LearnLoc: A Framework for Smart Indoor Localization with Mobile Devices

ABSTRACT
There has been growing interest in location-based services and indoor localization in recent years. While several smartphone-based indoor localization techniques have been proposed, these techniques have many shortcomings related to accuracy and consistency. These prior efforts also ignore energy consumption analysis which is a crucial quality metric in resource-constrained smartphones. In this work, we propose novel techniques based on machine learning algorithms and smart sensor management for real-time indoor localization using smartphones. We implement our proposed techniques as well as state-of-the-art techniques on real smartphones and evaluate their tracking effectiveness and energy overheads across several diverse real-world indoor environments. Our best technique improves upon prior work, achieving indoor localization accuracy between 1-3 meters.

1. INTRODUCTION
The abundance of sensors in contemporary smartphones has significantly improved user context estimation and led to the proliferation of context-aware apps [1]. Location is one of the most important components of a user’s context. Since the inception of smartphones, location-sensing GPS modules have been an integral part of these mobile devices, suggesting the interest of companies and consumers in providing and knowing the location of a user. Governments and corporations have made huge investments in creating infrastructure for map based location services [2].

Recent years have seen the emergence of a-GPS (assisted GPS) that uses the GPS sensor together with supplemental smartphone sensors (e.g., accelerometer, gyroscope, magnetometer) and wireless signals to improve outdoor navigation precision [3]. Researchers have attempted to utilize this approach to create a smartphone-based ‘indoor GPS’ for indoor navigation, but without significant success. This is because in contrast to outdoor environments, GPS signals are very weak indoors and therefore cannot be used for recalibration. Without regular recalibration, estimation errors accumulate over time, causing location accuracy to decay dramatically [4].

Despite many efforts in both academia and industry [5]-[18], highly accurate and practical smartphone-based indoor localization remains an open problem. Some solutions for accurate indoor localization cannot be readily converted to smartphone-based ones due to the limited resources available on smartphone platforms, while others require custom infrastructure (e.g., RF signaling beacons densely integrated into indoor environments) which is expensive and not scalable [5]-[9]. A few approaches (e.g., [12]) exploit existing infrastructure, such as Wi-Fi, for indoor localization. The abundance of Wi-Fi signals and their ability to penetrate through walls makes them an interesting option for trilateration or fingerprinting based localization. But Wi-Fi signals often suffer from signal attenuation and noise in indoor environments. This is due to certain materials in building structures, such as metal beams, which greatly influence electromagnetic waves and magnetic instruments. As a result, there can be distortion in readings provided by the Wi-Fi radio, which can result in erroneous indoor location estimation.

Clearly, new techniques are needed to enable robust and accurate indoor localization in the presence of these challenges. Moreover, the continuous use of Wi-Fi and other sensors requires the processor and memory subsystem to be active, which can quickly drain a smartphone battery. Thus, there is a critical need to analyze energy consumption for any viable indoor location sensing technique, to ensure that smartphones operating on a limited energy (battery) budget do not die before reaching the intended destination.

In this paper, we propose a new framework for indoor localization using smartphones. Our proposed framework (LearnLoc) combines machine-learning techniques with smart Wi-Fi fingerprinting and supplemental sensors to improve the accuracy of indoor localization. Our work makes the following novel contributions:

- We propose the integration of inertial and Wi-Fi fingerprinting with three regression-based machine learning techniques that we have adapted and enhanced for indoor location sensing to create a low-cost, infrastructure-less indoor navigation solution;
- We implement these techniques on an actual smartphone and quantify their performance in real-time for indoor localization;
- We compare the accuracy of our techniques with state-of-the-art techniques from prior work for indoor localization;
- We perform extensive benchmarking and testing of these techniques in different indoor building environments with more diverse structural compositions than any prior work;
- Unlike previous efforts, we also compare and contrast energy overheads of all considered indoor localization techniques.

2. RELATED WORK
There has been a lot of work in the area of indoor localization techniques [5]-[18]. Here we summarize a few key works.

Several techniques employ active techniques, which involve the deployment of custom hardware and software infrastructure for indoor localization. Examples include the deployment of custom signaling beacons based on ultrasonic [6], RFID [7], and wideband acoustic [8], [9] technology. These solutions make use of trilateration or fingerprinting for indoor localization. In trilateration, signal strength data from beacons is converted into a distance with a signal propagation model from the beacon to the mobile device. A circle where the radius equals distance from the measuring point is estimated. When at least 3 measurements are combined, the intersection of the circles gives the position of the mobile device. In fingerprinting, an offline phase creates a database of wireless signal characteristics at various locations, and at real-time, signal characteristics measured by the mobile device are compared to the database, to provide the best estimate of the location of the device. But setting up the custom infrastructure for these approaches is often impractical, expensive, and not easily scalable.

To keep costs manageable, several alternative passive techniques have been proposed that sense pre-existing wireless signals from the environment to enable indoor localization [10]-[12]. For example, PlaceLab [11] uses pervasive radio signals (e.g., GSM) and uses received signal strength (RSS) data to create a signal fingerprint database using crowdsourcing, for indoor environments. Radar [12] uses Wi-Fi signals for location tracking, with empirical methods that utilize Wi-Fi RSS to determine user position. But the accuracy of these techniques is generally not very high.

Several other techniques make use of inertial sensors for indoor localization [13]-[15]. For example, IndoorAtlas [13] proposes creating magnetic fingerprints for indoor maps and using magnetometers for localization. But our analysis shows that such systems do not work well in buildings with metallic structures, which create magnetic interference. Footpath [14] uses an accelerometer and magnetometer on smartphones for indoor localization. Their approach aims to overcome the major drawback of inertial navigation solutions – error accumulation over time – by using a
sequence alignment algorithm that resets these errors by matching detected steps with expected steps extracted from a map.

Several other techniques combine the approaches discussed above to create hybrid techniques [16]-[18]. For example, [16] combines Wi-Fi fingerprinting and inertial sensor based indoor navigation. SurroundSense [18] proposes using ambient fingerprinting with Wi-Fi triangulation, inertial sensors, ambient light, ambient sound, etc. and uses a Support Vector Machine (SVM) based classification-learning approach for indoor localization. However, the authors only present simulation analysis and do not implement their approach on smartphones. On implementing their approach on a smartphone, we observed that the classification method is too time-consuming and thus not a viable solution for real-time localization. Our LearnLoc framework uses low overhead regression-based learning techniques that are more viable for real-time implementation on smartphones.

3. LEARNLOC FRAMEWORK

In this section we first discuss the three fundamental components of LearnLoc needed for indoor localization (Sec. 3.1-3.3). Then we present three regression-based machine learning techniques to improve indoor localization performance with LearnLoc (Sec. 3.4).

3.1 Step Detection

Step detection is a vital component of any indoor localization method. We adapt the technique from [14] for step detection, where only the z-axis accelerometer value is required to detect a step. The accelerometer values display a regular pattern and a step is detected when there is a sharp drop in acceleration due to the jiggling of a phone in the hand or pocket. We apply a low pass filter to improve detection. A step is detected when the difference for consecutive z-axis acceleration values changes by 2 m/s². The difference is checked for a window of 5 consecutive readings. After each detected step a timeout of 333ms is used to avoid false detection.

3.2 Inertial Navigation

Inertial navigation (or dead reckoning) is a popular technique for indoor localization that can be accomplished by combining readings from two or more inertial sensors. Inertial navigation is fundamental to LearnLoc, where we use it to determine the heading angle of a person and step detection. The heading angle is the angle in which the user is facing with respect to the true North. We obtain the angle by combining the accelerometer, gyroscope, and magnetometer readings. The accelerometer provides the gravity vector, the magnetometer works as a compass, and the gyroscope provides angular rotation speed. The angular rotation speed is first integrated over a time interval to determine the orientation of the mobile device. Then the sensor data from all three sensors is combined using Kalman filtering to obtain precise orientation that avoids both gyro drift and noisy orientation. We use this resulting data to determine the change in position from the current position.

![Figure 1: Change in position calculation for inertial navigation](image)

Figure 1 shows how a new position is calculated. The angle ($\theta$) is the heading angle that is multiplied by a factor $d$, which is the step distance of the user (with a default value of 0.6 meters). The step distance is determined for each user by averaging the distance between their two feet for 5 consecutive steps, using inertial sensors. The equations to calculate the new position of $L_{n+1}(x_{n+1},y_{n+1})$ are:

$$x_{n+1} = x_n + d \cdot \cos(\theta)$$

$$y_{n+1} = y_n + d \cdot \sin(\theta)$$

3.3 Wi-Fi Fingerprinting

We also make use of IEEE 802.11 wireless signal strength as an ambient location fingerprint for indoor environments. The Media Access Control (MAC) addresses of visible Wi-Fi access points (AP’s), the Received Signal Strength Indication (RSSI) value, and location coordinates are stored in a tuple. Fingerprints are collected along the indoor path on the mobile device, by the user. This manual step is a prerequisite for all techniques that use Wi-Fi for indoor tracking. The fingerprint details are logged into an SQLite database that is accessed by our machine learning algorithms (Sec. 3.4). We found that a fingerprint gathered after every 3-4 meters on the path works well for the algorithms. Every point on a path typically has a large number of visible MAC addresses, which requires filtering out the addresses that are significant for tracking purposes. We filter out and select only those MAC addresses that are at least present at $j$ unique locations on the fingerprint map, to eliminate noisy signals.

3.4 Enhancements with Machine Learning

To improve indoor location prediction capabilities over prior work, we propose to integrate machine-learning techniques that intelligently make use of step detection, inertial navigation, and Wi-Fi fingerprinting. Machine learning algorithms search for patterns and regularities in any given data and have found wide usage across various application domains. These algorithms are typically implemented in two phases. In the first phase, data is gathered and provided to the algorithm, so it can learn patterns and create a model to classify data or predict data properties. This is called the training phase. In the second phase, new data is tested against the model that was built during the learning phase, and the effectiveness of the model is revealed. This is called the testing phase. Such two-phase learning algorithms are called supervised learning algorithms [19]. There are also algorithms in which the training phase is not used, and such algorithms are called unsupervised learning algorithms. These algorithms use unlabeled data to cluster the data in different classes.

In our LearnLoc framework, we adapt three supervised learning algorithms to assist with indoor localization: K-nearest neighbor (KNN), linear regression (LR), and non-linear regression with neural networks (NL-NN). We utilize regression-based variants of these algorithms instead of a more traditional classification-based approach. This is because a classification technique requires dividing the entire indoor map area into a fine-grained grid for classification towards accurate localization, which creates a prohibitively large input space that is impractical for resource-constrained smartphones. Indeed our implementation of [18] that proposes an SVM-based classification technique found that it was not possible to provide real-time localization on an actual smartphone because of the large memory footprint and slow performance (taking close to a minute for a prediction) with the approach. In contrast, regression can allow us to quickly predict a continuous dependent value with much lower resource demands, which is what is needed for real-time indoor localization with mobile devices. This is also why we do not use unsupervised learning algorithms, as there is no straightforward unsupervised learning algorithm that can be used for regression.

All three of our algorithms allow us to predict the location of a user over a majority of the map area with the few fingerprints that are collected in the training phase. We implement Linear Regression training on a server and then offload the predictions (testing phase) on the mobile device, while both the training and testing phases for the KNN and NL-NN algorithms are implemented on the mobile device. These algorithms are discussed below in more detail.

3.4.1 K-Nearest Neighbor

The K-Nearest Neighbor (KNN) algorithm is a non-parametric algorithm, and can be used for both regression and classification. Non-parametric methods assume that similar inputs have similar outputs. Most of the effort in this algorithm is expended in the testing phase as it is semi-supervised and hence has a minimal training
phase. As all of the work for obtaining the best prediction for the location of a user is done at every instance of the testing phase, KNN can be considered an instance based learning algorithm. Our KNN regression approach works as an extension of the KNN classification algorithm. In KNN classification, new samples are classified by assigning the class that is the most common among the k closest sample in the training set. In our KNN regression implementation, the output value is calculated as the average of the value of its k nearest neighbors. To determine the closest sample, some form of a distance function is required. We make use of the Euclidean distance (D) measure between any two points a and b, each containing n attributes. This measure is defined as:

$$D(a, b) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2}$$  \hspace{1cm} (3)

In the training phase of our KNN algorithm, only the training data is collected and rearranged for the testing phase. This greatly reduces the time needed for training, but the time during the testing phase increases as a majority of the mathematical calculations in the algorithm are performed during the testing phase. These calculations typically require high memory and processing overhead. However mobile devices such as smartphones have stringent energy and memory capacity constraints. Thus it was essential for us to optimize the KNN implementation for it to work on smartphones. We implemented approximation techniques that constrain the search space to a small subset of nearest neighbors, when estimating user location. Details of these techniques are omitted due to lack of space.

3.4.2 Linear Regression

Linear regression (LR) models capture relationships between output dependent variables and input variables. These linear models are built during the training phase and used to make predictions during the testing phase. The key assumption of this approach is that the output variable is a linear combination of certain weights and input variables. For non-linear relationships, these models would provide inaccurate predictions. However, efficient non-linear models are also harder to derive. We found that for our purposes of indoor location estimation, linear models provided good accuracy.

The creation of linear regression models requires fitting the input data using one of several functions. We make use of the least squares approach, which involves a mathematical procedure for finding the best fitting curve to a given set of points (training data) by minimizing the sum of the squares of the offsets or the residuals of the points from the curve, as expressed below:

$$y(x; w) = w_0 + \sum_{i=1}^{n} w_i x_i$$  \hspace{1cm} (4)

$$w_{best} = \arg\min_{w} \sum_{i=1}^{n} (t_i - y(x_i; w))^2$$  \hspace{1cm} (5)

Eq. (4) shows the output values (y) as a function of the inputs (x) and the weights (w) in our linear regression model. We determine the weight vectors by minimizing the error between the target values (t) and the output of the function y as shown in Eq. (5). The training phase for linear regression is time consuming and extremely compute intensive. Therefore, in LearnLoc this training phase is performed on a server and not on the smartphone. We however do utilize the smartphone in the testing phase to make predictions.

3.4.3 Non-linear Regression with Neural Networks

Neural networks are non-linear models inspired by the manner in which biological neurons work in our brain. The human brain is a large interconnected network of hundreds of billions of neurons that processes information and is very capable of learning as well as recognizing patterns. In a neural network model, the artificial neurons are referred to as perceptrons, and have many inputs that are all individually weighted. The perceptron weights can either amplify or deamplify the original input signals as shown in Eq. (6):

$$y = \sum_{i=1}^{n} w_i x_i + w_0$$  \hspace{1cm} (6)

The output from a perceptron is a weighted sum of its inputs, with a weighted bias. We used a feedforward backpropagation approach in the training phase to find the weight parameters, $w_i$. These weights minimize the mean squared errors between the neural network outputs, $y$, and the target outputs $t_i$. Once the neural network model is created we can compute the output $y$ given a sample $x_i$. As some of our localization-centric data showed non-linear relationships, we aimed to improve prediction accuracy by creating a non-linear neural network (NL-NN) model. This is done by applying a sigmoidal or hyperbolic tangent to hidden layer perceptrons as shown below:

$$y_i = \text{sigmoid}(y_i) = \frac{1}{1 + \exp(w^T x)}$$  \hspace{1cm} (7)

4. EXPERIMENTS

4.1 Experiment Setup

This section describes how LearnLoc was implemented on a mobile device, the algorithms chosen for comparison from prior work, and the benchmarks and models used for the experiments.

4.1.1 LearnLoc implementation

We designed a LearnLoc mobile app for the Android mobile ecosystem. The app allows fingerprinting paths in a given map as part of a training phase. In this phase, the user initially selects a map, sets the scale for the map, and specifies the starting position on the map. The app then performs Wi-Fi scans at regular intervals while the user moves along the indoor environment. This phase continues till the user explicitly indicates an end to the training phase in the app. The captured data is used to train our machine learning algorithms. The testing phase uses the trained learning algorithms to make predictions. In this phase, the app provides indoor location estimates that are highlighted on the map. Our mobile app implemented all three variants of the LearnLoc framework: using KNN, LR, and NL-NN. The app provides the flexibility to set parameters for the step detection algorithm, such as step distance and window size for step detection. It is also possible to set the Wi-Fi scan frequency and the maximum distance thresholds up to which the predictions from the learning algorithm are valid.

![Figure 2: (a) Wi-Fi MAC IDs detected along a path (blue dots indicate RSSI of the IDs) (b) convergence problem](image)

On testing our LearnLoc app, we encountered two challenges. Firstly, we found that for several indoor environments there are a large number of visible Wi-Fi MAC ID’s (Figure 2(a)). Considering all of these MAC ID’s increased prediction time and also caused overfitting of learning data, reducing prediction accuracy. To keep run-time overheads low and prevent overfitting of learning data, we filtered and considered only the most significant MAC ID’s, defined as those MAC ID’s that are at least present at 12 different points in the training data. For a location where a MAC ID was not present (after the filtering step), we set the signal strength to zero. Secondly, we also encountered a convergence problem when running LearnLoc and this problem is illustrated in Figure 2(b). It was observed that when a user is in a particular region (e.g., the circle in the figure), the learning algorithms predicted the same location repeatedly until the user had moved completely out of the region. To address this issue, we store the previous prediction from the learning algorithm and then
calculate the distance between the new and previous predictions. If we observe that the distance is below a particular threshold, we discard the prediction and use the inertial sensor trace to obtain the location prediction. We set the threshold in this case to be 120 cm.

4.1.2 Comparison with prior work

To compare and contrast the effectiveness of LearnLoc for indoor localization, we compared it against three prior works in the area. The first approach (PlaceLab) uses GSM RSSI for indoor localization [11]. The second approach (Radar) uses Wi-Fi RSSI based location estimation technique that uses empirical models with RSSI information for location estimation [12]. The third approach (Inertial_Nav) is an inertial navigation approach from Bitsch et al. [14] that uses the accelerometer and magnetometer to provide indoor location estimates. The accelerometer determines if a step is detected and the accelerometer and magnetometer data is combined to give the heading angle. The data is combined by calculating the cross product of the accelerometer and magnetometer values to give the angle. All of the compared techniques ([11], [12], [14], LearnLoc) were implemented and tested on an HTC Sensation [20] smartphone running the Android OS version 4.1.

![Figure 3: Paths chosen for indoor localization analysis](image)

4.1.3 Indoor paths for localization benchmarking

To quantify the performance of our LearnLoc framework as well as the prior works that we compare against, we selected four indoor paths in three buildings that are part of our University campus. These paths were used as benchmarks in our accuracy and energy comparisons between the indoor localization techniques. The paths are shown in Figure 3, highlighted in red against the backdrop of the indoor floor plan. The starting and end points are marked as “S” and “E” respectively. The path lengths range from 110 meters to 140 meters. Each building was chosen because of its unique characteristics that differed from other buildings. The Clark building is one of the oldest buildings on campus, and primarily made of wood and concrete. We chose two paths (Clark L2 South and North) in this building. The Library is a relatively new building that has a mix of metal and wooden structures with open spaces. We chose one path (Library L3) in this building. The Engineering building has a significant amount of metal in its structure as well as in the equipment in the labs. The presence of a large quantity of metal creates magnetic disturbances which can complicate indoor localization. We chose one path (Engineering B) in this building.

4.1.4 Location accuracy and energy estimation

We estimate the accuracy of the indoor localization techniques by checking for the deviation of prediction error along the path after every 10 meters. We implemented LearnLoc and techniques from prior work as apps. A widget in the apps calculates the distance between the traced path and the actual path (specified by the user). The widget then uses an appropriate scaling factor for the map to obtain the actual distances and then averages the piecewise errors to give an overall average error for each technique. To estimate energy for the indoor localization techniques, we used a Monsoon Solutions power meter [21] to characterize and model power dissipation of various modules in the smartphone (e.g., CPU, inertial sensors, Wi-Fi interface, 3G/4G radio, screen backlight). The power meter connects to a smartphone and provides a profile of power dissipated over time when the smartphone is in use. We logged statistics for the various smartphones modules (e.g., how frequently they were invoked and their utilization) during indoor location estimation with each of the localization techniques, and then used the power models to determine energy consumption for the indoor localization techniques on the HTC Sensation [20] smartphone. All energy values were averaged over five readings to improve result fidelity.

4.2 Experimental Results

In this section, we first present results that explore the impact of changing the Wi-Fi scan interval within LearnLoc. Subsequently, we present results comparing LearnLoc with prior work in the area.

4.2.1 Wi-Fi scan interval sensitivity analysis

Our energy analysis revealed that each instance of a Wi-Fi scan consumed a notable amount of energy (~2380 mJ). This motivated us to explore the most suitable value of a Wi-Fi scan interval for our LearnLoc framework. We therefore conducted a sensitivity analysis and recorded the indoor location estimation accuracy and energy costs for the KNN, LR, and NL-NF variants of LearnLoc. Figure 4(a) shows the average location estimation error and Figure 4(b) shows the energy consumed for indoor localization on the HTC Sensation smartphone with Wi-Fi scan intervals varying from 1-16 seconds, for the KNN, LR, and NL-NF variants. The results are shown averaged across all four paths. Figure 5 shows a detailed look at the predicted paths for different Wi-Fi scan intervals when using the KNN variant of LearnLoc on the Clark L2 South path.

![Figure 4: Impact of Wi-Fi scan interval on (a) indoor localization error, (b) location-tracking energy consumption](image)
behind this is the unavailability of GSM signals in many indoor locations. In Radar, the use of Wi-Fi signals only along with empirical methods leads to large errors, due to the inconsistency of RSSI for Wi-Fi signals in many indoor locations. The Inertial_Nav technique performs better than PlaceLab and Radar, but shows higher error than LearnLoc techniques due to error accumulation in the inertial sensors over time. Supplementing Wi-Fi signals with inertial sensors together with machine learning techniques, as done in LearnLoc, proves to be an overall better approach.

It is interesting to observe that the Inertial_Nav technique has one of its worst performances in the Engineering and Library buildings. This can be attributed to the high amount of magnetic interference due to the abundance of lab equipment in the Engineering building and metal walls and surfaces in both the Engineering and Library buildings. All the techniques work well in the Clark building. This is because Clark is a relatively old building with wooden and brick walls. There are very few metallic structures present in the building and hence there is a very little magnetic interference that can impact magnetometer or Wi-Fi readings. The lackluster performance of the LR variant of LearnLoc was found to be the result of non-linearities in the relationship between the Wi-Fi fingerprints and location data that a linear model is unable to capture accurately.

Figure 7 summarizes the paths traced by the four best performing indoor localization techniques for the Clark L2 South path. It can be observed that the path traced by the Inertial_Nav technique greatly deviates from the actual path for Clark L2 South, shown in Figure 3. This is due to error accumulation over time. The sequence alignment algorithm in the Inertial_Nav technique aims to overcome this error with periodic recalibration, but is not always successful in doing so. For our LearnLoc variants, the green points in the figure indicate instances where a Wi-Fi scan was performed. For the LR variant, the generated linear model is not very tolerant to noisy readings and thus its predictions are not consistently accurate along the path. The KNN variant performs the best, with an average error of 2.23 meters, while the NN variant has an average error of 4.38 meters.

The energy consumed by all indoor localization techniques is shown in Figure 6(b). The energy consumed on the Engineering B path was the lowest out of all other paths as it was the shortest path in the study. The energy values shown for the LearnLoc variants are for the testing phase of these variants. In the training phase for each of the four paths, KNN consumed approx. 0.95KJ while NL-NN

4.2.2 Localization Algorithm Comparison

The localization accuracy results for the three variants of LearnLoc and the approaches from [11], [12], and [14] are shown in Figure 6(a). Results are shown for the four indoor path benchmarks shown earlier. It can be observed that the KNN variant of our proposed LearnLoc framework achieves the best accuracy across all paths, out of all the techniques considered. The accuracy of the NL-NN variant is lower than for the KNN variant but higher than the LR variant. Our comparison with PlaceLab indicates that the use of GSM signals alone, as done in PlaceLab, gives poor results. The reason behind this is the unavailability of GSM signals in many indoor

2 sec 4 sec

Figure 5: Paths traced for various Wi-Fi scan intervals for the LearnLoc framework using KNN along the Clark L2 South path; green dots represent an instance of a Wi-Fi scan along the path.
consumed approx. 31.5KJ on the smartphone. For LR, training was not possible on the smartphone due to higher resource requirements; therefore we performed the training phase (regression model design) on a server and then ported the models to the smartphone.

From Figure 6(b) it can be observed that on average the LearnLoc techniques have a slightly higher energy overhead compared to the other techniques. However, this overhead can be easily justified in light of the significant benefits in indoor localization accuracy, associated with LearnLoc based techniques (Figure 6(a)). The most accurate technique (LearnLoc-KNN) also consumes more energy than other techniques. This is attributed to the high computation overhead to generate predictions during the testing phase of the KNN algorithm. However, the energy value is still low enough to enable viable implementation on a smartphone. The NL-NN technique provides a middle ground between the KNN and LR variants, for both energy consumption and localization accuracy. However, the high training overhead for NL-NN, as discussed earlier, is a limitation. The LR variant has slightly lower energy consumption on average compared to KNN and NL-NN, but its notably lower localization accuracy compared to the other two variants is a drawback. PlaceLab has higher energy consumption due to the high energy associated with GSM RSSI detection and processing. Radar has slightly lower energy consumption than PlaceLab, as it utilizes the less energy-hungry Wi-Fi interface. Inertial Nav has the lowest energy consumption as it does not utilize Wi-Fi or GSM radios.

**Figure 7: Paths traced by indoor localization techniques along the Clark L2 North benchmark path**

**Figure 8: (energy x accuracy) comparison, with both energy and accuracy normalized w.r.t RADAR (for Engineering B)**

We summarize the overall performance of all localization techniques with a single metric that captures both localization accuracy and energy consumption in Figure 8. The metric considers the product of the normalized energy and normalized accuracy for all techniques. All results are normalized to the energy and localization accuracy for the Radar technique in the Engineering building. It can be seen that the KNN variant of our LearnLoc framework provides a superior output across several diverse indoor environments, with the best localization accuracy and a competitive energy consumption overhead when compared to the other alternatives. Our work thus strongly motivates the integration of smart machine learning methods to enhance the effectiveness of traditional indoor navigation mechanisms such as dead reckoning and Wi-Fi fingerprinting. The ability to trade-off localization accuracy and energy in our approach also provides a unique benefit that can allow our techniques to be implemented on various mobile devices with different capabilities.

**5. CONCLUSION**

In this paper we presented the LearnLoc indoor localization framework that intelligently utilizes Wi-Fi fingerprinting and inertial sensors for accurate and energy-efficient localization. We presented three variants of our framework that use: K Nearest Neighbor (KNN) regression based learning, Non-Linear regression based Neural Networks (NL-NN), and Linear Regression (LR). We implemented our framework and several techniques from prior work on an actual smartphone. Our experimental studies show that the KNN machine learning based LearnLoc approach is robust to noise and magnetic interference and significantly outperforms other approaches from prior work. The KNN-based approach provides very accurate indoor localization with approximately 1 to 3 meters accuracy. Our ongoing work is attempting to explore other learning algorithms and perform more aggressive trade-offs between accuracy and energy.

**REFERENCES**


