Predicting User Mobility in Mobile Radio Networks to Proactively Anticipate Traffic Hotspots

Sebastian Göndör, Abdulbaki Uzun
Telekom Innovation Laboratories, TU Berlin
Service-centric Networking
Berlin, Germany
sebastian.goendoer@tu-berlin.de
abdulbaki.uzun@telekom.de

Till Rohrmann, Julian Tan, Robin Henniges
Technische Universität Berlin
Service-centric Networking
Berlin, Germany
till.rohrmann|robin.henniges@gmail.com
julian.tan@gmx.de

***Abstract*—With approx. 6 million macro cells worldwide and a gross energy consumption of approx. 100 TWh per year as of 2013, mobile networks are one of the major energy consumers in the ICT sector. As trends, such as cloud-based services and other traffic-intensive mobile applications, fuel the growth of mobile traffic demands, operators of mobile telephony networks are forced to continuously extend the capacity of the existing infrastructure by both implementing new technologies as well as by installing new cell towers to provide more bandwidth for mobile users and improve the network's coverage. In order to implement energy-efficient reconfiguration mechanisms in mobile telephony networks as proposed by the project *Communicate Green*, it is essential to anticipate traffic hotspots, so that a network's configuration can be adjusted in time accordingly. Hence, predicting the movement of mobile users on a cellular level of the mobile network is a crucial task. In this paper, we propose a *Movement Prediction System* based on the algorithm of Yavas et al. that allows to determine the future movement of a user on a cellular level using precomputed movement patterns. We extended the algorithm to be capable of preselecting patterns based on time and contextual data in order to improve prediction accuracy. The performance of the algorithm is evaluated based on real and live user movement data from the *OpenMobileNetwork*, which is a platform providing estimated mobile network topology data. We found that the algorithm's prediction quality is sufficient, but requires an extensive amount of recorded user movements to perform well.**

I. INTRODUCTION

To enable customers of mobile telephony providers to stay connected at all times at almost every place, network operators need to maintain a highly sophisticated and complex network infrastructure. The perpetual innovation in the area of mobile communications is forcing network operators to continuously extend the existing infrastructure by introducing new technologies, such as LTE or WiMAX in order to meet the constant growing capacity demands. Areas need to be covered by multiple technologies (e.g., GSM and UMTS) as well as the current state-of-the-art standards LTE/WiMAX in order to provide the most recent technology available and support full connectivity of legacy devices. Hence, a high number of network cells is required. As of late 2012, mobile network operators ran approx. 6 million base stations worldwide [1] to fulfill the desire for the steadily growing bandwidth demands in mobile telephony networks [2] [3]. The total power consumption of mobile networks of approx. 100 TWh per year is contributing to the overall carbon dioxide emissions, a number that is expected to almost triple by 2020 [4].

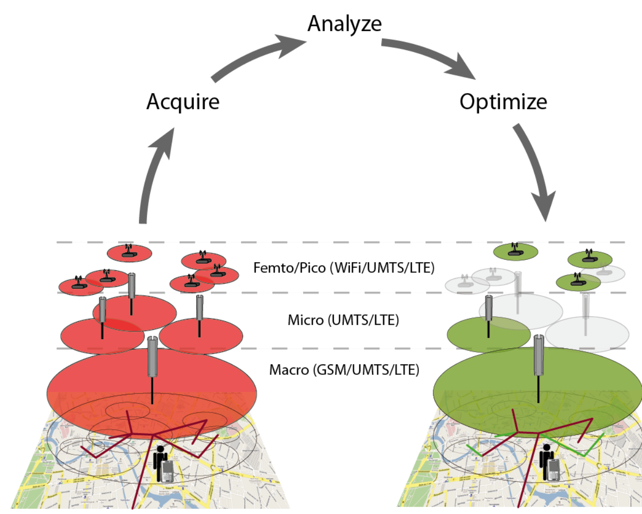

Fig. 1. The basic principle of the project *Communicate Green* is to dynamically suspend unused elements of mobile networks to reduce the overall energy consumption

Therefore, introducing mechanisms to reduce the overall energy consumption of mobile networks is considered to be a crucial task. As base stations of mobile telephony networks are operated 24/7 in an always-on-manner – regardless of the actual traffic demands in the network – a massive amount of energy could be saved by dynamically adapting the provided bandwidth to the actual requirements in an area.

The project *Communicate Green*[1], which is funded by the *German Federal Ministry of Economics and Technology (BMWi)* as part of the *IT2green*[2] initiative, addresses issues of energy efficiency in mobile telephony networks. This can be achieved by applying an adequate reconfiguration of entities in a network, where the network's current state is monitored and modeled by a *Context Management Architecture* as described by Göndör et al. [5]. The reconfiguration solutions comprise the de- and reactivation of a complete cell tower (see Figure 1) as well as switching to a *Bandwidth Expansion Mode (BEM)* [6] or sparse optimization [7].

[1] http://www.communicate-green.de/
[2] http://www.it2green.de/

MOBILWARE 2013, November 11-12, Bologna, Italy

DOI 10.4108/icst.mobilware.2013.254184

As presented by Bayer et al. [8], a certain amount of time $\tau > 0$ is required to apply a specific reconfiguration to network elements, caused by re-initialization of distinct components of the hardware as well as necessary maintenance tasks due to the changes in the network's structure. *MORFEO*, a flexible energy-saving decision algorithm, has been implemented to automatically determine the optimal time to de- or reactivate specific energy efficiency procedures [9]. Still, reactive algorithms could be improved by enabling them to foresee when and where critical situations may occur, i.e., a reconfiguration will become necessary. This requires mechanisms that are able to anticipate the behavior of users, so that the network can be adjusted to a situation that has yet to occur.

As network usage shows certain characteristic patterns as depicted in Figure 2, one can use knowledge of such pattern to estimate the load of a given base station. Dawoud [10] demonstrated how traffic patterns of radio cells can be used to compute a prediction of the future load of this radio cell, where patterns appear to be diverse depending on whether the cell is located in a city or rural area.

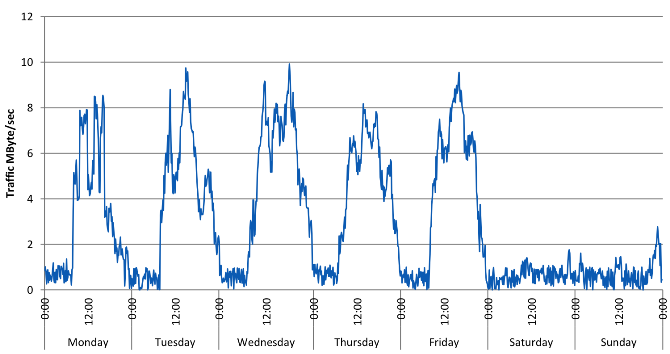

Fig. 2. Data traffic of a typical UMTS NodeB showing recurring patterns over the course of one week

Anyhow, suspending radio cells in order to save energy comes at the risk of some users experiencing an impaired quality of experience (QoE). Hence, in order to avoid situations where a user may be disconnected from the network or experiences a low connection quality in a potentially reconfigured mobile network, it is required to know where a user will be in the future, i.e., which cell he will be connected to. With this knowledge, suspended radio cells can be reactivated in time to ensure seamless connectivity.

In this paper, we address the question how to predict the mobility of users in mobile networks in order to be able to proactively reconfigure distinct network elements. For this purpose, we propose a *Movement Prediction System* that is able to determine the future movement of a user on a cellular level using precomputed movement patterns. This system uses real and live user movement data including mobile network information that has been acquired within the *OpenMobileNetwork* [11] with the purpose of estimating mobile network and WiFi access point topologies worldwide. Predictions are calculated by utilizing the approach proposed by Yavas et al. [12], which we have extended to be capable of preselecting patterns based on contextual data (e.g., time or weather) provided by the *Context Management Architecture* [5]. As a proof of concept, we have implemented the *OpenMobileNetwork Predictions Visualizer*[3] – a map that shows the predicted user movements on a cellular level.

The remainder of the paper is organized as follows: First, we elaborate on existing techniques for movement prediction and present previous work for energy optimizations in radio networks in Section II. Section III describes the *Movement Prediction System* comprising the *OpenMobileNetwork* as well as the prediction algorithm with the proposed improvements, whereas Section IV presents an evaluation of the algorithm. Section V concludes the paper.

II. Related Work

Human movement patterns are more regular than one might expect. In contrast to the assumptions of random trajectories predicted by the long prevailing Lévy flight and random walk models, Gonzalez et al. [13] found that individuals follow highly regular patterns and return with high certainty to a few highly frequented places. Hence, methods that allow discovery and extraction of regular patterns in user movements can be utilized to describe and predict future movements of this individual person.

Especially in urban environments, people tend to walk or drive mostly on streets in order to reach a certain destination. Hence, it is possible to confine the areas one is most likely to move in by eliminating all areas off the street and therefore adding even more regularity to one's movements. Personal habits result in daily routines, which let us choose the same set of paths and sub-paths most of the time. Temporal aspects also significantly contribute to the predictability of human movement. Daily routines, such as going to work, shopping, or attending other regular activities are likely to result in distinct movement patterns, which mostly depend on the time of the day, the day of the week or a day being a workday or a holiday. Furthermore, environmental aspects may affect human movement behavior. Depending on the weather, people often favor certain destinations over others or accept detours if they prevent from getting wet in rainy conditions. Big events also play a role in movement behavior as big crowds of people usually stream towards or from conventions, concerts or other major occasions. All of these features can be taken advantage of in order to identify patterns in human movement, although some are more complicated to make use of or require significantly more training data to be detected.

The rapid development of mobile networks and its services has inspired research of user movement prediction tremendously over the last decade. Knowing the future positions of users allow network operators to allocate required resources in advance, so that smooth handoffs and a continuously high service quality become possible. As such proactive reconfigurations may not be applied just-in-time, mechanisms that are able to predict the path a user will walk or drive enable network operators to estimate the number of mobile devices in a given area or cell and hence use this knowledge to derive the expected traffic demands.

Two of the most influential algorithms in the beginning of user movement prediction, which have also been used for a long time, are the *Mobile Motion Prediction* algorithm

[3] http://www.openmobilenetwork.org/predictions/

developed by Liu and Maguire Jr. [14] and the *LeZi-Update* algorithm proposed by Bhattacharya and Das [15]. The first approach considers human movement that comprises deterministic and random movement. Deterministic movement can be distinguished into circular movements and movements along a path. This kind of movement can be found by pattern extraction while the random part is modeled using Markov models. The prediction is then done by searching the best matching deterministic sequence in a given input sequence and returning the next deterministic position. The latter approach uses the Lempel-Ziv algorithm to minimize the data acquired by a mobile device. The collected data is then utilized to construct a k-order Markov model where the order is estimated implicitly by the Lempel-Ziv dictionary. Outputs from the k-order Markov model are returned as movement predictions.

Markov models were also used by Ashbrook and Starner [16] to predict the movements of individuals that where tracked by GPS enabled devices. If a user stays longer than a given threshold at one position, the position is considered to be a place, while these places are then clustered into higher-level entities, called locations. Locations are used to generate a second order Markov model for prediction purposes. Prasad and Agrawal [17] employ Hidden Markov Models (HMM) to model user movements. The HMMs are learned from the users' movement history and then used for prediction.

Another approach considers the movement history consisting of longitudinal, latitudinal, and altitudinal data to be a non-linear time series. With the means of a non-linear estimator, one then tries to predict future positions. A possible way to do this is delay embedding [18], which was applied by Domenico et al. [19] in the context of *Nokia's Mobile Data Challenge 2012*[4]. They showed that one can accurately predict user positions with delay embedding. Even more interestingly, they could considerably increase the accuracy by taking social ties into account and compute predictions on correlated movement histories. However, this comes at a price of high computational cost and bad scalability.

NextPlace, which is developed by Scellato et al. [20], also relies on delay embedding. The authors used GPS traces of individuals to determine important places being locations they spend a considerable amount of time. Once determined, statistics on the arrival and duration times are collected. These sequences are then transformed into the feature space by delay embedding and used for prediction purposes: Given the current visiting history (consisting of arrival and duration times), similar patterns in the processed history are extracted. The predicted arrival and duration time is the average of the next values of all matching patterns. The advantage of this approach is the considered spatio-temporal aspect, which allows the algorithm to estimate when the user will most likely arrive at a certain place and how long he will stay. However, this only works with a rather small number of places at which the user spends enough time. Hence, it will most likely not perform well on the scale of mobile radio network cells.

Another class of algorithms related to the *Mobile Motion Prediction* algorithm attempts to determine recurring patterns in a given set of movement traces, which are then used to compute the predicted movement of a person. Anagnostopoulos et al. [21] proposed to consider the prediction problem as a classification problem. The input data to this classification problem are subsequences c_1, ..., c_{i+l} of length l and the label of each data point is the subsequent cell c_{i+l+1}. The authors employed a voting scheme to achieve best classification results, i.e., they used machine learning methods to train several classifiers, such as *1-nearest-neighbor* and *C4.5* classifier, and let them vote the label of an input data point. The vote is decided by the majority rule.

Another way of learning the underlying movement regularities is pattern mining as proposed by Yavas et al. [12]. Here, sequential pattern mining methods are employed to detect frequently occurring patterns in a user's movement. These patterns are then utilized to create movement rules, which again are used to predict a user's movement decisions. The aforementioned movement rules are used as an implication: If the user has recently moved in a certain way, which matches a movement rule's premise, then he is expected to continue to move as the conclusion of the rule says. The proposed method works on the network cell level and can predict the next cell. However, it lacks the ability to estimate the arrival and the duration time in this cell, as all state-of-the-art algorithms do.

Refined versions of this algorithm use more exact GPS data but cannot make network cell predictions [22]. Furthermore, Abo-Zahhad et al. [23] presented a multi-scale version of the algorithm. The algorithm is able to compute movement predictions on different scales depending on the user's needs. However, the algorithm requires a data source providing location information on the different scales.

In the project *Communicate Green*, Göndör et al. [5] developed a *Context Management Architecture*, which is capable of acquiring contextual information in distributed environments, such as mobile radio networks. *Context Sources* (e.g., smart mobile devices or base stations) forward relevant data to a central management server, the *Context Manager*, which is responsible for data management and modeling. A policy-based, flexible decision algorithm uses this information to determine an optimal configuration for the network, so that all users are guaranteed to stay connected and the network reduces its overall energy consumption with fully preserved coverage [9].

Dawoud [10] showed that by using a pattern-based forecasting mechanism de- and reactivation of radio cells can be orchestrated before overprovisioning of distinct areas or bottlenecks in terms of bandwidth occur.

In order to improve the prediction capabilities of the mechanisms proposed within *Communicate Green* by adding functionality that allows us to predict a user's future position, we used the pattern mining approach presented by Yavas et al. [12] due to it's superior prediction accuracy in comparison to other prediction methods and the fact that the required training data has already been gathered by the *Context Manager* as well as the *OpenMobileNetwork* as described in Section III. The algorithm is extended by pattern preselection based on time and context information. In the next section, the details of the *Movement Prediction System* including the prediction algorithm and the implemented improvements are presented.

[4] http://research.nokia.com/page/12000

III. MOVEMENT PREDICTION SYSTEM

This section describes the functional entities of the proposed *Movement Prediction System*. The system is based on the prediction mechanism proposed by Yavas et al. [12], which consists of three main steps: First, available data is preprocessed to extract distinct paths (UAPs). Second, the resulting paths are used to extract movement patterns (UMPs). Finally, the extracted patterns are used to forecast a user's actual movements.

A. Data Preprocessing

In order to mine a significant number of paths for the following pattern extraction, a high amount of recorded movement data is necessary. As the algorithm predicts a user's path on the cell layer of mobile networks, the data needs to comprise at least a Cell-ID and a timestamp. Further information, such as neighboring cells, WiFi hotspots, or GPS coordinates have no direct impact on the prediction. For the path segmentation and pattern extraction, we used data from the *OpenMobileNetwork*[5].

The *OpenMobileNetwork* [11] is an open-source crowdsourcing platform for providing mobile network and WiFi access point topology data based on the principles of *Linked Data* [24]. The data is acquired by a crowdsourcing community using the *OpenMobileNetwork for Android (OMNApp)*[6] and *Jewel Chaser*[7] apps [25]. These applications collect network measurements on a user's mobile device, which are then sent to the *OpenMobileNetwork* server in order to compute cell locations as well as cell coverage information [26]. The estimated topology data is semantically modeled using *RDF*[8], *RDF Schema* and other vocabularies enabling an easy integration of the dataset into the *Linking Open Data Cloud*[9] by defined links to related datasets, such as *LinkedGeoData*[10], *Linked Food*[11], or *DBpedia*[12].

By the time of writing this paper, the *OpenMobileNetwork* comprised data with a total of 201,108 unique measurements for 11,214 mobile radio cells out of which 7,196 are located in Germany. Due to the fact that each unique measurement includes extensive information, such as GPS coordinates and data about the radio cell the mobile device was connected to at the time of the measurement, mobility traces can be generated for distinct users, which we used as training data for the algorithm as well as for evaluation purposes. Furthermore, the cell location and coverage information in the dataset allowed us to easily visualize both traces and cell layers on a map.

By using the dataset provided by the *OpenMobileNetwork*, our training data consisted of:

- Cell-ID, Location Area Code (LAC), Mobile Country Code (MCC), Mobile Network Code (MNC), Primary Scrambling Code (PSC), operator, network access type and signal strength of the connected base station
- Cell-ID, Location Area Code (LAC), Primary Scrambling Code (PSC), network access type and signal strength of nearby base stations if information is accessible
- The time at which the data was measured
- The location of the user in longitude and latitude determined by GPS
- The accuracy of the GPS measurement
- A list of services running on the user's cellphone and the traffic generated by those services
- Information about the user's device: operating system, brand, device name, hardware name, manufacturer, model name, product name, SIM card information (MCC, MNC, operator)

Using *OpenMobileNetwork for Android (OMNApp)*, information is polled from the mobile device every 15 seconds. However, a measurement is omitted if relevant sensor data is missing. A missing GPS location fix, for example, will cause a measurement to be considered incomplete and hence it would not be forwarded to the server. As a consequence, incomplete measurements can cause a path of a user not to be recorded end-to-end. Over time, data collection ideally results in a sequence of Cell-IDs in which only neighboring cells can follow its predecessor. Due to loss of signal, however, it can happen that consecutive cells are not adjacent to each other. Especially, when using underground transportation, sizeable jumps between two consecutive cells usually occur since the permanently missing view of sky makes it impossible to retrieve location data via GPS. Moreover, if a user remains within the same cell for a long period, then it is likely that consecutive cell measurements comprise the same Cell-ID. To wipe out such irregularities, a preprocessing step is required in order to extract "clean" paths from the available, possibly flawed data.

In addition, the measurement data is augmented with contextual information that has been acquired by the *Context Management Architecture*. This information includes:

- weather information tailored for the location of the device, including temperature, humidity, windspeed, and chance of rain
- information about regional holidays and local events

The contextual information is used by the prediction mechanism to preselect the best fitting movement patterns for the prediction calculation. In a second step, the raw movement data is transformed into an appropriate format to get rid off ambiguities in the data. First of all, the sequence of Cell-IDs is segmented into reasonable subsequences, which represent paths between two destinations. Considering only those paths as logical units makes sense, because once arrived at a destination, the next steps are usually not determined by the previous movement history, but by the decision which next destination one wants to reach.

[5] http://www.openmobilenetwork.org/
[6] Available at Google Play Store
[7] Available at Google Play Store
[8] Resource Description Framework, http://www.w3.org/RDF/
[9] http://www.lod-cloud.net/
[10] http://www.linkedgeodata.org/
[11] http://www.linkedfood.org/
[12] http://www.dbpedia.org/

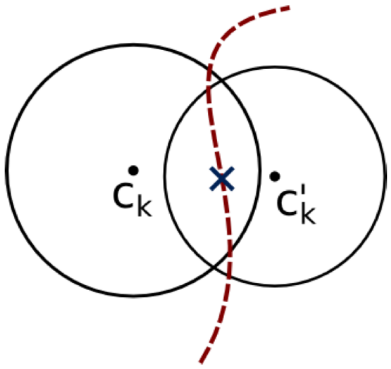

Fig. 3. Overlapping cells might cause different Cell-ID assignment for one and the same position. Furthermore, it might come to an oscillation phenomenon even if one does not move.

A dimorphic problem arises from overlapping network cells as visualized in Figure 3: On the one hand, it is possible to obtain for one and the same path different sequences of Cell-IDs as mobile devices may connect to a different cell. Hence, regular patterns lose their significance as they appear more often in a polymorphic form and will therefore be treated as distinct movement patterns. As a consequence, the prediction quality will not be as high as without overlapping network cells resulting in a significantly higher amount of data for path extraction. On the other hand, overlapping cells and a continuous network related handover to the other cell cause an oscillation phenomenon where it seems that a user moves from one cell to the other and back. This phenomenon implies a user movement even if the user does not move at all. Thus, the oscillation deteriorates the overall prediction performance as well. In the following, we present the path segmentation and the cell clustering procedure, which has been proposed by Bayir et al. [27].

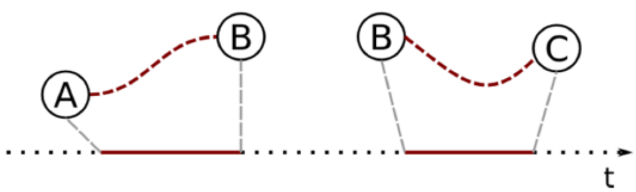

Fig. 4. The user actual paths (UAPs) are extracted from a sequence of Cell-ID measurements by the means of destination detection.

1) Path Segmentation: In order to segment sequences of Cell-IDs into subsequences between two destinations, distinct destinations have to be defined and extracted from the data. Destinations are considered to be places where a user spends a reasonable amount of time, such as at work, the shopping mall, at a restaurant, or at home. Thus, one natural approach would be to consider the resting time of a user in a given area or mobile cell. If this time exceeds a certain threshold δ_{rest}, then this indicates the arrival at a destination. The resting time is defined as the time difference between the moment a user enters a cell and the moment he leaves the same cell as illustrated in Figure 4. Hence, the resting time in the k-th cell in a given sequence is defined as $L^k_{rest} = L^k_{end} - L^k_{start}$.

However, situations in which the user's device loses network connectivity or is turned off (e.g., if the user attends a meeting or the device's battery dies), the sequence is interrupted and a regular destination is not detected. By considering the transition time between successively entered cells, these interruptions can be detected. The transition time between the k-th and the $k+1$-th cell in a given sequence is defined as $L^k_{tra} = L^{k+1}_{start} - L^k_{end}$. If the transition time exceeds a threshold δ_{tra}, then we assume that the user was at a hidden destination, which was not detected as a regular destination. Given these definitions of destinations, the UAPs being subsequences that denote a path between two destinations can be defined. The formal definition of a UAP is described in Equation 1.

$$\begin{aligned} UAP := {} & (L^{i-1}_{tra} > \delta_{tra} \vee L^{i}_{rest} > \delta_{rest}) \wedge \\ & (L^{i+1}_{tra} > \delta_{tra} \vee L^{i+1}_{rest} > \delta_{rest}) \wedge \\ & (\forall j, i, i \leq j < i + l : L^{j}_{tra} \leq \delta_{tra}) \wedge \\ & (\forall j, i, i < j < i + l : L^{j}_{rest} \leq \delta_{rest}) \end{aligned} \tag{1}$$

2) Cell Clustering: In order to avoid the Cell-ID oscillation phenomenon, overlapping cells in the mobile telephony network have to be detected. Since cell towers are usually located closely to each other to guarantee seamless coverage, cells involved into an oscillation can be clustered. By replacing oscillating cells in a UAP by a cluster ID, cell oscillation can be avoided at the cost of a loss in accuracy.

Given a user actual path, cell oscillation can be detected by checking every following pair of Cell-IDs (x, y) how often these two cells alternate. Since more than two cells can overlap, oscillation between more than two cells is possible. Hence, oscillations between three cells (x, y, z) need to be detected as well, e.g., in a sequence $C = (x, y, z, x, y, z, x, y, z)$. If the number of oscillations of a pair (x, y) exceeds a given threshold δ_{osc}, the pair is considered to be a possible merge candidate. Once all oscillating pairs from a UAP have been determined, the merge candidates are clustered by replacing each Cell-ID with a cluster ID, where for each set of merge candidates a new cluster is created if necessary.

3) Topology Construction: As a last preprocessing step, the topology of the mobile network cells is reconstructed from the available data. This is beneficial for the sequential pattern mining algorithm in the learning phase of the prediction algorithm. Knowledge of all neighboring cells for any given cell enables the algorithm to determine which cells can be reached from a given cell in the mobile network and hence prune "impossible" generated patterns, which then do not have to be checked for plausibility in the prediction phase. The topology is approximated by declaring cell y to be reachable from x, if and only if the subsequence (x, y) occurs in one of the UAPs.

B. Prediction Algorithm

The prediction algorithm utilizes a pattern mining approach to predict future movements as proposed by Yavas et al. [12]. It attempts to find subsequences in the UAPs, which are more common than others. These subsequences are the UMPs and contain the essence of the regularities in a user's movement history. By using these UMPs, movement rules can be derived, which consist of a *premise* and a *conclusion* part. The *premise* stands for the movement history while the *conclusion* represents the expected future movement. Thus, by matching a given user movement history with the premises of the movement rules, we can select the best fitting rule and use the conclusion part as our movement prediction. The algorithm is extended by adding pattern preselection functionality that favors patterns recorded under similar environmental conditions as sensed at the time the prediction is computed.

This prediction approach consists of two phases: the learning phase in which movement rules from the UAPs are extracted and the prediction phase in which a user's future movements are predicted on the basis of his latest movement traces.

1) Learning Phase: The learning phase is an offline algorithm, which can be run periodically or when new UAPs have been computed. As input, the algorithm uses the available UAPs and returns the calculated movement patterns (UMP) for each user. To determine the movement rules, sequential pattern mining is applied to find the most frequent user mobility patterns. In Algorithm 1, the pattern mining algorithm is given as pseudo code.

Algorithm 1 Sequential pattern mining algorithm as of Yavaş et al. [12]

```
R ← ∅
while P ≠ ∅ do
    T ← ∅
    for p ∈ P do
        for q ∈ UAPs do
            p.supp ← p.supp + support(p, q)
        end for
        if p.supp > δ_sup then
            T ← T ∪ {p}
        end if
    end for
    P ← generateNewCandidates(T, topology)
    R ← R ∪ T
end while
    return R
```

The algorithm starts initially with all patterns of length 1. For every pattern, it then calculates its support value, which indicates whether this pattern occurs frequently. The support is determined via Equation 2, where a denotes the pattern for which the support is being calculated and b is the support giving UAP.

$$support(a,b) := \begin{cases} \frac{1}{1+dist(a,b)} & \text{if pattern } a \text{ is contained in } b \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

This function takes the influence of noise into account. Usually, human movement consists of a highly deterministic component and a random walk component. The random component can also be considered to be noise in an otherwise regular movement scheme and hence is added to the regular component. If a UAP contains a lot of random movements, then it is considered to be highly corrupted. By using the function $dist(a, b)$, which measures the distance between the sequences a and b, we can assign different support values to different candidate patterns depending on how corrupted they are. In the context of this work, we used the so-called *Levenshtein* distance. This means that candidate patterns, which are only slightly corrupted, will have usually a lower distance to the UAPs than candidate patterns that are highly corrupted. Thus, higher support values are assigned to patterns, which are less corrupted.

After calculating the support values for every candidate pattern, only those patterns are kept whose support value is higher than a given threshold δ_{sup} and which are user mobility patterns of length of at least l. After the UMPs of length l are determined, the new candidate patterns of length $l + 1$ are calculated in the same fashion. Usually, this is done by appending to every UMP of length l all possible Cell-IDs. However, this is not efficient as this would also generate paths that contain sequences of non-neighboring cells. In order to avoid this, the computed topology is utilized that has been created in the third preprocessing step. Only Cell-IDs may be appended to a UMP p that are reachable from the last cell of p. This pruning helps to speed up the sequential pattern mining process significantly. The algorithm then restarts with the new set of candidate patterns of length $l + 1$.

The output of the pattern mining algorithm is a set of UMPs, which are used to generate the movement rules by splitting each UMP $C = c_1, c_2, \ldots, c_n$ at every possible position resulting in a set of $R = \{c_1, \ldots, c_i \rightarrow c_{i+1}, \ldots, c_n \mid 1 < i < n\}$ with a premise and a conclusion. The premises represent the movement history and the conclusions stand for the expected future movements for this user. Thus, given a movement history, one can check all available movement rules for this user for a matching premise and use the matching conclusion as a movement prediction.

As several matching movement rules might be found, a confidence value $conf(R)$ for every rule $R = c_1, \ldots, c_i \rightarrow c_{i+1}, \ldots, c_n$ is determined using Equation 3.

$$conf(c_1, \ldots, c_i \rightarrow c_{i+1}, \ldots, c_n) := \frac{\#c_1, \ldots, c_n}{\#c_1, \ldots, c_i} \tag{3}$$

Here, $\#c$ denotes the number of occurrences of a cell c in the given set of movement rules, i.e., the confidence value determines the significance of a rule. A threshold δ_{conf} is used to exclude rules with a worse confidence value from being used in the later prediction phase.

2) Pattern Preselection: As behavioral habits strongly depend on environmental parameters, such as the time of the day, the weather, or events in vicinity, we propose to privilege movement rules during the prediction phase of the algorithm that have been recorded with environmental parameters similar to the situation in which the prediction should be computed. We argue that this gives more relevant patterns an advantage and results in more accurate movement predictions. The remainder of this section describes the proposed timed- and contextual pattern preselection.

a) Timed Pattern Preselection: As movement patterns in everyday life strongly depend on time (e.g., in the morning people are moving to work and in the afternoon they go to the shopping centers), the prediction system utilizes temporal knowledge. For this purpose, the day is divided into four phases: morning (6 to 12 o'clock), afternoon (12 to 18 o'clock), evening (18 to 24 o'clock), and night (24 to 6 o'clock). For each of these intervals, it is determined for each movement rule how often it has been used during this time. After normalizing over all rules in one interval, a probability distribution is obtained about which rules are most adequate for the current day phase. The distribution is denoted by $P_t(R)$ with R being a movement rule and $t \in \{morning, afternoon, evening, night\}$. This knowledge is then incorporated into the prediction phase in order to preselect the most fitting movement rules, thus making the prediction more accurate.

b) Contextual Pattern Preselection: A similar mechanism is used to preselect movement rules with matching contextual data. Contextual data acquired by the *Context Management Architecture* is used to describe certain features of the environment for which a movement prediction is being calculated. The contextual data is classified into distinct features, e.g., $\texttt{temperature} \in \{cold, warm, hot\}$ or $\texttt{humidity} \in \{dry, humid, raining\}$, where for each recorded rule, a set of features is determined. In the later prediction phase, rules with a contextual feature set that match with the current surrounding of the user will have a greater impact on the prediction accuracy. For example, one might take the bus home from work when it's raining instead of walking home. A weighting factor $W(R) \in [1, 2]$ for a movement rule R is chosen depending on how many contextual features could be matched.

3) Prediction Phase: In the prediction phase, the current movement history $h = c_1, \ldots, c_{i-1}$ of a user for which movement rules have already been calculated is used to predict his future movements. This is done by searching for movement rules whose premise contains h such that the last cell of the premise equals c_{i-1}. These rules are called matching rules. Since there can be several matching rules, we have to select the best for our prediction. For this purpose, we calculate a weighting factor and choose the rule with the highest weight

$$w(R) := (support(R) + conf(R)) \cdot P_t(R) \cdot W(R) \qquad (4)$$

with $support(R)$ being the support of the UMP from which R has been generated, $conf(R)$ being the confidence of R, $P_t(R)$ the probability that the rule R has been used during the time interval t, and $W(R)$ being the weighting factor for contextual information.

C. System Architecture

The prediction mechanism has been implemented in C++ on a Debian-based system. The data preprocessing is triggered periodically and automatically fetches new measurement data from the external *OpenMobileNetwork*. Hence, new movement rules are generated continuously extending the dataset of available movement patterns for active users over time. The architecture (see Figure 5) features a visualization frontend that allows users to directly review the predicted paths on a map.

In the learning phase of the algorithm, movement patterns containing the movement regularities are extracted from the

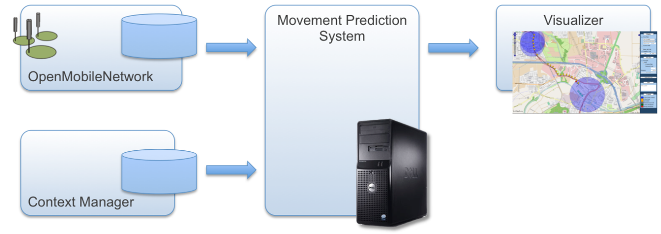

Fig. 5. Architecture of the *Movement Prediction System*. Data is acquired from both the *OpenMobileNetwork* and the *Context Manager* and used for predictions. The resulting predictions are then used by the visualizer.

calculated UAPs. For this purpose, sequential pattern mining is applied. Once a set of frequent UMPs has been determined, the movement rules representing the actual movement regularities are generated automatically, which are then used during the prediction phase to determine the user's next cell movement. Here, all matching rules are weighted according to their support and confidence values where the highest ranked rules are then used for the requested number of cell movement predictions.

1) OpenMobileNetwork Predictions Visualizer: In order to visualize the predicted user movements, a web-based demonstrator has been developed as shown in Figure 6. For this purpose, data acquired from the *OpenMobileNetwork* is used to predict user movements in cellular networks.

Utilizing the visualizer, known paths of *OpenMobileNetwork* users are displayed, which have been used as training data for the prediction algorithm. Hence, the visualizer attempts to predict the next cell association for each step a user made during recording. This allows us to evaluate the accuracy of the predictions by comparing the predicted next cell association from the last step with the actual recorded cell association in the current step.

The demonstrator displays the recorded locations of users and the network cell their mobile device was connected to at this moment. Each cell is represented by a circle with different colors depending on whether the cell has not yet been visited, i.e., is part of the path and will be visited later on (yellow color), has already been visited (blue color), or is the cell the users' device is currently connected to. Furthermore, the predicted cells are colorized depending on the question whether the calculated prediction was correct (green color) or false (red color). This allows a user to directly see how well the algorithm performed on a certain UAP. While using cell diameter data from the *OpenMobileNetwork*, the displayed cell diameter is adjusted to fit to the viewport. This is depicted using a dashed line for the cells and can be deactivated by unchecking the option *Simplify Cells* in the configuration menu on the top right. Also, cells that are not part of the selected UAP are not displayed by default, but can be shown using the option *Show all cells*.

The *OpenMobileNetwork Predictions Visualizer* enables the selection of a distinct path for a user that has been recorded via the *OpenMobileNetwork* mobile application as described in [25]. This is done by first choosing an anonymous user ID and then selecting one of the recorded paths from this user.

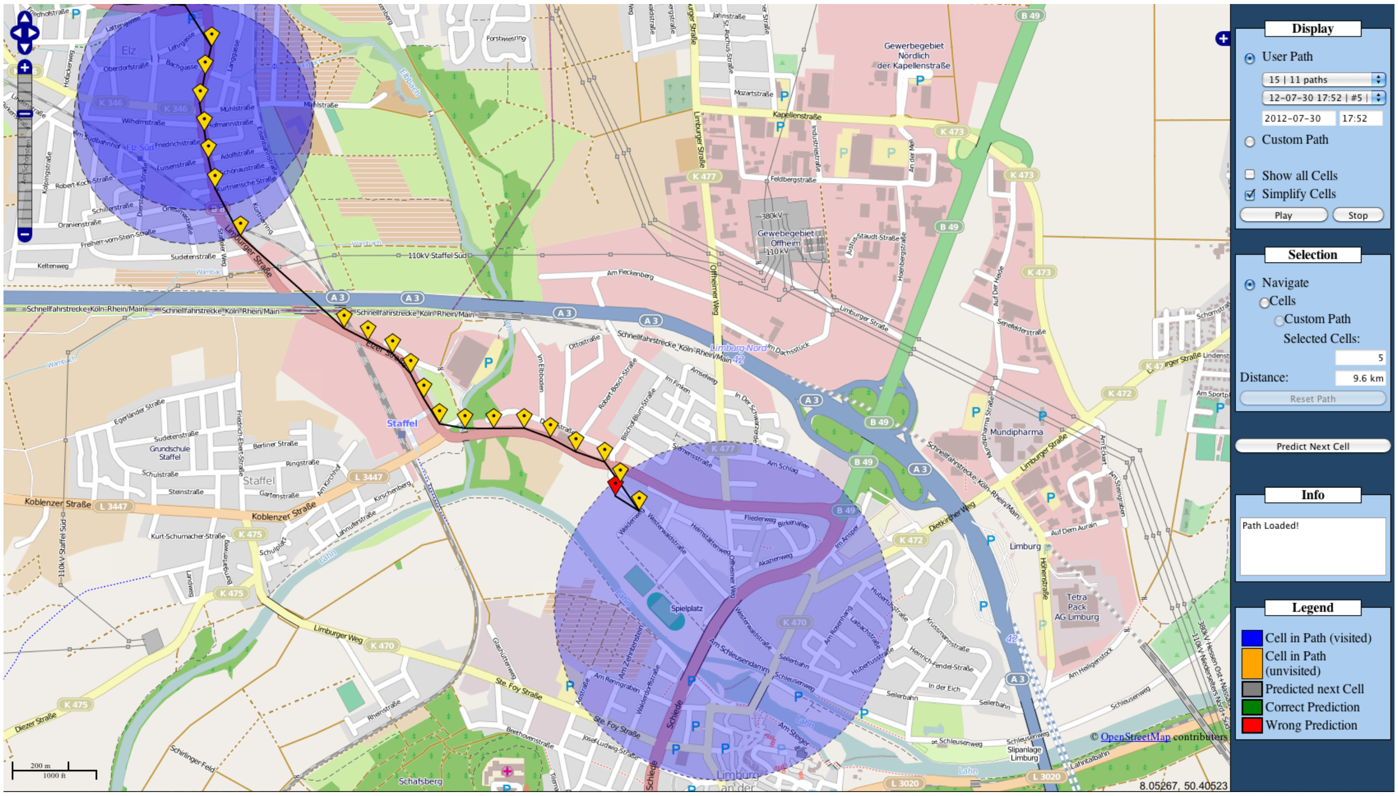

Fig. 6. The *OpenMobileNetwork Predictions Visualizer* showing a UAP and relevant radio cells. The visualizer uses OpenStreetMap data: © OpenStreetMap contributors.

By checking the option *Custom Path*, the visualizer allows to create mock paths by clicking on the map. In this case, the "simulated" path is then used to predict the movement of a non-existing user using an ignorant predictor that randomly chooses a cell from the neighboring cells as a prediction. Once a path has been selected, it is displayed on the map. The *Play* button displays an icon representing the user moving along this path, while the predictions are calculated and visualized in real time.

IV. EVALUATION

In this section, the performance of the implemented *Movement Prediction System* is evaluated using real and live user movement data from the *OpenMobileNetwork*. First, the dataset that has been used to generate the movement rules is described. Second, parameters chosen for the algorithm are determined. Third, the quality of the computed predictions is evaluated.

A. Movement Data

The used prediction method strongly depends on the number of collected paths for each user, i.e., a significant amount of UAPs is required to extract recurrent patterns and thus being able to detect implicit regularities. The dataset provided by the *OpenMobileNetwork* comprises data of 1,269 users out of which only 209 contributed with measurements at all. Only about 10 users contributed about 50% of the total measurements. Therefore, only data from users were utlilized, who provided a significant amount of measurements. Restricting the dataset to users with at least 10 recorded paths, paths with a length of at least one cell, and continuous measurements for at least 60 seconds, a total of 1,377 paths remained. As a comparison, for evaluation purposes, Yavas et al. [12] used a dataset of 10,000 UAPs for one user generating the data from a model. In our dataset, only 3 users recorded enough data to extract at least 100 paths. Hence, we focus our evaluation on data of these users, while omitting the pattern preselection technique as it would further reduce the number of available UAPs.

B. Threshold Values

The prediction model strongly depends on the values of the support and the confidence threshold, as they control which rules will be generated. Since there is no way to know a priori which values yield the highest prediction accuracy, reasonable values needed to be determined. In order to determine appropriate values, the path segmentation algorithm has been run using different parameter values. The resulting optimal values for the various thresholds are listed in Table I.

TABLE I. DETERMINED THRESHOLD VALUES

δ_{rest}	600s
δ_{tra}	600s
δ_{osc}	3

C. Evaluation

We applied our evaluation on the users with the most recorded paths using cross validation and compared the results to an ignorant prediction technique that randomly chooses one of the cells in vicinity of the currently visited cells as the predicted one. Here, we focus on the user with the *User ID 290*, who had 224 recorded path at the time of the evaluation. The accuracy is measured as the ratio of correctly predicted cells to the number of computed predictions.

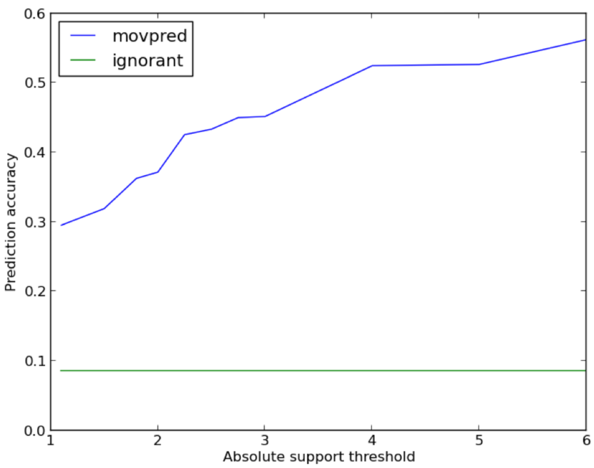

Fig. 7. Prediction accuracy for User 290 with respect to the support threshold

Figure 7 shows the prediction accuracy for *User 290* with respect to the support threshold, which filters movement rules with a high corruption due to a high amount of noise. Here, the confidence threshold has been fixed to $\delta_{sup} = 0.7$. As can be seen, a higher support threshold results in a higher portion of correctly predicted next cells and clearly outperforms the ignorant predictor, while the amount of failed predictions due to insufficient data increases with $\delta_{conf} > 2.5$.

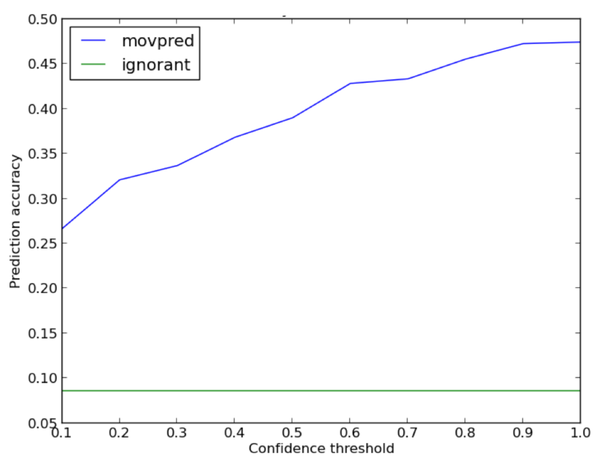

Fig. 8. Prediction accuracy for User 290 with respect to the confidence threshold

The same holds for the prediction accuracy with respect to the confidence threshold with $\delta_{sup} = 2.5$ (see Figure 8). Here, the prediction accuracy increases with a higher confidence threshold, but the number of failed predictions increases for $\delta_{sup} > 0.7$.

We achieved similar results for other users with less paths. The measured accuracy is between 30% and 50%, which is far less accurate than the results found by Yavas et al., who had accuracy values of up to 90% for a generated set of 10,000 UAPs. Hence, we argue that the performance of the *Movement Prediction System* is sufficient, but limited by the fact that an extensive amount of movement data has to be recorded per user for the prediction to function properly and reliable.

V. CONCLUSION

In this paper, we proposed a *Movement Prediction System* that extends the algorithm introduced by Yavas et al. [12] by a pattern preselection technique. The implemented system utilizes live user movement data (including information about the mobile network) collected within the *OpenMobileNetwork* and other contextual information provided by the *Context Management Architecture* in order to compute an estimation for the future movements of a user. By knowing the most likely future position of mobile internet users, network operators can anticipate traffic hotspots and reconfigure network elements accordingly, so bottlenecks or bandwidth overprovisioning can be avoided.

Results show that our system's performance is limited by the amount of available UMPs acquired from the recorded UAPs and hence shows a relatively low but acceptable accuracy compared to the results of Yavas et al. At the same time, however, it clearly outperforms an ignorant prediction technique based on simple guessing.

At the moment, our system focuses on individual user movements on a cellular level. However, for a power management solution in mobile networks, the movement of a large group of users has to be taken into consideration. Therefore, we plan to extend the *Movement Prediction System* by user group prediction mechanisms. We further work on solutions to make the *OpenMobileNetwork for Android* app more "attractive" in order to increase the number of user movement paths within our system. Our overall goal is to provide good prediction results based on real user movements rather than using simulated data.

ACKNOWLEDGMENT

This work has been done within the research project *Communicate Green* (grant number 01ME11010) funded by the *German Federal Ministry of Economics and Technology (BMWi)*. We thank all partners in the consortium of the project: Telekom Innovation Laboratories, Ericsson GmbH, Fraunhofer Gesellschaft zur Förderung der angewandten Forschung e.V., Universität Paderborn, Technische Universität Berlin.

REFERENCES

[1] "Small cell market status - Informa February 2013," White Paper, Small Cell Forum Ltd, Feb 2013. [Online]. Available: http://www.smallcellforum.org/smallcellforum_resources/pdfsend01.php?file=050-SCF_2013Q1-market-status%20report.pdf

[2] "Ericsson Mobility Report - On the Pulse of the Network Society," White Paper, Ericsson, Jun 2013. [Online]. Available: http://www.ericsson.com/res/docs/2013/ericsson-mobility-report-june-2013.pdf

[3] "Cisco Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2012-2017," White Paper, Cisco, Feb 2013. [Online]. Available: http://www.cisco.com/en/US/solutions/collateral/ns341/ns525/ns537/ns705/ns827/white_paper_c11-520862.html

[4] "Ericsson Energy and Carbon Report - On the Impact of the Networked Society," White Paper, Ericsson, 2013. [Online]. Available: http://www.ericsson.com/res/docs/2013/ericsson-energy-and-carbon-report.pdf

[5] S. Göndör, A. Uzun, and A. Küpper, "Towards a Dynamic Adaption of Capacity in Mobile Telephony Networks using Context Information," in *Proceedings of the 11th Intl. Conference on ITS Telecommunications (ITST 2011)*. St. Petersburg, Russia: IEEE, Aug 2011, pp. 606–612. [Online]. Available: http://dx.doi.org/10.1109/ITST.2011.6060128

[6] M. Butt, B. Schubert, M. Kurras, K. Borner, T. Haustein, and L. Thiele, "On the Energy-Bandwidth Trade-off in Green Wireless Networks: System Level Results," in *1st IEEE International Conference on Communications in China Workshops (ICCC 2012)*. IEEE, 2012, pp. 91–95. [Online]. Available: http://dx.doi.org/10.1109/ICCCW.2012.6316482

[7] E. Pollakis, R. Cavalcante, and S. Stanczak, "Enhancing energy efficient network operation in multi-RAT cellular environments through sparse optimization," in *14th IEEE Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2013)*. Darmstadt, Germany: IEEE, 2013.

[8] N. Bayer, D. Sivchenko, H.-J. Einsiedler, A. Roos, A. Uzun, S. Göndör, and A. Küpper, "Energy Optimisation in Heterogeneous Multi-RAT Networks," in *Proceedings of the 15th Intl. Conference on Intelligence in Next Generation Networks (ICIN 2011)*. Berlin, Germany: IEEE, Oct 2011, pp. 139–144. [Online]. Available: http://dx.doi.org/10.1109/ICIN.2011.6081062

[9] K. Gomez, C. Sengul, N. Bayer, R. Riggio, T. Rasheed, and D. Miorandi, "MORFEO: Saving Energy in Wireless Access Infrastructures," in *14th IEEE International Symposium and Workshops on a World of Wireless, Mobile and Multimedia Networks (WoWMoM 2013)*. IEEE, 2013, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/WoWMoM.2013.6583409

[10] S. Dawoud, "Predicting Network Usage to Support Dynamic Network Reconfigurations based on Contextual Information," Master's thesis, Technische Universität Berlin, Berlin, Germany, 2013.

[11] A. Uzun and A. Küpper, "OpenMobileNetwork - Extending the Web of Data by a Dataset for Mobile Networks and Devices," in *Proceedings of the 8th International Conference on Semantic Systems*, ser. I-SEMANTICS '12. New York, NY, USA: ACM, 2012, pp. 17–24. [Online]. Available: http://doi.acm.org/10.1145/2362499.2362503

[12] G. Yavas, D. Katsaros, O. Ulusoy, and Y. Manolopoulos, "A data mining approach for location prediction in mobile environments," *Data & Knowledge Engineering*, vol. 54, no. 2, pp. 121–146, Aug 2005. [Online]. Available: http://dx.doi.org/10.1016/j.datak.2004.09.004

[13] M. C. Gonzalez, C. A. Hidalgo, and A.-L. Barabasi, "Understanding individual human mobility patterns," *Nature*, vol. 453, no. 7196, pp. 779–782, Jun 2008. [Online]. Available: http://dx.doi.org/10.1038/nature06958

[14] G. Liu and G. Maguire Jr., "A Predictive Mobility Management Algorithm for Wireless Mobile Computing and Communications, year = 1995, pages = 268-272, doi = 10.1109/ICUPC.1995.496902, url = http://dx.doi.org/10.1109/ICUPC.1995.496902, publisher = IEEE," in *Fourth IEEE International Conference on Universal Personal Communications*.

[15] A. Bhattacharya and S. K. Das, "LeZi-Update: An Information-Theoretic Approach to Track Mobile Users in PCS Networks," in *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, ser. MobiCom '99. New York, NY, USA: ACM, 1999, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/313451.313457

[16] D. Ashbrook and T. Starner, "Using GPS to learn significant locations and predict movement across multiple users," *Personal Ubiquitous Computing*, vol. 7, no. 5, pp. 275–286, Oct 2003. [Online]. Available: http://dx.doi.org/10.1007/s00779-003-0240-0

[17] P. Prasad and P. Agrawal, "Movement Prediction in Wireless Networks Using Mobility Traces," in *7th IEEE Consumer Communications and Networking Conference (CCNC 2010)*. IEEE, 2010, pp. 1–5. [Online]. Available: http://dx.doi.org/10.1109/CCNC.2010.5421613

[18] A. Basharat and M. Shah, "Time series prediction by chaotic modeling of nonlinear dynamical systems," in *12th IEEE International Conference on Computer Vision*.

[19] M. D. Domenico, A. Lima, and M. Musolesi, "Interdependence and Predictability of Human Mobility and Social Interactions," *CoRR*, vol. abs/1210.2376, 2012. [Online]. Available: http://arxiv.org/abs/1210.2376

[20] S. Scellato, M. Musolesi, C. Mascolo, V. Latora, and A. T. Campbell, "NextPlace: A Spatio-Temporal Prediction Framework for Pervasive Systems," in *Proceedings of the 9th International Conference on Pervasive Computing*, ser. Pervasive '11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 152–169. [Online]. Available: http://dl.acm.org/citation.cfm?id=2021975.2021989

[21] T. Anagnostopoulos, C. Anagnostopoulos, S. Hadjiefthymiades, M. Kyriakakos, and A. Kalousis, "Predicting the Location of Mobile Users: A Machine Learning Approach," in *Proceedings of the 2009 International Conference on Pervasive Services*, ser. ICPS '09. New York, NY, USA: ACM, 2009, pp. 65–72. [Online]. Available: http://doi.acm.org/10.1145/1568199.1568210

[22] T. H. N. Vu, K. H. Ryu, and N. Park, "A method for predicting future location of mobile user for location-based services system," *Computers & Industrial Engineering*, vol. 57, no. 1, pp. 91–105, Aug 2009. [Online]. Available: http://dx.doi.org/10.1016/j.cie.2008.07.009

[23] M. Abo-Zahhad, S. Ahmed, and M. Mourad, "Future Location Prediction of Mobile Subscriber over Mobile Network Using Intra Cell Movement Pattern Algorithm," in *1st International Conference on Communications, Signal Processing, and their Applications (ICCSPA 2013)*. IEEE, 2013, pp. 1–6. [Online]. Available: http://dx.doi.org/10.1109/ICCSPA.2013.6487272

[24] T. Heath and C. Bizer, *Linked Data: Evolving the Web into a Global Data Space*, ser. Synthesis Lectures on the Semantic Web. Morgan & Claypool Publishers, 2011. [Online]. Available: http://dx.doi.org/10.2200/S00334ED1V01Y201102WBE001

[25] A. Uzun, L. Lehmann, T. Geismar, and A. Küpper, "Turning the OpenMobileNetwork into a Live Crowdsourcing Platform for Semantic Context-aware Services," in *Proceedings of the 9th International Conference on Semantic Systems*, ser. I-SEMANTICS '13. New York, NY, USA: ACM, 2013, pp. 89–96. [Online]. Available: http://doi.acm.org/10.1145/2506182.2506194

[26] E. Neidhardt, A. Uzun, U. Bareth, and A. Küpper, "Estimating Locations and Coverage Areas of Mobile Network Cells based on Crowdsourced Data," in *Proceedings of the 6th Joint IFIP Wireless and Mobile Networking Conference (WMNC 2013)*. Dubai, United Arab Emirates: IEEE, Apr 2013, pp. 1–8. [Online]. Available: http://dx.doi.org/10.1109/WMNC.2013.6549010

[27] M. Bayir, M. Demirbas, and N. Eagle, "Discovering SpatioTemporal Mobility Profiles of Cellphone Users," in *IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks Workshops (WoWMoM 2009)*. IEEE, 2009, pp. 1–9. [Online]. Available: http://dx.doi.org/10.1109/WOWMOM.2009.5282489