Extracting Context Information from Wi-Fi Captures

Lorenz Schauer
Mobile and Distributed Systems Group
Ludwig-Maximilians-Universität München
Munich, Germany
lorenz.schauer@ifi.lmu.de

Claudia Linnhoff-Popien
Mobile and Distributed Systems Group
Ludwig-Maximilians-Universität München
Munich, Germany
linnhoff@ifi.lmu.de

ABSTRACT

Inferring a user's current situation is the basis of context-aware services. However, users rarely provide access to their sensor data, and hence extracting context information remains challenging in real-world scenarios. In this paper, we present an overall concept for inferring mobility, location, and role information from users based on passively recorded Wi-Fi signals. Several methods are investigated and an extended Viterbi-based approach is presented to determine dwelling and motion periods. This information is used to enhance the mobility model for probabilistic indoor localization. In addition, we compute various features to classify users according to their role. The presented concept is evaluated on simulated data and discussed on real Wi-Fi captures. Our results show, that the proposed Viterbi-based approach performs best for inferring mobility states and can improve the localization accuracy in most instances. Furthermore, it helps to increase the classification performance and indicates strong cluster tendencies in our real-world dataset.

CCS Concepts

•Human-centered computing → Ubiquitous and mobile computing systems and tools;

Keywords

mobility state; localization; tracking; classification; Wi-Fi

1. INTRODUCTION

Nowadays, mobile devices are equipped with several sensors enabling context-aware services being useful for personal and commercial purposes. Beside location, Abowd et al. [1] discovered identity, time and activity as primary context types for characterization of a user's situation. A lot of work exists where sensor data is collected from mobile phones in order to determine the user's position [2] or his/her mobility status [6]. However, they require an active participation which is rarely given in real-world scenarios.

PETRA '17, June 21-23, 2017, Island of Rhodes, Greece

DOI: http://dx.doi.org/10.1145/3056540.3056551

Therefore, we investigate Wi-Fi as a promising way for inferring context information from mobile users without their consent or even awareness. Wi-Fi enabled devices broadcast IEEE 802.11 probe requests to discover access points in reach. These frames contain unencrypted device specific information and simple monitor units suffice to sniff them in an area of interest and in a passive manner. The technique has been already applied to infer context information, e.g., mobility status [22], location [13], or user role [17].

In contrast to these works, we present an overall concept to extract the mentioned primary types of context information out of passive Wi-Fi captures. More precisely, we firstly investigate deterministic and probabilistic approaches to infer the user's mobility state, such as dwelling and motion periods. The result is further processed to enhance the mobility model of our state particle filter, introduced in [20], to receive more accurate position fixes. Subsequently, we identify three feature groups out of raw Wi-Fi signal readings (1), the inferred mobility status (2), and extracted location information (3). For each group, we compute appropriate features and investigate their ability for user role classification. Overall, our main contributions are as follows:

- Investigation of various methods and enhancements of a Viterbi-based approach to infer mobility information.
- Extension to our previously presented state particle filter for estimating user trajectories.
- Presentation of a novel simulation tool for Wi-Fi sniffing providing data with ground truth for evaluation.
- Evaluation of the proposed concept and discussion of obtained results with real Wi-Fi captures.

In summary, the paper is structured as follows: Section 2 reveals related work. Our concept is described in detail within Section 3. Section 4 presents the evaluation and the obtained results which are briefly discussed in Section 5 related to real-world data. Finally, Section 6 concludes the paper and gives hints on future work.

2. RELATED WORK

Inferring motion and location information from Wi-Fi time series has gathered high interest in the last decade. LOCADIO, developed by Krumm and Horvitz [9], is seen as one of the first systems inferring a user's motion state by using Wi-Fi. Unlike us, it only considers the variance of received signal strengths (RSS) from the currently strongest access point and uses a two-state hidden Markov model (HMM)

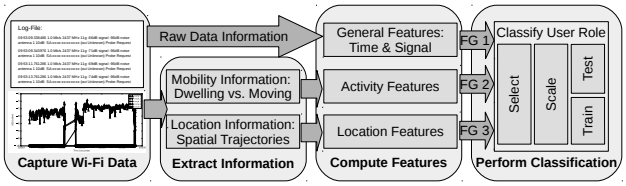


Figure 1: Schematic overview of our concept.

for smoothing. The mobility context is further used for another HMM to infer the user's location. According to the authors, they infer the mobility state with a 87% accuracy. Wind et al. [25] analyzed human mobility by inferring stop locations form Wi-Fi data. They periodically scan for visible access points and determine start and stop times for complete environments. In contrast, Shen et al. [22] considered dwell times for a feature-based room level localization. Their approach supports detecting the correct room in case of indistinguishable Wi-Fi fingerprints. They also determined sharp changes as walking periods in smoothed RSS time-series achieving a 10% improvement to the histogram method. Ruiz-Ruiz et al. [17] also extracted context information from passive Wi-Fi captures at a hospital environment. Beside users' role and locations, they determine different features, such as stationary or moving devices. Quin et al. [16] presented the Mo-Fi system for estimating human presence activities based on Wi-Fi sniffing data. They determined 4 activity patterns and reached a detection rate of 87.4%. Muthukrishnan et al. [14] also investigated several approaches for determining dwell times based on passively recorded Wi-Fi captures and used the extracted information to improve state-of-the art RSS-based localization algorithms. They reached over 90% of precision and recall. However, they do not consider probabilistic methods.

To the best of our knowledge, we present the first overall concept for extracting primary types of context information from Wi-Fi captures. It extends some of the described works, e.g., LOCADIO for inferring mobility states, and introduces novel results for role classification based on different feature groups.

3. CONCEPT

Our concept is illustrated in Figure 1 and consist of the following major steps which are subsequently described:

1. Capture Wi-Fi Data: monitor units listen for IEEE 802.11 probe requests in an area of interest.
2. Extract Information: based on Wi-Fi captures, we infer mobility and location information. Furthermore, time/signal information are extracted from raw data.
3. Compute Features: features are computed representing time/signal, activity and location information.
4. Perform Classification: based on the feature groups (FG) from the previous step, user roles are classified using common machine learning techniques.

3.1 Capture Wi-Fi Data

Wi-Fi, standardized in IEEE 802.11 [5], uses management frames for network discovery which can be performed either actively, or passively. Mobile clients prefer the active mode sending out probe request frames iteratively for each channel. These frames contain unencrypted device specific information, e.g., the MAC address, which can be captured by any Wi-Fi card in range which is set into monitor mode. The technique shows high potentials for user tracking [21].

According to [3], probe requests are sent out every two minutes on average, regardless of the device's connection status. Our own experiments confirm these results. At least for Android devices, we also observed a significant gain of probe request bursts if the device is used and not in idle mode. Furthermore, we always detect probe requests when a mobile device is activated, e.g., powered on, or home button is pressed. These observations are incorporated in our Wi-Fi sniffing simulation tool which is described in Section 4.1.

3.2 Extract Information

As shown in Figure 1, we extract 3 types of information out of captured Wi-Fi data: time/signal, mobility and location information. The former is easy to extract, because time and signal information can be directly read from raw captures. However, raw signal readings include noise and may be corrupted by multi-path propagations which commonly occur in buildings [19]. Therefore, we propose other methods to extract mobility and location information.

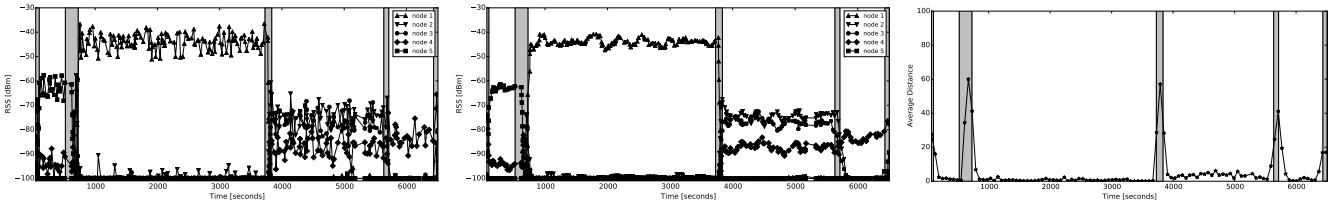
3.2.1 Mobility Information

Theoretically, mobility information could also be inferred by continuous and accurate position fixes. However, this is still a challenging task especially within buildings, due to missing reliable positioning systems for real-world scenarios. Hence, we concentrate on inferring the mobility information from raw Wi-Fi captures, rather than computing motion states on uncertain location estimations.

The goal is to detect sharp changes in signal readings indicating a user's motion, while constant RSS values over a period of time indicate that the user is dwelling [10]. In order to distinguish between both mobility patterns, we firstly smooth the raw RSS readings, due to outlier elimination. For a better imagination, Figure 2 depicts an example where 5 monitor nodes capture Wi-Fi data of a user moving through a building for nearly two hours and stays at 4 different rooms with variable dwell times. The ground truth is illustrated by gray areas denoting the user's movement and by white regions marking the dwelling periods.

In contrast to related work, we use the Savitzky–Golay filter for smoothing, because it keeps best the important features, e.g., distribution of maxima and minima. Other tested filters, such as running mean, or Kalman filter have shown more falsified results in our settings. The effect can be observed by comparing Figure 2a and 2b.

Finally, we use the smoothed RSS time-series and calculate an average distance value in a sliding window approach with window size t_w and step size $s = t_w/2$. Note, higher distance values represent an increased variance within the time-window indicating moving periods. For our example, Figure 2c depicts the obtained distance curve where high peaks are located around real moving periods denoting an adequate result. This allows us to distinguish between dwelling and moving times. It applies, the better the matching of sharp changes with ground truth, the more accurate the classification result. Hence, we consider readings from several Wi-Fi monitor units, rather than considering only the strongest signal, like LOCADIO. As denoted in [7], this should return



(a) Raw data (b) Smoothed data (c) Distances

Figure 2: Wi-Fi captures from 5 nodes observing one user. Gray areas denote real moving periods.

more distinguishable patterns. Overall, we investigate the following distance functions for our purpose:

- Cosine similarity: representing the similarity of 2 non zero vectors by considering the cosine of the angel. This measure was successfully used in our previous work [12] returning a similarity value between 0 (completely different) and 1 (equal). Here, the average of the cosine similarities between all pairs of RSS observation vectors within a time-window are considered.

- Euclidean distance: representing the difference of two n-dimensional vectors p and q by computing the distance $d(p,q) = \sqrt{\sum_{i=1}^{n}(p_i - q_i)^2}$. In contrast to [14], we calculate the average Euclidean distance between all pairs of RSS observation vectors within a time-window, rather than only between the first and last measurements.

- Sum of range: representing the sum of the maximum change of captures from each monitor within a time-window. In comparison to the former functions, we now consider each monitor records separately, rather than computing observation vectors.

- Variance: representing the mean variance over captures from each monitor within a time-window. The variance or similar measures, e.g., mean standard deviation, are commonly used for inferring mobility, like in [9, 23]. Hence, it can be used as benchmark.

Note, that if we have no observation during a certain time-window, which is very common in real-world scenarios, we interpolate the distance by using the distance function on the previous and the last observation. Finally, a binary decision is required whether the user is moving or dwelling.

On the one hand, this can be performed deterministically, like in [14, 22] by using a certain threshold θ. For distances greater than θ, the user is seen as "moving", or marked as "still" otherwise. On the other hand, a probabilistic approach on an underlying movement model could be used to infer more accurate mobility information without requiring any threshold. Therefore, we create an HMM $\lambda = (S; O; A; B; \pi)$ with two hidden states "still" and "moving" $S = \{s, m\}$, like the LOCADIO system [9] or ComPoScan [7]. In contrast, we consider the computed distance for each time-window as our observations O. Furthermore, we assume a higher probability for state changes in the real world and define the transition probability matrix A as:

$$A = \begin{pmatrix} a_{s,s} & a_{s,m} \\ a_{m,s} & a_{m,m} \end{pmatrix} = \begin{pmatrix} 0.99 & 0.01 \\ 0.01 & 0.99 \end{pmatrix}$$

In order to estimate the observation probabilities B, we assume that the distance observations for a mobility state S follow a normal distribution $X_S \sim \mathcal{N}(\mu_S, \sigma_S^2)$. We determine mean μ_S and standard deviation σ_S by observing the distributions of our distance values for both patterns. Obviously, μ_S and σ_S are higher in case of $S = m$ when a user is in motion. Overall, the observation probability $p(\text{obs}\,|S)$ for being in state S observing a distance value d is determined by using the probability density function f_{X_S}:

$$p(\text{obs}\,|S) = f_{X_S}(d; \mu_S, \sigma_S^2) \tag{1}$$

For the initial probability π, we consider $\pi_m = 0.9$, due to the assumption that the user has recently entered a monitor's coverage range, and thus he/she is in motion. Based on this model, we use the Viterbi algorithm and determine the most likely sequence of state changes for a mobile user.

3.2.2 *Location Information*

Location inference from Wi-Fi captures is a huge research field, but still a challenging task, especially for indoor scenarios. In our previous work [20], we have presented a novel particle filter which uses an underling state graph for estimating spatial user trajectories. The idea is to restrict the transition model for particles in order to get more accurate and robust positioning results while keeping the amount of particles low for scalability reasons. In principle, the particles are forced to move along the state graph with the freedom to estimate a position between the deployed state nodes. Please consider [20] for more details. However, the drawback of our so-called state particle filter is a limited solution space depending on the state graph which may lead to a lower accuracy for dwelling periods.

Hence, the novel idea is to reduce this problem by integrating the mobility information from the previous step. While the user is estimated as "moving", the state particle filter operates like before. However, if the user is denoted as "still", we use the model for free space movement according to Widyawan et al. [8], and adapt the velocity between 0 and 1 m/h. This renders small positioning corrections possible while the user is dwelling. Note, that adequate mobility information is required for every prediction step which is rarely given for passive Wi-Fi captures. Therefore, we interpolate between the estimated states, and use a small timespan for switching between both transition models.

3.3 Compute Features

We compute a set of statistical features for each information category, i.e., time/signal, activity, and location. The goal is to represent the information of each group for user role classification. According to [11], we consider maximum,

minimum, mean, median, standard deviation, entropy, root mean square error, and the 75th percentile as representative features for each group. Furthermore, we compute the following group specific features:

- General features (time/signal): we determine the time of both first and last observations. Furthermore, the total time of observations, the average of observations per second, and the percentage of non-observations for the total time are considered. Overall, 49 features are computed including the features from above.
- Activity features: out of the inferred mobility information, we extract the time spans for dwelling and motion periods resulting in 18 features for this group.
- Location features: we extract the covered distance from each estimated user trajectory. Based on this, we consider the amount of position fixes and the total length of the walking distance as features resulting in 10 features for this group.

With the proposed feature groups we are able to consider individual and combinations of features for classification. Totally, an amount of 77 features can be used.

3.4 Perform Classification

In order to obtain reasonable classification results and to avoid overfitting effects we investigate 3 approved methods for feature selection: variance threshold, k-best selection using chi-squared statistic, and the principal component analysis (PCA). The latter is the most promising method. Basically, it performs a decomposition of multivariate datasets into a set of successive orthogonal components representing the maximal variance.

Before executing the classification task, the selected features have to be scaled transforming them to standard normally distributed data with zero mean and unit variance. This is a common requirement for many machine learning estimators [15]. The selection of an adequate estimator for our purpose is investigated during evaluation.

4. EVALUATION

We evaluate our concept on simulated data. Therefore, we firstly introduce our simulation tool for generating realistic Wi-Fi observations with a known ground truth.

4.1 Simulation of Wi-Fi Captures

Our simulation tool consist of 3 major models: environment, mobility, and a probe request transmission model.

4.1.1 Environment Model

Like in [18], we use a 2-d bitmap representation of a downloaded building plan[1]. This image is further processed, removing doors, room labels, signs, etc. Afterwards, the image is converted into a binary bitmap, where walkable and non-walkable regions are represented by white and black pixels, respectively.

4.1.2 Mobility Model

We take the modified pathway mobility model form our previous work [18] in order to simulate the behavior of people for indoor scenarios: enter a building through an entrance door, walking to one or more rooms, leaving the building through an entrance door. The underlying assumption is, that a person always try to walk directly to a certain room to execute a particular objective. Hence, we simulate stay times, when the user arrives at a particular room.

4.1.3 Transmission Model

Probe requests are sent out in irregular intervals. Hence, we firstly determine the transmission probability p_{tx} for a certain point in time. On the basis of our observations, we determine 3 key factors with a significant impact on p_{tx}:

1. Activity: as mentioned in Section 3.1, probes are sent out, when the user starts an activity, and hence, this has to be considered separately.
2. Time: as mentioned before, probes are sent out every two minutes on average. Furthermore, the probability for a user's activity increases with time.
3. Device: the probing interval depends on the used device. E.g, Goodall [4] estimated an interval between 18 and 82 seconds for various phones. Additionally, we measured a doubled frequency for an iOS compared to an Android phone in normal usage pattern.

With respect to these factors, we determine p_{tx} on the basis of p_{activity}, the probability that a user's activity occurs, and p_{device}, the probability of device-specific transmissions:

$$p_{\text{activity}} = \text{time}_{\text{lastActivity}} \cdot c_{\text{activity}} \tag{2}$$

$$p_{\text{device}} = F_X(\text{time}_{\text{lastProbe}}) \tag{3}$$

where $\text{time}_{\text{lastActivity}}$ denotes the elapsed time since the last activity, and $\text{time}_{\text{lastProbe}}$ to the last probe transmission, respectively. Obviously, p_{activity} increases with time, and c_{activity} is a constant value determining an upper threshold for activities. In Equation 3, $F_X(\text{time}_{\text{lastProbe}})$ denotes the cumulative density function (cdf) of $X \sim \mathcal{N}(\mu, \sigma^2)$. This represents a normal distributed probing behavior of a certain device X with average time interval μ and its standard deviation σ which can be empirically determined. Due to [4], we use $\mu = 65$ seconds and $\sigma = 20$ seconds, and add a device specific value to respect the varying probing behavior.

At each simulation step, we determine $\text{time}_{\text{lastActivity}}$ and $\text{time}_{\text{lastProbe}}$ and decide if a probe will be sent with respect to p_{activity} or p_{device}. In case of a positive decision, we simulate the transmission using the standard logarithmic path loss model with wall attenuation factor WAF:

$$P_{\text{rx}}(d) = A - 10n \log_{10}(d) - l \cdot \text{WAF}$$

where A is the transmission power, n the path loss exponent, and l represents the amount of walls in the distance d between sender and receiver. The parameters have been determined empirically in [20]. Due to erroneous Wi-Fi transmissions, we decide with a probability of 0.75 that a package is successfully captured. In this case, each monitor saves the corresponding timestamp, the RSS value, the ground truth position and the device identifier. This information is aggregated to one observation vector and the sequence of corresponding vectors represents one user for the given time in the given environment. This is the same procedure like in real scenarios leading to comparable data.

[1] http://www.uni-muenchen.de/funktionen/gebaeudeplaene/7070_d_00.pdf

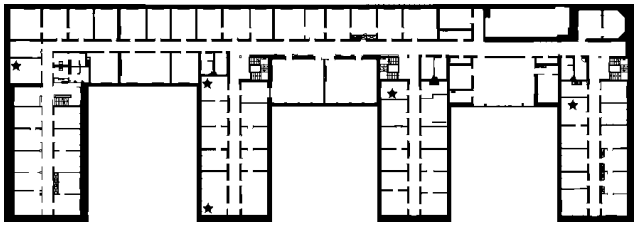


Figure 3: As setup, 5 Wi-Fi nodes (black stars) monitor the depicted section of our university building.

4.2 Experimental Setup

According to [20], we deploy 5 monitors within our university building capturing probe requests on a typical weekday. This dataset will be consulted for discussion in Section 5. Figure 3 depicts the setup on the used building plan, represented as 2-d bitmap. For evaluation, the same setup is created by our simulation tool and 480 users are created based on 4 typical user role patterns (120 users per pattern):

1. Worker: is working in the building. Arrives between 8:00 and 10:00 am, goes to the office, changes rooms several times during the day, and finally leaves the building between 5:00 and 7:00 pm.

2. Student: goes to lectures. Arrives between 10:00 am and 4:00 pm, goes to a lecture room, stays their for the lecture (80 to 120 minutes), and finally leaves and may come back later. Note, this group simulates users with little dwelling periods of long durations.

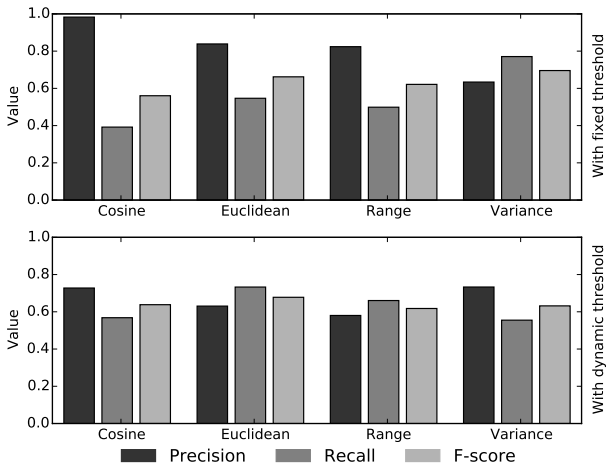
3. Cleaner: has to clean all rooms in the morning. Arrives between 7:00 and 12:00 am, goes to every room in the building and stays there for a short period of time (1 to 5 minutes). Note, this group simulates users with a lot of dwelling and moving periods of short durations.

4. Random: behaves randomly. Arrives between 7:00 am and 6:00 pm, goes to several rooms and stay there between 1 and 120 minutes. Note, this group simulates noisy data which is very common in real scenarios.

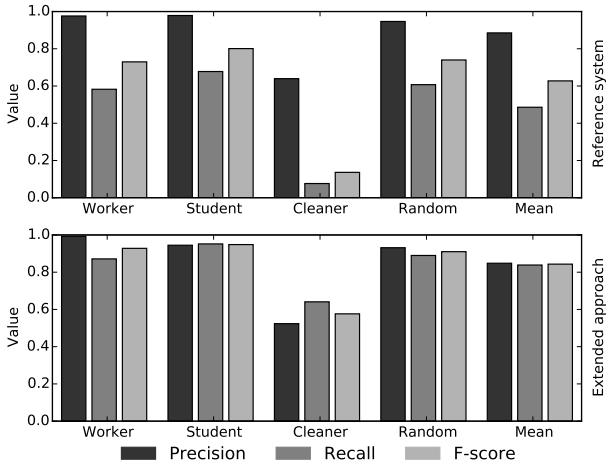
4.3 Inferring Mobility Information

Based on our experimental setup, we investigate the proposed methods of Section 3.2.1 for mobility inference. Hence, we compute the presented distance functions on our simulation data with $t_w = 90$ seconds. We assume, this time is long enough to cover at least one probe on average for all phones, and short enough to detect brief movements.

As a first step, we evaluate the accuracy related to ground truth when performing deterministic decisions. For this purpose, we empirically determine a fixed threshold θ for each distance function. Note, this may influence the overall classification result and is not generalizable. Therefore, we also determine a dynamic threshold θ_{var} by computing the average of all local maxima in the distance curve. For both fixed and dynamic thresholds we calculate precision, recall and F-score as well-adopted metrics for classification tasks. More precisely, we calculate the unweighted mean for each label, i.e., still and moving, in order to ignore label imbalance. This prevents biased results, especially in case of long dwelling and short moving periods or vice versa.



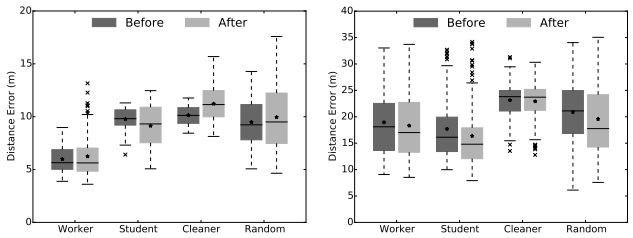
(a) Deterministic decisions



(b) Probabilistic decisions

Figure 4: Evaluation of different methods for mobility inference using precision, recall, and F-score.

Figure 4a depicts the results for our investigations. Obviously, the metric of precision (considering the amount of selected relevant items) is over 0.8 in case of a fixed threshold, except for the variance distance. In contrast, the values for the recall metric are very low (under 0.58) denoting that more false negatives than false positives decisions are made on average. Only in case of the variance distance, the chosen threshold seems more suitable returning the highest F-score (harmonic mean of precision and recall). However, the results demonstrate that a fixed threshold is not feasible for extracting accurate mobility information in general. In contrast, the results based on a dynamic threshold, as shown below in Figure 4a, depict more balanced values for precision and recall. In terms of the F-score metric, we observe comparable results related to the former with fixed threshold lying over 60% for each distance function. The best value of 68% is observed in case of the Euclidean distance which is not a reliable result. Hence, we conclude that our deterministic methods (even with dynamic threshold) are not suitable for our purpose. The reason is, that they are too inflexible



(a) Pairwise distance (b) Fréchet distance

Figure 5: Distributions of location errors before and after integrating mobility information.

to deal with a wide range of various user behaviors.

Next, we evaluate our probabilistic approach. We use the Euclidean distance for our observations O, as it performed best in the previous investigations. As reference system, we implement LOCADIO, described in [9], which differs to our approach in terms of O, A, B, and π. Figure 4b depicts the results for both systems in case of the presented user role patterns and the overall mean which is comparable to the Euclidean distance function of Figure 4a. Obviously, our approach performs more accurate and more balanced in terms of precision and recall for all patterns. Beside the cleaner behavior which models a high frequency of mobility state changes, we reach a high F-score of more than 0.91 (even for random) and a mean F-score of 0.84. This indicates a suitable performance in general. In comparison, our reference implementation has major difficulties with the cleaner behavior. For the others it achieves an F-score of more than 0.73, and 0.63 for the mean which is slightly worse than the value reached by our dynamic threshold. Only in case of the student class we achieve a more suitable value of nearly 80% which comes close to the result shown by Krumm et al. However, our reference system do not show a reliable classification performance in general, and hence, we use our approach for the following steps. Note, that the increased performance of 21% is mainly caused by the usage of the Euclidean distance over 5 monitored signals instead of using just the variance of the strongest signal.

4.4 Inferring Location Information

According to our previous work [20], we use the average pairwise and the discrete Fréchet distance to investigate the localization error of the state particle filter (sPF) before and after integrating the inferred mobility information (cf. Section 3.2.2). The former considers the mean error over pairwise Euclidean distances between estimated and ground truth positions. The latter is a common distance measure for trajectories and can be seen as the minimum length of a leash between a man and his dog while both walking on their paths without going backwards.

Figure 5 depicts the localization error distributions for each user pattern. In case of the pairwise distance (cf. Figure 4a), we observe a correlation between the accuracy of inferred mobility states and the improvement of our sPF, e.g., the student group performs more accurate while the random and cleaner group perform even worse. This indicates that additional mobility information could improve the underlying movement model, but only if the information is reliable, e.g., more than 0.92 for F-score in our case.

Table 1: Classification performance on all features.

Features		Classifiers				
FG	#	D. Tree	R. Forest	SVM	MLP	Vote
1	49	0.95	0.98	0.97	0.97	0.97
2	18	0.87	0.93	0.85	0.88	0.90
3	10	0.66	0.73	0.73	0.72	0.74
1+2	67	0.95	0.99	0.97	0.97	0.98
1+3	59	0.95	0.99	0.97	0.97	0.97
2+3	28	0.85	0.92	0.88	0.89	0.90
All	77	0.94	0.99	0.97	0.97	0.98

In case of the discrete Fréchet distance (cf. Figure 5b) we observe an improvement for all cases, except for the cleaner class. However, the differences are only marginal (between 1 and 1.5 meters for the mean) which indicates that the original movement model already fits the requirements for general scenarios. Overall, we make oppositional observations: the pairwise distance shows increased and the Fréchet distance shows decreased error distributions for most of the cases. This is due to the fact, that the Fréchet distance considers complete trajectories, rather than single position fixes. On the one hand, the reduced speed of particles indicates a positive effect on the overall path error. On the other hand, however, our extension leads to increased errors for several position fixes, in particular during the transition of the user's mobility state, or when the estimated mobility state differs from ground truth. In summary, we use the extended approach in the sequel for user role classification, due to the higher accuracy for trajectory estimations.

4.5 User Role Classification

The goal is to assign the correct user role label, i.e., worker, student, cleaner, or random, to each of our 480 simulated samples using the extracted feature groups (FG) or their combinations. Thus, we perform a ten-fold stratified cross-validation, which should reduce overfitting, as stated in [24].

First, we analyze the classification performance without feature selection. Training and testing are performed on scaled features using well-adopted classifiers provided by [15], such as: decision tree, random forest with 100 estimators, support vector machine (SVM), multi-layer perceptron (MLP), and a voting classifier which uses a majority vote over the combination of the named classifiers. Table 1 depicts the results where the best values for each classifier are highlighted in gray. Overall, we observe high values (> 0.93) for all classifiers when general features (FG 1) are involved. In contrast, activity (FG 2) or location (FG 3) features perform worse. This indicates that time/signal features include the most significant information for our user role patterns. We will discuss this point for real captures later in Section 5. With additional information from activity and/or location features, we could marginally improve the classification performance in case of random forest and voting. Due to the fact that our simulations consist of only 4 well-defined user role patterns, we still observe an overfitting effect, expressed by very high success rates, e.g., 99% for random forest.

In order to reduce this effect and to improve the generalization of our models, we now select a subset of significant features before analyzing the classification performance. Therefore, we perform feature selection, as described in Section 3.4 and repeat our last test. The PCA method

Table 2: Classification performance on selected features based on PCA with k=10.

Features		Classifiers				
FG	#	D. Tree	R. Forest	SVM	MLP	Vote
1	10	0.88	0.94	0.94	0.94	0.94
2	10	0.87	0.90	0.85	0.88	0.90
3	10	0.68	0.73	0.75	0.73	0.75
1+2	20	0.92	0.96	0.96	0.97	0.97
1+3	20	0.85	0.94	0.94	0.94	0.95
2+3	20	0.84	0.90	0.84	0.88	0.88
All	30	0.87	0.96	0.96	0.96	0.97

indicates the most promising results which are exemplarily depicted in Table 2. We chose $k = 10$, in order to balance the amount of features for each group. As expected, we observe lower success rates for nearly all combinations. Furthermore, FG 1 is still involved in the most accurate results (> 0.93), and hence we conclude that time/signal information is essential for user role classification. An interesting observation is that the combination of FG 1 and 2 performs best for all classifiers indicating that the inferred activity information can help to identify the correct user role more precisely. In contrast, the location features still return low success rates and could not improve the obtained results in any case, except for the voting classifier. This may be caused by our simulations and the simple structure of our building leading to similar trajectories for all user groups. Hence, we will also discuss this observation with unlabeled data from real Wi-Fi captures in the following section.

5. DISCUSSION

For discussion, we use our proposed methods and extract the same feature groups out of the real dataset. Due to missing ground truth, we have to deal with unlabeled data using methods from unsupervised learning, such as manifold learning, or clustering. As a first step, we visualize the n-dimensional features from all groups for both datasets using t-distributed Stochastic Neighbor Embedding (t-SNE), also provided by [15]. Figure 6 depicts the results for 2-d embedded features. Due to missing labels, we can only guess the user role for each data point. However, we can analyze the structure of both datasets showing interesting properties.

In case of general features (cf. first row), our simulations show more biased data. This is approved by a higher Hopkins statistic[2] of $H_{\text{sim}} = 0.66$ to $H_{real} = 0.64$ and confirms the assumption, that 4 well-defined user role patterns do not cover the variety of human behavior in real life.

For activity features (cf. second row), in contrast, we observe the highest cluster tendencies in both datasets, i.e., $H_{\text{sim}} = 0.73$, and $H_{real} = 0.81$. Especially for real captures this is interesting and should be further investigated. For the moment, we argue that dwelling and moving periods are suitable to characterize user behaviors which corresponds to our previous findings.

For location features (cf. last row), we observe the lowest cluster tendency for our simulations with $H_{\text{sim}} = 0.60$. This explains the decreased classification performance and confirms our assumption that the simulated trajectories are too similar to distinguish properly between different user roles. However, in case of real Wi-Fi captures, we observe a higher value for location than for general features, i.e., $H_{real} = 0.67$. Hence, we expect that the proposed method helps to increase the classification performance within real-world settings, which has to be proven by future work.

[2]Measures cluster tendencies in datasets by comparing nearest neighbor distances, from $H = 0$: poor, to $H = 1$: strong.

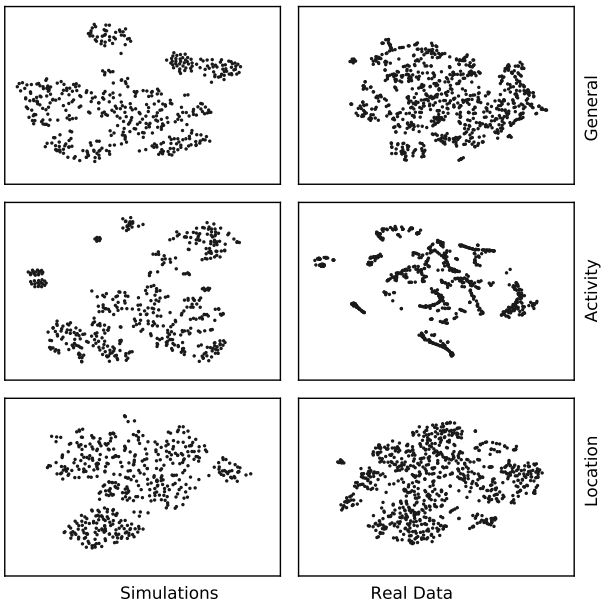


Figure 6: Comparison between simulated and real data according to the proposed feature groups.

In summary, both datasets show similar compositions indicating the feasibility of our simulation tool. Furthermore, the extracted feature groups, in particular the activity features, seem to be appropriate to distinguish between different user groups also within real-world scenarios. However, it has also been proven that the performed simulations do not represent the complete variety of real human behavior and further investigations have to be made in the near future.

6. CONCLUSION

With this paper, we have proposed an overall concept for extracting primary types of context information out of Wi-Fi captures. Novel methods for inferring the mobility state and the location of a user have been presented. Our evaluation on simulated data has shown that a certain threshold is not feasible for mobility inference in general. It has proven useful to perform a Viterbi-based approach using the Euclidean distance on several monitored signals leading to an improvement of up to 21% related to our reference system.

For location inference, we have shown that reliable mobility information can improve the movement model of our state particle filter. In case of user role classification, we state that time/signal features are essential, and a combination with selected activity features helps to increase the success rate of up to 3%.

Finally, we conclude that our concept seems to be suitable in real-world settings where activity and location features have shown strong cluster tendencies. Hence, we will enhance our efforts in the near future using a large set of labeled data from real captures to confirm this assumption.

7. REFERENCES

[1] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, and P. Steggles. Towards a better understanding of context and context-awareness. In *International Symposium on Handheld and Ubiquitous Computing*, pages 304–307. Springer, 1999.

[2] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 261–272. ACM, 2009.

[3] B. Bonné, A. Barzan, P. Quax, and W. Lamotte. Wifipi: Involuntary tracking of visitors at mass events. In *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2013 IEEE 14th International Symposium and Workshops on a*. IEEE, 2013.

[4] N. J. Goodall. Fundamental characteristics of wi-fi and wireless local area network re-identification for transportation. *IET Intelligent Transport Systems*, 2016.

[5] IEEE Computer Society, 3 Park Avenue, NY 10016-5997, USA. *IEEE Std 802.11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, June 2007.

[6] Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In *Proceedings of the 2012 ACM Conference on Ubiquitous Computing*, pages 441–450. ACM, 2012.

[7] T. King and M. B. Kjærgaard. Composcan: adaptive scanning for efficient concurrent communications and positioning with 802.11. In *Proceedings of the 6th international conference on Mobile systems, applications, and services*, pages 67–80. ACM, 2008.

[8] M. Klepal, S. Beauregard, et al. A novel backtracking particle filter for pattern matching indoor localization. In *Proceedings of the first ACM international workshop on Mobile entity localization and tracking in GPS-less environments*, pages 79–84. ACM, 2008.

[9] J. Krumm and E. Horvitz. Locadio: Inferring motion and location from wi-fi signal strengths. In *Mobiquitous*, pages 4–13, 2004.

[10] D. L. Lee and Q. Chen. A model-based wifi localization method. In *Proceedings of the 2nd international conference on Scalable information systems*, page 40. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2007.

[11] M. Maier and F. Dorfmeister. Fine-grained activity recognition of pedestrians travelling by subway. In *International Conference on Mobile Computing, Applications, and Services*, pages 122–139. Springer, 2013.

[12] M. Maier, L. Schauer, and F. Dorfmeister. Probetags: Privacy-preserving proximity detection using wi-fi management frames. In *Wireless and Mobile Computing, Networking and Communications (WiMob), 2015 IEEE 11th International Conference on*, pages 756–763. IEEE, 2015.

[13] A. Musa and J. Eriksson. Tracking unmodified smartphones using wi-fi monitors. In *Proceedings of the 10th ACM conference on embedded network sensor systems*, pages 281–294. ACM, 2012.

[14] K. Muthukrishnan, B. J. van der Zwaag, and P. Havinga. Inferring motion and location using wlan rssi. In *Mobile Entity Localization and Tracking in GPS-less Environnments*. Springer, 2009.

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

[16] W. Qin, J. Zhang, B. Li, and L. Sun. Discovering human presence activities with smartphones using nonintrusive wi-fi sniffer sensors: the big data prospective. *International Journal of Distributed Sensor Networks*, 2013, 2013.

[17] A. J. Ruiz-Ruiz, H. Blunck, T. S. Prentow, A. Stisen, and M. B. Kjærgaard. Analysis methods for extracting knowledge from large-scale wifi monitoring to inform building facility planning. In *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*, pages 130–138. IEEE, 2014.

[18] L. Schauer. Discovering hotspots: A placement strategy for wi-fi based trajectory monitoring within buildings. In *SAI Intelligent Systems Conference (IntelliSys), 2015*, pages 371–380. IEEE, 2015.

[19] L. Schauer, F. Dorfmeister, and M. Maier. Potentials and limitations of wifi-positioning using time-of-flight. In *Indoor Positioning and Indoor Navigation (IPIN), 2013 International Conference on*. IEEE, 2013.

[20] L. Schauer, P. Marcus, and C. Linnhoff-Popien. Towards feasible wi-fi based indoor tracking systems using probabilistic methods. In *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*. IEEE, 2016.

[21] L. Schauer, M. Werner, and P. Marcus. Estimating crowd densities and pedestrian flows using wi-fi and bluetooth. In *Proceedings of the 11th International Conference on Mobile and Ubiquitous Systems: Computing, Networking and Services*, pages 171–177. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering), 2014.

[22] J. Shen, J. Cao, X. Liu, J. Wen, and Y. Chen. Feature-based room-level localization of unmodified smartphones. In *Smart City 360Âř*, pages 125–136. Springer, 2016.

[23] T. Sohn, A. Varshavsky, A. LaMarca, M. Y. Chen, T. Choudhury, I. Smith, S. Consolvo, J. Hightower, W. G. Griswold, and E. De Lara. Mobility detection using everyday gsm traces. In *International Conference on Ubiquitous Computing*, pages 212–224. Springer, 2006.

[24] M. Werner, C. Hahn, and L. Schauer. Deepmovips: Visual indoor positioning using transfer learning. In *Indoor Positioning and Indoor Navigation (IPIN), 2016 International Conference on*. IEEE, 2016.

[25] D. K. Wind, P. Sapiezynski, M. A. Furman, and S. Lehmann. Inferring stop-locations from wifi. *PloS one*, 11(2):e0149105, 2016.