WI-FI-BASED INDOOR LOCALIZATION USING MODEL-BASED AND DATA-DRIVEN APPROACHES

by

AYOUB IDELHAJ

Presented to the Faculty of the Graduate School of
The University of Texas at Arlington in Partial Fulfillment
of the Requirements
for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON
August 2021

ACKNOWLEDGEMENTS

I would like to show my profound gratitude and appreciation to my supervising professor Dr. Yan Wan for her lasting contributions and guidance in the completion process of this project. We had very engaging and productive weekly meetings and discussions about multiple aspects of this project. Given the unprecedented circumstances resulting from COVID-19, Dr. Wan was very supportive, offering all the Dynamic Networks and Control lab resources to conduct the project's experimental design stages.

I appreciate Dr. Devarajan Venkat and Dr. Ioannis Schizas for taking the time to serve on my thesis committee.

I have a special appreciation to all of the undergraduate and graduate faculty for whom they have shaped and influenced my educational journey at UTA. They were a source of motivation and inspiration.

I thank my lab partner Chellappan Lakshmi for her assistance in the data collection of this project.

I want to show my greatest appreciation to my beloved family for their endless emotional and financial support and encouragement throughout the stages of my academic life, especially my brothers Miloud and Lahcen, my sisters Souad and Rabia, and my parents.

July 23, 2021

ABSTRACT

WI-FI-BASED INDOOR LOCALIZATION USING MODEL-BASED AND DATA-DRIVEN APPROACHES

AYOUB IDELHAJ, M.S.

The University of Texas at Arlington, 2021

Supervising Professor: Dr. Yan Wan

This thesis investigates model-based and data-driven approaches for indoor localization using the Received Signal Strength Indicator (RSSI) of Wi-Fi signals. We study multiple model-based indoor localization approaches, including the free space path loss model, the log-distance path loss model, the International Telecommunication Union (ITU) model, and a nonlinear regression model. We examine their indoor localization accuracy using raw RSSI values, and filter RSSI values passed through a Moving Average filter and a Kalman filter. For data driven approaches, we employ a family of Extreme Learning Machine (ELM) algorithms including Basic-ELM, Online Sequential-ELM (OS-ELM), Hierarchical-ELM (H-ELM), and Kernel-ELM (K-ELM), to find the indoor position. We provide simulation results comparing the performances of both the Machine-learning based approaches and model-based approaches in terms of localization error to identify the algorithms with the lowest localization error.

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

Figure Page

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Localization is one of the most fundamental tasks for autonomy and robotic applications. Global Positioning System (GPS) has been a popular localization system in open outdoor environments [1]. However, in indoor environments, GPS falls short due to the unavailability of line-of-sight between the satellite and the receiver. Indoor localization has attracted broad interests given its wide applications in construction, health, cargo transport [2, 3, 4]. A variety of indoor localization system technologies have been employed in practice [5] and are categorized as, inertial-based localization [6], magnetic-based localization [7], ultrasound-based localization [8], optical-based localization [9], and Radio Frequency (RF) based localization [10, 11]. RF-based devices are commonly implemented in indoor localization because of their availability, simplicity, and cost-effectiveness. These standard devices include WiFi routers, Bluetooth modules, Cellphones, and Ultra-Wide Band (UWB) modules. We consider using WiFi in this thesis, given its availability in most buildings, ease of use, range capability, and that no additional infrastructure is needed. Other than implementation devices, it is also important to consider the choice of measurement signals for indoor localization [12, 10]. Choices include Received Signal Strength Indicator (RSSI), Time of Arrival (ToA), Angle of Arrival (AoA), and Time Difference of Arrival (TDoA). All have been employed to estimate the distance between the target and the Access Point (AP) nodes. RSSI measures the signal strength intensity from several APs to a client device. We consider to use it in this thesis because of its ease in implementation, and that no time synchronization is needed.

The analysis of indoor localization problem can be approached using model-based and data-driven methods. Many studies have investigated model-based indoor localization are given as follows. Paper [13] proposes an algorithm that integrates the signal measurement from an Inertial Measurement Unit (IMU), UWB, and the velocity and the heights acquired by the lower body biomechanical model to track the location and the motion of humans under multiple activities. Paper [9] uses a vision system to estimate the 3-D pose of a UAV through its motion. Paper [14] introduces an accurate and scalable system to determine the user location. It uses the probability distribution function to address the noise and interference in wireless channels and enhances indoor localization accuracy. It also uses a common clustering technique to reduce the computational requirements. Paper [15] utilizes the Extended Kalman Filter based on TDoA measurement signals to estimate the location of the UAV in real-time. In this thesis, the model-based approaches are narrowed down to WiFi propagation models. The machine learning-based approaches are also considered in indoor localization due to the complexity of noise and interference. Once the training phase is completed, these algorithms have the potential to produce a better positioning accuracy. Paper [16] uses the Convolutional Neural Network (CNN) in indoor settings where a Stacked Auto Encoder is combined with a one-dimensional CNN. Paper [17] proposes a Gradient Fingerprinting method, which produces a more stable RSSI gradient adaptive to the time-variant RSSI in indoor environments. To estimate the unknown location in [18], Artificial Neural Networks (ANN) is applied as a function approximation to map the RSSI measurements to coordinates on the plane. Paper [19] adopts the Recurrent Neural Networks (RNN) for indoor localization, where the RNN aims at finding trajectory positions by taking into account the correlation among the RSSI samples in a trajectory. Paper [20] cascades a two-stage machine learning approach for robust indoor localization. In the first stage, k-Nearest Neighbor (k-NN) is employed to identify the type of indoor environment based on real data measurements. In the second stage, k-NN is used to distinguish the appropriate selection of RF features that produce the highest indoor localization accuracy. Paper [21] prposes a scalable Deep Neural Network (DNN) for indoor localization based on Wi-Fi fingerprinting method.

The primary focus of this study, Extreme Learning Machine (ELM) is a single hidden layer feed-forward network characterized by its good performance and fast learning capabilities that can reduce learning time while achieving accurate results with minimal training data [22]. ELM employs random mapping from the input to the single hidden layer, and then a single matrix inversion is calculated to determine the output weights from the hidden layer to the output of the network. ELM is considered an effective machine learning method that can easily fit to learning tasks in mobile devices with limited computational power, and has been used in many machine learning applications. Paper [23] combines CNN and ELM in the application of lane detection. [24] applies ELM-based probabilistic forecasting method for wind power generation. The study in [25] employs ELM to estimate the energy consumption, where it is applied to building material thicknesses and their thermal insulation capability. Paper [26] uses ELM to examine the relationship between the sales amount and some factors affecting the demand. Most relevant to our study, paper [27] employs the Online Sequential-ELM (OS-ELM) in the application of indoor localization, where the method is shown to shorten the time spent during the offline phase, and adapts to changes in the environment due to its online sequential stage.

The main contributions of the thesis are listed as follows: 1) comparing the model-based approaches to the ELM based approaches in indoor localization, 2) comprehensively exploring multiple ELM algorithms (Basic ELM, Online Sequential ELM (OS-ELM), Hierarchical ELM (H-ELM), Kernel ELM (K-ELM) and applying them to the indoor localization problem, 3) investigating the effect of filtering schemes in improving the accuracy in the model-based approaches, and 4) using the correlation coefficient as a measure to determine the quality of RSSI samples.

The rest of this thesis is organized as follows. Section 2 describes the RSSI model-based indoor localization propagation and regression models. Section 3 defines and analyzes basic ELM, OS-ELM, H-ELM, and K-ELM concepts. Section 4 shows a detailed experimental design and setup. Section 5 analyzes experimental results, and conducts comparative performance evaluation. Finally, the conclusion and additional remarks to consider are presented in Section 6.

CHAPTER 2

MODEL-BASED INDOOR LOCALIZATION

In the model-based approach, four commonly used methods are adopted for localization, including the path loss model, log-distance path loss model, International Telecommunication Union (ITU) model, and nonlinear regression-based model. To use these methods, the RSSI measurements from various AP nodes are obtained, and then incorporated as input signals into each of the four models. The models then provide the estimated distances between the AP nodes and the testing points to be used further to estimate the positions of testing points in the traveling path of the robot. In this section, we first describe each of these models and then the techniques to use these estimated distances to approximate positions.

2.1 Indoor Propagation Models:

Indoor propagation models express transmitted signal strengths as functions of distances between a pair of transceiver and receiver. We consider an area where AP nodes are placed and $R_{i,k}$ is the ith RSSI measurement made for the kth AP at a testing point. The estimated distance between the kth AP and the testing point is denoted as $\hat{d}_k$. We introduce four models in this section. In a real indoor environment, these models may not be ideal due to many factors (e.g. obstacles, signals interference, and multipath effect) that impact the localization accuracy. The indoor propagation models are closely dependent on several parameters that capture the dissipation effect of signal energy in amplitude, phase, and time in a varying environment.

2.1.1 Free Space Path Loss Model:

Equation (2.1) is the simplified free space model that links an RSSI measurement with the estimated distance [28]. This propagation model shows the decrease of RSSI as the distance increases logarithmically [29].

$$R_{i,k}[dBm] = -10 n_k log(\hat{d}_k) + C_k, \tag{2.1}$$

where n_k is the path loss exponent and the constant C_k captures RSSI measurements at a reference distance. Equation (2.1) leads to the estimated distance as

$$\hat{d}_k = 10^{\left(\frac{C_k - R_{i,k}}{10 n_k}\right)}. \tag{2.2}$$

The estimation of n_k and C_k can have a great effect on the correctness of this model. Considering that this model does not capture the shadowing and multi-path effects and their impact on the transmitted signals, the accuracy of the estimated positions can degrade significantly.

2.1.2 Log-Distance Path Loss Model:

Equation (2.3) takes into account the shadowing effects caused by different objects forcing the traveled signals to be reflected, absorbed, and refracted [29][30]. As a consequence, they generate a reduction in the signals' transmitted power,

$$R_{i,k}[dBm] = -10 n_k log(\hat{d}_k) + C_k + X_k. \tag{2.3}$$

Rearranging (2.3) gives the estimated distance of

$$\hat{d}_k = 10^{\left(\frac{C_k + X_k - R_{i,k}}{10 n_k}\right)}. \tag{2.4}$$

Here, X_k is the zero mean Gaussian random variable having a standard deviation σ_k in dBm, N is the number of RSSI samples, and $\overline{R}_k$ is the average value of the RSSI measurements,

$$\sigma_k = \frac{1}{N}\sqrt{\sum_{i=1}^{N}\left(R_{i,k} - \overline{R}_k\right)^2}, \tag{2.5}$$

$$\overline{R}_k = \frac{1}{N}\sum_{i=1}^{N} R_{i,k}. \tag{2.6}$$

2.1.3 International Telecommunication Union (ITU) model:

The ITU model was specifically designed to capture the path loss in indoor settings (e.g., residential, office, and commercial areas), where many components are considered, including the materials inside the building, the number of floors in a building, etc. This model uses several parameters such as, the frequency f that the AP nodes operate on, the distant power loss coefficient N, the floor penetration factor L_f, and n number of floors between the floor and base. The ITU model is captured by the equation below,

$$R_{i,k}[dBm] = 20log(f) + Nlog(\hat{d}_k) + L_f(n) - 28 \tag{2.7}$$

Solving for $\hat{d}_k$ yields

$$\hat{d}_k = 10^{\left(\frac{28+R_{i,k}-20log(f)-L_f(n)}{N}\right)}. \tag{2.8}$$

All the coefficients in (2.7) can be designed based on the recommendation of [31].

2.2 Nonlinear Regression Model:

We can also use nonlinear regression models to capture indoor propagation. Based on the RSSI data collected from several AP nodes at various testing locations, the model relates the transmitted signals to their transmitted distances. As the example shown in Figure 4.3, with the increase of distances, the RSSI values are dissipated, meaning that the strength of the signal weakens. Therefore, $\hat{d}$ and the RSSI values can be related by fitting the set of data from N AP nodes through a n^{th} degree polynomial as [32]

$$\hat{d}_k = \sum_{l=0}^{n} \alpha_l \overline{R}_k{}^l, \tag{2.9}$$

where α_l, $l \in \{1,...,n\}$ are the coefficients of the nonlinear polynomial. The error e_k is found by taking the difference between the true and estimated distances, which gives,

$$e_k = d_k - \hat{d}_k = d_k - \sum_{l=0}^{n} \alpha_l \overline{R}_k{}^l. \tag{2.10}$$

Here d_k is the true distance between AP node and a testing point. The goal here is to solve for the coefficients (α_l) by minimizing the sum of the error squares,

$$E = \sum_{k=1}^{N} e_k^2 = \sum_{k=1}^{N} \left(d_k - \sum_{l=0}^{n} \alpha_l \overline{R}_k{}^l \right)^2. \tag{2.11}$$

In order to minimize the error, (2.11) is differentiated with respect to α_j, $j \in \{0,...,n\}$ and set to zero

$$\frac{\partial E}{\partial \alpha_j} = -2 \sum_{k=1}^{N} \overline{R}_k{}^j \left(d_k - \sum_{l=0}^{n} \alpha_l \overline{R}_k{}^l \right) = 0, \tag{2.12}$$

where,

$$\sum_{k=1}^{N} \overline{R}_k{}^j d_k = \sum_{k=1}^{N} \overline{R}_k{}^j \sum_{l=0}^{n} \alpha_l \overline{R}_k{}^l. \tag{2.13}$$

For simplification purposes, j is set to 0 and we obtain

$$d = \begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix} = \begin{bmatrix} 1 & \overline{R}_1 & \overline{R}_1{}^2 & \dots & \overline{R}_1{}^n \\ 1 & \overline{R}_2 & \overline{R}_2{}^2 & \dots & \overline{R}_2{}^n \\ \vdots & \vdots & \vdots & \dots & \vdots \\ 1 & \overline{R}_N & \overline{R}_N{}^2 & \dots & \overline{R}_N{}^n. \end{bmatrix} \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix} = R \begin{bmatrix} \alpha_0 \\ \alpha_1 \\ \vdots \\ \alpha_n \end{bmatrix}. \tag{2.14}$$

The coefficients α_l are determined by using the least square estimation, $\alpha = (R^T R)^{-1} R^T d$. The solution of α_l leads to the estimated distances by employing (2.9).

2.3 Position Estimation:

The above methods provide the estimated distances between the AP nodes and the locations of the testing points. With these distances, linear and nonlinear least squares estimations can be applied to estimate the position of the testing points.

2.3.1 Linear Least Squares Estimation (LLS):

To relate the estimated distances to the estimated positions $(\hat{x}, \hat{y})$, the distance formula are utilized [33],

$$\begin{aligned} \hat{d_1}^2 &= (\hat{x} - x_1)^2 + (\hat{y} - y_1)^2 \\ \hat{d_2}^2 &= (\hat{x} - x_2)^2 + (\hat{y} - y_2)^2 \\ &\vdots \\ \hat{d_k}^2 &= (\hat{x} - x_k)^2 + (\hat{y} - y_k)^2 \end{aligned}, \tag{2.15}$$

where x and y capture the position of the AP nodes. Subtracting 1st equation from system of equations in (2.15) yields,

$$\begin{aligned} \hat{d_2}^2 - \hat{d_1}^2 &= 2\hat{x}x_1 - 2\hat{x}x_2 + 2\hat{y}y_1 - 2\hat{y}y_2 + x_2^2 - x_1^2 + y_2^2 - y_1^2 \\ &\vdots \\ \hat{d_k}^2 - \hat{d_1}^2 &= 2\hat{x}x_1 - 2\hat{x}x_k + 2\hat{y}y_1 - 2\hat{y}y_k + x_k^2 - x_1^2 + y_k^2 - y_1^2 \end{aligned}. \tag{2.16}$$

Expressing (2.16) in a matrix form

$$A\hat{X} = b, \tag{2.17}$$

where,

$$A = \begin{bmatrix} x_2 - x_1 & y_2 - y_1 \\ x_3 - x_1 & y_3 - y_1 \\ \vdots & \vdots \\ x_k - x_1 & y_k - y_1 \end{bmatrix}, and \quad b = \frac{1}{2} \begin{bmatrix} x_2^2 - x_1^2 + y_2^2 - y_1^2 + \hat{d}_1^2 - \hat{d}_2^2 \\ x_3^2 - x_1^2 + y_3^2 - y_1^2 + \hat{d}_1^2 - \hat{d}_3^2 \\ \vdots \\ x_k^2 - x_1^2 + y_k^2 - y_1^2 + \hat{d}_1^2 - \hat{d}_k^2 \end{bmatrix}. \tag{2.18}$$

$$\hat{X} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix}. \tag{2.19}$$

As observed in (2.18), equation (2.17) can be solved in a form of linear least squares estimation

$$Min||A\hat{X} - b||, \tag{2.20}$$

with the unique solution,

$$\hat{X} = \begin{bmatrix} \hat{x} \\ \hat{y} \end{bmatrix} = \left(A^T A\right)^{-1} A^T b. \tag{2.21}$$

To optimize the results of (2.21), a constrained equation can be introduced by taking the average of (2.15),

$$\frac{1}{N}\sum_{k=1}^{N}\left((\hat{x}-\hat{x}_k)^2+(\hat{y}-\hat{y}_k)^2\right) = \frac{1}{N}\sum_{k=1}^{N}\hat{d_k}^2. \tag{2.22}$$

Now matrices A, and b are expressed as follows,

$$A = \begin{bmatrix} x_1 - \frac{1}{N}\sum_{k=1}^{N}\hat{x}_k & y_1 - \sum_{k=1}^{N}\hat{y}_k \\ \vdots & \vdots \\ x_N - \frac{1}{N}\sum_{k=1}^{N}\hat{x}_k & y_N - \sum_{k=1}^{N}\hat{y}_k \end{bmatrix} \tag{2.23}$$

$$b = \frac{1}{2}\begin{bmatrix} x_1^2 - \frac{1}{N}\sum_{k=1}^{N}\hat{x_k}^2 + y_1^2 - \frac{1}{N}\sum_{k=1}^{N}\hat{y_k}^2 + \frac{1}{N}\sum_{k=1}^{N}\hat{d_k}^2 - \hat{d_1}^2 \\ \vdots \\ x_N^2 - \frac{1}{N}\sum_{k=1}^{N}\hat{x_k}^2 + y_N^2 - \frac{1}{N}\sum_{k=1}^{N}\hat{y_k}^2 + \frac{1}{N}\sum_{k=1}^{N}\hat{d_k}^2 - \hat{d_N}^2 \end{bmatrix}. \tag{2.24}$$

2.3.2 Non-Linear Least Squares Estimation (NLS):

An alternative to solving the localization problem is to consider the NLS approach, where the goal is to minimize the cost function J. The sum of the squares of errors in (2.25) is minimized through an iterative method. In this application, the gradient descent method is used [33, 34],

$$J = \frac{1}{2N}\sum_{k=1}^{N}\left(\sqrt{(x_k-\hat{x})^2+(y_k-\hat{y})^2} - d_k\right)^2. \tag{2.25}$$

Implementing the gradient descent method requires picking an appropriate learning factor ($0 < \alpha < 1$), where a smaller α will slow down the convergence to the global minimum, and a larger α may lead to the divergence from the global minimum. In addition, picking different starting points can help to avoid mistaken local minima as the global minimum. Algorithm 1 summarizes the procedures to implement the gradient descent method.

Algorithm 1 To Estimate The Position $(\hat{x}, \hat{y})$

```
Input:
    Define a starting point $(x_1, x_2)$.
    Set a learning factor $\alpha$ and $N$ maximum number of iterations.
    Set termination tolerance $M$ and gradient norm $G$.
    Set the counter $i$.
Output:
while $(i <= N$ and $G >= M)$ do
    for j= 1:2 do
        Calculate the gradient $g_j = \frac{\partial J}{\partial x_j}$.
        Minimize the error $temp_j = x_j - \alpha g_j$.
        Store $x_j = temp_j$.
    end for
    Calculate the norm $G = norm(g)$.
    Update the counter $i = i + 1$.
end while
```

where,

$$\begin{aligned} \frac{\partial J}{\partial \hat{x}} &= \frac{1}{N}\sum_{k=1}^{N}\left(\sqrt{(x_k-\hat{x})^2+(y_k-\hat{y})^2}-d_k\right)^2 \frac{(x_k-\hat{x})}{\sqrt{(x_k-\hat{x})^2+(y_k-\hat{y})^2}}. \\ \frac{\partial J}{\partial \hat{y}} &= \frac{1}{N}\sum_{k=1}^{N}\left(\sqrt{(x_k-\hat{x})^2+(y_k-\hat{y})^2}-d_k\right)^2 \frac{(y_k-\hat{y})}{\sqrt{(x_k-\hat{x})^2+(y_k-\hat{y})^2}}. \end{aligned} \tag{2.26}$$

The model-based methods have their limitations in solving the localization problem. The models include multiple environment-related parameters that are sensitive to the changes in the experimental setup, which affect the accuracy of final estimated testing point locations. It is worth noting that the testing points are equivalent to training points when processed using these methods. Figure 2.1 below gives detailed procedures for the implementation of the model based approaches.

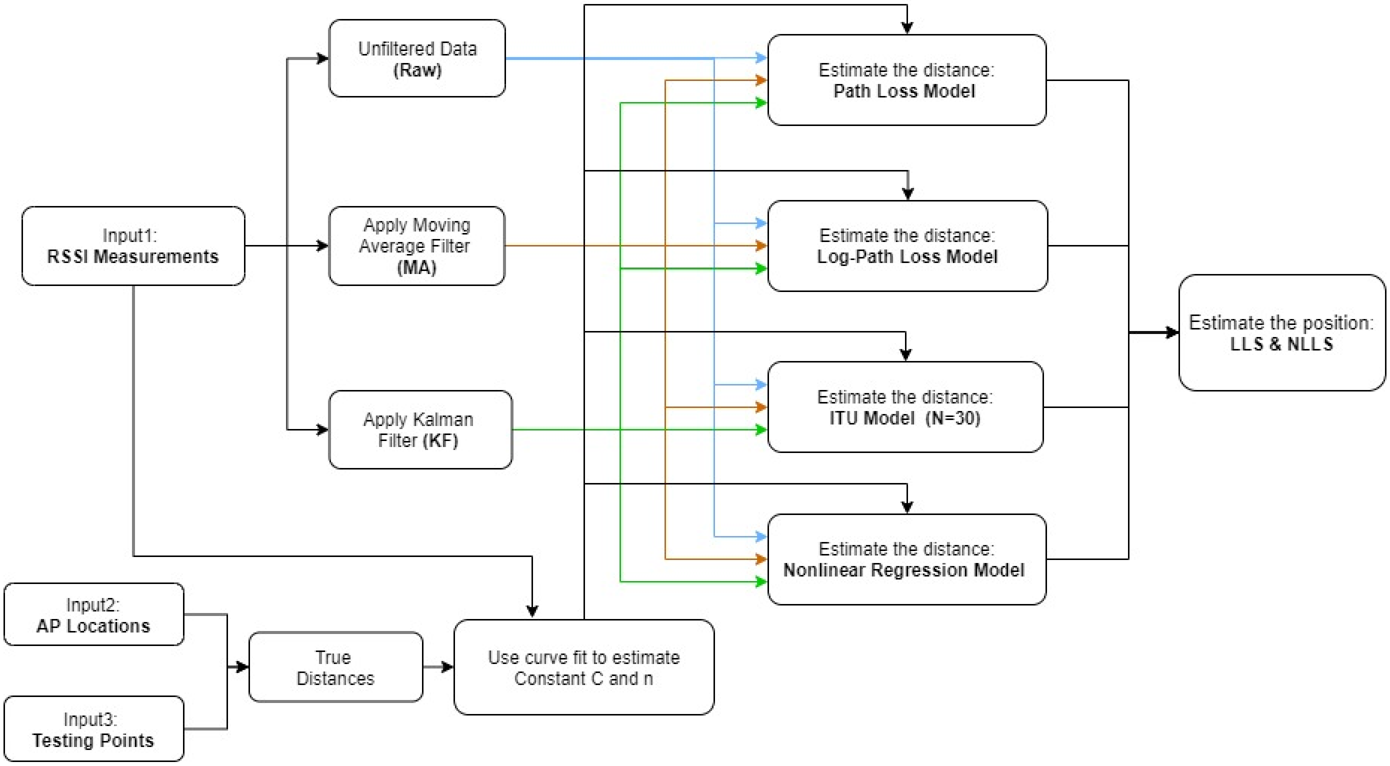
Unfiltered Data (Raw)
Input1: RSSI Measurements
Apply Moving Average Filter (MA)
Apply Kalman Filter (KF)
Estimate the distance: Path Loss Model
Estimate the distance: Log-Path Loss Model
Estimate the distance: ITU Model (N=30)
Estimate the distance: Nonlinear Regression Model
Estimate the position: LLS & NLLS
Input2: AP Locations
Input3: Testing Points
True Distances
Use curve fit to estimate Constant C and n

Figure 2.1. The overall block diagram of the model based approaches.

CHAPTER 3

MACHINE LEARNING-BASED INDOOR LOCALIZATION

In this section, we study machine learning-based methods to solve the localization problem. Unlike the model-based methods, machine learning algorithms can easily learn and adapt to a changing environment. Generally, machine learning methods are implemented in two steps. First, the model is trained by processing the data to solve the optimization problem. Second, the model is tested by evaluating the performance of the trained model. There are varieties of machine learning algorithms adopted in the the literature. Deep learning algorithms (e.g., Deep Neural Network and Convolutional Neural Network), when trained, require multiple hidden layers, adding complexity to the computation. In addition, deep learning algorithms suffer from two major issues; over-fitting and the long computation time. As an alternative, this section focuses on analyzing four algorithms that belong to the ELM family, which are characterized by their fast learning capabilities, making them suitable for mobile edge computing on low-computing power platforms such as robots and UAVs. The Basic-ELM is introduced first, which is an integral component of all types of ELM algorithms. Then the OS-ELM is studied, where it examines the data differently using a two-phase approach. The K-ELM is next analyzed, offering a single step kernel version of the ELM, making it simpler to implement than other kernel methods. Finally, The H-ELM is considered for multi-layer perceptron, which is effective in deep learning applications. H-ELM is divided into two main components, the unsupervised feature learning phase, which integrates the sparse ELM autoencoder, and the supervised feature learning phase that uses the original

3.1 Basic-ELM

ELM is a type of a single hidden layer feed-forward neural network (SLFN) with *L* numbers of hidden nodes (see figure 3.1). It is easier to implement than most gradient descent algorithms for feedforward neural networks. The main feature of the ELM algorithm is that it generally provides a good performance at a fast learning speed. The ELM network is trained by only estimating its output weights, while leaving the parameters in other layers random, thus guaranteeing a shorter learning time [22]. The randomness in the ELM was observed and supported in biological studies [35]. In contrary to the traditional learning algorithms, ELM learns without adjusting the parameter of the hidden neurons, thus making it effective across number of applications [22][36].

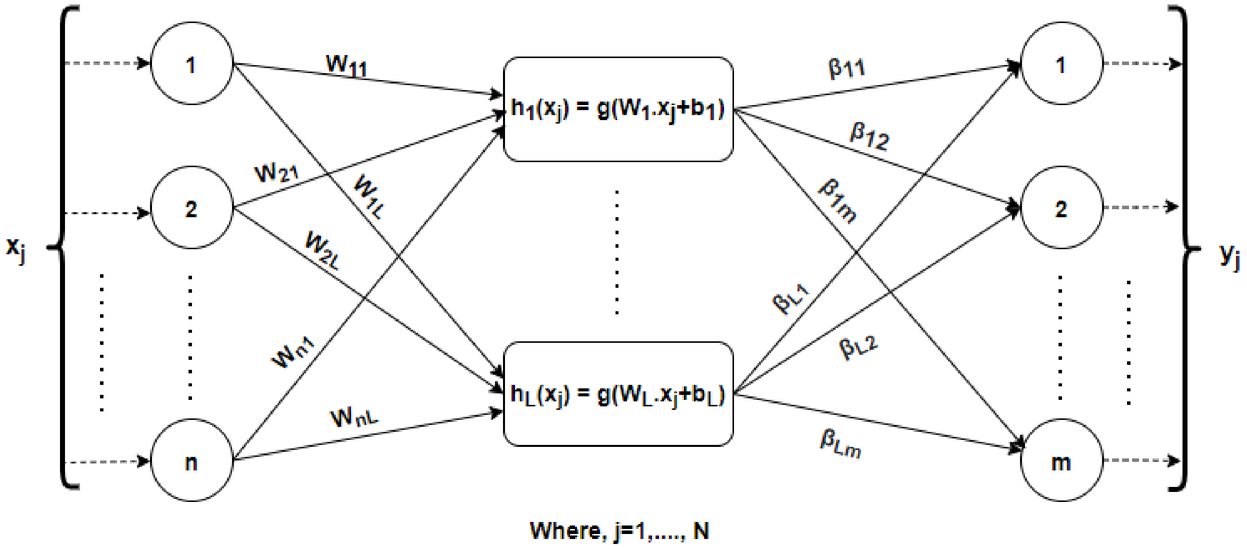

Figure 3.1. ELM network with a single hidden layer and L number of hidden nodes.

Let (x_j, t_j) denote the jth training samples, where $j = \{1,...,N\}$. Here the input $x_j = [x_{j1}, \ldots, x_{jn}]^T$ and the training targets $y_j = [y_{j1}, \ldots, y_{jm}]^T$. The output of a SLFN is defined as

$$\sum_{i=1}^{L} \beta_i g(W_i \cdot x_j + b_i) = y_j \quad j = 1, \ldots, N \tag{3.1}$$

where i is the i-th hidden node $i \in \{1,...,L\}$, the output weight vector $\beta_i = [\beta_{i1}, \ldots, \beta_{im}]^T$ connects the i-th hidden node to the output nodes, and $g(.)$ is an infinitely differentiable nonlinear activation function whose arguments include W_i and b_i. $W_i = [W_{1i}, \ldots, W_{ni}]^T$ expresses an input weight vector connecting the n input nodes to the i-th hidden node, and b_i represents a bias vector. Primarily, the core of the ELM algorithm is centered on estimating the output weights that minimize the error function. In order to estimate β, (3.1) can be written in a matrix form as follows:

$$H\beta = Y, \tag{3.2}$$

where,

$$H = \begin{bmatrix} h(x_1) \\ h(x_2) \\ \vdots \\ h(x_N) \end{bmatrix} = \begin{bmatrix} h_1(x_1) & h_2(x_1) & \ldots & h_L(x_1) \\ h_1(x_2) & h_2(x_2) & \ldots & h_L(x_2) \\ \vdots & \vdots & \ldots & \vdots \\ h_1(x_N) & h_2(x_N) & \ldots & h_L(x_N) \end{bmatrix} = \tag{3.3}$$

$$\begin{bmatrix} g(W_1 \cdot x_1 + b_1) & g(W_2 \cdot x_1 + b_2) & \ldots & g(W_L \cdot x_1 + b_L) \\ g(W_1 \cdot x_2 + b_1) & g(W_2 \cdot x_2 + b_2) & \ldots & g(W_L \cdot x_2 + b_L) \\ \vdots & \vdots & \ldots & \vdots \\ g(W_1 \cdot x_N + b_1) & g(W_2 \cdot x_N + b_2) & \ldots & g(W_L \cdot x_N + b_L). \end{bmatrix} \tag{3.4}$$

The elements in the first column of H equivalent to the samples of the first hidden layer $h_1(x_j)$ are expanded as

$$h_1(x_j) = \begin{bmatrix} h_1(x_1) \\ \vdots \\ h_1(x_N) \end{bmatrix} = \begin{bmatrix} g(W_1 \cdot x_1 + b1) \\ \vdots \\ g(W_1 \cdot x_N + b1) \end{bmatrix} = \begin{bmatrix} g(W_{11}x_{11} + W_{21}x_{12} + \ldots + W_{n1}x_{1n} + b_1) \\ \vdots \\ g(W_{11}x_{N1} + W_{21}x_{N2} + \ldots + W_{n1}x_{Nn} + b_1) \end{bmatrix}. \tag{3.5}$$

In addition,

$$\beta = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix} \qquad \mathit{and} \qquad Y = \begin{bmatrix} y_1^T \\ \vdots \\ y_N^T \end{bmatrix} \tag{3.6}$$

The least squares solution solves equation (3.2) by providing the best estimation in the sense that the sum of squares of the difference $H\beta - Y$ is minimized, giving the unique solution,

$$\hat{\beta} = \left(H^T H\right)^{-1} H^T Y. \tag{3.7}$$

As shown in the equations above, H is the hidden layer output matrix of the network which contains the output of each hidden node for each training sample, and Y is the target matrix. Since the input weights and the bias are generated randomly, the parameters to learn are $\hat{\beta}$. When implementing the ELM, the number of hidden nodes and the types of the activation function need to be carefully designed. The ELM algorithm below, summarizes the steps taken to estimate $\hat{\beta}$.

```
Algorithm 2 ELM Algorithm
Input:
    Set $W_i$, and $b_i$ randomly.
    Load input data $x_j$.
    Set $L$.
Output:
    Define non-linear activation function g(.).
    Calculate $H = g(W_i \cdot x_j + b_i)$.
    Calculate $\hat{\beta} = \left(H^T H\right)^{-1} H^T Y$.
```

3.2 OS-ELM

Building on the basis of ELM, Online Sequential-ELM (OS-ELM) operates in two phases: the initialization phase for offline training, and the online sequential learning phase for online localization. OS-ELM has features that help address two crucial problems for indoor localization. First, it reduces the lengthy process needed in building the offline database each time the experiment is executed differently. Second, it provides a solution that can adapt to changes in the environment through its online sequential learning phase. It is worth noting that online learning models are trained incrementally with data fed to the system either one by one or block by block [27]. Before examining the two phases of the OS-ELM, we introduce the following parameters. D is the training data in sequential order, $D = [(x_i,t_i)|x_i \varepsilon R^n, t_i \varepsilon R^m, i = 1,...,N]$. The hidden output matrix for the initialization phase is denoted as H_0, where $Rank(H_0) = L$. N_0 is the initial training data as a subset of D, where the length of $N_0 \geq L$. A special case of batch-ELM is when the length of N_0 is N. The steps and the processes in the initialization phase are similar to the ones described in the basic ELM algorithm. In order to initialize the learning, a small initial training set N_0 is used.

1. Initialization Phase (offline training):

   Step 1: Randomly assign input parameters W_i and b_i to each hidden node i.

   Step 2: Calculate the initial hidden layer output matrix H_0. Refer to (3.3), where N is replaced by N_0.

   Step 3: Estimate the initial output weights β^0. Again, the objective is to minimize $||H_0\beta^0 - Y_0||$ given that, Y_0 are the target values for N_0

   $$\beta^0 = (H_0^T H_0)^{-1} H_0^T Y_0. \tag{3.8}$$

   Step 4: Set $k = 0$ as the index of data blocks passed to the network.

2. Online Sequential Learning Phase (online training):

   During the online sequential learning phase, newly additional RSSI samples are collected and processed block by block. Each time a new block of data arrives, the OS-ELM is updated. The implementation of this phase is summarized in these steps.

   Step 1: Construct the partial hidden layer output matrix. Denote $G(x_j W_i, b_i) = g(W_i \cdot x_j + b_i)$, $(k+1)$ as the index of the block of new RSSI samples, and N_{k+1} as the number of samples in the $(k+1)$-th block

   $$H_{k+1} = \begin{bmatrix} G(x_1^{(k+1)}, W_1, b_1) & \dots & G(x_1^{(k+1)}, W_L, b_L) \\ \vdots & \dots & \vdots \\ G(x_N^{(k+1)}, W_1, b_1) & \dots & G(x_N^{(k+1)}, W_L, b_L) \end{bmatrix}. \tag{3.9}$$

   Step 2: Compute the output weight $\beta^{(k+1)}$ given the target values of $Y_{k+1} = \left[y_1^{(k+1)}, \dots, y_N^{(k+1)}\right]^T$

   $$\beta^{(k+1)} = \beta^{(k)} + P_{k+1} H_{k+1}^Y (Y_{k+1} - H_{k+1}\beta^k), \tag{3.10}$$

where $P_{k+1} = K_{k+1}^{-1}$. The term K_{k+1}^{-1} is introduced to avoid inverting the matrices. It is derived using the Woodbury formula,

$$P_{k+1} = P_k - P_k H_{k+1}^T \left(I + H_{k+1} P_k H_{k+1}^T \right) H_{k+1} P_k \tag{3.11}$$

Step 3: Update to next available block of data $k = k+1$ and repeat Step 1 in the online learning phase.

3.3 H-ELM

The Hierarchical based ELM (H-ELM) utilizes multiple layers for deep learning applications. The structure of the H-ELM training is divided into two subsystems, unsupervised training where the ELM sparse autoencoder is implemented, and supervised training where the original ELM is performed [37]. The main feature of the H-ELM is the unsupervised training phase that adopts the ELM sparse autoencoder. In general, the autoencoder is a neural network trained to learn a representation for a set of data [38]. The ELM-autoencoder consists of two parts, an encoder and a decoder. The encoder provides a compressed representation of the original input using the encoder activation function $g^{(1)}$,

$$y^{(1)} = g^{(1)} \left(W^{(1)} \cdot x + b^{(1)} \right), \tag{3.12}$$

where $W^{(1)}$ and $b^{(1)}$ are the encode weight matrix and encoder bias, respectively. A decoder produces a reconstructed input by mapping the encoded representation $y^{(1)}$ using a decoder activation function $g^{(2)}$,

$$z^{(2)} = g^{(2)}(W^{(2)} \cdot y^{(1)} + b^{(2)}), \tag{3.13}$$

where $W^{(2)}$ and $b^{(2)}$ are the decoding weight matrix and decoder bias, respectively. The objective of the ELM-autoencoder is to learn a function such that $h(x) \approx x$ to approximate the original input. Because that the ELM-autoencoder is trained following the ELM theory, once it is initialized, there is no need to fine-tune the network. In this specific application, the H-ELM uses the sparse ELM autoencoder to generate a reconstructed input. Sparse autoencoder acts as a feature selection mechanism that imposes constraints on the hidden neurons. In general, sparsity in the context of autoencoder reveals that many features in the input are redundant and do not need to be used [38]. Sparse ELM autoencoder performs an l_1 optimization to generate sparse and compact features of the input, resulting in a reduction of the number of the neural nodes. Due to the sparsity of l_1 optimization, an ELM sparse autoencoder learns the representation better, and its activation is more sparse, giving better performance and further improving the testing time of H-ELM [37]. The optimization model of the ELM sparse autoencoder is represented as,

$$E = ||H\beta - X||^2 + ||\beta||_{l_1}, \tag{3.14}$$

where X denotes the input data, H represents random mapping output, β is hidden layer weights, and $\beta||_{l_1}$ is the l_1 penalty term of the training model. In this case, that the optimization process is reduced to the optimization of l_1 using the FISTA (Fast Iterative Shrinkage-Threshold Algorithm) [39]:

Step 1: Calculate Lipschitz constant γ of the gradient of the first term $||H\beta - X||^2$ in Equation 3.14

Step 2: Initialize $y_1 = \beta_0$ and $t_1 = 1$. Then start the iteration for $k \geq 1$,

$$\beta_k = argmin\{\frac{\gamma}{2}||\beta - (\beta_{k-1} - \frac{1}{\gamma}\nabla(||H\beta_{k-1} - X||^2))||^2 + ||\beta||_{l_1}\} \tag{3.15}$$

$$t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2} \tag{3.16}$$

$$y_{k+1} = \beta_k + (\frac{t_{k-1}}{t_{k+1}})(\beta_k - \beta_{k-1}). \tag{3.17}$$

By completing the iterative steps, we achieve a compact representation of the original input. Then, the hierarchically encoded output from the unsupervised learning phase is randomly projected using the advantage of the ELM random features, where, the original ELM-based regression is performed for the supervised learning phase.

3.4 K-ELM

The Kernel based ELM (K-ELM) substitutes the inner product with a nonlinear kernel function, for processing in a high dimension space [40] [41]. The kernel computes the inner product in the feature space directly from the input data,

$$\Omega_{ELM_{i,j}} = h(x_i) \cdot h(x_j) = K(x_i, x_j), \tag{3.18}$$

where $h(.)$ is the hidden layer feature mapping, and (x_i, x_j) are the i-th and j-th input data point. In (3.18), there is no need to implement the transformation, since $h(.)$ does not need to be known. It is sufficient to only evaluate the nonlinear kernel function. The K-ELM algorithm is expressed in a single learning step as [42] [34],

$$f(x) = K(x)^T \left(\frac{I}{\lambda} + \Omega_{ELM} \right)^{-1} Y, \tag{3.19}$$

where λ is a tunable regularization parameter which helps stabilize the kernel matrix, $K(x)$ is a n vector of components $K(x) = K(x_i, x_j)$, and Ω_{ELM} is a n by n kernel matrix of

$$\Omega_{ELM} = HH^T = \begin{bmatrix} h(x_1)^T h(x_1) \dots h(x_1)^T h(x_n) \\ \vdots \\ h(x_n)^T h(x_1) \dots h(x_n)^T h(x_n) \end{bmatrix}. \tag{3.20}$$

The derivation of (3.19) involves the minimization of the regularized sum of squares error function

$$J(w) = \frac{1}{2} \sum_{n=1}^{N} \left(W^T h(x_n) - Y_n \right)^2 + \frac{\lambda}{2} W^T W. \tag{3.21}$$

Taking the partial derivative with respect to the learning weights W, and setting it to zero yields:

$$W = -\frac{1}{\lambda}\sum_{n=1}^{N}\left(W^T h(x_n) - Y_n\right)h(x_n) = \sum_{n=1}^{N} a_n h(x_n) = H^T a. \tag{3.22}$$

For simplification, a_n is defined as

$$a_n = -\frac{1}{\lambda}\sum_{n=1}^{N}\left(W^T h(x_n) - Y_n\right). \tag{3.23}$$

Substituting $W = H^T a$ into $J(w)$ yields

$$J(a) = Y^T Y - Y^T H H^T a - a^T H H^T Y + a^T H H^T H H^T a + \frac{\lambda}{2} a^T H H^T a, \tag{3.24}$$

where $J(w)$ is

$$J(w) = Y^T Y - Y^T H W - W^T H^T Y + W^T H^T H W + \frac{\lambda}{2} W^T W. \tag{3.25}$$

Under the Mercer's theorem, the kernel matrix is symmetric ($\Omega_{ELM} = \Omega_{ELM}^T$) and positive semi definite, where $\alpha^T \Omega_{ELM} \alpha \geq 0$ for any α vector. We obtain

$$J(a) = Y^T Y - Y^T \Omega_{ELM} a - a^T \Omega_{ELM} Y + a^T \Omega_{ELM}^2 a + \frac{\lambda}{2} a^T \Omega_{ELM} a. \tag{3.26}$$

. Minimizing the cost function $J(a)$ gives the optimal a as

$$a = \left(\Omega_{ELM} + \frac{I}{\lambda}\right)^{-1} Y. \tag{3.27}$$

Now, the following output function of the K-ELM is

$$f(x) = W^T h(x) = a^T H h(x) = K(x)^T \left(\frac{I}{\lambda} + \Omega_{ELM}\right)^{-1} Y. \tag{3.28}$$

By (3.28), we avoid using the feature vector $h(x)$ which requires us to use feature space of high dimensions. Again, when performing the K-ELM, the number of L and the hidden layer features mapping $h(.)$ does not need to be known.

CHAPTER 4

EXPERIMENTAL DESIGN AND SETUP

4.1 Experiment Setup

We conduct experimental studies in the Dynamical Networks and Control laboratory to evaluate the performance of the model-based approach and data-driven approach for indoor localization. Figure 4.1 shows a visual layout of the lab that serves as the experimental environment. As seen in Figure 4.1, the experiment setup involves designing a system of five different AP node locations (router locations) and multiple testing/training points. Initially, the RSSI signals get transmitted from various AP node (routers) locations to the mobile robot via multiple propagation paths. Each path interacts with different objects and surfaces in an indoor environment through the propagation effects (reflection, diffraction, scattering, and absorption). These effects dissipate the energy of the transmitted signals [43]. The locations of the AP nodes and the testing/training points are placed with reference to the corner closest to the lab door, and it is set as the origin in a standard X-Y plane. The RSSI fingerprints are collected at each data point (testing/training points), where the mobile robot scan the transmitted signals 100 times at five seconds intervals to build the database that will be used to provide an estimate for the testing locations. MATLAB R2017b is employed to carry out the simulation that will examine the performance of the suggested methods. The system characteristics where the simulations are completed are as follows: Intel(R) Core(TM) i7-9750H @ 2.60GHZ, with 16 GB RAM. The performance of the adapted localization techniques is measured by taking the distance error D_{error}. Equation 4.1 is calculated by taking the distance between the true locations (x, y)

and the estimated ones (x_0, y_0). Table 4.1 gives the exact locations of the AP nodes inside the DNC lab.

$$D_{error} = \sqrt{(x-x_0)^2 + (y-y_0)^2}. \tag{4.1}$$

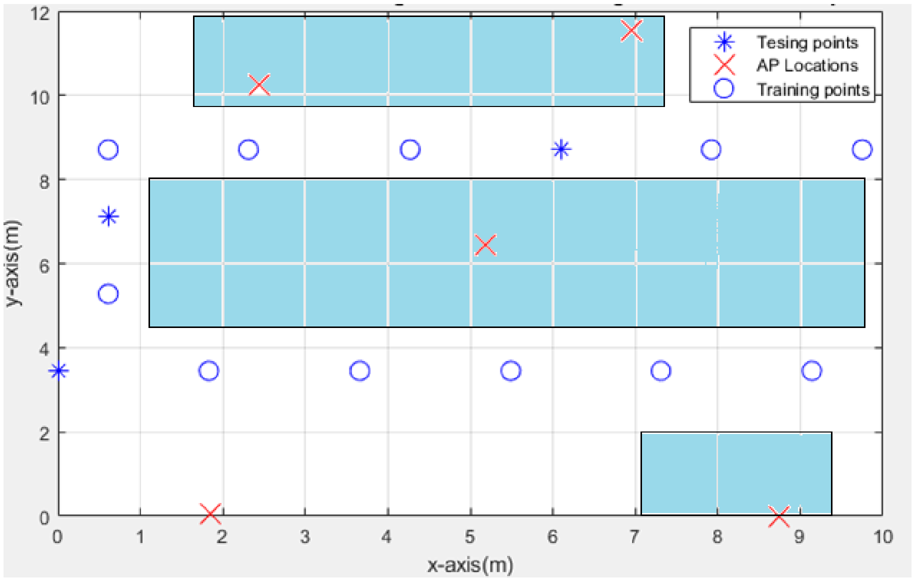

Figure 4.1. The layout of the DNC lab (light blue areas represent furniture and other physical objects).

Table 4.1. List and location of signal sources used inside the DNC lab

AP Nodes	Signal Sources	Sources Location (m)
AP1	Linksys098683	(1.85, 0.05)
AP2	Huawei	(5.18, 6.43)
AP3	Linksys29213	(6.96, 11.55)
AP4	DNC Lab	(8.74, 0)
AP5	TP Link	(2.44, 10.26)

4.2 Data Collection

We build a database to store the RSSI values. The RSSI fingerprinting technique is used since the RSSI samples can be measured from a mobile device. RSSI fingerprinting consists of two phases: offline training and online training. During the offline training phase, the RSSI measurements of each AP node are stored in a database along with the known coordinates of the mobile robot. During the online phase, the current RSSI location will be compared by an ELM matching algorithm to those stored in the database, and the closest match is returned as the estimated location. The RSSI fingerprinting approach is not robust to changes in the environment. Every time there is a change in the experimental setup (e.g., moving furniture, relocating the AP nodes,...etc.), a new database must be built to account for those changes. The figure 4.2 below illustrates the RSSI fingerprinting method.

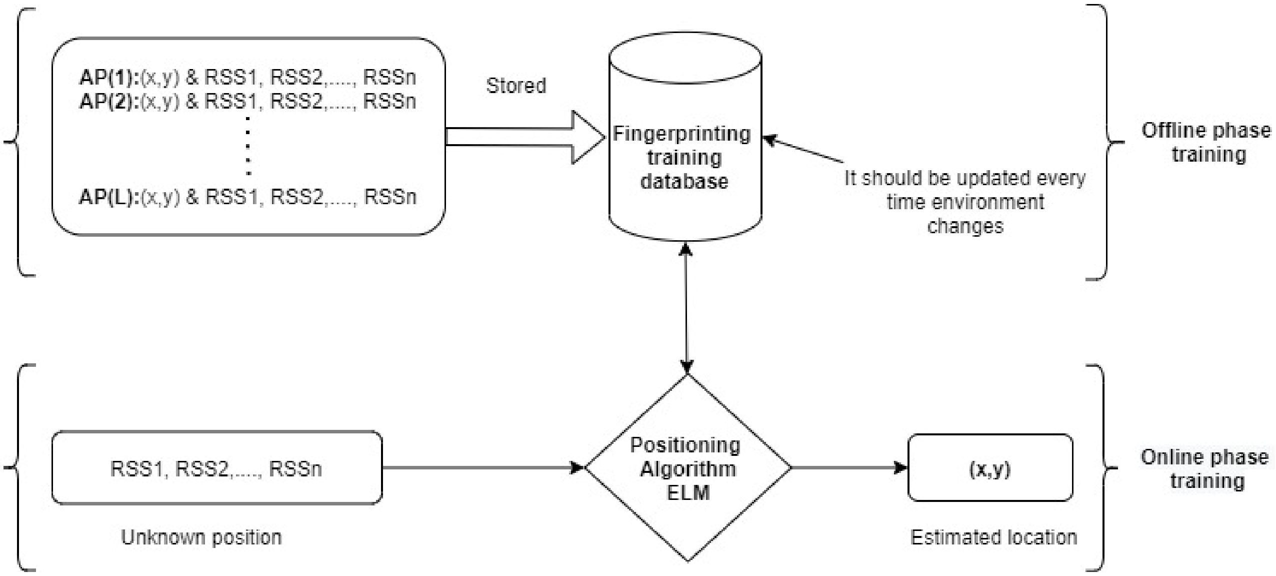
AP(1):(x,y) & RSS1, RSS2,...., RSSn
AP(2):(x,y) & RSS1, RSS2,...., RSSn
AP(L):(x,y) & RSS1, RSS2,...., RSSn
Stored
Fingerprinting training database
It should be updated every time environment changes
Offline phase training
RSS1, RSS2,...., RSSn
Unknown position
Positioning Algorithm ELM
(x,y)
Estimated location
Online phase training

Figure 4.2. RSSI fingerprinting method.

4.3 Data Quality and Filtering

After the data collection process is complete, the correlation coefficient $\rho_{(X,Y)}$ will be calculated to examine the quality of the collected data before the estimation procedures described in Section 2 and Section 3. $\rho_{(X,Y)}$ quantifies the relationship between two random variables X and Y [44]. X is the actual distance between the AP nodes and the testing points. Y is the average RSSI value. To determine $\rho_{(X,Y)}$, the RSSI measurements across all the AP nodes from all the testing points are considered, and the strength between X and Y is expressed with the best linear prediction.

$$\rho_{X,Y} = \frac{Cov[X,Y]}{\sigma_X \sigma_Y} = \frac{\sum((X-\bar{X})(Y-\bar{Y}))}{\sqrt{\sum((X-\bar{X})^2(Y-\bar{Y})^2)}}, \quad -1 \leq \rho_{X,Y} \leq 1, \tag{4.2}$$

where, $Cov[X,Y]$ is the covariance of X and Y, σ_x and σ_y are the standard deviation of X and Y, and $\bar{X}$, $\bar{Y}$ are the average values of X and Y. A ρ close to ± 1 represents a strong correlation between X and Y. A ρ close to 0 indicates a weak correlation between X and Y. As an example, the experiment below shows a correlation coefficient of -0.755. The ρ here is negative, indicating that the distance of transmission increases with a decrease in the strength of the transmitted signal as shown in Figure 4.3. ρ is relatively close to -1, indicating a good correlation between the distance and RSSI samples, and that the data quality of this experiment is acceptable for further analysis.

The RSSI signals can be very unstable and noisy due to the influence of the propagation effects. To control data quality, filtering techniques such as the Moving Average filter (MA) and Kalman Filter (KF) are applied to the raw data. This step is crucial to improve the performance of localization using model-based methods. The MA filter is an optimal solution to reduce the random white noise from the RSSI signals. The MA filter is simple

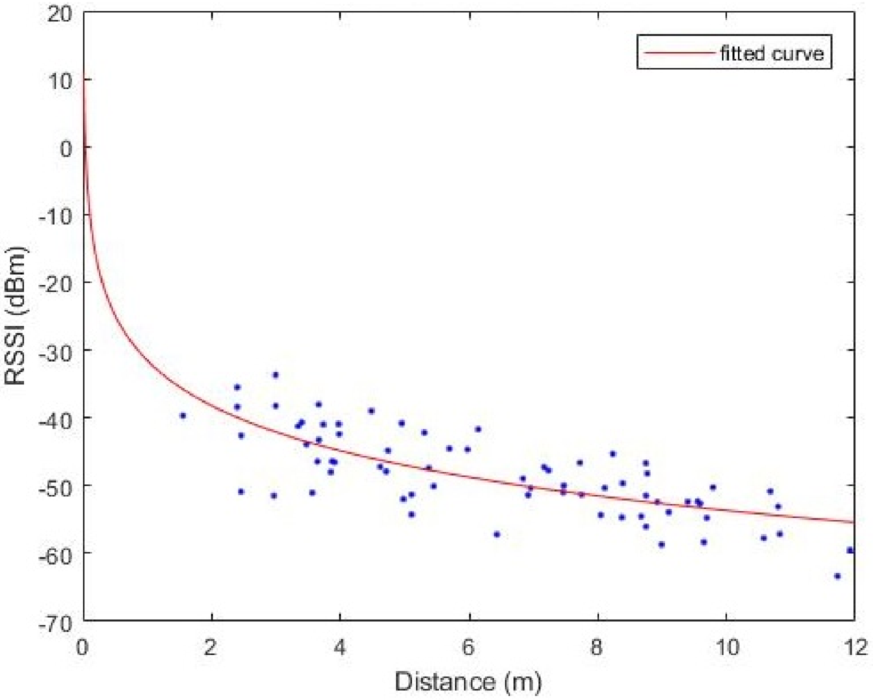

Figure 4.3. Plot of the RSSI Vs distance with a correlation coefficient of $\rho_{(X,Y)} = -0.755$.

to use; it operates by averaging the number of points from the RSSI signal to produce a filtered signal. Equation (4.3) summarizes the MA filter

$$y(i) = \frac{1}{M} \sum_{j=0}^{M-1} R(i+j), \tag{4.3}$$

where, y is the filtered output, M is the number of points used in moving average (moving window), and $R(i)$ is the i-th measurement of the RSSI input signal. The Kalman Filter (KF) is an optimal estimator to estimate the current state x_k of a system. It is implemented in two recursive steps, prediction and correction, to reduce the magnitude of the error covariance P_k. The prediction phase finds the state and error covariance in the next step. In the correction phase, we the state prediction is modified based on the current measurement Z_k [45]. To reduce the noise from the RSSI measurements, we choose to implement a linear scalar KF. The process noise covariance Q is set to a small value ($Q = 0.001$). The measurement noise covariance R is set to $R = 1$.

To start the KF estimation, we initialize the state and covariance x_{k-1} and P_{k-1}. k is the counter, and each time k is increment by 1. x_k and P_k denote the values after including measurement Z_k (posteriori). x_k^- and P_k^- denote the values before including measurement Z_k (priori). At the kth iteration, KF estimation is found as follows,

Prediction Phase:

$$x_k^- = x_{k-1}$$

$$P_k^- = P_{k-1} + Q$$

Correction Phase:

$$K_k = P_k^- \left(P_k^- + R\right)^{-1}$$

$$x_k = x_k^- + K_k(Z_k - x_k^-)$$

$$P_k = (1 - K_k)P_k^-$$

K_k is the Kalman gain, and the measurements Z_k are the actual RSSI values to be filtered out. The control input is omitted since it is zero in our system. Figure 4.4 displays the unstable nature of the RSSI measurements versus the smooth filtered RSSI samples.

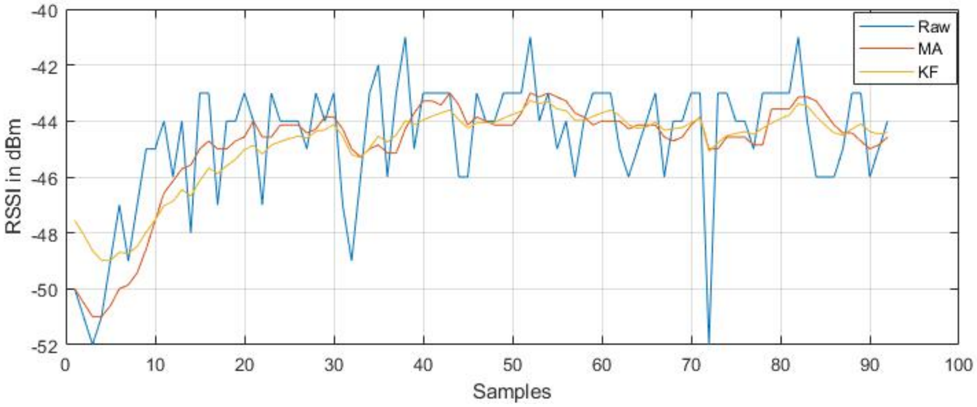

Figure 4.4. The filtered RSSI measurements from AP1.

CHAPTER 5

RESULTS

5.1 Results of the model based indoor localization

The localization errors between the actual and estimated positions using all the proposed model-based approaches are summarized in Table 5.1. A 5^{th}-degree polynomial is used in the implementation of the nonlinear regression model. The localization errors reported include the raw RSSI, the filtered (KF) RSSI, and the filtered (MA) RSSI data that are integrated using LLS and NLLS estimation techniques. We find that the NLLS estimation technique contributes to an increase in the overall performance in terms of localization error in all four models, especially in the free space models. Comparing the NLLS with the LLS method, the path loss and log-path loss models demonstrate a 29% increase in their localization performance, and the ITU model produces a 35.40% increase in its performance. When it comes to the nonlinear regression model, the NLLS shows a 12.80% improvement over the LLS method. Also, It is concluded that using the raw RSSI data in indoor propagation models results in a poor performance, as demonstrated by the large localization errors in the range of (2.94*m*, 3.69*m*). It is evident from the results shown in Table 5.1 that the filtering schemes (MA and KF) can improve the overall localization accuracy of the indoor propagation models. Also, we noticed an improvement of localization error when comparing the raw RSSI results of the log path loss model with the path loss model. We observed an improvement of 1.15*m* using the LLS method and 0.75*m* using the NLLS. This performance improvement is due to the variable added in Equation 2.3 that accounts for the multipath effect. On the other hand, the filtered results (KF and MA) show negligible differences in the range of (0.06*m*, 0.26*m*) when the path loss and log path loss models are compared. In summary, the nonlinear regression model produces the best performance with a localization error of $1.55m$ using the NLLS method. Among the indoor propagation models, the ITU model performs the best with a localization error of $2.00m$ (NLLS). Figure 5.3 shows the visual presentation of the estimated locations of the mobile robot to their corresponding true locations using the models of the overall best results.

Table 5.1. Localization error of the model based approaches in meters

	LLS			NLLS		
Proposed Method	Raw(m)	MA(m)	KF(m)	Raw(m)	MA(m)	KF(m)
Path Loss	5.37	4.26	4.75	3.69	2.92	3.62
Log Path Loss	4.22	4.18	4.60	2.94	2.86	3.36
ITU (N=30)	6.45	3.00	2.91	3.59	2.00	2.08
Nonlinear Regression	2.06	2.06	1.70	1.81	1.80	1.55

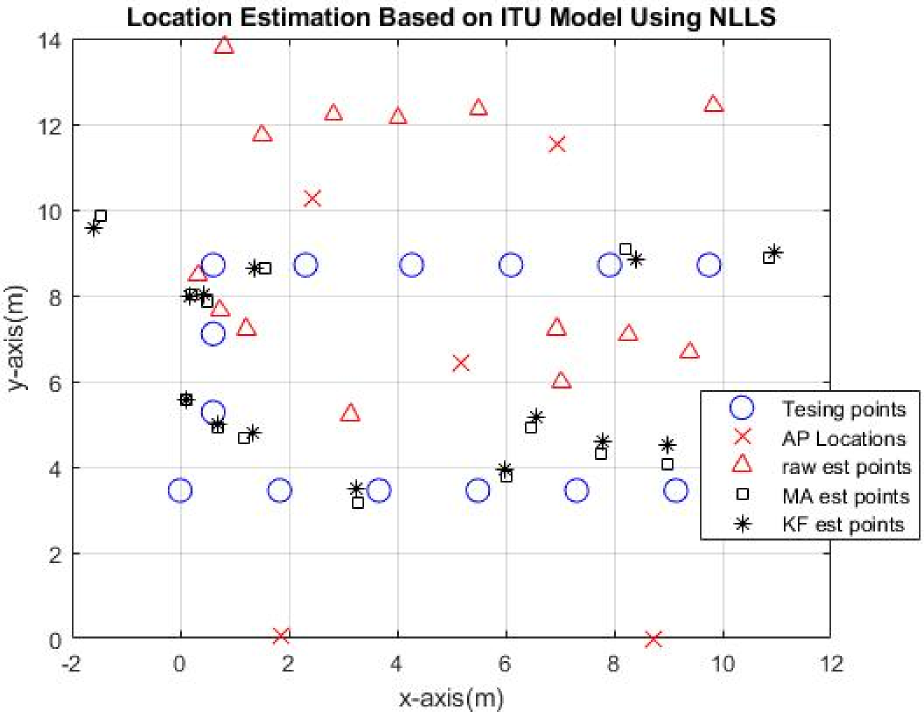

Figure 5.1. ITU-NLLS.

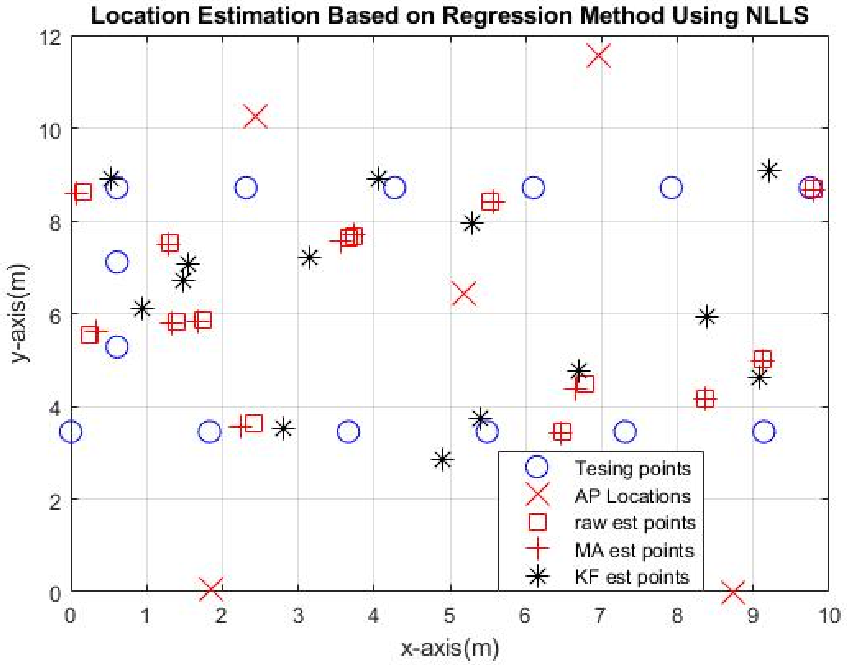

Figure 5.2. Regression-NLLS.

Figure 5.3. Plots of the estimated locations using ITU-NLLS and Nonlinear Regression-NLLS methods.

5.2 Results of the Machine Learning-based Indoor Localization:

As demonstrated in Table 5.2, the mean localization errors generated by Basic-ELM, OS-ELM, and H-ELM are 2.629*m*, 2.139*m*, and 2.568*m*, respectively. Generally, ELM algorithms can learn extremely fast, and this is apparent from the training time it takes to learn the network. The K-ELM provides the best localization accuracy of 1.498*m*. It shows an improvement in performance by 43.02%, 30%, and 41.67% when compared to the Basic-ELM, OS-ELM, and H-ELM. The four ELM algorithms are also compared in terms of he Cumulative Density Functions (CDFs) of the distance errors, as shown in Figure 5.4. In the CDF plots, K-ELM again shows the best performance, as it reaches probability of 1 at a smaller distance error value. Figures (5.5, 5.6, 5.7, 5.8) give a graphical display of the estimated locations using the four ELM algorithms. The Basic-ELM, OS-ELM, and H-ELM show higher variations giving the scattered nature of their estimated locations. K-ELM produces fewer variances and high certainty in estimating the testing points.

In this thesis, the indoor localization error of the data-driven approaches using multiple ELM algorithms and the model-based approaches are relatively similar, given the localization error between the nonlinear regression model (KF) using NLLS and the K-ELM where it shows a slight difference of 0.05*m*. In addition, the filtered ITU indoor localization error results using NLLS are approximately equal to the OS-ELM result with a small difference of 0.06*m* (KF).

Table 5.2. Performance comparison between the proposed ELM algorithms

Proposed Method	Training Time(s)	Testing Time(s)	Accuracy (m)
Basic-ELM	0.260	0.0036	2.629
OS-ELM	1.389	0.017	2.139
H-ELM	0.382	0.018	2.568
K-ELM	0.330	0.0072	1.498

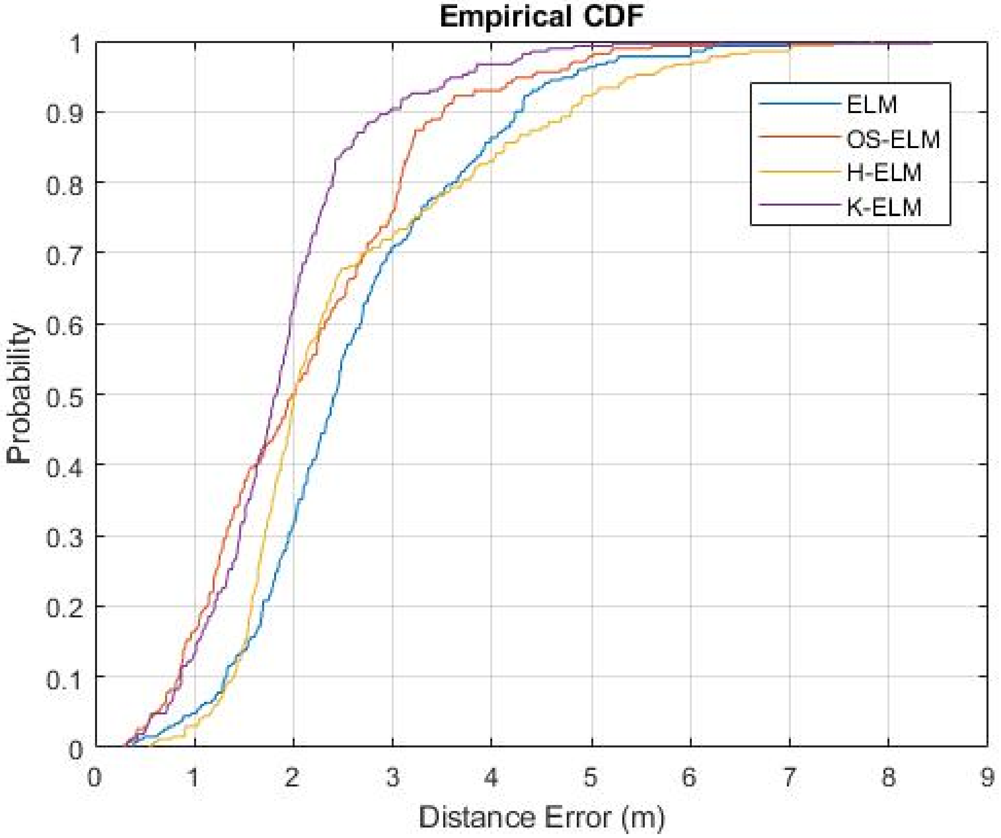

Figure 5.4. Localization accuracy CDFs across multiple ELM algorithms.

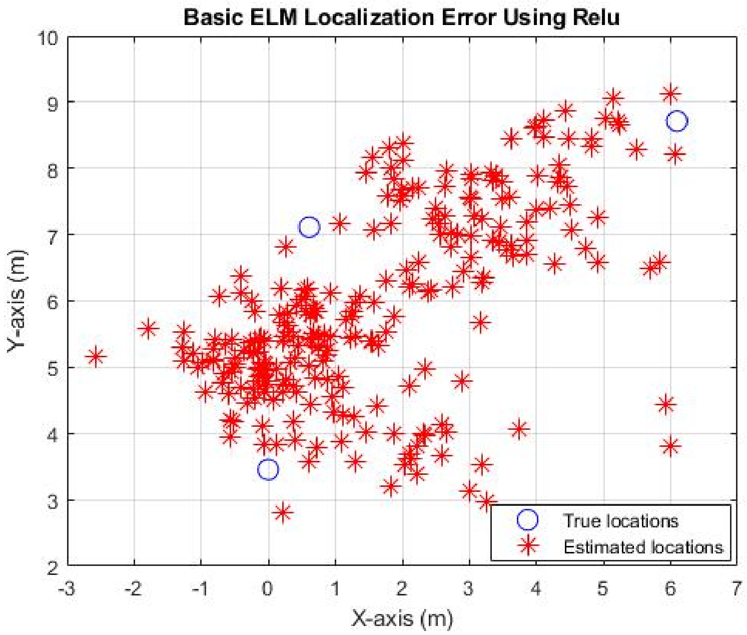

Figure 5.5. Estimated locations using ELM.

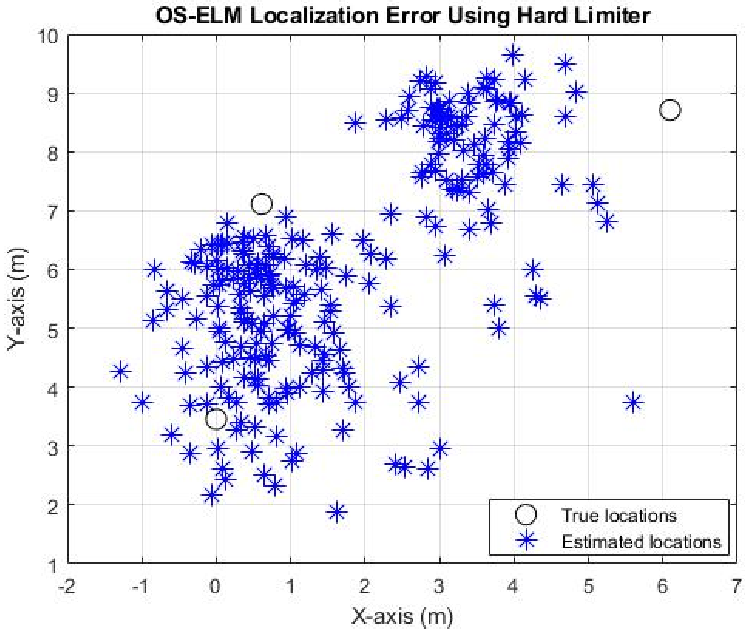

Figure 5.6. Estimated locations using OS-ELM.

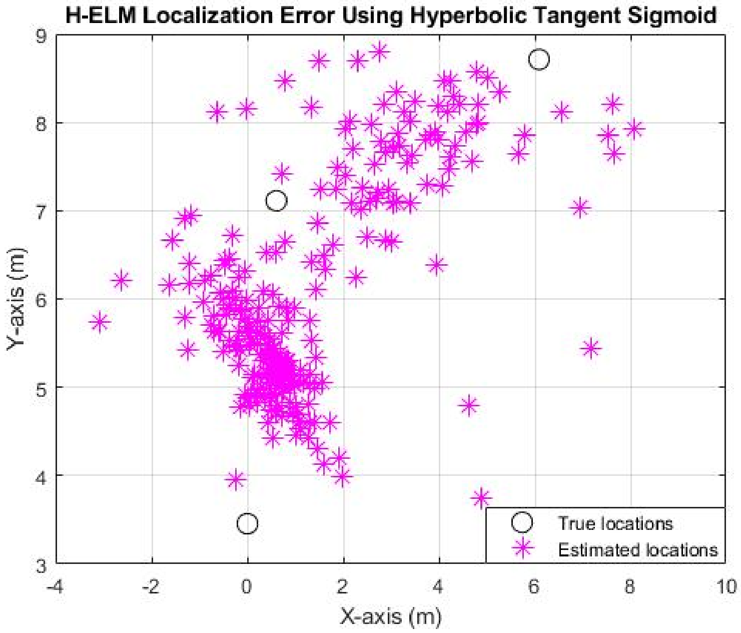

Figure 5.7. Estimated locations using H-ELM.

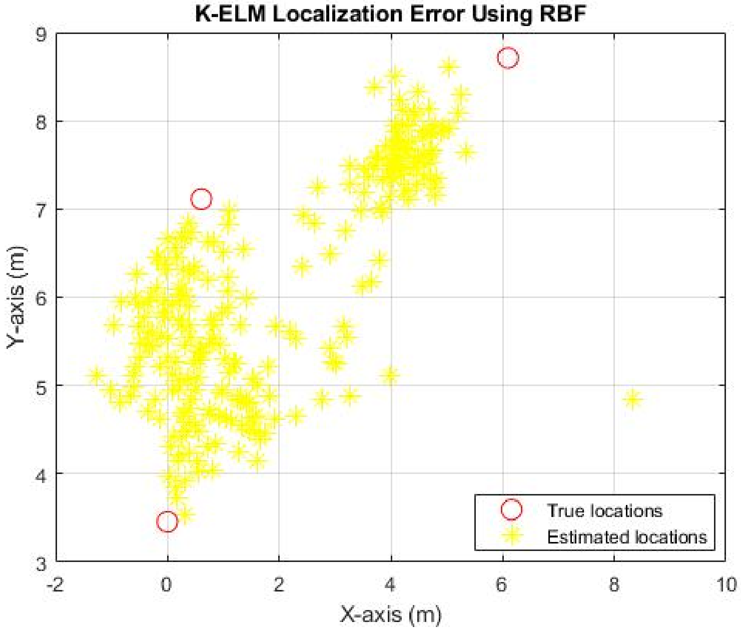

Figure 5.8. Estimated locations using K-ELM.

CHAPTER 6

CONCLUSION AND FUTURE WORK

This paper analyzed the indoor localization problem using both model-based and machine-learning methods including multiple ELM algorithms. For the model-based approaches, multiple steps are involved, including filtering, parameters estimation, distance calculation, and position estimation. On the other hand, the main focus of ELM algorithms is toward the processing of data. The model-based approaches are closely dependent on the estimation of parameters, which must be updated each time the environment changes. ELM algorithms remove the need of using filtering schemes, where are important steps in indoor propagation models-based location to improve the localization accuracy. The data collection and the quality of data must be examined first before analyzing the performance of the proposed methods. As concluded in this paper, the two approaches produce approximately similar results when we closely compared their indoor localization accuracies.

There are several directions of the future work. First, the geometric layout and height to place the AP nodes and the number of AP nodes used for signal coverage area inside the lab can be optimized. Second, with the understanding of WiFi localization performance, the fusion of data from multiple sensor sources can be used to improve the indoor localization accuracy [10], Furthermore, the performance of localization can be improved by investigating other adaptive filtering and estimation techniques in the model-based methods and expanding the number of training and testing samples in data-driven approaches.

REFERENCES

[1] T. Menard, J. Miller, M. Nowak, and D. Norris, “Comparing the gps capabilities of the samsung galaxy s, motorola droid x, and the apple iphone for vehicle tracking using freesimmobile,” *Conference Record - IEEE Conference on Intelligent Transportation Systems*, 2011.

[2] A. Karaagac, J. Haxhibeqiri, M. Ridolfi, W. Joseph, I. Moerman, and J. Hoebeke, “Evaluation of accurate indoor localization systems in industrial environments,” in *2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 2017, pp. 1–8.

[3] S. Woo, S. Jeong, E. Mok, L. Xia, C. Choi, M. Pyeon, and J. Heo, “Application of wifi-based indoor positioning system for labor tracking at construction sites: A case study in guangzhou mtr,” *Automation in Construction*, vol. 20, pp. 3–13, 01 2011.

[4] T. V. Haute, E. D. Poorter, P. Crombez, F. Lemic, V. Handziski, N. Wirström, A. Wolisz, T. Voigt, and I. Moerman, “Performance analysis of multiple indoor positioning systems in a healthcare environment,” *International Journal of Health Geographics*, vol. 15, 2016.

[5] H. Obeidat, W. Shuaieb, O. Obeidat, and R. Abd-Alhameed, “A review of indoor localization techniques and wireless technologies,” *Wireless Personal Communications*, vol. 119, 07 2021.

[6] G. Fusco and J. M. Coughlan, “Indoor localization using computer vision and visual-inertial odometry,” in *Computers Helping People with Special Needs*, K. Miesenberger and G. Kouroupetroglou, Eds. Cham: Springer International Publishing, 2018, pp. 86–93.

[7] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," in *in Proceedings of the 9th international conference on Mobile systems, applications, and services*. Maryland, USA: ACM, 2011, pp. 141–154.

[8] C.-C. Hsiao and P. Huang, "Two practical considerations of beacon deployment for ultrasound-based indoor localization systems," in *2008 IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (sutc 2008)*, 2008, pp. 306–311.

[9] C. Teulire, E. Marchand, and L. Eck, "3-d model-based tracking for uav indoor localization," *IEEE Transactions on Cybernetics*, vol. 45, no. 5, pp. 869–879, 2015.

[10] J. Shang, X. Hu, F. Gu, D. Wang, and S. Yu, "Improvement schemes for indoor mobile location estimation: A survey," *Mathematical Problems in Engineering*, vol. 2015, pp. 1–32, 2015.

[11] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, 2019.

[12] T. Kim Geok, K. Zar Aung, M. Sandar Aung, M. Thu Soe, A. Abdaziz, C. Pao Liew, F. Hossain, C. P. Tso, and W. H. Yong, "Review of indoor positioning: Radio wave technology," *Applied Sciences*, vol. 11, 2021.

[13] P. K. Yoon, S. Zihajehzadeh, B.-S. Kang, and E. J. Park, "Robust biomechanical model-based 3-d indoor localization and tracking method using uwb and imu," *IEEE Sensors Journal*, vol. 17, no. 4, pp. 1084–1096, 2017.

[14] M. Youssef, A. Agrawala, and A. Udaya Shankar, "Wlan location determination via clustering and probability distributions," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, 2003, pp. 143–150.

[15] J.-L. Rullán-Lara, S. Salazar, and R. Lozano, "Real-time localization of an uav us-

ing kalman filter and a wireless sensor network," *Journal of Intelligent & Robotic Systems*, vol. 65, no. 1-4, pp. 283–293, 2012.

[16] X. Song, X. Fan, C. Xiang, Q. Ye, L. Liu, Z. Wang, X. He, N. Yang, and G. Fang, "A novel convolutional neural network based indoor localization framework with wifi fingerprinting," *IEEE Access*, vol. 7, 2019.

[17] Y. Shu, Y. Huang, J. Zhang, P. Cou, P. Cheng, J. Chen, and K. G. Shin, "Gradient-based fingerprinting for indoor localization and tracking," *IEEE Transactions on Industrial Electronics*, vol. 63, 2016.

[18] C. Laoudias, D. Eliades, P. Kemppi, C. Panayiotou, and M. Polycarpou, "Indoor localization using neural networks with location fingerprints," in *ICANN*, 2009.

[19] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate rssi indoor localization," *IEEE Internet of Things Journal*, vol. 6, 2019.

[20] M. I. AlHajri, N. T. Ali, and R. M. Shubair, "Indoor localization for iot using adaptive feature selection: A cascaded machine learning approach," *IEEE Antennas and Wireless Propagation Letters*, vol. 18, no. 11, pp. 2306–2310, 2019.

[21] K. S. Kim, S. Lee, and K. Huang, "A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on wi-fi fingerprinting," *Big Data Analytics*, vol. 3, 04 2018.

[22] G.-B. Huang, Q.-Y. Zhu, and C.-K. Siew, "Extreme learning machine: theory and applications," *Neurocomputing*, vol. 70, no. 1-3, pp. 489–501, 2006.

[23] J. Kim, J. Kim, G.-J. Jang, and M. Lee, "Fast learning method for convolutional neural networks using extreme learning machine and its application to lane detection," *Neural Networks*, vol. 87, pp. 109–121, 2017.

[24] C. Wan, Z. Xu, P. Pinson, Z. Y. Dong, and K. P. Wong, "Probabilistic forecasting of wind power generation using extreme learning machine," *IEEE Transactions on*

Power Systems, vol. 29, 2014.

[25] S. Naji, A. Keivani, S. Shamshirband, U. J. Alengaram, M. Z. Jumaat, Z. Mansor, and M. Lee, "Estimating building energy consumption using extreme learning machine method," *Energy*, vol. 97, pp. 506–516, 2016.

[26] Z.-L. Sun, T.-M. Choi, K.-F. Au, and Y. Yu, "Sales forecasting using extreme learning machine with applications in fashion retailing," *Decision Support Systems*, vol. 46, no. 1, pp. 411–419, 2008.

[27] H. Zou, X. Lu, H. Jiang, and L. Xie, "A fast and precise indoor localization algorithm based on an online sequential extreme learning machine," *Sensors*, vol. 15, no. 1, pp. 1804–1824, 2015.

[28] S. Sadowski and P. Spachos, "Rssi-based indoor localization with the internet of things," *IEEE Access*, vol. 6, pp. 30 149–30 161, 2018.

[29] T. S. Rappaport *et al.*, *Wireless communications: principles and practice*. prentice hall PTR New Jersey, 1996, vol. 2.

[30] H. K. Rath, S. Timmadasari, B. Panigrahi, and A. Simha, "Realistic indoor path loss modeling for regular wifi operations in india," in *2017 Twenty-third National Conference on Communications (NCC)*, 2017, pp. 1–6.

[31] P. Series, "Propagation data and prediction methods for the planning of indoor radiocommunication systems and radio local area networks in the frequency range 900 mhz to 100 ghz," *Recommendation ITU-R*, pp. 1238–7, 2012.

[32] J. Yang, Y. Chen, R. P. Martin, W. Trappe, and M. Gruteser, *On the Performance of Wireless Indoor Localization Using Received Signal Strength*. John Wiley and Sons, Ltd, 2018.

[33] R. Zekavat and R. M. Buehrer, *Impact of Anchor Placement and Anchor Selection on Localization Accuracy*, 2019, pp. 435–465.

[34] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and*

Statistics). Berlin, Heidelberg: Springer-Verlag, 2006.

[35] M. Rigotti, O. Barak, M. R. Warden, X. J. Wang, N. D. Daw, E. K. Miller, and S. Fusi, "The importance of mixed selectivity in complex cognitive tasks," *Nature*, pp. 585–590, 2013.

[36] S. Shubathra, P. Kalaivaani, and S. Santhoshkumar, "Clothing image recognition based on multiple features using deep neural networks," pp. 166–172, 2020.

[37] J. Tang, C. Deng, and G.-B. Huang, "Extreme learning machine for multilayer perceptron," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 4, pp. 809–821, 2016.

[38] A. C. Ian Goodfellow, Yoshua Bengio, *Deep Learning*, 1st ed. The MIT Press, 2015.

[39] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm with application to wavelet-based image deblurring," in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2009, pp. 693–696.

[40] G. Huang, H. Zhou, X. Ding, and R. Zhang, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.

[41] ——, "Extreme learning machine for regression and multiclass classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 42, no. 2, pp. 513–529, 2012.

[42] G. Huang, D. Wang, and Y. Lan, "Extreme learning machines: a survey," *International Journal of Machine Learning and Cybernetics*, vol. 2, pp. 107–122, 2011.

[43] D. M. Dobkin, *RF Engineering for Wireless Networks: Hardware, Antennas and Propagation*. Elsevier, 2011.

[44] D. J. G. Yates, Roy D, *Probability and Stochastic Processes: A Friendly Introduction for Electrical and Computer Engineers*, 2nd ed. Wiley, 2005.

[45] D. I. Wilson, *Advanced Control using MATLAB*. Auckland University of Technology

New Zealand, 2015.