SHERPA: A Lightweight Smartphone Heterogeneity Resilient Portable Indoor Localization Framework

Saideep Tiku, Sudeep Pasricha, Branislav Notaros
Department of Electrical and Computer Engineering
Colorado State University, Fort Collins, CO, U.S.A.
{saideep, sudeep, notaros}@colostate.edu

Qi Han
Department of Computer Science
Colorado School of Mines, Golden, CO, U.S.A.
qhan@mines.edu

***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* 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.**

I. 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 [3-7]. 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].

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, *S*martphone *He*terogeneity *R*esilient *P*ortable (SHERPA) Wi-Fi RSSI fingerprinting framework that is portable across smartphones with minimal accuracy loss. In summary, 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 design the SHERPA framework for portable Wi-Fi fingerprinting-based indoor localization; SHERPA employs a lightweight software-based approach to combine noisy fingerprints over distinct smartphones and pattern matching/filtering to improve location accuracy;
- We evaluate SHERPA against state-of-the-art localization techniques, across a variety of Android-based smartphones that are used for indoor localization along paths in different buildings.

II. BACKGROUND AND RELATED WORK

Since the establishment of RF 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. 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 investigated multipath effects for triangulation

978-1-7281-2437-7/19/$31.00 ©2019 IEEE

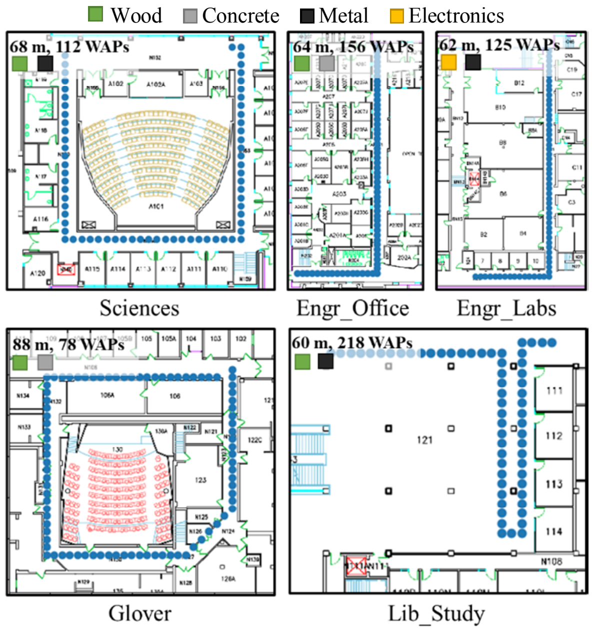


Figure 1: Benchmark paths for indoor localization (with path lengths and WAP density, and salient path features).

Table I: 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

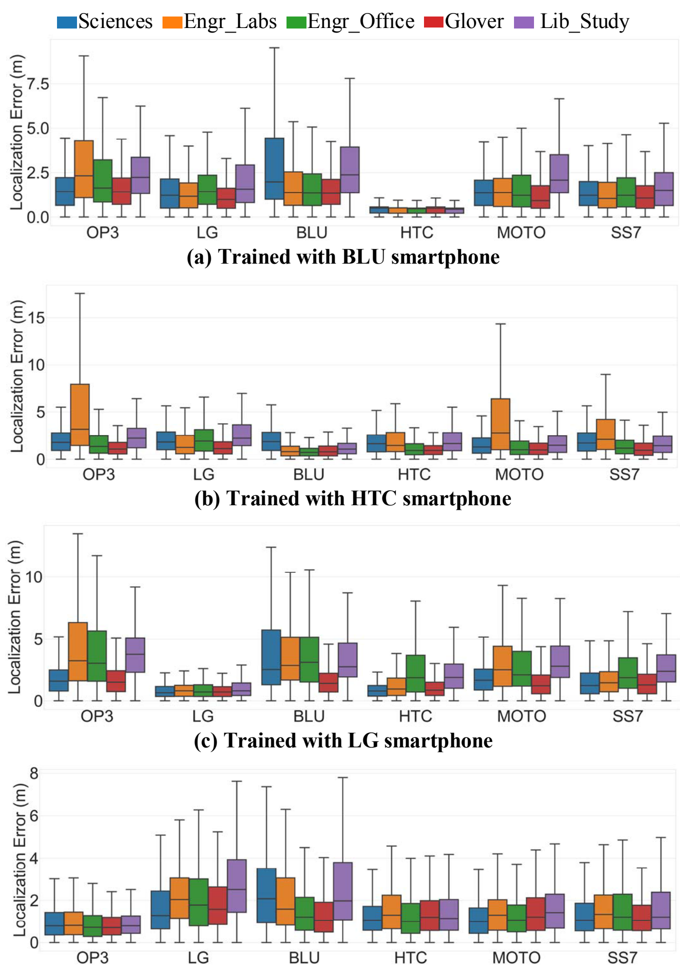
[13], but most work does not apply to commodity smartphones and hence, has limited scalability.

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. Another issue with such techniques is their severe energy imposition on mobile devices. In [3], an energy-efficient 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.

The most intuitive approach for calibration to address device heterogeneity is to acquire RSSI values and location data manually for each new mobile device [14]. This is however 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. 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.

In contrast, our *SHERPA* 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.



(a) Trained with BLU smartphone

(b) Trained with HTC smartphone

(c) Trained with LG smartphone

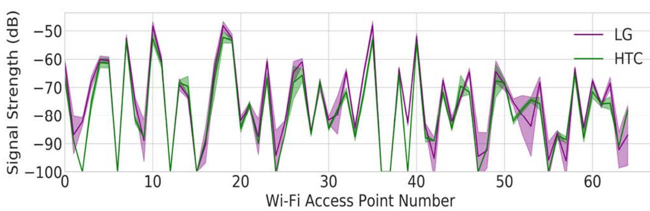
(d) Trained with OP3 smartphone

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

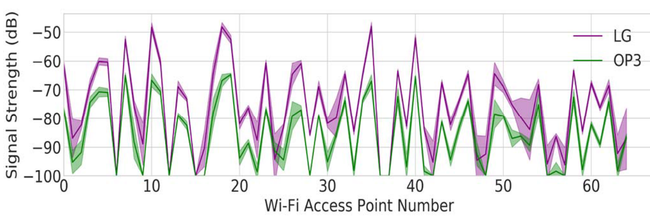
III. Heterogenous Fingerprint Analysis

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 (Figure 1) using six distinct devices (Table I).

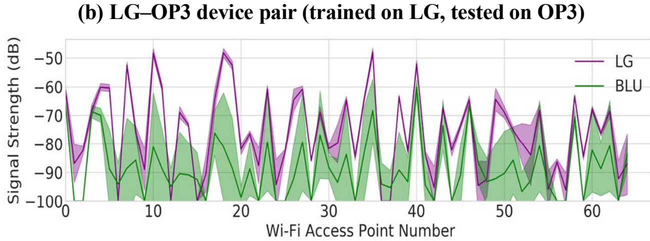
Figure 2 shows the boxplots (distribution) for localization error (in the online/testing phase) across all smartphones and indoor paths, for four scenarios where the KNN model was trained on four different smartphones. The most interesting observation is that, in general, the least error is achieved 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 meters) when trained and tested with OP3 on all paths (figure 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 (figure 2(c)). For the *Engr_Labs* path in figure 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.*



(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)

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

The RSSI values for the best and the two poorly performing training-testing device pairs are shown in figure 3. The solid lines represent the mean values, whereas the shaded regions represent the standard deviations of RSSI values. From figure 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 figure 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 figure 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 figure 3 is that the overall shape of the fingerprints is similar, including in figure 3(c), where the shape is similar to the mean fingerprint for the BLU device.

From figure 3(c), the greater amount of noise from the BLU device is apparent as compared to other devices, such as HTC. Identifying and quantifying such noise in the online phase 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.

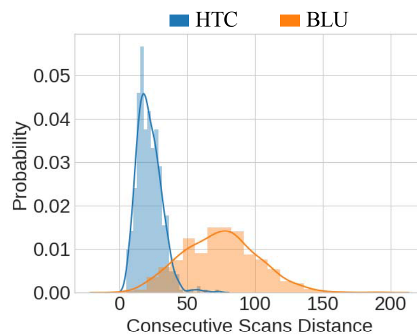


Figure 4: Probability distribution of the Euclidian distance across consecutive pairs of scans using the HTC and BLU smartphones on the *Engr_Labs* indoor path.

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. Figure 4 depicts the distribution of the Euclidian distance between consecutively captured Wi-Fi fingerprints for the BLU and HTC device over the *Engr_labs* path. From figure 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 heterogeneous 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 primary motivation for our proposed *SHERPA* framework for lightweight and portable localization, as discussed next.

IV. SHERPA Framework

In this section, we first discuss the Wi-Fi fingerprinting phase (section IV.A) and fingerprint pre-processing (section IV.B) required by *SHERPA*. Section IV.C describes the offline training phase database created in *SHERPA*. Section IV.D describes the *SHERPA* software-based framework and its main components: a noise resilient fingerprint sampling, a pattern matching metric, a preliminary location predictor, and a prediction filter; that are used in the online testing phase.

A. 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. Note also that the deliverable accuracy from any fingerprinting-based approach is correlated to the granularity of sampling along a path. We chose to fingerprint at 1-meter intervals along indoor paths, with the eventual goal of achieving a localization accuracy of within 2 meters.

B. 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/times-of-day). This pre-processing step helps to improve the overall stability of the *SHERPA* framework.

C. SHERPA 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 figure 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 figure 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.

D. SHERPA Online/Testing Phase

D.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* tracks the number of steps taken by the user as he or she walks along a path. *SHERPA* 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 6, we know that the average localization error over all paths for our framework is close to 2 meters and also the average step length of 0.5 meters can be assumed based on [21]. Therefore, *SHERPA* 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].

D.2. Noise Resilient Fingerprint Sampling

Noise in the testing phase presents a problem as it leads to degraded localization accuracy. As observed in figure 3(c), scanned Wi-Fi fingerprints in the testing phase can be significantly impacted by noise. Also, the shape of a single offline (training) fingerprint, based on only one Wi-Fi scan, may not match that of the online (testing) fingerprint. 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 figure 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 I) 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 reduces the number of scans required per prediction to two using the approach described in the next section (section IV.D.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 IV.D.4). A preliminary location prediction is then made using a lightweight reference point selection process (discussed in section IV.D.5). The final location prediction in the online phase is made after passing the preliminary location from the previous step through a filter (discussed in section IV.D.6).

Table II. Description of symbols used to describe *SHERPA* framework

Symbol	Description
τ	Opportunistic scan reduction threshold
T	Template (training) RSSI vector
X	Sample (online) RSSI vector
σ_T	Template vector standard deviation
σ_X	Sample vector standard deviation
Z_{xy}	Z-Score at reference point (*x, y*)
L_{xy}	Location of reference point (x, y)
Φ	Reference point selection cut-off
μ_Φ	Mean location of top Φ reference points
σ_Φ	Standard deviation of top Φ reference points
D_{max}	Maximum movement between consecutive scans
T_{scan}	Time taken for Wi-Fi scan to execute
$T_{predict}$	Time taken to make location prediction
S_{gait}	User gait speed (walking speed)

D.3. Opportunistic Reduction in 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 conserve battery, *SHERPA* opportunistically reduces the number of scans per prediction cycle by checking for noise over consecutive Wi-Fi scans. If the noise over consecutive scans is determined to be below a certain threshold (τ), further scanning is skipped to save energy. The value of τ 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 (figure 4) the value of τ was set to 25dB. For our setup with the *SHERPA* framework, if the Euclidian distance between the first two consecutive scans is below τ, a third scan is not performed, thereby saving one-third of the energy spent on Wi-Fi scanning in a given prediction cycle.

D.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:

$$PCC = \frac{cov(T,X)}{\sigma_T\, \sigma_X} \quad (1)$$

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 (1), 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*.

D.5. PCC-based Reference Point Selection

Once we have the *PCC* values associated with each reference point from the previous step, we sort them in descending order and select the top Φ *PCC* values that are associated with reference points that have a similar shape to the mean testing (online) fingerprint. The top Φ fingerprints are then passed through a light-weight Z-score based outlier detection algorithm. As shown in figure 5, the selected top Φ reference points (red and green "+" symbols in the figure) may include some outliers (red "+" symbols). To address this issue, the weighted sum of the top Φ reference points is taken to produce the initial predicted location (depicted by the red star symbol in the figure). Based on this mean location (red star symbol) and the standard deviation of the top Φ reference points, a Z-score is calculated for each selected reference point using the following equation:

$$Z_{xy} = \frac{L_{xy} - \mu_\Phi}{\sigma_\Phi} \quad (2)$$

where L_{xy} is the reference point at location (x, y), μ_Φ is the weighted mean location represented by the red star symbol in figure 5, and σ_Φ is the standard deviation of the coordinates of the top Φ reference points (red and green "+" symbols in figure 5). In statistics, Z-score is used as a tool to describe deviation of a sample from its distribution's mean. Therefore, a reference point with a high Z-score would be an outlier from the cluster of selected reference points.

Next, we compute the weighted-sum of all reference point locations that are above a Z-score threshold to establish a preliminary location prediction (shown by the blue star symbol in figure 5). The values of Φ and the Z-score threshold were empirically established after analyzing data across multiple indoor paths that we considered for our study. The computed weighted sum generates a preliminary location prediction, which is then passed to a location prediction filter in the last step, as discussed next.

D.6. Location Prediction Filter

The preliminary location prediction is passed through a lightweight filter designed for resource constrained platforms. It takes into account the previously predicted user location and the maximum distance a user can move within two consecutive predictions. The maximum movable distance (D_{max}) is governed by the following equation:

$$D_{max} = (T_{scan} + T_{predict}) \times S_{gait} \quad (3)$$

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 second for the devices shown in Table I. 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 3 consecutive Wi-Fi scans (number of upper bound scans) was heavily dependent on the smartphone being employed and even varied for each smartphone itself. Therefore, *SHERPA* 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 equation (3). This lightweight filter reduces the occurrence of highly erroneous predictions in corner cases and in doing so improves the quality and fidelity of the indoor localization prediction.

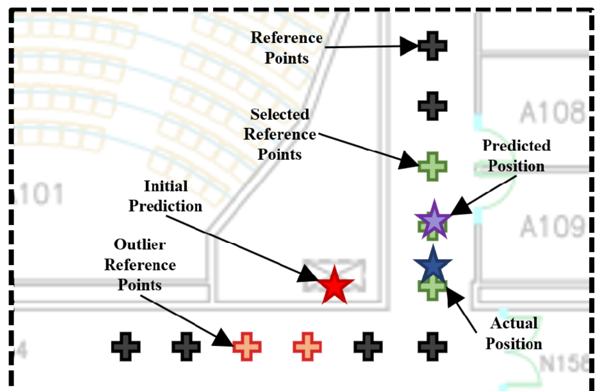


Figure 5: Representation of Z-score based run-time outlier detection on preliminary location prediction in the proposed *SHERPA* framework.

V. EXPERIMENTS

A. Experimental Setup

Heterogeneous Devices and Fingerprinting: To investigate the impact of smartphone heterogeneity, we employed six different smartphones (shown in Table I). 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 IV.A), and then pre-processed (section IV.B).

Indoor Paths for Localization Benchmarking: We compared the accuracy and stability of *SHERPA* and frameworks from prior work on five indoor paths in different buildings at a University campus. These paths are shown in Figure 1; with each fingerprinted location or reference point denoted by a blue dot. The path lengths varied between 60 to 80 meters, 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.

Comparison with Prior Work: We selected three prior works to compare against *SHERPA*. 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.

VI. Experimental Results

A. 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* using a single RSSI vector and the vectors formed by considering the mean of 1 to 5 scanned fingerprints. Figure 6 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 VI.B), 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 IV.D.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 scans per prediction for *SHERPA*'s online phase. If the value is too small, such as the case for the *Lib_Study* path in figure 6, there might not be a significant improvement in localization accuracy. However, if the value is too large, the smartphone may end up consuming a significant amount of energy for an insignificant improvement. From figure 6, 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 be 3 for our framework setup. We opportunistically reduce this to approximately 2 scans per prediction, as discussed in section IV.D.3.

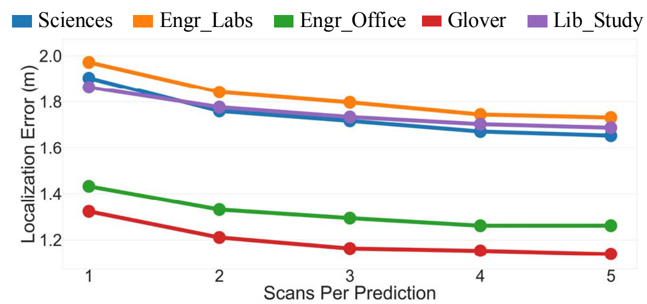


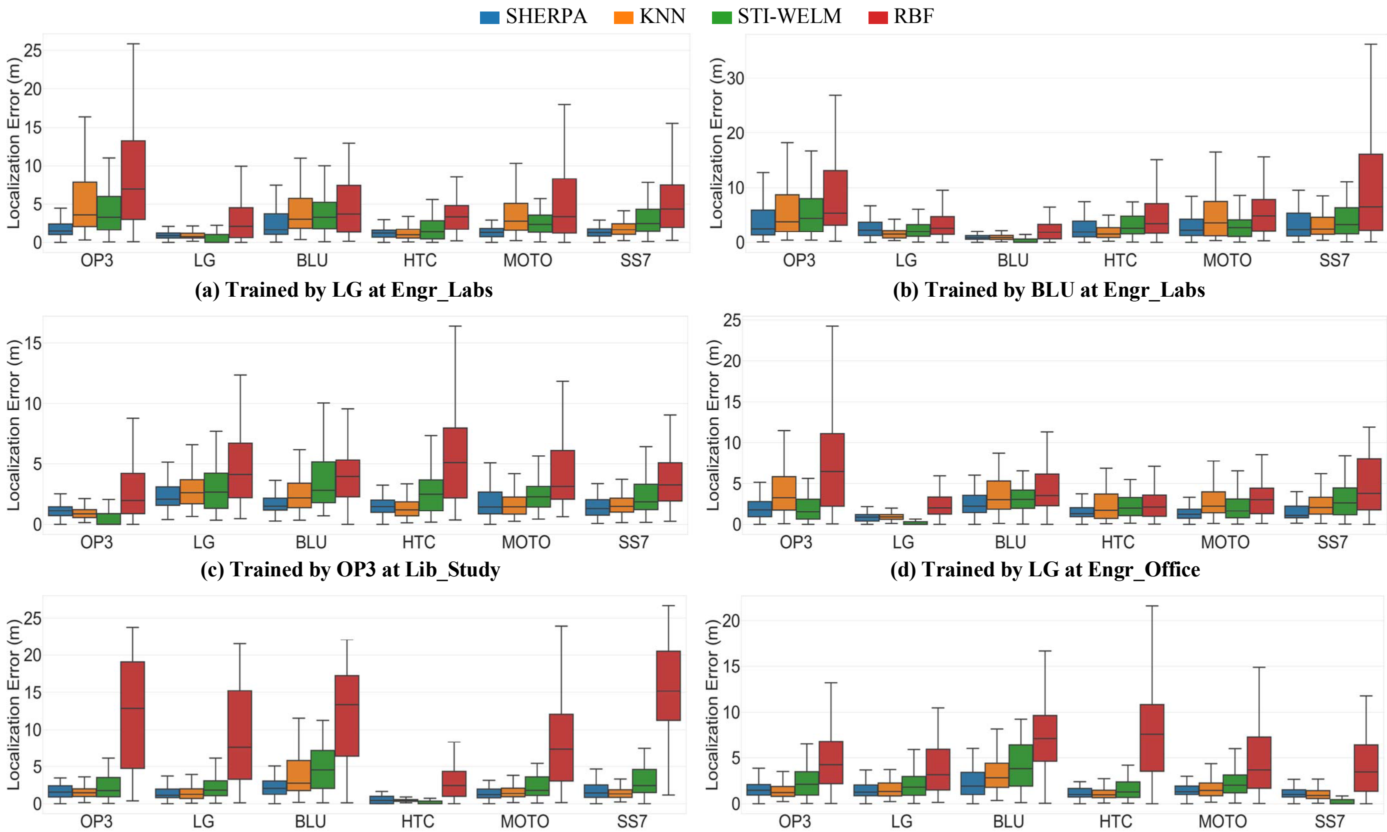
Figure 6: Variation in localization error over scans per prediction for various path benchmarks

B. Performance of Localization Techniques

Figure 7 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 figure 2. We focus on a subset of cases that demonstrate significant deterioration in error (> 2 meters) for the KNN technique.

From figure 7(a), it can be observed that HTC is the most stable device for KNN, i.e., is least affected by heterogeneity. In all other situations, localization error is heavily impacted by heterogeneity. Overall, in figures 7(a) and (b), *SHERPA* can be seen to outperform RBF and STI-WELM whenever the localization error from KNN is > 2 meters. 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 figure 7, we also observe that *SHERPA* outperforms STI-WELM in most training-testing device pairs, other than the non-heterogeneous cases (e.g., LG boxplot in 7(a), BLU boxplot in 7(b), etc.). *SHERPA* is able to deliver better performance in most cases as it is a purely pattern matching approach. 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 online phase. The work in [20] (STI-WELM) assumes a constant gain across heterogeneous devices which is not the case (from figure 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* is outperformed by STI-WELM, *SHERPA* 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 figure 7, *SHERPA* outperforms KNN. In the few cases where it is outperformed by KNN, its accuracy loss is very low.



(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

Figure 7: Localization error for various techniques on benchmark paths across training devices.

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* 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.

C. Comparison of Execution Times

To highlight the lightweight design of our approach, we show the mean execution time of location predictions for *SHERPA* 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 meters × 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 figure 8. The RBF technique is found to take over 2 seconds 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 figure 7) is a major limitation for STI-WELM. After STI-WELM (figure 8), *SHERPA* is the quickest localization framework with an average prediction time of 0.43 seconds that is slightly lower than the lightweight Euclidean-based KNN approach that takes 0.47 seconds for a prediction.

In summary, from the results presented in this section, it is evident that our proposed *SHERPA* 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.

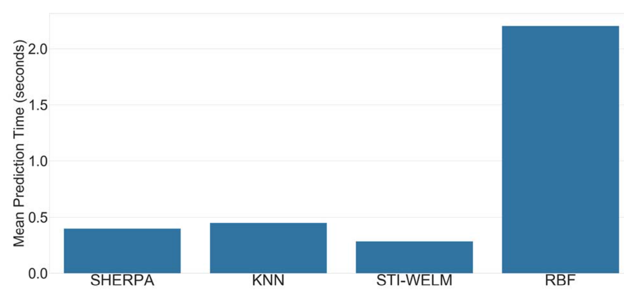


Figure 8: Mean indoor location prediction time for *SHERPA* and frameworks from prior work for the *Lib_Study* path using the OnePlus3 device.

VII. CONCLUSION

In this paper, we proposed the *SHERPA* 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* 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.

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 Consumer Electronics*, vol. 6, no. 4, 2017.

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

[4] A. Mittal, S. Tiku, S. Pasricha, "Adapting Convolutional Neural Networks for Indoor Localization with Smart Mobile Devices," *ACM Great Lakes Symposium on VLSI (GLSVLSI)*, May 2018.

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

[6] X. Wang, X. Wang and S. Mao, "CiFi: Deep convolutional neural networks for indoor localization with 5 GHz Wi-Fi," *ICC*, 2017.

[7] W. Xue, X. Hua, Q. Li, K. Yu and W. Qiu, "Improved Neighboring Reference Points Selection Method for Wi-Fi Based Indoor Localization," *Sensors Letters*, vol. 2, no. 2, pp. 1-4, 2018.

[8] J. Park, D. Curtis, S. Teller and J. Ledlie, "Implications of device diversity for organic localization," *INFOCOM*, 2011.

[9] H. Zou, B. Huang, X. Lu, H. Jiang and L. Xie, "A Robust Indoor Positioning System Based on the Procrustes Analysis and Weighted Extreme Learning Machine," *Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1252-1266, 2016.

[10] K. Chintalapudi, A. Padmanabha Iyer and V. N. Padmanabhan, "Indoor localization without the pain," Mobile Computing Networks, 2010.

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

[12] J. Xiong and K. Jamieson, "Towards fine-grained radio-based indoor location," *Mobile Computing Systems & Applications (HotMobile)*, 2012.

[13] E. Soltanaghaei, A. Kalyanaraman and K. Whitehouse, "Multipath Triangulation: Decimeter-level WiFi Localization and Orientation with a Single Unaided Receiver," *Mobile Systems, Applications, and Services (MobiSys)*, 2018.

[14] M. B. Kjærgaard, "Indoor location fingerprinting with heterogeneous clients," *Pervasive and Mobile Computing*, vol. 7, no. 1, pp. 31-43, 2010.

[15] C. Figuera, J. L. Rojo-Alvarez, I. Mora-Jimenez, A. Guerrero-Curieses, M. Wilby and J. Ramos-Lopez, "Time-Space Sampling and Mobile Device Calibration for Wi-Fi Indoor Location Systems," *Transactions on Mobile Computing*, vol. 10, no. 7, pp. 913-926, 2011.

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

[17] S. Fang, C. Wang, S. Chiou and P. Lin, "Calibration-Free Approaches for Robust Wi-Fi Positioning against Device Diversity: A Performance Comparison," *Vehicular Technology Conference (VTC)*, 2012.

[18] M. B. Kjærgaard and C. V. Munk, "Hyperbolic Location Fingerprinting: A Calibration-Free Solution for Handling Differences in Signal Strength," *Pervasive Computing and Communications (PerCom)*, 2008.

[19] A. K. M. Mahtab Hossain, Y. Jin, W. Soh and H. N. Van, "SSD: A Robust RF Location Fingerprint Addressing Mobile Devices' Heterogeneity," *Transactions on Mobile Computing*, vol. 12, no. 1, pp. 65-77, 2013.

[20] H. Zou, B. Huang, X. Lu, H. Jiang and L. Xie, "A Robust Indoor Positioning System Based on the Procrustes Analysis and Weighted Extreme Learning Machine," *Transactions on Wireless Communications*, vol. 15, no. 2, pp. 1252-1266, 2016.

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

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

[23] R. C. Browning, E. A. Baker, J. A. Herron and R. Kram, "Effects of obesity and sex on the energetic cost and preferred speed of walking," *Journal of Applied Physiology*, vol. 100, no. 2, pp. 390-398, 2006.

[24] J. Machaj, P. Brida and R. Piché, "Rank based fingerprinting algorithm for indoor positioning," *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", Journal of Pervasive and Mobile Computing, vol. 20, pp. 47-63, 2015.

[26] B. Donohoo, C. Ohlsen, S. Pasricha, C. Anderson, Y. Xiang, "Context-Aware Energy Enhancements for Smart Mobile Devices", IEEE Transactions on Mobile Computing (TMC), Vol 13, No. 8, pp. 1720-1732, 2014.

[27] B. Donohoo, C. Ohlsen, S. Pasricha, C. Anderson, "Exploiting Spatiotemporal and Device Contexts for Energy-Efficient Mobile Embedded Systems", IEEE/ACM 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", IEEE International Conference on Computer Design (ICCD), 2011.

[29] S. Tiku, S. Pasricha, "Energy-Efficient and Robust Middleware Prototyping for Smart Mobile Computing," IEEE 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", ACM/IEEE International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2017

[31] F. L. Coolidge, "An Introduction to Correlation and Regression," in Statistics: A Gentle Introduction, 3rd edition, LA, USA, SAGE, 2012, pp. 211-267.