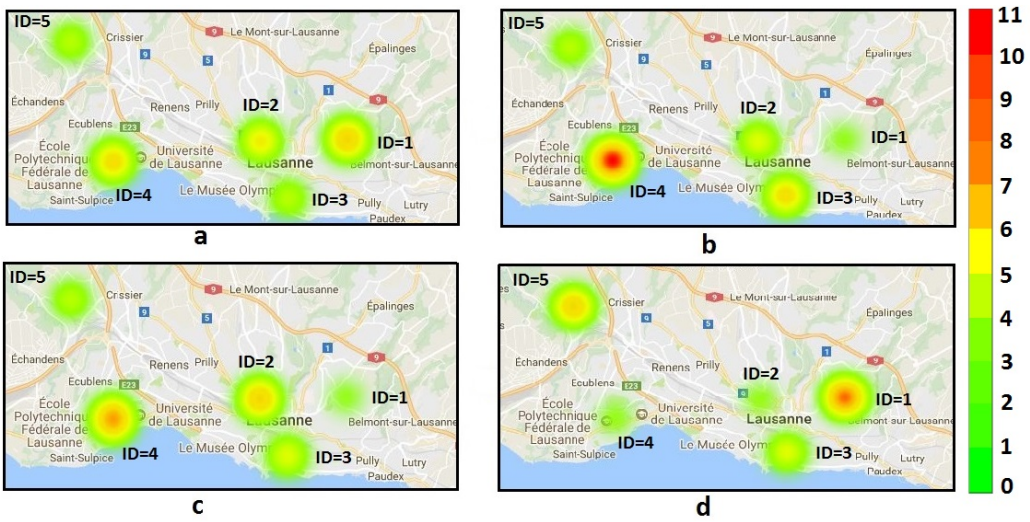


Fig. 11: Density of users in each Common-ZOI. a) Wednesday at 08:00 h. b) Wednesday at 12:00 h. c) Wednesday at 18:00 h. d) Wednesday at 22:00 h.

For future improvements we will focus on predicting trajectories of mobile users that they use for transition among I-ZOIs. We will also improve our area congestion prediction algorithm by applying congestion classification models to classify the predicted congestion to *slight congestion*, *moderate congestion* and *severe congestion*.

ACKNOWLEDGMENT

This work has been supported by the Swiss National Science Foundation with project number 154458.

REFERENCES

[1] G. Khodabandelou, V. Gauthier, M. A. El-Yacoubi, and M. Fiore, "Population estimation from mobile network traffic metadata," *CoRR*, vol. abs/1610.06947, 2016. [Online]. Available: http://arxiv.org/abs/1610.06947

[2] X. Chen, J. Pang, and R. Xue, "Constructing and comparing user mobility profiles," *ACM Trans. Web*, vol. 8, no. 4, pp. 21:1–21:25, Nov. 2014. [Online]. Available: http://doi.acm.org/10.1145/2637483

[3] V. Kulkarni, A. Moro, and B. Garbinato, "Mobidict: A mobility prediction system leveraging realtime location data streams," in *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming*, ser. IWGS '16. New York, NY, USA: ACM, 2016, pp. 8:1–8:10. [Online]. Available: http://doi.acm.org/10.1145/3003421.3003424

[4] T. N. Maeda, K. Tsubouchi, and F. Toriumi, "Next place prediction in unfamiliar places considering contextual factors," in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL'17. ACM, 2017, pp. 76:1–76:4.

[5] R. Montoliu and D. Gatica-Perez, "Discovering human places of interest from multimodal mobile phone data," in *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '10. ACM, 2010, pp. 12:1–12:10.

[6] "Identifying locations from geospatial trajectories," *J. Comput. Syst. Sci.*, vol. 82, no. 4, pp. 566–581, Jun. 2016. [Online]. Available: https://doi.org/10.1016/j.jcss.2015.10.005

[7] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personal gazetteers: An interactive clustering approach," in *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems*, ser. GIS '04. New York, NY, USA: ACM, 2004, pp. 266–273. [Online]. Available: http://doi.acm.org/10.1145/1032222.1032261

[8] R. Hariharan and K. Toyama, "Project lachesis: Parsing and modeling location histories," in *Geographic Information Science*, M. J. Egenhofer, C. Freksa, and H. J. Miller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 106–124.

[9] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Semantic trajectory mining for location prediction," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '11. New York, NY, USA: ACM, 2011, pp. 34–43. [Online]. Available: http://doi.acm.org/10.1145/2093973.2093980

[10] S. Gambs, M.-O. Killijian, and M. N. n. del Prado Cortez, "Next place prediction using mobility markov chains," in *Proceedings of the First Workshop on Measurement, Privacy, and Mobility*, ser. MPM '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:6. [Online]. Available: http://doi.acm.org/10.1145/2181196.2181199

[11] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson, "Approaching the limit of predictability in human mobility," *Scientific Reports*, vol. 3, p. 2923, Oct. 2013.

[12] L. Li, Y. Wang, G. Zhong, J. Zhang, and B. Ran, "Short-to-medium term passenger flow forecasting for metro stations using a hybrid model," *KSCE Journal of Civil Engineering*, Jul 2017. [Online]. Available: https://doi.org/10.1007/s12205-017-1016-9

[13] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with arima-garch model." in *Intelligent Vehicles Symposium*. IEEE, 2011, pp. 607–612. [Online]. Available: http://dblp.uni-trier.de/db/conf/ivs/ivs2011.html#ChenHMZ11

[14] P. S. Olszewski, "Modeling probability distribution of delay at signalized intersections," *Journal of advanced transportation*, vol. 28, no. 3, pp. 253–274, 1994.

[15] M. Karimzadeh Motallebi Azar, Z. Zhao, L. Hendriks, R. de Oliveira Schmidt, S. la Fleur, H. van den Berg, A. Pras, T. Braun, and M. Julian Corici, *Mobility and Bandwidth Prediction as a Service in Virtualized LTE Systems*. IEEE, 10 2015, pp. 132–138, 10.1109/CloudNet.2015.7335295.

[16] I. Yaqoob, I. A. T. Hashem, A. Gani, S. Mokhtar, E. Ahmed, N. B. Anuar, and A. V. Vasilakos, "Big data," *Int. J. Inf. Manag.*, vol. 36, no. 6, pp. 1231–1247, Dec. 2016. [Online]. Available: https://doi.org/10.1016/j.ijinfomgt.2016.07.009

[17] B. Sousa, Z. Zhao, M. Karimzadeh Motallebi Azar, D. Palma, V. Fonseca, P. Simoes, T. Braun, H. van den Berg, A. Pras, and L. Cordeiro, *Enabling a Mobility Prediction-aware Follow-Me Cloud Model*. United States: IEEE Computer Society, 11 2016, pp. 486–494, eemcs-eprint-27394.

[18] Z. Zhao, M. Karimzadeh Motallebi Azar, T. Braun, A. Pras, and H. van den Berg, *Cloudified Mobility and Bandwidth Prediction in Virtualized LTE Networks*. IEEE, 7 2017.

Pedestrians Complex Behavior Understanding and Prediction with Hybrid Markov Chain

Mostafa Karimzadeh, Zhongliang Zhao, Florian Gerber, Torsten Braun
Institute of Computer Science, University of Bern, Switzerland
Email : {karimzadeh, zhao, braun}@inf.unibe.ch, florian.gerber@students.unibe.ch

***Abstract*—The prevalence of smartphones equipped with global positioning system (*GPS*) has enabled researchers to excavate users mobility patterns in the cities. The knowledge of users' behavior, such as their locations, plays significant role in location-based services (*LBSs*), resource management, logistic administration and urban planning. To understand complex behavior of humans we utilize spatio-temporal analysis on collected geo-location points to exploit Individual Zone of Interests (*I-ZOIs*) in urban areas. In addition, we designed a hybrid Markov chain model to forecast future locations of pedestrians. Compared to existing mobility prediction methodologies, our predictor can adapt it's behavior constantly based on the quality of existing traced data to switch between first-order or second-order Markov chain. Besides, we propose a model to predict city area congestion. More specifically, the model predicts the number of users in a specific area of a city by discovering the regular mobility patterns of a group of users. We conducted comprehensive empirical experiments using a real-life dataset, namely the Mobile Data Challenge (*MDC*) dataset, which was collected in the city of Lausanne in Switzerland with around 180 participants. We found a satisfactory user future location prediction accuracy of $70-84\%$ and area congestion prediction accuracy of $65-73\%$ for the users.**

***Index Terms*—Mobile analysis, Mobility and Congestion Prediction, Mobility Behavior, Location based Services.**

I. Introduction

Extracting meaningful information from collected trace data of users to determine their movement pattern is an important part of location-based services (*LBSs*). For example, to predict future behavior of a mobile user, mobility predictors rely on clustering techniques to capture a user's Individual Zone of Interests (*I-ZOIs*) from collected trace data. Intuitively, an I-ZOI is a city area that an individual user visits frequently and the user spends considerable time in this region. Typically, LBSs are using mobility prediction as a means to improve quality of service by providing context-aware information to users beforehand.

In the last decade, with the increasing adoption of services such as *Google Now*, which proactively collects data, such as Bluetooth/WiFi connectivity, call/SMS log, information about running applications, large size of heterogeneous data is accumulated. This knowledge is essential for humans in their daily life activities. Another popular service, *Moves* enables automatic recording of any walking, cycling, and running of users and displays pertinent information, such as traveled distance, duration and calories burned for each activity. Similarly, *Google Maps* is a web mapping service to predict future location of users based on the movement history. It is apparent from the above examples that location based services are prospering, giving a notable chance to collect contextual data about visited location of users. This source of user mobility provides new possibilities to probe human mobility in large cities. In addition to mobility prediction, area congestion prediction in large cities is also of great importance. The past decades have witnessed a rapid development of modern cities accompanied with an increasing demand for mobility [1], accounting for the conflict between the limited resource capacities and the increase of traffic demand reflected by severe user congestion in hot spot regions. Induced by such a problem, several negative impacts arise for citizens, e.g., economic losses, reduction of travel efficiency and accessing to resources. Fortunately, accumulated data from smartphones and LBSs have been used to forecast congested areas in smart cities.

The type of dataset plays an important role in accurate location prediction as the prediction algorithms learn user movement patterns from collected data [2]. To examine prediction performance of proposed models, we use a large scale, real-world dataset from the Nokia Mobile Data Challenge, collected in the city of Lausanne by almost 180 participants.

In this work, the fundamental goal is to formulate the location prediction problem using a Mobility Markov Chain model (*MMC*). We propose a dynamic Markov chain-based model, which adaptively selects the first-order or the second-order Markov chain model based on the available trace quality. We propose a clustering algorithm to discover frequently visited places by the user and integrate it with a mobility predictor to predict a user's future behavior. Moreover, in order to estimate number of users in frequently visited places we propose a congestion prediction algorithm. The system overview is depicted in Figure 1. The contributions of our paper are as follows:

- We design a mobility prediction algorithm that benefits from both the first-order Markov chain and the second-order Markov chain to forecast users' future location.
- We propose the Zone of Interest discovery scheme, which helps us to model the mobility behavior of users.
- We introduce a mechanism to predict congestion areas that are frequently visited by users.
- We evaluate our mobility and congestion predictors using a real-life dataset, obtaining consistently satisfactory results.

The remainder of the paper is organized as follows. Section II discusses research efforts. Section III presents our system model and introduces some preliminaries used in the paper. Sections IV and V discuss the methodology to evaluate the mobility model and demonstrate obtained results respectively. Finally, in Section VI we conclude our paper by sketching future research directions.

II. Related Work

Understanding human mobility by mining raw GPS logs has been a long-standing subject in academic research [3], [4]. These approaches rely on clustering user visits to extract hot spots. A clustering algorithm attempts to partition m observed objects into n clusters, where each cluster is characterized with the similarity of objects within a cluster. The premier research in adapting the data clustering algorithms for modeling mobility behavior [5] proposes to iteratively extract hot spots of users. Montoliu et al. proposed a clustering algorithm [6], where GPS coordinates (*latitude and longitude*) are clustered in the temporal domain to detect the stay points that are used to derive frequently visited regions using a grid-based clustering approach. Several other clustering algorithms such as Density-Time (DT) clustering [7], Density-Join able (DJ) [8] and Time-Density (TD) clustering [9] have also been proposed to detect clusters, which are then considered as frequently visited places.

The above techniques use several temporal bounds to classify a particular region as a cluster. Some parameters include maximum distance between the collected locations, maximum/minimum time bound of visited places and cluster shapes. However, if we only take into account the temporal bounds, this leads to some inaccuracies in estimating the total number of clusters belonging to a user. As opposed to using only temporal metrics to cluster the individual regions, we form the clustering algorithm by benefiting both temporal and spatial metrics (*instantaneous velocity, average velocity*) to quantify the correlation between visited regions.

Regarding the mobility prediction methods, a majority of proposed models first explore movement patters and consequently employ it to predict next movements [10],[11]. Numerous map-matching algorithms have been proposed to predict user mobility [12],[13]. However, most existing map-matching algorithms are not always feasible and need continuous network connection [14]. In recent years mobility Markov chain (*MMC*) algorithms have been used widely to forecast future behavior of users, due to their simplicity, low execution time and good prediction performance [15]. In [16] authors have found that Markov-based algorithms are performing better than more complex and more memory consuming algorithms such as Sampled Pattern Matching (*SPM*) or Prediction by Partial Matching (*PPM*). The author in [2] proves that Neural network based approaches suffer from high computation complexity. In [17] authors proposed Markov-based algorithm to predict next location using non-Gaussian data. Using higher order Markov-based algorithms proposed in [18]. The authors in [19],[20],[21] integrate Markov predictors with other algorithms to improve prediction performance. Such schemes completely ignore the trace data with poor quality and just consider users with good quality of trace data. Our work consists of estimating the frequently visited locations, and then attaining the hybrid Markov chain model, which adaptively chooses from the first-order or the second order Markov chain, based on the quality of mobility trace to predict future behavior of the users.

Due to continued developments in location tracking techniques, forecasting of urban congestion has become an active research topic. In [22] and [23] authors proposed to use Autoregressive Integrated Moving Average (*ARIMA*) algorithm to analyze time series data. In [24], the author uses Markov-based algorithm to calculate the probability distribution of overflow queue. The algorithm is able to estimate mean queue and its variance for stationary and non-stationary arrival processes. However, the existing techniques are not well suited to predict time of congestion. Therefore, we utilize the area congestion predictor with a tunable time threshold, which means according to our requirements the algorithm can determine time of users' congestion in scales of *seconds* or *minutes*.

III. System Model

The system overview is depicted in Figure 1. The proposed system model involves three main layers: Common Zone of Interest (*C-ZOI*) Discovery, Individual Zone of Interest (*I-ZOI*) Prediction and Common Zone of Interest (*C-ZOI*) Congestion Prediction. The relevant notations and system component definitions are explained in the following subsections.

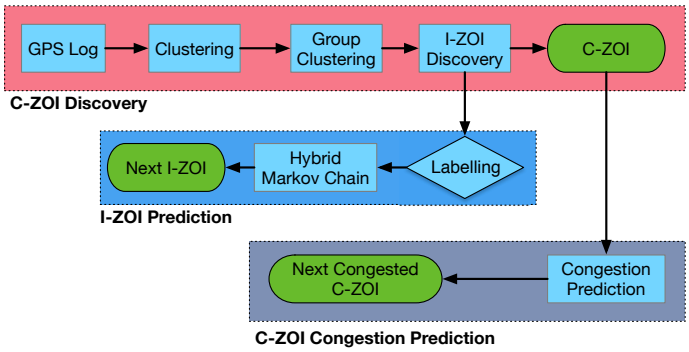


Fig. 1: Overview of the System Model.

A. Locations and Places

Mobile devices have the potential to track users' visited locations using the Global Positioning System (*GPS*). The device regularly collects a user's raw location logs as a list $L = [loc_1, loc_2, loc_3, ..., loc_n]$, where $loc_i = (\alpha, \beta, t, v)$ is a tuple representing a location point in the format $(latitude, longitude, timestamp, velocity)$. The rest of the paper uses the notations $loc.\alpha$, $loc.\beta$, $loc.t$ and $loc.v$ for the location point elements. In addition to GPS coordinates, users' daily activities consist of some places that they find useful or where they spend a considerable amount of time. This work

focuses on places, which we refer to Zone of Interest (*ZOI*) hereafter.

B. Common Zone of Interest (C-ZOI) Discovery

Intuitively, an Individual Zone of Interest (*I-ZOI*) is a cluster group that is frequently visited by a user and a Common Zone of Interest (*C-ZOI*) depicts a region covering multiple I-ZOIs. Conceptually, each C-ZOI represents a region, where several users are interested to visit frequently and spend considerable amount of time. In order to extract the C-ZOIs of users based on the location history L, we must first introduce the notations of cluster and cluster group. A cluster is a set of visited location points with similar temporal (*e.g., visiting time*) and spatial (*e.g., instantaneous velocity*) features. A cluster group is an aggregation of intersected clusters.

1) Cluster Discovery: A cluster represents a subset of successive location points in L, which are confined with similar temporal and spatial features. Δd_{max}, e_{max}, e_{min} and $v \in \mathbb{R}$ represents the distance expressed in meters, maximum velocity, minimum velocity and instantaneous velocity expressed in meters per second, respectively. In addition, $\Delta t_{min} \in \mathbb{N}$ is a time duration expressed in minutes. We introduce two functions: $ClusterCentroid\left([loc_1, loc_2, loc_3, ..., loc_n]\right)$, to compute the centroid of visited location points, and $Distance\left([loc_i, loc_j]\right)$, which measures the Euclidean distance between the two locations loc_i and loc_j. A subset $l \subseteq L$ becomes a cluster if the following conditions in Equation 1, 2, 3 and 4 are met:
$\forall loc_i, loc_{i+1} \in l$:

$$Distance\left(centroid\left(loc_1, ..., loc_i\right), loc_{i+1}\right) \leq \Delta d_{max} \quad (1)$$

$$loc_n t - loc_1 t \geq \Delta t_{min} \quad (2)$$

$$e_{min} \leq \sum_{i=1}^{n} \frac{v_i}{n} \leq e_{max} \quad (3)$$

$$\nexists l' \neq \quad : \quad l \subset l' \quad (4)$$

A cluster is a 4-item tuple $c = (\alpha_c, \beta_c, r, l)$, where α_c and $\beta_c \in \mathbb{R}$ are the latitude and longitude coordinates of the centroid, $r \in \mathbb{R}$ is it's radius in meters, and $l \in L$ is the subset of locations belonging to c. The average of all $loc.\alpha$ and $loc.\beta$ of the locations contained in the subset l is the centroid (α_c, β_c) of the cluster, which is designated as $c.centroid$. We introduce C, the set of clusters extracted from the location log of a user as $C = \{c_1, c_2, c_3, ..., c_n\}$. Disjointness of discovered clusters can not be guaranteed by equations 1, 2, 3 and 4. Therefore, construction of cluster group is required and explained in the next step.

2) Cluster Group Discovery: A cluster group includes a set of overlapped clusters. Thus, we define equation 5 to check whether two clusters c_i, $c_j \in C$ are intersected or not.

$$Distance\left(c_i.centroid, c_j.centroid\right) - \left(c_i r + c_j r\right) < 0 \quad (5)$$

A cluster group is a 4-item tuple $cg = (\alpha_{cg}, \beta_{cg}, r, \{c_1, c_2, c_3, ...\})$, where α_{cg}, β_{cg} and $r \in \mathbb{R}$, $\{c_1, c_2, c_3, ...\} \in C$ are latitude, longitude, radius and array of clusters constituting g respectively. $(\alpha_{cg}, \beta_{cg})$ represents the centroid of the cluster group, which is the mean of all the clusters formed g, and r must be compared to enclose all the individual clusters present in g. G contains the n cluster groups belonging to a user as $G = \{cg_1, cg_2, cg_3, ..., cg_n\}$

3) Individual Zone of Interest (I-ZOI): An I-ZOI refers to a frequently visited region by a user during daily activities. We define two constants $minCountThreshold$ and $maxTimeDifference \in \mathbb{N}$ representing the minimum threshold of visits and the maximum time difference threshold between two consecutive visits, respectively. Then, $CountVisits(cg)$ is a function that counts the number of clusters included in cluster group cg, and $timeDuration(G)$ is a function that returns the duration between two consecutive visited dates of cluster group cg in G. A cluster group $cg \in G$ is transformed into an I-ZOI z if the conditions of Equation 6 and 7 are met:

$$CountVisits(cg) \geq minCountThreshold \quad (6)$$

$$timeDifference(G) \leq maxTimeDifference \quad (7)$$

An I-ZOI z is a 6-item tuple $z = (\alpha_z, \beta_z, r, ID_{zone}, \{g_1, g_2, g_3, ..., g_n\}, T_{ID})$, where α_z, β_z and $r \in \mathbb{R}$, $ID_{zone} \in \mathbb{N}$ and $\{g_1, g_2, g_3, ..., g_n\} \in G$ are the latitude, longitude, radius, Zone-ID and group clusters becoming an I-ZOI. T_{ID} represents visiting dates of the I-ZOI by each user. The set Z is finally the set of I-ZOIs of the user such that $Z = \{z_1, z_2, z_3, ..., z_n\}$. As shown in Figure 2, the sets of discovered cluster groups are depicted by intersected yellow circles. Finally, the cluster groups that could satisfy above conditions are considered as I-ZOIs.

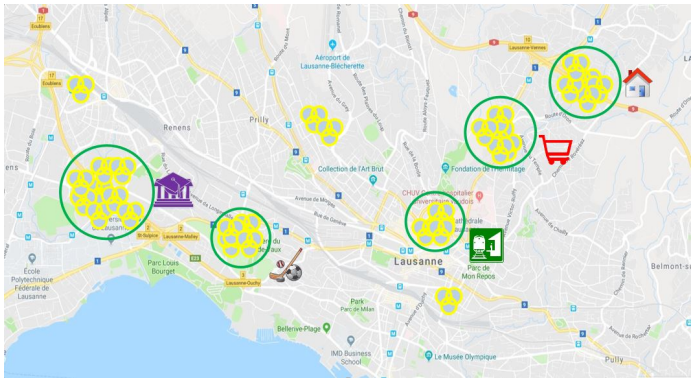


Fig. 2: I-ZOI construction from cluster groups of location points.

4) Common Zone of Interest (C-ZOI): A C-ZOI is an aggregation of adjacent I-ZOIs. We introduce a constant $distanceThreshold \in \mathbb{N}$ to represent the maximum threshold of distance between each I-ZOIs. $Distance\left(I - ZOI_i centroid, I - ZOI_j centroid\right)$ is a function to compute the Euclidean distance between the $I - ZOI_i$ and $I - ZOI_j$. I-ZOIs are grouped whenever the following

condition in Equation 8 is satisfied:

$$Distance\,(I-ZOI_i centroid, I-ZOI_j centroid) \leq distanceThresold \quad (8)$$

A C-ZOI is a 6-item tuple $C-ZOI = (\alpha_{cz}, \beta_{cz}, r, ID_{cz}, \{ZOI_1, ZOI_2, ZOI_3, ..., ZOI_n\}, T_{ID})$, where α_{cz}, β_{cz} and $r \in \mathbb{R}$, $ID_{cz} \in \mathbb{N}$ and $\{ZOI_1, ZOI_2, ZOI_3, ..., ZOI_n\} \in Z$ are the latitude, longitude, radius, C-ZOI-ID and group of I-ZOIs, respectively. The last item of the tuple indicates visiting dates of the C-ZOI by each user. Finally, we introduce CZ, which contains the n C-ZOIs belonging to users as $CA = \{C-ZOI_1, C-ZOI_2, C-ZOI_3, ..., C-ZOI_n\}$. Figure 3 shows an example of extracted C-ZOIs for a group of users. Each C-ZOI encapsulates a set of nearby I-ZOIs.

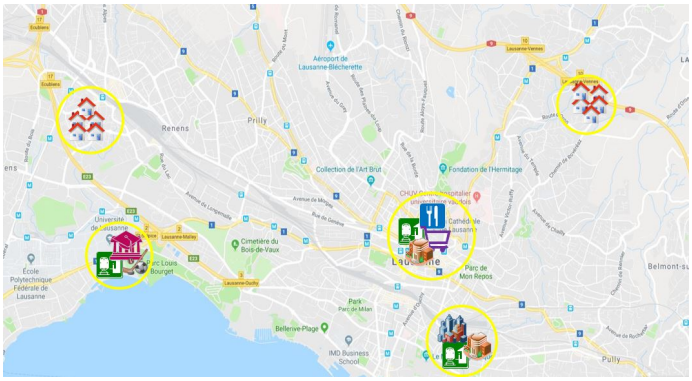


Fig. 3: C-ZOIs construction from I-ZOIs.

C. Individual Zone of Interest (I-ZOI) Prediction

This module predicts the users' future locations (*I-ZOI*). The mobility model represents the movement of mobile users and how their locations change over time. Collected mobility traces of users during their movements could be used to explore some sort of regularities in their daily life. This knowledge is utilized by a mobility predictor to forecast locations that a user may visit in future. The proposed mobility prediction scheme in this paper is based on a hybrid Markov chain model, which adaptively switch between the first-order or the second-order Markov chain, depending on the availability and quality of collected user traces. The proposed hybrid Markov model benefits from both the first-order Markov chain [25] and the second-order Markov chain. The rationale behind using a hybrid predictor is that the standard first-order Markov chain algorithms are memoryless models [26], which means that the mobility predictor only benefits from current temporal (*time and day of week*) and spatial (*location*) to predict next movement. However, the second-order Markov chain model benefits from previous state in addition to current state to predict future location. Actually, this information is really beneficial: if a user is currently at a city center, e.g., a restaurant, knowing whether he/she was at work or at home just before greatly helps in estimating his/her future behavior. However, we observe for some users trace data with discrete gaps (*ranging from a few seconds to a few minutes*). In these cases the 2-order state information conditions will not met, which led to poor performance for the second-order Markov predictor [26].

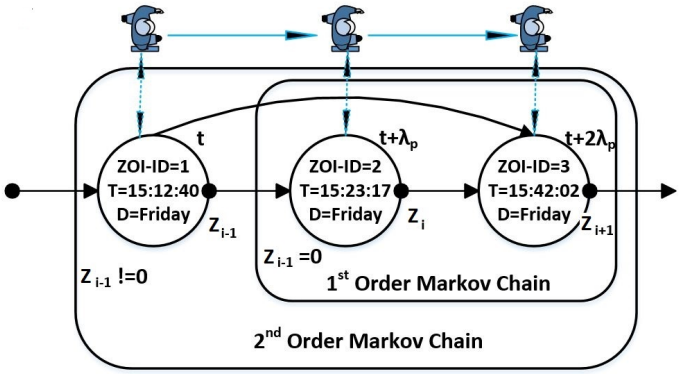


Fig. 4: Hybrid Markov Chain.

The proposed hybrid model is illustrated in Figure 4, in which a Markov chain state consists of a time step and an I-ZOI ID. Equation 9 defines the calculation of a future location probability, in which Z_i represents an I-ZOI with ID i, D indicates the day of the week (*e.g., Saturday*), T_i defines the time of the day D (*e.g., 13:22:43 h*), and λ_p determines the future time interval.

$$Pr(Z_{i+1}(t+2\lambda_p)) = \begin{cases} Pr(Z_{i+1}|Z(t+\lambda_p)=Z_i, & Z_{i-1}=0 \\ \quad D, T(t+\lambda_p)=T_i) & \\ Pr(Z_{i+1}|Z(t)=Z_{i-1}, Z(t+\lambda_p)=Z_i, & Z_{i-1}\neq 0 \\ \quad T(t)=T_{i-1}, T(t+\lambda_p)=T_i, D) & \end{cases} \quad (9)$$

Equation 9 can be considered as a location-dependent distribution and a time-dependent distribution (as expressed in Equations 10 and 13). As shown in Equation 13, both the time-dependent and location-dependent distributions in the second-order Markov chain model benefit from current and previous state information (*e.g., time, day and location*). The location dependent distribution can be modeled as a Mobility Markov Chain (*MMC*). The MMC is described by a transition matrix, which includes discovered I-ZOIs per each single user and all the calculated transitions between them. The mobility predictor obtains these transitions during training step by considering a tunable time threshold (*e.g., each minutes*) on the given days of the week (*e.g., all Wednesdays, all Thursdays and etc.*). The second-order Markov chain calculates transition probabilities among states only if two successive states are present in traced data.

For the case of the second-order Markov chain, the counting of transition frequency happens only when the user's movement is continuously following the sequence of two states.

$$Pr(Z_{i+1}|Z(t+\lambda_p)=Z_i, T(t+\lambda_p)=T_i, D) \quad (10)$$
$$=Pr(Z_{i+1}|Z(t+\lambda_p)=Z_i) \quad (11)$$
$$+Pr(T(t+\lambda_p)=T_i, D) \quad (12)$$

$$Pr(Z_{i+1}|Z(t)=Z_{i-1},Z(t+\lambda_p)=Z_i,T(t)=T_{i-1},T(t+\lambda_p)=T_i,D) \quad (13)$$

$$=Pr(Z_{i+1}|Z(t)=Z_{i-1},Z(t+\lambda_p)=Z_i) \quad (14)$$

$$+Pr(T(t)=T_{i-1},T(t+\lambda_p)=T_i,D) \quad (15)$$

D. Common Zone of Interest (C-ZOI) Congestion Prediction

From the probability distribution of the users' future visited I-ZOIs, we can further estimate the number of users that may visit and stay in a C-ZOI together at a future moment. Therefore, next, we target at predicting the probability distribution of the number of users that may visit together a specific C-ZOI within a given time. As explained in Section III, nearby I-ZOIs are covered by C-ZOIs. In these regions users either do not move or they move very slowly and users are spending a considerable amount of time together in each C-ZOI. Each of these hot spot regions are candidates to host a significant number of users (*pedestrians*). Then, we define the *AreaCongestionThreshold (M)*, which refers to the number of predicted users in each C-ZOI. The congestion prediction model counts the number of predicted users in each C-ZOI. If the number of users exceeds the defined threshold, we assume that this region will experience congestion within the next λ_c minutes. Predicting the number of users will help us to facilitate tasks such as resource management, logistic administration, and urban planning. For instance, if we know how many users will be in a specific C-ZOI between time t and $t+\lambda_c$ we could optimize placement of resources (*e.g., bandwidth allocation, public transportation*) in the city or dynamically adapt those resources, while taking into account the number of users. Equation 16 defines the probability of having M users visiting $C-ZOI_i$ at time $t+\lambda_c$, which is derived from the estimated number of upcoming and outgoing of users in $C-ZOI_i$ at time $t+\lambda_c$. The parameters used in this equation are listed in Table I.

$$\begin{aligned} P\left\{N_{C-ZOI_i}(t+\lambda_c)=M\right\} = \\ \sum_m P\left\{N_{C-ZOI_i}(t+\lambda_c)=M \mid N_{C-ZOI_i}(t)=m\right\} \times \\ P\left\{N_{C-ZOI_i}(t)=m\right\} = \\ \sum_m \langle \sum_{n_1,n_2,n_1-n_2=\triangle m} P\left\{N_{in,C-ZOI_i}(t+\lambda_c)=n_1\right\} \times \\ P\left\{N_{out,C-ZOI_i}(t+\lambda_c)=n_2\right\}\rangle \times P\left\{N_{C-ZOI_i}(t)=m\right\} \end{aligned} \quad (16)$$

In Equation 16, $P\left\{N_{in,C-ZOI_i}(t+\lambda_c)\right\}$ describes the probability of having n_1 users that may move into $C-ZOI_i$ at time $t+\lambda_c$ and $P\left\{N_{out,C-ZOI_i}(t+\lambda_c)\right\}$ indicates the probability of having n_2 users may moving out from $C-ZOI_i$ at time $t+\lambda_c$. These probabilities can be calculated using Equation 17.

$$\begin{aligned} P\left\{N_{in,C_i}(t+\lambda_c)=n_1\right\} = \sum_{A_1 \in F_{C_{i'}(t)}} \prod_{j_1 \in A_1} P_{j_1} \prod_{j_1' \in A_1^c} \left(1-P_{j_1'}\right) \times \\ P\left\{N_{out,C_i}(t+\lambda_c)=n_2\right\} \\ = \sum_{A_2 \in F_{C_i(t)}} \prod_{j_2 \in A_2} P_{j_2} \prod_{j_2' \in A_2^c} \left(1-P_{j_2'}\right) \end{aligned} \quad (17)$$

TABLE I: Area congestion prediction algorithm parameters.

Parameter Name	Parameter Definition
Z_i	State i in the Markov chain
$Pr\left\{Z_{i+1}(t)\right\}$	Probability of at State $(i+1)$ at t
$C-ZOI_i$, U_j	C-ZOI ID i, User ID j
D, T_i	Weekday and time of being at State i
λ_c, t	Future time interval and current time
$C=\{C-ZOI_1,..,i\}, \lvert C\rvert = I$	Set of C-ZOIs, I is the total numbers
$U=\{User_1,..,j\}, \lvert U\rvert = J$	Set of Users, J is the total numbers
$N_{C-ZOI_i}(t)$, $N_{C-ZOI_i}(t+\lambda_c)$	Number of Users in C-ZOI i at time t and t+λ_c (*e.g.,* m and M)
$N_{in,C-ZOI_i}(t+\lambda_c)$	Number of Users that may move to C-ZOI i at time t+λ_c (*e.g.,* n1)
$N_{out,C-ZOI_i}(t+\lambda_c)$	Number of Users that may move from C-ZOI i at time t+λ_c (*e.g.,* n2)
$F_{C-ZOI_i(t)}$	Subset of all users in $C-ZOI_i$ at time t
$F_{C-ZOI_{i'}(t)}$	Subset of all users out of $C-ZOI_i$ at time t
P_{j_1} , $User_{j_1} \in F_{C-ZOI_{i'}(t)}$	$P\{User_{j_1}$ *is in* $F_{C-ZOI_{i'}(t)}$ *at time* $t\}$ $\times$ $P\{User_{j_1}$ *moves to* $C-ZOI_i$ *at time* t+$\lambda_c\}$
P_{j_2} , $User_{j_2} \in F_{C-ZOI_i(t)}$	$P\{User_{j_2}$ *is in* $F_{C-ZOI_i(t)}$ *at time* $t\}$ $\times$ $P\{User_{j_2}$ *moves from* $C-ZOI_i$ *at time* t+$\lambda_c\}$

IV. Evaluation

In this section, we present an evaluation methodology to validate the proposed user mobility and area congestion prediction models.

1) Dataset: In order to train mobility and congestion prediction algorithms, accumulated traced data of users' movements is needed. In this research, we are relying on collected trace data in Nokia Mobile Data Challenge (*MDC*) dataset. [27]. This dataset contains records of almost 180 smartphones conducted by residents around the lake Geneva in Switzerland. The data records on which our work is based cover a duration over 17 months from October 2009 to March 2011. The basic demographic documents show that the participants are mostly young individuals and university students [27]. The dataset includes data generated from sensors and applications, such as visited locations (*GPS coordinates*), movement (*instantaneous velocity*), proximity (*Bluetooth*), communication (*Cell-IDs, WLAN-IDs*), etc. However, for mobility and congestion predictions, GPS coordinates of visited places and corresponding time stamps are required, which are nearly 10 million location points. To evaluate prediction performance of both mobility and congestion predictors, we divided collected mobility trace data of each single user to two parts: (i) dataset (L): which contains 70% of data as learning dataset. (ii) dataset (T): which contains the rest of traced data (30%) as testing dataset. Learning dataset is used to obtain states for both algorithms and to determine their transition probability matrix. The testing dataset is used to test and evaluate the accuracy of the proposed prediction algorithms.

2) User Trace Quality: Quality of collected traces depends to behavior of pedestrians. Some users keep the smartphone with them self everyday. However, others sometimes forgot to carry the devices or had to charge them, such that data recordings are non-continuous. As learned from our previous

experiences [28] [29], the number of valid states (*with a time stamp and I-ZOI ID*) in the drived hybrid Markov chain for each user depend on the quality of data trace in each day. Therefore, we first classify the dataset into two groups (*good or poor quality*) based on the number of recorded instances during the whole data collection period. We choose five users with good quality of trace data (*e.g., 500000-400000 records*) and five users with poor quality of trace data (*e.g., 250000-350000 records*).

3) Evaluation Metrics: Prediction accuracy measures the accuracy of the location prediction algorithm. We select states out of all the Markov chain states (*e.g., states from* 9 *AM to* 11 *AM*) derived for each particular weekday from the training dataset L for each user. Afterwards, the prediction algorithm is performed for each of the selected states to estimate the possible future visited I-ZOI(s) for mobility prediction in the next λ_p minutes. We check the transition probability for states during the same period of time in the testing data set T as well. Afterwards, the Mean Absolute Error (*MAE*) of the possible transitions of the corresponding testing points is calculated according to Equation 18. To evaluate performance of the area congestion predictor we define two metrics: (*i*) density of users, which counts the number of users that may move to each C-ZOI; (*ii*) area congestion prediction accuracy, which represents probability of moving users to a C-ZOI in a specific day of week and is calculated by average of future location prediction accuracies of users in each C-ZOI drived from Equation 18.

$$MAE = \frac{1}{N}\sum_{i=1}^{N} \mid Pr_iL - Pr_iT \mid, Accuracy = (1 - MAE) \times 100 \quad (18)$$

4) Experimental Settings: We describe the experimentation parameters of the discussed clustering, mobility prediction, and congestion prediction algorithms. In order to determine the parameters we analyze traced data for users with at least 10 months duration of collected data. Then, we read the data-points sequentially according to the recorded time stamps. Table II shows the experiment parameters and the associated values in our assessment.

TABLE II: Experiments parameters.

Parameter	Definition	Value
Δd_{max}	Maximum Euclidean distance between centroid of location points and next location	60 m
Δt_{min}	Minimum time threshold of staying in each location point	15 min
e_{max}	Maximum instantaneous velocity threshold	50 *m/s*
e_{min}	Minimum instantaneous velocity threshold	0
M	Number of predicted users in each C-ZOI	6
minCountTh	Minimum number of visits of each cluster group	60
maxTimeDiff	Maximum time difference between two consecutive visits of a cluster group by a user	24 *h*
DistanceTh	Maximum Euclidean distance between two I-ZOIs	500 *m*
λ_p	Time threshold for hybrid-MC algorithm	1 *min*
λ_c	Time threshold for area congestion prediction algorithm	15 *min*

V. Evaluation Results

1) Mobility Prediction Accuracy Results: This subsection details the prediction accuracy results of the proposed hybrid predictor, the first order and the second order Markov chain. We first present the average prediction accuracy of all the users with different trace qualities. Then, we discuss more details about the prediction accuracy per day, for users with poor and good trace qualities.

Figures 5 and 6 show the prediction accuracy of different MMC predictors for users with good and poor quality of

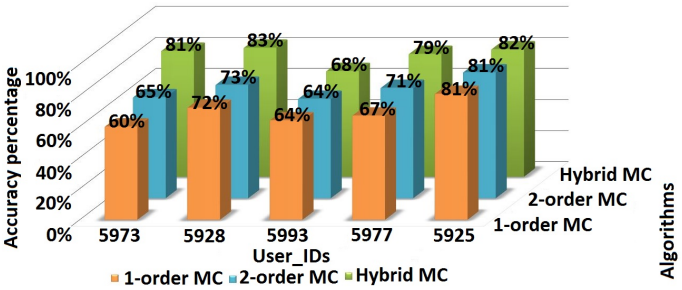


Fig. 5: Prediction accuracy for users with good quality.

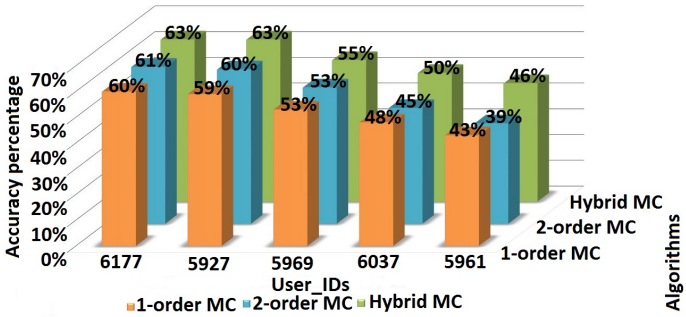


Fig. 6: Prediction accuracy for users with poor quality.

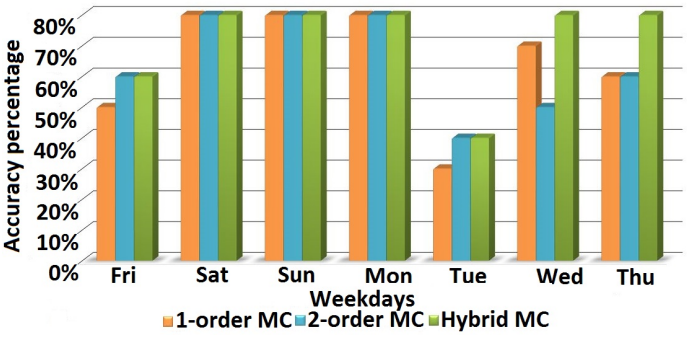


Fig. 7: Prediction accuracy per day for User-5928.

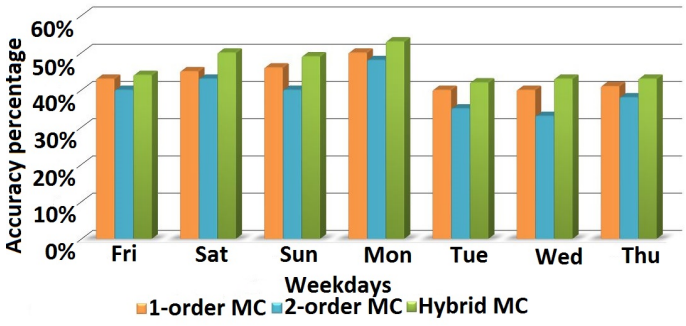


Fig. 8: Prediction accuracy per day for User-6037.

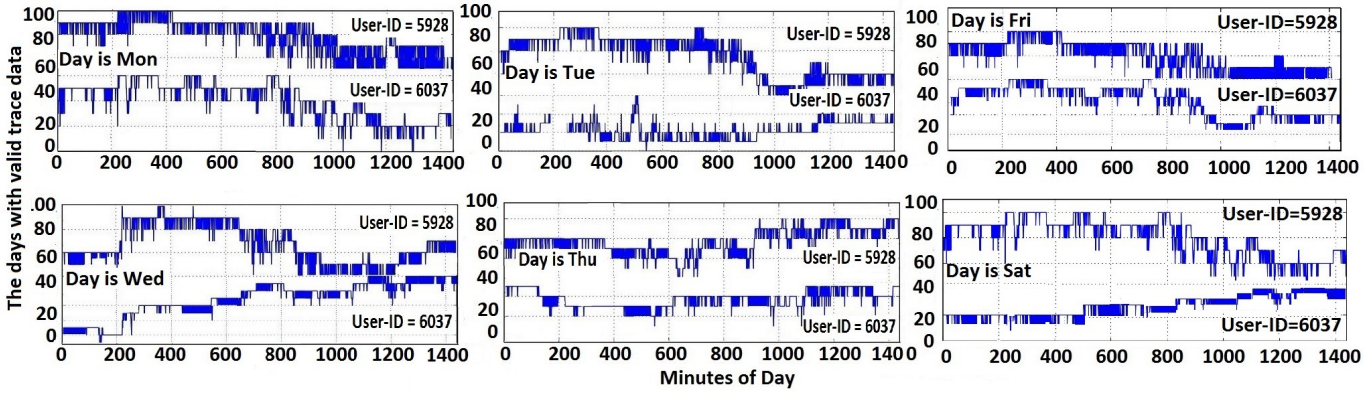


Fig. 9: Trace qualities of User_IDs 5928/6037 on weekdays.

mobility traces. We define two categories of quality depending on the number of instances recorded in a user's movement traces. We randomly choose 5 User IDs (5973, 5928, 5993, 5977, 5925) from the group of good quality trace data, and 5 User IDs (6177, 5927, 5969, 6037, 5961) from the group of poor quality trace data. As we can see from Figure 5, the hybrid predictor can deliver an average prediction accuracy over all weekdays and weekends of nearly 83% for User-ID 5928. Moreover, it can be observed from Figure 6 that the estimated accuracy is improved significantly when the hybrid predictor used for users with poor quality trace data. For instance, it delivers an average accuracy of 63% for User-IDs 6177 and 5927. The results clearly demonstrate that the hybrid predictor outperform others, while using the traced data with either poor or good quality. Figures 7 and 8 show the prediction accuracy of three different predictors for each day. This helps us to explain the advantages of the hybrid predictor compared to the first-order and second-order Markov chains. From defined user categories we randomly select User-ID 5928 and User-ID 6037 as the representatives of users with good and poor qualities. The graphs show that the hybrid predictor performs better than the first-order and the second-order MC predictors for both categories. To explain the performance difference of mobility predictors for these two users, we next discuss the data quality of User-ID 5928 and 6037. Figure 9 depicts mobility traces of users over a year, shown as a matrix, where each column is a minute of the day and each line indicates the number of days with valid trace data (*with time stamp and GPS coordinates*). We map each interval of valid records to continuous pulses, and leave blank intervals time during which we have no information about users' locations. To count the number of days with valid trace records, we introduce a *threshold*, which counts the days with more than 1500 records as valid days for prediction. If a user has less than 1500 records in one day, the data of that specific day will not be included in the prediction. This is because such a low number of records happen most probably because of network connectivity issues or defective sensors. Therefore, collected traces for these users are not valid and should not be included in the prediction procedure. Figure 6 illustrates that for User-ID 6037, the hybrid predictor can only deliver an accuracy of around 52% for Monday. This situation arises typically because location data are partly available. For User-ID 5928, due to having continuous intervals of collected GPS records at a high number of days with valid trace data between 80 to 100, the hybrid predictor has improved performance (81% *to* 83%) for all weekdays.

2) Congestion Prediction Accuracy Results: In addition to estimating future locations of mobile users, we are also interested in area congestion prediction. In this subsection, we present the prediction accuracy of the congestion prediction algorithm. Then we discuss more details about the number of predicted users in each C-ZOI.

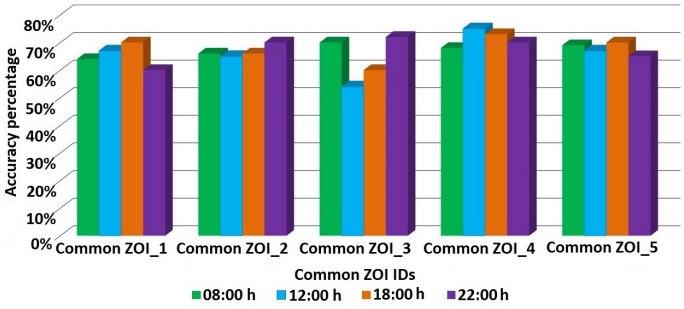


Fig. 10: Area congestion prediction accuracy.

We examine the predictability of congestion by employing recorded GPS coordinates of all available users in the dataset. We focus on the extracted C-ZOIs in the city of Lausanne by predicting the number of users that may move and stay together in each common hot spot. Figure 10 depicts the results of the congestion prediction algorithm for time-of-days (*08:00 h, 12:00 h, 18:00 h and 22:00 h*). The graph shows that the congestion prediction algorithm achieves accuracies exceeding 70% for C-ZOIs. In addition to congestion prediction accuracy, we also count the number of users in each C-ZOI for time-of-days (*08:00 h, 12:00 h, 18:00 h and 22:00 h*). Figure 11 shows the density of pedestrians for different hours. We observe that in the evening the users have tendency to travel towards the city suburbias (*C-ZOI 1, C-ZOI 5*), going back home for dinner. An inverse behavior is detected at 08:00 AM and 12:00 PM, when the most of flows are toward the universities or city center (*C-ZOI 2, C-ZOI 3 and C-ZOI 4*). Although we can not

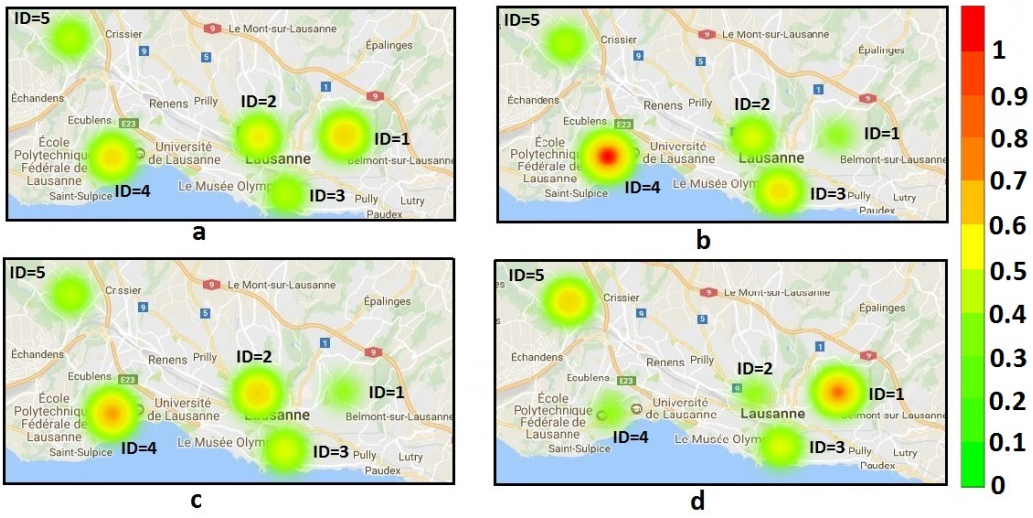


Fig. 11: Density of users in each C-ZOI. a) Wednesday at 08:00 h. b) Wednesday at 12:00 h. c) Wednesday at 18:00 h. d) Wednesday at 22:00 h.

compare these population densities against a proper ground truth, we remark that the model represents very reasonable results that match well to the movements of inhabitants in the city of Lausanne.

VI. CONCLUSIONS

With the explosive growth of location-based service on mobile devices, predicting users' future locations is of increasing importance to support proactive information services. In this paper, we introduce a hybrid predictor to estimate future locations of a user. Further, we propose a technique to discover hot spot regions for users by relying on spatial and temporal constraints. The achieved results over real world mobility traces validate our proposed algorithms, which achieve more than 81% correct predictions for users. More important, we present a novel approach to predict congestion in hot spot regions using GPS coordinates. This achieves accuracies exceeding 70% for discovered C-ZOIs from the available dataset.

For future enhancements we will concentrate on predicting trajectories of mobile users that they use for transition among I-ZOIs. We will also improve our area congestion prediction algorithm by applying congestion classification models to classify the predicted congestion to *slight congestion*, *moderate congestion* and *severe congestion*. Furthermore, to foster our hybrid predictor we are planning to employ a time series-based periodicity detection algorithm to recognize variations in pedestrians' behaviors, and apply the appropriate mobility predictor accordingly. We will also conduct extensive practical experiments to compare our mobility predictor with other Markov chain-based algorithms on other large scale datasets.

ACKNOWLEDGMENT

This work has been supported by the Swiss National Science Foundation with project number 154458.

REFERENCES

[1] G. Khodabandelou, V. Gauthier, M. A. El-Yacoubi, and M. Fiore, "Population estimation from mobile network traffic metadata," *CoRR*, vol. abs/1610.06947, 2016. [Online]. Available: http://arxiv.org/abs/1610.06947

[2] Z. Zhao, M. Karimzadeh, F. Gerber, and T. Braun, "Mobile crowd location prediction with hybrid features using ensemble learning," *Future Generation Computer Systems*, 2018. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0167739X17318058

[3] X. Chen, J. Pang, and R. Xue, "Constructing and comparing user mobility profiles," *ACM Trans. Web*, vol. 8, no. 4, pp. 21:1–21:25, Nov. 2014. [Online]. Available: http://doi.acm.org/10.1145/2637483

[4] V. Kulkarni, A. Moro, and B. Garbinato, "Mobidict: A mobility prediction system leveraging realtime location data streams," in *Proceedings of the 7th ACM SIGSPATIAL International Workshop on GeoStreaming*, ser. IWGS '16. New York, NY, USA: ACM, 2016, pp. 8:1–8:10. [Online]. Available: http://doi.acm.org/10.1145/3003421.3003424

[5] T. N. Maeda, K. Tsubouchi, and F. Toriumi, "Next place prediction in unfamiliar places considering contextual factors," in *Proceedings of the 25th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL'17. ACM, 2017, pp. 76:1–76:4.

[6] R. Montoliu and D. Gatica-Perez, "Discovering human places of interest from multimodal mobile phone data," in *Proceedings of the 9th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '10. ACM, 2010, pp. 12:1–12:10.

[7] "Identifying locations from geospatial trajectories," *J. Comput. Syst. Sci.*, vol. 82, no. 4, pp. 566–581, Jun. 2016. [Online]. Available: https://doi.org/10.1016/j.jcss.2015.10.005

[8] C. Zhou, D. Frankowski, P. Ludford, S. Shekhar, and L. Terveen, "Discovering personal gazetteers: An interactive clustering approach," in *Proceedings of the 12th Annual ACM International Workshop on Geographic Information Systems*, ser. GIS '04. New York, NY, USA: ACM, 2004, pp. 266–273. [Online]. Available: http://doi.acm.org/10.1145/1032222.1032261

[9] R. Hariharan and K. Toyama, "Project lachesis: Parsing and modeling location histories," in *Geographic Information Science*, M. J. Egenhofer, C. Freksa, and H. J. Miller, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004, pp. 106–124.

[10] J. J.-C. Ying, W.-C. Lee, T.-C. Weng, and V. S. Tseng, "Semantic trajectory mining for location prediction," in *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '11. New York, NY, USA: ACM, 2011, pp. 34–43. [Online]. Available: http://doi.acm.org/10.1145/2093973.2093980

[11] S. Gambs, M.-O. Killijian, and M. N. n. del Prado Cortez, "Next place prediction using mobility markov chains," in *Proceedings of the*

First Workshop on Measurement, Privacy, and Mobility, ser. MPM '12. New York, NY, USA: ACM, 2012, pp. 3:1–3:6. [Online]. Available: http://doi.acm.org/10.1145/2181196.2181199

[12] H. Aly and M. Youssef, "semmatch: Road semantics-based accurate map matching for challenging positioning data," in *Proceedings of the 23rd SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL '15. New York, NY, USA: ACM, 2015, pp. 5:1–5:10. [Online]. Available: http://doi.acm.org/10.1145/2820783.2820824

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

[14] X. Lu, E. Wetter, N. Bharti, A. J. Tatem, and L. Bengtsson, "Approaching the limit of predictability in human mobility," *Scientific Reports*, vol. 3, p. 2923, Oct. 2013.

[15] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini, and J. Widmer, "A survey of anticipatory mobile networking: Context-based classification, prediction methodologies, and optimization techniques," *IEEE Communications Surveys Tutorials*, vol. 19, no. 3, pp. 1790–1821, thirdquarter 2017.

[16] L. Song, D. Kotz, R. Jain, and X. He, "Evaluating next-cell predictors with extensive wi-fi mobility data," *IEEE Transactions on Mobile Computing*, vol. 5, no. 12, pp. 1633–1649, Dec 2006.

[17] Y. Qiao, Z. Si, Y. Zhang, F. B. Abdesslem, X. Zhang, and J. Yang, "A hybrid markov-based model for human mobility prediction," *Neurocomputing*, vol. 278, pp. 99–109, 2018. [Online]. Available: https://doi.org/10.1016/j.neucom.2017.05.101

[18] M. H. Sun and D. M. Blough, "Mobility prediction using future knowledge," in *Proceedings of the 10th ACM Symposium on Modeling, Analysis, and Simulation of Wireless and Mobile Systems*, ser. MSWiM '07. New York, NY, USA: ACM, 2007, pp. 235–239. [Online]. Available: http://doi.acm.org/10.1145/1298126.1298167

[19] M. Chen, X. Yu, and Y. Liu, "Mining moving patterns for predicting next location," *Information Systems*, vol. 54, pp. 156 – 168, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0306437915001295

[20] D. Barth, S. Bellahsene, and L. Kloul, "Combining local and global profiles for mobility prediction in lte femtocells," in *Proceedings of the 15th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '12. New York, NY, USA: ACM, 2012, pp. 333–342. [Online]. Available: http://doi.acm.org/10.1145/2387238.2387295

[21] S. Bellahsene and L. Kloul, "A new markov-based mobility prediction algorithm for mobile networks," in *Computer Performance Engineering*, A. Aldini, M. Bernardo, L. Bononi, and V. Cortellessa, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2010, pp. 37–50.

[22] L. Li, Y. Wang, G. Zhong, J. Zhang, and B. Ran, "Short-to-medium term passenger flow forecasting for metro stations using a hybrid model," *KSCE Journal of Civil Engineering*, Jul 2017. [Online]. Available: https://doi.org/10.1007/s12205-017-1016-9

[23] C. Chen, J. Hu, Q. Meng, and Y. Zhang, "Short-time traffic flow prediction with arima-garch model." in *Intelligent Vehicles Symposium*. IEEE, 2011, pp. 607–612. [Online]. Available: http://dblp.uni-trier.de/db/conf/ivs/ivs2011.html#ChenHMZ11

[24] P. S. Olszewski, "Modeling probability distribution of delay at signalized intersections," *Journal of advanced transportation*, vol. 28, no. 3, pp. 253–274, 1994.

[25] M. Karimzadeh Motallebi Azar, Z. Zhao, L. Hendriks, R. de Oliveira Schmidt, S. la Fleur, H. van den Berg, A. Pras, T. Braun, and M. Julian Corici, *Mobility and Bandwidth Prediction as a Service in Virtualized LTE Systems*. IEEE, 10 2015, pp. 132–138, 10.1109/CloudNet.2015.7335295.

[26] Z. Zhao, L. Guardalben, M. Karimzadeh, J. Silva, T. Braun, and S. Sargento, "Mobility prediction-assisted over-the-top edge prefetching for hierarchical vanets," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2018.

[27] I. Yaqoob, I. A. T. Hashem, A. Gani, S. Mokhtar, E. Ahmed, N. B. Anuar, and A. V. Vasilakos, "Big data," *Int. J. Inf. Manag.*, vol. 36, no. 6, pp. 1231–1247, Dec. 2016. [Online]. Available: https://doi.org/10.1016/j.ijinfomgt.2016.07.009

[28] B. Sousa, Z. Zhao, M. Karimzadeh Motallebi Azar, D. Palma, V. Fonseca, P. Simoes, T. Braun, H. van den Berg, A. Pras, and L. Cordeiro, *Enabling a Mobility Prediction-aware Follow-Me Cloud Model*. United States: IEEE Computer Society, 11 2016, pp. 486–494, eemcs-eprint-27394.

[29] Z. Zhao, M. Karimzadeh Motallebi Azar, T. Braun, A. Pras, and H. van den Berg, *Cloudified Mobility and Bandwidth Prediction in Virtualized LTE Networks*. IEEE, 7 2017.

Pedestrians Trajectory Prediction in Urban Environments

Mostafa Karimzadeh, Florian Gerber, Zhongliang Zhao, Torsten Braun
Institute of Computer Science, University of Bern, Switzerland
Email : {karimzadeh, zhao, braun}@inf.unibe.ch, florian.gerber@students.unibe.ch

***Abstract*—Increasing adoption of cellular phones equipped with global positioning system (*GPS*) chips enables the exploration of pedestrians' mobility patterns. Tasks such as discovering *hot-spots* in large cities can be addressed through the usage of accumulated *GPS* coordinates. In this work we utilize spatio-temporal analysis on collected geo-location points to discover Zone of Interests (*ZOIs*) of pedestrians in large cities to understand people's dynamics. We design an adaptive Markov model to forecast long distance trajectories of pedestrians, which adapts it's behavior constantly by switching from a first or second order Markov chain based on the quality of traced data and user's mobility patterns. From the predicted trajectories, we further introduce a mechanism to predict congested trajectories by estimating the number of pedestrians who may take the same trajectory in a future moment. We conduct comprehensive empirical experiments using a real-life dataset, namely the Mobile Data Challenge (*MDC*) dataset with 185 participants. Our mechanisms can deliver a satisfactory pedestrian trajectory prediction with a precision of 86% and a recall of 84%.**

***Index Terms*—Mobile analysis, Mobility and Congestion Prediction, Mobility Behavior, Location based Services.**

I. Introduction

Due to prevalence of location-based applications huge amount of mobility trace of pedestrians collected in urban cities. Such data encapsulates all visited locations and mimics the identity, behaviors, and interests of an individual or a group of users [1]. Therefore, analyzing collected data could reveal frequently visited places of users, formally called Zone of Interests (*ZOIs*). A *ZOI* is a region that an entity is interested to spend a significant duration of time and visits it frequently. Detecting such regions is an essential component for making cities smart, particularly with regards to traffic management, mobile network resource allocation, early congestion warning, etc.

User trajectory prediction has great research value and broad application prospects. For users, trajectory prediction can provide opportunities for better travel planning, such as informing drivers on the highway about the traffic condition beforehand. For service providers, trajectory prediction can help them to offer users personalized location-based services (*LBSs*) in real time and update user geographic information. In addition to trajectory prediction, congestion prediction in trajectories is an important precondition to alleviate traffic congestion in large-scale urban areas. Currently, the growth of location-aware technologies and location based Internet services enrich the variety of human mobility information. This knowledge has been employed to explore and predict pedestrian congestion in trajectories. In this paper, we introduce a spatio-temporal based clustering algorithm to extract users' *ZOIs*. The discovered *ZOIs* are used to train our mobility predictor to predict next visited *ZOI*. The implemented algorithm constantly chooses the first order or the second order Markov chain, based on the quality of mobility trace to predict future location of the users. The utilized mobility predictor has a tunable time threshold, which means according to our requirements the algorithm can determine next location of the users in scales of *minutes* or *hours*. More details regarding to the process of feeding the mobility predictor to estimate next location of users can be found in our previous works [2] [3]. We then extract observed trajectories among *ZOIs*, which will be used to train our trajectory predictor to predict a future trajectory that the user will take to move from a specific *ZOI* to another one. We propose an adaptive Markov chain-based model to predict future trajectories, which uses a periodicity detection algorithm to capture the trend of user's mobility and then adapts it's behavior constantly based on the user's movement to switch between the first-order or the second-order Markov chain. More important, we propose a trajectory congestion predictor to estimate the number of users in each trajectory at different time granularities. In this work, we use a real life dataset, namely, the Nokia Mobile Data Challenge (*MDC*) [4]. The dataset collected in Switzerland around lake Geneva from October 2009 to March 2011 by almost 185 volunteers. In a series of experiments to evaluate the prediction performance of proposed algorithms, the results show superior performance over other trajectory prediction in terms of precision and recall.

The rest of this paper is organized as follows. We briefly review the related work in Section II and provide an overview of our prediction framework in Section III. Sections IV and V discuss the methodology to evaluate the trajectory prediction model and demonstrate obtained results respectively. Finally, in Section VI we discuss about our conclusions and future work.

II. Related Work

The proliferation and ubiquity of collected *GPS* location points (*e.g., latitude and longitude*) have generated notable interest in the analysis of moving object's trace data to extract *hot-spots*. Clustering is one of the most popular data-mining methods, not only due to its exploratory power but also because it is the preprocessing step or subroutine for other techniques [5]. A clustering algorithm attempts to partition

m observed objects into *n* clusters, where each cluster is characterized with the similarity of objects within a cluster. The premier contribution in adopting the data clustering techniques for *hot-spot* detection was made by Ashbrook et al [6]. They propose an iterative approach to extract *hot-spots* by imposing a set of temporal constraints. *k*-shape and *k*-Multishape clustering techniques are proposed in [5]. The techniques relay on a scalable iterative refinement procedure.

Most of the introduced time-series mining algorithms, including clustering, critically rely on some temporal constraints (*e.g., distance measure*). However, according to our extensive experiments on the MDC dataset, using only temporal bounds can cause some uncertainties in regards to the detected *hot-spot* regions of users. Therefore, in this article, we define a novel algorithm, which uses both temporal and spatial metrics (*e.g., average velocity, instantaneous velocity*) to detect user's *ZOIs*.

Time granularity is an important aspect of future trajectory prediction, which can be classified into near future and distant future predictions. The majority of the published works up to now concentrated on near future prediction in the order of minutes (e.g.,10 *to* 15 *minutes*) by benefiting from the R-tree based indexing technique [7], [8]. However, our work is able to predict future distant trajectories with granularity of several hours. Tayeb et al. [9] uses PMR-Quadtrees [10] for predicting the future linear trajectories. This method assumes that objects move according to a linear function (*e.g., homogeneous movement*), which severely limits their applicability as in practice movements are more complex and individual objects may follow different motion patterns. In order to overcome this drawback, we utilize an adaptive Markov chain model to perform distant future trajectory prediction by using periodicity detection to detect movement patterns. Depending on the periodicity of movement the predictor selects the first-order or the second-order Markov chain adaptively.

For pedestrians, traveling through overcrowded trajectories is one of the major reasons of discomfort. Therefore, the problem of predicting the trajectories congestion has attracted a great amount of attention in recent years. Various congestion prediction methods have been implemented to help effective resource management in dense smart cities. Vinay et al designed *ARIMA* to forecast traffic congestion in urban areas [11]. Chuishi et al. [12] uses collected location data of smartphones to study crowdedness of mobile users. In [13] authors attempted to reveal urban crowding patterns using Automated Fare Collection (*AFC*) system data. They studied spatial and temporal patterns of crowds to predict congested routes. Pan et al. implemented a method to detect and predict traffic congestions using mobile users social network data and mobility patterns [14].

These techniques have some deficiencies for trajectory congestion prediction. First, the main drawback of these models is their inability to estimate the specific time of predicted congestion in trajectories, since these models are just working on spatial granularity (*location of congestion*), which means the algorithm outputs only the number of users in each specific trajectory without any timing information. Second, these works focus on near future trajectory prediction on the order of meters. Third, current methods like tree structures demand high complexity and memory usage. To overcome these shortcomings we propose a model, which improves prior methods with the ability to predict congestion in distant trajectories with estimating the time of congestion. Moreover, the propsed technique dose not demand costly actions as opposed to existing tree structures.

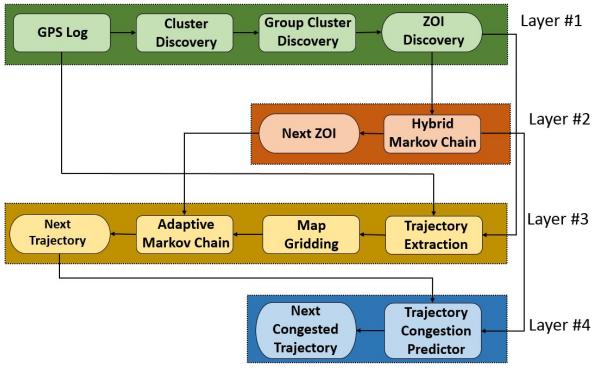


Fig. 1: Overview of the System Model.

III. System Model

Figure 1 shows the overall system architecture. The proposed system model involves four main layers: (1) Zone of Interest (*ZOI*) discovery, (2) Predicting next *ZOI*, (3) Predicting next trajectory among discovered *ZOIs* and (4) Trajectory congestion prediction. In the first layer of the system model, we utilize spatio-temporal analysis to detect *ZOIs* for each individual user. The discovered *ZOIs* are the input for the second layer, where our mobility predictor estimates future *ZOIs* for each user. The third layer relies on the outputs of layers 1 and 2 to predict trajectories among two *ZOIs*. Furthermore, layer 3 is using *GPS* logs to discover trajectories among two *ZOIs*. The discovered trajectories serve as training data for the trajectory predictor to exploit pedestrians' movement patterns. Finally, our trajectory congestion predictor in layer 4 utilizes the outputs of our mobility and trajectory predictor to estimate density of pedestrians in each trajectory. The relevant notations and system component definitions are explained in the following subsections.

A. Zone of Interest (ZOI)

Conceptually, a *ZOI* specifies a geographical area in which the user in question spends a significant amount of time (*e.g., home, workplace, gym*). We first define *GPS* trace data as a chronologically ordered list $L = [loc_1, loc_2, loc_3, ..., loc_n]$, where $loc_i = (lat, lng, t, v)$ is a tuple representing a location record specified by the latitude, longitude, time stamp and the velocity at the given point in time, respectively. Extracting *ZOIs* of a user based on the location trace *L* then is a multi-step process. To understand the process we must first introduce the terms cluster and group cluster. A cluster consists of a subset of successive location points in the trace data *L*, which are confined according to defined temporal and spatial constraints.

Intersecting clusters are aggregated as a group cluster. Group clusters, where the user spends a significant amount of time, are selected as a *ZOI*.

1) Cluster Discovery: For each single user, the trace data (L) is split into subsets $l_i \in L$ where $l_i = [loc_i, loc_{i+1}, \ldots, loc_{i+n}]$. We first define the time interval $\Delta t_{max} \in \mathbb{N}$ (*e.g.,* 10 *to* 20 *minutes*). Each subset l_i consists of a series of chronologically successive *GPS* locations that confine to $|loc_{i_t} - loc_{i+n_t}| \leq \Delta t_{max}$. We then define the parameter $r_{cluster} \in \mathbb{R}$ as a distance in meters (*e.g.,* 40 *to* 60 *meters*) denoting the maximum radius of a cluster. Furthermore, we define $min_{points} \in \mathbb{N}$ as the minimal number of locations points in a cluster (*e.g.,* 10 *to* 20 *location points*). Lastly, $v_{max} \in \mathbb{R}$ denotes the maximum average acceleration (*e.g.,* 4 *to* 8 *m/s*). A subset l_i is considered as a cluster if the following constraints are fulfilled:

$$distance(centroid(loc_1, ..., loc_i), loc_i + 1) \leq r_{cluster} \, \forall loc_i \in l_i \quad (1)$$

$$|l_i| \geq min\_points \quad (2)$$

$$\sum_{loc_i \in l_i} \frac{v_i}{|l_i|} \leq v_{max} \quad (3)$$

A cluster is then specified by a $4-item$ tuple c = (c_{lat}, c_{lng}, $r_{cluster}$, T_{c_i}). c_{lat},c_{lng} denote the latitude and longitude of the cluster. $r_{cluster}$ measured in meters denotes the maximum distance between the centroid and any given location in the subset l_i the cluster was discovered from. Lastly T_{c_i} is a list containing the time stamps for all location records in l_i. Since the clusters are discovered only for locations within a given time period Δt multiple clusters for a single user may geographically intersect, which leads to the next step of the group cluster discovery.

2) Group Cluster Discovery: Two clusters c_i and c_j belong to the same group cluster if Equation 4 is fulfilled:

$$distance(centroid(c_i), centroid(c_j)) \leq c_i.\Delta r + c_i.\Delta r \quad (4)$$

A group cluster (gc) is specified by a 5-tuple gc = (c_{lat}, c_{lng}, Δr , T_c). c_{lat} and c_{lng} denote the central position of the circular group cluster, which is given by the average c_{lat} and c_{lng} of the clusters belonging to this group cluster. Δr denotes the radius in meters. The radius is chosen to be minimum distance which still encloses all the clusters belonging to this group cluster. T_c is a set of time stamps given by equation 5.

$$T_{gc} = \bigcup_{c_i \in gc} T_{c_i} \quad (5)$$

$G = \{gc_1, gc_2, \ldots, gc_n\}$ is the set of all discovered group clusters for a given user.

3) ZOI Discovery: A Zone of Interest (*ZOI*) is denoted as z_i =(c_{lat}, c_{lng}, Δr, id), where the values of c_{lat}, c_{lng} and Δr remain the same as for the group cluster that is selected as a *ZOI*. The id is a unique *ZOI* identifier for a specific user. We consider a group cluster to be suitable as a *ZOI* if the number of time stamps (*cardinality of the set* T_c) is greater than a specific threshold $ts_{min} \in \mathbb{N}$ (900 *time stamps*). The discovered *ZOIs* are aggregated as the set Z = $\{z_1, z_2, \ldots z_n\}$, with n as the total number of *ZOIs* discovered for a specific user. As shown in Figure 2, the set of discovered *ZOIs* are depicted by intersected blue circles. Each blue circle represents extracted clusters.

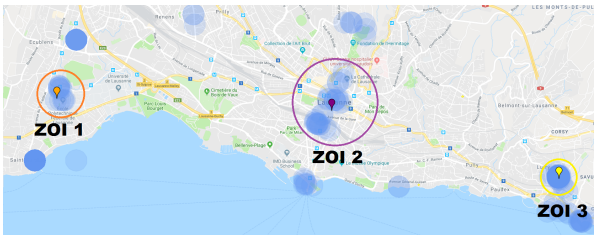


Fig. 2: Discovered ZOIs for a specific user

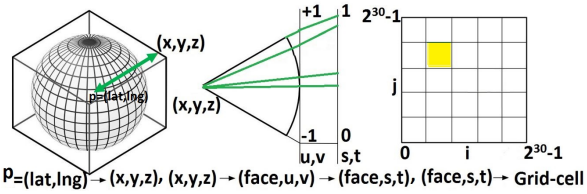


Fig. 3: Projecting a location point onto the grid-cell with Google S2 geometry library.

B. Trajectory Extraction

A trajectory is an observed path when a user travels from one *ZOI* to another one. A pedestrian can choose different paths to travel between his/her *ZOIs*. A list of all observed trajectories per each single user among ZOI_i to ZOI_j are accumulated in $Tr_{i,j} = [tr_1, tr_2, tr_3, .., tr_n]$. After discovering the paths, the next step is space partitioning. We use the *Python Google S2 geometry Library* [1] to partition the geographical space into grid-cells. Each grid-cell is a four corners cell, which covers a specific region on earth. As shown in Figure 3 the library hierarchically maps spatial location points of a sphere (*the earth in our case*) into grid-cells in several steps. First of all, the library surrounds the earth with a cube. Then, the location point p is projected onto faces of the cube to transform *GPS* coordinates of p to the three coordinates x, y and z. Since same area grid-cells on the cube have different sizes when mapped back to the sphere, a quadratic-transform is performed $i.e.$, $(face, u, v)$ is transformed to $(face, s, t)$ before partitioning the region into a grid-cell. Generated grid-cells are accumulated into a set of Grid-cells = { $gCell_1$, $gCell_2$,..., $gCell_m$}, which demonstrates distinct locations on earth. Each observed path tr_k is partitioned into a sub-list of l visited grid-cells such that $tr_k \in Tr_{i,j} : \{gCell_1, gCell_2, ..., gCell_l\}$. Given the *GPS* coordinates, the location of any moving object at a specific time can be mapped to a grid-cell. In this research, we generate cells with 1 square kilometer area, which means every 1 km^2 on earth's surface is represented by a rectangular grid-cell. Each grid-cell is uniquely identified by a 64-bit Cell-ID. Figure 4 illustrate an example of possible trajectories between two discovered *ZOIs*, which are represented by successive cells.

[1] http://s2geometry.io/

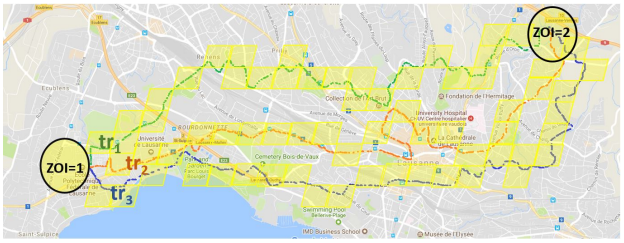


Fig. 4: Detected trajectories between two discovered ZOIs for a specific user.

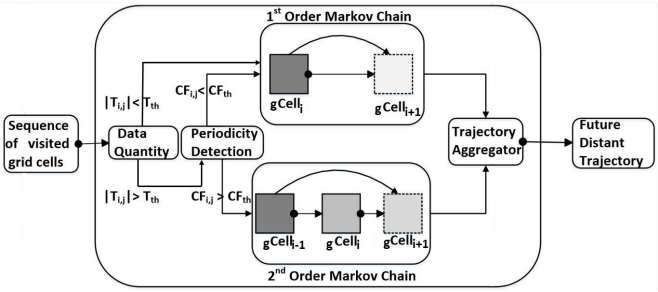


Fig. 5: Adaptive Markov chain

C. Trajectory Prediction

In this section, we introduce a novel trajectory predictor to estimate the future trajectory of pedestrians, which is defined as a series of grid-cells between two *ZOIs*. The proposed trajectory prediction scheme in this paper is based on the adaptive Markov chain, which integrates the first-order Markov chain and the second-order Markov chain with a decision process to select one of the available predictors [2]. The proposed model is able to adapt it's behavior according to the availability of paths (tr_k) in $T_{i,j}$ and the user's movement behavior among his/her *ZOIs* to maximize prediction performance, while reducing memory usage and execution time. The second order Markov chain has gained wide popularity in trajectory prediction tasks compared to the first order Markov chain because this predictor incorporates the current state and one state before to estimate next movement of users [2]. Actually, this information is really beneficial: some users have repetitive transitions, they take certain trajectories to travel between *ZOIs (e.g., home and workplace)*, and therefore, learning and predicting their patterns is easy. On the other hand, some other users travel at random and choose different trajectories to reach *ZOIs*. It is tough to predict their next trajectories, but knowing previous states of these users greatly helps in estimating their future movements. However, the second order Markov chain has a high complexity. Memory demand of the second order Markov chain is $O(N^3)$ and size of generated transition probability matrix is $N^2(N-1)$ [15]. Unlike the second order Markov chain the first order Markov chain is slightly different, as only the current state is taken into account to estimate future movements of users. Therefore, it has lower complexity, memory demand of the first order Markov chain is $O(N^2)$ and the size of the generated transition probability matrix is $N(N-1)$. In the *MDC* dataset we observed some users' trace data with discrete gaps (*ranging from a few seconds to a few minutes*). In these cases the second order state information conditions were not be met, because the second order Markov chain calculates transition probabilities among states only if two successive states are present in $Tr_{i,j}$. Therefore, it leads to poor performance for the second order Markov predictor. Based on these observations, we add a decision process to the trajectory predictor to choose either the first order or the second order Markov chain according to the availability of data and type of movements. Details about this multi-step decision process are explained in sections III-C1 and III-C2.

The proposed adaptive model is illustrated in Figure 5, in which the sequence of visited grid-cells between ZOI_i and ZOI_j is the input for the model. Data quantity and periodicity detection units are part of the decision process. The first order and the second order Markov based algorithms are available predictors in the adaptive Markov chain. Finally, the trajectory aggregator stores sequence of consecutive predicted grid-cells to form the next trajectory. Equation 6 defines the calculation of a next trajectory probability, in which $gCell_i$ represents a grid-cell with ID i, $CF_{i,j}$ is the calculated confidence factor of a trajectory among ZOI_i and ZOI_j, CF_{th} is the confidence factor threshold, $|Tr_{i,j}|$ indicates the number of observed paths between two ZOI i and j, and T_{th} is the threshold of observed paths between to ZOI_i and ZOI_j. These parameters and thresholds are explained in the next sections.

Equation 6 is an integration of the first and the second order Markov chains (*as expressed in Equations 7 and 8*). As shown in Equation 7, to estimate the next grid-cell, the first order Markov chain only benefits from current state information (*grid Cell-IDs*), where the second order Markov chain benefits from both current and previous grid-cells (Equation 8).

$$Pr(cell_{i+1}) = \begin{cases} Pr(cell_{i+1}|cell_i) & \text{if } CF_{i,j} > CF_{th} \vee |Tr_{i,j}| < T_{th} \\ Pr(cell_{i+1}|cell_i, cell_{i-1}) & \text{otherwise} \end{cases} \tag{6}$$

$$Pr(cell_i|cell_{i-1}) = \frac{\sum\limits_{tr \in Tr_{i,j}} (tr_i = cell_i \wedge tr_{i-1} = cell_{i-1})}{\sum\limits_{tr \in Tr_{i,j}} (tr_{i-1} = cell_{i-1})} \tag{7}$$

$$Pr(cell_i|cell_{i-1}, cell_{i-2}) = \frac{\sum\limits_{tr \in Tr_{i,j}} (tr_i = cell_i \wedge tr_{i-1} = cell_{i-1} \wedge tr_{i-2} = cell_{i-2})}{\sum\limits_{tr \in Tr_{i,j}} (tr_{i-1} = cell_{i-1} \wedge tr_{i-2} = cell_{i-2})} \tag{8}$$

1) Data Quantity: The number of collected valid records (*with GPS coordinates and timestamp*) in traces depends on the behavior of users. Some users keep smartphones with themselves everyday. However, others sometimes forgot to carry the devices or had to charge them, such that data recordings are non-continuous. As learned from our previous experiences [2] [16], the utilization of two state information by the second order Markov chain could increase prediction performance, when current and previous visited states information are available. However, when trace data includes

discrete gaps, the second order state transition conditions will not be met, which leads to poor performance for the second order predictor. Based on these experiences, data quantity analysis serves as the first decision step in future trajectory prediction. We use the cardinality of $Tr_{i,j}$ ($|Tr_{i,j}|$) as a metric to measure quantity of collected traced data. When the number of observed paths between two *ZOIs* i and j in $Tr_{i,j}$ during collecting the dataset for a given user is below a certain threshold (Tr_{th} *=150 paths*), then the adaptive Markov chain always utilizes the first order Markov chain for trajectory prediction. Otherwise, the decision process is passed onto the periodicity detection algorithm, which is presented in the next section.

2) Periodicity Detection: In this section, we present a periodicity detection algorithm to efficiently classify the users movement between two *ZOIs* as either *homogeneous* or *heterogeneous*. The periodicity detection algorithm that we use in this work is a modification of the segment periodicity detection algorithm presented in [17], where the authors defined a time series to be periodic with a period P_T, if the time series can be divided into equal-length segments, each of length P_T, that are almost identical. In our application we define C to be a series of grid-cells, given by combining the trajectories in $T_{i,j}$. An example of such a series would be $C = (1, 2, 3, 4, 1, 2, 3, 4, 1, 2, 3, 4)$, where the integer numbers indicate the unique ID for each grid-cell. To formalize the periodicity detection we define C_i as the time series C shifted by i positions to the right. In this case, leaving the first i entries undefined (*indicated by the '*' character*) and discarding the last i entries (e.g. $C_4 = (*, *, *, *, 1, 2, 3, 4, 1, 2, 3, 4)$). To compare the similarity of two sequences we utilize the Hamming distance [18] as defined by Equation 9.

$$H(C, C_{P_T}) = \sum_{j=0}^{n-1} \begin{cases} 1 & \text{if } C_j = C_{P_{T_j}} \\ 0 & \text{if } C_j \neq C_{P_{T_j}} \end{cases} \tag{9}$$

Intuitively, H measures the number of identical grid-cells at the same position in the original and the shifted time series. Furthermore, we define the confidence factor $CF_{i,j} \in [0, 1]$ for a given period P_T and a time series C according to Equation 10.

$$CF_{i,j} = \max_{CF_{i,j}} : \frac{H(C, C)}{|C| - P_T} \geq CF_{i,j} \tag{10}$$

The confidence factor measures the number of matching symbols at the same position, between the original time series C and the shifted time series C_{P_T}, in comparison to the number of possible matches. In their work [17], the authors describe an algorithm to find P_T with the highest $CF_{i,j}$ across all possible periods, with a time complexity of $O(nlogn)$. In our case, it is apparent that the periods with the highest confidence factors need to be reasonably close to the average length of the transitions in $Tr_{i,j}$. Therefore, we only consider the period lengths $P_T \in [1, \lambda$ * avg length $Tr_{i,j}]$, with λ (*e.g., 1.1 to 1.2*) as a fixed parameter to ensure that period lengths slightly above the average length of the transitions in $Tr_{i,j}$ are considered as well. Our modified periodicity algorithm runs in $O(n)$ as specified by Algorithm 1. The algorithm outputs the period $P_T \in [1, \lambda$ * avg length $Tr_{i,j}]$ with the highest confidence factor and the corresponding confidence factor itself. If $CF_{i,j} \geq CF_{th}$ with $CF_{i,j}$ being the confidence

Algorithm 1: Periodicity detection algorithm to find highest confidence factor $CF_{i,j}$ and corresponding period P_T

find $CF_{i,j}$ $(Tr_{i,j}, \lambda)$;
Input : Set of Trajectories $T_{i,j}$, λ coefficient
Output: max. confidence factor and corresponding period P_T
Define C = $[tr_1, \ldots tr_{n-1}]$ $tr_i \in Tr_{i,j}$;
Define x-dimensional Vector X, x = λ * `avg length` $T_{i,j}$;
for $i \in [0, \lambda * \texttt{avg length}\ Tr_{i,j}]$ **do**
  $X_i = H(C, C_i)$
end
CF = 1 ;
while *True* **do**
  for $P_T \in [0, \lambda * \texttt{avg length}\ Tr_{i,j}]$ **do**
    if $\frac{X_i}{|Tr_{i,j}| - P_T} \geq CF_{i,j}$ **then**
      return $(CF_{i,j}, P_T)$;
    else
      reduce $CF_{i,j}$
    end
  end
end

factor returned by the algorithm and CF_{th} being a previously defined threshold (*e.g., 0.25 to 0.35*) the movement is classified as homogeneous. Therefore, the first order Markov chain is selected, otherwise the second order Markov chain is selected. Figure 6 shows the period lengths and the corresponding confidence factors for different users. For example for user 5993 and the transitions between *ZOIs* 2 and 3, the period P_T with the highest confidence factor is P_T = 8 and the corresponding confidence factor $CF_{i,j}$ = 0.28. Considering CF_{th} = 0.25, determined through our experiments, the first order Markov chain is used in this case, because the movement behavior of the user between the two *ZOIs* is classified to be homogeneous.

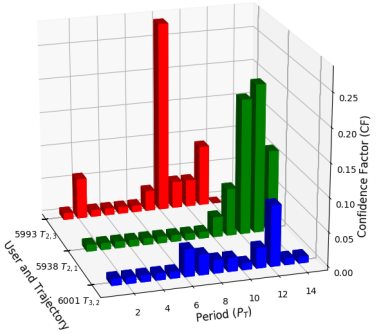


Fig. 6: Confidence Factor for different periods and trajectories

D. Trajectory Congestion Prediction

In this subsection we illustrate how mobility and trajectory predictors can be combined to predict congestion in trajectories. The key aspect is to predict the future trajectories between predicted future *ZOIs* for multiple users during the same time frame. By storing and aggregating the predicted trajectories for multiple users we can determine from which grid-cells users pass most often. Consequently, some means of storing

the trajectories of multiple users needs to be introduced. We use an inverted index which is made up of n different entries, n denotes the number of unique grid-cells in all the predicted trajectories that are currently stored. Each entry is indexed by a specific, unique grid $Cell - ID$ which occurs in at least one predicted trajectory. The entries contain at least one and up to m $(4 - item)$ tuples (user-id, $p_{i,i+1}$,t_{start}, t_{end}). The $User - ID$ is the unique identifier of a user and $p_{i,i+1}$ $\in [0, 1]$ is the probability of moving to the next *ZOI*. t_{start}, t_{end} denotes the time the user leaves the current ZOI_i and the time the user will arrive at the predicted ZOI_{i+1}. The system as seen in Fig 7 integrates the adaptive Markov chain and the hybrid Markov chain [3] with the inverted index. The trajectory congestion prediction is a multi-step process:

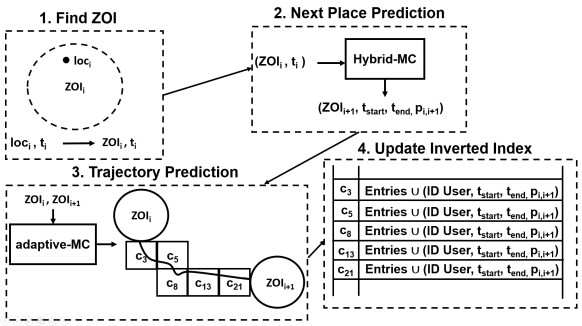


Fig. 7: Trajectory congestion predictor

1) Current location record of the user is received. Using the latitude and longitude of the location record, the system checks whether the user is currently in a *ZOI*. This step hands the id of the ZOI_i where the user is in.
2) The current ZOI_i and the current time stamp (t_{start}), which are part of the location record, are passed to the hybrid Markov chain. The algorithm predicts the next ZOI_{i+1}, as well as t_{end}, which is a time stamp, and indicate when the user will arrive to ZOI_{i+1}. Lastly, $p_{i,i+1} \in [0, 1]$ is returned, which gives us the probability of this next place prediction. From the outputs of the hybrid Markov chain the tuple that will be stored in the inverted index can already by formed. The tuple is (user-id, $p_{i,i+1}$,t_{start}, t_{end}), where the first item is the user-id and the other items can be directly assumed from the hybrid Markov chain output.
3) The next step is to predict the next trajectory for the user. Using the current ZOI_i from step 1 and the predicted ZOI_{i+1} from step 2, the trajectory between these two places can be predicted using the adaptive Markov chain. The output of the algorithm is an ordered list of grid-cells which the user will pass through when moving from ZOI_i to ZOI_{i+1}. These grid-cells correspond to the indexes in the inverted index where we will store the tuple formed in step 2.
4) At last, the inverted index is updated. First, old tuples for the user in question are removed from the inverted index as they hold deprecated information. Then for each grid-cell in the predicted trajectory from step 3 the tuple (user-id, $p_{i,i+1}$,t_{start}, t_{end}) is added to the corresponding entry in the inverted index.

These 4 steps are repeated for each user every time a new location record is received. The inverted index is organized to contain one entry for each grid-cell. For each user that is predicted to pass through a given grid-cell an additional tuple is added to the corresponding entry. Therefore, by monitoring the number of tuples of each entry we can deduct how many users will pass through the respective grid-cell. Considering t_{start} and t_{end} stored in the tuples we can make a prediction about number of users who may move through certain grid-cells at the next time threshold λ_c (*e.g., next* 10 *or* 30 *minutes*).

IV. Evaluation

In this section, we discuss an evaluation methodology to validate the proposed trajectory and congestion predictors.

A. Dataset

To evaluate the prediction performance of our approach, we are relying on the Mobile Data Challenge (*MDC*) dataset [4], which contains large-scale records conducted by 185 participants from the city of Lausanne in Switzerland. The collected data on which our research is based spans a period over 14 months. Since for some pairs of *ZOIs* only a small number of transitions were recorded, a *10-fold cross-validation* method was chosen in order to maximize the training data available to our trajectory predictor. The available transitions between two *ZOIs* were divided into 10 parts with an equal number of transitions. Then 9 parts were used for training the trajectory predictor while the last part was used for the testing of the algorithm. The results were subsequently averaged over all possible combinations of the different parts as training and testing data.

B. Evaluation Metrics

To interpret the success of the proposed adaptive Markov chain we used metrics which are commonly accepted in *information retrieval* [19]: *precision* and *recall*. These metrics indicate the relationship between the numbers of *True Positive* (11), *False Positive* (12), and *False Negative* (13).

$$TP = gCell_i \in Tp_{i,j} \wedge gCell_i \in Tr_{i,j} \tag{11}$$

$$FP = gCell_i \in Tp_{i,j} \wedge gCell_i \notin Tr_{i,j} \tag{12}$$

$$FN = gCell_i \notin Tp_{i,j} \wedge gCell_i \in Tr_{i,j} \tag{13}$$

$Tp_{i,j}$, $Tr_{i,j}$ are trajectories predicted by adaptive Markov chain and the observed trajectory, which is extracted according to a user's movement history among ZOI_i and ZOI_j, respectively. Besides, $gCell_i$ is a grid-cell such that $gCell_i \in$: $\{gCell_1, gCell_2, gCell_3, ..., gCell_m\}$ $i = 0, 1, 2, 3, ..., m$. Using these values *precision* and *recall* can be transformed to our domain and defined as follows:

- **Precision:** The part of the predicted trajectory ($Tp_{i,j}$) that truly belongs to the observed trajectory ($Tr_{i,j}$).
- **Recall:** The part of the observed trajectory ($Tr_{i,j}$) that is correctly estimated.

As we explain in section III-B, the observed trajectory ($Tr_{i,j}$) includes several discretized transitions (tr_k), which show different paths among ZOI_i and ZOI_j. To measure average *precision* and *recall* for each set of observed trajectories we are defining functions $AvgPrecision(Tp_{i,j}, Tr_{i,j})$ and $AvgRecal(Tp_{i,j}, Tr_{i,j})$, in Equations 14 and 15, respectively.

$$AvgPrecision\,(Tp_{i,j}, Tr_{i,j}) = \frac{\sum_{tr_k \in Tr_{i,j}} TP\,(Tp_{i,j}, tr_k)}{\sum_{tr_k \in Tr_{i,j}} (TP\,(Tp_{i,j}, tr_k) + FP\,(Tp_{i,j}, tr_k))} \quad (14)$$

$$AvgRecall\,(Tp_{i,j}, Tr_k) = \frac{\sum_{tr_k \in Tr_{i,j}} TP\,(Tp_{i,j}, tr_k)}{\sum_{tr_k \in Tr_{i,j}} (TP\,(Tp_{i,j}, tr_k) + FN\,(Tp_{i,j}, tr_k))} \quad (15)$$

C. Experimental settings

The parameters used to form the system model's components (*clustering, trajectory prediction and trajectory congestion prediction*) and their associated values are shown in Table I. These values are determined by analyzing trends of pedestrians with at least 10 months collected mobility trace in the *MDC* dataset.

TABLE I: Experiments parameters.

Parameter	Definition	Value
t_{max}	Time window for cluster discovery	10 *min*
Δr_{max}	Maximal radius of a cluster	60 *m*
e_{max}	Maximal velocity	8 ms^{-1}
$minpoints$	Minimal number of location records per cluster	10
ts_{min}	Minimal time stamps per ZOI	200
T_{th}	Threshold for observed paths between two ZOI_i and ZOI_j	150
CF_{th}	Threshold for confidence factor of observed trajectory between two ZOI_i and ZOI_j	0.25
λ_c	Time threshold for trajectory congestion prediction algorithm	20 *min*

V. Evaluation Results

A. Trajectory Prediction Results

In this subsection, we examine the trajectory prediction performance of the proposed adaptive Markov chain. We conducted detailed experiments to compare the performance of the proposed predictor to various trajectory estimation methods [20]. Each of these estimation methods generate a non-sequential set of grid-cells to represent future trajectories between two *ZOIs*. The set of selected grid-cells consists of all grid-cells, which occur more often than a certain threshold in the training data. The three methods of *adaptive threshold, F-score optimization threshold and mean threshold* generate different thresholds.

Through the experiments, we were able to obtain the average prediction precision and average prediction recall. We choose 4 User IDs (5927, 5938, 5976, 5993) and one source *ZOI* and destination *ZOI* for each user. Figures 8 and 9 present the average precision and average recall results for the different users and combination of *ZOIs*. Figure 8 shows the better performance of the adaptive Markov chain in terms of precision when compared to the other methods. The adaptive Markov chain achieved the highest average precision of 86%. Figure 9 shows that no predictor achieved the highest average recall over all test cases.

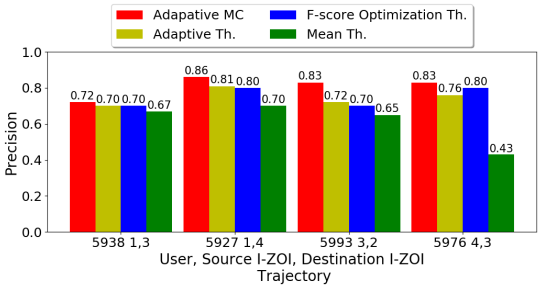


Fig. 8: Prediction precision

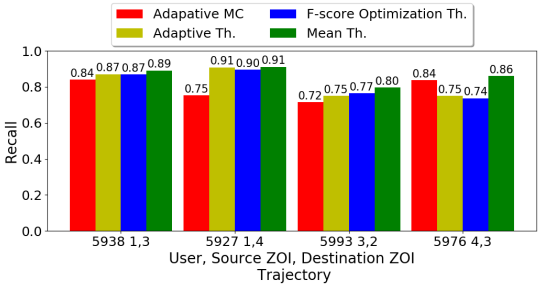


Fig. 9: Prediction recall

B. Trajectory Congestion Prediction Results

In addition to predicting next trajectories of pedestrians, we are also interested to predict the number of users that may take the same trajectory to travel among two *ZOIs*. When the density of predicted users in a trajectory is more than other trajectories we can assume that this trajectory will be congested. Figure 10 displays the predicted density of users for different time interval of a day. We found that in the early morning $(08:00\ \ to\ \ 08:20)$ the pedestrians have a tendency to take trajectories that are mostly in the city center, going to transportation hubs. We observed that in the morning to afternoon $(10:00\ \ to\ \ 10:20, 15:40\ \ to\ \ 16:00\ \ and\ \ 17:40\ \ to\ \ 18:00)$ most of the pedestrians' flows are around universities. This is because, most of the participants during collecting the *MDC* dataset were associated with the Universities of Lausanne and EPFL, therefore, predicting congestion in trajectories around those universities can be expected. Although we can not compare these population densities against a proper ground truth, we remark that the model represents very reasonable results that match well to the movement of inhabitants in the city of Lausanne.

VI. Conclusions

A huge volume of geo-location points are being ubiquitously accumulated as pedestrians' traced data. Extracting

Fig. 10: Congestion in Trajectories at different times of day

meaningful information from collected data to feed predictors (*e.g., mobility predictor, trajectory predictor, trajectory congestion predictor*) is indeed at the heart of many *LBSs*, such as traffic congestion forecasting, public-transportation optimization, etc. Through this work, we address three key challenges associated to collected datasets: (i) We propose a technique to discover *ZOIs* for pedestrians by relying on spatial and temporal analysis. (ii) Our extensive experiments on the *MDC* dataset show that the trajectories taken by pedestrians are often complex and as such, the pedestrians' movement types should be considered to predict future trajectories. Therefore, we propose an adaptive Markov chain as a trajectory predictor, which can constantly adapts it's behavior according to regularities of pedestrians. (iii) To estimate number of users in trajectories we introduce a novel approach based on inverted indexing, which is well suited in the context of large scale datasets. To examine our algorithms we conducted comprehensive experiments using a real-life dataset, namely the Mobile Data Challenge (*MDC*) dataset. We found satisfactory pedestrians future trajectory prediction precision of 86% and recall of 84% for the available users.

We observed that certain families of algorithms are more suited for particular mobility behavior. Our future work will be an attempt to have an ensemble trajectory predictor to select a suitable predictor according to behavioral changes, to attain higher prediction performance.

REFERENCES

[1] G. Joachim, M. v. Kreveld, and S. Frank, "Algorithms for hotspot computation on trajectory data," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL'13, 2013.

[2] Z. Zhao, L. Guardalben, M. Karimzadeh, J. Silva, T. Braun, and S. Sargento, "Mobility prediction-assisted over-the-top edge prefetching for hierarchical vanets," *IEEE Journal on Selected Areas in Communications*, pp. 1–1, 2018.

[3] M. Karimzadeh, Z. Zhao, F. Gerber, and T. Braun, "Pedestrians complex behavior understanding and prediction with hybrid markov chain," in *2018 Eleventh International Workshop on Selected Topics in Mobile and Wireless Computing (STWiMob'2018)*, Limassol, Cyprus, Oct. 2018.

[4] J. K. Laurila, D. Gatica-Perez, I. Aad, B. J., O. Bornet, T.-M.-T. Do, O. Dousse, J. Eberle, and M. Miettinen, "The mobile data challenge: Big data for mobile computing research," 2012.

[5] P. John and G. Luis, "Fast and accurate time-series clustering," *ACM Trans. Database Syst.*, vol. 42, no. 2, Jun. 2017.

[6] D. Ashbrook and T. Starner, "Learning significant locations and predicting user movement with gps," in *Proceedings. Sixth International Symposium on Wearable Computers,*, 2002, pp. 101–108.

[7] T. Alasdair, G. Nathan, and S. Victor, "Predicting interactions and contexts with context trees," in *Proceedings of the 24th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. GIS '16, 2016.

[8] R. Vineeth, J. Niranjan, K. Alexander, and R. C. K., "Probabilistic social sequential model for tour recommendation," in *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, ser. WSDM '17, 2017.

[9] T. Jamel, U. zgr, and W. Ouri, "A quadtree-based dynamic attribute indexing method," vol. 41, pp. 185–200, 01 1998.

[10] X. Mengwen, W. Dong, and L. Jian, "Destpre: A data-driven approach to destination prediction for taxi rides," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16, 2016.

[11] G. V. B., G. Gagan, G. Ajay, and A. Rajeev, "Assessment of arima-based prediction techniques for road-traffic volume," in *Proceedings of the Fifth International Conference on Management of Emergent Digital EcoSystems*, ser. MEDES '13, 2013.

[12] M. Chuishi, J. Wenjun, L. Yaliang, G. Jing, S. Lu, D. Hu, and C. Yun, "Truth discovery on crowd sensing of correlated entities," in *Proceedings of the 13th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '15, 2015.

[13] F. Stefan, K. Gerd, R. Reza, P. Santi, V. Marco, and B. Carlos, "Mining temporal patterns of transport behaviour for predicting future transport usage," in *Proceedings of the 2013 ACM Conference on Pervasive and Ubiquitous Computing Adjunct Publication*, ser. UbiComp '13 Adjunct, 2013.

[14] P. Bei, Z. Yu, W. David, and S. Cyrus, "Crowd sensing of traffic anomalies based on human mobility and social media," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, ser. SIGSPATIAL'13, 2013.

[15] X. Yu, Y. Liu, D. Wei, and M. Ting, "Hybrid markov models used for path prediction," in *Proceedings of 15th International Conference on Computer Communications and Networks*, Oct 2006, pp. 374–379.

[16] M. Karimzadeh, Z. Zhao, F. Gerber, and T. Braun, "Mobile users location prediction with complex behavior understanding," in *2018 IEEE 43rd Conference on Local Computer Networks (LCN) (LCN 2018)*, Chicago, USA, Oct. 2018.

[17] M. G. Elfeky, W. G. Aref, and A. K. Elmagarmid, "Periodicity detection in time series databases," *IEEE Transactions on Knowledge and Data Engineering*, vol. 17, no. 7, pp. 875–887, 2005.

[18] D. Chase, "Class of algorithms for decoding block codes with channel measurement information," *IEEE Transactions on Information Theory*, vol. 18, no. 1, pp. 170–182, 1972.

[19] M. Alistair and Z. Justin, "Rank-biased precision for measurement of retrieval effectiveness," *ACM Trans. Inf. Syst.*, vol. 27, no. 1, Dec. 2008.

[20] C. Bertil, M. Arielle, K. Vaibhav, and G. Benoît, "Capturing complex behaviour for predicting distant future trajectories," in *Proceedings of the 5th ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems*, ser. MobiGIS '16, 2016.

Paper Overview

Accepted:

1. Zhao, Zhongliang; Karimzadeh Motallebiazar, Mostafa; Gerber, Florian; Braun, Torsten (2018). Mobile Crowd Location Prediction with Hybrid Features using Ensemble Learning. Future Generation Computer Systems Elsevier 10.1016/j.future.2018.06.025 DOI: 10.7892/boris.98674
2. Karimzadeh Motallebiazar, Mostafa; Zhao, Zhongliang; Gerber, Florian; Braun, Torsten (4 October 2018). Mobile Users Location Prediction with Complex Behavior Understanding. In: IEEE Conference on Local Computer Networks (IEEE LCN). Chicago, USA. October 1-4, 2018. DOI: 10.7892/boris.116297
3. Karimzadeh Motallebiazar, Mostafa; Zhao, Zhongliang; Gerber, Florian; Braun, Torsten (20 August 2018). Pedestrians Complex Behavior Understanding and Prediction with Hybrid Markov Chain. In: The Eleventh International Workshop on Selected Topics in Wireless and Mobile computing (STWiMob-2018). DOI: 10.7892/boris.118504

Submitted:

4. Karimzadeh Motallebiazar, Mostafa; Gerber, Florian; Zhao, Zhongliang; Braun, Torsten (15 October 2018). Pedestrians Trajectory Prediction in Urban Environments. DOI: 10.7892/boris.120477