Contents lists available at ScienceDirect

Journal of Systems Architecture

journal homepage: www.elsevier.com/locate/sysarc

A Hidden Markov Model based smartphone heterogeneity resilient portable indoor localization framework

Saideep Tiku[a,*], Sudeep Pasricha[a,*], Branislav Notaros[a], Qi Han[b]

[a] Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO, USA
[b] Department of Computer Science, Colorado School of Mines, Golden, CO, USA

ARTICLE INFO

Keywords:
Indoor localization
Indoor navigation
HMM
Heterogeneity
Wi-Fi
Fingerprinting

ABSTRACT

Indoor localization is an emerging application domain that promises to enhance the way we navigate in various indoor environments, as well as track equipment and people. Wireless signal-based fingerprinting is one of the leading approaches for indoor localization. Using ubiquitous Wi-Fi access points and Wi-Fi transceivers in smartphones has enabled the possibility of fingerprinting-based localization techniques that are scalable and low-cost. But the variety of Wi-Fi hardware modules and software stacks used in today's smartphones introduce errors when using Wi-Fi based fingerprinting approaches across devices, which reduces localization accuracy. We propose a framework called *SHERPA-HMM* that enables efficient porting of indoor localization techniques across mobile devices, to maximize accuracy. An in-depth analysis of our framework shows that it can deliver up to 8× more accurate results as compared to state-of-the-art localization techniques for a variety of environments.

1. Introduction

The arrival of Global Positioning System (GPS) technology within smartphones has revolutionized the way we navigate in the outdoor world. Today, indoor localization technology holds a similar potential to disrupt the way we navigate within indoor spaces that are unreachable by GPS. An example scenario is localizing patients, staff, and equipment in large hospitals and assisted living facilities. Precise location information can allow first responders closest to a patient to be notified in emergencies. Some startups (e.g., Shopkick, Zebra) are also beginning to provide indoor localization services that can help customers locate products inside a store [1].

Unlike GPS for outdoor localization, no standardized solution exists for indoor localization. Therefore, a myriad of techniques have been developed that use various sensors and radio frequencies. Some commonly utilized radio signals are Bluetooth, ZigBee, and Wi-Fi [2]. Among these, Wi-Fi based indoor localization has been the most widely researched, due to its low setup cost and easy availability. Today, Wi-Fi access points are deployed in most indoor locales around the world and all smartphones support Wi-Fi connectivity.

Despite the advantages of Wi-Fi based indoor localization, there are also some drawbacks. Many prior solutions perform indoor localization by measuring Wi-Fi Received Signal Strength Indicator (RSSI) values and calculating distance from Wi-Fi Access Points (WAPs). These works assume that wireless signal strength reduces in a deterministic manner as a function of distance from a signal source (i.e., WAP). But Wi-Fi signals suffer from weak wall penetration, multipath fading, and shadowing effects in real-world environments, making it difficult to establish a direct mathematical relationship between RSSI and distance from WAPs. These issues have served as a motivation for using fingerprinting-based techniques. Fingerprinting is based on the idea that each indoor location exhibits a unique signature of WAP RSSI values. Due to its independence from the RSSI-distance relationship, fingerprinting can overcome some of the aforementioned drawbacks with Wi-Fi based indoor localization.

Fingerprinting is usually carried out in two phases. In the first phase (called offline or training phase), the RSSI values for visible WAPs are collected along indoor paths of interest. The resulting database of values may further be used to train models (e.g., machine learning-based) for location estimation. In the second phase (online or testing phase), the models are deployed on smartphones and used to predict the location of the user carrying the smartphone, based on real-time readings of WAP RSSI values on the smartphone.

A majority of the literature that utilizes fingerprinting employs the same smartphone for (offline) data collection and (online) location prediction [3–]. This assumes that in a real-world setting, users would have access to the same smartphone as the one used in the offline phase. But today's diverse smartphone market, with various brands and models, largely invalidates such an assumption. In reality, the smartphone user base is a distribution of heterogeneous devices that vary in antenna gain, Wi-Fi chipset, OS version, etc. [8,25–30].

[*] Corresponding author.
E-mail addresses: saideep@colostate.edu (S. Tiku), sudeep@colostate.edu (S. Pasricha), notaros@colostate.edu (B. Notaros), qhan@mines.edu (Q. Han).

https://doi.org/10.1016/j.sysarc.2020.101806
Received 31 August 2019; Received in revised form 11 May 2020; Accepted 12 May 2020
Available online 25 May 2020
1383-7621/© 2020 Elsevier B.V. All rights reserved.

Recent work has shown that the perceived Wi-Fi RSSI values for a given location captured by different smartphones can vary significantly [9]. This variation degrades the localization accuracy of conventional fingerprinting. Therefore, there is a need for portable and device heterogeneity-aware fingerprinting techniques. In this paper, we present a lightweight Wi-Fi RSSI fingerprinting framework for *S*martphone *He*terogeneity *R*esilient *P*ort*a*ble localization with *H*idden *M*arkov *M*odels (*SHERPA-HMM*) that is portable across smartphones with minimal accuracy loss. The novel contributions of our work are:

- We conduct an in-depth analysis of Wi-Fi fingerprinting across smartphones to emphasize the importance of device heterogeneity-resilient indoor localization;
- We formulate the indoor localization problem as a Hidden Markov Model (HMM) that utilizes heterogeneity resilient metrics for user path prediction;
- We design the SHERPA-HMM framework for portable Wi-Fi fingerprinting-based indoor localization; SHERPA-HMM employs a lightweight software-based approach to combine noisy fingerprints over distinct smartphones and pattern matching/filtering to improve location accuracy;
- We evaluate SHERPA-HMM against state-of-the-art localization techniques, across a variety of Android-based smartphones that are used for indoor localization along paths in real buildings.

2. Background and related work

Since the establishment of wireless RF signal based indoor localization a few decades ago, a significant level of advancement has been achieved in this area. In general, most indoor localization techniques fall under three major categories: 1) static propagation model-based, 2) triangulation/trilateration-based, and 3) fingerprinting-based. Early indoor localization solutions used static propagation model-based techniques that relied on the relationship between distance and Wi-Fi RSSI gain [10]. These techniques only work well in open indoor areas as they do not take into consideration any form of multipath effects or shadowing due to walls and other indoor obstacles that invalidate the direct distance-RSSI relationship. This method also required the creation of a gain model for each individual Wireless Access Point (WAP) or Wi-Fi router, which is a cumbersome undertaking. Triangulation/Trilateration-based methods use geometric properties such as the distance between multiple APs (Trilateration) and the smartphone [11] or the angles at which signals from two or more WAPs are received [12]. Such methodologies may be more resilient to smartphone heterogeneity but are not resilient to multipath and shadowing effects. Some recent work has also investigated multipath effects for triangulation [13], but the proposed approach cannot be implemented on commodity smartphones, and hence has limited scalability.

Wi-Fi fingerprinting-based approaches associate several sampled locations (reference points) with the RSSI measured with respect to multiple WAPs [2–6]. These techniques are relatively resilient to multipath reflections and shadowing as the reference point fingerprint captures the characteristics of these effects leading to improved indoor localization. Fingerprinting techniques use some form of machine learning techniques to associate Wi-Fi RSSI captured in the online phase to the ones captured at the reference points in the offline phase. Recent work on improving Wi-Fi fingerprinting exploits the increasing computational capabilities of smartphones. For instance, sophisticated Convolutional Neural Networks (CNNs) have been proposed to improve indoor localization accuracy on smartphones [4]. One of the concerns with utilizing such techniques is the vast amounts of training data required by these models to achieve high accuracy. This is a challenge as the collection of fingerprints for training is an expensive manual endeavor and often the lack of training data leads to poor accuracy.

To overcome this limitation, researchers often resort to building more complex frameworks that utilize hybrid techniques such as combining fingerprinting with dead reckoning [32–34]. Dead reckoning refers to the use of inertial sensors and a previous known location to predict a future location. However, dead reckoning accumulates errors over time, and needs to be further augmented via map matching to be useful. Map matching utilizes compute intensive particle filtering based approaches along with the knowledge of known physical features on a map to improve localization accuracy [35,36]. These systems assume that the location of a user in real time is given by a distribution of particles. The location of every particle is then individually updated at every location prediction cycle and interaction of these particles with known physical features such as walls is also captured. Such methodologies often lead to highly compute intensive solutions. Utilizing such complex frameworks levy high energy and computational requirements on resource constrained smartphone platforms, despite their improving capabilities. In [3], an energy-efficient hybrid fingerprinting approach was proposed. However, most prior work, including [3], is plagued by the same drawback, i.e., lack of support for smartphone heterogeneity across both the offline and online phases. This leads to solutions that perform poorly in real-world scenarios.

Coping with device heterogeneity is a significant research challenge in most sensing domains. The recent improvements in the field of deep learning have motivated researchers to apply these models to overcome heterogeneity challenges. For example, the work in [45] suggests using a probabilistic heterogeneity generator for training DNNs for speech and sensor sampling applications, whereas the work in [46] extends this idea to include the use of cycleGANs for alleviating heterogeneity across microphones. Unfortunately, none of these works can be directly applied to the domain of fingerprinting-based indoor localization. While these works attempt to overcome the device heterogeneity challenge through probabilistic data augmentation of heterogeneity features by comparing signals of two devices, the heterogeneity features of each device could be unique thereby limiting the scalability of this approach across devices. Further, hyperparameters for GAN based techniques are known to be difficult to adjust such that they produce meaningful information. In contrast our work focuses on utilizing global similarities across heterogeneous devices and an intelligent combination of optimization techniques to deliver a framework that performs consistently across a verity of smartphones.

The most intuitive approach to address device heterogeneity in the domain of fingerprinting-based indoor localization is to acquire RSSI values and location data manually for each new mobile device [14]. This is unfortunately not very practical. Once RSSI information is collected, manual calibration can be performed through transformations such as weighted-least squares optimizations and time-space sampling [15,16]. These techniques can be aided by crowdsourcing schemes. However, such approaches still suffer from accuracy degradation across devices [19].

In calibration-free fingerprinting, the fingerprinting data is translated into a standardized form that is portable across devices [17]. One such approach, known as Hyperbolic Location Fingerprint (HLF) [18] uses the ratios of individual WAP RSSI values to form the fingerprint. But HLF significantly increases the dimensionality of the training data in the offline phase. The Signal Strength Difference (SSD) approach [19] reduces dimensionality by taking only independent pairs of WAPs into consideration. Improvement in accuracy over this approach through Procrustes-based shape analysis and uniform scaling of RSSI values was proposed in [20]. The RSSI values are standardized via a Signal Tendency Index (STI), while maintaining the dimensionality of the training data. The STI-based technique was shown to perform better than SSD and HLF. However, as STI is used in conjunction with Weighted Extreme Learning Machines (WELMs) for best performance, it is very computationally expensive. Also, the experiments in [20] are performed with a limited set of smartphones, in a one-room-environment that is heavily controlled by the authors. An extension of this work, WinIPS [43], adds to STI-WELM by collecting more data over time using additionally deployed Wi-Fi APs whose sole purpose is to extract RSSI information from

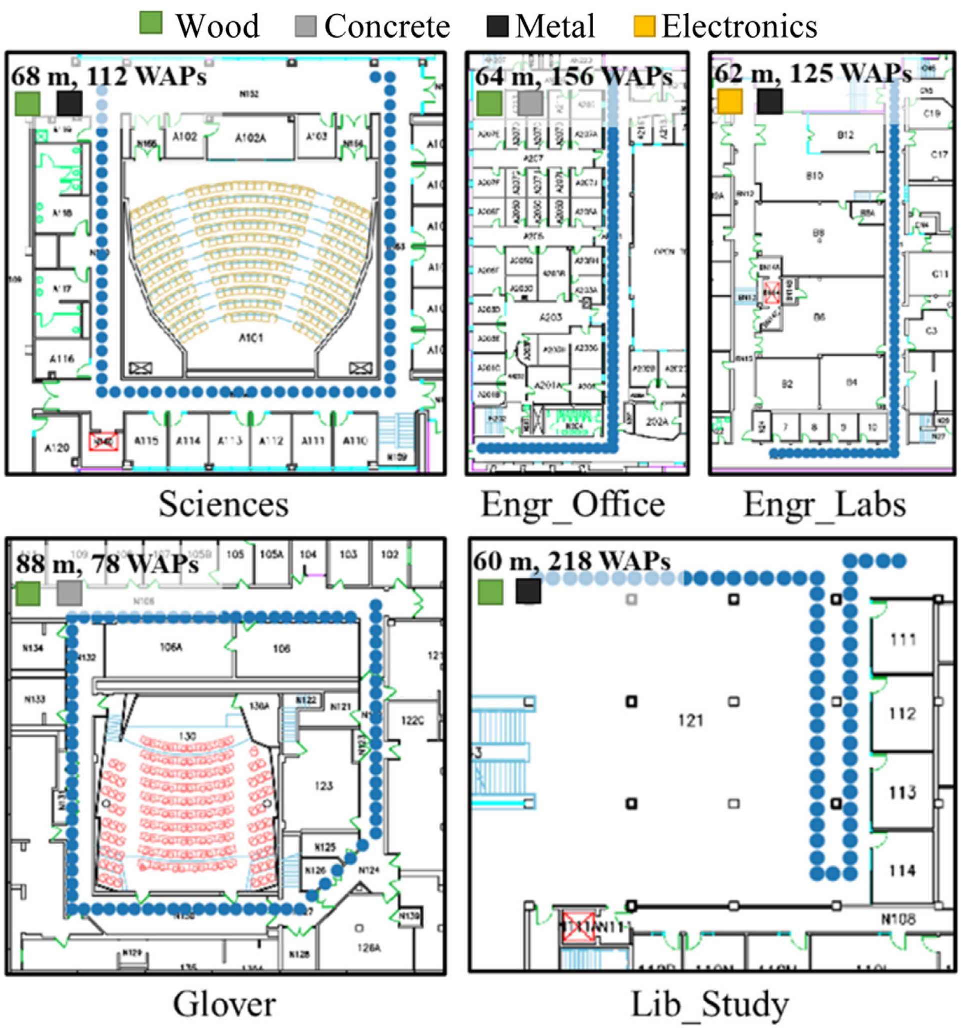


Fig. 1. Benchmark paths for indoor localization (with path lengths and WAP density, and salient path features).

Wi-Fi packets. The newly collected data is then adapted to maintain reliability of the deployed indoor localization framework over time. This work not only has all of the limitations of [20], but it also introduces some new concerns. The WinIPS framework comes at a cost of deploying additional Wi-Fi access points. It improves the resilience to temporal variation of the STI-WELM technique overtime, which is not the focus of our work. The work in [44] is another data adaption technique to support heterogenous devices by translating crowdsourced information of one device into another through multivariate linear regression. This work also employs HMMs to improve overall stability of the results. The major drawback of the work in [44] lies in its very limited resolution of localization accuracy, i.e., at the room or section level.

In contrast, our *SHERPA-HMM* framework provides a novel and computationally inexpensive approach that is tested for a wider set of environments and multiple mobile devices in realistic indoor settings. Unlike some previous works, it delivers accurate results in the resolution of a few meters.

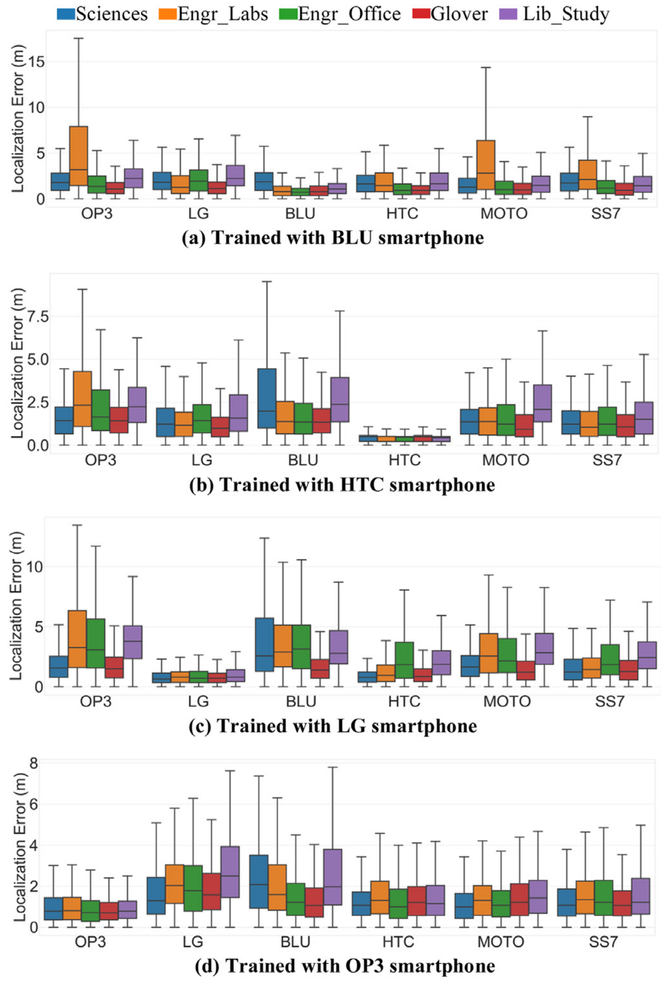
3. Heterogenous fingerprint analysis

We begin with an analysis of the impact of smartphone heterogeneity on a state-of-the-art indoor localization technique: Euclidean-based KNN [3]. To capture the impact of device heterogeneity we observe the performance of the KNN technique to localize six users on five benchmark paths (Fig. 1) using six distinct devices (Table 1).

Fig. 2 shows the boxplots (distribution) for localization error (in the online/testing phase) across all smartphones and indoor paths, for four scenarios where the KNN model was trained on four different smartphones. The most interesting observation is that, in general, the least error is achieved when the device under test is identical in the (offline) training and (online) testing phases. For example, the average localization error of KNN remains stable (< 2 m) when trained and tested with the OP3 mobile device on all paths (Fig. 2(d)). But this trend does not hold when the training device is not the same as the testing device. For example, training on the LG device leads to severe deterioration in accuracy in the *Engr_Labs* path when testing with the OP3, BLU, and MOTO smartphones (Fig. 2(c)). For the *Engr_Labs* path in Fig. 2(a), the average error can be 6× between the best-case training-testing scenario (BLU–BLU), and worst-case scenario (BLU–OP3). *This suggests that a fingerprinting-based indoor localization framework can be extremely unreliable and unpredictable, due to device heterogeneity.*

Table 1
Details of smartphones used in experiments.

Smartphone	Chipset	Android version
OnePlus 3 (OP3)	Snapdragon 820	8.0
LG V20 (LG)	Snapdragon 820	7.0
Moto Z2 (MOTO)	Snapdragon 835	8.0
Samsung S7 (SS7)	Snapdragon 820	7.0
HTC U11 (HTC)	Snapdragon 635	8.0
BLU Vivo 8 (BLU)	MediaTech Helio P10	7.0



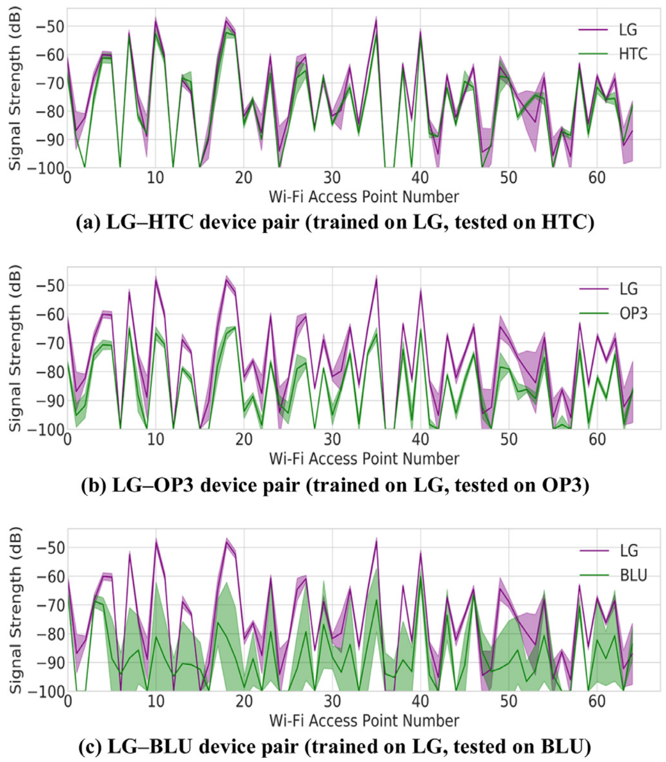
(a) Trained with BLU smartphone

(b) Trained with HTC smartphone

(c) Trained with LG smartphone

(d) Trained with OP3 smartphone

Fig. 2. Error distribution for benchmark paths using KNN [3].



(a) LG–HTC device pair (trained on LG, tested on HTC)

(b) LG–OP3 device pair (trained on LG, tested on OP3)

(c) LG–BLU device pair (trained on LG, tested on BLU)

Fig. 3. RSSI values of each WAP for training and testing pairs. Shaded regions depict the standard deviation.

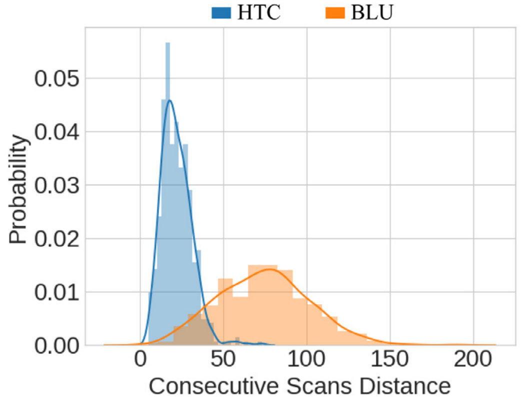


Fig. 4. Probability distribution of the Euclidian distance across consecutive pairs of scans using the HTC and BLU smartphones on the *Engr_Labs* indoor path.

The RSSI values for the best and the two poorly performing training-testing device pairs are shown in Fig. 3. The solid lines represent the mean values, whereas the shaded regions represent the standard deviations of RSSI values. From Fig. 3(a), it can be observed that there is a significant overlap in the RSSI values for the LG and HTC devices. This translates into a shorter Euclidian distance and therefore, produces good results using KNN. On the other hand, in Fig. 3(b) we observe almost no overlap in the RSSI fingerprints. Instead, an inconsistent gain difference can be observed across the two devices. Further, in Fig. 3(c), it can be seen that the BLU device exhibits a significant amount of noise due to variation in the WAP RSSI values for consecutive scans, which can be attributed to its less stable Wi-Fi chipset, compared to the other mobile devices. This leads to severe misprediction when using Euclidian-based KNN. An interesting observation that can be made from looking at Fig. 3 is that the overall shape of the fingerprints is similar, including in Fig. 3(c), where the shape is similar to the mean fingerprint for the BLU device.

From Fig. 3(c), the greater amount of noise from the BLU device is apparent as compared to the other devices, such as the HTC. Identifying and quantifying such noise when using a device for localization (i.e., in the online phase, which is distinct from the offline phase where the localization technique is trained) would allow us to take additional steps to improve localization accuracy. However, it is difficult to identify if a device is capturing noisy fingerprints in the online phase, given a limited set of fingerprints along a path. One approach to quantifying noisy readings could be to check for the Euclidian distance across consecutive scans in the online phase. Since consecutive online scans are conducted using the same device, they should not change significantly over short distances and be similar in terms of Euclidian distance.

To test this hypothesis, we walked over the *Engr_Labs* indoor path with the BLU (most noisy fingerprints) and HTC (most stable fingerprints) smartphones while capturing Wi-Fi fingerprints with consecutive scans during the walk. Fig. 4 depicts the distribution of the Euclidian distance between consecutively captured Wi-Fi fingerprints for the BLU and HTC devices over the *Engr_labs* path. From Fig. 4, we observe that the consecutive scan distances for the HTC device are distributed over a very short range, denoting a stable collection of Wi-Fi fingerprints. However, the distances for the BLU device are distributed over a much wider range due to the variation/noise over consecutive Wi-Fi scans. *This approach can be used to identify mobile devices that capture unstable fingerprints during the online phase.*

The discussion in this section suggests that a portable methodology that captures the pattern of similarity across fingerprints from hetero-

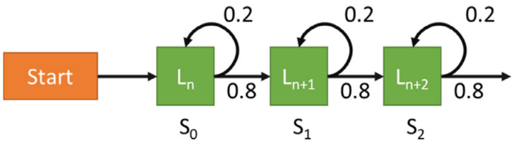


Fig. 5. Reference points represented as states in a Hidden Markov Model with given transition probabilities from one state to another.

geneous smartphones and is able to overcome the noisy behavior of the testing devices, in an energy efficient manner, should deliver better accuracy for indoor localization. These observations serve as the motivation for our proposed *SHERPA-HMM* framework for lightweight and portable localization, as discussed in Section 5. The next section provides a background on HMMs that are used by *SHERPA-HMM*.

4. Hidden Markov Model (HMM) formulation

In this section, we discuss the formulation of the indoor localization process as a Hidden Markov Model (HMM). An HMM statistical prediction model is one that estimates the next hidden state given the transition probability of moving from the current hidden state to the next hidden state and probabilities of observable states [39]. HMMs are particularly renowned for identifying patterns that change with time and have applications in the area of handwriting recognition [38], activity recognition [41], speech synthesis [42], etc. In this paper, we utilize Wi-Fi RSSI pattern similarity as observable (non-hidden) states and predict the user's location or path taken by user which are not directly observable (hidden states).

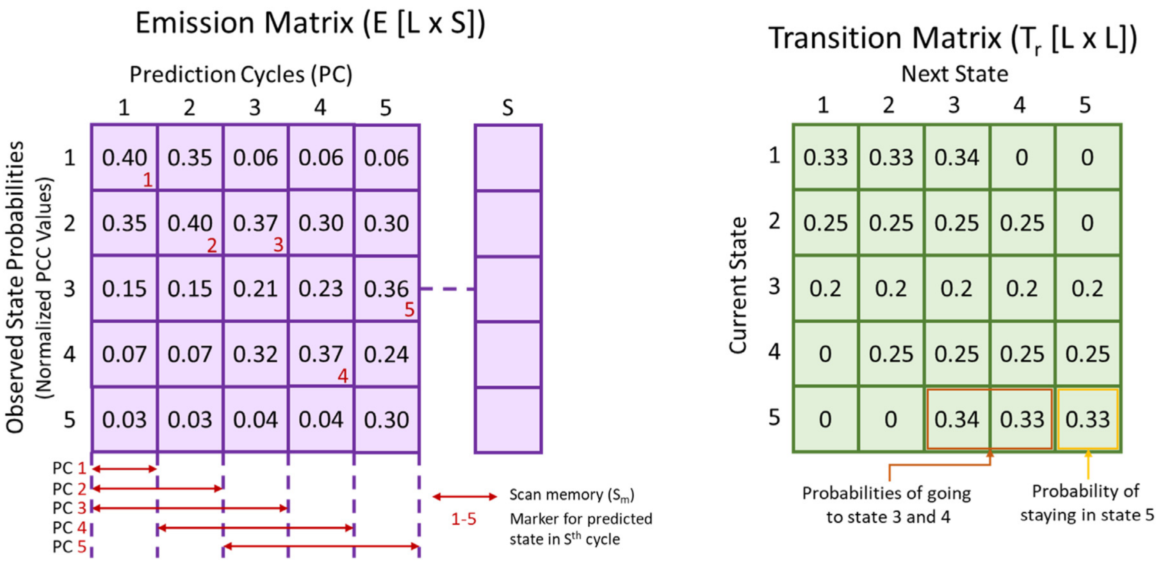
As shown in Fig. 5, we can translate the indoor localization process into a Markov process by first assuming that discrete localizable locations (denoted by L_n, L_{n+1}, L_{n+2}…) on the indoor floor plan are the states. As there is no direct way of checking if the predicted position or state in the online phase is correct, these states are referred to as hidden states. Further, for a given path taken by a user in the online phase, there may be certain known probabilities of going from one hidden state to another. From Fig. 5, we observe that a user is 80% likely to go to the next state and 20% likely to stay on the same states at any given time-step (S_n). In our case, we assume that a user moving on a path is equally likely to move in all directions by a finite amount.

Fig. 6 represents an example of transition and emission matrices for a given path that are critical components of our HMM formulation. The probabilities of transitioning from one state to another are also referred to as the transition probabilities and are mathematically represented as a matrix. The transition matrix T_r is of size $[L \times L]$, where L is number of discrete hidden states (locations in our case). The transition matrix shown in Fig. 6 describes one such example that contains a total of 5 locations or states on a path, thereby producing a matrix of size [5 × 5]. In Fig. 6, the current states are listed as rows and the next states are represented by columns. So, the probabilities of transitioning from state 5 (current state) to state 3 or state 4 (next state) would be 0.34 and 0.33, respectively, as per the transition matrix. Therefore, the transition probability of going from any state i to a state j would be given by the value of $T_r[i,j]$ in the transition matrix.

The observable state information is represented through the emission matrix and is mathematically expressed by $E\ [K \times S]$ (as shown in Fig. 6), where K is the number of observable states and S is the number of subsequent measurements of the observable states (prediction cycles in our framework). In the context of our work, the observable states are the “Wi-Fi pattern similarity” of a scanned unknown Wi-Fi fingerprint (online RSSI vector) with respect to the Wi-Fi fingerprints associated with known locations (offline RSSI vectors). As the number of known locations is L, the size of the emission matrix in the context of SHERPA-HMM becomes [L × S]. In Fig. 6, as we have 5 locations on the path, each measurement of the subsequent state or prediction cycle contains 5 probabilities ($K = L = 5$) in the emission matrix, each associated with being at a specific state or location. The methodology for computing the emission probabilities in our work is dependent on Pearson's Cross Correlation and is explained in greater detail in the next section.

An HMM based framework utilizes information from the observable states (emission matrix) and known transition probabilities (transition matrix) to identify the most likely path or series of hidden states. This is achieved through the Viterbi algorithm [40]. The Viterbi algorithm identifies the most likely sequence of hidden states, also known as the Viterbi path, given the probabilities of observed states.

Here we explain the behavior of the Viterbi algorithm in the context of our framework through a working example using Fig. 6. In the initial state, we already have the user defined transition matrix of size [5 × 5]. However, the emission matrix is empty with 5 rows ($K = L = 5$) and 0 columns. In the first prediction cycle ($S = 1$), a column with emission



Emission Matrix (E [L x S])

Observed State Probabilities (Normalized PCC Values) / Prediction Cycles (PC)

	1	2	3	4	5	S
1	0.40 1	0.35	0.06	0.06	0.06	
2	0.35	0.40 2	0.37 3	0.30	0.30	
3	0.15	0.15	0.21	0.23	0.36 5	
4	0.07	0.07	0.32	0.37 4	0.24	
5	0.03	0.03	0.04	0.04	0.30	

Transition Matrix (T_r [L x L])

Current State / Next State

	1	2	3	4	5
1	0.33	0.33	0.34	0	0
2	0.25	0.25	0.25	0.25	0
3	0.2	0.2	0.2	0.2	0.2
4	0	0.25	0.25	0.25	0.25
5	0	0	0.34	0.33	0.33

PC 1
PC 2
PC 3
PC 4
PC 5
Scan memory (S_m)
1-5
Marker for predicted state in S^{th} cycle
Probabilities of going to state 3 and 4
Probability of staying in state 5

Fig. 6. The emission and transition matrices with populated probabilities over various prediction cycles. The values in the emission matrix represent the probabilities associated with each observable state and are based on Pearson's Cross Correlation. The values in the transition matrix describe the user defined probabilities of going from one hidden state to another.

probabilities is added to the emission matrix, such that the emission matrix now has 1 column. The methodology for populating this column with probabilities is described in Section 5.4.5.

For any of the future prediction cycles given by $S = n$, s.t. $n > 1$, we calculate the probabilities associated with all possible sequences of states or locations that the user could have visited in the n-1 state transitions. The probability of a sequence of states given by $N = \{g_1, g_2, g_3 \ldots g_i \ldots g_n\}$ is computed as:

$$P(N) = \prod_{i=1}^{i=(n-1)} \left(E[N[i],\ i] \cdot E[N[i+1],\ i+1] \cdot T_r[i,\ i+1] \right) \tag{1}$$

where $E[N[i], i]$ represents the emission probability of the observed state at N[i] in the i^{th} prediction cycle, and $T_r[i,\ i+1]$ represents the transition probability of moving from state i to $i + 1$. The sequence of states or locations with the highest probability across various prediction cycles is reported as the path taken by the user, and the last state in the reported sequence is produced as the current location of the user. For example, as per the emission and transition probabilities given in Fig. 6, the most likely sequence of states or path taken by the user after 3 prediction cycles would be {1, 2, 2} (these are the states in the emission matrix with the highest probabilities for each of the first 3 prediction cycles) and after 5 prediction cycles would be {1, 2, 2, 4, 3}. In this manner, given the emission and transition matrices, the Viterbi algorithm is able to search for the most likely path taken by the user based on Eq. (1). More details on the Viterbi algorithm can be found in [40].

5. SHERPA-HMM framework

In this section, we first discuss the Wi-Fi fingerprinting phase (Section 5.1) and fingerprint pre-processing (Section 5.2) required by *SHERPA-HMM*. Section 5.3 describes the offline training phase database created in *SHERPA-HMM*. Section 5.4 describes the software-based *SHERPA-HMM* framework and its main components that are used in the online testing phase: a noise resilient fingerprint sampling, a pattern matching metric, HMM-based location predictor, and additional optimizations.

5.1. Wi-Fi fingerprinting

We utilize both the 2.4 GHz and 5 GHz Wi-Fi bands to capture the RSSI of a WAP along with its Media Access Control (MAC) address and the location (x-y coordinate) at which the sample (fingerprint) was taken. The MAC address allows us to uniquely identify a WAP. The average RSSI values for WAPs obtained through multiple scans at each location are stored in a tabular form, such that each row of RSSI values (fingerprint vector) characterizes a unique location. Fingerprints are collected along indoor paths with a smartphone. This step is essential for any fingerprinting technique.

Through SHERPA-HMM we aim deliver a lightweight indoor localization solution that is as good as, if not better, than the non-heterogenous case (KNN example in Section 3). Therefore, our goal was to eliminate the impact of heterogeneity from the indoor localization framework. Further, it should be noted that more complex frameworks may be able to deliver higher accuracies but that would come at a cost of longer prediction times that would further negatively impact the real-time behavior of the framework. Additionally, the achievable localization accuracy is also limited by other factors such as radio signal (Wi-Fi) density, sampling reference point granularity and choice of radio signal in use. The use of freely available Wi-Fi based radio signal in public buildings limits the achievable accuracy. Higher accuracies could be delivered by deploying custom radio beacons based on UWB or Bluetooth technologies, but at a high cost.

With these considerations as a guide, we decided to establish a realistic localization accuracy objective of 2 m which we try to achieve through the fingerprint sampling granularity of 1 m. The 2 m accuracy objective is small enough to differentiate the user from being in a corridor or a room and large enough to accommodate the saving that can be achieved through our lightweight framework.

5.2. Fingerprint database pre-processing

The captured fingerprints can be easily polluted by temporarily visible untrusted Wi-Fi hotspots. Utilizing such RSSI values in our fingerprints can significantly reduce the overall reliability and security of our localization framework. Therefore, we only capture and maintain RSSI values for trusted MAC addresses that are found to be reliable WAP sources (e.g., by checking for visible WAPs across several days and times-of-day). This pre-processing step helps to improve the overall stability of the *SHERPA-HMM* framework.

5.3. Sherpa-hmm offline/training phase

In the training phase, a dataset containing the means of all fingerprints taken at each sampled reference point (x-y coordinates shown as blue dots in Fig. 1) is established and is stored in a tabular form identical to the fingerprinting dataset. Instead of storing multiple RSSI vector fingerprints for each reference point location, the mean RSSI dataset represents a collection of RSSI vectors where the noise in individual samples has been averaged out. The noise in the training phase dataset is heavily dependent on the smartphone used (as was observed in Fig. 3). Therefore, storing the mean of RSSI vectors per reference point is an essential step to ensure the portability of the training database across heterogeneous mobile devices.

5.4. SHERPA-HMM online/testing phase

5.4.1. Motion-aware prediction deferral

Scanning for Wi-Fi fingerprints is one of the most energy intensive aspect of fingerprinting-based indoor localization frameworks. In the real-world, the user may choose to stop and look at the surroundings while on a path. Any Wi-Fi scans or location prediction cycles that may take place while the user has stopped would be wasted. To avoid such a scenario, *SHERPA-HMM* tracks the number of steps taken by the user as he or she walks along a path. *SHERPA-HMM* defers scanning for Wi-Fi fingerprints until it detects that a significant number of steps have been taken since the last location of the user was predicted. Based on the experiments performed in Section 7, we know that the average localization error over all paths for our framework is close to 2 m and also the average step length of 0.5 m can be assumed based on [21]. Therefore, *SHERPA-HMM* only scans for Wi-Fi fingerprints once the user has taken at least four steps since the last location prediction started. We utilized the default step detector in the Android API to achieve this functionality [22].

5.4.2. Noise resilient fingerprint sampling

Noise in the testing phase presents a problem as it leads to degraded localization accuracy. As observed in Fig. 3(c), scanned Wi-Fi fingerprints in the testing phase can be significantly impacted by noise. Also, the extent of noise observed varies from device to device. Therefore, the shape of a single offline (training) fingerprint, based on only one Wi-Fi scan, may not match that of the online (testing) fingerprint from a noisy device. To overcome this challenge, we propose a methodology to reduce the impact of observed noise across heterogeneous smartphones and establish a prominent pattern match across the training dataset and the online phase samples.

As previously addressed, the mean RSSI vectors shown in Fig. 3 are more reliable for establishing a pattern match across heterogeneous devices instead of individually scanned RSSI fingerprints. Furthermore, recent advances in smartphone technology have led to the development of robust Wi-Fi support in smartphones. From our preliminary experiments, we found that some smartphones (Table 1) can deliver up to 1 scan in a second. These observations support the idea of executing multiple Wi-Fi scans in the online phase and using their mean for each location prediction.

Our framework opportunistically increases the number of scans required per prediction from 1 to 3 using the approach described in the next section (section 5.4.3). Once multiple consecutive Wi-Fi scans are completed, their mean fingerprint is calculated and used to predict a user's location. The online phase mean fingerprint is compared with the mean fingerprint vectors from the offline database in the next step which uses Pearson's Cross-Correlation (*PCC*; discussed in Section 5.4.4). The location prediction is then made using a lightweight HMM model with PCC-based values embedded in the emission matrix (discussed in Section 5.4.5).

5.4.3. Smart noise reduction with boosted scans per prediction

The key motivation behind considering multiple Wi-Fi scans per location prediction is to overcome any unpredictable noise across fingerprints from heterogeneous devices. However, too many Wi-Fi scans can undesirably reduce the battery life of a smartphone. To strike a balance between battery life and indoor localization accuracy, *SHERPA-HMM* identifies situations in the localization process where consecutive fingerprints are noisy and lead to degraded localization performance. In such situations, *SHERPA-HMM* boosts the number of Wi-Fi scans per prediction from one to up to three scans. To achieve this, *SHERPA-HMM* keeps a track of two quantities: maximum movable distance (D_{max}) and consecutive scan distance threshold (*CSDT*).

The maximum distance a user can move within two consecutive predictions is limited. From preliminary analysis and our previous work [37], we found that in the situations where noisy fingerprints lead to highly erroneous localization predictions, the distance between consecutive predictions is over a threshold of distance a human can move in the allotted time. If the distance between consecutive location predictions is larger than D_{max}, its respective flag is set and *SHERPA-HMM* resorts to conducting a second scan. The maximum movable distance (D_{max}) threshold is governed by the following equation:

$$D_{max} = \left(T_{scan} + T_{predict}\right) \times S_{gait} \tag{2}$$

where T_{scan} and $T_{predict}$are the times to complete the consecutive Wi-Fi scans and to predict the user's location respectively, and S_{gait} is the average gait speed of the user. In our case, $T_{predict}$was not significantly variable across smartphones and therefore, an upper bound value for $T_{predict}$was empirically set to be 0.5 s for the devices shown in Table 1. Also, an upper bound gait speed of 2 m/s was used for S_{gait} based on a large-scale study performed on human gait speeds [23]. A preliminary analysis found that the time taken for 1 Wi-Fi scan (number of default scans) was heavily dependent on the smartphone being employed and even varied for each smartphone itself. Therefore, *SHERPA-HMM* utilizes a timer on the smartphone to record the time taken for consecutive Wi-Fi scans at run-time and uses that value as T_{scan} in Eq. (2).

The consecutive scan distance threshold (*CSDT*) is the maximum allowable noise across consecutive scanned fingerprints above which we label the fingerprints as noisy. The value of *CSDT* is estimated based on the Euclidian distance between the fingerprints collected by the training device at each reference point. The assumption is that if the noise over consecutive scans is low, consecutive Wi-Fi fingerprints captured by the same device should be very close in terms of Euclidian distance. Based on a preliminary analysis performed on the HTC and BLU devices (Fig. 4) the value of *CSDT* was set to 25 dB. For our setup with the *SHERPA-HMM* framework, if the Euclidian distance between the first two consecutive scans is above *CSDT*, the noise threshold flag is set, and a third Wi-Fi scan is conducted. The mean of all three Wi-Fi scans is then used to predict the user's location. However, it is important to note that some of the noise resilience comes from the use of HMMs, therefore noise threshold alone may not guaranty degraded localization performance.

If both the noise threshold flag and the distance threshold flags are set, then *SHERPA-HMM* resorts to conducting three scans per location prediction until at least one of the flags are reset. In contrast to our previous work *SHERPA* [37] that utilizes three scans per prediction by default, the revised *SHERPA-HMM* framework only utilizes one scan per prediction by default, two scans in the relatively uncommon case of highly noisy devices, and very rarely boosts up to three scans per prediction. In this manner, our revised framework delivers low-latency predictions in real-time. It is important to note that the second scan only occurs when the last two consecutive location predictions are too far away from each other, which usually only occurs when cheaper low-quality instruments are in use. The condition for a third scan is only met when there is sufficient noise in consecutive scans. The number of times a third scan actually got triggered was found to be very limited. Further, in the rare case that the user moves by a significant amount by the time the third scan finishes, the resultant location slightly lags behind. However, our Viterbi formulation is able to overcome this issue by gaining confidence from later prediction cycles.

5.4.4. Heterogeneity resilient pattern matching: pcc

Pearson's Cross-Correlation (*PCC*) [31] is measure of linear correlation between two vectors. It is a popular metric in the field of signal processing and pattern matching for voice. A 2D version of *PCC* is also used in image processing for template matching, a method used for identifying any incidences of a pattern or an object within a template image. *PCC* between a template vector (*T*) and a sample vector (*X*) can be expressed as:

$$PCC = \frac{cov(T,\ X)}{\sigma_T\ \sigma_X} \tag{3}$$

where, *cov(T, X)* represents the covariance and σ_T and σ_X are their respective standard deviations. *PCC* is limited to a range of −1 to 1, where the sign represents negative or positive linear relationship, respectively, and the magnitude represents the strength of a linear relationship. For our purposes, a positive high value of *PCC* would suggest a strong similarity between the template (offline database in our case) and the sample (online mean fingerprint in our case). From (3), we observe that *PCC* is directly proportional to covariance (dot product of fingerprints) and inversely proportion to the standard deviation of sample *X* and *T*. Therefore, a sample exhibiting a high level of covariance with the template and a low standard deviation is likely to produce a stronger *PCC*.

5.4.5. Shape similarity focused Hidden Markov Model

As discussed in Section 4, there are two inputs to a Hidden Markov Model: the transition matrix and the emission matrix. The transition matrix remains the same for a given path, whereas the emission matrix is updated and fed to the Viterbi algorithm in each prediction cycle.

The transition matrix describes the probability of moving from one location (hidden state) to the next. We set up the transition matrix such that a user at a location can move in any direction by two steps in each prediction cycle. For example, on a linear path a user at the location with label *l* has equal probability to go to the locations with label: $l-2$, $l-1$, l, $l+1$, $l+2$ (0.2 each) in the next prediction cycle.

The formulation of the emission matrix is the most critical component of the proposed framework. The emission matrix at any stage of the prediction cycle is given by $E\,[L \times S]$, where *L* is the number of locations and *S* is the number of Wi-Fi scans conducted so far. At each location prediction cycle once one or more Wi-Fi scans have been completed (as discussed in Section 5.4.3), the PCC for each of the RSSI vectors of training data and the online mean RSSI vector is calculated. These PCC values now form a column vector of length *L*. The PCC column vector is normalized such that the sum of its values is 1. The normalized PCC column vector is now appended at the end of the emission matrix and fed to the Viterbi algorithm along with the transition matrix. The Viterbi algorithm in turn produces a series of the most likely reference points or locations (Viterbi path) that the user has visited in the last S prediction cycles. The last location of the series of reference points is the predicted location of the user.

5.4.6. Optimizing emission matrix for prediction time

In the real-world, a user may walk a very long path before reaching their final destination. This would result in a very large emission matrix, as each location prediction event will add one new column to the emission matrix as discussed in Section 5 and represented in Fig. 6. This will improve the overall localization accuracy of the user at each prediction cycle, however, it will also slow down the time it takes to produce a location prediction.

Even though we expect the location prediction of the user to improve as the emission matrix size increases, it may take its toll on battery life and prediction time. Therefore, to maintain the QoS for the *SHERPA-HMM* framework, we limit the maximum number of columns for the emission matrix to a limit called Scan Memory (S_m). In the example shown in Fig. 6, we observe that the width (number of columns) of emission matrix increases by one in every prediction cycle up to the Scan Memory limit of three. Once this limit is reached (PC 3), in each consecutive prediction cycle, only the portion of emission matrix inside the scan memory window is passed on to the Viterbi algorithm. This process limits the width of the emission matrix to a constant. Now, as the prediction cycles progress, the emission probabilities from old prediction cycles are no longer in consideration, allowing SHERPA-HMM to “forget” past noisy observations.

Based on our analysis in Section 7, we set the S_m to a value of 3. In this manner, the Viterbi algorithm at max predicts the last 3 locations the user has been to, based on the last 3 Wi-Fi scan events. This optimization limits the location inference time in a predictable manner and in-effect optimizes our framework for energy consumption. Further, this optimization enables our framework to disregard any errors that may have been accumulated due to delays in Wi-Fi scans.

6. Experimental setup

6.1. Heterogeneous devices and fingerprinting

To investigate the impact of smartphone heterogeneity, we employed six different smartphones (shown in Table 1). This allows us to explore the impact of device heterogeneity based on varying chipsets and vendors. We created an Android application that recorded the x-y coordinate from the user and included a scan button. Once the scan button was pressed, multiple Wi-Fi scans were performed. The RSSI value and MAC address for each WAP were recorded in an SQLite database (Section 5.1), and then pre-processed (Section 5.2).

6.2. Indoor paths for localization benchmarking

We compared the accuracy and stability of *SHERPA-HMM* and frameworks from prior work on five indoor paths in different buildings at a University campus. These paths are shown in Fig. 1; with each fingerprinted location or reference point denoted by a blue dot. The path lengths varied between 60 and 80 m, and the number of visible WAPs along these paths varied from 78 to 218. Each path was selected due to its salient features that may impact indoor localization. The *Glover* building is one of the oldest buildings on campus and constructed from wood and concrete. This path is surrounded by a combination of labs that hold heavy metallic equipment as well as large classrooms with open areas. The Behavioral Sciences (*Sciences*) and Library (*Lib_Study*) are relatively new buildings on campus that have a mix of metal and wooden structures with open study areas and bookshelves. The *Engr_Office* path is on the second floor of the engineering building that is surrounded by small offices. The *Engr_Labs* path is in the engineering basement and is surrounded by labs consisting a sizable amount of electronic and mechanical equipment. Both engineering paths are in the vicinity of large quantities of metal and electronics that lead to noisy Wi-Fi fingerprints and can hinder indoor localization. A total of 6 users, each carrying a smartphone from a different vendor, walked on each indoor path and collected samples (fingerprints) for each location on that path. This set of data was utilized in the training phase. For the testing/online phase, each of these 6 users walked on each of these paths in a random manner, generating 10 walks each varying from 20 to 50 m in length.

6.3. Comparison with prior work

We selected four prior works to compare against *SHERPA-HMM*. The first work (LearnLoc/KNN [3]) is a lightweight non-parametric approach based on the idea that similar data when observed as points in a multi-dimensional space would be clustered together. Thus, given a vector of Wi-Fi fingerprints in the testing phase, KNN identifies the K closest fingerprints based on Euclidean distance within its training model and produces the weighted sum of the coordinates of those K fingerprints. The second work (Rank Based Fingerprinting (RBF) [24]) claims that the rank of WAPs in a vector of ranked WAPs based on RSSI values remains stable across heterogeneous devices. It is functionally similar to KNN with the only difference being that each RSSI fingerprint vector in the training and testing phases is sorted and re-populated to store the rank of WAPs instead of raw RSSI values. The third work combines Procrustes analysis and Weighted Extreme Learning Machines (WELM) [22] to predict the location of a user. Procrustes analysis allows the technique to scale and superimpose the RSSI fingerprints of heterogeneous devices and denote the strength of this superimposition as the Signal Tendency Index (STI). The STI metric is used to transform the original RSSI fingerprints, and then used to train a WELM model in the online phase (STI-WELM) with the help of cloud servers. Lastly, we also compare *SHERPA-HMM*, to our previous work *SHERPA* [37], that utilizes a Pearson Correlation-based pattern matching metric to identify locations that are associated with offline Wi-Fi fingerprints, and employs lightweight optimizations to deliver high accuracy indoor localization predictions in real-time.

7. Results

7.1. Sensitivity analysis on scans per prediction

To quantify the potential improvement of using mean RSSI vectors in our framework, we conducted a sensitivity analysis to compare the accuracy results for *SHERPA-HMM* using a single RSSI vector and the vectors formed by considering the mean of 1 to 5 scanned fingerprints. Fig. 7 depicts the overall localization error for various values of scans per prediction over individual benchmark paths. Even though the overall errors for the *Engr_Office* and *Glover* paths are significantly lower than the other paths (discussed further in Section 7.3), there is a similar trend in reduction of localization error for all paths as the number of scans per prediction increases. The most significant reduction is observed when moving from 1 to 2 scans per prediction, whereas there is almost no reduction as we move from 4 to 5 scans. This observation solidifies our claim of improvement in accuracy by using more than one scans per prediction, as was discussed in detail in Section 5.4.2.

It is important to note that scans per prediction not only impacts the localization accuracy but also the energy consumed per prediction. A single Wi-Fi scan can consume a notable amount of energy (∼2400mJ when using LG). This motivated us to explore the most suitable value of maximum scans per prediction for *SHERPA-HMM*’s online phase. If the value is too small, such as the case for the *Lib_Study* path in Fig. 7, there might not be a significant improvement in localization accuracy.

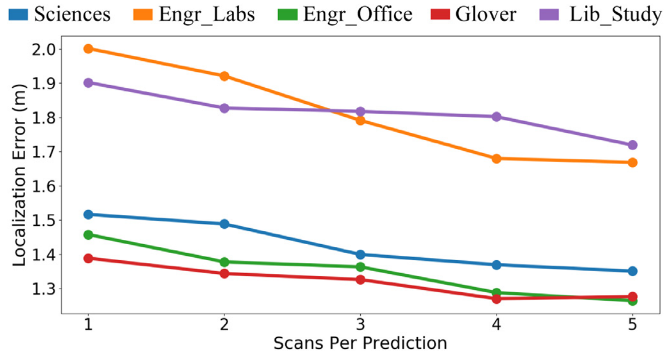
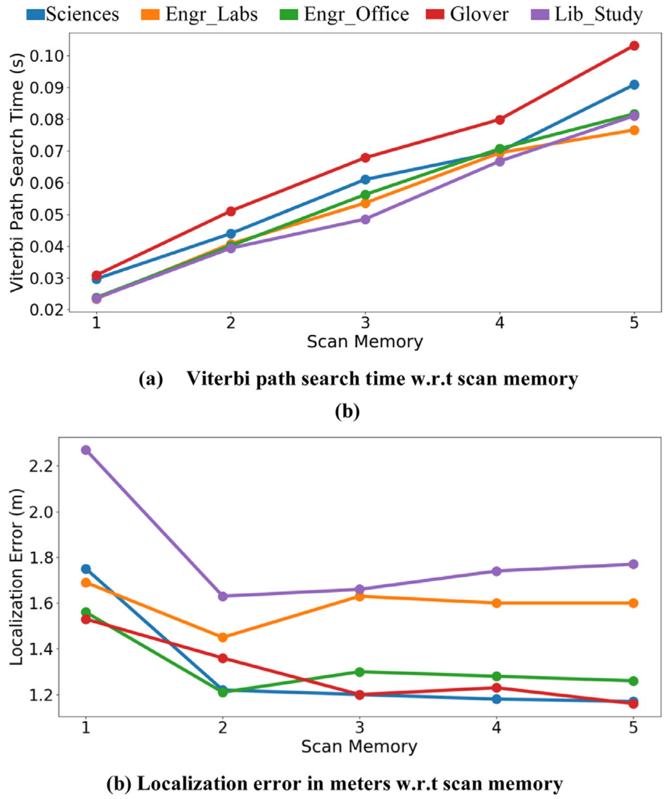


Fig. 7. Variation in localization error for different values of scans per prediction (x axis) across various path benchmarks.



(a) Viterbi path search time w.r.t scan memory

(b) Localization error in meters w.r.t scan memory

Fig. 8. Variation in localization error and Viterbi path search time over scan memory for various benchmark paths.

However, if the value is too large, the smartphone may end up consuming a significant amount of energy for an insignificant improvement. From Fig. 7, we observe that for most benchmark paths, a majority of the improvement is achieved by conducting only 3 consecutive scans. Therefore, the upper limit on scans per prediction is set to 3 for our framework. We increase the number of scans per prediction from 1 to 3 in an intelligent manner, as discussed in Section 5.4.3.

7.2. Sensitivity analysis on scan memory

The scan memory variable discussed in Section 5.4.6 can significantly impact the performance characteristics of the proposed *SHERPA-HMM* framework. To quantify this, we perform a sensitivity analysis on the scan memory variable in an effort strike a balance between prediction latency and localization accuracy.

Fig. 8(a) and (b) present the trends on Viterbi path search times and average localization error across all devices on various paths in our benchmark suite. For this experiment, we analyze the change in Viterbi path search time and localization error when the scan memory (emission matrix width) ranges from 1 to 5. Setting the value of 1 for scan memory translates into only using the latest Wi-Fi scan for location prediction without any historical knowledge, whereas a value of 5 suggests that the latest Wi-Fi scan along with previous four Wi-Fi scan events were utilized to identify the current location. The results for this experiment were averaged out over all the devices.

From Fig. 8(a), we observe that the time taken by the Viterbi algorithm to deduce the most likely path taken increases linearly as scan memory is increased in the range from 1 to 5. This trend is consistent across the paths. We observe that the overall search time is generally the highest for the Glover path. This is mainly due to the fact that the Glover path is the longest benchmark path with 88 reference locations. Each reference location translates into a unique state in the Hidden Markov model. This increases the number of rows in the emission matrix. In Fig. 8(a), we also observe that the search time grows by 5× as scan memory is increased from 1 to 5.

From Fig. 8(b), we observe that as we increase scan memory the drop in localization error is most significant up to the point where scan memory is 3, beyond which we observe diminishing returns. Another notable aspect is that the most improvement is observed in the Lib_Study path. This can be attributed to the fact that the Lib_Study has a more complex zig-zag like path. This observation also highlights the prospective improvements that can be gained by using HMM models in more complex paths and dynamically increasing scan memory at run-time in an intelligent manner.

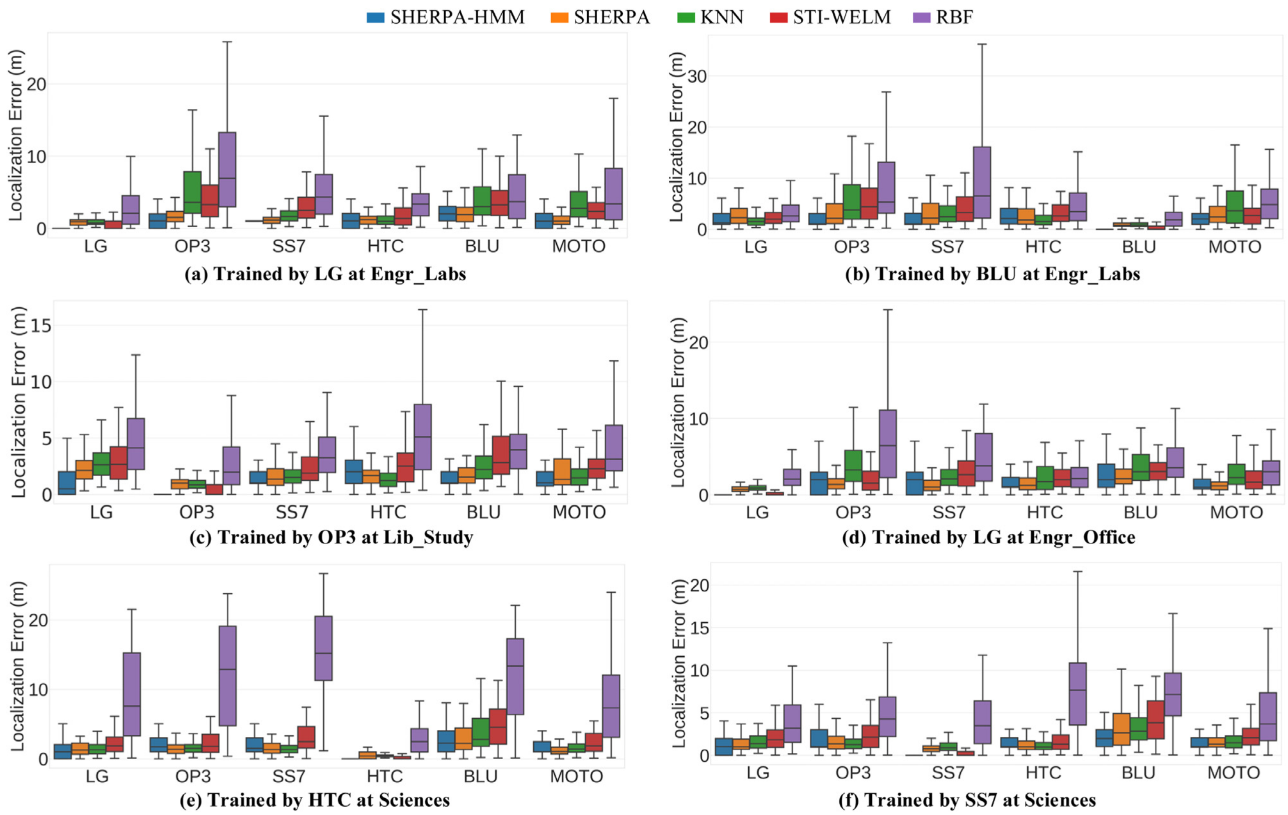
From our observations in Fig. 8(a) and 8(b), we set the value of scan memory for our HMM formulation to 3. This allows us to minimize the localization error without significantly impacting the overall prediction time of our proposed indoor localization framework. It is also important to note that the value of scan memory that delivers the best accuracy highly depends on the state space of the path. The user is responsible for identifying a good value of state space for each path individually.

7.3. Performance of localization techniques

Fig. 9 shows the individual plots that represent the contrast in the localization experiences of six users carrying smartphones from distinct vendors. The paths along with the training phase device combinations were chosen based on the analysis of the plots in Fig. 2. We focus on a subset of cases that demonstrate significant deterioration in error (> 2 m) for the KNN technique.

From Fig. 9(a), it can be observed that HTC is the most stable device for KNN, i.e., is least affected by heterogeneity, by delivering an average accuracy close to 2 m. In all other situations, localization error is heavily impacted by heterogeneity. Overall, in Figs. 9(a) and (b), *SHERPA-HMM* can be seen to outperform RBF and STI-WELM with average accuracies in the range of 1–2 m whenever the localization error from KNN is > 2 m. *SHERPA-HMM* is also better than our *SHERPA* in most cases. We observe that RBF performs the worst when there is a significant amount of metal structures in the environment. This is the case for the engineering building paths (*Engr_Labs, Engr_Office*) and the path in the *Sciences* building. The perturbations in the Wi-Fi WAP RSSI values due to the metallic surroundings cause the ranks of the WAP RSSI values to become highly unstable. We noted that RBF performed better than KNN for a few walks, but this was averaged out by poor results from other iterations of the same walk.

From Fig. 9, we also observe that *SHERPA-HMM* outperforms STI-WELM in most training-testing device pairs, other than the non-heterogeneous cases (e.g., LG boxplot in 9(a), BLU boxplot in 9(b), etc.). *SHERPA-HMM* is able to deliver better performance in most cases as it is a purely pattern matching approach along a path. STI-WELM identifies the closest sampled locations from the offline phase using the scaling and shape matching based STI metric. The fingerprints of these closest locations are then used to train a WELM based neural network in the



(a) Trained by LG at Engr_Labs

(b) Trained by BLU at Engr_Labs

(c) Trained by OP3 at Lib_Study

(d) Trained by LG at Engr_Office

(e) Trained by HTC at Sciences

(f) Trained by SS7 at Sciences

Fig. 9. Localization error for various techniques on benchmark paths across training devices.

online phase. The work in [20] (STI-WELM) assumes a constant gain across heterogeneous devices which is not the case (from Fig. 2) and does not compensate for noise across smartphones. The neural network model itself is not especially designed for pattern matching, and sacrifices predictability of localization error for faster training time in the online phase. Further, a neural network-based localization framework such as STI-WELM requires extremely large sets of training data which may not a be a realistic and scalable approach for indoor environments. In the few cases that *SHERPA-HMM* is outperformed by STI-WELM, *SHERPA-HMM* still performs within the acceptable range of accuracy and is very close to STI-WELM in terms of median error. We also note that for most paths considered in Fig. 9, *SHERPA-HMM* outperforms KNN. In the few cases where it is outperformed by KNN, its accuracy loss is very low.

In some of the cases such as in Fig. 9(d), we observe that *SHERPA-HMM* delivers relatively higher localization error as compared to *SHERPA*. We found that the major cause of this was that the HMM model falsely predicts that a user has turned back when the user is actually moving forward along a path. This is caused by noisy fingerprints and the fact that we are using a simple transition matrix where the probability of the user moving in any direction is the same. Also, we do not utilize other motion sensors such as magnetic and gyroscope to identify situations where the user is changing directions [35]. However, even with this drawback *SHERPA-HMM* is able to meet our target accuracy of 2 m across the board.

The experiments performed in this work revealed that certain devices such as the low-cost BLU smartphone produce particularly noisy and inconsistent Wi-Fi RSSI measurements. Even though *SHERPA-HMM* attempts to minimize the impact of noise by taking into account multiple Wi-Fi scans for each location prediction, users should be wary of the quality limitations of such low-cost devices, especially when using them for indoor localization and navigation.

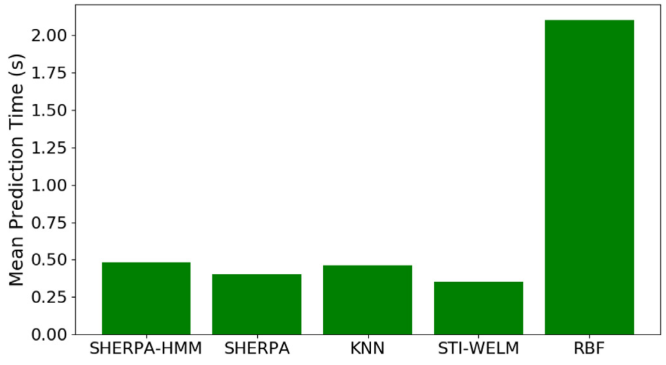


Fig. 10. Mean indoor location prediction time for *SHERPA-HMM* and frameworks from prior work for the *Lib_Study* path using the OnePlus3 device.

7.4. Comparison of execution times

To highlight the lightweight design of our approach, we show the mean execution time of location predictions for *SHERPA-HMM* and prior work frameworks executing on the OP3 device. For brevity, results for only one path (*Lib_Study*) are shown. The specific path was chosen for this experiment as it was the largest one with 13,080 data points (60 *m* × 218 WAPs) available. The OP3 device was randomly chosen as we expect the overall trends of this experiment to remain the same across smartphones.

The results of this experiment are shown in Fig. 10. The RBF technique is found to take over 2 s to execute. This behavior can be attributed to the fact that RBF requires sorting of Wi-Fi RSSI values for every scanned fingerprint in the testing phase, unlike any of the other techniques. STI-WELM takes the least time to predict locations. However, the highly degraded accuracy with STI-WELM, especially in the presence of device heterogeneity (as seen in Fig. 9) is a major limitation for STI-WELM. After STI-WELM (Fig. 10), *SHERPA* is one of the quickest localization frameworks with an average prediction time of 0.43 s that is slightly lower than the lightweight Euclidean-based KNN approach that takes 0.47 s for a prediction. Finally, *SHERPA-HMM* delivers its prediction results in 0.48 s which is only slightly higher than KNN. As compared to *SHERPA, SHERPA-HMM* takes ~0.05 s longer but has proven to deliver significantly better results as shown in Section 7.3.

In summary, from the results presented in this section, it is evident that our proposed *SHERPA-HMM* framework for is a promising approach that provides highly accurate, lightweight, smartphone heterogeneity-resilient indoor localization. A major strength of this framework is that it can be easily ported across smartphones without the need of any calibration effort or cloud-based service to execute.

8. Conclusion and future work

In this paper, we proposed the *SHERPA-HMM* framework that is a computationally lightweight solution to the mobile device heterogeneity problem for fingerprinting-based indoor localization. Our analysis in this work provides important insights into the role of mobile device heterogeneity on localization accuracy. *SHERPA-HMM* was able to deliver superior levels of accuracy as compared to state-of-the-art indoor localization techniques using only a limited number of samples for each fingerprinting location. We also established that developing algorithms that can be easily ported across devices with minimal loss in localization accuracy is a crucial step towards the actuation of fingerprinting-based localization frameworks in the real world.

As part of our future work, we would like to focus on improving the reliability of the proposed framework through incorporating inertial and magnetic information in the HMM formulation. This would greatly reduce the chances of the Viterbi algorithm predicting false user movement direction changes. Another improvement could be to dynamically increase the scan memory variable such that user predictions are made with higher confidence in situations where the online fingerprint is noisy.

References

[1] "Enterprise Indoor LBS Market in the US 2016-2020", 2016 [online] technavio.com/report/usa-machine-machine-m2m-and-connected-devices-enterprise-indoor-location-based-services

[2] C. Langlois, S. Tiku, S. Pasricha, Indoor localization with smartphones, IEEE Consum. Electron. 6 (4) (2017).

[3] S. Pasricha, V. Ugave, Q. Han, C. Anderson, LearnLoc: a framework for smart indoor localization with embedded mobile devices,, Proceedings of International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2015.

[4] A. Mittal, S. Tiku, S. Pasricha, Adapting convolutional neural networks for indoor localization with smart mobile devices, in: Proceedings of Great Lakes Symposium on VLSI (GLSVLSI), 2018.

[5] V. Singh, G. Aggarwal, B.V.S. Ujwal, Ensemble based real-time indoor localization using stray WiFi signal, in: Proceedings of International Conference on Consumer Electronics (ICCE), 2018.

[6] X. Wang, X. Wang, S. Mao, CiFi: deep convolutional neural networks for indoor localization with 5GHz Wi-Fi, in: Proceedings of International Conference on Communications (ICC), 2017.

[7] W. Xue, X. Hua, Q. Li, K. Yu, W. Qiu, Improved neighboring reference points selection method for wi-fi based indoor localization, Sens. Lett. 2 (2) (2018) 1–4.

[8] J. Park, D. Curtis, S. Teller, J. Ledlie, Implications of device diversity for organic localization, in: Proceedings of INFOCOM, 2011.

[9] H. Zou, B. Huang, X. Lu, H. Jiang, L. Xie, A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine, Trans. Wirel. Commun. 15 (2) (2016) 1252–1266.

[10] K. Chintalapudi, A. Padmanabha Iyer, V.N. Padmanabhan, Indoor localization without the pain, in: Proceedings of Mobile Computing Networks, 2010.

[11] J. Schmitz, M. Hernández, R. Mathar, Real-time indoor localization with TDOA and distributed software defined radio: demonstration abstract, in: Proceedings of Information Processing in Sensor Networks (IPSN), 2016.

[12] J. Xiong, K. Jamieson, Towards fine-grained radio-based indoor location, in: Proceedings of Mobile Computing Systems & Applications (HotMobile), 2012.

[13] E. Soltanaghaei, A. Kalyanaraman, K. Whitehouse, Multipath triangulation: decimeter-level wifi localization and orientation with a single unaided receiver, in: Proceedings of Mobile Systems, Applications, and Services (MobiSys), 2018.

[14] M.B. Kjærgaard, Indoor location fingerprinting with heterogeneous clients, Pervas. Mob. Comput. 7 (1) (2010) 31–43.

[15] C. Figuera, J.L. Rojo-Alvarez, I. Mora-Jimenez, A. Guerrero-Curieses, M. Wilby, J. Ramos-Lopez, Time-space sampling and mobile device calibration for Wi-Fi indoor location systems, Trans. Mob. Comput. 10 (7) (2011) 913–926.

[16] A. Haeberlen, E. Flannery, A.M. Ladd, A. Rudys, D.S. Wallach, L.E. Kavraki, Practical robust localization over large-scale 802.11 wireless networks, in: Proceedings of Mobile computing and networking (MobiCom), 2004.

[17] S. Fang, C. Wang, S. Chiou, P. Lin, Calibration-free approaches for robust wi-fi positioning against device diversity: a performance comparison, in: Proceedings of Vehicular Technology Conference (VTC), 2012.

[18] M.B. Kjærgaard, C.V. Munk, Hyperbolic location fingerprinting: a calibration-free solution for handling differences in signal strength, in: Proceedings of Pervasive Computing and Communications (PerCom), 2008.

[19] A.K.M. Mahtab Hossain, Y. Jin, W. Soh, H.N. Van, SSD: a robust RF location fingerprint addressing mobile devices' heterogeneity, Trans. Mob. Comput. 12 (1) (2013) 65–77.

[20] H. Zou, B. Huang, X. Lu, H. Jiang, L. Xie, A robust indoor positioning system based on the procrustes analysis and weighted extreme learning machine, Trans. Wirel. Commun.s 15 (2) (2016) 1252–1266.

[21] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, F. Zhao, A reliable and accurate indoor localization method using phone inertial sensors, in: Proceedings of Ubiquitous Computing (UbiCom), 2012.

[22] "Android Step Counter API", 2018 [online]. Available: https://developer.android.com/reference/android/hardware/Sensor#TYPE_STEP_C-OUNTER

[23] R.C. Browning, E.A. Baker, J.A. Herron, R. Kram, Effects of obesity and sex on the energetic cost and preferred speed of walking, J. Appl. Physiol. 100 (2) (2006) 390–398.

[24] J. Machaj, P. Brida, R. Piché, Rank based fingerprinting algorithm for indoor positioning, in: Proceedings of Indoor Positioning and Indoor Navigation (IPIN), 2011.

[25] B. Donohoo, C. Ohlsen, S. Pasricha, A middleware framework for application-aware and user-specific energy optimization in smart mobile devices, J. Pervas. Mob. Comput. 20 (2015) 47–63.

[26] B. Donohoo, C. Ohlsen, S. Pasricha, C. Anderson, Y. Xiang, Context-aware energy enhancements for smart mobile devices, Trans. Mob. Comput. 13 (8) (2014) 1720–1732.

[27] B. Donohoo, C. Ohlsen, S. Pasricha, C. Anderson, Exploiting spatiotemporal and device contexts for energy-efficient mobile embedded systems, in: Proceedings of Design Automation Conference (DAC), 2012.

[28] B. Donohoo, C. Ohlsen, S. Pasricha, AURA: an application and user interaction aware middleware framework for energy optimization in mobile devices, in: Proceedings of International Conference on Computer Design (ICCD), 2011.

[29] S. Tiku, S. Pasricha, Energy-efficient and robust middleware prototyping for smart mobile computing, in: Proceedings of International Symposium on Rapid System Prototyping (RSP), 2017.

[30] S. Pasricha, J. Doppa, K. Chakrabarty, S. Tiku, D. Dauwe, S. Jin, P. Pande, Data analytics enables energy-efficiency and robustness: from mobile to manycores, datacenters, and networks, Proceedings of International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2017.

[31] F.L. Coolidge, An introduction to correlation and regression, Statistics: A Gentle Introduction (2012) 211–267.

[32] Q. Lu, X. Liao, S. Xu, W. Zhu, A hybrid indoor positioning algorithm based on WiFi fingerprinting and pedestrian dead reckoning, in: Proceedings of Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), 2016.

[33] Y. Hu, X. Liao, Q. Lu, S. Xu, W. Zhu, A segment-based fusion algorithm of WiFi fingerprinting and pedestrian dead reckoning, in: Proceedings of International Conference on Communications in China (ICCC), 2016.

[34] U. Bolat, M. Akcakoca, A hybrid indoor positioning solution based on Wi-Fi, magnetic field, and inertial navigation, in: Proceedings of Workshop on Positioning, Navigation and Communications (WPNC), 2017.

[35] S. Lamy-Perbal, N. Guénard, M. Boukallel, A. Landragin-Frassati, An HMM map–matching approach enhancing indoor positioning performances of an inertial measurement system, in: Proceedings of International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2015.

[36] C. Ascher, C. Kessler, R. Weis, G.F. Trommer, Multi-floor map matching in indoor environments for mobile platforms, in: Proceedings of International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2012.

[37] S. Tiku, S. Pasricha, B. Notaros, Q. Han, SHERPA: a lightweight smartphone heterogeneity resilient portable indoor localization framework, in: Proceedings of International Conference on Embedded Software and Systems (ICESS), 2019.

[38] B. Yoon, P.P. Vaidyanathan, Context-sensitive hidden markov models for modeling long-range dependencies in symbol sequences, Trans. Signal Process. 54 (11) (2006) 4169–4184.

[39] M.I. Mohd-Yusoff, I. Mohamed, M.R.A. Bakar, Hidden Markov models: an insight, in: Proceedings of International Conference on Information Technology and Multimedia, 2014.

[40] D. Han, H. Rho, S. Lim, HMM-based indoor localization using smart watches' ble signals, in: Proceedings of International Conference on Future Internet of Things and Cloud (FiCloud), 2018.

[41] S. Poh, Y. Tan, X. Guo, S. Cheong, C. Ooi, W. Tan, LSTM and HMM comparison for home activity anomaly detection, in: Proceedings of Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), 2019.

[42] K. Oura, K. Tokuda, J. Yamagishi, S. King, M. Wester, Unsupervised cross-lingual speaker adaptation for HMM-based speech synthesis, Proceedings of International Conference on Acoustics, Speech and Signal Processing, 2010.

[43] H. Zou, et al., WinIPS: wiFi-based non-intrusive indoor positioning system with online radio map construction and adaptation, Trans. Wirel. Commun. 16 (12) (2017) 8118–8130.

[44] Y. Li, S. Williams, B. Moran, A. Kealy, A probabilistic indoor localization system for heterogeneous devices, Sens. J. 19 (16) (2019) 6822–6832.

[45] A. Mathur, T. Zhang, S. Bhattacharya, P. Velickovic, L. Joffe, N.D. Lane, F. Kawsar, P. Lio, Using deep data augmentation training to address software and hardware heterogeneities in wearable and smartphone sensing devices, in: Proceedings of International Conference on Information Processing in Sensor Networks (IPSN), 2018.

[46] A. Mathur, A. Isopoussu, F. Kawsar, N. Berthouze, N.D. Lane, Mic2Mic: using cycle-consistent generative adversarial networks to overcome microphone variability in speech systems, in: Proceedings of International Conference on Information Processing in Sensor Networks (IPSN), 2019.

Saideep Tiku is a Ph.D. student in the ECE Department at Colorado State University, Fort Collins, Colorado, USA. His research interests include indoor localization, and energy efficiency for fault tolerant embedded systems. He is a Student Member of the IEEE.

Sudeep Pasricha received his Ph.D. in computer science from the University of California, Irvine in 2008. He is currently a Rockwell-Anderson Professor of ECE at Colorado State University. His-research interests include energy-efficiency and fault-tolerance for embedded and mobile computing. He is a Senior Member of the IEEE.

Branislav M. Notaroš received the Dipl.Ing. (B.S.), M.S., and Ph.D. degrees in electrical engineering from the University of Belgrade, Belgrade, Yugoslavia, in 1988, 1992, and 1995, respectively. He is currently a Professor with the Department of Electrical and Computer Engineering, Colorado State University (CSU), His research interests and activities are in computational electromagnetics, higher order numerical methods, antennas, scattering, microwaves, metamaterials, characterization of snow and rain, surface and radar precipitation measurements, RF design for MRI at ultra-high magnetic fields, and electromagnetics education.

Qi Han received the Ph.D. degree in computer science from the University of California, Irvine, in 2005. Currently, she is an associate professor with the Department of Electrical Engineering and Computer Science, Colorado School of Mines, Colorado. Her research interests include distributed systems, middleware, mobile and pervasive computing, dynamic data management, and cyber physical systems. She was the program chair of IEEE PerCom'16 and MobiSPC'14. She is a member of the IEEE and the ACM.