Mobility Prediction Using Future Knowledge

Michael H. Sun and *Douglas M. Blough*
School of Electrical and Computer Engineering
Georgia Institute of Technology
Atlanta, GA USA
{msun,doug.blough}@ece.gatech.edu

ABSTRACT

Anticipating user mobility can be a critical feature for today's mobile systems. We introduce a novel location predictor which incorporates knowledge of a user's potential future locations to improve prediction accuracy. Such future knowledge is often available through contextual sources such as a user's calendar, e-mail, or instant messaging conversations. Simulation results show that our future knowledge leveraging location predictor can improve prediction accuracy by 3% to 95% over history-only Markov predictors, depending on the amount of future knowledge that is available and the type of mobility exhibited by users.

Categories and Subject Descriptors: H.4. [Information Systems Applications]: Miscellaneous

General Terms: Algorithms, Experimentation, Measurement, Performance

Keywords: mobility prediction, location prediction, mobility management

1. INTRODUCTION

The ability to track and predict a mobile user's location is a critical need for today's mobile systems and applications. To support the growing need for accurate location prediction in mobile applications, we have designed and implemented a location prediction system built upon a novel location predictor that leverages knowledge about a user's potential future locations in determining its predictions. The notion of locations is abstract in nature and can be configured in the system (e.g. access points, IP addresses, etc.). Taking the popular and relatively simple order-k ($O(k)$) Markov predictor [7] as a baseline example, we demonstrate how future knowledge derived from contextual sources such as a user's calendar can be utilized to improve prediction accuracy. We believe our techniques are applicable as well to other history-based location prediction schemes such as those utilizing Bayesian networks and neural networks [2,3].

2. RELATED WORK

Our work falls under the class of domain-independent location predictors which do not require any semantic interpretation of their locations, and in general, observe past movements in order to form predictions [1,6–8]. They operate on the premise that there is temporal regularity in the movements of nodes and that by accurately modeling these patterns, future movements can be predicted.

Several history-based predictors [1,8] have been proposed that are based on a text compression algorithm by Ziv and Lempel [9]. The LZ parsing algorithm assumes that movements are generated by a finite-state Markov source–where the next symbol is dependent only on the current state–and constructs an LZ tree with nodes comprised of substrings produced by parsing the sequence of locations visited by a user [7]. Predictions of a user's next locations can then be derived by traversing the LZ tree and searching for the most frequently seen substring given the current state.

The $O(k)$ Markov family of predictors can be seen as a simplified class of LZ-based predictors where the finite-state Markov source is constrained to a specific k. Song et al. evaluated in [7] several $O(k)$ Markov predictors with an empirical set of traces collected on the Dartmouth campus wireless network. Their findings show that with a *fallback optimization*–which recursively uses the result of the $O(k-1)$ predictor when no prediction was produced by the $O(k)$ predictor–the predictor performed as well as or better than their LZ-based relatives. It was also observed that the $O(2)$ predictor with the fallback optimization provided the best accuracy as higher order Markov predictors suffered from an increasing number of prediction misses.

All of the LZ-based and Markov-based predictors cited base their predictions on past movement histories only. We introduce a predictor that utilizes future information about a user in order to improve prediction accuracy and evaluate our predictor's performance with real world traces of *various location granularities* which, to the best of our knowledge, has not been previously performed. Our results indicate that in many cases, significant performance gains can be achieved by incorporating potential future knowledge into the predictor.

3. BACKGROUND

In this section, we discuss the nature of location prediction, provide a general model for the order-k Markov-based family of predictors, and define the metrics used to evaluate the predictors. We adopt the notation and terminology of [7].

MSWIM'07, October 22–26, 2007, Chania, Crete Island, Greece.
Copyright 2007 ACM 978-1-59593-851-0/07/0010 ...$5.00.

3.1 Location prediction

At at any point in time, a user is deemed to be in a given *location*. A location can refer to a number of actual locales—an access point, a building, a subnet—but is represented by a symbol a drawn from a finite alphabet A, the set of all possible locations in which a user can reside. The user may move between any of the locations at any point in time. A *location history* $L_n = a_1a_2...a_n$ is attributed to a user and consists of a sequence of location symbols drawn from A representing the first n locations held by the user since the start of observation. The goal of location prediction is to accurately predict a_{n+1}, the next location a user will reside at, given a location history L_n.

3.2 Order-k Markov predictors

The $O(k)$ Markov family of predictors attempt to predict the next location from a current context comprised of the k most recent symbols in the location history L_n. The current context is defined to be $c = a_{n-k+1}, ..., a_n$ and represents a state in the Markov model. Transitions from states represent the possible next locations a user may take. Let a user's location be the random variable X and let $X(i,j)$ be the sequence of random variates $X_iX_{i+1}...X_j$ for any $1 \leq i \leq j \leq n$. The $O(k)$ Markov predictor operates by assuming for all $a \in A$ and $i \in 1, 2, ..., n$ that,

$$\begin{aligned} &P(X_{n+1} = a|X(1,n) = L_n) \\ &\quad = P(X_{n+1} = a|X(n-k+1,n) = c) \\ &\quad = P(X_{i+k+1} = a|X(i+1,i+k) = c) \end{aligned}$$

where $P(X_i = a_i|...)$ denotes the probability that X_i takes the value a_i.

Since the transition probabilities are not known ahead of time, on-line predictors maintain running estimates of these probabilities, $\widehat{P}$, from the current history L_n and the current context according to the equation,

$$\widehat{P}(X_{n+1} = a|L_n) = \frac{N(ca, L_n)}{N(c, L_n)}$$

where $N(s_1, s_2)$ denotes the number of times the substring s_1 occurs in the string s_2.

For $L_n = a_1a_2 \ldots a_n$, let $L_n(i,j) = a_i \ldots a_j$. The transition probabilities can be represented by a matrix M, where rows are indexed by length-k strings from A^k and the columns are indexed by symbols from A so that $P(X_{n+1} = a|X(1,n) = L_n) = M(s,a)$ where $s = c = L_n(n-k+1,n)$ and a is the next location. The predictor operates by examining the row indexed by the current context c and selecting the column indexed by location a with the highest probability.

3.3 Fallback optimization

The fallback optimization involves recursing to an $O(k-1)$ Markov predictor when the $O(k)$ predictor fails to make a prediction. For the remainder of this paper, references to the $O(k)$ predictor will imply a basic $O(k)$ predictor with the fallback optimization.

3.4 Predictor operation and metrics

The location predictor operates by tracking a user's movements and predicting the user's next location whenever his/her current context changes. Prediction attempts can result in three outcomes: a *correct prediction*, an *incorrect prediction*, and a *prediction miss*. When a prediction is made and is equivalent to the actual next location of a user, a correct prediction has been made; otherwise it is deemed an incorrect prediction. If the predictor is unable to make a prediction, a prediction miss occurs. A predictor's *accuracy* is defined to be the number of correct predictions divided by the number of locations visited by a user. A predictor's *performance* is taken to be its prediction accuracy. The *conditional accuracy* of a predictor is the number of correct predictions divided by the number of predictions made. Conditional accuracy reveals how accurate a predictor is when it *does* make a prediction as prediction misses are ignored.

4. FUTURE-ENHANCED PREDICTION

Current $O(k)$ Markov predictors operate only on states derived from past user movements. Our *future-enhanced Markov predictor* utilizes knowledge of a user's potential presence at a future location at a specified future time; we refer to each record of such information as a *future event*. Future event information can be amassed from a number of readily available sources such as Microsoft Outlook, Lotus Notes, and Google Calendar. We incorporate future events into the context of an $O(k)$ Markov predictor, enriching the states in the underlying Markov model.

4.1 Basic $O(k,f)$ predictor

In this subsection, we define the basic $O(k,f)$ Markov predictor, which extends the formal model of an $O(k)$ Markov predictor to incorporate future events. For a given step n in the location history, we define the future location list, $FL_{(m,n)} = b_1b_2 \ldots b_m$, where $b_i \in A, 1 \leq i \leq m$. $FL_{(m,n)}$ represents the sequence of possible future locations that are in the user's calendar (or other future information source) when the location history is L_n. Note that there is a different future location list for each possible point in the location history.

Let $FL_{(m,n)}(i,j) = b_i \ldots b_j$. We redefine the context to include some locations from the current location history and some locations from the future location list. For an $O(k,f)$ Markov predictor, we define the context to be $c = L_n(n-k+f+1,n), FL_{(m,n)}(1,f)$, i.e. the last $k-f$ locations from the location history and the first f locations from the future location list. Just as before, the states of the Markov model are the possible contexts and the transitions from a current context (state) represent the possible next locations.

As in the $O(k)$ Markov predictor, we keep a running estimate of transition probabilities over time as the predictor executes. Now,

$$\widehat{P}(X_{n+1} = a|L_n, FL_{(m,n)}) = \frac{N(ca, L_n, FL_{(m,n)})}{N(c, L_n, FL_{(m,n)})}$$

where $N(ca, L_n, FL_{(m,n)})$ represents the number of times that the context c occurs in the pair $(L_n, FL_{(m,n)})$ and the next location is a, and $N(c, L_n, FL_{(m,n)})$ represents the total number of occurrences of the context c in $(L_n, FL_{(m,n)})$.

4.2 Optimizations

We describe a number of variations to the basic $O(k,f)$ predictor that aim to increase the likelihood that available future knowledge can be utilized in a prediction attempt.

4.2.1 Fallback to $O(k)$ predictor

When the future-enhanced $O(k,f)$ Markov predictor fails to make a prediction, the predictor defaults to using the history-only $O(k-f)$ Markov predictor. This fallback allows the predictor to take advantage of the greater *conditional accuracy* of the $O(k,f)$ predictor without sacrificing overall accuracy because of prediction misses by the more selective $O(k,f)$.

4.2.2 Fallback f

If a prediction miss results with the $O(k,f)$ predictor, fallback recursively to the $O(k-1,f-1)$ predictor.

4.2.3 Wildcard

The wildcard optimization provides a regular expression type of operation on contexts. When searching for prediction candidates, rather than requiring that the current context exactly match a row's index in M, the future portion of the current context, $FL(1,f) = b_1 b_2 ... b_f$, is treated as a regular expression $FL_{wildcard}(1,f) = A^* b_1 A^* b_2 A^* ... A^* b_f A^*$. If $FL_{wildcard}(1,f)$ matches a row's index context, elements in that row will be considered prediction candidates. The wildcard optimization is used in conjunction with the fallback f optimization so that as fallback occurs with an $O(k,f-i)$ predictor, $FL_{wildcard}(1,f-1)$ is used to match the row indices in M.

4.3 Hybrid $O(k,f)$ predictor

One potential weakness of the $O(k,f)$ predictor is that it assumes future knowledge is available consistently throughout a user's history. To mitigate this issue, we propose a *hybrid $O(k,f)$ predictor* that does not require future information to be consistently present in the states of the Markov model. We represent Markov states in the transition matrix M as contexts of an $O(k+f)$ history-only Markov predictor. The current context remains $c_{current} = L(n-k+1,n), FL(1,f)$. To make a prediction, $c_{current}$ is treated as a single sequence and matched against the row indices in M as a normal $O(k)$ predictor would operate. The intuition behind such a mechanism is that when future information has not been consistently available, it will be more likely to find matches between a user's current $O(k,f)$ context and his/her $O(k+f)$ history contexts. The hybrid $O(k,f)$ predictor can make use of the same optimizations as its basic analogue.

5. EVALUATION

In this section, we evaluate the performance of our future-enhanced predictors in comparison with their history-only counterparts. The *median accuracy* is defined as the median prediction accuracy of all users in a trace set.

5.1 Future-enhanced prediction

The accuracy of the *basic $O(k,f)$ predictor* and its optimizations are examined by running the predictors on the Dartmouth movement traces [4]. It can only be beneficial to fallback to the history-only $O(k)$ predictor when no prediction can be made by a future-enhanced $O(k,f)$ predictor, so all $O(k,f)$ predictors utilize this fallback. To facilitate direct comparisons with the results in [7], we utilize the same data set which contains only user traces with more than 1000 movements. There are 2,195 users in this trace set.

Since users' movements are re-enacted through traces, synthetic future information must be created for each user. p% of the movements found in a user's trace are randomly extracted and inputed as future events to the predictor.

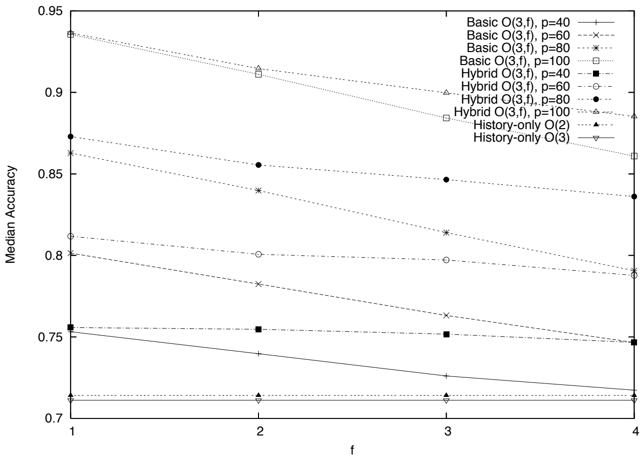

Figure 1: Median accuracies of the basic $O(3,f)$ and hybrid $O(3,f)$ predictors

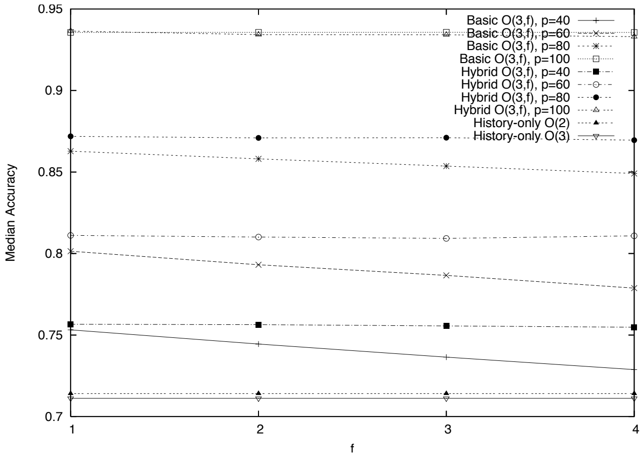

Figure 2: Median accuracies of the basic $O(3,f)$ and hybrid $O(3,f)$ predictors with fallback f

The performance of the *basic $O(k,f)$ predictor* is shown in Figure 1 for $k = 3, f = 1,2,3,4$, and $p = 100, 80, 60, 40$. For comparison purposes, the median accuracies of the $O(3)$ and $O(2)$ predictors have been included; the $O(2)$ predictor was the highest performing predictor in [7]. Clearly, the use of future information has results in greater prediction accuracy as the basic $O(k,f)$ predictor outperforms the $O(k)$ predictor for all parameters shown. For the basic $O(k,f)$ predictor, median accuracy is highest at $f = 1$ and declines linearly for increasing values of f. The decreasing median accuracy can be attributed to prediction misses resulting from the increased amount of state that must be matched for successful prediction. The amount of future information available to the predictor is the significant factor for prediction performance. Note that $p = 100$ represents the optimal case in which the predictor has complete future knowledge of all of a user's movements.

Figure 2 shows the performance of the *basic $O(k,f)$ predictor with fallback f*. Note that at $f = 1$, this predictor degenerates into the basic $O(k,f)$ predictor since no fallback on f can occur. Though the fallback f optimization

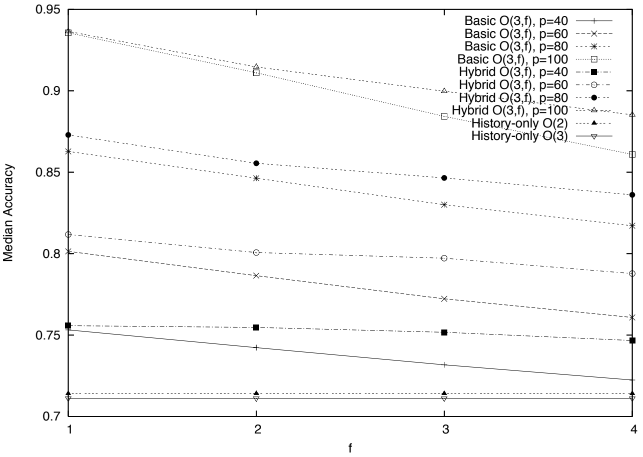

Figure 3: Median accuracies of basic $O(3, f)$ and hybrid $O(3, f)$ predictors with wildcard

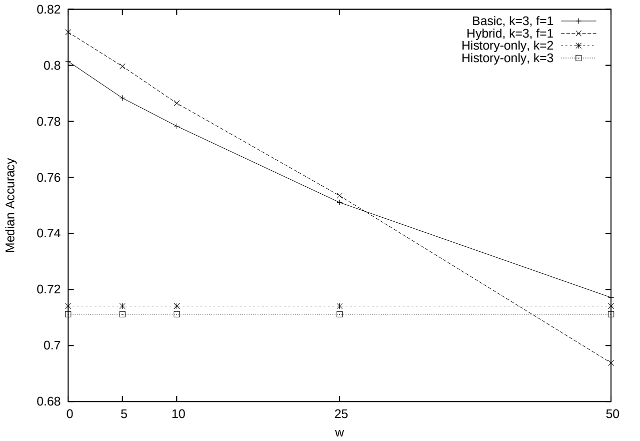

Figure 4: Effect of incorrect future information on the basic and hybrid $O(3, 1)$ predictors for $p = 60$

does not improve upon the best accuracy achieved at $f = 1$, the fallback f allows the predictor to mitigate prediction misses when higher values of f are used by falling back to predictors requiring less matched state. The slight losses in accuracy when higher values of f are used is a puzzling result we discuss in Section 6.

Figure 3 shows the performance of the *basic $O(k, f)$ predictor with wildcard*. Intuitively, the wildcard optimization should reduce the number of prediction misses by allowing close, but imperfect matches to occur. Also note that the wildcard optimization automatically requires the fallback f optimization in order to operate. The results show however, that the wildcard optimization is not effective in improving performance as the median accuracy actually falls significantly for increasing values of f, i.e. the situations in which wild cards take effect. In Section 6, we discuss several modified schemes which we believe might better leverage the potential benefits of the wildcard optimization.

We then evaluate the performance of our *hybrid $O(k, f)$ predictor*, which should provide better performance when lower levels of future knowledge are available, i.e. p. Figures 1, 2, and 3 compare the performances of the *hybrid $O(3, f)$ predictor and its optimizations* with their *basic $O(3, f)$ analogues*. The hybrid predictor continuously outperforms the basic version, especially for values of p lower than $p = 100$. Our intuition that the hybrid predictor can operate with lower levels of future knowledge is confirmed.

We conclude that the best performing predictor is the the hybrid $O(3, 1)$ predictor. Note that when $f = 1$, the fallback f and wildcard optimizations have no effect. Performance gains ranging from 6% to 30% are achieved by using the hybrid $O(3, 1)$ predictor as opposed to the $O(3)$ and $O(2)$ predictors. The amount of available future knowledge is the key factor affecting performance improvement, though even at moderate levels of future knowledge ($p = 60$), accuracy improvements of roughly 14% are achieved.

Thus far, our simulations have assumed that for a given p, all future knowledge available was correct. In reality, users may not have completely accurate information in their calendars. We define the parameter w to represent the percentage of future events that are incorrect, i.e. the wrong location noted in a user's calendar or other contextual source. Figure 4 shows the decline in median accuracy of the basic and hybrid $O(3, 1)$ predictors for $p = 60$ for various values of w. Even when a significant portion of the future events are incorrect ($w = 25$), the basic and hybrid $O(3, 1)$ predictors continue to outperform the $O(3)$ and $O(2)$ predictors. Only when half of the reported future events are incorrect does the use of future knowledge hinder the accuracy of prediction.

5.2 Traces with ping-pong effects removed

Up to this point, all simulations have been performed using the set of access point traces used in [7]. As noted by [5], these traces contain many *ping-pong transitions*—frequent re-associations between two or three access points in a short period of time. The authors of [5] believe the removal of ping-pong transitions from the traces provides a more accurate portrayal of user movements across wireless networks. Given these concerns, we have processed the Dartmouth traces in the same fashion as [5] to remove ping-pong transitions and "OFF" locations.

We are interested in how a different and potentially more realistic trace set will affect our hybrid $O(k, f)$ predictor's performance. Figure 5 shows the improvement possible by the hybrid $O(3, 1)$ predictor when compared to a $O(2)$ predictor. Optimally, a 95% performance increase is possible when $p = 100$, but even for lower values of $p = 80, 60, 40$, we see improvements of 73%, 52%, and 31%, respectively. Note that these large improvements of the hybrid predictor over the history-only predictor are a result of the poor performance of the history-only predictor. Much of the regularity that the history-only predictor relies on for its accuracy disappears with removal of the ping-pong effects. These results demonstrate that the relative performances of future-enhanced and history-only predictors depend not only on the amount of future information available, but also on the nature of the mobility patterns.

5.3 Location granularity

We evaluate how the hybrid $O(k, f)$ predictor performs with regard to various location granularities; the predictors are run through trace sets representing movements between buildings and between subnets across the internet.

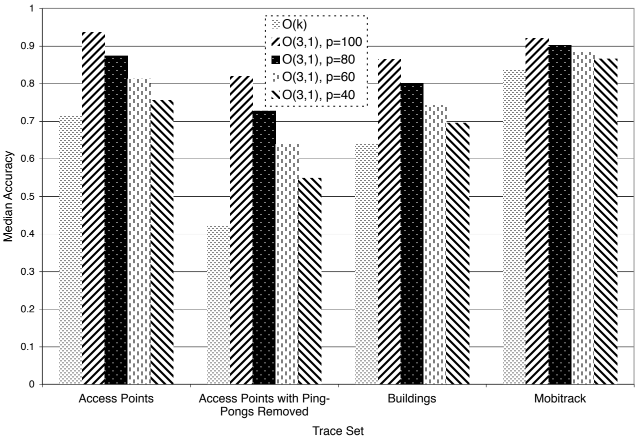

Figure 5: Performance of the hybrid $O(3,1)$ predictor on various trace sets

5.3.1 Reduced Dartmouth traces

We have processed the Dartmouth user traces so that movements between access points in the same building are coalesced into a single location bearing the building's name. The "OFF" location has also been eliminated. The result is the *Buildings* trace set.

Figure 5 illustrates that with coarse-grained locations, the hybrid $O(3,1)$ predictor continues to outperform the $O(2)$ predictor.

5.3.2 Mobitrack traces

We have implemented the Mobitrack agent to collect movements across the Internet of four users over a seven month period. For each user, the agent runs in the background on the laptop, reporting to the server each time the laptop associates with a new sub-net or loses network connectivity.

The significance of these traces is that they are *host-centric*, offering a log of movements from the user's viewpoint. Other traces such as those collected on the Dartmouth campus wireless network are *domain-centric* as they reflect all of the movements of users observed while on the Dartmouth network only. For pervasive applications, *host-centric* traces are critical in providing the application an accurate picture of users' total movement patterns.

Figure 5 reveals that both predictors—the $O(2)$ and the hybrid $O(3,1)$—perform very well on the Mobitrack traces. However, the hybrid $O(3,1)$ predictor still provides some improvement in median accuracy. While the number of users is small and the trace lengths are relatively short (due primarily to coarser location granularity), these preliminary results indicate that host-centric movements tend to be more easily predicted than domain-centric ones. This is likely due to the high degree of regularity in work day patterns, where users move regularly between office and home locations.

6. DISCUSSION & FUTURE WORK

Our evaluations show that the *hybrid $O(k,f)$ predictor* consistently outperforms the best history-only $O(k)$ Markov predictor. More importantly, the hybrid $O(k,f)$ predictor maintains good accuracy across varying location granularities. We were surprised to find that larger values of f—reflecting the use of more future information—and the wildcard optimization were not helpful to performance. The increased number of prediction misses due to the more stringent matching criteria cannot fully explain why predictor performance fails to increase even when the fallback f optimization is incorporated to mitigate these misses. We speculate that an optimization providing greater weight to the first future event as compared to subsequent future events could potentially leverage larger f values more effectively, especially when the wildcard optimization is utilized. We also plan to incorporate information about the scheduled times of future events into the prediction schemes, which might help to identify cases where the use of wildcards can improve prediction performance. The element of time was not included in our initial evaluation, primarily because we first wanted to provide a direct comparison of future-enhanced Markov versus history-only Markov predictors, which is not possible if time is included. However, knowledge of scheduled event times is clearly an untapped source of highly relevant information, which has the potential to boost the accuracy of future-enhanced prediction significantly.

7. REFERENCES

[1] A. Bhattacharya and S. Das. LeZi-update: an information-theoretic approach to track mobile users in PCS networks. *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 1–12, 1999.

[2] J. Biesterfeld, E. Ennigrou, and K. Jobmann. Location Prediction in Mobile Networks with Neural Networks. *Proc. IWANNT*, 97:207–214.

[3] C. Cheng, R. Jain, and E. van den Berg. Location prediction algorithms for mobile wireless systems. *Wireless internet handbook: technologies, standards, and application table of contents*, pages 245–263, 2003.

[4] D. Kotz, T. Henderson, and I. Abyzov. CRAWDAD trace dartmouth/campus/movement/infocom04 (v. 2004-08-05). Downloaded from http://crawdad.cs.dartmouth.edu, Aug. 2004.

[5] J. Lee and J. Hou. Modeling steady-state and transient behaviors of user mobility:: formulation, analysis, and application. *Proceedings of the seventh ACM international symposium on Mobile ad hoc networking and computing*, pages 85–96, 2006.

[6] G. Liu and G. Maguire. A class of mobile motion prediction algorithms for wireless mobile computing and communications. *Mobile Networks and Applications*, 1(2):113–121, 1996.

[7] L. Song, D. Kotz, R. Jain, and X. He. Evaluating location predictors with extensive Wi-Fi mobility data. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, volume 2, pages 1414–1424, March 2004.

[8] F. Yu and V. Leung. Mobility-based predictive call admission control and bandwidth reservation in wireless cellular networks. *Computer Networks*, 38(5):577–589, 2002.

[9] J. Ziv and A. Lempel. Compression of individual sequences via variable-rate coding. *Information Theory, IEEE Transactions on*, 24(5):530–536, 1978.