

Predicting a User's Next Cell With Supervised Learning Based on Channel States

Xu Chen, François Mériaux, Stefan Valentin

▶ **To cite this version:**

Xu Chen, François Mériaux, Stefan Valentin. Predicting a User's Next Cell With Supervised Learning Based on Channel States. IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC 2013), Jun 2013, Darmstadt, Germany. pp.1-5, 10.1109/spawc.2013.6612007 . hal-00821207

HAL Id: hal-00821207
https://hal.science/hal-00821207

Submitted on 13 May 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Predicting a User's Next Cell With Supervised Learning Based on Channel States

Xu Chen*‡, François Mériaux†‡, and Stefan Valentin‡

*Department of Electrical Engineering, Princeton University, NJ, USA
†Laboratoire des Signaux et Systèmes, SUPELEC, Paris, France
‡Bell Labs, Alcatel-Lucent, Stuttgart, Germany
xuchen@princeton.edu, meriaux@lss.supelec.fr, and stefan.valentin@alcatel-lucent.com

***Abstract*—Knowing a user's next cell allows more efficient resource allocation and enables new location-aware services. To anticipate the cell a user will hand-over to, we introduce a new machine learning based prediction system. Therein, we formulate the prediction as a classification problem based on information that is readily available in cellular networks. Using only Channel State Information (CSI) and handover history, we perform classification by embedding Support Vector Machines (SVMs) into an efficient pre-processing structure. Simulation results from a Manhattan Grid scenario and from a realistic radio map of downtown Frankfurt show that our system provides timely prediction at high accuracy.**

I. Introduction

Localizing a user and tracking its trajectory change today's cellular networks. At application layer, many services make heavy use of the user's current position. At the physical layer, accurate tracking is crucial for many beamforming approaches.

In this paper, we focus on a coarse localization of the wireless user. Instead of obtaining accurate geographical coordinates, we are satisfied in expressing the user's location in terms of a cell index. However, rather than obtaining the user's current location or a short-term trajectory, we propose a framework to *predict* the next cell the user will hand-over to. Although this prediction is performed at low geographical accuracy and at large time scale, it has to be provided early enough to allow the system to adapt. This requires an accurate prediction before the user enters the next cell, thus, rendering simple, linear predictors inapplicable with many applications.

Such applications include various adaptation functions at the medium access and network layer. New, *context-aware* schedulers can provide seamless quality-of-service by knowing in advance that a user will join a congested cell [1]. Interference management schemes can blank subframes of interfering users even before they join an interfered cell and context-aware handover schemes can use our long-term prediction to make faster and better decisions.

Unlike previous work, based on the Global Positioning System (GPS) or Time Difference Of Arrival (TDOA) [2], our solution combines Channel State Information (CSI) with limited handover history. This combination of long-term handover information with short-term CSI, has several benefits:

- It is fully functional within buildings. This is not the case for GPS-based solutions.
- It employs information that is readily available in cellular networks. As handhelds perform frequent CSI feedback and the network stores handover history, no signaling costs are added to the radio link and implementation effort is limited.
- It predicts the user's next cell early and at high accuracy. This high performance was observed in a synthetic and in a realistic scenario, without strong assumptions on user mobility.

Our paper is structured as follows. We discuss related approaches in Sec. II and introduce the system model in Sec. III. Sec. IV details the proposed prediction framework, which is studied in Sec. V. Sec. VI concludes the paper.

II. Related Work

Two fields in literature are related to our work: predicting the user's trajectory and estimating its current location. Some work on localization [2], [3], [4] uses CSI as input or shares the application of Support Vector Machines (SVMs) [5] with our approach. In contrast to this work, we complement CSI by the users' handover history to predict the user's next cell. Thus, our approach differs in objective as well as in input from the state-of-the-art in localization.

Following our objective, [6], [7] employ machine learning to predict a user's next cell. These predictors are solely based on handover history, while our work combines this information with CSI. Both types of information are efficiently available in cellular networks but substantially differ in time scale. While CSI is periodically reported within several milliseconds [8], obtaining a sufficiently long vector of previous handovers requires minutes or, for slow users, hours.

Unlike handover history, large CSI vectors can be obtained quickly and capture the scatterers in a user's propagation environment. We will see that such CSI "fingerprints" provide substantially higher prediction accuracy than handover history alone.

III. System Model

We study a cellular network with $\mathcal{I} = \{1, \ldots, I\}$ users and a base station (BS), under the following modeling assumptions.

This work was supported by the program *RISE professional* of the DAAD and by the *PhD@Bell Labs* internship program.

A. Channel Model

We assume that the wireless communication channel of user $i \in \mathcal{I}$ within a discrete time slot t is characterized by the channel gain

$$g_i(t) = |h_i(t)|^2 d_i(t)^{-\alpha(x_i(t))}. \quad (1)$$

To account for long-term propagation loss, $d_i(t)$ denotes the distance between user i and BS. The path-loss exponent is denoted by $\alpha(x_i(t))$ and depends on the user's geographical position $x_i(t)$.

Fast fading is reflected by $|h_i(t)|^2$, which we assume to be an exponentially distributed random variable that is independent among the users. We adopt the Jakes-like fading model where the time-correlation of $d_i(t)$ follows the zeroth-order Bessel function of the first kind, which is parametrized by the maximum Doppler shift defined by the user's speed. This model is widely used to reflect an isotropic scattering environments or ensembles of multiple environments [9, Sec. 2.4.3].

B. CSI Feedback

High-rate communication systems such as IEEE 802.16 and LTE, adapt the modulation scheme, code rate, and allocated time-frequency blocks to the current state of the wireless channel. As in those systems the channel cannot be assumed to be reciprocal, users perform periodic feedback of CSI. To accurately adapt to a time-variant channel, this reporting should be performed once per channel coherence time, i.e., between 1 and 20 ms in many scenarios. While typically a quantized channel state is reported, we assume the reporting of the channel gain $g_i(t)$ for tractability.

C. Mobility Model

We assume that a user's mobility is fully described by its speed and its motion path. In each cell, we assume that there is a finite amount of paths each user can take. Each path has an entry point and an exit point in the cell, where the path is a continuous line joining those points. Paths may have common properties, such as the same entry point, same exit point, or even complete sections.

To select a user's path and speed, we adopt a random model. Any time a users enters a cell it randomly chooses a speed value from a given interval and a path value from a given set (cp. Sec. V). The choices are independent among the users and follow a uniform distribution. Consequently, path and speed may change at any handover.

While moving on a given path in the cell, a user $i \in \mathcal{I}$ reports the channel gain $g_i(t)$ to the base station. According to (1), $g_i(t)$ depends on $|h_i(t)|^2$, representing fast fading, and on $d_i(t)^{-\alpha(x_i(t))}$, which accounts for slow fading. As this slow fading term depends only on the user's location, it is fully determined by the user's path. However, even if multiple users travel on the same paths, it is very likely that their CSI sequences differ due to different speeds and fast fading.

In Sec. V, two maps are used to perform simulations to test the proposed trajectory prediction. The first map is a Manhattan grid. It is an abstract map in which the path-loss exponent α is set arbitrarily to model different shadowing effects in different areas of the map. In the second map, however, this function is not set arbitrarily since $d_i(t)^{-\alpha(x_i(t))}$ is obtained from real measurements in the German city Frankfurt am Main.

IV. Proposed Prediction Framework

We propose to predict the user's next cell by solving a classification problem via supervised learning. To do so, we build classifiers from the user's CSI sequence as it travels through the current cell and from the index of its next cell. While this index is used as a label, the CSI sequence serves as an input vector. Before we describe the real-time prediction during system operation, let us formulate the classification problem.

A. Next Cell Prediction as Classification Problem

A user's trajectory can be represented by a set of cells it travels through. Focusing on the cell an arbitrary user i is currently associated with, we define a user's trajectory using the index p_i of its previous cell and index n_i of its next cell. Furthermore, the trajectory can be associated to CSI which is reported periodically while the user is traversing its current cell. Formally, this CSI sequence can be stated as

$$H_i \triangleq \begin{cases} \emptyset & \text{if } t < t_i^{\text{in}} \ ; \\ (g_i(t_i^{\text{in}}), \ldots, g_i(t)) & \text{if } t_i^{\text{in}} \leq t < t_i^{\text{out}}; \\ (g_i(t_i^{\text{in}}), \ldots, g_i(t_i^{\text{out}})) & \text{if } t \geq t_i^{\text{out}}. \end{cases} \quad (2)$$

Here, CSI is represented as the channel gain $g_i(t)$, t_i^{in} denotes the time when user i enters the current cell, and t_i^{out} is the time when the users leaves this cell.

Given the input data H_i and p_i, we can predict the user's next cell by finding an accurate mapping between the tuple $\langle p_i, H_i \rangle$ to the associated next cell n_i. As there will only be a limited number of possible combinations of adjacent cell indices, this is a classification problem and can be solved by supervised learning methods.

The benefit of using this input information is that it is readily available in current cellular networks. The CSI vector H_i is known at the user and reported to the base station. The previous cell index is known at user and base station after the handover. This simplifies the implementation of our framework either at the base station or at the handset.

In our approach, multiple classifiers are learned. Each classifier corresponds to a possible previous cell and each predicts a next cell out of all possible next cells. Consequently, the number of classifiers is equal to the number of possible previous cells. Compared to using only a single classifier, a learning process using multiple classifiers is faster and the prediction is more accurate. This is because each single classifier only needs to perform a much smaller-scale classification task.

This idea combined with supervised learning is illustrated in Fig. 1. The overall scheme consists of two phases, training

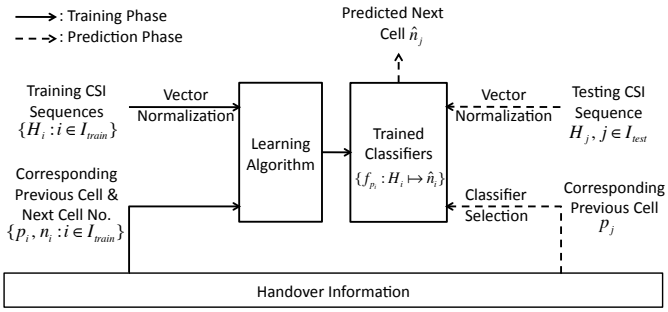


Fig. 1. Concept of the proposed prediction scheme

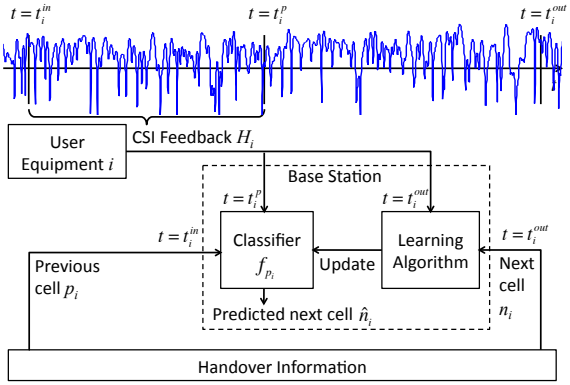


Fig. 2. Real-time system integration of the proposed prediction scheme

phase and prediction phase. In the training phase, the first input to the system is a set of training CSI sequences $\{H_i : i \in \mathcal{I}_{\text{train}}\}$, their associated previous cell $\{p_i\}$ and their next cell indices $\{n_i\}$. Here, p_i specify particular classifiers and $\{n_i\}$ are the corresponding labels that the learning algorithm aims to match the CSI sequences $\{H_i\}$ to. Note that a vector normalization function applies subsampling and interpolation to assure that the CSI sequences are of equal length.

Based on this training data, a learning algorithm derives multiple classifiers $\{f_{p_i}\}$ corresponding to all the possible previous cells. Using this result of the training phase, the prediction phase now selects a specific classifier f_{p_j} to match a previously unknown CSI sequence $H_j, \forall j \in \mathcal{I}_{\text{test}}$, to a predicted index of the next cell $\hat{n}_j$.

B. Classification

We employ a multi-class Support Vector Machine (SVM) as a classifier, using the LIBSVM [10] library.

SVMs [11] seek for the decision boundary between any two classes by constructing a hyperplane in a high-dimensional space such that the hyperplane has the largest distance to the nearest training sample of either class. Although a linear SVM is fast to train, in our case, it does not achieve an accurate early prediction, when feature dimension is small. This situation is common when users traverse the cells faster (e.g., at vehicular speed). To this end, we choose a nonlinear SVM with a Gaussian kernel $K(\mathbf{x}, \mathbf{y}) = \exp\left(-\gamma \|\mathbf{x} - \mathbf{y}\|^2\right)$ that maps sample vectors $\mathbf{x}, \mathbf{y}$ into a higher dimensional space. The Gaussian kernel parameter γ and the SVM slack variable C are optimized with cross-validation on a subset of the training set based on grid search.

C. Real-Time Prediction Scheme

Based on the above classification problem, we now describe a system that predicts the user's next cell in real time during system operation. The main idea is to predict a user's next cell from current classifiers that are learned from all user's traversing the cell. After a user has left the cell, the true next cell index is known and the classifier can be updated. This feedback approach includes more and more users in the training set and increases prediction accuracy over time. By frequently renewing the training set and re-training the classifiers, the prediction is continuously updated during system operation.

We illustrate this online prediction scheme in Fig. 2 for an arbitrary user i. The user enters the current cell at entering time t_i^{in} and exits at time t_i^{out}. Additionally, we define the time when the prediction is performed at t_i^p.

This leads to two time intervals. Interval $[t_i^{\text{in}}, t_i^p]$ defines the duration over which the input data is obtained until the prediction is performed. Interval $[t_i^p, t_i^{\text{out}}]$ represents the time the predictor can look ahead.

When the prediction is performed at t_i^p, the CSI sequence H_i observed during $[t_i^{\text{in}}, t_i^p]$ is used with classifier f_{p_i} to derive the index of the next cell $\hat{n}_i$. After user i exits the current cell at time t_i^{out}, the true next cell index n_i as well as CSI sequence H_i will be fed back to the learning algorithm and thus update the corresponding classifier f_{p_i}.

The smaller t_i^p, the less input data is required and the earlier the prediction can be used by network adaptation algorithms. Performing the adaptation at $t_i^p + \epsilon$ – allowing a very small ϵ for computation time – assures that prediction and adaptation are performed in real time. We will study these timing aspects in the following section.

V. Simulation Results

To obtain tractable results, we first study our system in a simple Manhattan grid scenario. Then, we move to a realistic radio map for the German city Frankfurt am Main. For both scenarios, our results consistently show high performance and fast convergence for our prediction framework.

A. Manhattan Grid Scenario

We design this simple scenario based on quadratic cells, as illustrated in Fig. 3. Each cell is of $75\,\text{m} \times 75\,\text{m}$ size and includes 16 distinct paths. As all cells are assumed to be equal, we can focus on a single cell without loss of generality. This exemplary cell is surrounded by 4 neighboring cells at each cardinal direction. Hence we have four different labels for the users moving through this cell.

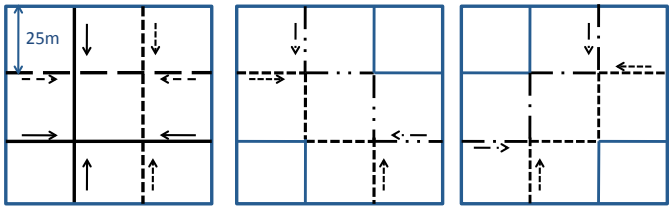


Fig. 3. Illustration of the Manhattan Grid scenario: A quadratic area of $5625\,\mathrm{m}^2$ size

The paths within the studied cell traverse the cell either horizontally, vertically, or diagonally. The users choose the path randomly when they enter the cell and follow the path at a speed, randomly chosen from between 5 m/s and 40 m/s. This random choice follows a uniform distribution and all users make independent choices, as described in Sec. III-C.

To evaluate the accuracy of the proposed solution, we first generate classifiers in the learning phase. To assure accurate learning of the classifiers, we collect 500 data samples per path and use 90% for training and 10% for testing.

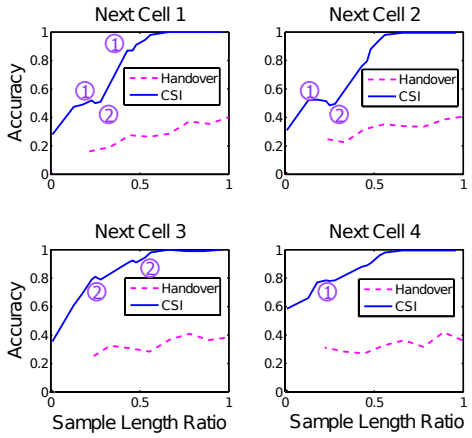
Fig. 4(a) presents the prediction accuracy for each label, i.e., the next cell the user is predicted to go to. The results are shown for a varying sample length ratio, which represents the fraction of input data used from the available data. For our scheme, this is the proportion of CSI values used out of all CSI values that a user reports in the current cell.

We form a baseline by applying SVM only on the user's handover history. Here, the sample length ratio is the length of the used handover history divided by the absolute length of the history. We generate handover history using the random mobility model from Sec. III-C. With this model, all possible next cells have the same probability of being part of a trajectory, if the history length is sufficiently large. We, thus, consider no history length smaller than 2, i.e., a sample length ratio below 0.25.

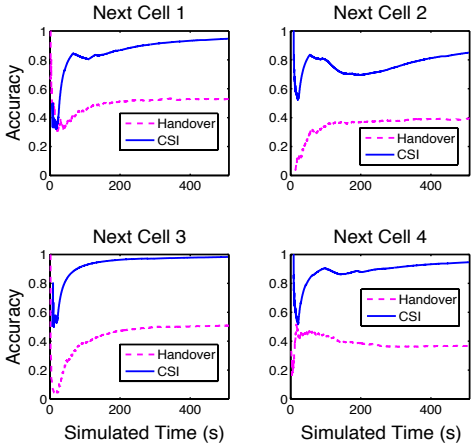
As shown in Fig. 4(a), our method substantially outperforms the handover history-based approach. This illustrates that older handover values do not accurately reflect the decision of the user's current trajectory. Also, our method is shown to profit quickly from more input CSI data, it only requires a small fraction of input data to reach a certain accuracy threshold.

In Fig. 4(a), we notice several intervals with slowly increasing accuracy. We mark those phases by (1) and use (2) to mark phases where the accuracy even decreases for higher sample ratios. Studying the Euclidean distance among the paths explains both effects as a result of the overlapping sections of different paths. As different paths can use the same section of a street, the long-term average of the CSI values for this section is similar. Unless the paths separate, using a large sample length ratio does not increase prediction accuracy.

Nonetheless, the prediction accuracy reaches 1 when the sample length ration exceeds 0.6. This means that once a user travels 60% of its path through the cell, maximum prediction accuracy is reached. With the considered speeds, this allows to predict the next cell at 100% accuracy between 1 s and 20 s



(a) Prediction accuracy versus sample length ratio



(b) Prediction accuracy versus simulated time

Fig. 4. Prediction accuracy of the proposed scheme and the baseline in the Manhattan Grid scenario

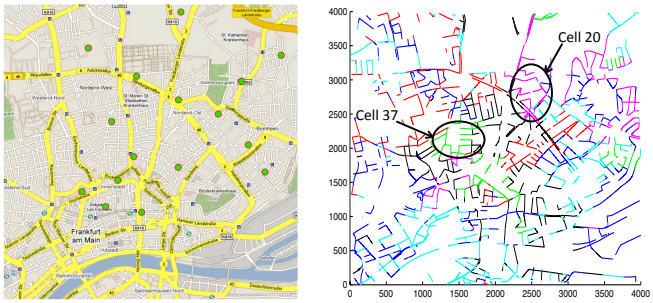
in advance. This provides sufficient time for many adaptation schemes before the user leaves the cell.

Fig. 4(b) shows prediction accuracy per label for varying simulated time. These results where generated for the real-time prediction scheme with users entering the current cell with Poisson distributed inter-arrival times at $\lambda = 1\,\mathrm{s}$. We notice that prediction accuracy converges very fast. With approximately 100 samples per path, our learning scheme obtains stable classifiers. This fast convergence to 100% accuracy shows, again, the advantage of our proposed CSI-based prediction over using handover history alone.

B. Frankfurt Scenario

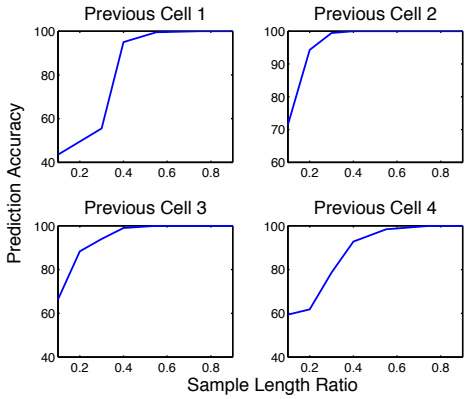
In this scenario, we study an area of $4\,\mathrm{km} \times 4\,\mathrm{km}$ in the downtown area of the German city Frankfurt am Main. A map and the 16 base station locations are shown in Fig. 5(a). As each base station has 3 sectors, drive tests in this area recorded the average channel gain from 48 cells. Using those measurements to parametrize our fast fading model, Sec. III-A provides the channel gain H_i for user $i \in \mathcal{I}$.

This scenario differs from Manhattan grid by its more

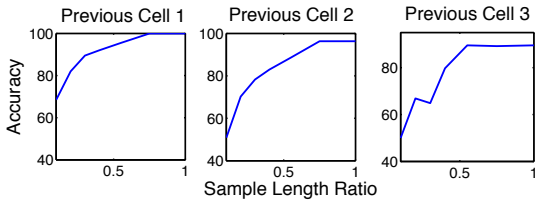


(a) Base station locations (b) Cell distribution and studied cells

Fig. 5. Overview of the Frankfurt scenario: A 16 km^2 area in downtown Frankfurt am Main, Germany



(a) Results for Cell 37



(b) Results for Cell 20

Fig. 6. Prediction accuracy of the proposed scheme in the Frankfurt scenario

complicated cell geometry and trajectories. However, the remaining parts of the system model are not changed. The baseline could not have been studied in this scenario due to insufficient handover data.

We test our predictor for two different cells, which are marked in Fig. 5(b). The first cell with index 37 includes 26 paths. Here, 100% prediction accuracy is reached when more than 70% of the CSI input values are used; cp. Fig. 6(a). With the second cell, number 20, we study a complicated example. This cell contains 46 paths and has a more complex geometry than Cell 37. As shown in Fig. 6(b), the prediction accuracy does not reach 100% even when the complete CSI sequence is used. However, considerable high accuracy is reached when more than 75% of the input is employed. This accuracy of more than 95% should suffice for many practical purposes.

VI. Conclusion

We described a system to predict a user's next cell. Formulating this prediction as a classification problem allowed us to apply Support Vector Machines (SVMs) on the last handover event and a CSI vector. This interesting combination of (i) long-term handover information and (ii) short-term CSI shows promising results for a synthetic Manhattan grid scenario and for a realistic radio map of Downtown Frankfurt. From these simulation results, we conclude that

1) A realistic number of users is sufficient to train the system. In the studied scenarios, 100 users per path sufficed to build accurate classifiers.
2) SVMs predict the next cell substantially more accurately with CSI than with handover history alone. In the studied scenarios, our method more than doubled the accuracy of the classic handover history-based approach.
3) This high accuracy is already reached with a small fraction of the CSI input vector. In the studied scenarios, not more than 60 % of the CSI vector was required to reach 100 % prediction accuracy.

From these observations, we can conclude immediate practical benefits. Using only a small part of the input vector allows early but accurate prediction. Fast training makes the system reactive to changes in the environment. Finally, the prediction accuracy is high under various practical conditions. To this end, the proposed prediction framework is highly promising and deserves further study in realistic scenarios.

Acknowledgment

We thank Lutz Ewe from Bell Labs, Stuttgart for supporting us with the Frankfurt radio map.

References

[1] M. Proebster, M. Kaschub, T. Werthmann, and S. Valentin, "Context-aware resource allocation for cellular wireless networks," *EURASIP Wireless Communications and Networking*, no. 216, Jul. 2012.
[2] E. Elnahrawy, X. Li, and R. P. Martin, "The limits of localization using signal strength: A comparative study," in *Proceedings of the IEEE SECON*, 2004, pp. 406–414.
[3] J. Kim, S. Kim, N. Y. Kim, J. Kang, Y. Kim, and K. T. Nam, "A novel location finding system for 3GPP LTE," in *Proceedings of the IEEE PIMRC*, 2009, pp. 3213–3217.
[4] K. Li, P. Jiang, E. L. Bodanese, and J. Bigham, "Outdoor location estimation using received signal strength feedback," *IEEE Communications Letters*, vol. 16, no. 7, pp. 978–981, 2012.
[5] D. A. Tran and T. Nguyen, "Localization in wireless sensor networks based on support vector machines," *IEEE Trans. on Parallel and Distributed Systems*, vol. 19, pp. 981–994, 2008.
[6] M. Kyriakakos, N. Frangiadakis, and L. Merakos, "Enhanced path prediction for network resource management in wireless LANs," *IEEE Wireless Communications*, vol. 10, pp. 62–69, 2003.
[7] T. Anagnostopoulos, C. B. Anagnostopoulos, S. Hadjiefthymiades, A. Kalousis, and M. Kyriakakos, "Path prediction through data mining," in *Proceedings of the IEEE ICPS*, 2007, pp. 128–135.
[8] 3GPP, "E-UTRA medium access control (MAC) protocol specification," 3GPP, Technical Specification TS 36.321 V 11.0.0, Sep. 2012.
[9] D. Tse and P. Viswanath, *Fundamentals of Wireless Communication*. Cambridge University Press, May 2005.
[10] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *Available at http://www.csie.ntu.edu.tw/cjlin/libsvm*, 2001.
[11] V. Vapnik and C. Cortes, "Support vector networks," *Machine Learning*, vol. 20, pp. 273–297, 1995.