Proceedings of the Twenty-Ninth AAAI Conference on Innovative Applications (IAAI-17))

Real-Time Indoor Localization in Smart Homes Using Semi-Supervised Learning

Negar Ghourchian
McGill University
negar@cs.mcgill.ca

Michel Allegue-Martinez
Aerîal Technologies Inc./Tandemlaunch
michel.allegue@aerial.ai

Doina Precup
McGill University
dprecup@cs.mcgill.ca

Abstract

Long-term automated monitoring of residential or small industrial properties is an important task within the broader scope of human activity recognition. We present a device-free wifi-based localization system for smart indoor spaces, developed in a collaboration between McGill University and Aerîal Technologies. The system relies on existing wifi network signals and semi-supervised learning, in order to automatically detect entrance into a residential unit, and track the location of a moving subject within the sensing area. The implemented real-time monitoring platform works by detecting changes in the characteristics of the wifi signals collected via existing off-the-shelf wifi-enabled devices in the environment. This platform has been deployed in several apartments in the Montreal area, and the results obtained show the potential of this technology to turn any regular home with an existing wifi network into a smart home equipped with intruder alarm and room-level location detector. The machine learning component has been devised so as to minimize the need for user annotation and overcome temporal instabilities in the input signals. We use a semi-supervised learning framework which works in two phases. First, we build a base learner for mapping wifi signals to different physical locations in the environment from a small amount of labeled data; during its lifetime, the learner automatically re-trains when the uncertainty level rises significantly, without the need for further supervision. This paper describes the technical and practical issues arising in the design and implementation of such a system for real residential units, and illustrates its performance during on-going deployment.

Problem Description

Localization is an essential function of a smart indoor environment, as it enables discovering knowledge about the behaviour and preferences of residents, especially those who need long-term monitoring or care. Location-aware applications include surveillance and security, health and sleep monitoring, assisted living for elderly people or patients with disabilities and entertainment. For example, knowing the location of an elderly person who lives alone enables offsite caregivers to observe daily routines in order to detect hazardous events such as long stays in bed or in an unusual area (e.g., shower or bathroom).

The majority of research in indoor localization has been focused on device-based systems, in which the location of a moving subject within the space is determined and represented by an associated device such as a wifi-enabled smart phones or radio-frequency identification (RFID) tags (Azizyan, Constandache, and Roy Choudhury 2009; Saab and Nakad 2011). These technologies are usually accurate and reliable, but require the user to wear the device on their body. Existing commercial solutions also include surveillance cameras. This leads to practical problems such as the need for cooperation from subjects, privacy concerns, and high implementation and maintenance cost. For instance, in assisted living for elderly people or patients, carrying/wearing an external device 24/7 is uncomfortable and infeasible. We focus instead on an emerging research area that focuses on localization through device-free passive (DFP) approaches. This does not require users to carry any devices or participate actively in the positioning process, which is an attractive property from the point of view of deployment. Most of the DFP localization systems adopt Radio-frequency sensing infrastructures (such as wifi, RFID, microwave, FM signals, etc.) and rely on the influence of the human body presence and movement on these signals, e.g. through reflection (Popleteev 2013; Pirzada et al. 2014; Kotaru et al. 2015).

In this paper, we present a novel technology that utilizes only off-the-shelf wifi-enabled devices such as access points, laptops, smart TV, for passive sensing in the environment of interest. This design is mainly enabled by wireless technology improvements (introduced in the IEEE 802.11n and IEEE 802.11ac standards) and motivated by the fact that wifi signals are pervasive at home, work and even in public places. The idea is to create a smart sensing environment by using off-the-shelf devices and monitoring the distortions in the strength and patterns of signals between two nodes of communication (transmitter and receiver). State-of-the-art studies suggest that information gleaned from the physical layer in the wireless infrastructures (e.g. wifi signals), such as channel state information (CSI) and received signal strength indicator (RSSI) values, have the potential to characterize the environment, which includes human movements and their locations (Yang, Zhou, and Liu 2013; Azizyan, Constandache, and Roy Choudhury 2009). A wide range of DFP localization systems use RSSI measurements as their base modality, due to simplicity and low hardware requirements. However, RSSI values are coarse-grained and do not exploit the frequency diversity of wide-band wifi channels. Instead, we work with CSI values, which contain fine-grained information that describes how the wifi signal propagates from the transmitter to the receiver and provide richer frequency content (Yang, Zhou, and Liu 2013).

Finding a mathematical characterization of the disturbance created by human motion in CSI signals is a challenging problem due to the complexity of the wireless signal propagation in indoor environments. Therefore, the first challenge of designing an accurate CSI-based localization system is to characterize statistically the correlation between the location of motion events and the CSI values; this can be thought of as a supervised learning problem. However, wifi signal components are sensitive to many internal and external factors such as multipath interference, building attenuation, device and antenna orientation issues, changes in the environment (such as changing the position of objects) and signal interference. This temporal instability and high variance of the raw measurements introduces problems even when trying to predict signal strength between two stationary devices in a motion-free environment. This is a major challenge, which means that any fixed predictor will degrade in performance over time or at certain instants. This problem is an instance of concept drift, because the distribution of input features changes over time in unforeseen ways, strongly affecting the mapping from the input data to the target concept.

A few other studies have employed the same data type to perform active or passive localization indoors. For instance, PinLoc (Sen et al. 2012) and DeepFi (Wang et al. 2015) are recent systems developed base on the CSI data to detect the location of an active mobile device at meter level resolution. Other examples are Pilot (Xiao et al. 2013) and SpotFi (Kotaru et al. 2015), which are passive localization systems, and use multiple pairs of transmitter-receiver to determine the location of a static target. All of these studies have been implemented and evaluated in controlled environments, such as a university laboratory or classroom, with a large volume of human annotated data and using predefined scenarios and predetermined floor plans. We believe that our work provides the first deployment of device-free CSI-based DFP systems in a relatively large-scale manner to typical environments, in which the system is used "in the wild" for continual monitoring . Moreover, to the best of our knowledge, none of the existing CSI-based systems have mentioned or addressed the issue of the drift in the signals, which only appears in long-term evaluations.

Our platform utilizes only one pair of off-the-shelf wifi-compatible devices, which can be placed inside a dwelling (of surface area up to 900 ft^2) to perform room-level localization of a moving target. In the setup phase, we need a very small amount of labeled data for training. In order to address the practical challenges of CSI-based indoor localization, we developed a semi-supervised learning approach which:

- applies various data mining techniques to create a base supervised learner that can predict location of movement from CSI values
- leverages the probabilistic output of the base learner to improve real-time localization
- employs a change-point detection algorithm to detect structural changes in the feature space distribution, i.e. drifts, over the streaming data
- automatically updates the decision boundaries of the base learner in order to maintain high performance localization in spite of signal non-stationarity.

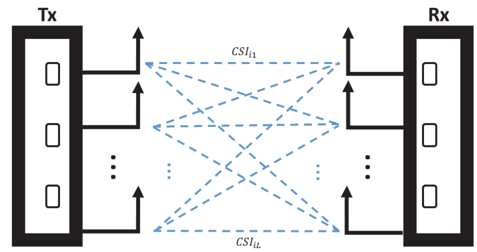

Figure 1: Schematic representation of signal propagation between Tx-Rx antennae

In the remainder of this paper, we present in more detail CSI data (in hopes that other AI researchers would also consider working with it), we describe the structure of proposed application of machine learning to create a robust localization system, we provide technical details about the implementation and development of the deployed application, and summarize the results obtained "in the wild", and finally we discuss the lessons learned and conclude.

Data Description

Channel State Information

In modern wireless communications, a wifi signal propagates between a transmitter (Tx) and receiver (Rx) through multiple transmission channels using Orthogonal Frequency Division Multiplexing (OFDM); this means that within each channel, the transmitter broadcasts simultaneously on several narrowly separated sub-carriers at different frequencies, in order to increase the data rate. Channel state information (CSI), which can be obtained at the receiver, describes how the transmitted signal is propagated through the channel and reveals channel variations and signal distortions experienced during propagation caused by, e.g. scattering, fading and power decay with distance. The quantitative analysis of signal propagation behavior within a wifi-covered area can identify and measure different types of disturbances, including human motion; we will use it to identify the location of a subject.

For each pair of Tx-Rx, let $\ell \in \{1, \cdots, L\}$ denote the antenna link, and $CSI_{i\ell}(t)$ denote a complex number describing the signal received at subcarrier $i \in \{1, \cdots, I\}$ at time t, which is defined by:

$$CSI_{i\ell} = |CSI_{i\ell}| \exp\{j \sin \angle CSI_{i\ell}\} \quad (1)$$

where, $|CSI_{i\ell}|$ and $\angle CSI_{i\ell}$ denote the amplitude response and the phase response of subcarrier i of link ℓ, respectively.

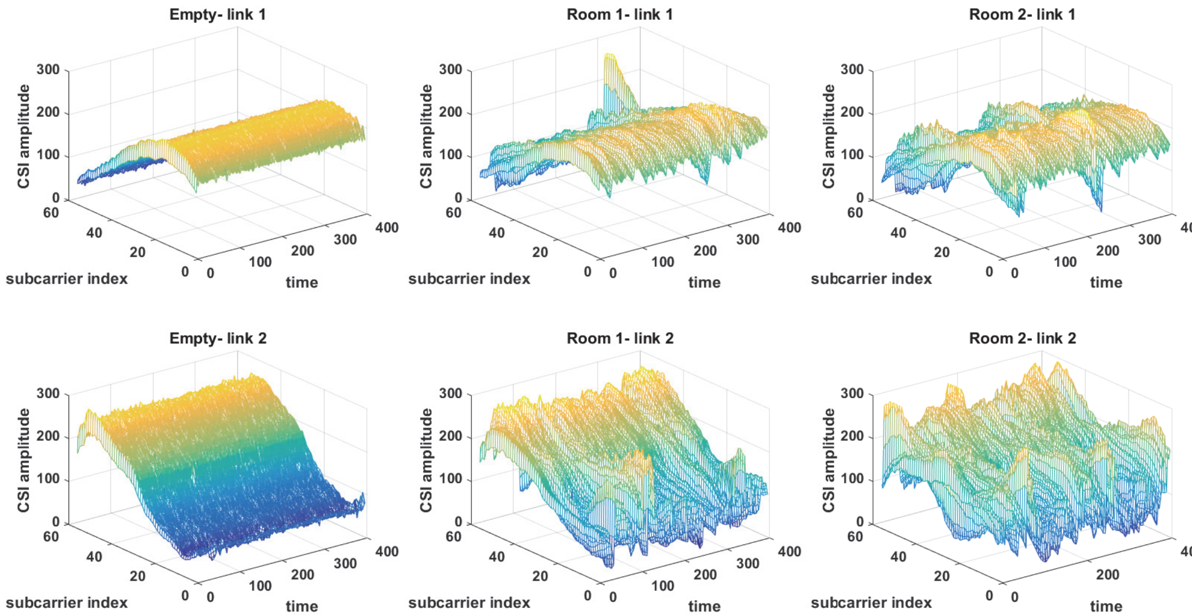

Figure 2: An example of CSI amplitudes captured over 20 seconds (with sampling rate of 20pks/s) for two different Tx-Rx links representing *No Motion* and walking in *Room1* and *Room2*.

Figure 1 gives an overview of an $n \times m$ multiple input and multiple output (MIMO) system with n transmitting antennae and m receiving antennae. Environmental changes and human body movements affect the CSI values of different links independently, but affect the different subcarriers of each link in a similar manner; this correlation will be exploited in our system.

Figure 2 gives an example of CSI amplitude streams from two different links obtained while a user is walking inside two different rooms of a residential apartment, as well as a capture from an empty apartment. As can be seen, the data is very noisy and differs in the two spaces, but also has regularities. The first challenge is to infer a mapping between the CSI values over multiple links and subcarriers, and the location of a moving entity.

Preparing the CSI Data

The obtained CSI packets from all links and subcarriers are indexed over time, so the input of our localization system is a group of synchronized sequences of observations, i.e. data streams. We carry out multiple steps of data pre-processing on these streams in order to eliminate or tame redundant and noisy samples and prepare a stable feature vector before feeding it to the base classifier for localization.

Noise removal and standardization: The raw data contain high-frequency noise from a variety of sources (as seen above). Also, the duration of typical human activities and gestures happens at low frequencies (no more than 2Hz). Therefore, we apply a low-pass filter with cut-off frequency of 2Hz to each CSI stream individually, in order to remove the high-frequency noise as well as the static components.

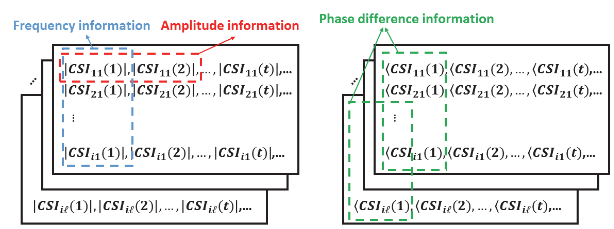

Figure 3: Feature extraction strategies from CSI magnitude (left) and CSI phase (right) values.

On the other hand, at each time stamp multiple CSIs values for different Rx-Tx links can take values in different dynamic ranges, while the values of different subcarriers within each link can get shifted and scaled over time. These irrelevant and unwanted changes are removed by introducing a fixed-score scaling normalization module, which standardizes the CSI feature space to a predefined reference range, so we can reliably track meaningful variations in the signals. The L_2-norm of the CSI vector was calculated for each link to rescale all values to the reference range.

Feature acquisition: We begin by sliding a moving window with overlap over the stream of samples, in order to extract correlated features that describe environmental events. This creates a vector of the form:

$$W(t) = \{CSI_{i\ell}(t-w+1), \cdots, CSI_{i\ell}(t-1), CSI_{i\ell}(t)\}, \tag{2}$$

where w is the size of the moving window and t is the time stamp of the CSI values of subcarrier i of link ℓ. This data is then further processed using different feature extraction technique, which are illustrated in Figure 3, and described below:

- **Amplitude information:** Statistics computed from per-subcarrier CSI amplitudes are the most widely used features in CSI-base systems, since they exhibit more temporal stability (Wang et al. 2015; Wu et al. 2013). We calculated the moving variance and moving average of all CSI amplitudes within each sliding window $W_{i\ell}(t)$; these have appeared to be very useful in our application.
- **Phase information:** Several prior papers reported that the CSI phase values are very noisy (Sen et al. 2012; Yang, Zhou, and Liu 2013; Wang et al. 2015). Therefore, in order to infer meaningful behaviour from phase variations, an extensive amount of preprocessing and enhancement needs to be performed. In general, in wireless communication applications, the phase difference between the received signals at each antenna array is roughly correlated to the angle of arrival (AoA), which yields a method for determining the direction of RF wave propagation. Through exploratory experimentation, we realized that the phase differences between various pairs of Rx-Tx links can actually help localize human movement with respect to the positions of the transmitter and receiver devices. Therefore, we track the variance of the phase differences between the subcarriers of all pairs of Rx-Tx over the moving window $W_{\ell,\ell'}(t)$ as another group of relevant features for our localization system.
- **Frequency information:** As we mentioned earlier, various CSI values describe channel properties in the frequency domain and a moving subject can change signal reflections differently based on his or her location, resulting in different delay profiles. This frequency information is embedded in the correlations among (CSI values of) subcarriers in each Rx-Tx link, which can be inferred by computing statistics such as variance, log energy entropy, standard deviation, kurtosis and skewness over the moving window $W_{\ell}(t)$.

Machine Learning System Architecture

We aim to perform localization in real-time, using the stream of CSI data. As explained earlier, drift in the distribution of input features is expected, so learning needs to continue over the lifetime of the system. However, it is cumbersome to query the end-user and request a new batch of labeled data after the drift to retain the accuracy, especially because we do not expect end users to understand this calibration step. Hence, our ultimate goal is to avoid involving the user for as long as possible. We assume that labeled data obtained initially would be during a setup phase, which is acceptable (as this is analogous to a technician coming to set up a new cable box, for example), but in this phase, even if the end-user would be involved, there would be guidance.

In order to maintain the performance of the system in spite of the drift, the first step is to automatically detect significant changes in the distribution of features extracted from the CSI stream in a timely manner, and then update the out-dated model. We developed a real-time semi-supervised learning framework that divides the localization problem into two phases: initial offline training using a batch of labeled data and online evaluation and adaptation using the streaming data. An overview of the system architecture is given in Figure 4. In the following sections we provide details of the different components.

Initial training

The training process initiates by fitting a base supervised learner to obtain a mapping between features extracted from CSIs and different locations in the sensing environment, using a small amount of labeled data. In order to simplify the problem, localization is performed at the level of discrete "areas" inside a dwelling, which could be rooms, but also finer grained than rooms (e.g., a class could be "on the couch" or "in the reading chair", for example). After a small amount of preliminary experimentation, we chose Random Forests (Breiman 2001) as the base learner. Random Forest is an ensemble estimator that builds several decision trees on random subsets of the samples from the original training set and then aggregates their individual predictions, usually by averaging, to form final decisions. Therefore, besides predicting a label, the obtained classifier also provides a measure of the uncertainty in its prediction, expressed through the proportion of votes given by all trees for each class. We use the proportion of votes that agree on the outcome to estimate a *confidence score*, which quantifies how certain the classifier is of its decision.

After building the classifier, real-time evaluation on the streaming data begins. Arriving CSI measurements are processed as described above, and the obtained features are then fed into the base classifier frame by frame, which results into a stream of predicted location labels and their associated confidence scores. From a practical point of view, it is important to have a stable localization system, which smoothly transits between different classes. Thus, in order to reduce the variance in the sequence of predicted labels and minimize the error when outputting final decisions to the end user, we also use some additional strategies that can be used to increase stability of the labeling.

Consider a K-class classification problem, where for each time frame $W(t)$ (from Eq. 2) a class label c_t is independently obtained from the base learner with confidence scores (prediction probability) of p_t. We consider a larger decision frame $W' \geq W$ with length w', where given a prediction history, $\{c_{t-w'+1}, \cdots, c_{t-1}, c_t\}$ and $\{p_{t-w'+1}, \cdots, p_{t-1}, p_t\}$, a final class decision C_T is made for time buffer $T = \{t - w' + 1, \cdots, t-1, t\}$ through the following steps:

- **Outlier removal:** discarding *rare* class labels that last less than α consecutive samples
- **Uncertainty removal:** discarding any class label with confidence score less than β
- **Transition bias:** imposing an extra bias towards keeping the current predicted class label until the average confidence score for switching to another class reaches a certain level γ.

At the end of the localization process, only the final decision C_T appears on the user interface.

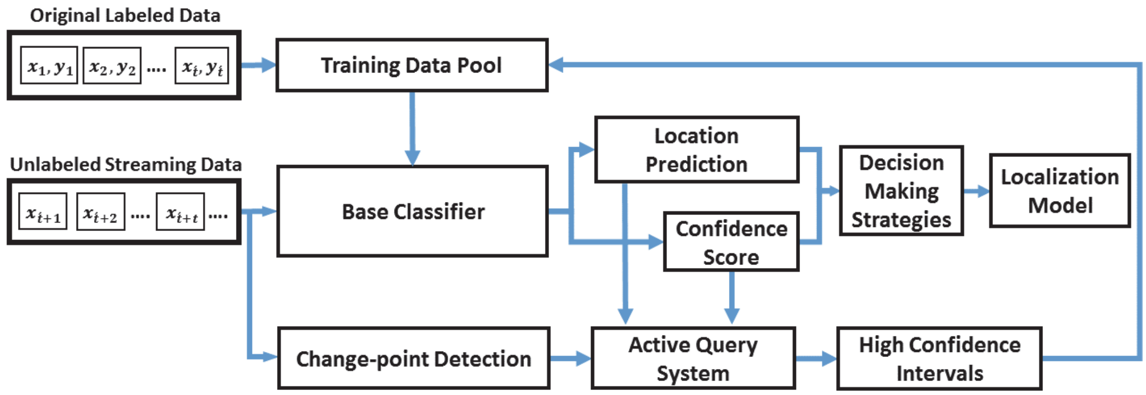

Figure 4: General overview of the proposed localization system

To see the intuition behind the first strategy, suppose the average speed of a human walk roughly 1 m/s (and not exceeding 2 m/s) and the fact that in our application, the average distance between location classes (rooms) is in the scale of a few meters, we can easily ignore the classes that appear for less than 0.5 seconds in the decision frame as outliers. Secondly, we take advantage of the confidence measure produced by the Random Forest classifier to ensure the behavior of the labels is sufficiently stable for practical use. For instance, using the confidence score of the base classifier, we apply weighted voting to smooth the transition between locations when the user walks between rooms.

Concept Drift Detection

Over time, gradual changes happen to the distribution of CSI magnitude (i.e, frequency information) over each Tx-Rx link independently, and can affect one, some or all of these links over time. Therefore, a change-point detection algorithm is required to constantly estimate and monitor the stability of all links individually. Our method uses Kullback-Leibler (KL) divergence as a distance metric to track substantial changes in the distribution of the features, $\{|CSI_{1\ell}|, \cdots, |CSI_{i\ell}|\}$.

The KL-divergence between two distributions $CSI_\ell(t)$ and $CSI_\ell(t+\delta)$ is estimated as:

$$D_\ell(\delta) = \sum_{i=1}^{I} CSI_{i\ell}(t) \log \frac{CSI_{i\ell}(t)}{CSI_{i\ell}(t+\delta)}, \qquad (3)$$

where D_ℓ corresponds to the drift measure of link ℓ, at time stamp δ after the initial training set captured at time t. We set an empirical threshold θ to automatically detect any significant divergence in any element of vector $\boldsymbol{D_\ell(\delta)} = \{D_1, \cdots, D_L\}$.

Once a significant drift in any of the links is detected, the algorithm asks for an update. In the following section we will explain how to update the classifier without involving the user.

Adapting the Classifier to Track Signal Distribution Changes

Although the unwanted changes in CSI magnitude and their timing are not predictable, they usually happen over a short period of time and do not involve all signals simultaneously. Therefore, many samples still get correctly classified even once drift has occurred, as some partial mappings between the feature space and class labels still hold. Intuitively, we aim to select these "good" representative samples from the history to update the training data. The main idea is to use confidence scores provided by the base classifier to establish *high confidence intervals* over the stream of unlabeled data and accumulate a batch of the most representative samples and their associated inferred labels over time.

When the change detection block identifies a significant drift that demands retraining, a query for updating the base classifier is formed. An *active query module* receives these demands and pushes sub-samples from the most recent high confidence intervals into a *pool* of labeled training data. In this way, there is no need to query the user to avoid deterioration in prediction accuracy and the system can maintain its performance even after drifts.

Let $\boldsymbol{X} = \{X(1), X(2), \cdots, X(t'), \cdots\}$ be the stream of features extracted from CSI values, and let $\boldsymbol{Y} = \{Y(1), Y(2), \cdots, Y(t)\}$ be the true labels of $X(t) : t \in \{1, \cdots, t'\}$. We define a sliding window P of length $\mu \gg w$ over the streaming unlabeled data starting from $t > t' + 1$, in which we keep a history of prediction labels$\{c_{t-\mu+1}, \cdots, c_{t-1}, c_t\}$ and confidence scores $\{p_{t-\mu+1}, \cdots, p_{t-1}, p_t\}$.

The system narrows the collection of samples by setting a relatively high confidence threshold. Shortly after the change-point detection block produces an alert, the system queries the samples in high confidence intervals and updates the base classifier with a fusion of the original training data and these new high-confidence samples.

As the real-time platform needs to provide long-term functionality, we need to keep the size of this repository of samples limited, in order to provide good scalability of data storage and retrieval. Thus, we need to force the system to discard old data when the size of the pool exceeds a certain

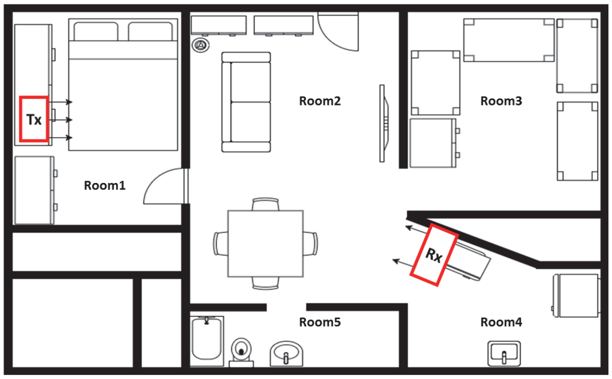

Figure 5: An example of the apartment layout and device placement (Tx:Access point, Rx: mini PC).

point.

Application Development and Deployment

Aerîal is a company that develops device-free activity recognition platforms and technologies for creating intelligent environments like smart homes. The indoor localization system that was introduced here is deployed within one of their smart home applications, and is the result of collaborative research between the university and this industrial unit.

Our system was initially designed and developed base on a significant amount of real data collected under several different setups and experimental scenarios. The initial version of the system, including the signal processing, data mining and machine learning blocks, was coded using MATLAB and WEKA. After many stability evaluations and modifications, the final version of the core algorithms, user interface and central data processing units were implemented and tested in Python by a team of developers. It took about 4 months to design, improve and integrate the proposed approach into the existing Aerîal application platform. The system is currently functioning as a part of Aerîal home security and surveillance products.

Performance Evaluation

Operation in Real Environments

The localization platform can operate in any residential apartment using a mini PC equipped with the CSI collection tool as receiver and a commercial access point as transmitter to create a wifi connection for room-level localization. Figure 5 depicts an example of device placement and floor plan in one residential apartment. After device placement, a short period of initial training is performed by asking the user to simply walk inside each room (or inside each area of interest) and record the labels. Also, a capture from the empty apartment is needed for the *No motion* class. The user interacts with the system through a web-based interface, which can be accessed from a computer or a portable device such as tablet or smartphone. Figure 6 illustrates the user interface in the *Training* mode. Once the initial training is over, the real-time localization system is activated and the user is able to

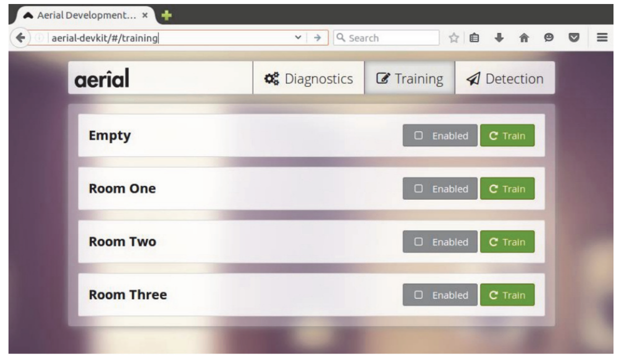

Figure 6: Aerîal web interface, *Training* mode.

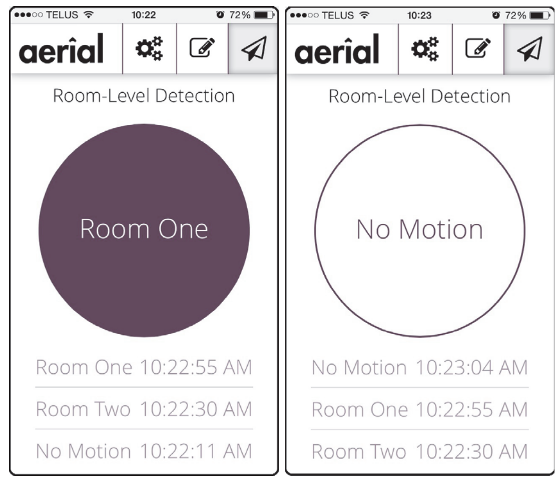

Figure 7: Aerîal interface in *Detection* mode.

track the location of a moving person within the apartment in the *Detection* mode as shown in Figure 7. Beside location identification, this tool can be used as an intruder alarm that notifies the user as soon as a person enters in their empty apartment.

Experiments

In practice, the initial training of the localization system usually results in very robust performance, but it only lasts about an hour before the accuracy begins to drop due to the unexpected signal changes. The early version of the platform used to require manual re-training 90 minutes after the initial training, regardless of whether drifts occurred or not. In contrast, the method we described here detects significant changes and reacts in a timely fashion to maintain accuracy.

We evaluated our algorithm on real data collected during experiments conducted in 7 different residential apartments. In each round of an experiment, an initial training set was recorded, where the CSI values were captured while the user was asked to walk inside each room for 45 sec. A 45 sec. capture from the empty apartment was also taken. The number of classes was from 4 to 6 in different apartments, including the empty or No motion class. In order to

Dataset	Initial Test	Diagn.1	Diagn.2	Diagn.3	Adaptive learner
Apart.1	$98\% \pm 1$	87%	84%	74%	$94\% \pm 3$
Apart.2	$94\% \pm 2$	86%	80%	77%	$91\% \pm 2$
Apart.3	$95\% \pm 2$	85%	85%	84%	$92\% \pm 1$
Apart.4	$96\% \pm 2$	87%	85%	79%	$91\% \pm 2$
Apart.5	$97\% \pm 1$	92%	84%	82%	$93\% \pm 2$
Apart.6	$94\% \pm 2$	90%	89%	82%	$93\% \pm 2$
Apart.7	$95\% \pm 2$	92%	81%	78%	$90\% \pm 1$

Table 1: Accuracy of the room-level location identification at different time intervals with and without adaptive solution.

obtain examples of drift in the input, a couple of diagnostic sets were captured in various time intervals from 60 min. to 11 hours after the initial set, and these diagnostic sets were used to evaluate the adaptive algorithm. We performed 3-10 rounds of evaluation per apartment in order to obtain averages for the results. We note that while this was specifically the design of the evaluation, the software is deployed and in use on a permanent basis in these apartments.

As described above, the modules in our proposed system contain parameters and thresholds that need tuning in order to achieve robust localization, which we now explain in more detail. The CSI values were logged with the sampling frequency of 20 packets per second on the mini PC and after preprocessing, 904 features were extracted from the raw measurements within the moving window of $w = 40$ packets. These features were fed into a Random Forest classifier with 100 trees, whose depth was set to 2. During the decision making phase, the moving voting window length was $w' = 40$ samples. The strategies for pruning and adjusting decisions used parameters set to $\alpha = 15$, $\beta = 0.5$ and $\gamma = 0.75$, respectively. The length of the two windows w and w' had to be set carefully, because they both directly contribute to the delay in the real-time prediction system. A very long window size yields a noticeable delay in the localization, especially when transiting from one room to another, which can lead a user to be dis-satisfied with the system. On the other hand, if the window lengths are not long enough, they might not capture relevant events such as human walking. We found that the parameters above worked well for all the units tested. We note that we obtained these parameters through prior experimentation in the space occupied by Aerîal.

The evaluation results from the experiments are shown in Table 1. The accuracy of the localization system, before any drift occurs, is evaluated on an test set, which is captured right after the original training set with no delay (indicated in first column of the table). The accuracy rate in the last column of each row represents the average performance of the adaptive learner on all diagnostic sets of that row. The first diagnostic sets were captured after between 60-90 min in all apartments, whereas the second and third sets were taken between 90 min. to 11 hours.

The results show that the accuracy obtained by the base learner is very high (as evaluated using a test set), but using the resulting classifier over an extensive period of time leads to significant accuracy loss. Our semi-supervised learner is able to maintain accuracy close to that of the base learner in the face of signal drift. As explained, this is done with no additional new labeled training data. The standard deviations indicated are quite small.

Conclusions, Limitations and Lessons Learned

Conclusion

Recently, the Internet of Things (IoT) has received a lot of attention, due to its potential to provide ubiquitous connectivity, enabling context-aware computing, and its ability to produce massive amounts of data, which provides insight into different aspects of human behavior. Many AI techniques, such as machine learning, decision making, heuristic search and optimization, can contribute significantly to the understanding and visualization of this type of data. Our wifi-based indoor localization application fits into this research area. Our work relies on existing commercial devices, and relies heavily on machine learning in order to perform location detection and intruder detection. Our quantitative experimental results, as well as qualitative user feedback on the current platform confirm that this approach for indoor localization is very promising. We note that, while our approach requires several parameters, finding good values for these was not difficult. We also wish to note that in earlier version of the system, we tried Bayesian probabilistic modeling of various types for this data, with much less success. The empirical confidence measure provided by the Random Forrest classifier, together with the strategies for maintaining label quality, performed better than theoretically-motivated Bayesian inference. We suspect this is due to the noise in the CSI signals, which renders many approaches brittle. We found random forest to be both reliable, as well as able to handle the large amount of real-time data required.

Assumptions

At this stage of the technology, our system is designed to accurately track variations in CSI measurements from a single user at a time. To the best of our knowledge, all of the studies working with the same type of data are in preliminary stages of development and their techniques are implemented in a single-user setting as well. The key reason for this limitation is the sensitivity of the CSI values to ambient changes, including non-target subjects. We followed the same setting for the localization application and assumed that at each time stamp there is only one main target user, which we track. However, the intruder detection application can sense the presence of any movement in the environment, and it is not limited to a single-user setting. In the future, we would like to generalize our platform to achieve a robust, multiple people tracking system.

CSI measurements are affected significantly by the location and distance between the transmitter and receiver, as well as the orientation of their antennae. Therefore, at this point we require that during both the initial supervised training and the real-time functioning of the device, the location of these endpoints and the position of the antennae are predetermined and fixed at all time. If any displacement of the devices occurs, the initial training needs to be repeated in order to preserve the accuracy of the system.

Beyond Localization

Although we primarily developed a platform for localization and intruder detection, the collected measurements have the potential to reveal finer information about the activities and events in the sensing area. Beside localization, CSI measurements have recently been used as sensing technology for high-level and low-level activity recognition, since they do not require special-purpose hardware. For example, in (Zeng, Pathak, and Mohapatra 2015) the authors leveraged CSI signals to analyze shopper behaviour and browsing patterns. The study in (Ali et al. 2015) claims to perform keystroke extraction on a keyboard from CSI measurements. These studies have been carried out in very controlled environments and their results are mainly reported on a batch of pre-recorded data instead of real-time data streams. However, these results are intriguing and potentially transferable to a naturalistic setting.

Aerîal has also developed other algorithms for application scenarios which require more precise analysis on the CSI values. Some preliminary evaluations on the data obtained from our current experimental setup have shown encouraging results for leveraging CSI in user identification from walking patterns, learning physical activities such as *walking*, *sitting* and *standing*, and monitoring and tracking breathing and heart rates (which can be used, eg., for monitoring babies who are predisposed to apneas, or elderly living in their own homes). However, deployment of stable systems that perform in real-time and naturalistic situations requires further improvement of the current implementation, as well as extensive research on the learning techniques, in order to cope with the noisy and non-stationary nature of these signals. We hope that this paper will motivate other AI researchers to consider working with this new type of signal data.

Finally, we would like to mention that the semi-supervised learning approach we proposed is useful more generally, beyond this application, to any situations in which real-time classification of a time series is necessary and the application includes concept drift which affects some of the inputs drastically, but not all inputs at the same time. In such situation, the strategy we outlined for incorporating in the new training set both old data and new data which is still confidently classified is applicable. This could be useful e.g., in clinical monitoring from multiple physiological signals.

Acknowledgments

This research was partially supported by MITACS. We thank our colleagues at Aerîal that greatly assisted the research.

References

Ali, K.; Liu, A. X.; Wang, W.; and Shahzad, M. 2015. Keystroke recognition using wifi signals. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 90–102. ACM.

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

Breiman, L. 2001. Random forests. *Machine learning* 45(1):5–32.

Kotaru, M.; Joshi, K.; Bharadia, D.; and Katti, S. 2015. Spotfi: Decimeter level localization using wifi. In *ACM SIGCOMM Computer Communication Review*, volume 45, 269–282. ACM.

Pirzada, N.; Nayan, M. Y.; Hassan, F. S. M. F.; and Khan, M. A. 2014. Device-free localization technique for indoor detection and tracking of human body: A survey. *Procedia-Social and Behavioral Sciences* 129:422–429.

Popleteev, A. 2013. Device-free indoor localization using ambient radio signals. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, 549–552. ACM.

Saab, S. S., and Nakad, Z. S. 2011. A standalone rfid indoor positioning system using passive tags. *IEEE Transactions on Industrial Electronics* 58(5):1961–1970.

Sen, S.; Radunovic, B.; Choudhury, R. R.; and Minka, T. 2012. You are facing the mona lisa: spot localization using phy layer information. In *Proceedings of the 10th international conference on Mobile systems, applications, and services*, 183–196. ACM.

Wang, X.; Gao, L.; Mao, S.; and Pandey, S. 2015. Deepfi: Deep learning for indoor fingerprinting using channel state information. In *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, 1666–1671. IEEE.

Wu, K.; Xiao, J.; Yi, Y.; Chen, D.; Luo, X.; and Ni, L. M. 2013. Csi-based indoor localization. *IEEE Transactions on Parallel and Distributed Systems* 24(7):1300–1309.

Xiao, J.; Wu, K.; Yi, Y.; Wang, L.; and Ni, L. M. 2013. Pilot: Passive device-free indoor localization using channel state information. In *Distributed computing systems (ICDCS), 2013 IEEE 33rd international conference on*, 236–245. IEEE.

Yang, Z.; Zhou, Z.; and Liu, Y. 2013. From rssi to csi: Indoor localization via channel response. *ACM Computing Surveys (CSUR)* 46(2):25.

Zeng, Y.; Pathak, P. H.; and Mohapatra, P. 2015. Analyzing shopper's behavior through wifi signals. In *Proceedings of the 2nd workshop on Workshop on Physical Analytics*, 13–18. ACM.