# Wifi Indoor Positioning with Genetic and Machine Learning Autonomous War-Driving Scheme

Pham Doan Tinh
School of Electrical and Electronics Engineering
Hanoi University of Science and Technology
No.1 Dai Co Viet street, Hanoi, Vietnam

Bui Huy Hoang
School of Electrical and Electronics Engineering
Hanoi University of Science and Technology
No.1 Dai Co Viet street, Hanoi, Vietnam

***Abstract*—Wifi Fingerprinting is a widely used method for indoor positioning due to its proven accuracy. However, the offline phase of the method requires collecting a large quantity of data which costs a lot of time and effort. Furthermore, interior changes in the environment can have impact on system accuracy. This paper addresses the issue by proposing a new data collecting procedure in the offline phase that only needs to collect some data points (Wi-fi reference point). To have a sufficient amount of data for the offline phase, we proposed a genetic algorithm and machine learning model to generate labeled data from unlabeled user data. The experiment was carried out using real Wi-fi data collected from our testing site and the simulated motion data. Results have shown that using the proposed method and only 8 Wi-fi reference points, labeled data can be generated from user's live data with a positioning error of 1.23 meters in the worst case when motion error is 30%. In the online phase, we achieved a positioning error of 1.89 meters when using the Support Vector Machine model at 30% motion error.**

***Keywords*—Wifi fingerprinting; indoor positioning; machine learning; genetic algorithm**

## I. Introduction

As people spend more time indoors, many location-based applications and services require to known user indoor location. While the global positioning system (GPS) is a popular positioning method, it can hardly be applied to indoor environments because lacking in line of sight (LOS). Therefore, many indoor positioning techniques have emerged and been proposed in recent years.

An approach that many researchers have taken is to use the network infrastructure like Wi-fi [1], [2], Bluetooth [3], [4], Zigbee [5], Ultra-Wideband [6] to perform indoor localization. Wi-fi is the most common because it is likely to be installed in most public indoor places like malls, stations, airports and nearly every smartphone is equipped with a Wi-fi transceiver module. A popular method that utilizes Wi-fi signal is Wifi Receive Signal Strength Indicator (RSSI) Fingerprinting [7], [8], [9], [10]. The approach assumed that the RSSI measurement from access points (APs) for every location is unique. For that reason, RSSI measurements are recorded and stored in a database called the radio map. Whenever a new RSSI measurement is generated by the user, it will be matched to the similar one in the database. One of the popular matching algorithms is The Nearest Neighbors in Signal Space (NNSS) [11], which calculates the distance of signal space between the observed data and the recorded data. The step of constructing the database is also known as the offline phase and the matching step is known as the online phase. In the matching step, machine learning can be also be used to take advantage of the powerful pattern recognition ability to produce more accurate results [10].

Another popular approach focus on exploiting the inertial measuring units (IMU) because of their availability in mobile devices and fast measurement update. A well-known method that takes advantage of the IMU measurements is Pedestrian Dead Reckoning (PDR) [12], [13], [14]. PDR extract step event [15], step length [16] and heading angle [17] from IMU raw measurements and output the location of the user. The major benefit of this approach is that no infrastructure needs to be installed. In addition, measurements are regularly updated which enables real-time localization. However, the initial position of the target is required to be predetermined in PDR because PDR can not locate the current position of the target without the knowledge of the target's previous location in the environment. Furthermore, IMU sensors are subjected to noise, interference and disturbance thus produces accumulated errors over time. For that reason, sensor fusion methods like Kalman Filter [18] or Madgwick Filter [19] were employed to counter the error but due to the complexity of those filters, they raise the amount of computation to solve the indoor localization problem.

Wi-fi Fingerprinting has advantages when it comes to precision and deployment cost compared with other indoor positioning schemes. However, it is worth mentioning the drawbacks of the method. First, Wi-fi Fingerprinting method performance is suffered from multipath, shadowing, and interference in the environment [20]. Secondly, the number of data points in the radio map can affect the positioning accuracy. Therefore, in the offline phase of conventional Wi-fi Fingerprinting, the coordinate system of the whole area needs to be built, then numerous data points are marked to have their RSSI measurements collected. This procedure consumes a lot of time and effort. Additionally, changes like adding or moving objects in the environment could make the radio map no longer reliable and have to be reconstructed to maintain accuracy. While the offline phase of the Fingerprinting method poses difficulties in collecting data, the PDR method creates a lot of data but they are unlabeled. Aiming to reduce the amount of data needed to be collected in the offline phase of the Wi-fi Fingerprinting method and taking advantage of the results from the PDR method, this paper proposed a new architecture for the offline phase of the Wi-fi Fingerprinting method. From the results of PDR, a genetic algorithm is

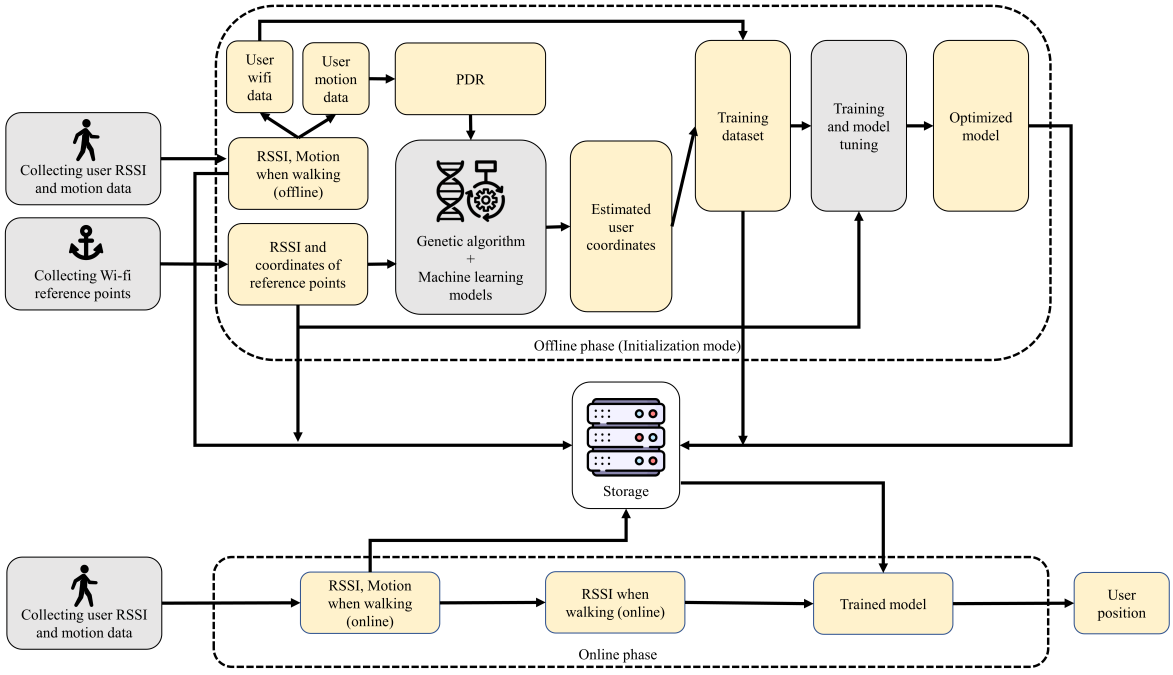



Fig. 1. Offline Phase Initialization Mode and Online Phase of the Proposed Method

implemented to combined with machine learning algorithms to find labels (locations) for the user unlabeled Wifi RSSI data. By creating more labeled data from user data, the proposed method reduces the spent time and effort to collect labeled data while still achieve good results.

Several approaches that used motion sensors and Wi-fi signals for indoor positioning have been studied in the past. Wi-fi SLAM proposed by Brian Ferris et al. in [21] take advantage of a technique called Simultaneous Localization and Mapping (SLAM), which is a popular navigation method in an unknown environment. The authors used Gaussian Process Latent Variable Model to find the location of unlabeled signal strength data in the latent space and combine it with motion data to rebuild the topological connectivity graph to perform indoor localization. Naguib in [22] proposed an indoor positioning scheme that combines multi sources of information, which are motion sensors and Wi-fi using a low-complexity version of particle filter to increase accuracy. Constandache in [23] utilized electronic compass and accelerometers in phones to measure user speed and orientation. Then the recorded data is matched against possible path signatures in the local electronic map. The proposed method is different from others because the user motion data are not used together with Wi-fi to directly predict the user location but they are only utilized in the offline phase to generate more labeled data for the machine learning model. As the motion data is only utilized in the offline phase, the proposed method does not contain any heavy computation in the online phase, which enables real-time localization.

The rest of the paper is arranged as follows. Section 2 presents the proposed system architecture and detail implementation of the genetic algorithm and machine learning models. Section 3 presents the experimental design. Section 4 presents the results and the discussion. The last section concludes the paper with future direction.

## II. The Proposed Scheme

Similar to the architecture of an indoor Wi-fi Fingerprinting method, the system is divided into two-phase: the offline phase and the online phase. In addition, there is central storage which is responsible for saving collected data and trained machine learning models.

In the offline phase, it is divided into two modes: the initialization mode and the update mode. The initialization mode is used on the first run of the system when the storage is empty and it is illustrated in Fig. 1. First, a set of positions is designated for Wi-fi RSSI measurements, this set is denoted as Wi-fi Reference Points (WRP)s. Because the number of collected WRPs is small, their locations should cover the whole area. Then, for each WRP in the environment, the Wi-fi RSSI signals are carefully measured from all the access points and saved them to the central storage. The next step is letting users with a device equipped with Wi-fi transceiver modules and IMU sensors move around in the area. The raw motion data and Wi-fi RSSI are recorded and saved to the storage. When the system decided that it has collected enough user data, it would start analyzing and creating machine learning models. User raw motion data is separated from the user Wi-fi RSSI and they are sent to the PDR block to output processed motion data (step event, step length, and heading angle). Next, the WRP data and user processed motion data are forwarded into the block where a genetic algorithm and machine learning model is implemented to estimate user positions. Details of the implementation are presented in the later sections. Then, the estimated positions are used as the labels for the user Wi-fi RSSI data to form a new dataset. The new dataset is

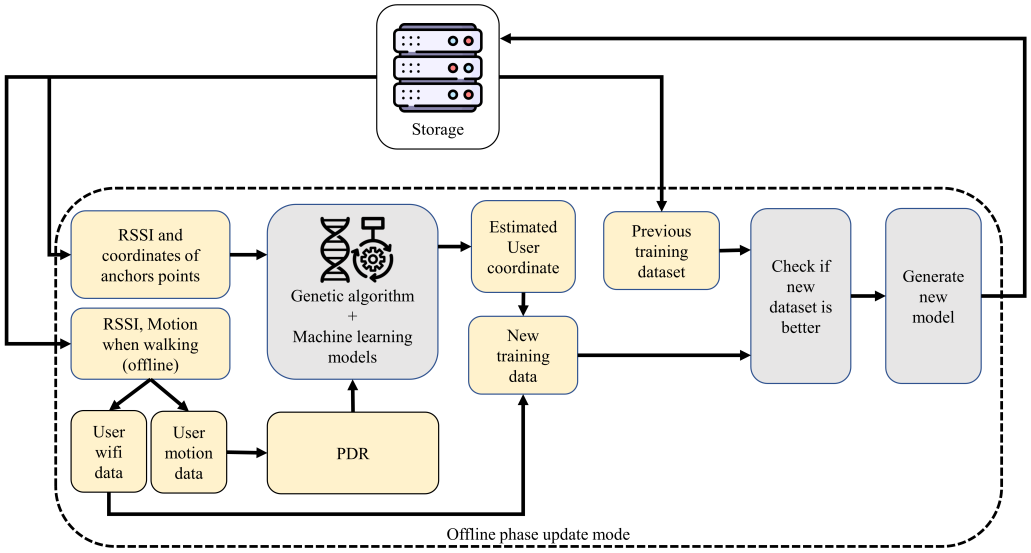



Fig. 2. Offline Phase Update Mode

combined with the WRP data to become the training dataset. Next, machine learning models were used to fit the training data then perform model tuning for better performance. The optimized model is later saved to the central storage.

The online phase is mentioned first for better chronological order. The online of the proposed system is used to estimate the position of the user using the trained model from the offline phase and it is illustrated in Fig. 1. First, user Wi-fi RSSI and motion data are recorded while the user moves. Then, the Wi-fi RSSI data are extracted from the user data and forwarded into the trained machine learning model to output user location. User data are saved to the central storage for further analysis in the offline phase.

The offline phase update mode is presented in Fig. 2. It is utilized when there are new user data collected during the online phase and system operator want to analyze the potential of this data to be used with the previous training dataset. First, the WRP data and user data are loaded from the storage. Then the user data is split into Wi-fi RSSI and raw motion data. The raw motion data is forwarded to PDR to extract processed motion data (step event, step length, and heading angle). Then, the WRP and processed motion data are sent to the genetic algorithm and machine learning block to create user positions similar to the initialization mode. Next, the previous training dataset is loaded from the central storage and compared with the new one. If the new dataset satisfies the metric, then it will be used as training data to generate a new model. Details of the metric are described in the next section.

### *A. Implementation of Genetic Algorithm and Machine Learning Block*

The input of this block is WRP data and user motion data, the output is the estimated user position. WRP is defined as (1):

$$WRP_i = \{(RSSI_i^1, ..., RSSI_i^j, ..., RSSI_i^M), (x_i, y_i)\} \quad (1)$$

where $i = 0$ to $N_{WRP}$ and $N_{WRP}$ is the number of WRPs, $M$ is the number of Wi-fi AP, $RSSI_i^j$ is the RSSI measurement from the $j^{th}$ AP of $i^{th}$ WRP, $(x_i, y_i)$ is the coordinates of $i^{th}$ WRP.

An assumption was made that a PDR algorithm was implemented and it processed the raw motion data from IMU sensors and output the step event, step length, and heading angle. Using the output, the distance and the angle are calculated between two consecutive Wi-fi RSSI measurement points. Then the user Wi-fi RSSI data and the processed motion data is in the form (2):

$$U_i = \{(RSSI_i^1, ..., RSSI_i^j, ..., RSSI_i^M), (\alpha_i, d_i)\} \quad (2)$$

Where $i = 0$ to $N_U$, $N_U$ is the number of user data point, $\alpha_i$ is the angle formed by two vector $\overrightarrow{U_iU_{i+1}}$ and $\overrightarrow{U_{i+1}U_{i+2}}$ and $d_i$ is euclidean distance between $U_i$ and $U_{i+1}$

From the processed motion data $\{(\alpha_1, d_1), (\alpha_2, d_2), ...(\alpha_{N_U}, d_{N_U})\}$, the user route shape can easily be obtained. However, because of not knowing the user's starting point, it is not possible map the route to the environment. As the starting point can be any position in the area, if a searching algorithm is implemented to search for every possible location, it would cost a lot of time and computation resources. To tackle this, a genetic algorithm was proposed.

Genetic algorithm (GA) was first introduced by John Holland in 1960 in [24] and it has been applied as a method to solve optimization and search problems. GA can be divided into three main steps which are population initialization, fitness evaluation [25] and applying genetic operations. The algorithm repeats steps 2 and 3. For each repetition, a new generation is created. The solution is obtained and the algorithm stops when the fitness value has converged or GA has reached the maximum number of generations. Based on the structure of a standard GA, a version of GA was implemented to find the user position in the 2-dimensional coordinate system from

processed motion data. The following are the details of the implementation.

*1) Population Initialization:* First, the representation of an individual (chromosome) in the population is considered. Although searching for the user's starting point is the goal, the whole route also has to be considered because when calculating the error, it is important to calculate for the whole track. An array with each element consisting of two floating-point numbers representing the coordinates of the user in the environment is chosen and shown in (3). No encoding method method like Binary Encoding was used because it requires further computation for binary conversion and loss of precision.

$$p_i = \{(x_1, y_1); (x_2, y_2); ...(x_{N_U}, y_{N_U})\} \tag{3}$$

While randomly generating the initial population (first generation), it is important to determine the population size. The number varies in real cases because of factors like search space, processing capability, and environmental constraints. After trials, the population size is selected to be 100. Constraints were also put on the whole route that every point must be inside the range $[x_{min}, x_{max}], [y_{min}, y_{max}]$. This range is to prevent the generated track from not being too far off the indoor area. To get a random track, the starting point is randomly generated, then the rest of the track positions are calculated using processed motion data mentioned earlier.

**Algorithm 1** Proposed Genetic Algorithm

**INPUT:** WRP data, user motion data
**OUTPUT:** Estimated user positions

**begin**
Set $N_P$, mutation rate $\epsilon_M$, convergence condition $E$
$i = 0$
Initialize first population $P(i)$
Calculate fitness of $P(i)$
$p$ = individual with highest fitness of $P(i)$
**while** $E$ is not satisfied **do**
    Create empty $P(i+1)$
    Populate $P(i+1)$ using selection operator on $P(i)$
    Apply crossover operator on $P(i+1)$
    Apply mutation operator on $P(i+1)$
    Calculate fitness of $P(i+1)$ using ML and WRP
    $\widetilde{p}$ = individual with highest fitness of $P(i+1)$
    **if** (Fitness($\widetilde{p}$) > Fitness($p$)) **then**
        $p = \widetilde{p}$
    **end if**
    Replace $P(i)$ by $P(i+1)$
    $i = i + 1$
**end while**
Output best $p$
**end**

*2) Fitness Evaluation:* The fitness value of an individual needs to show how well that individual perform compared to others. After the initial population is generated, a method to calculate the fitness of the user track by taking advantage of machine learning models was proposed. It is known that if a machine learning model was trained using accurate training data, then the error on the testing data would be small. Using this idea, the random track position is used as the label for the user Wi-fi RSSI data as the training set and the testing set is the WRP dataset. Let's say that the model prediction is (4).

$$\overline{WRP} = \{(\overline{x_1}, \overline{y_1}), (\overline{x_2}, \overline{y_2}), ..., (\overline{x_{N_U}}, \overline{y_{N_U}})\} \tag{4}$$

Then the positioning error between $WRP$ and $\overline{WRP}$ is calculated using (5):

$$E(\overline{WRP}) = \frac{\sum_{i=1}^{N_{(WRP)}} \sqrt{(x_i - \overline{x_i})^2 + (y_i - \overline{y_i})^2}}{N_{(WRP)}} \tag{5}$$

The intuition that an individual which performs better would have a higher fitness value is better for comprehension. Therefore, the fitness value is computed using (6).

$$F(p_i) = \frac{1}{E(\overline{WRP_i})} \tag{6}$$

where $i = 1$ to $N_p$ and $N_p$ is the population size, $\overline{WRP_i}$ is the predicted WRP using the $i^{th}$ individual in the population.

A simulation is carried out to prove that the fitness function in (6) can show how close and accurate the randomly generated track is compared to the real user track. Details are described in the experiment section.

*3) Genetic Operations:* Three operators that need to implement are selection, crossover, and mutation. For the selection operator, Roulette Selection was used. The probability of an individual being selected for the next generation is calculated using (7).

$$\mathbb{P}(p_i) = \frac{F(p_i)}{\sum_{j=1}^{N_p} F(p_j)} \tag{7}$$

The crossover operator was designed so that between two individuals, the one with higher fitness would have more contribution to the offspring. The amount of contribution is evaluated using the metric called fitness weight. If individual A $p_A = \{x_i^A, y_i^A, i \in [1, N_U]\}$ and individual B $p_B = \{x_i^B, y_i^B, i \in [1, N_U]\}$ are selected as parents, then the fitness weight of A and B is computed as in (8):

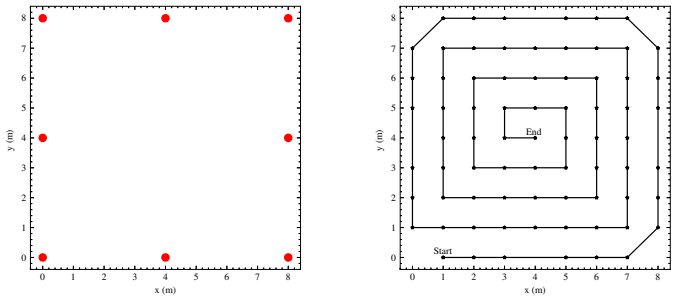
$$w_A = \frac{F(p_A)}{F(p_A) + F(p_B)}; w_B = \frac{F(p_B)}{F(p_A) + F(p_B)} \tag{8}$$

If the offspring of A and B is denoted as C $p_C = \{x_i^C, y_i^C, i \in [1, N_U]\}$ then the coordinates in C are calculated using (9), (10):

$$x_i^C = w_A * x_i^A + w_B * x_i^B \tag{9}$$

$$y_i^C = w_A * y_i^A + w_B * y_i^B \tag{10}$$

For mutation, mutations in the population are created by randomly created 2 values $(dx, dy)$ which represent how far the whole track will be shifted in a 2-dimensional area. The mutation rate is chosen to be a small constant number because it helps the algorithm to converge faster. Details of the configuration are described in the experiment section. The steps of the genetic algorithm is shown in Algorithm 1.




(a) (b)

Fig. 3. (a)Wi-Fi Reference Points marked as Red (b) User Route where Black Marks Indicate Locations of RSSI Measurements

TABLE I. GENETIC ALGORITHM PARAMETERS

| Parameter | Value |
|---|---|
| Population size | 100 |
| Maximum number of generation $N_G$ | 50 |
| Coordinate boundary | $[-5, 15]$ meters |
| Mutation $[dx, dy]$ boundary | $[-3, 3]$ meters |
| Mutation rate | 0.001 |
| Convergence condition | Reaching $N_G$ |

## III. EXPERIMENTAL DESIGN

The experiment was carried out in the laboratory which is a square area of 10 by 10 meters. The room has tables, chairs, computers, and other networking devices. The Wi-fi network is set up and 8 APs are placed around the room. First, The offline phase was performed by designating 8 points in the testing site as the WRP, their locations are illustrated in Fig. 3(a) Then, at each WRP, the RSSI measurements from 8 APs are collected and saved to the central storage. Because the processed motion data (heading angle and distance) were applied from other research and to ease the implementation of the experiment, the user motion data are simulated with random Gaussian noise (noise ranges from 5% to 30%). Then one person would follow the path of the simulated route. While going, the RSSI measurements are collected at marked points in Fig. 3b and saved the data to the central storage.

After acquiring all the necessary data, the next step is to run the genetic algorithm. In the population initialization step, the boundary constraints on generated coordinates are set to be in the range of $[-5, 15]$ (meter) in both axes. The constraints prevent the generated coordinates from being too far off and the negative range while creating a diverse population. Table I shows details of genetic algorithm parameters.

The fitness function needs predictions from the machine learning model to calculate fitness value. However, machine learning models usually take time to train and predict. As the result, it is not recommended to use too many models and perform hyperparameter tuning in this situation. Two machine learning models that have been widely used in indoor positioning research were selected which are Support Vector Machine Regression (SVR) and K-Nearest Neighbors Regressor (KNN). Fixed hyper-parameters were selected for both model in Table II and Table III with no tuning while calculating the fitness.

TABLE II. SVR MODEL PARAMETERS

| Parameter | Value |
|---|---|
| Kernel | Radial basis function (RBF) kernel |
| Regularization parameter | 1.0 |
| Epsilon-SVR | 0.1 |

TABLE III. KNN MODEL PARAMETERS

| Parameter | Value |
|---|---|
| Number of neighbor | 5 |
| Weight function | Uniform |
| Algorithm | Select from Ball-Tree, KD-Tree and Brute force |
| Leaf size | 30 |
| Metric | Minkowski |

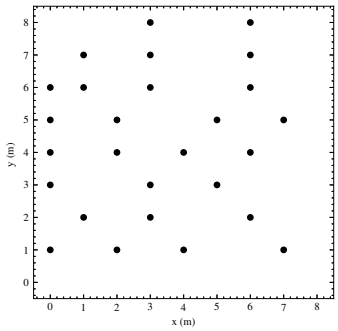



Fig. 4. Testing Point in the Experiment Area

When GA outputs the estimated user position, those positions are mapped to the user Wi-fi RSSI data and combined with the WRP to become training data. At this step, four machine learning models were selected: SVR, KNN, Multi-Layer Perceptron (MLP), and Random Forest (RF) to train and perform optimization on hyper-parameters to achieve the best possible results. The optimization method is Grid Search which was implemented in the sci-kit learn library.

Finally, the performance of the trained models is tested by collecting 27 random points with Wi-fi RSSI measurements as

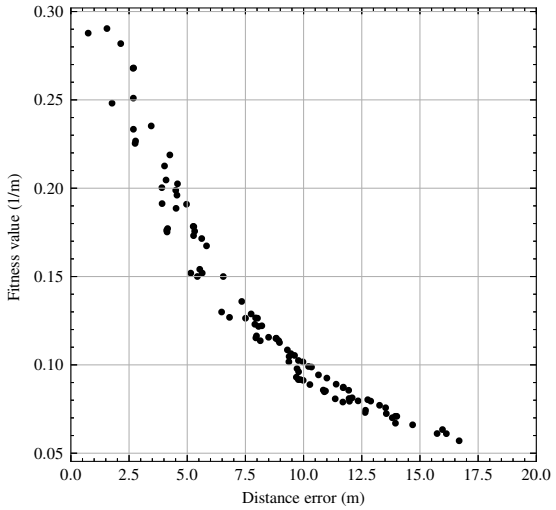



Fig. 5. Relationship between the Fitness Value and the Distance Error of 100 Random Tracks

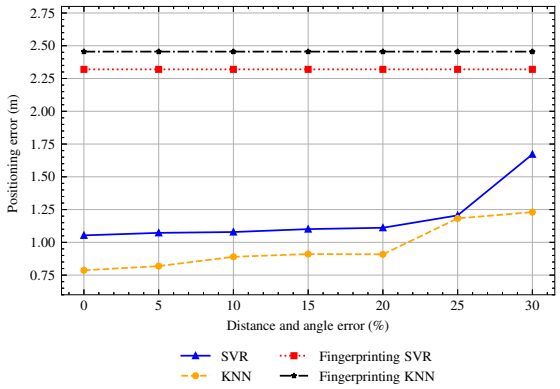



Fig. 6. Positioning Error of the Proposed Method and Conventional Fingeprinting using 8 WRPs with respect to Different Amount of Motion Error

illustrated in Fig. 4. Then the prediction error of each model is calculated and compared.

## IV. Results and Discussion

### A. Fitness Function Evaluation

In this part, the effectiveness of the fitness function on the collected data is illustrated. Using the processed motion data, the obtained route shape is similar to the one in Fig. 3(b). Then, 100 initial points in the environment were randomly generated. From those points, 100 user tracks were acquired. Equation (6) was used to calculate the fitness value of each track and (5) was used to measure the error of the randomly generated track to the real one. The relationship between the two values is shown in Fig. 5.

From Fig. 5 it is clear that when the fitness value is high, the distance error of the randomly generated track and the real track is low. This shows that the fitness function can create value that reflects how close a random track is to the real track. Looking closely at the top right of Fig. 5, it is noticeable that the track with the highest fitness value is not the one with the lowest positioning error. This shows that the fitness function can not find the absolute best because when the machine learning models were used to make a prediction, it is important to also account for the error in the WRP testing set. As it is impossible create an error-free dataset, the error is unavoidable and the only way to counter it is to carefully measure each point in the WRP dataset. Although the track with the lowest positioning error may not be found, the track with the highest fitness is guaranteed to be its neighbors, which gives a reasonable estimation.

### B. Offline Phase Results

The positioning error of the estimated user track coordinates is computed using (5). The same machine learning models with the same configurations as in Table II and Table III were used in the conventional Fingerprinting method for comparison. In conventional Fingerprinting, ML models were trained on the WRP dataset, and then they were tested with the user Wi-fi RSSI dataset. Fig. 6 illustrated positioning error in relationship with different amounts of processed motion error (from 0 % to 30 %) of our proposed method and conventional Fingerprinting method.

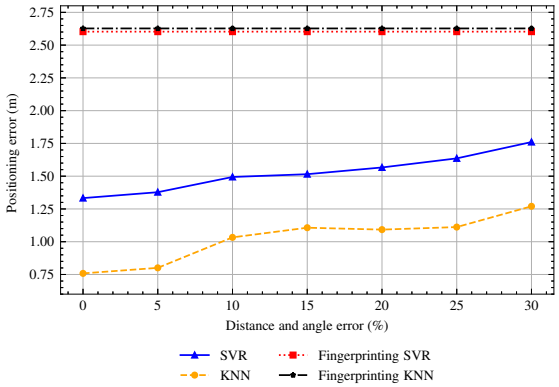



Fig. 7. Positioning Error of the Proposed Method and Conventional Fingeprinting using 6 WRPs with respect to Different Amount of Motion Error

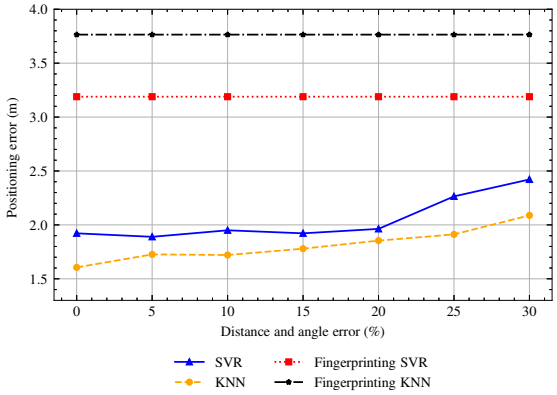



Fig. 8. Positioning Error of the Proposed Method and Conventional Fingeprinting using 4 WRPs with respect to Different Amount of Motion Error

From Fig. 6, the proposed genetic algorithm had better performance in both cases using the SVR and the KNN. With traditional Fingerprinting, it does not use motion data so the results are the same across the different amounts of motion error. For Fingerprinting the positioning error of the SVR is 2.32 meters and the KNN is 2.45 meters, which had been optimized by the Grid Search search algorithm. On the other hand, GA relies on motion data so when processed motion error rises, the positioning error of the proposed method also increases. From 0% to 20% motion error, the positioning error of the proposed method increased slightly from 1.05 meters to 1.11 meters for the SVR model, which is only 5.7% of the increased error. The same trend can be seen with the GA and KNN model where it rises from 0.78 meters to 0.9 meters. From 20% motion error onward, both GA models had a sharp rise especially for SVR at 30% motion error and KNN at 25% motion error. However, even at 30% motion error, the results of GA are 1.67 meters for SVR and 1.23 meters for KNN, which is still much better than the conventional Fingerprinting method. For comparison, our GA with the unoptimized SVR and KNN has positioning error 28% and 46% lower than that

of the Fingerprinting with the optimized models. However, it is worth noticing that although the SVR model is more complex and is supposed to have better performance than the KNN model, GA with KNN has better performance. As mentioned earlier no hyperparameters tuning was performed inside GA so both models may not be optimized and KNN initial parameter may be better than SVR.

To analyze the impact of the number of WRPs on the positioning error, the experiment was carried out with a different number of WRP. Fig. 7 and Fig. 8 show the results when applying the same method and configuration but with 6 and 4 Wi-fi Reference Points. When lowering the number of WRPs, it is expected that the conventional Fingerprinting method would have positioning error increased because of the smaller training dataset. The situation can be observed in both Fig. 7 and Fig. 8. The fingerprinting SVR at 6 WRPs has a positioning error of 2.60 meters, which increases by 12% compared to the similar one at 8 WRPs. In the case of Fingerprinting KNN, the positioning error is 2.62 meters, which is close to the Fingerprinting SVR. The proposed GA with SVR and KNN also depend on the WRP dataset to calculate fitness so the performance is also affected. In Fig. 7, GA with KNN has a lower positioning error than GA with SVR. Compared to the performance of GA with KNN using 8 WRPs, the one using 6 WRP is similar. On the other side, GA with SVR using 6 WRPs has positioning error significantly higher than the one using 8 WRPs. The difference between the two cases ranges from 0.3 to 0.4 meters.

Looking at Fig. 8, the difference in positioning error while using less WRP is even more noticeable. For all cases in proposed GA and Fingerprinting, they all experienced a sharp increase. Although GA with KNN using 4 WRPs still be able to maintain its position to be the best solution, the positioning error has seen rises ranging from 0.7 to 0.9 meters compared to positioning error of the same one using 6 WRPs, which makes the result become 1.6 meters at 0% motion error and reach up to 2.1 meters at 30 % motion error. GA with SVR using 4 WRPs comes behind KNN and the result is not too far off, it has a positioning error of 1.92 meters at 0 % motion error and gets up to 2.42 meters at 30 % motion error. In the case of conventional Fingerprinting, KNN has the worst result with a positioning error of 3.76 meters, which is 35 % higher than the same version that uses 6 WRPs. Fingerprinting with SVR using 4 WRPs achieves a result at 3.19 meters, which is better than the one with KNN but poor compared to the proposed GA with both models.

From the above observations of the conducted experiment, the proposed GA achieves better results than the conventional Fingerprinting approach. However, as the proposed GA uses user-processed motion data to create labels for the user Wi-fi RSSI data, motion error can greatly affect the estimated labels. Another factor that also has a big impact on the proposed GA performance is the number of collected WRPs. Although using fewer WRPs means the degradation of the results, the experiment had demonstrated that even with just 8 WRPs in a 10 by 10 meters area, the proposed GA was able to produce accurate estimations.

After GA outputs the estimated user track position, the track generated by the KNN model was selected because it has the lowest positioning error. Then, mapping from the estimated positions to user Wi-fi RSSI and combining with WRP dataset was done to create training data for 4 machine learning models: SVR, KNN, MLP, and RF. After the training and model optimization process, all models were saved to the central storage for evaluation in the online phase.

### C. Online Phase Results

After loading the trained model from the central storage, 27 points with RSSI measurements illustrated in Fig. 4 were used as the testing dataset for evaluating model performance. For comparison, Fingerprinting method was employed, 4 ML models similar to the last training step in the offline phase were used and they were trained and optimized on the WRP dataset. Positioning error (5) was still used as the metric to measure the distance between the estimated points and the ground truth. In addition, positioning error was calculated at different amounts of motion data in the training data from the previous section. Results of the proposed SVR, KNN, MLP, and RF along with its Fingerprinting version were shown in Fig. 9a, Fig. 9b, Fig. 9c, and Fig. 9d respectively.

Looking at the 4 graphs in Fig. 9, it is clear that all 4 models which were trained using the estimated positions achieved lower positioning error than the Fingerprinting version. It can be seen that motion error of the training set reflects on the performance of the model on the testing set and they form a linear relationship. Across 4 models, although there were some exceptions, the general trend is when the motion error of the training set grows, the positioning error of the trained model in the testing dataset increases.

In Fig. 9(a) the positioning error of the proposed SVR model on the testing set was 1.65 meters at 0% motion error and it reached 1.89 meters at 30% motion error. Conventional Fingerprinting using the SVR model had a positioning error of 2.24 meters across all levels of motion error. The difference between the two models in the worst case is 0.35 meters, which makes the Fingerprinting SVR positioning error 18% higher than the proposed SVR model. In Fig .9b, the result of the Fingerprinting KNN is 2.23 meters which is close to the Fingerprinting SVR. The proposed KNN positioning error gradually increased from 1.67 meters at 0% motion error to 1.76 meters at 25% motion error. Then a sudden jump at 30% motion error happened, which made the positioning error of the proposed KNN become 2.02 meters. Results of MLP models were illustrated in Fig. 9c. The proposed MLP error at 0% was 1.75 meters, which was the highest among the 4 proposed models but from 5% motion error, the performance was close to the proposed SVR and KNN. Finally, Fig. 9d illustrates the results of the RF model. It can be seen that the positioning error of the Fingerprinting RF was 2.3 meters and it was the highest across all models and methods used in this section. While the fingerprinting RF result was poor, the proposed RF achieved good results with positioning error at 0% motion error was 1.63 meters and got up to 1.93 at 30% motion error.

Similar to the previous section, the impact of the number of WRPs on the proposed model performance is analyzed. Training data with the lowest positioning error using 6 WRPs and 4 WRPs were selected from the previous section and became the training data for 4 machine learning models. Fig. 10 shows the comparison of positioning error among 4

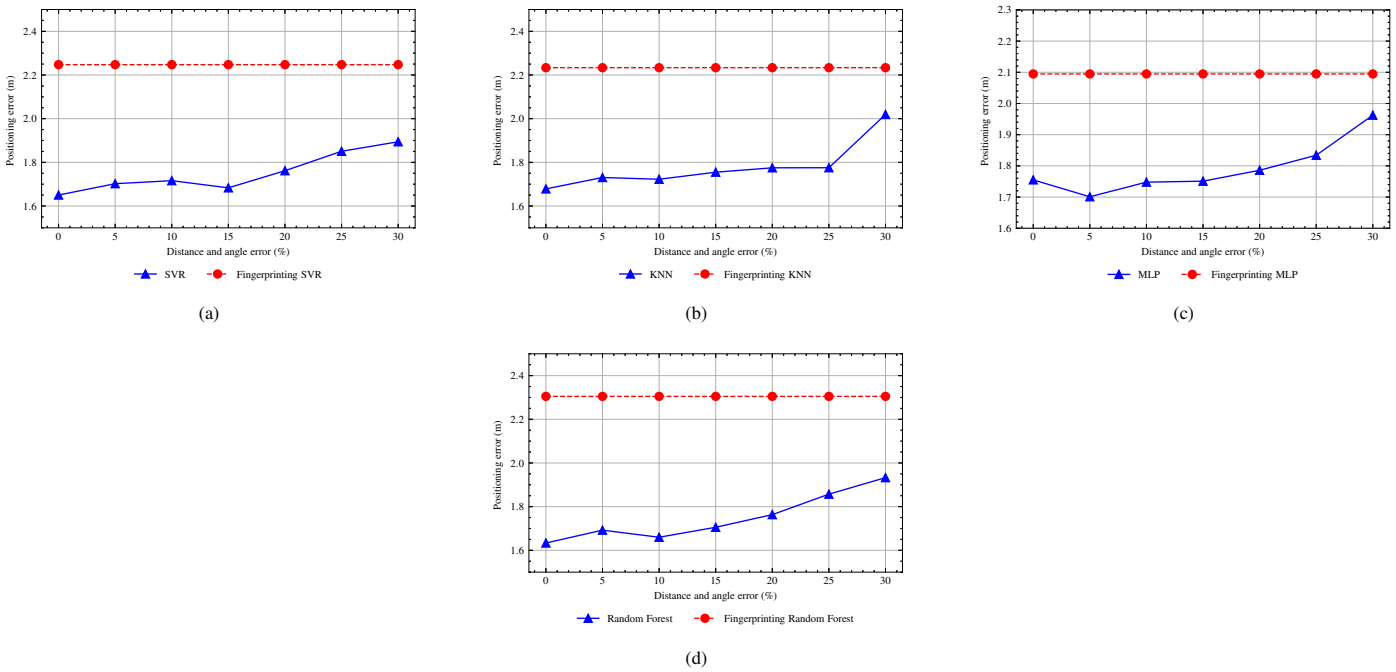



Fig. 9. Positioning Error of 4 Machine Learning Models using the Proposed Method and Conventional Fingeprinting with respect to Different Amount of Motion Error. (a) SVR, (b) KNN, (c) MLP, (d) RF

proposed models with different numbers of WRPs across 6 levels of motion error.

From Fig. 10 it can be seen that using fewer WRPs would result in the increase of positioning error for all models. However, the difference between using 8 and 6 WRPs is not significant while there is a huge gap between using 8 WRPs and 4 WRPs.

At 5% motion error in Fig. 10(a), the positioning error of all models that use 8 WRPs are close, ranging from 1.69 meters (RF) to 1.73 meters (KNN). When using 6 WRPs, all model's positioning errors had slight increases. The proposed KNN model still had the worst result at 1.86 meters following are MLP, SVR, and RF. At 6 WRPs, a big jump of positioning error can be observed across all 4 proposed models. This time, RF had the most significant increase and peaks at 2.45 meters, which was also the highest positioning error among others.

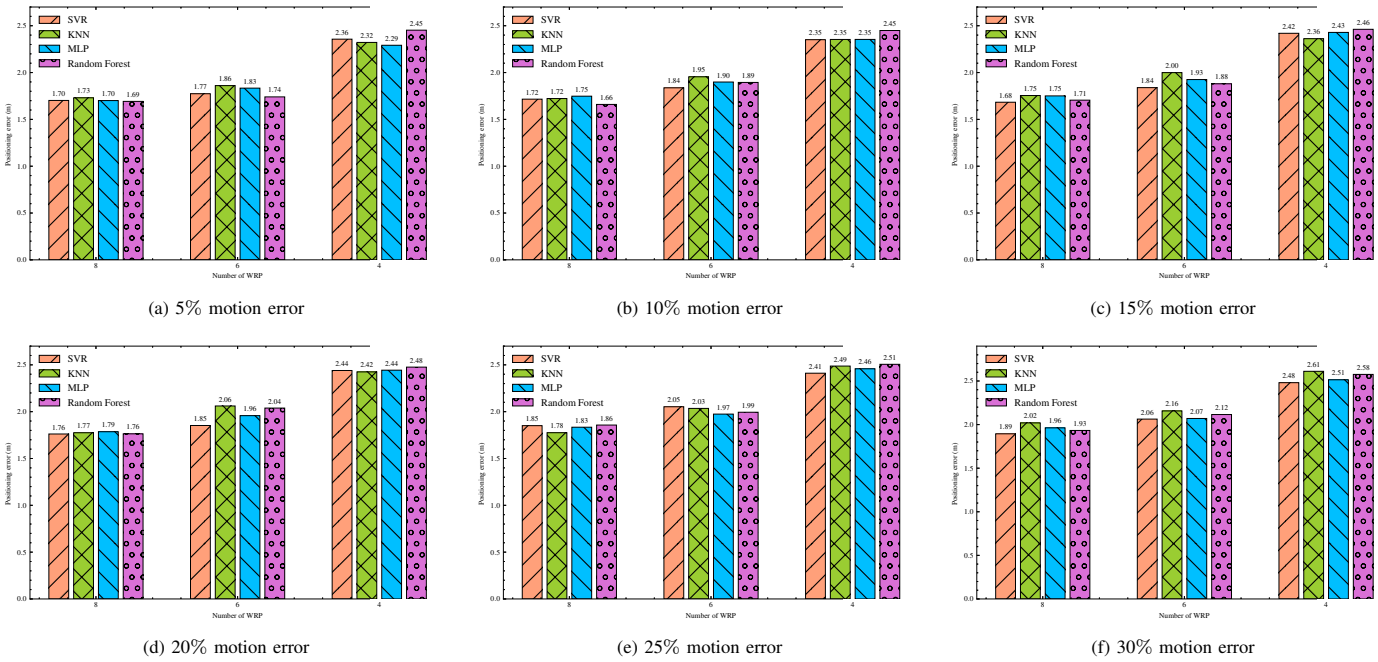
As can be seen, the trend in Fig. 10(b) to Fig. 10(c) is similar to Fig. 10(a), especially with the proposed RF model. While its results among the lowest positioning error at 8 WRPs and 6 WRPs, the performance dramatically decreased at 4 WRPs. Other proposed models had close positioning error and they only had a mild rise across levels of motion error. The proposed models at 10 % motion error that use 6 WRPs had errors ranging from 1.84 meters (SVR) to 1.95 meters (KNN) and they reached the range from 1.85 meters to 2.06 meters at 20 % motion error. A similar case can be observed with models that used 4 WRPs at 10 % motion error had positioning error ranging from 2.35 meters to 2.45 meters and they jumped to the range from 2.42 meters (KNN) to 2.48 meters (RF). As the motion error gradually increased and peaked at 30 %, Fig. 10(f) shows the worst case for all models. From Fig. 10(f), it can be seen that SVR has the best performance among others across different numbers of WRPs. It has a positioning error of 1.89 meters at 8 WRPs and peaks at 2.61 meters at 4 WRPs. On the other hand, KNN has the worst performance with positioning error at 2.02 meters at 8 WRPs and got up to 2.61 meters at 4 WRPs.

From the above analysis, it is clear that the number of WRP and proposed model performance has a close relationship. When less number WRP were used, the positioning error of the estimated user positions will rise. In addition, the training data would have less accurate data points than before. For those reasons, it is expected that the trained model performance would get poorer when the number of WRP drops. In real cases, it is important to select an appropriate number of WRP to collect. When system users want to have more accurate results, then more WRP may need to be collected. Another recommendation is the designation of the WRPs location should form a grid that covers the whole area so that they can capture the pattern of Wi-fi signal in every position.

The above experiment presented the online phase in the first run of the system. In practice, both user motion data and Wi-fi RSSI are collected and analyzed for potential usage. If the positioning error of a new user data is lower than the one that was used previously, then the model can be retrained for better performance. This creates a close loop system where user data are collected and models are updated continuously to suit the indoor environment.

## V. Conclusion

This paper proposed a new architecture in the offline phase of the Fingerprinting method. Our proposed architecture takes advantage of the user motion data and GA to create labels for the user's Wi-fi RSSI data. It enables our models to have




(a) 5% motion error (b) 10% motion error (c) 15% motion error

(d) 20% motion error (e) 25% motion error (f) 30% motion error

Fig. 10. Positioning Error of 4 Machine Learning Models with respect to the Different Number of WRPs Across Different Amount of Motion Error

more training data to accurately predict user location while reducing the amount of data that needs to be collected in the conventional Fingerprinting method. However, it is worth mentioning that our proposed offline phase procedure requires heavy computation as the fitness value of every individual in the population of GA needs to perform machine learning model training and prediction. In addition, the proposed method relies on motion data so any error, noise, and interference in the motion data can affect the output of GA and machine learning block. In the future, more studies on the application of the genetic algorithm and also the field of evolutionary computation will be conducted to improve the performance and reduce the computational complexity of the existing system.

## REFERENCES

[1] S. A. Golden and S. S. Bateman, "Sensor Measurements for Wi-Fi Location with Emphasis on Time-of-Arrival Ranging," *IEEE Trans. on Mobile Comput.*, vol. 6, no. 10, pp. 1185–1198, Oct. 2007. [Online]. Available: http://ieeexplore.ieee.org/document/4294899/

[2] C. Yang and H.-r. Shao, "WiFi-based indoor positioning," *IEEE Commun. Mag.*, vol. 53, no. 3, pp. 150–157, Mar. 2015. [Online]. Available: http://ieeexplore.ieee.org/document/7060497/

[3] F. Subhan, H. Hasbullah, A. Rozyyev, and S. T. Bakhsh, "Indoor positioning in Bluetooth networks using fingerprinting and lateration approach," in *2011 International Conference on Information Science and Applications*. Jeju Island: IEEE, Apr. 2011, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/document/5772436/

[4] D. Ahmetovic, M. Murata, C. Gleason, E. Brady, H. Takagi, K. Kitani, and C. Asakawa, "Achieving Practical and Accurate Indoor Navigation for People with Visual Impairments," in *Proceedings of the 14th International Web for All Conference*. Perth Western Australia Australia: ACM, Apr. 2017, pp. 1–10. [Online]. Available: https://dl.acm.org/doi/10.1145/3058555.3058560

[5] C. Jihong, "Patient Positioning System in Hospital Based on Zigbee," in *2011 International Conference on Intelligent Computation and Bio-Medical Instrumentation*. Wuhan, China: IEEE, Dec. 2011, pp. 159–162. [Online]. Available: http://ieeexplore.ieee.org/document/6131776/

[6] H. Kobayashi and A. F. Molisch, "Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks," *IEEE Signal Processing Magazine*, vol. 22, no. 4, p. 15, Jul. 2005.

[7] M. N. Husen and S. Lee, "Indoor human localization with orientation using WiFi fingerprinting," in *Proceedings of the 8th International Conference on Ubiquitous Information Management and Communication - ICUIMC '14*. Siem Reap, Cambodia: ACM Press, 2014, pp. 1–6. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2557977.2557980

[8] S. He and S.-H. G. Chan, "Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons," *IEEE Commun. Surv. Tutorials*, vol. 18, no. 1, pp. 466–490, 2016. [Online]. Available: http://ieeexplore.ieee.org/document/7174948/

[9] F. Zafari, A. Gkelias, and K. K. Leung, "A Survey of Indoor Localization Systems and Technologies," *IEEE Commun. Surv. Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019. [Online]. Available: https://ieeexplore.ieee.org/document/8692423/

[10] D. Tinh Pham and T. T. Ngoc Mai, "Ensemble learning model for Wifi indoor positioning systems," *IJ-AI*, vol. 10, no. 1, p. 200, Mar. 2021. [Online]. Available: http://ijai.iaescore.com/index.php/IJAI/article/view/20603

[11] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2. Tel Aviv, Israel: IEEE, 2000, pp. 775–784. [Online]. Available: http://ieeexplore.ieee.org/document/832252/

[12] A. Anjum and M. U. Ilyas, "Activity recognition using smartphone sensors," in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*. Las Vegas, NV: IEEE, Jan. 2013, pp. 914–919. [Online]. Available: http://ieeexplore.ieee.org/document/6488584/

[13] W. Kang and Y. Han, "SmartPDR: Smartphone-Based Pedestrian Dead Reckoning for Indoor Localization," *IEEE Sensors J.*, vol. 15, no. 5, pp. 2906–2916, May 2015. [Online]. Available: http://ieeexplore.ieee.org/document/6987239/

[14] A. Jimenez, F. Seco, C. Prieto, and J. Guevara, "A comparison of Pedestrian Dead-Reckoning algorithms using a low-cost MEMS IMU," in *2009 IEEE International Symposium on Intelligent Signal Processing*. Budapest, Hungary: IEEE, Aug. 2009, pp. 37–42. [Online]. Available: http://ieeexplore.ieee.org/document/5286542/

[15] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*. Zurich Switzerland: ACM, Sep. 2013, pp. 225–234. [Online]. Available: https://dl.acm.org/doi/10.1145/2493432.2493449

[16] Q. Wang, L. Ye, H. Luo, A. Men, F. Zhao, and Y. Huang, "Pedestrian Stride-Length Estimation Based on LSTM and Denoising Autoencoders," *Sensors*, vol. 19, no. 4, p. 840, Feb. 2019. [Online]. Available: http://www.mdpi.com/1424-8220/19/4/840

[17] M. J. Abadi, L. Luceri, M. Hassan, C. T. Chou, and M. Nicoli, "A collaborative approach to heading estimation for smartphone-based PDR indoor localisation," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. Busan, South Korea: IEEE, Oct. 2014, pp. 554–563. [Online]. Available: http://ieeexplore.ieee.org/document/7275528/

[18] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear," *Proc. SPIE 3068, Signal Processing, Sensor Fusion, and Target Recognition VI*, p. 12, Jul. 1997.

[19] S. O. H. Madgwick, "An efficient orientation filter for inertial and inertial/magnetic sensor arrays," 2010.

[20] A. Khalajmehrabadi, N. Gatsis, and D. Akopian, "Modern WLAN Fingerprinting Indoor Positioning Methods and Deployment Challenges," *IEEE Commun. Surv. Tutorials*, vol. 19, no. 3, pp. 1974–2002, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7874080/

[21] B. Ferris, "WiFi-SLAM Using Gaussian Process Latent Variable Models," *Proceedings of the 20th international joint conference on Artifical intelligence*, Jan. 2007.

[22] A. Naguib, P. Pakzad, R. Palanki, S. Poduri, and Y. Chen, "Scalable and accurate indoor positioning on mobile devices," in *International Conference on Indoor Positioning and Indoor Navigation*. Montbeliard, France: IEEE, Oct. 2013, pp. 1–10. [Online]. Available: http://ieeexplore.ieee.org/document/6817856/

[23] I. Constandache, R. R. Choudhury, and I. Rhee, "Towards Mobile Phone Localization without War-Driving," in *2010 Proceedings IEEE INFOCOM*. San Diego, CA, USA: IEEE, Mar. 2010, pp. 1–9. [Online]. Available: http://ieeexplore.ieee.org/document/5462058/

[24] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*, 1st ed., ser. Complex adaptive systems. Cambridge, Mass: MIT Press, 1992.

[25] A. L. Nelson, G. J. Barlow, and L. Doitsidis, "Fitness functions in evolutionary robotics: A survey and analysis," *Robotics and Autonomous Systems*, vol. 57, no. 4, pp. 345–370, Apr. 2009. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S0921889008001450