UNIVERSITI PUTRA MALAYSIA

APPLICATION OF MULTILAYER PERCEPTRON AND RADIAL BASIS FUNCTION NETWORKS, TOGETHER WITH VARIABLE REGION OF INTEREST GEOMETRY

AMIN SOLEHUDDIN BIN ADLI

AN IMPROVED INDOOR LOCATION SYSTEM BY HYBRID APPLICATION OF MULTILAYER PERCEPTRON AND RADIAL BASIS FUNCTION NETWORKS, TOGETHER WITH VARIABLE REGION OF INTEREST GEOMETRY

By

AMIN SOLEHUDDIN BIN ADLI

Thesis Submitted to the Department of Physics, Universiti Putra Malaysia, in partial Fulfilment of the Requirements for the Degree of Bachelor of Science in Instrumentation Science with Honours
February 2022

DEDICATION

This book is dedicated to my supervisor, beloved family, and friends for your kindness, devotion, and endless support. Thank you for all.

ABSTRACT

AN IMPROVED INDOOR LOCATION SYSTEM BY HYBRID APPLICATION OF MULTILAYER PERCEPTRON AND RADIAL BASIS FUNCTION NETWORKS, TOGETHER WITH VARIABLE REGION OF INTEREST GEOMETRY

By

AMIN SOLEHUDDIN BIN ADLI

February 2022

Supervisor: Dr. Edgar Scavino

Faculty: Faculty of Science

The development of precise wireless localization algorithms, a critical enabler technology for future Location-Based Services (LBS), is currently generating much buzz. As interest in universal computing and location-aware techniques develops, so does the necessity for a viable and precise solution to localization. In various location-based services (LBS) applications, GPS designs are standard. Due to the weak GPS signals from satellites that cannot penetrate buildings and structures or through the earth, GPS performance in urban canyons is worse than in open areas, mainly in-house and indoor, as well as underground circumstances or surroundings. As a result, GPS is no longer valid for interior navigation. Due to this issue, Indoor Positioning Systems (IPS) has been developed. Due to impediments in the indoor environment, such as walls

and partitions. Positional issues are exacerbated by these irregularities, which occur naturally in any indoor space. IPS encompasses a wide range of techniques. This method, however, generates a large amount of RSS sample data. The problem with using a big data sample is that the data is highly nonlinear. Artificial Neural Networks (ANNs) can help solve these problems. Hence, in this study, the work focuses on reducing the error for indoor location systems using an artificial neural network. A total of 576 data were collected for this experiment. This study used a radial basis function network to process the data. The software used in this study is MATLAB. The data consist of 96 points and six signal strengths. Each of the 96 points has six signal strengths.

ABSTRAK

SISTEM LOKASI DALAMAN YANG DIPERBAIKI OLEH APLIKASI HIBRID PERCEPTRON BERBILANG LAPISAN DAN RANGKAIAN FUNGSI ASAS JARI, BERSAMA-SAMA DENGAN GEOMETRI KEPENTINGAN WILAYAH PEMBOLEH UBAH

Oleh

AMIN SOLEHUDDIN BIN ADLI

Februari 2022

Penyelia : Dr Edgar Scavino

Fakulti : Fakulti Sains

Pembangunan algoritma navigasi nirwayar yang tepat, adalah teknologi pemboleh yang kritikal untuk Perkhidmatan Berasaskan Lokasi (LBS) pada masa hadapan, sedang menjana banyak perhatian. Semakin perhatian dalam pengkomputeran universal dan teknik menyedari lokasi berkembang, begitu juga keperluan untuk penyelesaian yang berdaya maju dan tepat untuk navigasi dalam bangunan. Dalam pelbagai aplikasi perkhidmatan berasaskan lokasi (LBS), reka bentuk GPS biasanya digunakan. Disebabkan oleh isyarat GPS yang lemah daripada satelit yang tidak dapat menembusi bangunan dan struktur atau melalui bumi, prestasi GPS di bandar adalah lebih teruk daripada di kawasan pedalaman, terutamanya dalam rumah dan dalam bangunan , serta keadaan bawah tanah atau sekeliling. Akibatnya, GPS tidak lagi berguna untuk navigasi dalam bangunan. Untuk mengatasi isu ini, Sistem Kedudukan Dalaman (IPS) telah dibangunkan. Disebabkan halangan dalam persekitaran

dalaman, seperti dinding dan sekatan, penyetempatan nirwayar terdedah kepada pengurangan prestasi. Isu kedudukan diburukkan lagi oleh ketidakaturan yang disebabkan isu isu tersebut yang manaberlaku secara semula jadi di mana-mana ruang dalaman. IPS merangkumi pelbagai teknik. Kaedah ini walaubagaimanapun, menjana sejumlah besar data yang besar. Masalah dengan menggunakan sampel data besar ialah data itu sangat tidak linear. Rangkaian Neural Buatan (ANN) boleh membantu menyelesaikan masalah ini. Oleh itu, dalam kajian ini, kerja-kerja memfokuskan kepada mengurangkan ralat untuk sistem lokasi dalaman menggunakan rangkaian AI. Dalam kajian ini rangkaian fungsi asas jejari digunakan untuk memproses data. Perisian yang digunakan dalam kajian ini ialah MATLAB. Sebanyak 576 data telah dikumpul untuk eksperimen ini. Data terdiri daripada 96 titik dan 6 kekuatan isyarat. Setiap daripada 96 titik mempunyai 6 kekuatan isyarat.

ACKNOWLEDGEMENT

All praise to Allah the Almighty who have given me the strength, guidance, and blessing more than I deserved.

I want to thank my beloved family that has always supported me throughout working on this project. I also want to express my deepest gratitude to my supervisor, Dr. Edgar Scavino, who guide me in completing this project. Thank you for always making time to assist me, and thank you for all the moral support and guidance for me during the planning and development of this project.

Besides, I am also thankful to my friends for helping me throughout this project. Last but not least, thank you to those who took part in this project.

DECLARATION

Declaration by student

I hereby confirm that:

- This thesis is my original work;
- Quotations, illustrations, and citations have been duly referenced;
- This thesis has not been submitted previously or concurrently for any other degree at any other institution;
- Intellectual property from the thesis and copyright of thesis are fully-owned by Universiti Putra Malaysia, as according to the Universiti Putra Malaysia (Research) Rules 2012;
- Written permission must be obtained from the supervisor and the office of Deputy Vice-Chancellor (Research and Innovation) before the thesis is published (in the form of written, printed, or electronic form), including books, journals, modules, proceedings, popular writings, seminar papers, manuscripts, posters, reports, lecture notes, learning modules or any other materials as stated in the Universiti Putra Malaysia (Research) Rules 2012;
- There is no plagiarism or data falsification/fabrication in the thesis, and scholarly integrity is upheld as according to the Universiti Putra Malaysia (Research) Rules 2012.

Signature: ________________________ Date: ___________________

Name and Matric No.: Amin Solehuddin Bin Adli

TABLE OF CONTENT

LIST OF FIGURES

LIST OF ABBREVIATION

RBF	Radial Basis Function Network

CHAPTER 1

INTRODUCTION

1.1 Background

Nowadays, the pursuit of precise wireless localization algorithms, a vital enabler technology for future Location-Based Services (LBS), has garnered considerable interest. The need for a practical and precise solution to localization is growing as interest in universal computing and location-aware approaches grow. GPS designs are widely used in a range of location-based services (LBS) applications. However, GPS performance in urban canyons is lower than in open areas, mainly in-house and indoor, and underground situations or surrounds, due to the weak GPS signals from satellites that cannot penetrate houses and structures or through the earth. As a result, GPS is made obsolete for indoor localization. By merging two or more wireless technologies, Indoor Positioning Systems (IPS) are proposed to avoid this limitation. Wireless localization is nevertheless susceptible to fading due to barriers in the internal environment, such as walls and partitions. These anomalies, which occur naturally in any interior environment, contribute significantly to positional difficulties.

IPS includes a plethora of approaches. However, this method generates a considerable amount of RSS sample data. The issue with a big data sample is that the data in the sample is highly nonlinear. These challenges can be addressed by utilizing Artificial Neural Networks (ANNs) such as the Multilayer Perceptron (MLP) and Radial Basis Function Network (RBF). This is demonstrated in (Ullah Khan et al., 2020), in which the author employs an Artificial Neural Network (ANN) to solve the nonlinear spatial distribution of signal intensities.

MLPs are a sort of feed-forward artificial neural network (ANN). MLP is vague; it might refer to any feed-forward ANN or networks comprising many perceptron layers (with threshold activation). Multilayer perceptrons are frequently jokingly referred to as "vanilla" neural networks, mainly when only one hidden layer is present. An MLP comprises at least three layers of nodes: an input layer, a hidden layer, and an output layer. Each node represents a neuron with a nonlinear activation function except for the input nodes. MLPs are trained by a supervised learning technique known as backpropagation. MLP differs from a linear perceptron in that it has multiple layers and nonlinear activation, and it is capable of separating non-linearly separable data.

The term "multilayer perceptron" refers to a collection of multilayer perceptrons, not to a single multilayer perceptron. Rather than that, it is made up of numerous layers of perceptrons. Another idea is a "multilayer perceptron network." Furthermore, MLP "perceptrons" are not actual perceptrons in the strictest sense of the term. True perceptrons are a subclass of artificial neurons that use a threshold activation function such as the Heaviside step as their activation function. Any function can activate MLP perceptrons. While a simple perceptron does binary classification, an MLP neuron can perform either classification or regression depending on its activation function. Later on, the term "multilayer perceptron" was used without respect for the nodes/layers, which might be formed of arbitrarily defined artificial neurons rather than perceptrons. This interpretation avoids using the term "perceptron" in a broader sense to encompass any artificial neuron.

A radial basis function network is a neural network that approaches design as a curve fitting (approximation) problem in a high-dimensional environment. Learning is comparable to determining a multidimensional function that fits the training data the best, with the criterion for "best fit" being quantified statistically, as discussed in (Jefferson et al., 1995). Similarly, the author notes that regularisation is comparable to interpolating the test data using this multidimensional surface. This is the actual reason behind the RBF approach, as it is based on research on tight interpolations in a multidimensional space. The hidden units in a neural network form a collection of "functions" that serve as a random "basis" for the input patterns (vectors). These are referred to as radial basis functions. Radial basis functions were introduced in (Powell, M. J. D. 1977) to answer the problem of real multivariate interpolation. This is a significant area of research in numerical analysis at the moment. In its simplest form, an RBFN is composed of three distinct layers. The input layer consists of source nodes (sensory units). The second layer is a high-dimensional concealed layer. Finally, the output layer demonstrates the network's response to the activation patterns given to the input layer—the nonlinear transition from the input space to the hidden-unit space.

On the other hand, as discussed in (Jefferson et al., 1995), the transfer from the hidden space to the output space is linear. This is mathematically justified in (Cover, T. M. 1965), where Cover asserts that a pattern classification problem cast in a nonlinear high-dimensional space is more likely to be linearly separable than one cast in a low-dimensional space. This is referred to as

Cover's Theorem on Pattern Separability. This is also why the dimension of the hidden-unit space in an RBFN is large.

1.2 Problem statement

According to a study, ANNs can solve problems when confronted with significant data. This entails tens of thousands, if not more, of cases. The author (Ullah Khan et al., 2020) employs ANN to analyze sample data with a highly nonlinear spatial distribution of signal intensities. The researcher employs fingerprinting techniques in both WLAN and WSN environments. However, this method generates a considerable amount of RSS sample data. The ANN is employed because it classifies using nonlinear discriminant functions. ANN approaches have been applied to positioning applications, most notably in conjunction with fingerprinting techniques. Additionally, the author explains that when location-tagged calibration data from both WLAN and WSN are used, the concept of a model and associated hyperparameters lead. By utilizing continually available signal strength data, this trained architecture is turned into a mathematical model used to solve the location problem.

Indoor GPS positioning was previously used. However, it is incapable of performing well. This is demonstrated in (Tsvi Kulfik et al., 2011), which concludes that indoor GPS location faces two substantial difficulties. The first objective is to detect weak signals; the second objective is to improve the quality of measurements, which is harmed by multipath and signal bending.

Artificial Neural Networks have been trained on indoor locating systems and can deal with large data sets. Thus, the demonstration established that in (Ullah Khan et al., 2020), the author employs ANN to analyze sample data with a highly nonlinear spatial distribution of signal intensities. The author can anticipate the tile placement using Multilayer Perceptron (MLP) based on the hybrid signal strength. Thus, utilizing an Artificial Neural Network to assist in developing the Indoor Location System is appropriate.

1.3 Objective

1.3.1 General Objective

The general objective is to produce an improved indoor location system by hybrid application of Multilayer Perceptron and Radial Basis Function networks, together with the variable region of interest geometry.

1.3.2 Specific objective

The specific objectives are to:

1. To create a set of RBF per each signal
2. To evaluate the average error between RBF and signal's actual values among the grid position
3. To evaluate the global error

1.4 Scope of study

RBF can help analyze the large set of data gathered during the indoor location system in the research. The performance RBF was observed. To analyze the sample data, knowledge of machine learning is required. The procedure and preparation are essential to ensure that a program or algorithm can run when analyzing sample data. The performance metrics analyzed are average errors between collected data and RBF. These data were collected at the Faculty of Engineering and Built Environment basement at the National University of Malaysia (UKM), Malaysia.

1.5 Significant of study

The research study helps to improve indoor location systems by using a hybrid application of MLP and RBF. The algorithm for RBF can represent based on a graph. From that study, the MATLAB algorithm system based on MLP and RBF can help improve the indoor location system.

1.6 Thesis outline

This research study was conducted to evaluate MLP and RBF analyze the data from indoor location systems. Altogether, the thesis consists of six chapters where the first chapter covers the background, statement of problems, objectives, scopes, and significance of the study.

Chapter 2 lists the literature review based on the finding of the previous research studies about indoor location systems. This chapter explains the theory that was related to this field of study. The theories related to this field of study include the indoor location system, Multilayer perceptron, and radial basis function network. The instrument used, data collection, and method analysis for this research project are stated in Chapter 3. This study is conducted by using software known as MATLAB.

Result and discussion are elaborated in Chapter 4, while Chapter 5 mainly emphasizes the limitation and conclusion of this study for future reference.

CHAPTER 2

LITERATURE REVIEW

2.1 Introduction

This chapter covers the theory related to the field of study, which is the indoor location system, Multilayer perceptron, and radial basis function network.

2.2 Indoor Location System

2.2.1 History of Indoor location System

An indoor location system (ILS) is a system that delivers an accurate position within a closed space, similar to how a GPS works. On the other hand, GPS cannot operate effectively within a building due to its reliance on GPS satellites. (2013) (Farid et al.) GPS is inefficient in urban canyons, close to walls, buildings, and trees, indoors, and in underground locations, as the signal from the GPS satellites is too weak to penetrate most buildings, rendering GPS worthless for indoor localization.

As urbanization forces people to spend more time indoors, it becomes vital to identify and lead humans through complex structures such as malls. Additionally, accurate user location information benefits the government and industry by allowing more accurate information dissemination. As a result, an indoor-based localization system that utilizes both ambient magnetic field and Wi-Fi as fingerprints is presented in (Zhang et al., 2014), intelligently merging two fingerprinting methodologies that complement one another. Due to the small size of the fingerprint database, this technology enables scaling while maintaining a high level of positioning accuracy compared to other state-of-the-art systems. Hybrid approaches improve localization accuracy while decreasing the time required to acquire indoor datasets. The author refers to their system as MaWi, an indoor localization technology based on a smartphone and utilizes magnetic fields and Wi-Fi as fingerprints. The two fingerprints are used in a complementary manner as a "duet." MaWi provides scalability through a clever combination of low demand on the fingerprint database and extremely competitive localization accuracy compared to state-of-the-art methods. Additionally, MaWi is a low-cost option because it does not require dedicated hardware or infrastructure adaption. Finally, because MaWi uses a smartphone as both the survey instrument and the location client, it is accessible to widespread adoption.

GPS is a widely used location system. However, due to GPS signal constraints, this level of performance is only possible outdoors. As a result, a hybrid indoor localization and navigation (HILN) system for pedestrians has been developed. Which includes a component for map matching, was proposed in (Tian et al., 2015). According to the author, this system combines pedestrian dead reckoning with a low-range proximity access control approach, such as RFID, that includes adaptive drift compensation at exchange control points using a particle-filter map-matching algorithm. This technology combines the advantages of both radiofrequency and inertial navigation systems.

Additionally, the author indicated that the step-based PDR system based on cellphones is enlarged by integrating a short-range proximity access control system, referred to as SRP, that can be RFID- or NFC-based and is connected to the interior map's boundaries. While SRP is limited by the short-range over which a smartphone can communicate with an SRP device, the function becomes advantageous when the target to be monitored is within the effective range of an SRP device. When used in conjunction with conventional PDR, the precise position obtained can calibrate the inaccuracy imposed by sensor drift in PDR, allowing PDR to give long-term accurate positioning performance as long as a dense enough density of SRP devices is available. Simultaneously, the PDR system contributes to reducing required SRP device density by correcting for tracking between adjacent SRP nodes.

Additionally, unlike an external SRP network, which requires the user to connect with nodes/tags, the interaction occurs effortlessly within the access control system, a common scenario in buildings, particularly office and public buildings. As a result, the proposed HILN system can easily be integrated into existing SRP access control systems. While utilizing commercially available hardware, the proposed system can provide long-term reliable localization and tracking performance with acceptable precision.

2.3 Multilayer perceptron

The multilayer perceptron is a technique for dealing with large amounts of data. This is demonstrated in (Ullah Khan et al., 2020), where the author explains that after obtaining all experimental data, the author analyses and constructs a mathematical model of the behavior of signal intensities in the 2D experimental area by training a suitably designed Multilayer Perceptron (MLP) network, which has inherent capabilities for classifying nonlinear data. Many pre-processing steps are required to apply the proposed tessellation approach to the MLP algorithm utilized to handle the indoor localization scenario. The primary processes are to create the tiling array, pre-process the raw MLP input signals, and associate the tiles with binary codes that correspond to their places in the conventional MLP output format.

2.3.1 Multilayer perceptron training

This is demonstrated in (Ullah Khan et al., 2020), where the MLP network was trained using the well-established approach of error reduction via backpropagation. The objective of MLP training is to identify appropriate weight values W1 and W2 such that any input vector in the training set matches its target within a specified error margin. The author continued by repeating the training of the MLP network an unlimited number of times to study its behavior and record the typical parameters corresponding to minima in the value of the testing data's error. The initial few iterations were used to fine-tune the learning and momentum rates to minimize the number of training steps and compute time. During this phase, training was conducted utilizing all 84 points without regard for testing data selection. The purpose of this study was to determine the algorithm's speed, the type of mismatches between actual and predicted tiles, and the stability of the error reduction technique based on backpropagation.

In (Ullah Khan et al., 2020), each run of the MLP training was conducted using a different set of random testing points to explore any possible effect of such an initial decision on the accuracy of the findings. However, no significant effect on the random selection of the testing data was detected in multiple runs in terms of accuracy. The learning rate and momentum are rapidly established through trial and error to strike a compromise between the speed of descent of the training error and the stability of the backpropagation process. A Multilayer

Perceptron Layer (MLP) network (Ullah Khan et al., 2020) was trained to forecast the tile placement based on the hybrid signal strength. Positioning accuracy was significantly improved as a result of the achieved position error.

2.4 Radial Basis Function Network

Radial Basis Function (RBF) networks are a subclass of Artificial Neural Networks (ANNs) comprised of three layers: an input layer, a hidden layer, and an output layer. It is typically designed as a single hidden layer with a nonlinear RBF activation function and a linear output layer, as discussed in (Salim & Mohammed, 2020). RBF networks have been applied to a wide variety of problems, including system identification, nonlinear function approximation, adaptive control, speech recognition, real-time approximation, and pattern classification. RBF Networks exhibit characteristics that distinguish them from other neural networks. It has been widely employed in a wide variety of sectors of research and engineering due to its capacity for global approximation and faster learning, as well as its ability to avoid solving for local minimums.

Another journal article (Guo et al., 2014) describes how Radial Basis Function Neural Networks (RBFNNs) was built for the indoor location. In the reference (Ding et al., 2010), an RBFNN is used to learn the object location based on received RSSI values. According to the reference (Moreno-Cano et al., 2013), a hybrid RFID/IR indoor localization mechanism is presented for usage in intelligent buildings. An RBF network is used to estimate the location of inhabitants, and a particle filter is utilized to track their subsequent positions. Because RBFNN has several advantages over other types of ANNs, including simple network topology, rapid learning, and good approximation capability, it was chosen for RFID indoor location. The RBFNN architecture comprises three layers: an input layer, a hidden layer, and an output layer. As illustrated in the picture below, RBFNN is characterized by the number of hidden neurons, the center vector and width value used to determine the Gaussian activation of each hidden neuron, and the connection weights between hidden neurons and output neurons. The RSSI vector is imported into the input layer, the number of neurons in the hidden layer is M, and the number of neurons in the output layer is equal to 2, corresponding to the estimated coordinate(x,y).

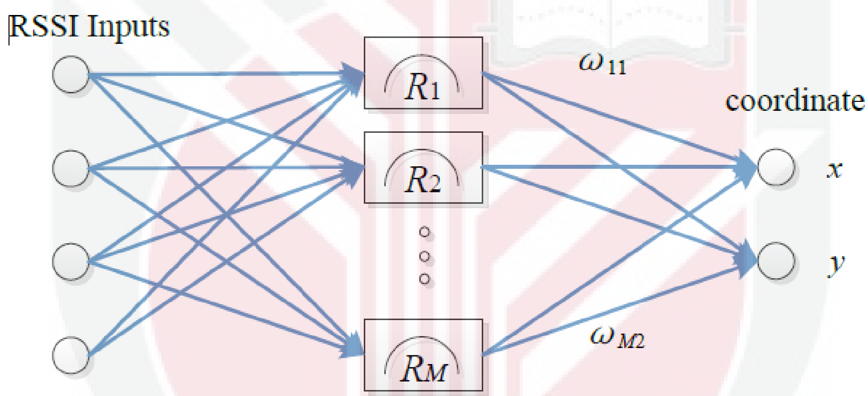

Figure 2.4 : structure of RBF neural network.

2.4.1 Radial Basis Function Network Training

The training of RBF is described in detail in (Salim & Mohammed, 2020), where the network is trained in two phases: The algorithm determines the weights from the input layer to the hidden layer in the first phase and then from the hidden layer to the output layer in the second phase. The following diagram illustrates the building of a fully connected RBF network.

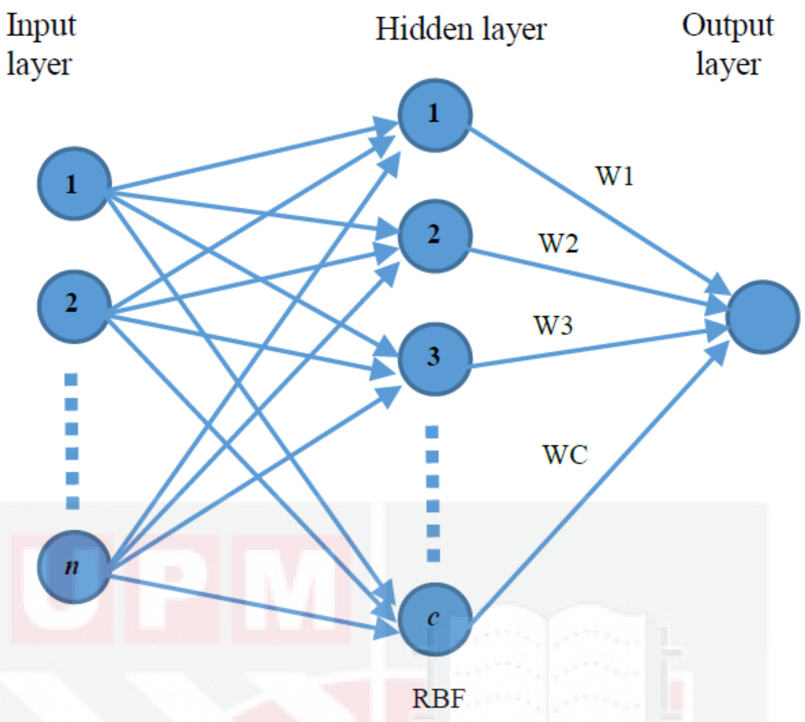

Figure 1.4.1: structure of RBF neural network.

The input vector n is the input for all radial basis functions. f(x) is the output which is expressed as:

$$f(x) = \sum_{n=1}^{c} W_n \, \sigma \left(||x - c_n||\right)$$

Where C is the number of neurons in a hidden layer, $||x - c_n||$ is the Euclidean distance between x and the n-dimensional basis function center c_n, we are the network weights.

Radial basis function has been used with fingerprinting in indoor localization. The input layer represents the received signal strength (RSS) measurements from three access points. In the second layer, the number of neurons is determined according to the requirements of the actual application. The Gaussian function was utilized in the Wi-Fi positioning project due to the positive definite function in any dimensional space and having a single solution. The formula of the Gaussian h(x) is as follows:

$$h(x) = \exp(-\frac{(x-c)^2}{r^2}$$

In the output layer, there are two outputs (x and y) that represent the location of the user inside the building. RBFNN has significantly been used to achieve indoor positioning systems. It can be trained to reduce the localized generalization error and is adopted to estimate locations based on RSS values. Compared to other types of ANNs, the advantage of RBFNN is that they are simple to be structured, fast to learn, and they can estimate locations with more minor errors. RBFNN is a neural network structure of three consequently connected layers: an input, a hidden, and an output layer. It should be identified by defining the number of hidden neurons, center vector, and width value which defines the Gaussian activation of each hidden neuron and the connection weights between the hidden and the output neurons.

The gathered RSS values in the database are loaded to the network, which is received by the input layer. The number of neurons in the hidden layer is N, and two neurons in the output layer represent the estimated coordinates x and y. RBF is a function whose value is affected by the distance from a certain point, in this context, a signal sample point. RBFs are triggered in the second layer, i.e., the hidden layer, to calculate the distance between the input vector and the weights. In the summation layer, a summation operation is done for the hidden output values of each class, and their connection weights are created instantly from the targets. If the input pattern belongs to the specific class, connection weight will equal 1; otherwise, it will equal 0. A decision output will be finally generated according to the competitive rule (winner takes all). The author continues in (Salim & Mohammed, 2020) by stating that the RBF network lacks the iterative feature required to generate weights for matching the inputs and target outputs. It is a significant aspect of this ANN network that contributes to its rapid training speed. When the network is in the training stage, it adjusts the weights of the connections between the input and hidden layers to match the previously provided samples. The hidden layer computes the distance between the input vector and the training patterns during the test stage and generates a vector of inputs that are near the training pattern. The summation layer consists of a vector of elements, each of which describes the probability of the "input vector to each class." The summation vector's values will be watched, and the summation node with the highest value in the network will be a competitive winner. In the decision layer, a class winner is chosen by representing the winner as '1'. The remainder of the classes is represented by '0's. Thus, a winner is always present in this instance.

The most extensively used RBFNN training processes, according to (Ding et al., 2010), are separated into two phases: unsupervised and supervised. The number of hidden neurons (M), the center vector (U), the width value (v) used to determine the Gaussian activation of each hidden neuron, and the connection weights (w) between the hidden neurons and the output neuron constitute an RBFNN. In the unsupervised phase, with a fixed M value, centers are assigned by clustering the training samples into M clusters and using their centroids as the center vectors for the RBFNN's hidden neurons. To determine the center vectors in this study, the conventional clustering algorithm k-means is used. The buried neuron's width is determined by the average distances between all centroids. The connection weights are determined using the least-square approach during the supervised learning phase.

2.5 Previous Study

No	Author	Year	Method	Findings
1.	Ullah Khan et al	2020	MLP	Predict the tile location using MLP. The achieved position the error resulted in a significant improvement of the positioning accuracy.
2.	Jefferson et al	1995	ANN	Introduction to neural network
3.	Powell, M. J. D	1977	RBF	Use RBF to solve the real multivariate interpolation problem.
4.	Tsvi Kulfik et al	2011		challenges and solutions for indoor cultural heritage sites
5.	Farid et al	2013		Recent Advances in Wireless Indoor Localization Techniques and System
6.	Zhang et al	2014	MaWi	Using Hybrid Magnetic and Wi-Fi System for Scalable Indoor Localization
7.	Tian et al	2015	HILN	hybrid in-building localization and navigation (HILN) system for pedestrians
8.	Salim & Mohammed	2020	RBF	Radial Basis Function Neural Network-Based Indoor Positioning System Radial Basis Function Neural Network-Based Indoor Positioning System
9	Guo et al	2014	RBF	Develop RBFNN to accomplish indoor positioning

10	Ding et al	2010	RBF	RFID indoor positioning using RBFNN with L-GEM
11	Moreno-Cano et al	2013	ANN	An indoor localization system based on artificial neural networks and particle filters applied to intelligent buildings
12	Sahin, F	1997		A Radial Basis Function Approach to a Color Image Classification Problem in a Real-Time Industrial Application

CHAPTER 3

METHODOLOGY

3.1 Introduction

In this chapter, the method and list equipment for this research study were explored. To accomplish that, previous studies on Indoor Location systems are used as references. Also, the steps and order of this study were illustrated in this chapter. This research study was conducted using MATLAB. A total of 576 retrospective data were collected throughout the 96 points. The data was then trained, and error was calculated. The workflow is shown in the figure below.

3.2 Data Collection

In this study, the dataset was used from UKM, which was prepared for refs. This dataset consists of signal strength on each of 96 points from 6 signals. Ninety-six points were set as a grid of 16 x 6. The six signals consist of 3 WSN and 3 WAP. in the grid, all signals measured strictly at 100 cm of height. In this dataset, there are 576 data for training.

3.3 Learning strategies

In an RBF network, different layers perform different tasks. Therefore, it is useful to separate the optimization of the remote unit and output layers of the network by using different techniques as explained in (Sahin, F., 1997). There are different learning strategies in the design of an RBF network depending on how the centers of RBFs of the network are determined. There are three major approaches to determine the centers, which are Fixed Centers Selected at Random, Self-Organised Selection of Centers, and Supervised Selection of Centers. We will go for the third approach.

In the third approach, a supervised learning process is employed to obtain the centers of the radial basis functions and all other free parameters of the network. In (Jefferson et al., 1995), the author claims that the RBF network goes into its most generalized form in this approach. He also states that error-correction learning is a natural candidate for such a process, which is most conveniently implemented using a gradient-descent procedure that indicates a generalization of the LMS algorithm. Defining the instantaneous value of the cost function can be the first step in the generation of such a learning process. The cost function can be given by

$$\xi = \frac{1}{2}\sum_{j=1}^{N} e_j^2$$

where N is the number of training samples used to undertake the learning procedure, and ej is the error signal given by

$$e_j = d_j - F(x_j)$$

$$e_j = -\sum w_i G(\left\| x_j - t_i \right\|)_{c_i}$$

In this approach, the calculation of the free parameters w_i, t_i, and $\sum_i^{-1}$ is necessary in order to minimize ξ. The covariance matrix $\sum_i$ is related to the norm-weighting matrix C.

The Gaussian function used in this study is as shown in (Sahin, F., 1997), where the author used the function as means to solve the regularization problem. The Gaussian function is given by

$$F(x) = \sum_{i=1}^{N} w_i \exp(-\frac{1}{2\sigma_i^2}||x - x_i||^2)$$

3.4 Training Phase

In this research, MATLAB was the programming language that chose the algorithm to run in RBF. There are two sets of code developed for this study. The first set of codes is used to run the experiment one variable at a time. The second set of experiments is used to run the experiment with many variables in one run. There are five steps in developing the code for this study. The summary of the process is illustrated below.

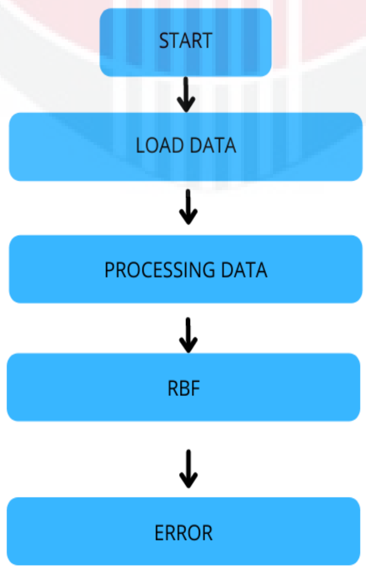

Figure 2.4: Code summary.

3.4.1 Data Loading

Firstly, the data is required to be loaded into MATLAB. In order to achieve this, a load command was used. The way load command function is loading data from the selected file. The file needs to be in the same folder as the MATLAB workspace. The general line of code for load command is "load(filename)".In this study, the file is named "Hybrid_data". In MATLAB, the line of code will be typed as "load Hybrid_data". This will load the data into the MATLAB workspace.

After the data has been loaded successfully, the data need to be in the workspace. The data is then categorized into signal strength, position X, and position Y. The loaded data will be denoted by "A ."In the data, there are eight columns and 96 rows of data. First, the signal strength is categorized. There is six signal, and the signal strength includes column 1 until column 6. 6 signal will be denoted as "Sig". The line of code for this is written as " Sig=A(:,1:6)". Next, the points were categorized. There are 96 points in position x and position y. Position x is in column 7, and position y is in column 8. The line of code for points is " X=A(:,7) and Y=(:,8).

The final code for loading data into MATLAB is shown below.

```
%------------- Load data ----------------------------------------%

load Hybrid_data             %-- Get the data in A

Sig=A(:,1:6);                %-- Signals
X=A(:,7);                    %-- Position X [m]
Y=A(:,8);                    %-- Position Y [m]
```

Figure 3.4.1: Data loading code.

3.4.2 Data Pre-processing

After loading, data has been completed. This study then proceeds to pre-process the data. This step is important to simplify the process. This is due to the original value of signal strength being too big. In this step, the signal strength is categorized in the range of maximum and minimum. The range is from 1 until 10. The lowest signal strength will be one, and the maximum signal strength will be 10. The processed signal will be a normalized signal. This can be achieved using the Normalization formula shown below.

$$Normalized\ signal = \frac{(signal\ strength - minimum\ \ value)\ 10}{(maximum\ value - minimum\ value}$$

In order for this equation to work the maximum and minimum need to be denoted. This is achieved by using similar step in data loading. In order to normalized the signal strength the maximum and minimum will be denoted for signal 1. The line of code is written as “ma=max(Sig(:,1))”, “mi=min(Sig(:,1));” the line of code for the equation is “Norm_Sig(:,1)=(Sig(:,t)-mi)*10/(ma-mi)”. Since there are six signal, manual method can be used for all six signal as shown below

```
ma=max(Sig(:,1));
mi=min(Sig(:,1));
Norm_Sig(:,1)=(Sig(:,1)-mi)*10/(ma-mi);
ma=max(Sig(:,2));
mi=min(Sig(:,2));
Norm_Sig(:,2)=(Sig(:,2)-mi)*10/(ma-mi);
ma=max(Sig(:,3));
mi=min(Sig(:,3));
Norm_Sig(:,3)=(Sig(:,3)-mi)*10/(ma-mi);
```

ma=max(Sig(:,4));

mi=min(Sig(:,4));

Norm_Sig(:,4)=(Sig(:,4)-mi)*10/(ma-mi);

ma=max(Sig(:,5));

mi=min(Sig(:,5));

Norm_Sig(:,5)=(Sig(:,5)-mi)*10/(ma-mi);

ma=max(Sig(:,6));

mi=min(Sig(:,6));

Norm_Sig(:,6)=(Sig(:,6)-mi)*10/(ma-mi);

However, the code can be simplified using the for loop command. For loop function is used to repeat a specified number of times. For this to work, general code needs to be used instead of specific code. To generalize the code, the data loading step was referred to where the number represents signal one until six is denoted as "t". The final code for data pre-processing is shown below.

```
for t=1:6

    ma=max(Sig(:,t));
    mi=min(Sig(:,t));

    Norm_Sig(:,t)=(Sig(:,t)-mi)*10/(ma-mi);

end
```

Figure 3.4.2:final code for data pre-processing.

3.4.3 Radial Basis Function Network

In this study, the Gaussian function was used for RBF. The RBF formula used in this experiment is as shown below.

$$rb = Aexp\frac{\frac{-1}{2}}{s^2}((cx - xp)^2 + (cy - yp)^2)$$

A is the signal strength, cx and cy is the coordinate of random center, xp and yp are coordinates, and s is sigma.

For each signal, RBF will be defined. Each rbf uses a random center in the data. The RBF consists of amplitude, position x, position y, and sigma. Position x and position y are the coordinates for the random center. The position of the random center is selected randomly using the rand command. Rand command is used to uniformly distribute random numbers. The line of code for the first rbf is “RBF(1,:)=[2 rand*15-5 rand*20 10];”

In this study, 4 RBF will be defined for each signal. In order to do that similar technique used in data pre-processing was referred to where for loop will be used. The code will be generalized in order to use for loop. The final code for RBF is shown below.

```
for t=1:4

    rbf(t,:)=[2 rand*15-5 rand*20 10];
    plot(rbf(t,2),rbf(t,3),'r*');

end
```

Figure 3.4.3.1: The final code for RBF.

After that, this study proceeds to calculate the rbf value. The code for the formula was divided into two codes. The first is the general code, and the second is the calculation code. This code includes the four RBF and for loop once again used. The final code is shown below.

```
rb=@(A,cx,xp,cy,yp,s) A*exp(-1/2/s^2*((cx-xp)^2+(cy-yp)^2));

for t=1:4
    sum_part(t) = rb(rbf(t,1),rbf(t,2),X(1),rbf(t,3),Y(1),rbf(t,4));
end
```

Figure 3.4.3.2: The final code for RBF value.

From this code, the RBF value for each RBF can be evaluated from the first rbf until the fourth RBF.

3.4.4 Error

After we got the RBF value, this study then proceeded to calculate the error. The general formula for error is the signal value minus the rbf value. The formula used is normalized signal strength minus the total of rbf value. By using this method, the error value can be calculated. The formula for error is shown below.

$$error = normalized\ Signal\ strength - (sum(sum\ part))^2$$

The code for the error is shown below.

```
err = (Norm_Sig(1,1)- sum(sum_part))^2;
```

Figure 3.4.4: Code for the error.

The (1,1) in "Norm_Sig(1,1)" refer to the column and row of where the signal is called from. (1,1) means this signal is referring to the signal value from column 1 and row 1.

After that, the global error will be calculated. This is due to there are six errors on each of the 96 points. The sum will be done for this.

3.4.5 MATLAB Code

The final code for all the steps explained before is shown below.

```
%------------- Load data ---------------------------------------%

load Hybrid_data            %-- Get the data in A

Sig=A(:,1:6);               %-- Signals
X=A(:,7);                   %-- Position X [m]
Y=A(:,8);                   %-- Position Y [m]

%------------- Preprocess signals ---------------------------------%
for t=1:6

    ma=max(Sig(:,t));
    mi=min(Sig(:,t));

    Norm_Sig(:,t)=(Sig(:,t)-mi)*10/(ma-mi);

end
%-- Plot

figure(1);
plot(X,Y,'bo');
hold on; grid on; axis equal;
% for cycle on all 6 signals
for t=1:4

    rbf(t,:)=[2 rand*15-5 rand*20 10]; %-- Amplitude, cx, cy, sigma)
    plot(rbf(t,2),rbf(t,3),'r*');

end

%for cycle on all points

rb=@(A,cx,xp,cy,yp,s) A*exp(-1/2/s^2*((cx-xp)^2+(cy-yp)^2));

for t=1:4
    sum_part(t) = rb(rbf(t,1),rbf(t,2),X(1),rbf(t,3),Y(1),rbf(t,4));
end

err = (Norm_Sig(1,1)- sum(sum_part))^2;

%end
```

Figure 3.4.5.1: Final code for all the steps.

This method, however, can only run one variable at one time. In order to increase efficiency, for loop is added to the existing code. The aim in the first run of the experiment is to find the error of all six signals for all 96 points. In order to do this, there are a few lines of code that need to be changed into generalized form. For this position x, position y and normalized signal denoted as “X(po)”, “Y(po)” and “Norm_Sig(po,pox)" respectively. In order to run multiple variables at the same time, for loop is used. The final code is shown below.

```
%------------- Load data ---------------------------------------%

load Hybrid_data            %-- Get the data in A

Sig=A(:,1:6);               %-- Signals
X=A(:,7);                   %-- Position X [m]
Y=A(:,8);                   %-- Position Y [m]

%------------- Preprocess signals --------------------------------%


for t=1:6
    ma=max(Sig(:,t));
    mi=min(Sig(:,t));

    Norm_Sig(:,t)=(Sig(:,t)-mi)*10/(ma-mi);
end
%-- Plot
figure(1);
plot(X,Y,'bo');
hold on; grid on; axis equal;
% for cycle on all 6 signals
rb=@(A,cx,xp,cy,yp,s) A*exp(-1/2/s^2*((cx-xp)^2+(cy-yp)^2));

for pox=1:1:6
for po=1:1:96

for t=1:4
    rbf(t,:)=[2 rand*15-5 rand*20 10];  %-- Amplitude, cx, cy, sigma)
    plot(rbf(t,2),rbf(t,3),'r*');
end

%for cycle on all points
for t=1:4
    sum_part(t) = rb(rbf(t,1),rbf(t,2),X(po),rbf(t,3),Y(po),rbf(t,4));
end

err = (Norm_Sig(po,pox)- sum(sum_part))^2;

end
end
%end
```

Figure 3.4.5.2: Second final code for all the steps.

CHAPTER 4

RESULTS AND DISCUSSION

4.1 Introduction

In this chapter, the results based on the objective of the research mentioned in chapter 1 were presented as follows. In section 4.2, discussion on the data trained with radial basis function network and further discussion on error optimization-based of the trained data. The aim of this section is to reduce the error as low as possible. This can be achieved in two methods. In 4.2.1, the first method discuss code alteration, and in 4.2.2, the second method discuss sigma substitution

4.2 Error optimization

In this study, 576 data were trained with a radial basis function network. For each signal, a total of 4 radial basis function has been defined, and the code was run using the same sigma value of 10. The error for all points is illustrated in the graph below.

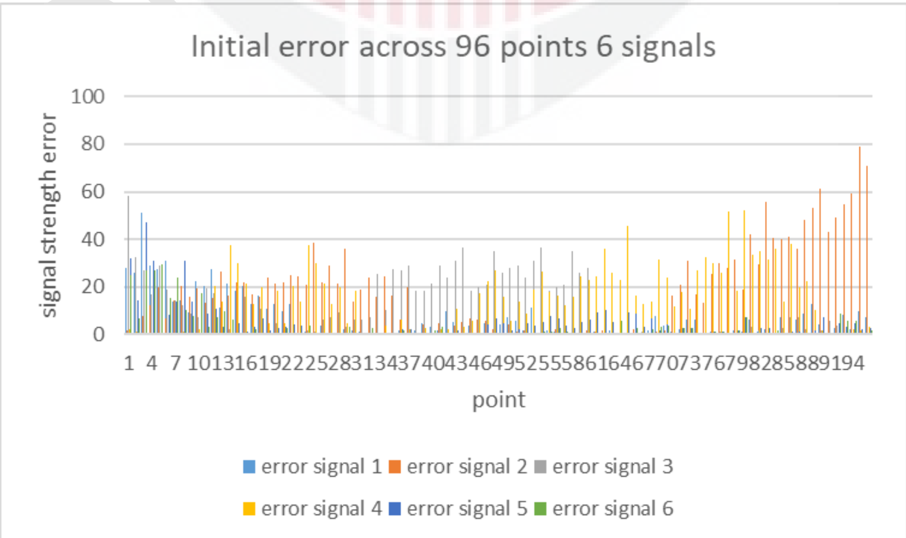

Figure 4.2: Initial error across 96 points 6 signals.

4.2.1 Code Alteration

Based on graph 1, error optimization was executed based on the initial trained data. In this section, code alteration will be done. During the initial training, it is observed that when the random majority of the RBF center stack too close to each other, it will make the error value not stable when the code was run a few times. In order to prevent this, the RBF center allocated into their own range coordinate. An area divided for each random center is similar to the quadrant technique where each center occupies one quadrant. Instead of using for loop, a manual method was used. The technique is shown below.

```
rbf(1,:)=[2 rand*15 rand*20 s];
plot(rbf(1,2),rbf(1,3),'r*');
rbf(2,:)=[2 rand*-10 rand*20 s];
plot(rbf(2,2),rbf(2,3),'r*');
rbf(3,:)=[2 rand*-10 rand*11.25 s];
plot(rbf(3,2),rbf(3,3),'r*');
rbf(4,:)=[2 rand*15 rand*11.25 s];
plot(rbf(4,2),rbf(4,3),'r*');
```

Figure 4.2.1.1: RBF manual code.

The result of these changes is illustrated below.

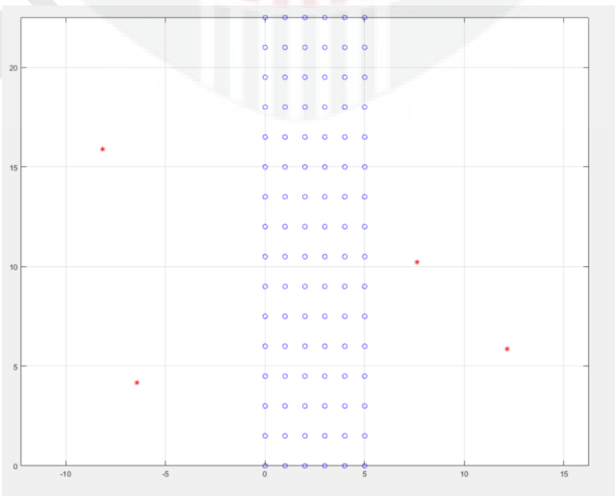

Figure 4.2.1.2: New RBF center.

As shown above, the RBF center is still randomly selected, but it will stay at its designated range. This will reduce the probability of all the four rbf center stacks to close to each other. The center is shown as a red star, and 96 points are shown as a small blue circle.

4.2.2 Sigma Substitution

Other than the code alteration method, another method was utilized in order to reduce error further. In this section, the sigma value will be adjusted until the lowest error value is achieved. This method will substitute the sigma value in the code. The initial sigma value that will be tested is from one until 300. The first run will only be done on the first point across all six signals. Sigma value will be changed from one until 200 in the interval of one. The experiment is started with the first point and only the first signal strength. The code is as below.

```
for pox=1:1:1
for po=1:1:1
for s=1:1:300
    rbf(1,:)=[2 rand*15 rand*20 s];                    %-- Amplitude, cx, cy, sigma)
    plot(rbf(1,2),rbf(1,3),'r*');
    rbf(2,:)=[2 rand*-10 rand*20 s];                    %-- Amplitude, cx, cy, sigma)
    plot(rbf(2,2),rbf(2,3),'r*');
    rbf(3,:)=[2 rand*-10 rand*11 s];                    %-- Amplitude, cx, cy, sigma)
    plot(rbf(3,2),rbf(3,3),'r*');
    rbf(4,:)=[2 rand*15 rand*11 s];                    %-- Amplitude, cx, cy, sigma)
    plot(rbf(4,2),rbf(4,3),'r*');

%for cycle on all points
for t=1:4
    sum_part(t) = rb(rbf(t,1),rbf(t,2),X(po),rbf(t,3),Y(po),rbf(t,4));
end

err = (Norm_Sig(po,pox)- sum(sum_part))^2;

end
end
```

Figure 4.2.2.1: Code for first point first signal.

The result for the code above will be shown in the graph below.

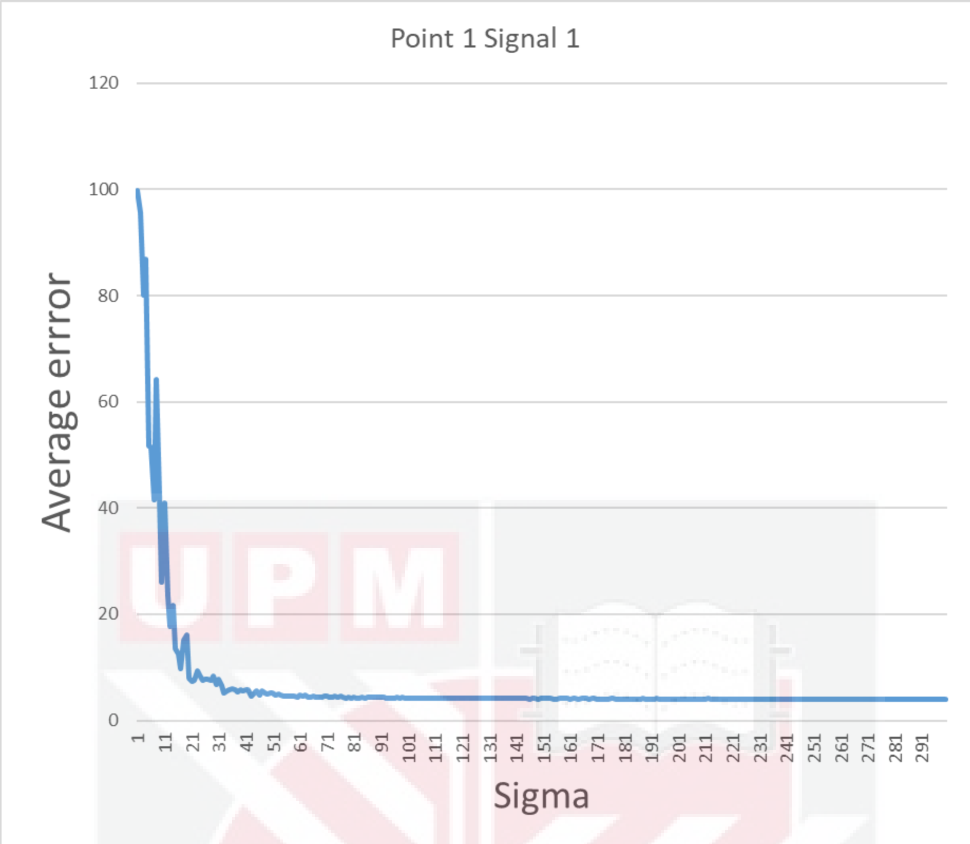

Figure 4.2.2.2: Point 1 Signal 1.

In the first run, the error was calculated three times, and the average was taken. Based on the graph above, the error decreased significantly from sigma value one until 23. A minor decrease in error occurs until sigma 100. The error still continues to decrease, albeit in a really small value until sigma 200. As shown in the graph sigma value bigger than 200 will only provide a diminishing return on the error value. In sigma 199, the error value is 4.05, while the initial error is 27.9839 with sigma 10.

The study then proceeds to the second signal while keeping the point and sigma value range constant—the graph for the second signal is shown below.

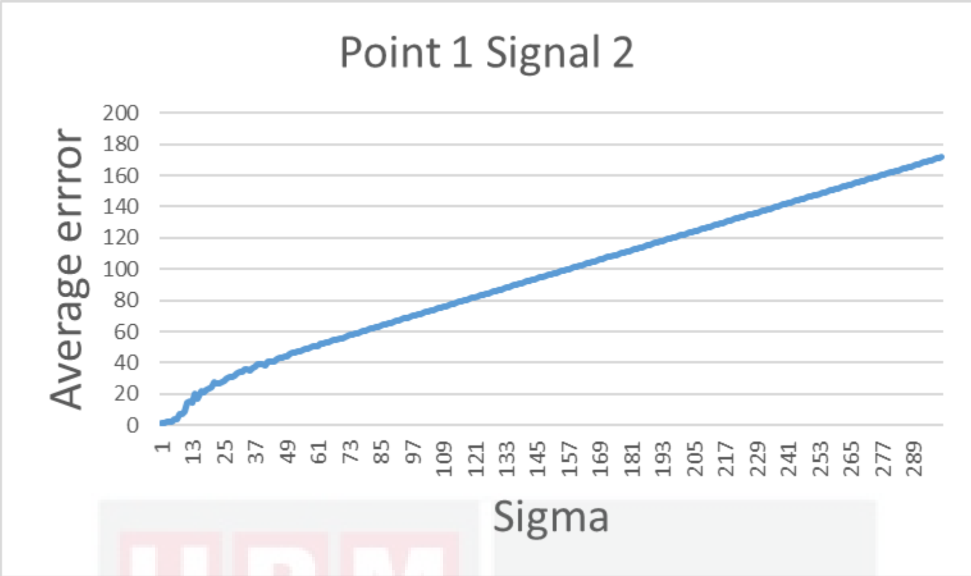

Figure 4.2.2.3: Point 1 Signal 2.

Based on the graph above, the error decrease when the sigma value decreases. Due to this condition, the sigma value will be tested, including the negative sigma value. New sigma will range from -50 until 50. The same method is used. The result is shown below.

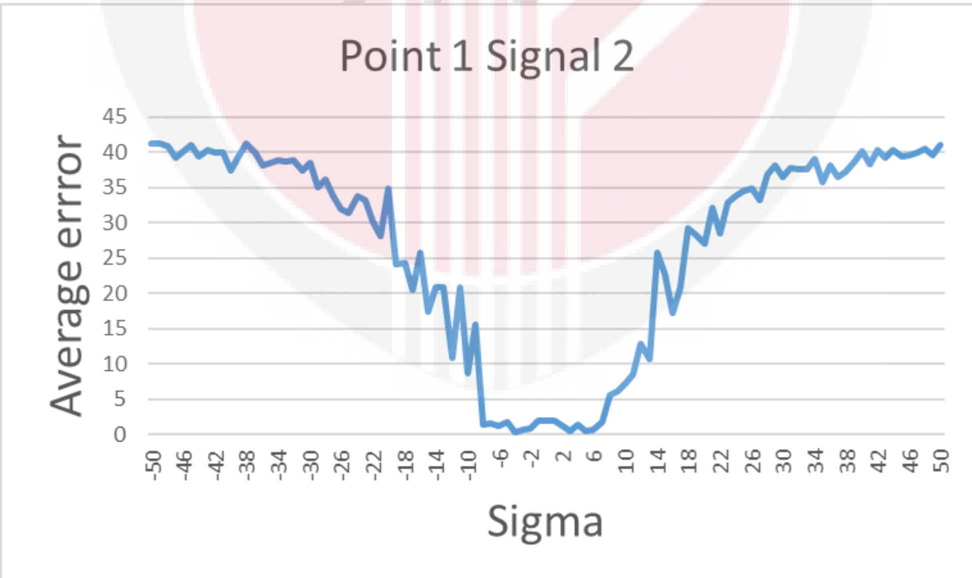

Figure 4.2.2.4: Point 1 Signal 2.

Based on the graph, we can observe that errors still decrease even when the sigma value is negative. Based on this graph, we conclude that the lowest error is 0.382 at sigma=-4 while the initial error is 1.764 at sigma=10

The study then proceeds to the third signal while keeping the point and sigma value range constant as per the first signal. The graph for the third signal is shown below.

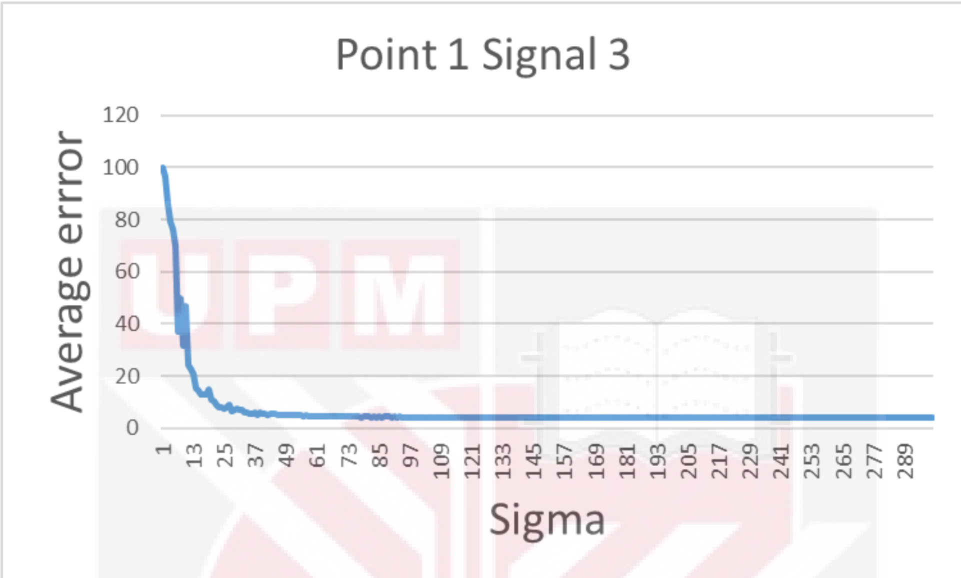

Figure 4.2.2.5: Point 1 Signal 3.

Based on the graph for signal 3, the trend for the error is the same as signal 1. The error decreases significantly from sigma value one until 25. A minor decrease in error occurs until sigma 100. The error still continues to decrease, albeit in a really small value until sigma 200. As shown in the graph sigma value bigger than 200 will only provide a diminishing return on the error value. In sigma=200, the error value is 4.05, while the initial error is 58.8 with sigma=10.

The study then proceeds to the fourth signal while keeping the point and sigma value range constant as per the first signal. The graph for the fourth signal is shown below.

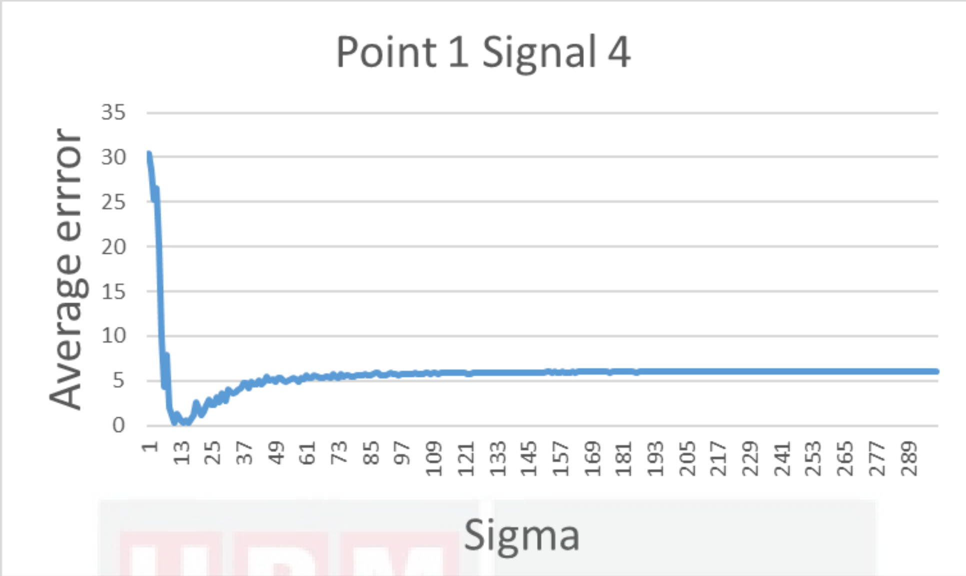

Figure 4.2.2.6: Point 1 Signal 4.

Based on the graph for signal 4, the trend for the error is quite similar to signal 1. However, in signal 4, the lowest error is more evident. The error decreases significantly from sigma=1 until 16. After that, the error value increase. In sigma=16, the error value is 0.294, while the initial error is 2.24 with sigma=10.

The study then proceeds to the fifth signal while keeping the point and sigma value range constant as per the first signal. The graph for the fifth signal is shown below.

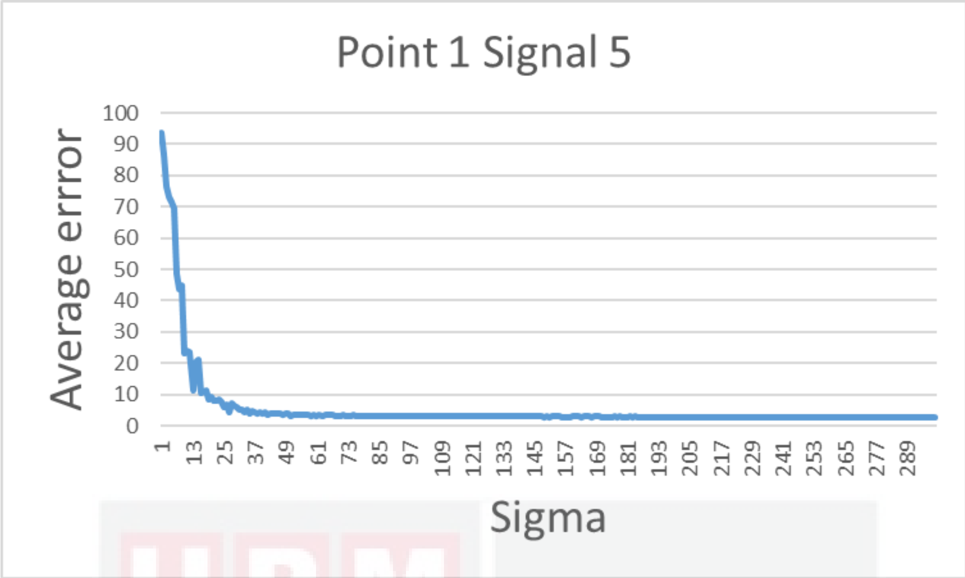

Figure4.2.2.7:Point 1 Signal 5.

Based on the graph above, the error decreased significantly from sigma value one until 13. A minor decrease in error occurs until sigma 100. The error still continues to decrease, albeit in a really small value until sigma 200. As shown in the graph sigma value bigger than 200 will only provide a diminishing return on the error value. In sigma=200, the error value is 2.88, while the initial error is 32.24 with sigma=10.

The study then proceeds to the sixth signal as the final signal for the first point while keeping the point and sigma value range constant as per the first signal. The graph for the sixth signal is shown below.

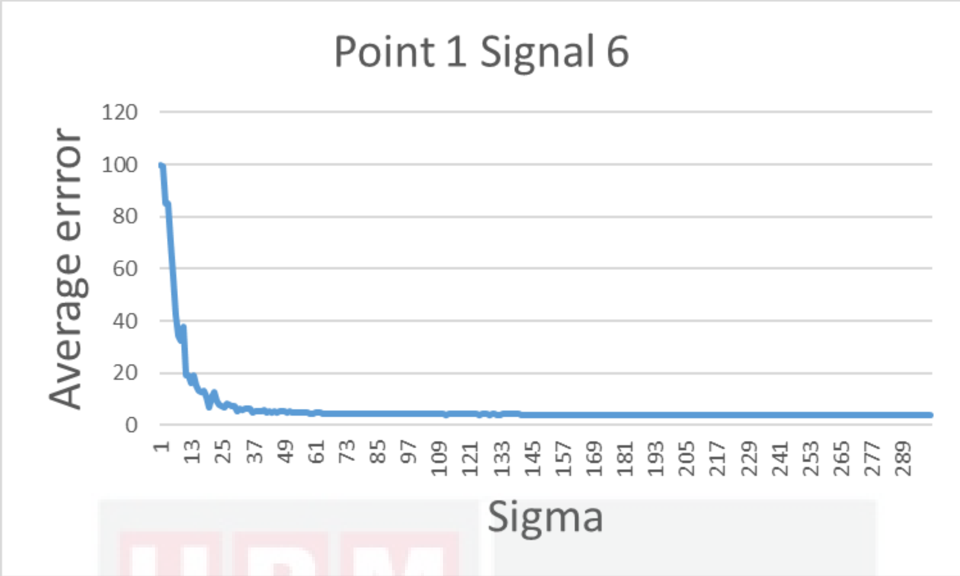

Figure4.2.2.8: Point 1 Signal 6.

Based on the graph above, the error decreased significantly from sigma value one until 13. A minor decrease in error occurs until sigma 100. The error still continues to decrease, albeit in a really small value until sigma 200. As shown in the graph sigma value bigger than 200 will only provide a diminishing return on the error value. In sigma=200, the error value is 4.039, while the initial error is 24.69 with sigma=10.

Based on the error optimization method for all six signals, a few conclusions can be made in order to increase efficiency when running the experiment for all 96 points. Firstly the sigma value range will be changed from sigma=1 to sigma=300 into sigma=-50 to sigma=200. This is due to more than 200 only giving a diminishing return. Sigma =-50 is just a random range just in case it needs a bigger negative value. The study proceeds to calculate the error for the rest of 96 points, in order to achieve this, for loop once again added to the code. Another line of code is also added to increase efficiency. The code is "disp(err)". This code function as it automatically displays the error in the command window. The final code is shown below.

```
%------------- Load data ---------------------------------------%

load Hybrid_data            %-- Get the data in A

Sig=A(:,1:6);               %-- Signals
X=A(:,7);                   %-- Position X [m]
Y=A(:,8);                   %-- Position Y [m]

%------------- Preprocess signals ---------------------------------%
for t=1:6
    ma=max(Sig(:,t));
    mi=min(Sig(:,t));
    Norm_Sig(:,t)=(Sig(:,t)-mi)*10/(ma-mi);
end
%-- Plot
figure(1);
plot(X,Y,'bo');
hold on; grid on; axis equal;
% for cycle on all 6 signals
for pox=1:1:6     %signal 1-6
for po=1:1:96     %point 1-96
for s=-50:1:200   %sigma value

    rbf(1,:)=[2 rand*15 rand*20 s];          %-- Amplitude, cx, cy, sigma)
    plot(rbf(1,2),rbf(1,3),'r*');
    rbf(2,:)=[2 rand*-10 rand*20 s];         %-- Amplitude, cx, cy, sigma)
    plot(rbf(2,2),rbf(2,3),'r*');
    rbf(3,:)=[2 rand*-10 rand*11.25 s];      %-- Amplitude, cx, cy, sigma)
    plot(rbf(3,2),rbf(3,3),'r*');
    rbf(4,:)=[2 rand*15 rand*11.25 s];       %-- Amplitude, cx, cy, sigma)
    plot(rbf(4,2),rbf(4,3),'r*');
%for cycle on all points
rb=@(A,cx,xp,cy,yp,s) A*exp(-1/2/s^2*((cx-xp)^2+(cy-yp)^2));
for t=1:4
    sum_part(t) = rb(rbf(t,1),rbf(t,2),X(po),rbf(t,3),Y(po),rbf(t,4));
end
err = (Norm_Sig(po,pox)- sum(sum_part))^2;
disp(err);
end
end
end
%end
```

Figure 4.2.2.9: Final code on error optimization.

This study then proceeds to reduce error on all points and all six signals. The end result is illustrated in the graph below.

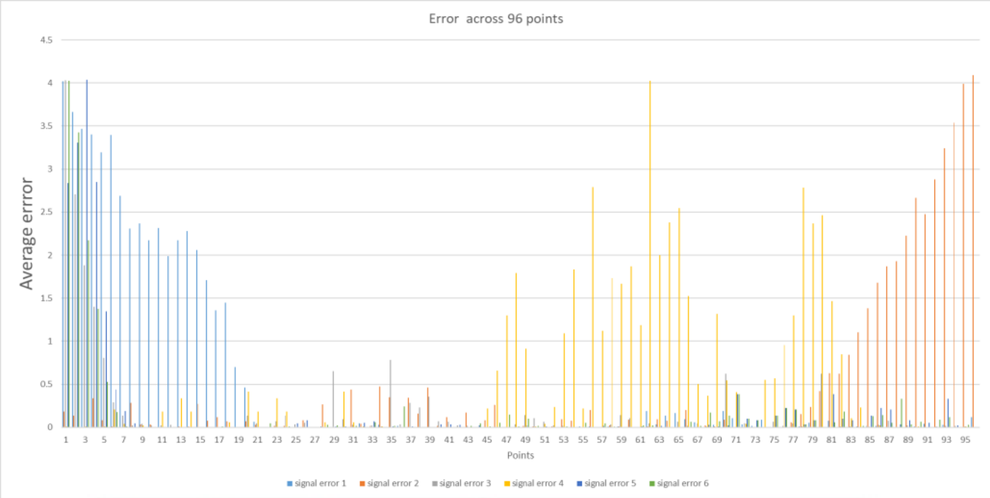

Figure 4.2.2.10: Error across 96 points.

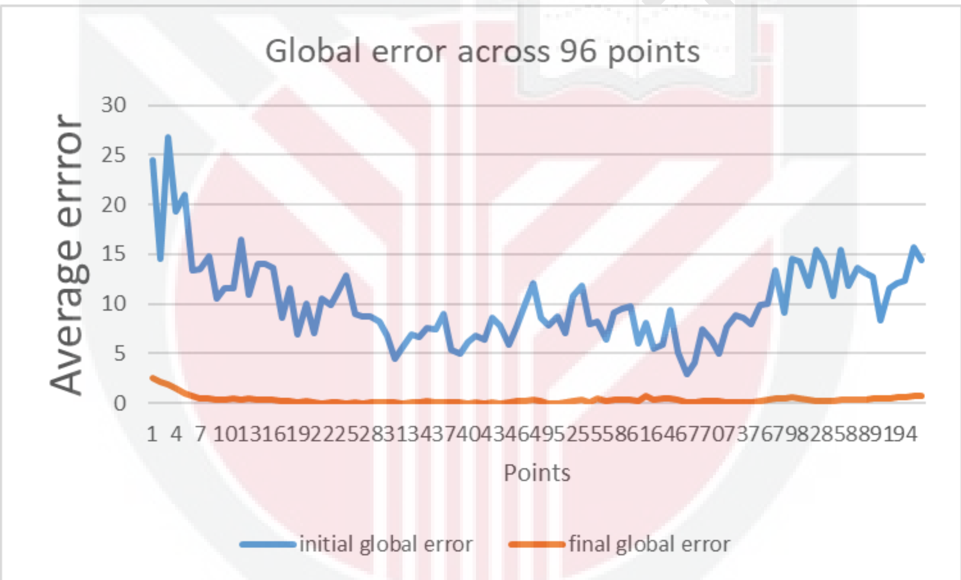

Figure 4.2.2.11: Global error across 96 points

CHAPTER 5

CONCLUSION

5.1 Conclusion

This project explores the possibilities of using a radial basis function network as an alternative in processing the data from indoor positioning inside buildings. The radial basis function network method is highly cost-effective because there is no need for additional investment. This technology is inexpensive, easily applied, and fastly deployed. This is due to the required computer to run the training is low. Even the low spec computer can run the training. From the experimental results, the RBFNN method provides fast training. Moreover, the underlying RBF architecture is scalable and can be easily applied to different indoor positioning setups,

5.2 limitation of study

Even though previously mentioned that RBF is easy to run, hardware limitations cannot be avoided when dealing with a huge amount of data. In this study, my computer could only handle 2000 data before the computer crash. Originally, 144000 data was intended to run in the first run. This caused the computer to crash, and a lot of time was spent redoing the training. This problem is caused by RAM capacity limitation because it will run at 96% capacity before the computer crash. The RAM capacity for the computer is 16GB.

5.3 Recommendations for future study

The data of research can be improved by using a computer with a higher RAM capacity. By having that, the training can be done much quicker, and more parameters can be added to the same training.

REFERENCES

Baala, O., Zheng, Y., & Caminada, A. (2009). The impact of AP placement in WLAN-based indoor positioning system. *Proceedings of the 8th International Conference on Networks, ICN 2009*, 12–17. https://doi.org/10.1109/ICN.2009.50

Ding, H. L., Ng, W. W. Y., Chan, P. P. K., Wu, D. L., Chen, X. L., & Yeung, D. S. (2010). RFID indoor positioning using RBFNN with L-GEM. *2010 International Conference on Machine Learning and Cybernetics, ICMLC 2010*, *3*(July), 1147–1152. https://doi.org/10.1109/ICMLC.2010.5580925

Farid, Z., Khan, I. U., Scavino, E., & Abd Rahman, M. A. (2019). A WLAN Fingerprinting Based Indoor Localization Technique via Artificial Neural Network. *International Journal of Computer Science and Network Security*, *19*(7), 157–165.

Farid, Z., Nordin, R., & Ismail, M. (2013). Recent advances in wireless indoor localization techniques and system. *Journal of Computer Networks and Communications*, *2013*. https://doi.org/10.1155/2013/185138

Guo, D., Zhang, Y., Xiang, Q., & Li, Z. (2014). Improved radio frequency identification indoor localization method via radial basis function neural network. *Mathematical Problems in Engineering*, *2014*(July 2014). https://doi.org/10.1155/2014/420482

Jefferson, M. F., Pendleton, N., Lucas, S., Horan, M. A., & Tarassenko, L. (1995). Neural networks. *The Lancet*, *346*(8991–8892), 1712. https://doi.org/10.1016/S0140-6736(95)92880-4

Moreno-Cano, M. V., Zamora-Izquierdo, M. A., Santa, J., & Skarmeta, A. F. (2013). An indoor localization system based on artificial neural networks and particle filters applied to intelligent buildings. *Neurocomputing*, *122*, 116–125. https://doi.org/10.1016/j.neucom.2013.01.045

Sahin, F., & Bay, J. S. (1998). A Radial Basis Function Approach to a Color Image Classification Problem in a Real-Time Industrial Application. *1st International Conference on Multisource-Multisensor Information Fusion*, 873–879.

Salim, A. T., & Mohammed, E. A. (2020). *Radial Basis Function Neural Network-Based Indoor Positioning System Radial Basis Function Neural Network-Based Indoor Positioning System*. *November 2019*.

Tian, Q., Salcic, Z., Wang, K. I. K., & Pan, Y. (2015). A hybrid indoor localization and navigation system with map matching for pedestrians using smartphones. *Sensors (Switzerland)*, *15*(12), 30759–30783. https://doi.org/10.3390/s151229827

Ullah Khan, I., Ali, T., Farid, Z., Scavino, E., Amiruddin Abd Rahman, M., Hamdi, M., & Qiao, G. (2020). An improved hybrid indoor positioning system based on surface tessellation artificial neural network. *Measurement and Control (United Kingdom)*, *53*(9–10), 1968–1977. https://doi.org/10.1177/0020294020964242

Zhang, C., Luo, J., & Wu, J. (2014). Poster abstract: MaWi: A hybrid Magnetic and Wi-Fi system for scalable indoor localization. *IPSN 2014 - Proceedings of the 13th International Symposium on Information Processing in Sensor Networks (Part of CPS Week)*, 275–276. https://doi.org/10.1109/IPSN.2014.6846760

Huang, G.-B., & Siew, C.-K. (2005). Extreme learning machine with randomly assigned RBF kernels. International Journal of Information Technology, 11(1), 16–24.

Er, M. J., Wu, S., Lu, J., & Toh, H. L. (2002). Face recognition with radial basis function (RBF) neural networks. IEEE Transactions on Neural Networks, 13(3), 697–710. https://doi.org/10.1109/TNN.2002.1000134

Huang, G. Bin, Saratchandran, P., & Sundararajan, N. (2005). A generalized growing and pruning RBF (GGAP-RBF) neural network for function approximation. IEEE Transactions on Neural Networks, 16(1), 57–67. https://doi.org/10.1109/TNN.2004.836241

Calise, A., Hovakimyan, N., & Lee, H. (2000). Adaptive output feedback control of nonlinear systems using neural networks. Proceedings of the American Control Conference, 5(1), 3153–3157.

Huang, G. Bin, Saratchandran, P., & Sundararajan, N. (2004). An efficient sequential learning algorithm for growing and pruning RBF (GAP-RBF) networks. IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 34(6), 2284–2292. https://doi.org/10.1109/TSMCB.2004.834428