IMPROVING THE ACCURACY OF INDOOR POSITIONING SYSTEM

Mohammed Muwafaq Noori Hameez

MASTER'S THESIS

Submitted to the School of Graduate Studies of Kadir Has University in partial fulfillment of the requirements for the degree of Master of Science in the Program of Computer Engineering

ISTANBUL, JULY, 2019

DECLARATION OF RESEARCH ETHICS /

METHODS OF DISSEMINATION

I, Mohammed Muwafaq Noori Hameez, hereby declare that;

- this master's thesis is my own original work and that due references have been appropriately provided on all supporting literature and resources;
- this master's thesis contains no material that has been submitted or accepted for a degree or diploma in any other educational institution;
- I have followed "Kadir Has University Academic Ethics Principles" prepared in accordance with the "The Council of Higher Education's Ethical Conduct Principles"

In addition, I understand that any false claim in respect of this work will result in disciplinary action in accordance with University regulations.

Furthermore, both printed and electronic copies of my work will be kept in Kadir Has Information Center under the following condition as indicated below:

☐ The full content of my thesis will not be accessible for two years. If no extension is required by the end of this period, the full content of my thesis will be automatically accessible from everywhere by all means.

Mohammed Muwafaq Noori Hameez

31.JULY.2019

KADIR HAS UNIVERSITY

SCHOOL OF GRADUATE STUDIES

ACCEPTANCE AND APPROVAL

This work entitled IMPROVING THE ACCURACY OF INDOOR POSITIONING SYSTEM prepared by Mohammed Muwafaq Noori Hameez has been judged to be successful at the defense exam held on 31. JULY.2019 and accepted by our jury as master's thesis.

APPROVED BY:

Asst. Prof. Dr. Taner Arsan (Advisor)
Kadir Has University

Assoc. Prof. Dr. Osman Kaan Erol
Istanbul Technical University

Asst. Prof. Dr. Arif Selçuk Öğrenci
Kadir Has University

I certify that the above signatures belong to the faculty members named above.

Prof. Dr. Sinem Akgül Açkmeşe
Dean of School of Graduate Studies
DATE OF APPROVAL: 31. JULY.2019

TABLE OF CONTENTS

IMPROVING THE ACCURACY OF INDOOR POSITIONING SYSTEM

ABSTRACT

Indoor positioning applications needs high accuracy and precision to overcome the existing obstacles and relatively small areas. There are several methods which could be used to locate an object or people in an indoor location. Specifically, Ultra-wide band (UWB) sensor technology is a promising technology in indoor environments because of its high accuracy, resistance of interference and better penetrating.

This thesis is focused on improving the accuracy of UWB sensor based indoor positioning system. To achieve that, optimization and machine learning algorithms are implemented. The impact of Kalman Filter (KF) on the accuracy is introduced in the implementation of the algorithms.

The average localization error is reduced by approximately 54.53% (from 16.34 cm to 7.43 cm), when combining the big bang - big crunch algorithm (BB-BC) with Kalman Filter. Finally, a Hybrid (BB-BC KF K-Means) algorithm is improved and implemented separately, and the best results are obtained from this Hybrid algorithm. Thus, it has been obtained that the average localization error is reduced significantly by approximately 64.26% (from 16.34 cm to 5.84 cm).

Keywords: Indoor positioning, Ultra-wide band, Big bang-big crunch algorithm, Genetic algorithm, K-Means algorithm, Fuzzy C-Means algorithm, Mean Shift algorithm, Clustering, Average silhouette method, Kalman Filter.

İÇ KONUM BELİRLEME SİSTEMİNİN DOĞRULUĞUNUN İYİLEŞTİRİLMESİ

ÖZET

İç mekan konum belirleme uygulamaları, nispeten daha küçük alanlarda kullanılmak ve mevcut engellerle başa çıkmak için dış mekan konum belirleme yöntemlerinden daha yüksek doğruluk ve hassasiyet gerektirir. İç mekandaki bir nesnenin veya insanın konumlarını belirlemek için kullanılabilecek çeşitli yöntemler bulunmaktadır. Özellikle, Ultra geniş bant (UWB) sensör teknolojisi, yüksek doğruluğu, bozuculara olan direnci ve iç mekan uygulamalarında geniş bant sinyallerinin her taraftan algınabilmesi özelliği sayesinde iç mekan konum belirlemede gelecek vaad eden bir teknolojidir.

Bu tez çalışması, UWB sensör tabanlı iç mekan konum belirleme sisteminin doğruluğunu arttırmaya odaklanmıştır. Bunu başarmak için, optimizasyon ve makine öğrenmesi algoritmaları kullanılmıştır. Kalman Filtresi (KF)'nin konum belirleme doğruluğu üzerindeki etkisi algoritmaların uygulanması esnasında görülmüş ve açıklanmıştır.

Büyük patlama - büyük çöküş algoritması (BB-BC), Kalman filtresiyle birleştirildiğinde, ortalama konum belirleme hatasının yaklaşık %54,53 oranındığı görülmüştür (16,34 cm'den 7,43 cm'ye düşer). Son olarak, bir Hibrit (BB-BC KF K-Ortalamalar) algoritma ayrı olarak geliştirilmiş ve uygulanmıştır, en iyi sonuçlar bu Hibrit algoritmadan elde edilmiştir. Bu sayede, ortalama lokalizasyon hatasının yaklaşık %64,26 oranında (16,34 cm'den 5,84 cm'ye) önemli ölçüde azaldığı belirlenmiştir.

Anahtar kelimeler: İç mekân konum belirleme, Ultra geniş bant, Büyük patlama - büyük çöküş algoritması, Genetik algoritma, K-Ortalamalar algoritması, Bulanık C-Ortalamalar algoritması, Ağırlıklı Ortalama Öteleme Algoritması, Kümeleme, Ortalama silhouette yöntemi, Kalman Filtresi.

ACKNOWLEDGEMENTS

I would like first to thank my thesis advisor Asst. Prof. Dr. Taner Arsan, Computer Engineering Department at Kadir Has University. This work wouldn't have been possible without his guidance and valuable instructions. The door to his office was always open whenever I run into trouble or had question about my thesis, and for that I am grateful.

I must express my very profound gratitude to my parents for providing me with unfailing support and continuous encouragement throughout my years of study.

Finally, to my wife Mareb, I would like to thank you for your patient and support throughout my study, and also for taking care of our beautiful newly born girl during my absence.

LIST OF TABLES

LIST OF FIGURES

LIST OF SYMBOLS/ABBREVIATIONS

a_i	The average dissimilarity between the *i*th
A	Status transition matrix in Kalman Filter
α	Standard normal distribution
$b_i(k)$	Average distance
c_j	Clusters center
$f(x_0, x_n)$	Objective function
H	Observation matrix in Kalman Filter
I	Filter deviation matrix
k	The iteration step in Big bang-big crunch algorithm
K	Number of clusters in K-Means algorithm
$K(t)$	Kernel in Mean Shift algorithm
K_k	Kalman gain matrix
L	Lower boundary in Big bang-big crunch algorithm
N	Population size in Big bang-big crunch algorithm
$Npop$	Population size in Genetic algorithm
Q	Process noise in Kalman Filter
r	Normal random number
R	Covariance matrix
s_i	Silhouette coefficient of the *i*th data point
u_{ij}	The degree of membership in clustering
u_{k-1}	System control vector in Kalman Filter
U	Upper Boundary in Big bang-big crunch algorithm
v_k	Observation noise vector in Kalman Filter
w_k	System noise vector in Kalman Filter
x_k	Status vector in Kalman Filter
x_{ir}	Real location value in x-dimension
x_{io}	The required offset value in x-dimension
x_{im}	The measured value in x-dimension
$\hat{x}_k$	Optimum filter value (posterior status estimation)

$\vec{x}^c$	Center of mass in Big bang-big crunch algorithm
y_{io}	The required offset value in y-dimension
y_{im}	The measured value in y-dimension
y_{ir}	Represent the real location in x-dimension
z_k	Observation vector in Kalman Filter
ε	Termination criterion
σ	The bandwidth in Mean Shift algorithm

AAL	Ambient Assistant Living
AOA	Angle of Arrival
AT &T	American Telephone & Telegraph
APIT	Approximate Point In Triangle
BB-BC	Big Bang – Big Crunch
BLE	Bluetooth low energy
CDF	Cumulative probability functions
CL	Centroid Localization
FCM	Fuzzy C-Means
FFD	Full Function Device
FIS	Fuzzy Inference System
GA	Genetic Algorithm
GNSS	Global Navigation Satellite System
GPS	Global Positioning System
GSM	Global System for Mobile Communications
ICL	Intelligent Centroid Localization
IPS	Indoor Positioning System
KDE	Kernel Density Estimates
KF	Kalman Filter
KNN	K-Nearest Neighbor
LBS	Location-based services
LOS	Line-of-Sight
MS	Mean Shift
NLOS	Non-Line-of-Sight

RF	Radio frequency
RFD	Reduced Function Device
RFID	Radio-frequency Identification
RSS	Received Signal Strengths
RTOF	Roundtrip Time of Flight
SVM	Support Vector Machine
TDOA	Time Difference of Arrival
TOA	Time of Arrival
ToF	Time of Flight
UWB	Ultra-wide band
WCSS	Within-cluster sum of squares
WSN	Wireless Sensor Network

1. INTRODUCTION

1.1 Indoor Positioning

Indoor positioning determines the location of objects, people, and other equipment in an indoor area. Indoor positioning has been the subject of widely growing interest in the last few years because of the demand for more accurate and reliable location-based services (LBS) (Cai et al., 2017).

Position of a device or user in a given environment is considered an important part of contextual information. And the extensive spread of sensors has produced growing wealth of such information. Location by itself, has generated great attention due to its potential to support a variety of applications (Brena et al., 2017).

Position estimation solutions are based on multi-lateration and triangulation methods using ultrasound, light, or radio signals, and they manage to provide locational information. Triangulation uses the geometric properties of triangles to estimate the target position. It includes two derivations: lateration and angulation. The lateration derivations estimates the location of an object by measuring its distances from multiple reference points. Instead of measuring the distance directly using Received Signal Strengths (RSS), the Time of Arrival (TOA) or Time Difference of Arrival (TDOA) is usually measured, and the distance is derived by computing the attenuation of the emitted signal strength or by multiplying the travel time and radio signal velocity. Roundtrip Time of Flight (RTOF) is a method that can be used to perform range estimation function in some systems. Whereas, in Angulation the object is located by computing the angles that is relative to a number of reference points. There are also other techniques and methods, which provide relative positioning such as, inertial methods. However, they accumulate errors in require periodic recalibration and in time. So, to locate an indoor object; tags, labels, or tokens can be used (Liu et al., 2007). Positioning systems have different architectures configurations, accuracies, and reliabilities.

Some of the indoor positioning system are Global Positioning system (GPS) AT&T Cambridge Ultrasonic Bats, Active Badges, active Bats, Wi-Fi, Radio Frequency Identification (RFID) technology, Bluetooth low energy (BLE) and Ultra-wide band (UWB) (Koyuncu and Yang, 2010).

In outdoor environments, location detection has been very successfully using GPS technology. The GPS technology has made huge impact on our lives by supporting a wide range of applications in mapping, guidance and other beneficial applications. Nonetheless, in indoor environments, the use of the GPS or other equivalent satellite-based location systems is restricted or limited, because of the lack of attenuation and line of sight of GPS signals while they cross through walls (Brena et al., 2017).

High sensitivity GPS can provide positioning in some indoor locations. Although the signals are heavily reflected and attenuated by building materials. It was observed that highly sensitive GPS receivers can track people through three layers of brick wall, but positioning accuracy were very low. The accuracy of some 50 meters inside a place with commercial setting is useless with respect to a job of locating specific products on the shelf. Thereby, the demand for specialized technologies and methods for indoor location systems has become widely accepted (Gu et al., 2009).

1.2 Indoor Positioning Applications

The following applications shows the necessity for indoor positioning and location- based technology in our daily life. However, more applications will be found from the future generation of indoor positioning and even more use cases to utilize its capabilities, in which at the moment are not possible (Mautz, 2012).

1.2.1 Location Based Services (LBS)

LBS are required in both outdoors and indoors. An example of indoor use is acquiring topical or safety on cinemas, events or concerts. Also, LBS applications provide navigation to stores in a shopping mall or office in large building. In general, the location-based advertisements, local search services and location-based billing have a high commercial value. Another use of LBS is to provide guidance for the guests to the exposition booths. The bus stations or train applications that include the directing to the bus stop or platform.

1.2.2 Private homes

The applications in houses include LBS at home, item detections, and physical game gesture. For example, the Ambient Assistant Living (AAL) systems provide help for old people in their house. The core function of this systems is positioning which enabled by an indoor positioning functionality.

Other Applications at houses are detection of emergencies, Patient monitoring such as monitoring vital signs (Zetik et al. 2010). It also personalized and service entertainment systems, for example, smart audio systems

1.2.3 Context detection and awareness

Mobile devices offer wide range of helpful functions, in which it is appealing to have an automated adaptation of the user device depending on the change of the user's context. Indoor positioning system technologies can utilize smart personal mobile devices, and non-smart non-personal mobile devices (beacons and object tags), for purposes of tracking and locating people and objects.

The greatest interest is given to technologies that incorporate smart mobile devices because users with these devices are the largest class for indoor positioning systems. For example, a smart event guide that provide information about the subject that it's been held in nearby auditoriums.

1.2.4 Medical service

In medical facilities the position determining of medical personnel in emergency situations become very important. Other applications in medical facilities also include tracking of patient and medical equipment. Other example is fall detection of the patients, providing an accurate positioning is essential for robotic assistance while operating surgeries.

1.2.5 Logistics and Optimization

To achieve optimization specially in complex systems, it’s highly necessary to obtain valuable information regarding the position of the staff members and assets. Thus, when it come to complex and large storage areas, it is very important that the needed products are located without any delay.

1.2.6 Police forces and firefighters’ services

Indoor positioning provides benefits for the rescue services, law enforcement, and provide fire services. For example, the position determining of firemen in building that are on fire. Whereas, the police benefits from variety of applications, for example, immediate detection of burglary or theft, locating of stolen products for incident investigations and develop of smart alarm systems that can detect if someone or an asset left unauthorized area.

1.3 Indoor and Outdoor Positioning Systems characteristics

Many characteristics makes the indoor positioning systems differ from the outdoor positioning systems. The indoor environments consider to be more complex due to the multiple objects (for example, walls, equipment and people) which reflect signals and produce multi-path and delay problems. Although, because of the presence of objects, indoor environments depend on Non-Line-of-Sight (NLoS) propagation in which the signals cannot move directly in straight way from the transmitter end to the receiver end, that will cause delays in the receiver end. The presence of objects produces signal scattering and high attenuation. Figure 1.1 shows the difference between Line-of-Sight (LOS) and NLOS (Alarifi et al., 2016).

Indoor positioning experiences a signal stability, as the signal power fluctuate easily because of the presence of interference sources such as mobile devices, Zigbee devices, Bluetooth devices, cordless phones, wireless devices, fluorescent lights, and microwave ovens. Also, the indoor environments suffer from structural movements in a way that structures may be there location changed from one area to another. As a result, this might calibrate and tune the positioning system to overcome with any recent changes in the structure. Also, indoor environments tend to be less dynamic due to objects movement at a slower speed within them (Mautz, 2012).

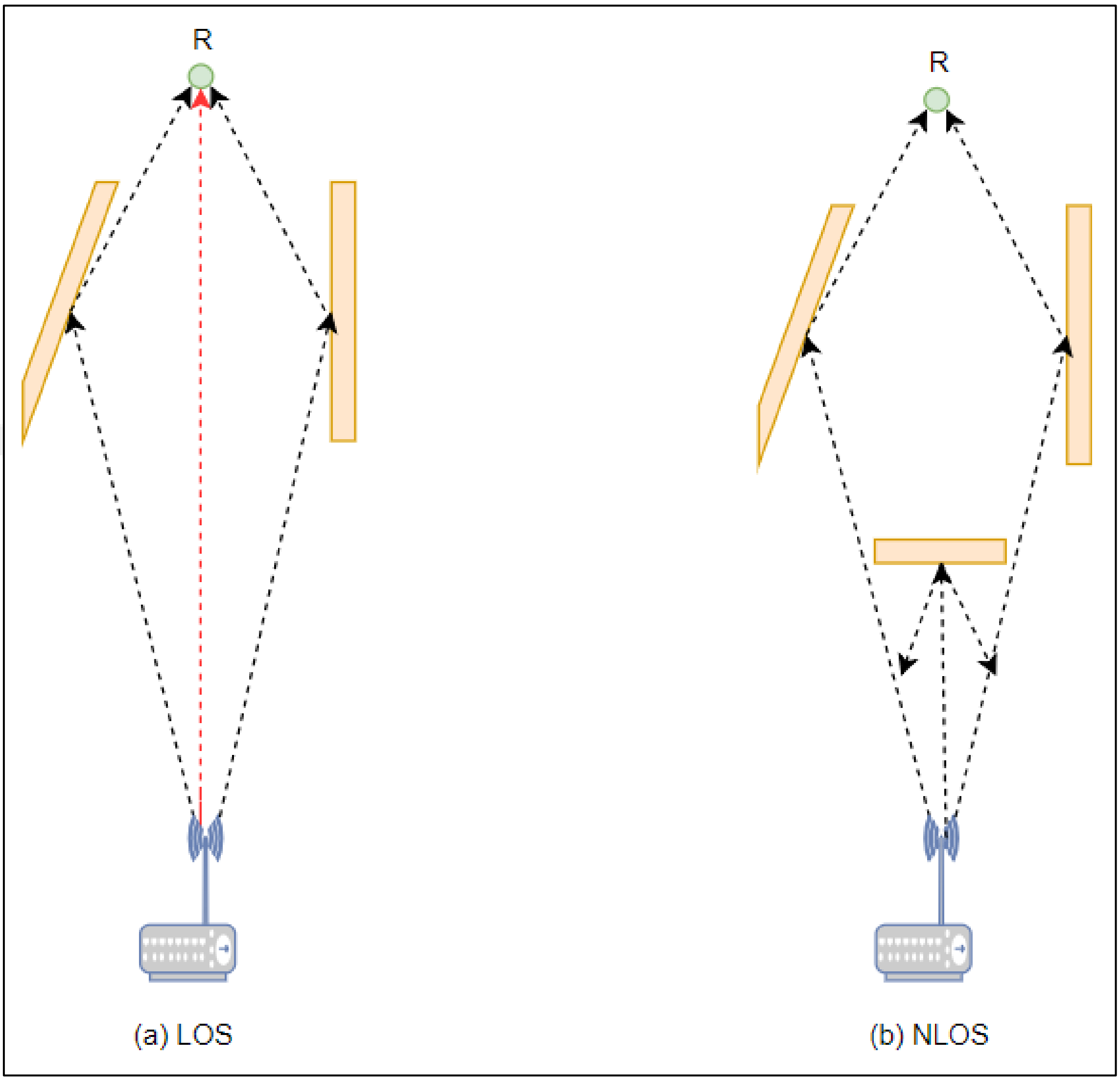

Figure 1.1 Line-of-Sight and Non-Line-of-Sight.

The outdoor positioning area has been dominated by Global Navigation Satellite System (GNSS). In their basic version, these systems provide precision in the order of meters. There have been other methods developed to increase the positioning precision. Most of them are based on using a reference station, or a network of stations, in order to improve the systems performance and overcome their limitations. With some of these methods, sub meter accuracy can be achieved while using a GNSS system. Outdoor positioning can be also achieved by using the ubiquitous mobile network base stations. In this case, the precision lies in the order of several meters, and depends on the number of surrounding base stations. The outdoor positioning is dominated by the use of GNSS, even though there are already integrated circuits which combine GNSS and cellular positioning. The indoor positioning domain is a bit more chaotic than the outdoor Positioning. In which, there is no prevailing standard for indoor positioning and several technologies have been used to provide position data. One of the technologies that have been utilized the most is IEEE 802.11 (Wi-Fi) (Sedlacek et al., 2016).

1.4 Indoor Positioning Technologies

The first indoor positioning technology that developed by AT&T Cambridge were Active badges. Each employee wears a device in this system that able to transmit an infrared signal. Then, all the outcome data from the infrared sensors are collected by central database and with help of RF tags which are worn each employee, the positions of all users are identified. As disadvantage of this technology, it can only be utilized for short-range communications because the infrared technique needs a LOS between both end the transmitter end and the receiver end (Want et al., 1992).

Active Bats, which is an ultrasonic technology named were developed also by AT&T Cambridge. This technology can provide an accuracy that is higher then what found in active badges. The user in this technology wears badges that transmit ultrasonic pulses for the transmitter end. Then, it uses a triangulation method and measures the Time-of-Flight (ToF) of this pulse from the transmitter end to point in the ceiling. Using such technique, we can calculate the distance between bats to each receiver. However, the implementation of this system is difficult because of the large number of transmitters devices that need to be installed and also the adjustment they require (Ward et al., 1997).

Radio-Frequency Identification (RFID) is a means of storing and also retrieving data over electromagnetic transmission to an RF compatible integrated circuit. The RFID reader can read the data emitted from RFID tags. The RFID readers use protocol and RF to transmit and also receive the data. RFID tags can be either active or passive. The advantage of RFID technology over ultrasonic positioning systems is the lower cost (Ni et al. 2004). Figure 1.2 shows a typical passive RFID system, while Figure 1.3 shows active RFID system.

The ZigBee offers security, networking, and services regarding the application support.

It is a low rate and short distance wireless personal area network. The ZigBee node is small with low cost and complexity. It includes microcontroller and also a multichannel two-way radio. The Zigbee is developed for applications that don't require high data throughput and high-power consumption. Two physical devices used in ZigBee nodes; (1) Full Function Device (FFD); (2) Deduced Function Device (RFD) (Mautz, 2012).

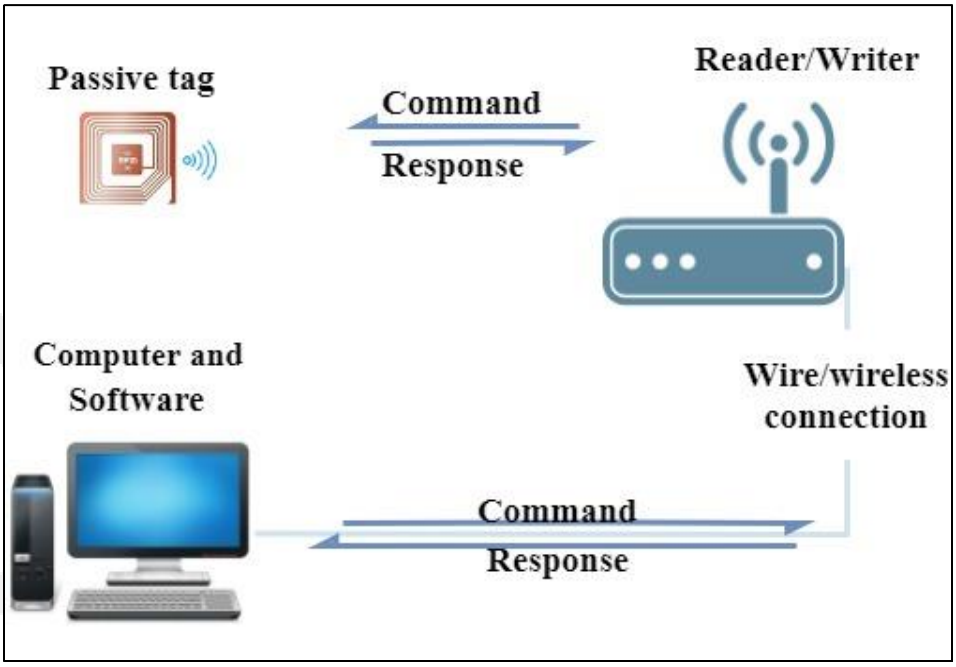

Figure 1.2 Passive RFID system.

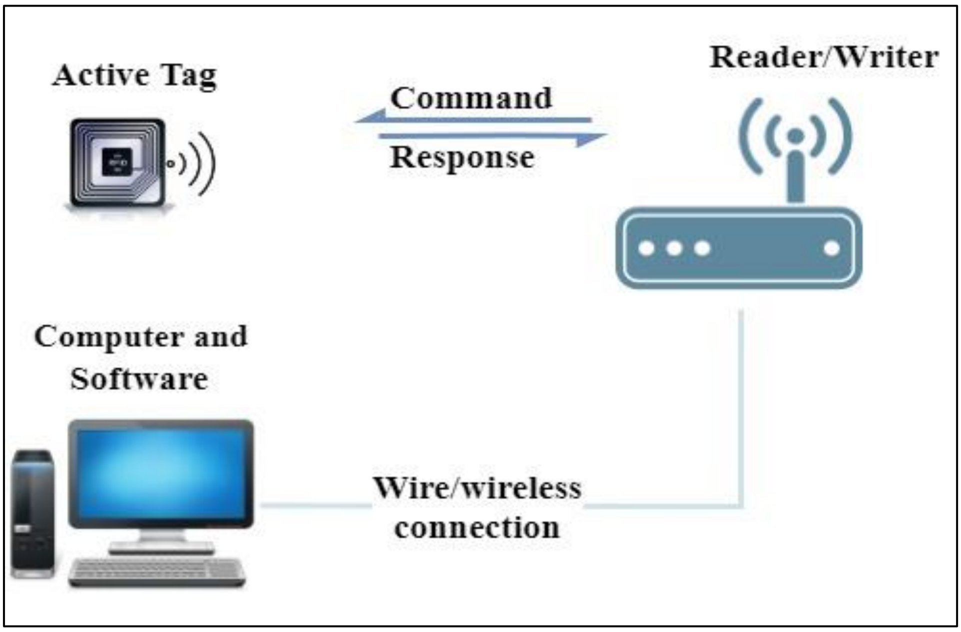

Figure 1.3 Active RFID system.

Wi-Fi can be considered to be very popular technology that can be used for wireless communication. Wi-Fi is very popular in enterprise locations and public hotspots during the last few years. Wi-Fi operates on Industrial, Scientific and Medical (ISM) band including 2.4 GHz and a range of (50 m to 100 m). IEEE 802.11 become the dominant local wireless networking standard. Therefore, it's desirable to use the already existing WLAN infrastructure for indoor positioning by adding a location server (Jekabsons et al., 2011). Figure 1.4 shows indoor Wi-Fi based localization, which utilize received signal strength in indoor Wi-Fi environment.

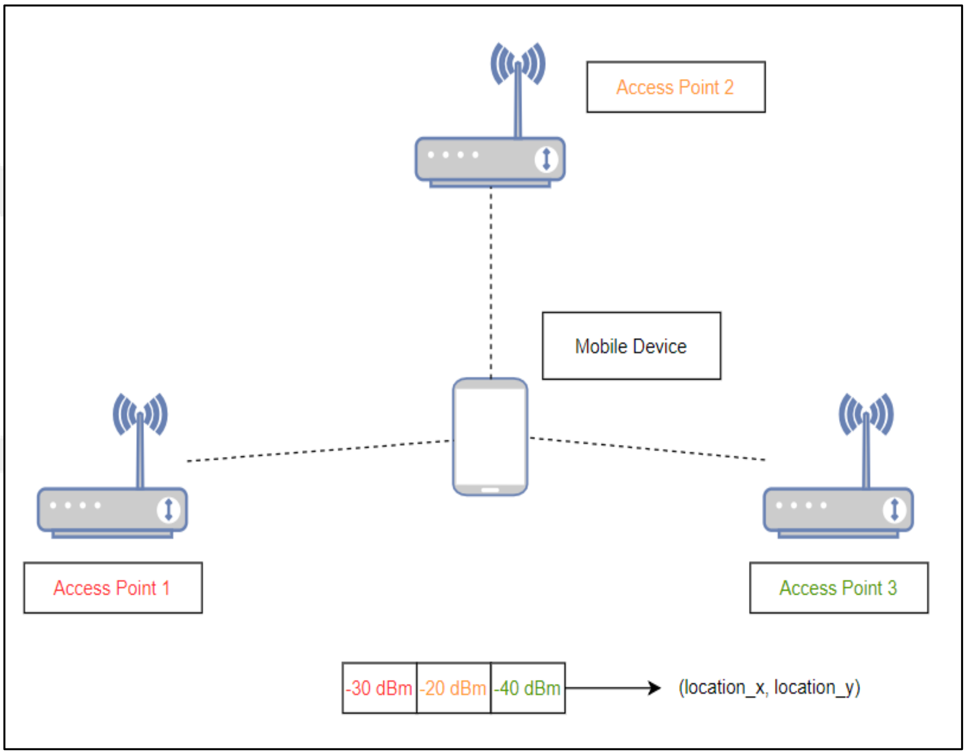

Figure 1.4 Indoor Wi-Fi based localization.

In the recent years there were increase of interest to use Bluetooth low energy (BLE) beacons for tracking and locating objects. The BLE beacon-based positioning methods include two types: fingerprinting-based and range-based. BLE beacons range is about (15 m), which is significantly wider by comparing it with RFID sensor. Utilizing RSSI is recommended to help in positioning. Since, the distance between both the sender end and receiver end decreases, the RSSI value decreases. Then, the user's position can be solved by trilateration according to the distances estimated accordingly (Zuo et al., 2018). Figure 1.5 shows Bluetooth low energy (BLE) beacon.

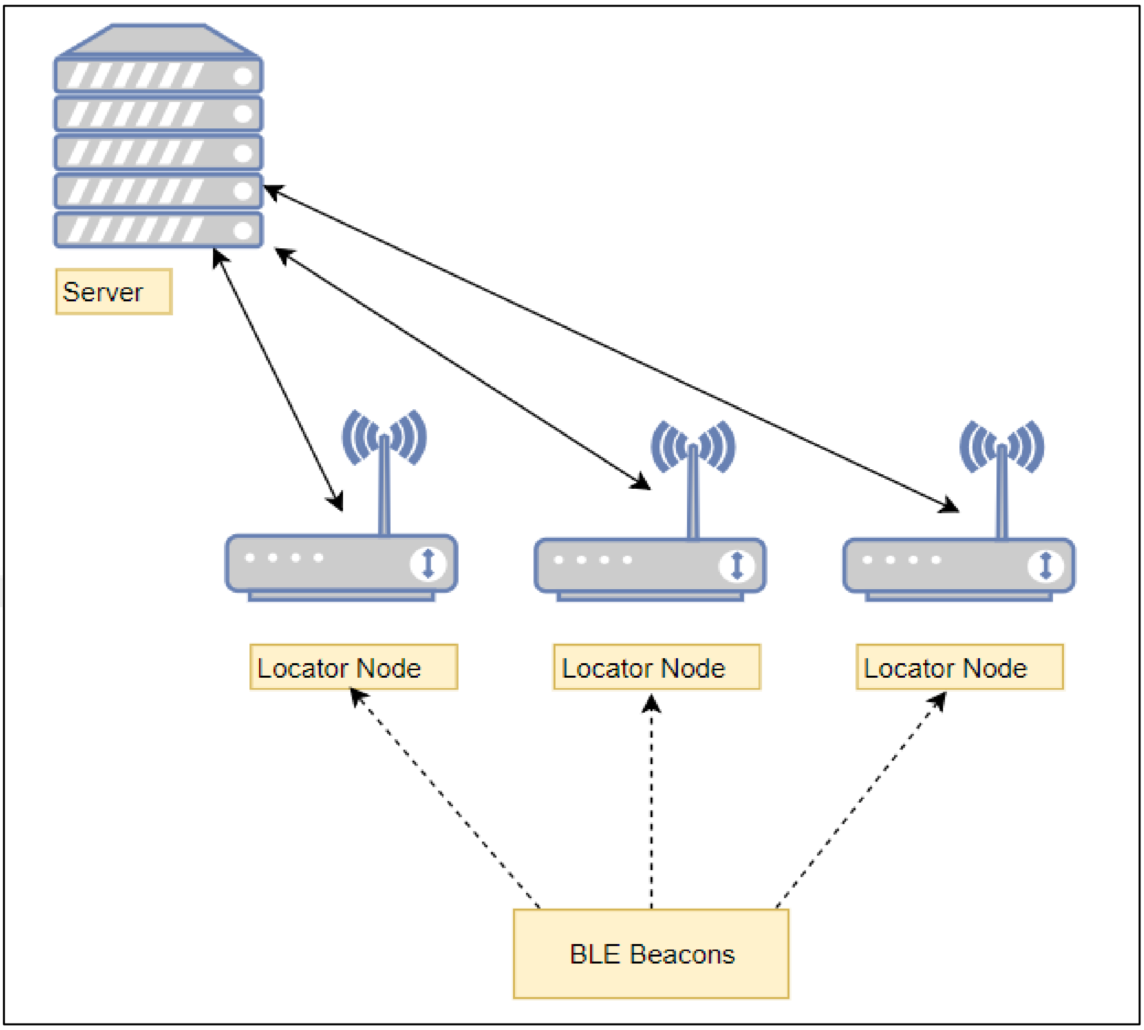

Figure 1.5 Server-based Indoor Positioning using BLE.

Several systems have utilized global system of mobile/code division multiple access mobile cellular network to estimate the position of outdoor mobile users. In term of accuracy, using cell-ID is quite low in range between (50 m to 200 m), according to the cell size. However, the accuracy is higher in densely covered areas. Indoor localization using mobile cellular network is workable if the building is covered by base stations or on base station with strong RSS received by indoor mobile users (Alarifi et al., 2011).

In cellular-based positioning the Global System for Mobile Communications (GSM) are obtainable in most countries that able to outperform the coverage of WLAN, but with lower localization accuracy. The GSM network operates in bands that is licensed and block any interference at a similar frequency. Fingerprinting is a method of GSM indoor localization that is based on the power level (RSS) (Mautz, 2012). The cellular based positioning system is shown in Figure 1.6.

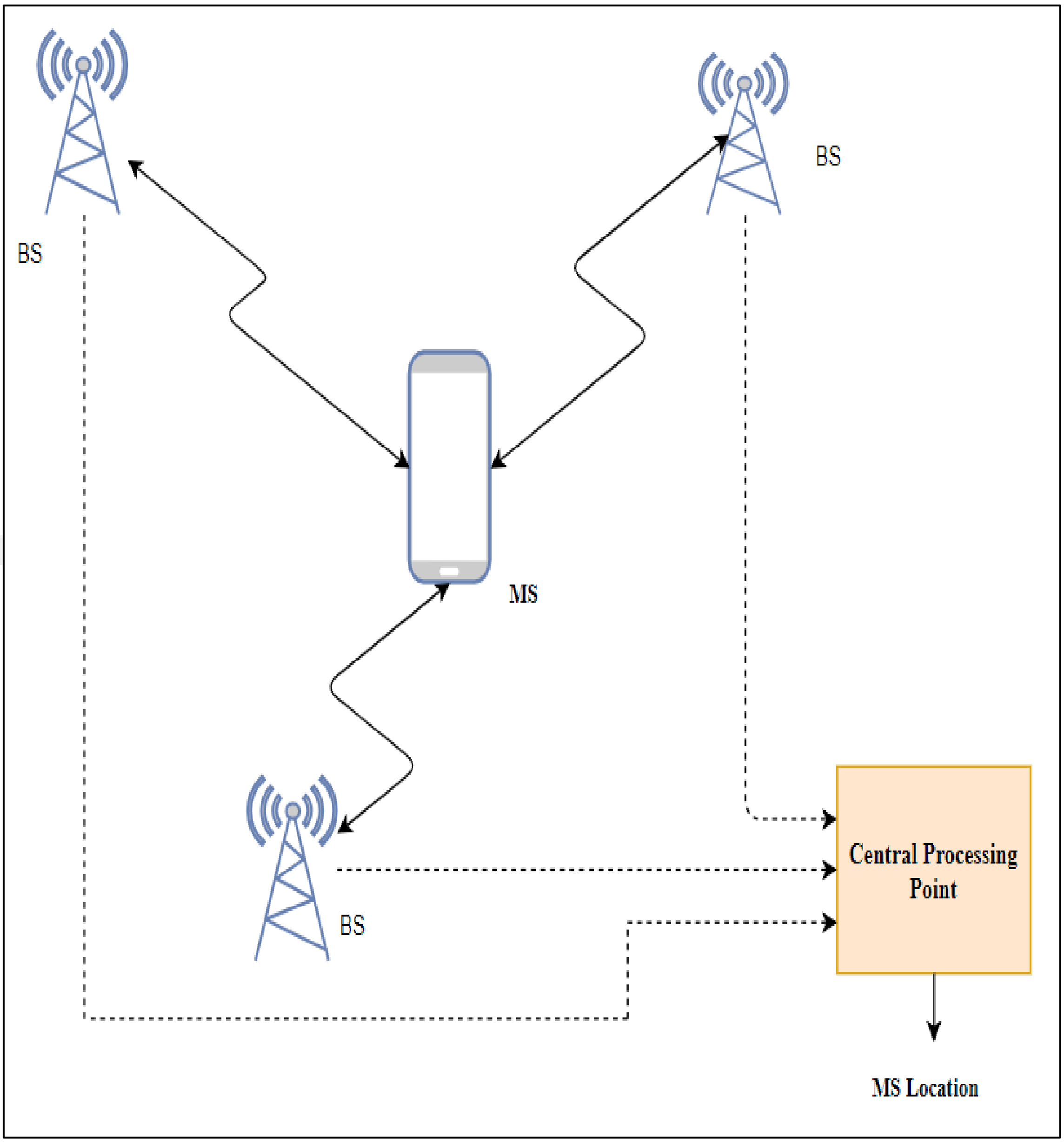

Figure 1.6 Cellular-based Positioning.

Ultra-wide band (UWB) signals have very large bandwidth, which is more than 500 MHz. UWB transmitters allow better power efficiency, because the consumption of power is low in comparison with other indoor positioning technologies. UWB provide excellent multipath resolution, since the indoor wireless system must overcome with sever multipath situations. Such a wide band width offers many advantages for communications and radar applications. In both cases, a large bandwidth improves reliability, since the signal contains different frequency components, so it will increase the probability that at least some of them can go around or through obstacles (Gezici et al., 2005).

UWB is considered to be very auspicious technologies. UWB technology it does not require LOS and also it does not affect by the presence external noise because of to its properties, which are the high bandwidth and signal modulation. UWB became commercially available in 1990. UWB based on transmitting short pulses that utilize techniques causing the spreading of the radio energy with low power spectral density. The high bandwidth of UWB provide high data throughput for communication and the low frequency of UWB pulses will make the signal to pass over barrier such as walls effectively (Ghavami et al., 2006). Figure 1.7 shows the UWB positioning system. Hence, the UWB enable more reliable and accurate positioning.

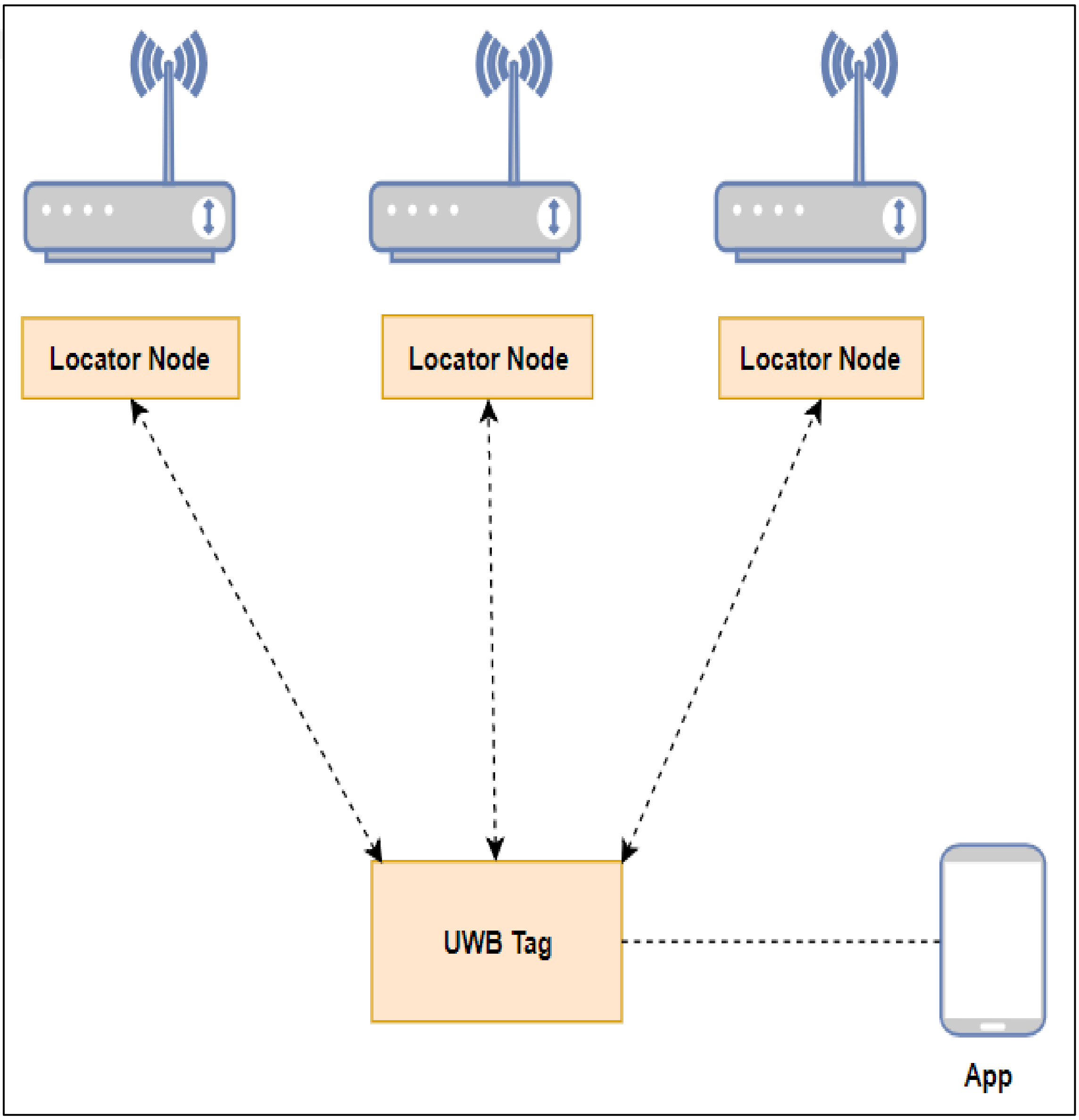

Figure 1.7 UWB positioning system.

1.5 Indoor Positioning Performance Metrics

Measuring the performance of a positioning technology only by its accuracy is not enough. Hence, the performance benchmarking for indoor positioning technology were provided as follow: accuracy, precision, complexity, scalability, robustness, and cost (Tekinay et al., 1998).

1.5.1 Accuracy

Accuracy is important requirement for any indoor positioning systems. The average Euclidean distance between the measured location and the real location is used as performance metric for evaluation purpose.

Accuracy can be systematic effect/offset, or a potential bias of a positioning system. When the accuracy is higher, it refers to good system. However, sometimes there is going to be a compromise between accuracy and some other related characteristics. In which such a compromise is highly needed.

1.5.2 Precision

Location precision reflect the consistently of the system works, thus it is the measure of the robustness of positioning technology as it shows the variation in its performance over many experiments. Whereas, the accuracy considers the value of mean distance errors.

1.5.3 Complexity

The complexity of a positioning system can be referring to software, hardware, and operation factors. For example, if the positioning algorithm computations is running on a centralized server side, then, the positioning calculation can be performed quickly because of the sufficient power supply and the powerful processing capability. However, if it is processed on the mobile unit side, then, the complexity effects could be clearer.

1.5.4 Robustness

The high robustness of a positioning technique means that it could function in normal way even if some signals are not available. Signal from transmitter unit in some cased is blocked, thus the signal can’t be acquired from some of the measuring units, that is the signal from other measuring units is the only information that can be used order to estimate the location.

1.5.5 Scalability

When the positioning scope gets large, the scalability of system will ensure the normal positioning function. In term of positioning performance, it decreased when the distance between the transmitter end and receiver increases.

1.5.6 Cost

In term of the cost of positioning system, it relies on several factors, such as time, money space, energy, and weight. The time factor is referring to both the installation and also the maintenance. Mobile units may have weight constraints and strict space.

1.6 Indoor Positioning System Classification

Indoor positioning technologies can be classified into two categories; first, building dependent and second, building independent. When it come to building dependent indoor positioning, it will indicate the technologies rely on the building in which they operate in. They utilize the existing technology in that building. Furthermore, it can be divided into two classes when it comes to building dependent: indoor positioning system that utilize the buildings and indoor positioning system that require dedicated infrastructure. Whereas, the building independent doesn’t require an existing infrastructure in order to operate. Figure 1.8 shows the classification of indoor positioning system technologies (Alarifi et al., 2016).

1.7 Comparison of Indoor Positioning Technologies

Table 1.1 characterizes the sensor technologies according to its accuracy, coverage, the measuring principle, and its application.

Table 1.1 Comparison of indoor positioning technologies.

Indoor Technology	**Accuracy**	**Measuring Principle**	**Application**
Infrared	1 cm to 5 m	thermal imaging, active beacons	people detection, tracking
WLAN / WiFi	20 m to 50 m	fingerprinting	pedestrian navigation, LBS
RFID	1 dm – 50 m	Fingerprinting, Proximity detection	pedestrian navigation
Ultra-wide band	1 cm – 50 m	time of arrival, body reflection	robotics, automation
GNSS	10 m (global)	assistant GPS, parallel correlation	location based services
Pseudolites	10 cm – 1000 dm	carrier phase ranging	GNSS challenged pit mines

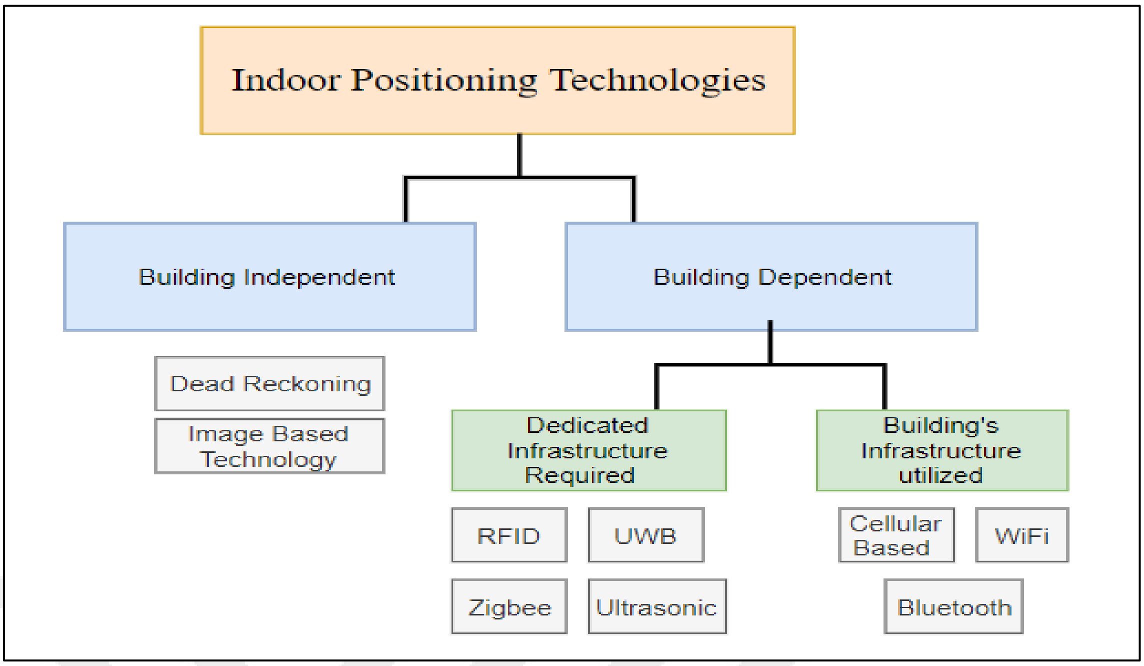

Figure 1.8 Indoor positioning system classification.

1.8 UWB Positioning Algorithms

UWB positioning can be categorized into Received Signal Strength (RSS) based systems, Time of Arrival (ToA), Angle of Arrival (AoA), Time Difference of Arrival (TDOA) and Hybrid-based Algorithms (Alarifi et al., 2016).

1.8.1 RSS-based algorithms

When using RSS-based algorithms, the object that been identified measures the signal power for received signals from numerous transmitters, to estimate the distance between both the transmitters end and receivers end, by using signal strength. Now the receiver end is able to identify its location relative to the transmitter end nodes.
The accuracy of Resaved signal strength for NLOS environment is relatively low, this mean that the RSS is not a suitable identification method for indoor positioning systems despite its advantages. Such as, the mobile tags act as receivers only, hence, it depends on the power of received signals from several transmitters in order to define their location. In this case, the RSS-based method will have lower communication traffic which eventually will help in improving the positioning accuracy (Wang, 2010).

1.8.2 AOA-based algorithms

The estimation of received signal angles, from two sources or more, is been compared with carrier phase in multiple antennas or the signal amplitude. The position is determined from the crossing of the angle line in each signal source. These algorithms are sensitive to number of elements, which can cause errors in their determining of object position. (Al-Jazzar et al., 2011). To increase its accuracy, the AOA is compatible to be used with other algorithms (Reddy and Sujatha, 2011). Figure 1.9 show the AOA-based method.

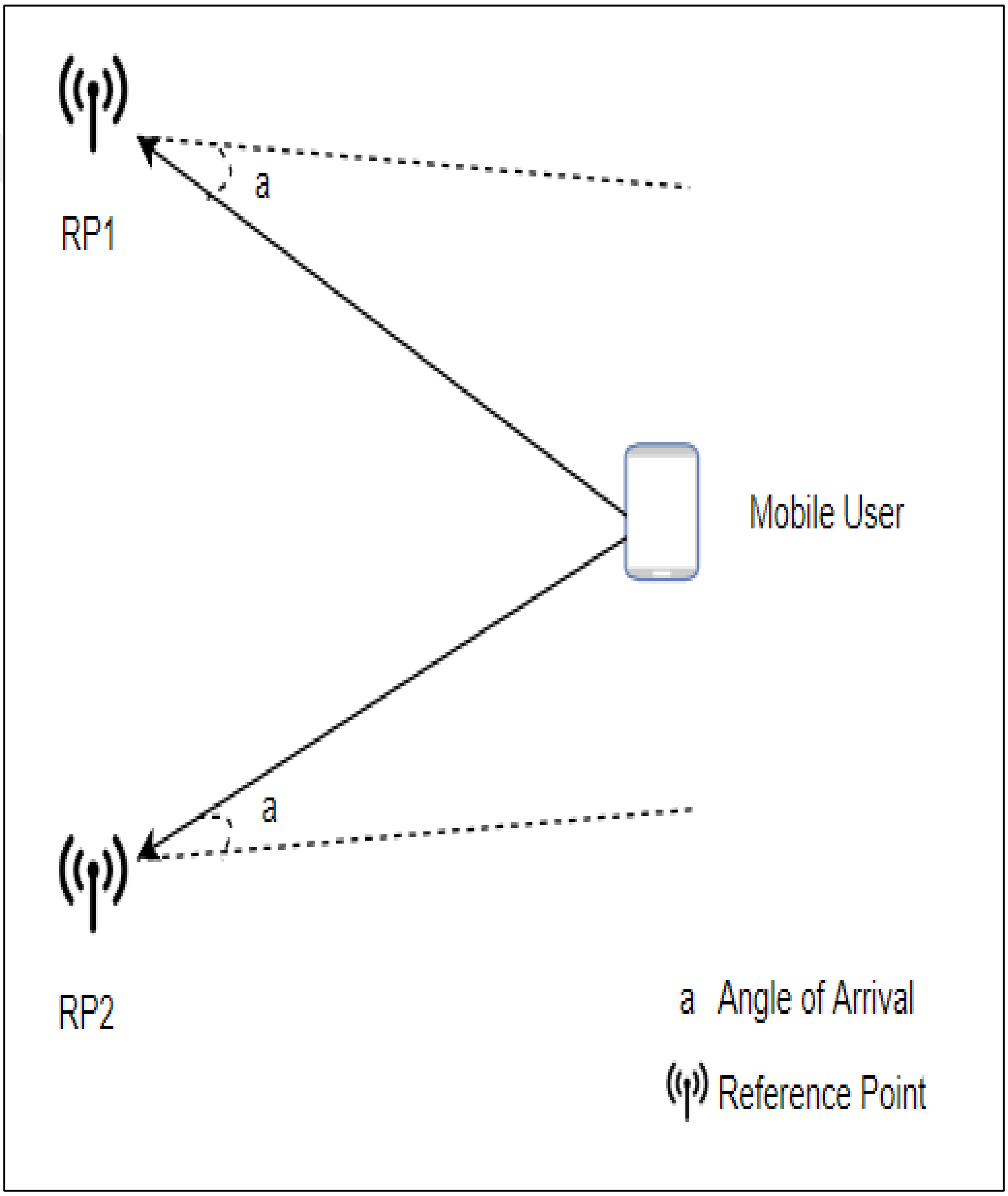

Figure 1.9 Angle of Arrival (AOA)-based method.

1.8.3 TOA-based algorithms

TOA Algorithms are based on the crossing of circles for number of transmitters. In which, the diameter of circles is the distance between both the transmitter end and receiver end. This distance is acquired by the calculation of the one-way propagation time between them (Reddy and Sujatha, 2011). Figure 1.10 shows ToA-based method.

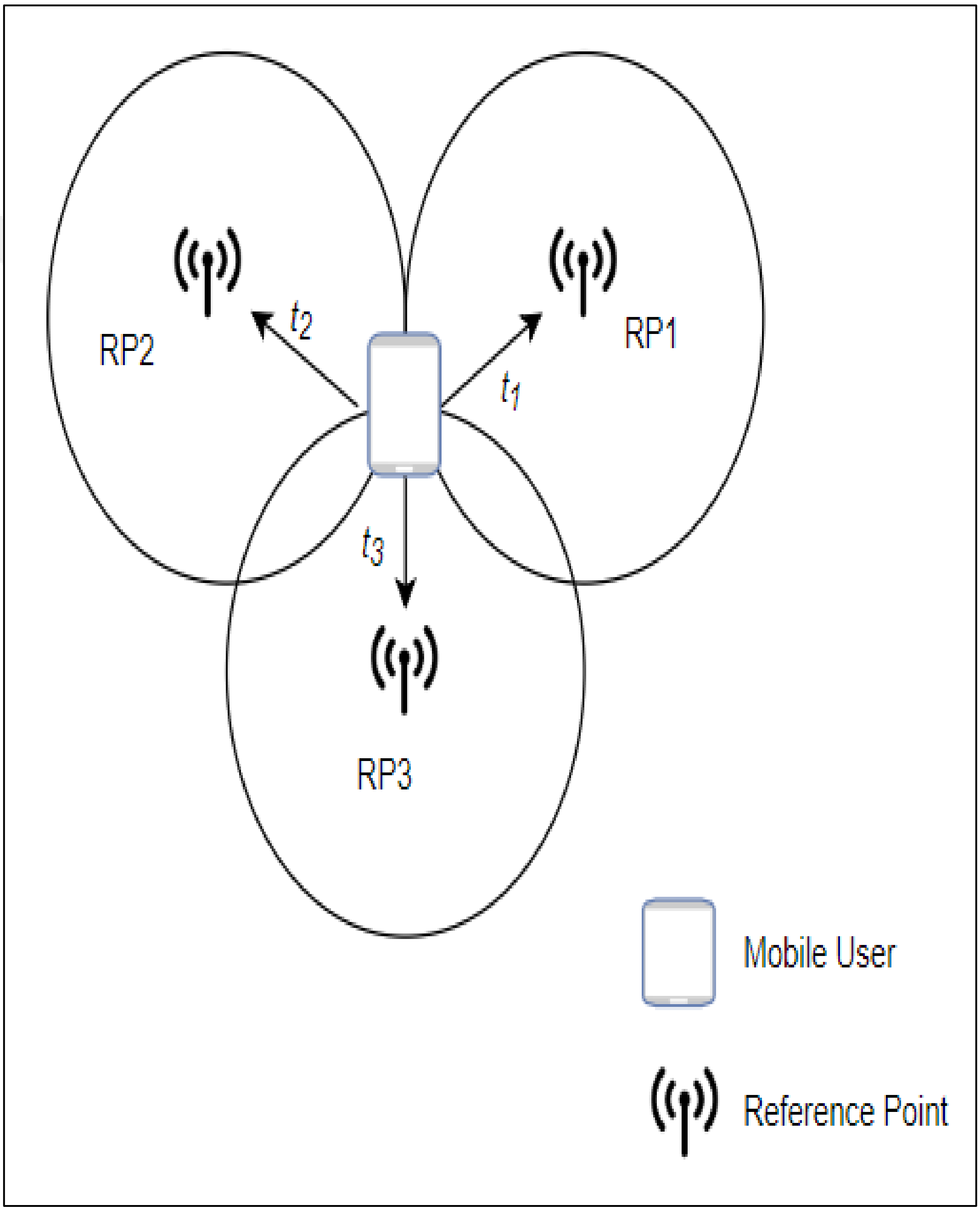

Figure 1.10 Time of Arrival (ToA)-based method.

1.8.4 TDOA-based algorithms

It measures the time difference of arrival of a signal that is been sent by target and then received by more than two receivers. In this scenario, the position of the transmitter end will be found. The scenario can be altered so a single receiver end can determine the object position by measuring the delta in arrival times of two transmitted signals. Usually, one transmitter end requires the multiple receivers end to work together to determine the position of the transmitter and share the data. This requires high bandwidth when compared with other algorithms (Alarifi et al., 2016). Figure 1.11 shows TDOA-based method.

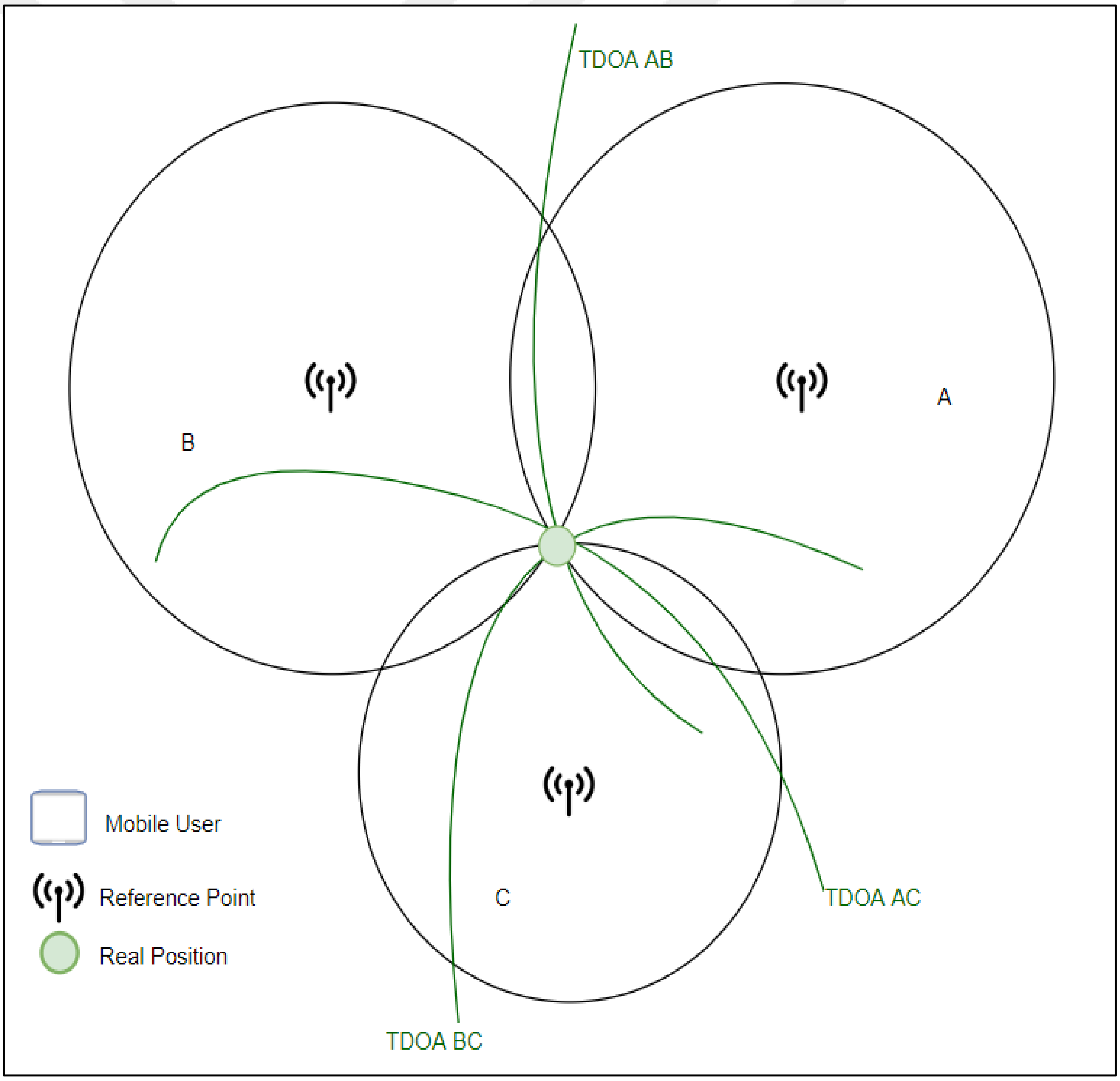

Figure 1.11 TDOA-based method.

1.8.5 Hybrid-based algorithms

Multiple localization techniques are utilized in a way that complement each other, or when multiple positioning techniques aim at multiple parts of the site that adequate with their capabilities. In this manner, the accuracy will increase as well as cost and complexity (Jiang et al., 2010).

1.9 Related Work

The performance of RSS algorithms is investigated for positioning using UWB technology and also explore the effect of small scale fading on the system accuracy (Gigl et al., 2007).

Bekkali et al. (2007) present an algorithm for detrmining the location of the tag by using the multi-lateration with RFID map-based technique and enhance the position estimation of the tag by using Kalman Filter.

The advantages and drawbacks of the TDoA method were analyzed. In which different simulations were presented to show the position errors of TDoA method according to time synchronization errors and anchor and clock errors (Syed Ahmed and Yonghong Zeng, 2017).

A method was proposed by Mahfouz et al. (2014) to combine machine learning with Kalman Filter to estimate instantaneous positions of a moving object. The application of this method can obtain the accurate estimation of position and the accelerations.

An indoor positioning system using BLE beacons was developed. The Big Bang – Big Crunch (BB-BC) method were applied to the experimental indoor positioning system with aim to average locational error. As a result, accuracy increased from 26.62% to 75.69% (Arsan, 2018, J).

Using Ultra-wide Band (UWB) sensors, an indoor positioning system was developed, and the purpose was to increase the accuracy level of the standard equipment. The BB-BC optimization algorithm was implemented to achieve that, by reducing the average location error. As a result, the average error was reduced by 27.51 % (Arsan, 2018, D).

Sunantasaengtong and Chivapreecha (2014) proposed algorithm to apply K-Means clustering and Genetic Algorithm (GA) as engine to prepare offline information. As a result, the accuracy was increased and decrease the computational cost of fingerprint technique for indoor positioning.

Hao Zhou and Nguyen Ngoc Van (2014) address the problem of GPS poor accuracy in indoor environments, they presented radio frequency (RF) based system in order to locate users inside buildings. However, this approach is sensitive to body movements and multipath. This means it will cost much computation time, hence, the Fuzzy C-Means (FCM) clustering algorithm were proposed to lower the computation time. As a result of such implementation, the computation time were reduced, and the accuracy were improved.

The indoor wireless position algorithm based on Wi-Fi K-Means was proposed by Zhong et al. (2016). The improved formula is utilized to consider the effect of attribute values, and the difference between different objects which can be computed accurately.

Suroso et al. (2011) proposes a technique using Fuzzy C-Means (FCM) clustering algorithm, this technique is in Radio Frequency (RF) fingerprint-base indoor positioning. Using such a technique offer positioning system that is capable to provide benefit in low power consumption and in time efficient.

Alata et al. (2008) used a subtractive clustering method to find the optimal number of clusters for the fuzzy C-Means algorithm. They optimize the parameters of the subtractive clustering algorithm by using iteration-based search approach in order to find weighting exponent to the fuzzy C-Means algorithm. The iteration-based search is used to find the optimal single-output Sugeno-type Fuzzy Inference System (FIS) model by optimizing the parameters of the subtractive clustering method that in return provide the a minimized least square error, that is between the real dataset and also the Sugeno fuzzy model.

Yesilbudak (2016) present similarity analysis, by utilizing K-Mean algorithm and Squared Euclidian. Silhouette coefficient value was utilized to check how well-separated the outcome clusters.

Paivinen and Gronfors (2006) study the problem of selecting the right number of clusters. k-Means clustering methods were used, whereas the number of clusters was determined with the largest average silhouette width. As a result, they were able to automatically find the optimal number of clusters from the given dataset without needing to use any user-defined parameters.

Tuncer (2017) present the Intelligent Centroid Localization (ICL) Method. This method is conversion of Centroid Localization (CL). The goal is to determine the position of a sensor position. The RSSI values are used as an input to the fuzzy system and the values of fuzzy system's produced membership functions tuned by performing Genetic Algorithm (GA) to minimize the average location error. In returned, the location error reduced by 65% and 57% and when it was compared with Approximate Point In Triangle (APIT) algorithm and Centroid Localization method.

1.10 Structure of the Thesis

This introduction is going to be followed by an overview of the proposed methods that were implemented in this work in Chapter 2. Which include the optimization algorithms, machine learning algorithms, Kalman Filter and additional tool to define the number of clusters in the clustering algorithms. The experimental setup that were applied to collect the dataset as well as the implementation and evaluation of the optimization methods are in Chapter 3. Whereas the implementation and evaluation of machine learning algorithms and the Hybrid Big bang-big crunch K-Means algorithms are in Chapter 4.

Chapter 5 closes this thesis with conclusions drawn from the work that presented throughout the thesis, and suggestions for future work to develop the current implemented work.

2. PROPOSED METHODS

2.1 Optimization Methods

The optimization problem is finding a set of parameters which minimizes an objective function, it can also consider as fitness function in the evolutionary algorithms (Erol and Eksin, 2006).

2.1.1 Big bang-big crunch algorithm

BB-BC is essentially consisting of two stages: a big bang (BB) stage and a big crunch (BC) stage. In BB stage, candidate solutions will be distributed uniformly over the search space with respect to the limit of the search space. The BC stage can be visualized as transformation from disordered state of energy to ordered state of energy (Erol and Eksin, 2006). The big crunch phase can be visualized as transformation from disordered state of energy to ordered state of energy. The BC has multiple inputs and one output, namely, center of 'mass'. The BC is a concurrence operator and the word 'mass' indicate the inverse of the objective function value. The center of mass is calculated as follow (Biradar and Hote, 2016):

$$\vec{x}^c = \frac{\sum_{i=1}^{N} \frac{\vec{x}^i}{f_i}}{\sum_{i=1}^{N} \frac{1}{f_i}} \tag{2.1}$$

where x_i, f_i, N is a point within an n-dimensional search space, the objective function value of this point, the population size, respectively. After the BC stage, the optimization algorithm creates new members to be used in BB stage in the next iteration. This process can be achieved by jumping to the first step and generate an initial population.

For an optimization algorithm to be classified as global, it must converge to an optimal point; However, it must include certain points that have a decreasing probability within its search population. So, the large amount of the solutions that is been produced must be around the optimal point, however, the few points that is remaining are distributed within the search space after a fixed number of steps. As the number of iterations increases, the ratio of solution points around the optimal value to points away from optimal value must decrease. The center of mass can be utilized by spreading new off springs around it.

After that, the center of mass will be recomputed. These contraction steps are keep performed until a specified stopping rule. The new candidates around the center of mass are calculated as follow (Labbi and Attous, 2010). The new candidates around the center of mass are calculated as follow (Erol and Eksin, 2006):

$$x^{new} = x^{c} + \frac{L\,r}{k} \tag{2.2}$$

where x^c, r, L, k is center of mass, normal random number, the upper bounds on the values of the optimization problem variables, the iteration step, respectively. As iterations go to infinity, the deviation term will reach zero, hence there will be always off-springs located far from the center of mass with probability that is decreasing but will never equal to zero. This will assure the global convergence of the algorithm. Figure 2.1 shows the flow chart of BB-BC algorithm. The BB-BC algorithm can is summarized in Figure 2.2.

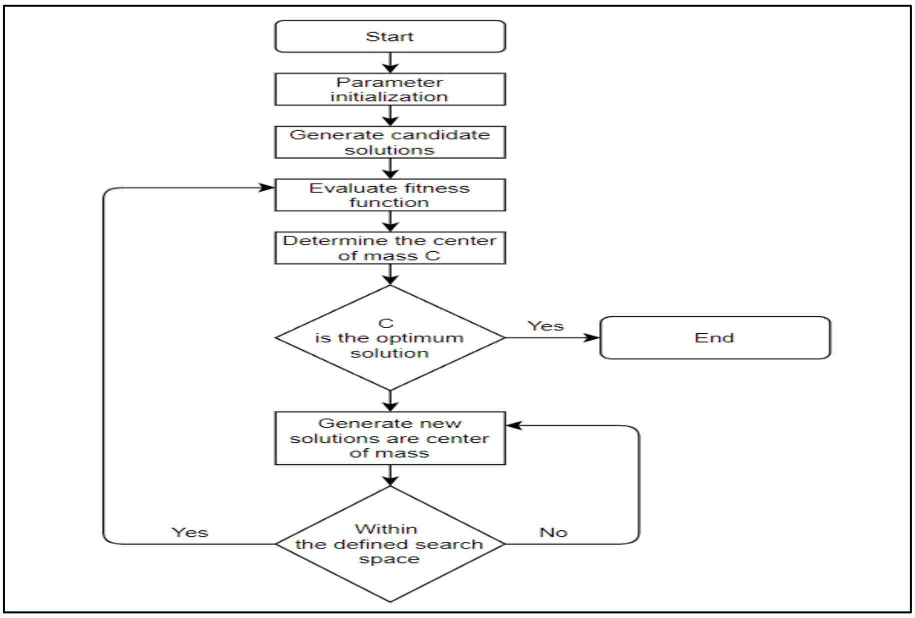

Figure 2.1 BB-BC algorithm flow chart.

Step 1: Initialize:

r: Normal random number,

N: Population size,

UB: Upper Boundary,

$iter$= 1,

Max iteration: Maximum number of iterations.

Step 2: Generate population X_i of size N with respect to the defined limits.

Step 3: For each candidate evaluate the fitness function.

Step 4: Calculate the center of mass C, using Eq. (2.1).

Step 5: Generate new solutions around the center of mass using Eq. (2.2).

Step 6: $iter \leftarrow iter+1$

Step 7: Return to step number 3, until stopping criteria is been met, which is (iter=max iteration).

Figure 2.2 BB-BC algorithm pseudo code.

2.1.2 Genetic algorithm

GA is a heuristic algorithm which can be applied in a straightforward manner. GA is implemented in a wide spread of problems. Due to their population approach, GA have been extended to solve search and optimization problems efficiently, that including multi-objective and multimodal (EL- Sawy et al., 2014).

GA based on genetics and biological evolution. In GA, the design variables are represented as genes on a chromosome. It features a group of candidate individuals, that is called population on the response surface.

Because of its genetic operators and environmental selection, mutation and recombination, chromosomes that have an optimal fitness are obtained (Deb, 1991).

In the 1960s, genetic algorithm was invented by "John Holland" and it was later developed by Holland and colleagues and his students at Michigan University. GA comprises by these four important steps (Michalewicz, 1996):

(i) The initial candidate population of chromosomes are formed by two way, in random way or by perturbing an input chromosome. The way the initialization step is done is not critical if the initial population extent a wide range of design variable settings. Hence, if there is a knowledge about the system being that is been optimized, then, this information can be adopted in the initial population. In the binary representation, every chromosome is a string zeros or ones. The length of the string depends on the required accuracy.

(ii) Evaluation were the fitness is computed in this step. The fitness function aims to numerically encode the performance of the chromosome. In real world applications, the selection of the fitness function is considered a critical step.

(iii) Then, the chromosomes with the highest scores when it come to the fitness, are placed once or more times into a mating pool subset. This placement is in semi-random manner. The low fitness chromosomes are removed from the population.

(iv) Exploration, which include the crossover and mutation operators. Two chromosomes are selected randomly from the mating pool subset to be mated. The probability that these parents are mated is initialized to high value usually, and also its user-controlled option. If the parent chromosomes can mate, then, a crossover operator is utilized to exchange the genes between the two parents to output two offspring. If they cannot mate, then, the parents are copied into the next generation unchanged.

Figure 2.3 shows a flowchart of GA working (Sunantasaengtong and Chivapreecha, 2014). GA algorithm is summarized in Figure 2.4.

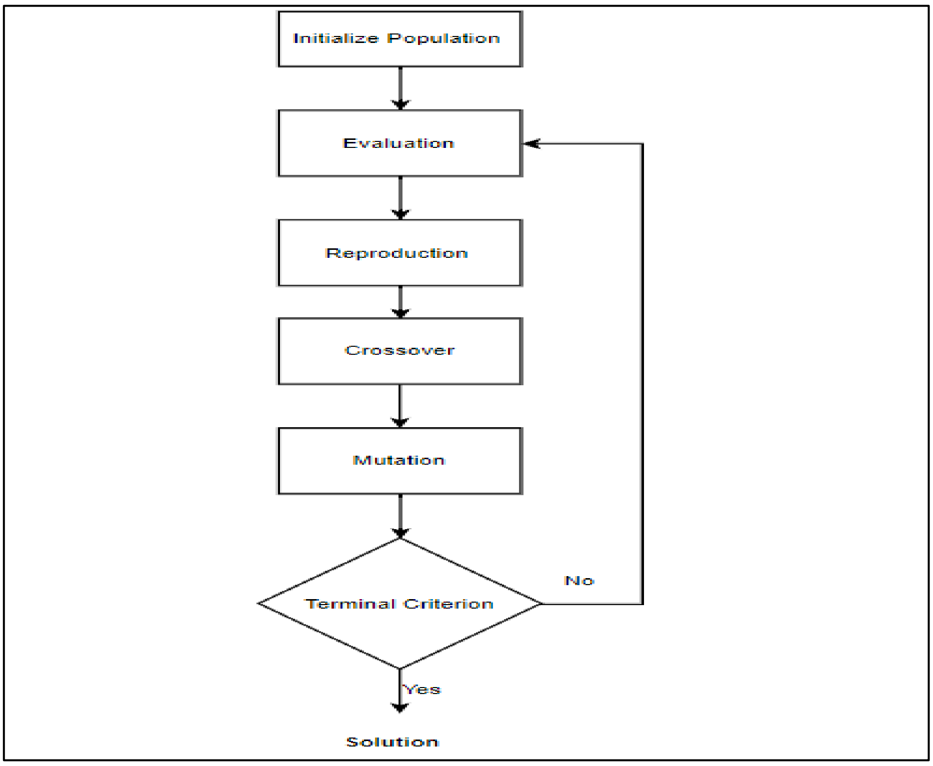

Figure 2.3 Genetic Algorithm flow chart.

```
Input:
        nP: Population Size,
        nVar: Number of Variables,
        Initial Population rang,
        mG: Max Generation.
Output: The best individual in all generation.
Step1: generate initial population of size nP.
 while (Number of generations is less than mG).
        Step 2: evaluate the initial population according to the fitness
        function.
        Step 3: select the individual according to their fitness(selection).
        Step 4: Do Crossover with Pc probability.
        Step 5: Do Mutation with Pm probability.
        Step 6: Update population (population=selected individual after
        Step 4 and 5).
End while
Step 7: Return the best individual.
```

Figure 2.4 Genetic Algorithm pseudocode.

2.2 Machine Learning Algorithms

In this work, centroid-based clustering model was used, since it's the most appropriate for the UWB data set. Three Clustering algorithms are proposed, K-Means, Fuzzy C-Means (FCM), and Mean Shift algorithms.

2.2.1 K-Means algorithm

K-Means clustering algorithm is considering to be one of the important clustering methods. K-Means algorithm randomly select k initial number of centroids (centers), where k is the total number clusters that is defined by the user. Then each point is assigned to a closest cluster center. According to points in the cluster the centroid gets Updated. The process continues till points stop changing their clusters. (Shedthi et al., 2017) formally, the aim of the algorithm is to partition the *n* entities into *k* sets *Si* where, *i =1, 2… k* , so that the within-cluster sum of squares (WCSS) is minimized, defined as :

$$\sum_{j=1}^{k}\sum_{i=1}^{n}\left\|X_i^j - c_j\right\|^2 \tag{2.3}$$

Where, the term $\left\| x_i^j - C_j \right\|^2$ provides the distance between cluster's centroid and an entity point. K-Means algorithm flow chart is shown in Figure 2.5. The algorithm is composed of the following steps (Shedthi et al., 2017):

(i) Selecting the number of clusters i.e. K.
(ii) Choosing Randomly N cluster centroids.
(iii) Calculated the distance between data points and cluster centroids.
(iv) Similar data points which is close to centroid, then move that cluster.
(v) Acquire new cluster centers by averaging the observations in each cluster.
(vi) Steps (iii) to (v) are repeated until cluster centroids do not change or reach the maximum number of iterations.

(Namratha and Prajwala, 2012) the main advantages of K-Means algorithm are: (1) the simplicity; (2) K-Means is computationally faster than hierarchical clustering, which allows it to run on large datasets; (3) if large number of clusters is specified, it can find pure sub clusters. Whereas the disadvantages of K-Means algorithm are: (1) it's difficult to identify the initial clusters; (2) since the number of clusters is fixed at the beginning, the prediction of value of *K* is difficult; (3) the final cluster pattern is dependent on the initial patterns; (4) It does not produce the same result with each run, since the outcome clusters depend on the initial random assignments (Singh et al., 2011).

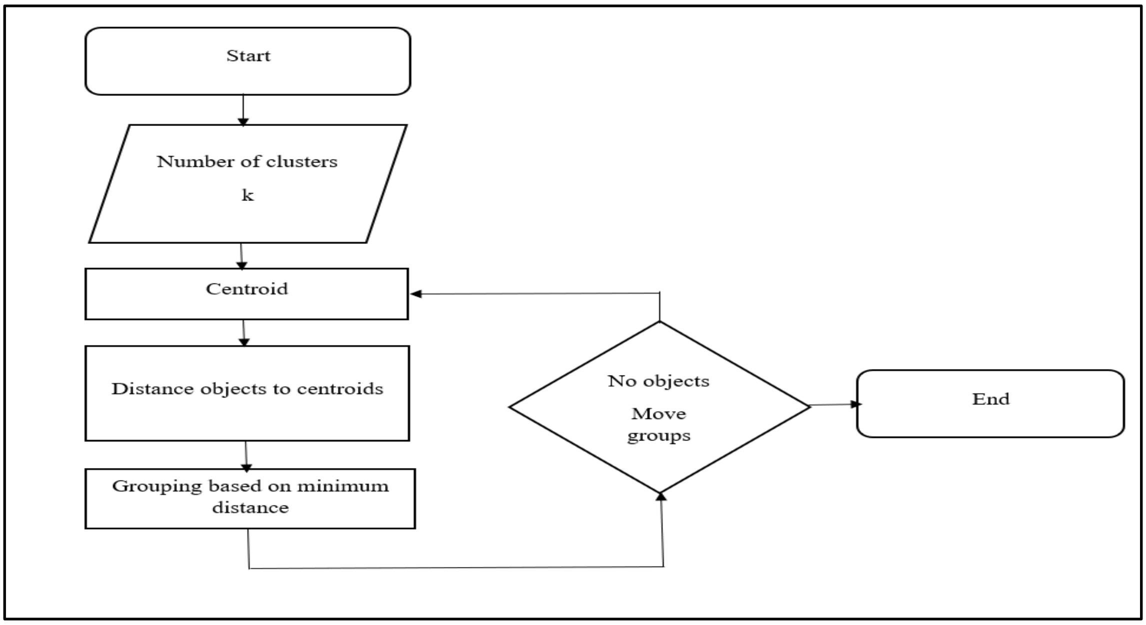

Figure 2.5 K-Means Algorithm flow chart.

2.2.2 Fuzzy C-Mean algorithm

Fuzzy C-Means (FCM) is algorithm for data clustering. Based on fuzzy set theory that allows one piece of data belongs to two or more clusters. Where fuzzy means "unclear" or "not defined" and "C" denotes clustering. The main advantages of this algorithm are its robust behavior, ability of uncertainty data modeling, applicability to multi-channel data, and its straight forward implementation (Suroso et al., 2011). It is based on minimization of the following objective function (Alata et al., 2008):

$$\sum_{i=1}^{N} \sum_{j=1}^{C} u_{ij}^{m} \left\| x_i - C_j \right\|^{2} \tag{2.4}$$

Where m refers to real number greater than 1; u_{ij} refer to the degree of membership of x_i in the cluster j; x_i is the *i*th of d-dimensional measured data; c_j is the *d*-dimension center of the cluster and ||*|| is norm expressing the similarity between any measured data and the center. Fuzzy partitioning is process through an iterative optimization of the objective function shown above, with the update of membership u_{ij} and the c_j cluster centers by:

$$u_{ij} = \frac{1}{\sum_{k=1}^{C} \left(\frac{\left\| x_i - c_j \right\|}{\left\| x_i - c_k \right\|} \right)^{\frac{2}{m-1}}} \tag{2.5}$$

$$c = \frac{\sum_{i=1}^{N} u_{ij}^{m} \cdot x_i}{\sum_{i=1}^{N} u_{ij}^{m}} \tag{2.6}$$

This iteration will stop when

$$max_{ij}\left\{ \left| u_{ij}^{k+1} - u_{ij}^{k} \right| \right\} < \ \varepsilon \tag{2.7}$$

Where ε is a termination criterion between 0 and 1 and *k* are the iteration steps. This procedure converges to a local minimum or a saddle point of *Jm*. Fuzzy C-Means Flow chart is given in Figure 2.6. The algorithm is composed of the following steps:

(i) Initialize U = [u_{ij}] matrix, U (0).

(ii) At k-step: calculate the centers vectors $C(k)=[c_j]$ with $U(k)$ using Eq. (2.6).

(iii) Update $U(k)$, $U(k+1)$ in Eq. (2.5).

(iv) *If* $|| U(k+1) - U(k)|| < \varepsilon$ *then STOP; otherwise return to step (ii).*

The main advantages of FCM algorithms are (Suganya and Shanthi, 2012):

(i) Converges.

(ii) Unsupervised.

The Disadvantages of FCM algorithm are:

(i) The computational time is long.

(ii) Very sensitivity to noise and One expects low (or even no) membership degree for outliers (noisy points).

(iii) Sensitivity to the initial guess (speed, local minima).

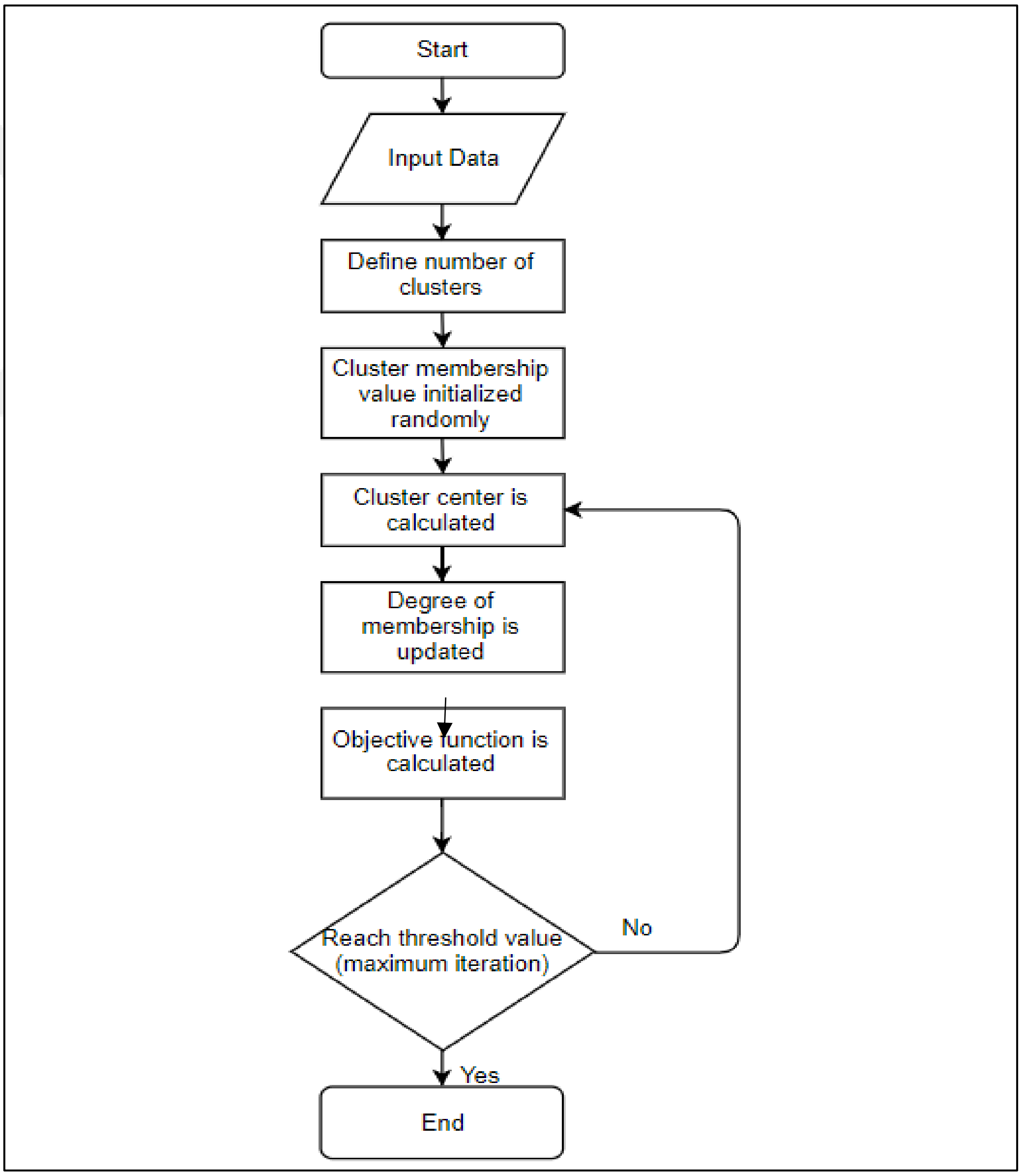

Figure 2.6 Fuzzy C-Means Algorithm flow chart.

2.2.3 Mean Shift algorithm

Mean shift algorithm is based on the general idea that locally averaging data results inmoving to higher density, and therefore more typical, regions (Carreira-Perpiñán, 2015). Mean shift is a nonparametric estimator of density gradient. The local maximum can be gotten by the iterative method. The algorithm now has been widely used, such as clustering analysis, image segmentation, object tracking, discontinuity preserving smoothing, filtering, edge detection, and information fusion. Mean shift algorithm used kernel function to calculate the step of the mean shift and estimate point gradient orientation (Guo et al., 2007). Mean shift algorithm is very attractive because it based on nonparametric Kernel Density Estimates (KDE). In which, the user doesn't need to define the number of clusters. The only parameter the user needs to specify is the scale of the clustering (band width) but not the number of clusters itself. In Mean shift clustering, the input of the algorithm are the data points and the bandwidth or scale. Call $\{x_n\}_{n=1}^{N} \subset \mathbb{R}^D$ the data points to be clustered. The kernel density estimate is defined as follow (Carreira-Perpiñán, 2015):

$$p(x) = \frac{1}{N}\sum_{n=1}^{N} K\left(\left\|\frac{x-x_n}{\sigma}\right\|^2\right) \quad x \in \mathbb{R}^D \tag{2.8}$$

With bandwidth σ > 0 and kernel *K(t), K(t) = e−t/2* for the Gaussian kernel. There are several ways to estimate the bandwidth of a KDE, for example, making the bandwidth proportional to the average distance of each point to its k*th* nearest neighbor.

In term of choosing of kernel, in practice, the Gaussian kernel produces better results than the Epanechnikov kernel, that generates KDEs that are only piecewise differentiable and can contain spurious modes. The results of mean shift were carried over to kernels where each test point has its own weight and its own bandwidth. Gaussian kernels were utilized, since it's easier to analyze and give rise to simpler formulas. Gaussian kernel steps are summarized in Figure 2.7.

The advantages of Mean shift algorithms are listed as follow (Carreira-Perpiñán, 2015):

(i) It doesn't make model assumptions,

(ii) It can model complex clusters having nonconvex shape.

(iii) Only one parameter is needed to set which is the bandwidth.

(iv) The clustering it produce is uniquely determined by the bandwidth, thus, there is no need to run this algorithm with different initializations.

(v) Identify the outliers.

The main Disadvantages Mean shift clustering algorithm are (Carreira-Perpiñán, 2015):

(i) KDEs tend to break down when performing on high dimensions dataset, in which the number of clusters changes abruptly from one for large σ to many, with only a minute decrease in σ. The most successful applications of Mean Shift are in low-dimensional problems.

(ii) In some applications for example, medical image segmentation or figure-ground the user may want a specific number of clusters, but in Mean shift, the user has no control over the number of clusters. Thus, in order to obtain specified number of clusters, the user must search over σ. This is computationally costly and not defined well.

<u>for</u> $n \in \{1, \ldots\ldots, N\}$

$\mathrm{x} \leftarrow \mathrm{x_n}$

<u>repeat</u>

$$\forall n{:}\, p(n|\mathrm{x}) \leftarrow \frac{\exp\left(-\frac{1}{2}\left\|\frac{(\mathrm{x}-\mathrm{x}_n)}{\sigma}\right\|^{2}\right)}{\sum_{n`=1}^{N} exp\left(-\frac{1}{2}\left\|\frac{(\mathrm{x}-\mathrm{x_{n`}})}{\sigma}\right\|^{2}\right)}$$

$$\mathrm{x} \leftarrow \sum_{n=1}^{N} p(n|\mathrm{x})\mathrm{x_n}$$

<u>until</u> stop

$z_n \leftarrow \mathrm{x}$

<u>end</u>

connected-components $(\{z_n\}_{n=1}^{N}, \varepsilon)$

Figure 2.7 Gaussian Mean Shift algorithm.

2.3 Kalman Filter

Kalman Filter algorithm uses a series of data that is observed over time, that may contain noise, with the aim to estimates unknown variables with better accuracy (Li et al., 2015). It was firstly proposed by (Kalman, 1960), then Kalman Filter become a standard approach to achieve optimal estimation. Kalman Filter is considered as one of the famous

Bayesian filter theories (Woods and Radewan, 1977). The Status equation and observation equation is a linear representation of w_k, u_{k-1}, x_{k-1} and x_k, v_k, respectively. Status equation and observation equation represent a dynamic model by the reliable estimation corrected by measurements (Salmond, 2011). The status equation of Kalman Filter is represented as follow (Li et al., 2015):

$$x_k = Ax_{k-1} + Bu_{k-1} + w_k \tag{2.9}$$

Whereas the observation equation is represented as follow:

$$z_k = Hx_k + v_k \tag{2.10}$$

where in the above equations: A, x_k, H, w_k, z_k, v_k, u_{k-1} is the transition matrix, status vector, the matrix of observation, noise vector of the system, observation vector, noise vector of the observation , system control vector, respectively.The w_k and v_k are supposed to satisfy the positive definite, uncorrelated and symmetric, zero mean Gaussian white noise vector; k is a subscript; w_k and v_k are satisfied:

$$E(w) = 0, cov(w) = E(ww^T) = Q \tag{2.11}$$

$$E(v) = 0, c0v(v) = E(vv^T) = R, E(wv^T) = 0 \tag{2.12}$$

$x_k^{\wedge -} \in R^n$ is the prior status estimation which is derived from status transition equation at the moment of k-1, where $\hat{x}_k$ is the posterior status estimation that combines the measurements at the moment of k. The deviations are in following Eq. (2.13) and Eq. (2.14):

$$e_k^- = x_k - x_k^{\wedge -} \tag{2.13}$$

$$e_k = x_k - \hat{x}_k \tag{2.14}$$

The priori and posterior estimation deviation covariance equations are defined in Eq. (2.15) and Eq. (2.16) :

$$P_k^- = E\ [e_k^- e_k^{-T}] \tag{2.15}$$

$$P_k = E\ [e_k e_k^T] \tag{2.16}$$

The following prediction and update equations are obtained from the Kalman Filter theory. Prediction equations are defined as follows:

$$\hat{x}_k^- = A\hat{x}_{k-1} + Bu_{k-1} \tag{2.17}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2.18}$$

Update equations are defined as follows:

$$K_k = P_k^- H^T (HP_k^- H^T + R\ )^{-1} \tag{2.19}$$

$$\hat{x}_{k-1} = \hat{x}_k^- + K_k(z_k - H\hat{x}_k^-) \tag{2.20}$$

$$P_k = (I - K_k H)P_k^- \tag{2.21}$$

Where K_k, $\hat{x}_k$, P_k, I are the Kalman gain matrix, optimum filter value, the matrix of filter deviation, unit matrix, respectively. Figure 2.8 shows the Kalman Filter in pseudocode, and Figure 2.9 shows the flow chart of Kalman Filter.

```
Input: Q, R, z, x_est, p_est
Output: s_t^-, P_t^-
Step 1: Initialize A matrix and H matrix.
Step 2: Predicted state vector and covariance:
        x_prd = A * x_est
        p_prd = A * p_est * A' + Q
Step 3: Estimation:
        S = H * p_prd' * H' + R
        B = H * p_prd'
Step 3: Compute Kalman gain factor
        klm_gain = (S \ B)'
Step 4: Correction based on observation:
        s_t^-  = x_prd + klm_gain * (z - H * x_prd)
        P_t^-  = p_prd - klm_gain * H * p_prd
Step 5: return s_t^-, P_t^-
```

Figure 2.8 Kalman Filter algorithm in pseudocode.

2.4 The Average Silhouette Method

The average silhouette is a way for defining the optimal number of clusters. It measures the quality of a clustering. That is, it determines how well each object lies within its cluster. The silhouette ranges from −1 to +1, where a high value indicates a good clustering. The closer silhouette coefficient to 1, the higher the observation belongs to its cluster (Yesilbudak, 2016) .If a_i is the average dissimilarity between the *i*th data point and all other points in the cluster, and $b_i(k)$ is the average distance from the *i*th point to points in another cluster k, then the silhouette coefficient of the *i*th data point is (Paivinen and Gronfors, 2006):

$$s_i = \frac{min_k b_i(k) - a_i}{max(a_i, min_k\, b_i(k))} \tag{2.22}$$

The average silhouette method can be computed as follow:

(i) Compute clustering algorithm (e.g., K-Means clustering or Fuzzy C-Means) for different values of k.

(ii) For each k, calculate the average silhouette of observations.

(iii) The location of the maximum is considered as the appropriate number of clusters.

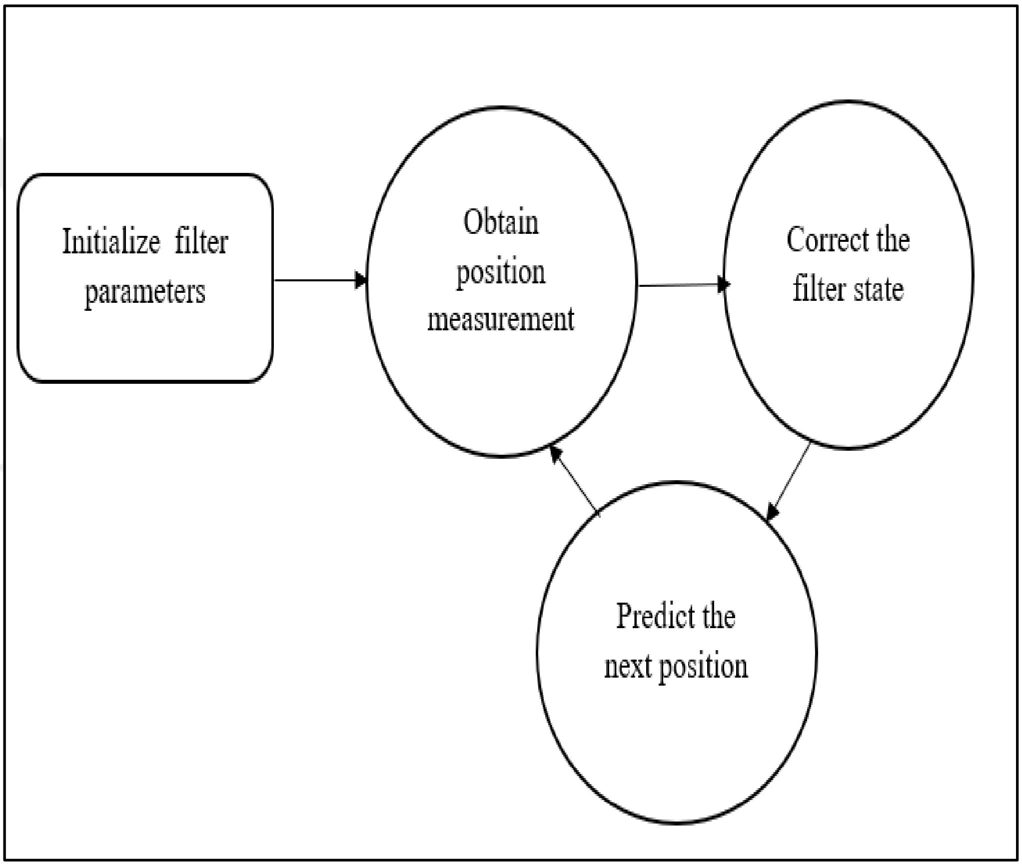

Figure 2.9 Flow chart of Kalman Filter.

3. EXPERIMENTAL SETUP, WORK AND EVALUATION OF THE OPTIMIZATION ALGORITHMS

3.1 Experimental Setup

In this work, the dataset that was collected from an active learning classroom (ALC). The classroom contains a moveable tables, chairs, and desks, so it will provide multiple choices for seating. The class is limited to 28 students, and the area is designed to provide maximum control to the users. Total of 12 student's setup is used when the dataset is collected as shown in Figure 3.1. The design features are expected to support users' use of all locations in the classroom while performing different activities.

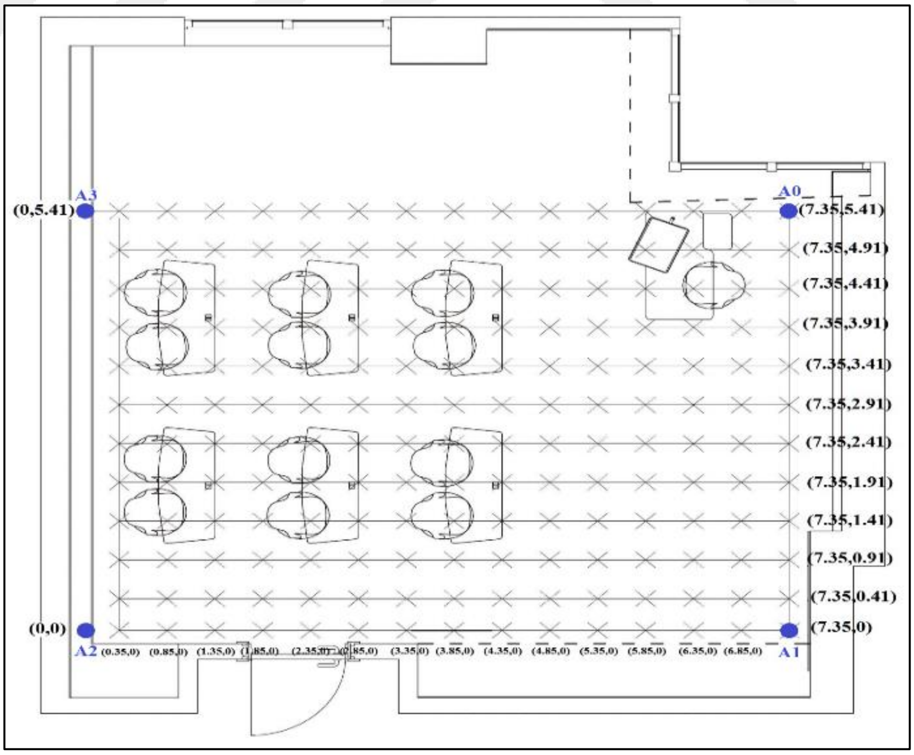

Figure 3.1 Active learning classroom, measuring 7.35 m x 5.41 m, installation of the four anchors expressed as A0, A1, A2 and A3, the test points expressed as ×.

While the active learning class, measuring 7.35 m x 5.41 m, is designed as a test bed for collecting data, a ceiling system, which is attached to the ceiling and held anchors on exactly corners of the testbed at 2.85 m constant height, is established. As shown in Figure 3.2, a ceiling system is established to provide better LOS and direct path between both the tags and the anchors.

Figure 3.2 Ceiling installation of the anchors.

As shown in Figure 3.3, Decawave MDEK 1001 UWB development kit is utilized to conduct this experiment, by including 4 anchors in the established ceiling system and a test tag for the test user.

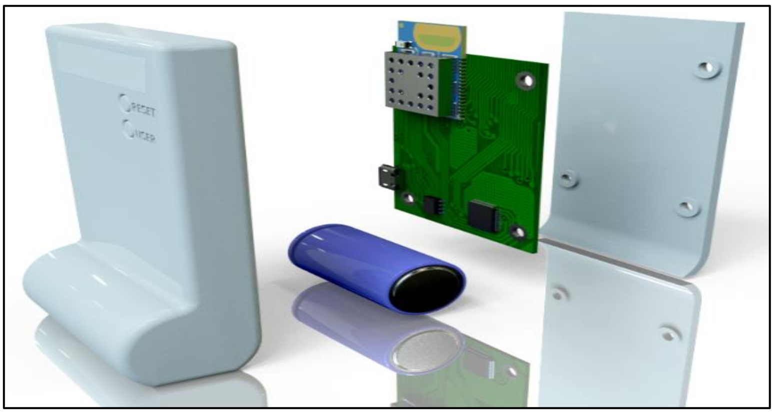

Figure 3.3 A sensor kit of Decawave MDEK1001 development kit which can be assigned as an anchor or a tag.

Total of 180 locations of the test user were marked, the test user was given a UWB sensor tag to wear around his neck and then the location data of the test user were collected. The test user stayed in the testbed for 3 minutes for each location providing 150 samples for each marked location. Total time of data collection is 9 hours excluding the time for set up and change of observation cycles. Total of 27,000 location measurements were collected (Arsan and Kepez, 2017).

The dataset was partitioned randomly into training set and test set. In which the training set include 70% of the samples (105 samples) and test set has 30% of the samples (45 samples). The average location error for the training set is 16.3378 cm, whereas the average location error for the test set is 16.3442 cm.

3.2 Experimental work and Evaluation of the Big Bang-Big Crunch Algorithm

The population size of BB-BC algorithm is set to 100 and the number of generations is also set to 500, which is also refer to maximum number of iterations. The experiments were performed on Intel Core i7 dual core with 4 threads, using MATLAB R2018a. In population-based approaches, it is known that the fitness function value is calculated for every candidate solution in each population, thus, it has a great impact on the algorithm's speed. The fitness function used in BB-BC algorithm as follow:

$$\sum_{i=1}^{N}((x_{im} - x_{ioffset} - x_{ir})^{2} + (y_{im} - y_{ioffset} - y_{ir})^{2}) \tag{3.1}$$

Where x_{ir}, y_{ir} represent the real location values, $x_{ioffset}$, $y_{ioffset}$ represent the required offset values, whereas x_{im} and y_{im} represent the measured UWB values in both x and y dimensions. Figure 3.4 show the flow chart of proposed system for BB-BC algorithm.

3.2.1 BB-BC algorithm standalone simulation

In the first simulation, the raw training set and test set were used as input to the proposed system as shown in Figure 3.4. The goal was to obtain (x_{offset}, y_{offset}) from the training set for each test point of the 180 test points and apply these values to the test set. The x_{offset} *and* y_{offset} values are coordinate dependent. In order to identify which (x_{offset}, y_{offset}) values belong to which test point in the test set, the average for each test *point* (x_{Avg}, y_{Avg}) in both the training set and the test set were calculated. Then, the average of test point (X_{Avg}, Y_{Avg}) in the test set that has nearest distance to the test point (X_{Avg}, Y_{Avg}) in the training set, uses the corresponding (x_{offset}, y_{offset}).

The offset obtained from the training set for the 180 test points are shown in Table 3.1. Figure 3.5 shows the improvement to the measured UWB test points (180 test points) for the test set. As a result, the average location error was reduced by 48.16 % (from 16.34 cm to 8.47 cm) for the test set.

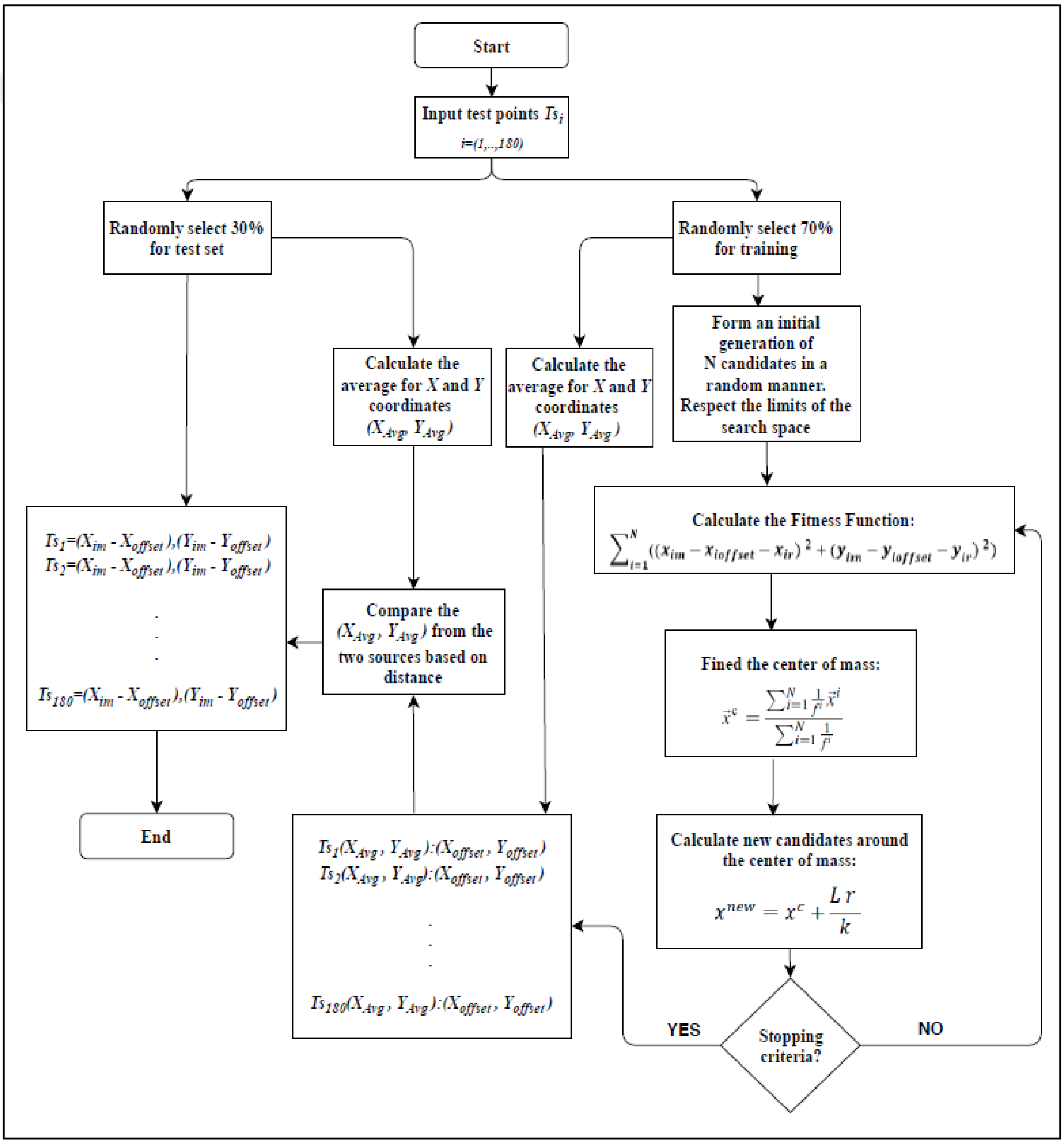

Figure 3.4 Proposed system for BB-BC algorithm.

Table 3.1 BB-BC offset values.

P.	X_{Offset}	Y_{Offset}	P.	X_{Offset}	Y_{Offset}	P.	X_{Offset}	Y_{Offset}
1	0.11534297	0.010691031	**61**	-0.139527798	0.272273972	**121**	0.120746649	0.086233055
2	0.011153233	0.001772856	**62**	0.061191721	-0.001110148	**122**	0.027088403	0.174906851
3	0.033775413	0.093884262	**63**	-0.072289955	-0.047935546	**123**	0.032346032	0.097945255
4	0.144149114	0.059178585	**64**	0.029885692	-0.107334795	**124**	0.058845891	-0.008461347
5	0.013981438	0.014213967	**65**	-0.228620752	-0.004151056	**125**	-0.073543642	0.133662001
6	0.025948707	0.023924808	**66**	-0.099349568	-0.101285281	**126**	-0.166364929	0.268415751
7	0.259778149	0.138007764	**67**	0.039579576	0.052863899	**127**	0.003986926	0.234548896
8	0.102347405	0.026860465	**68**	0.053694768	0.172095205	**128**	0.054993217	0.196720408
9	0.0932238	0.006225316	**69**	0.045984869	0.07215579	**129**	0.077061033	0.131563141
10	0.042170076	-0.470636028	**70**	0.067271314	0.085952028	**130**	0.077100746	0.100399409
11	0.078260469	-0.097363387	**71**	0.043085274	0.234667959	**131**	0.123378515	0.187617083
12	0.031380195	-0.252758446	**72**	0.050485832	0.073681031	**132**	0.048352486	0.155082641
13	0.356756709	-0.283321151	**73**	-0.092181744	0.396648217	**133**	0.16883339	0.396740095
14	0.000223164	-0.014900107	**74**	0.169675341	0.076607523	**134**	0.168720508	0.070474398
15	0.006596666	-0.12677083	**75**	-0.017532096	0.162956859	**135**	0.21630277	0.080254677
16	0.005680077	0.126100947	**76**	0.010432579	0.000292791	**136**	0.009522132	0.066771649
17	0.035955262	0.109745996	**77**	0.046699266	0.161710622	**137**	0.008319361	0.076502414
18	0.066909749	0.096249148	**78**	-0.17681075	0.067118373	**138**	0.012236464	0.201577989
19	0.181632427	-0.003151484	**79**	0.053216976	0.019098514	**139**	0.149440527	0.069453217
20	0.166353152	0.134384152	**80**	-0.145110179	-0.026194614	**140**	0.033438042	0.132771313
21	0.042659778	0.171867204	**81**	0.000171591	0.003023353	**141**	0.048942002	-0.006854175
22	0.058961847	0.057956027	**82**	0.009396841	0.013570732	**142**	0.041941409	0.233981827
23	0.052444442	0.13840581	**83**	0.0825571	0.029256507	**143**	0.061097416	0.078149291
24	0.07571815	0.023856579	**84**	-0.036048786	0.19294221	**144**	0.088670448	0.078599777
25	0.043679699	0.065891964	**85**	0.094266325	0.036783629	**145**	0.123729261	0.118024051
26	0.187804053	0.046776616	**86**	0.225096847	0.208415174	**146**	0.175081242	0.296943416
27	0.11972936	-0.091105251	**87**	0.216674747	0.22340662	**147**	0.200921163	0.155730319
28	0.079104968	0.059200073	**88**	0.220999317	-0.04382997	**148**	0.075428795	0.148993782
29	0.482457858	0.203845528	**89**	0.065456327	0.040766391	**149**	0.197906665	0.124759113
30	0.269062046	0.085777628	**90**	0.120590237	0.074951578	**150**	0.262231689	0.112623284
31	0.129735561	-0.00667473	**91**	0.003217512	0.02231684	**151**	0.018282563	0.029163401
32	0.029478048	0.134656837	**92**	0.000407513	0.041113111	**152**	0.083975607	0.191496341
33	0.040538682	0.002150369	**93**	0.09421124	0.006348759	**153**	0.00488178	0.149765367
34	0.017467886	0.04551327	**94**	0.070943299	-0.019202475	**154**	0.053088322	0.190280798
35	0.068561293	0.117111093	**95**	0.013241994	0.115995122	**155**	0.047600947	0.018212672
36	0.027178287	0.00431424	**96**	0.014105574	0.112738835	**156**	-0.036859424	0.094403495
37	0.048145096	-0.046118851	**97**	0.114189217	0.143408625	**157**	0.294525319	0.04006355
38	0.202234792	0.062478151	**98**	0.071558145	0.095234457	**158**	0.106801534	0.116203942
39	0.114548187	0.033747446	**99**	0.115875578	0.164704283	**159**	0.061716641	0.185538645
40	0.023142137	0.129951515	**100**	0.082685129	0.068549636	**160**	0.02397817	0.160128094
41	0.122544784	0.130915346	**101**	0.223900491	0.103605531	**161**	0.103100588	0.10843004
42	0.15104913	0.00372624	**102**	0.103123895	0.021258083	**162**	0.23222763	0.101299154
43	0.13400872	0.098360107	**103**	0.122297013	0.327171573	**163**	0.086305333	0.139601505
44	0.167912296	0.016906972	**104**	0.007120431	0.226871772	**164**	0.271850434	0.091154014
45	0.00768539	0.079753041	**105**	0.08771891	0.060366461	**165**	0.165481233	0.043669002
46	0.026936784	0.28813043	**106**	0.018636846	0.22053659	**166**	0.136457147	0.072343543
47	0.09095503	0.070559775	**107**	0.124070539	0.078187664	**167**	0.00421834	0.186925912
48	0.070974163	0.079160427	**108**	0.005691673	0.059029838	**168**	0.087019717	0.063383491
49	-0.018657545	-0.097474024	**109**	0.017775324	0.088848095	**169**	0.023338648	0.18871428
50	0.099406882	0.050910016	**110**	0.134837179	0.157019838	**170**	0.022039795	0.184450503
51	0.006706928	-0.011234497	**111**	0.064152053	0.104532165	**171**	0.020224854	0.155884735
52	0.008695962	0.212476437	**112**	0.089159428	0.046067622	**172**	0.137984277	-0.015895096
53	0.202938441	0.06513069	**113**	0.160355296	0.081417845	**173**	0.067649842	0.309365139
54	0.065564774	0.090929739	**114**	0.064413709	0.018205585	**174**	0.156233783	0.048204229
55	0.000726929	0.115047794	**115**	0.003465052	0.087313429	**175**	0.09125539	0.142290503
56	0.16838268	0.125490889	**116**	0.199259056	0.057405928	**176**	0.073300241	0.172366586
57	-0.068283701	0.293799255	**117**	0.048930969	0.221185088	**177**	0.147563607	0.285174929
58	0.221952072	0.016388281	**118**	0.192297252	0.133293202	**178**	0.147437	0.062293595

59	0.151508364	0.009834489	**119**	0.290046354	0.064799622	**179**	0.138933808	0.039111129
60	0.312387507	0.018248686	**120**	0.1893541	0.166586164	**180**	0.046140047	0.019041453

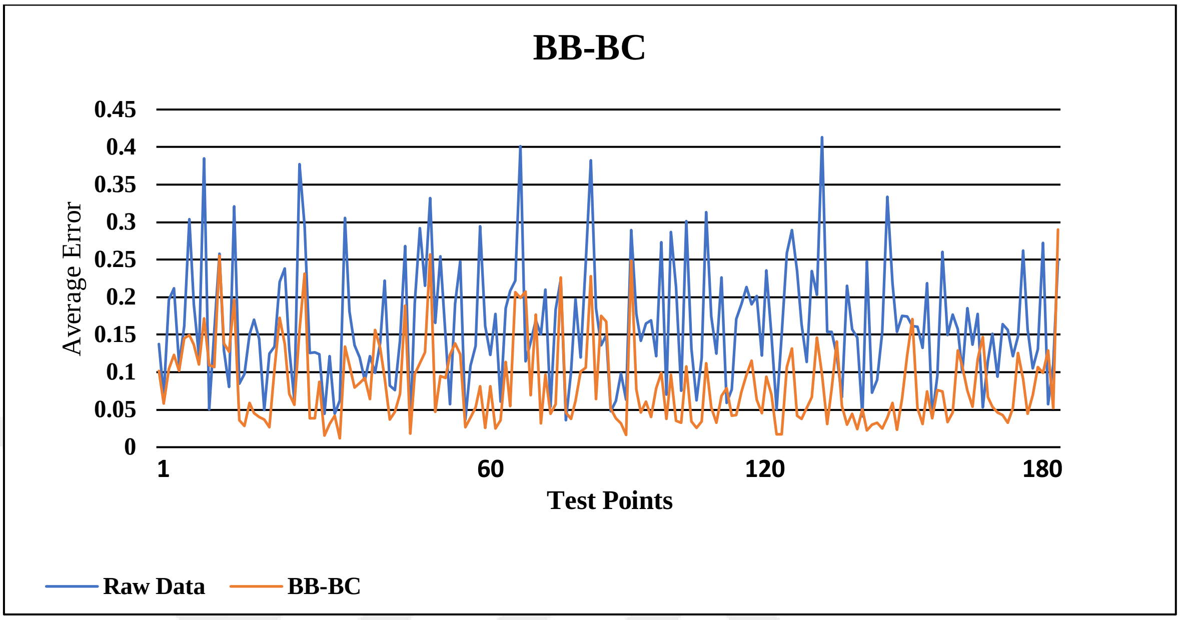

Figure 3.5 The improvement after applying BB-BC.

3.2.2 Kalman Filter then BB-BC algorithm simulation

To acquire a better optimized result and improve the accuracy of BBBC algorithm, in the second simulation the raw UWB test points for the training set and test set were used as input to Kalman Filter (KF) first, then the Kalman Filtered output data were used as input to the proposed system for the BB-BC algorithm. For the training set, the real location of the test point was used as performance metric for the prediction of Kalman Filter, then the number of iterations the produce the best prediction were used in the test set. Filtering noisy signals is important since many sensors have an output that is too noisy to be used directly and utilizing Kalman Filtering let you take account for the uncertainty in the signal/state. The H matrix was initialized, it's called the measurement matrix, which is a model of the sensors, however it is hard to determine. A popular approach is to initialize it as a diagonal identity matrix and tweak it to improve the final filter results. The covariance of the process noise Q was also initialized, and it does not get updated by the filter. This matrix tells the Kalman Filter how much error is in each action. R is covariance matrix of the measurement noise. It represents (electronic, random) noise characteristics of the sensor. Now, since this value is not defined, R is set to identity matrix. Finally, A matrix was initialized as well, which is the state transition matrix as following:

A= [1 0 dt 0 0 0;
0 1 0 dt 0 0;
0 0 1 0 dt 0;
0 0 0 1 0 dt;
0 0 0 0 1 0;
0 0 0 0 0 1]

Where dt = 0.1, which define the sample time. As a result of applying Kalman Filter, the average location error was reduced by approximately 31.03 % (from 16.34 cm to 11.27 cm) for the test set. Figure 3.6 shows the improvement when applying Kalman filter for the test set. Table 3.2 show the BB-BC obtained offset values for the Kalman Filtered UWB measurements.

As a result of using the Kalman Filtered UWB test points as an input to the BB-BC algorithm, the average location error was reduced by approximately 51.29 % (from 16.34 cm to 7.96 cm). Figure 3.7 shows the improvement to the Kalman Filtered UWB test points for the test set.

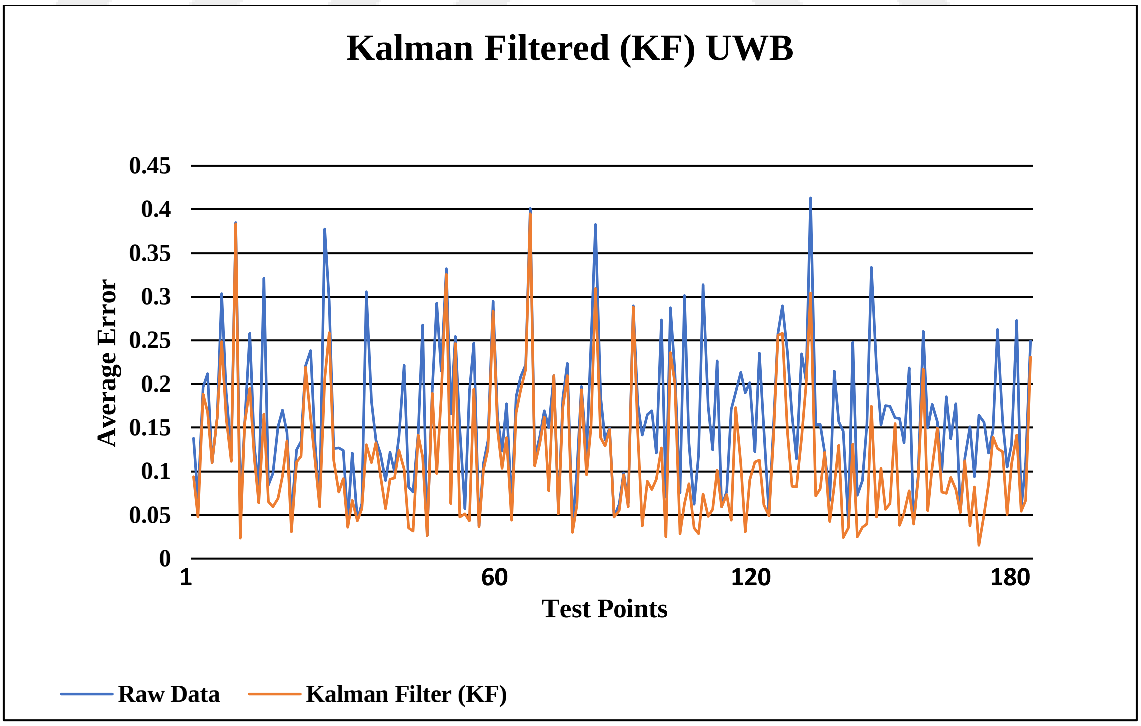

Figure 3.6 The improvement after applying Kalman Filter.

Table 3.2 BB-BC offset values for the Kalman Filtered UWB.

P.	X_{Offset}	Y_{Offset}	P.	X_{Offset}	Y_{Offset}	P.	X_{Offset}	Y_{Offset}
1	-0.004452698	-0.009401121	**61**	-0.171277364	0.095507099	**121**	0.111973805	-0.008996036
2	0.000839167	0.001653004	**62**	0.034890696	0.005573469	**122**	0.001360074	0.008043363
3	-0.006682247	0.086934956	**63**	-0.06419529	-0.058741689	**123**	0.004715827	0.000484926
4	-0.019335066	0.04634062	**64**	0.046240722	-0.116613893	**124**	0.055961679	0.004710373
5	0.000375786	0.013592907	**65**	-0.230744487	-0.00413322	**125**	-0.07359033	0.111706635
6	0.004988152	0.022557516	**66**	-0.073169615	-0.083701229	**126**	-0.183606367	0.226629881
7	-0.02866962	0.120433803	**67**	0.000227647	0.016212597	**127**	-0.157230828	0.253118367
8	0.004012279	0.024373697	**68**	-0.040268419	0.126709753	**128**	-0.049419812	0.147519909
9	0.009443672	0.005712742	**69**	0.018395743	0.061062175	**129**	0.002211581	0.004470769
10	0.052420459	-0.470636028	**70**	0.005832844	0.174214335	**130**	0.000240857	0.010569956
11	0.033423794	-0.097363387	**71**	-0.009324888	0.200538721	**131**	0.002669591	0.01866403
12	0.041473827	-0.252758446	**72**	0.021913634	0.029972283	**132**	0.002391118	0.043677708
13	0.16680138	-0.278905284	**73**	-0.102057993	0.372255906	**133**	-0.053726929	0.350513736
14	0.020462401	-0.014900107	**74**	0.011145366	0.048716771	**134**	-0.033752347	-0.038796434
15	0.026792012	-0.12677083	**75**	-0.01282896	0.146876449	**135**	0.011370974	-0.041144303
16	0.00384804	0.024265975	**76**	0.014175579	0.010014929	**136**	0.012717631	0.007007545
17	0.005742717	0.091913909	**77**	-0.030567989	-0.013282736	**137**	0.001537599	0.001732766
18	0.026793962	0.079508068	**78**	-0.186271509	0.055197448	**138**	0.000241868	0.01119839
19	0.017893566	-0.040392179	**79**	0.062854862	0.015556888	**139**	0.118269568	-0.04191768
20	-0.034829739	0.087813907	**80**	-0.1375657	-0.00917044	**140**	8.65E-05	0.001448221
21	0.022582856	0.162619713	**81**	0.014371669	0.022550305	**141**	0.037123247	0.024132376
22	0.02887165	0.057956027	**82**	0.006229408	0.017156305	**142**	0.002147991	0.019574982
23	0.022318645	0.132191698	**83**	0.000433164	0.00224732	**143**	0.001473452	0.001317913
24	0.028275673	0.142869189	**84**	-0.005756617	0.206040197	**144**	0.002884834	0.001458563
25	0.000117259	0.050433818	**85**	0.005086666	0.007046109	**145**	0.000704837	0.000396903
26	0.02787295	0.169351077	**86**	0.075495318	0.163856882	**146**	0.000259324	0.039368024
27	0.064399723	-0.064363789	**87**	0.035121694	0.071494298	**147**	0.00020227	0.004446765
28	0.074895298	0.063577167	**88**	0.051474964	-0.106255722	**148**	0.009502585	0.110038477
29	0.24651728	0.154268067	**89**	0.006473769	0.007789005	**149**	0.007595827	0.000174783
30	0.007070836	0.057056491	**90**	0.003602568	0.024290029	**150**	0.000702664	0.000119922
31	0.112775867	-0.034190462	**91**	0.001541214	0.002473021	**151**	0.011847162	0.009562367
32	-0.060049798	0.049468896	**92**	0.001551442	0.002315901	**152**	0.044520357	0.061474435
33	0.000947459	0.007106905	**93**	0.116224259	-0.012406822	**153**	0.001703576	0.000444605
34	-0.002157359	0.037613174	**94**	0.056998726	-0.016330696	**154**	0.005687507	0.004387977
35	0.001823338	0.078379671	**95**	-0.203502526	0.217413216	**155**	0.043867063	0.034657589
36	0.000749297	0.002626839	**96**	0.009123985	0.012149825	**156**	-0.033378341	0.090276781
37	0.019429281	-0.038228394	**97**	0.008630217	0.059758427	**157**	0.182202768	-0.107155226
38	0.003805108	0.011352742	**98**	0.000982674	0.007838648	**158**	0.000970993	0.000866079
39	0.004208097	0.013572841	**99**	-0.019465614	0.077750915	**159**	0.005949223	0.003865816
40	0.047617851	0.125164415	**100**	0.002729734	0.008598892	**160**	0.009878362	0.178533156
41	0.063937036	0.118602132	**101**	0.010579026	0.0113317	**161**	0.001660478	0.008379103
42	0.004671029	0.003916972	**102**	0.037012372	0.013807051	**162**	0.001926295	0.002561696
43	0.108304289	0.107382993	**103**	-0.054654001	0.253790149	**163**	0.028519074	0.101043134
44	0.00775339	0.007744819	**104**	-0.035460729	0.198651487	**164**	0.079836547	-0.050591518
45	0.042955856	0.093640455	**105**	0.015778246	0.04361748	**165**	0.190927139	0.040381104
46	-0.009293252	0.143597836	**106**	0.00419509	0.000414068	**166**	0.13095723	0.009455248
47	0.028271411	0.030590266	**107**	0.079475274	-0.013245461	**167**	0.001313554	0.002007672
48	0.020129235	0.025599499	**108**	0.000479318	0.003308769	**168**	0.085570773	0.04449075
49	-0.003197772	-0.092361494	**109**	0.000229375	0.001623825	**169**	0.000175806	0.001265529
50	0.039793532	0.003202767	**110**	0.001679027	0.002861347	**170**	0.001410503	0.001129085
51	0.019924649	-0.006706932	**111**	0.001247548	0.003798385	**171**	0.000346617	0.011483319
52	0.006576172	0.213688391	**112**	0.001432279	0.001247761	**172**	0.134396149	0.010988292
53	0.003952511	0.002970903	**113**	0.010876828	0.003665011	**173**	0.001132151	0.027332893
54	-0.068654742	0.176183259	**114**	0.002885001	0.002103791	**174**	0.136579852	0.003671197
55	0.033866584	0.067858752	**115**	-0.002036484	0.055769475	**175**	0.053659362	0.099518549
56	0.027312904	0.063795769	**116**	0.002427645	0.000202824	**176**	0.001279004	0.019392931
57	-0.034722934	0.306916801	**117**	-0.030886929	0.186725606	**177**	0.002505174	0.019705998
58	0.000618154	0.000664686	**118**	0.001619481	0.010209193	**178**	0.007429037	0.000254779
59	0.001274997	0.002621633	**119**	0.000580519	9.78E-05	**179**	0.08446876	0.005648342

60	0.002082215	0.000630607	**120**	-0.007788741	0.072487993	**180**	0.011343544	0.026349545

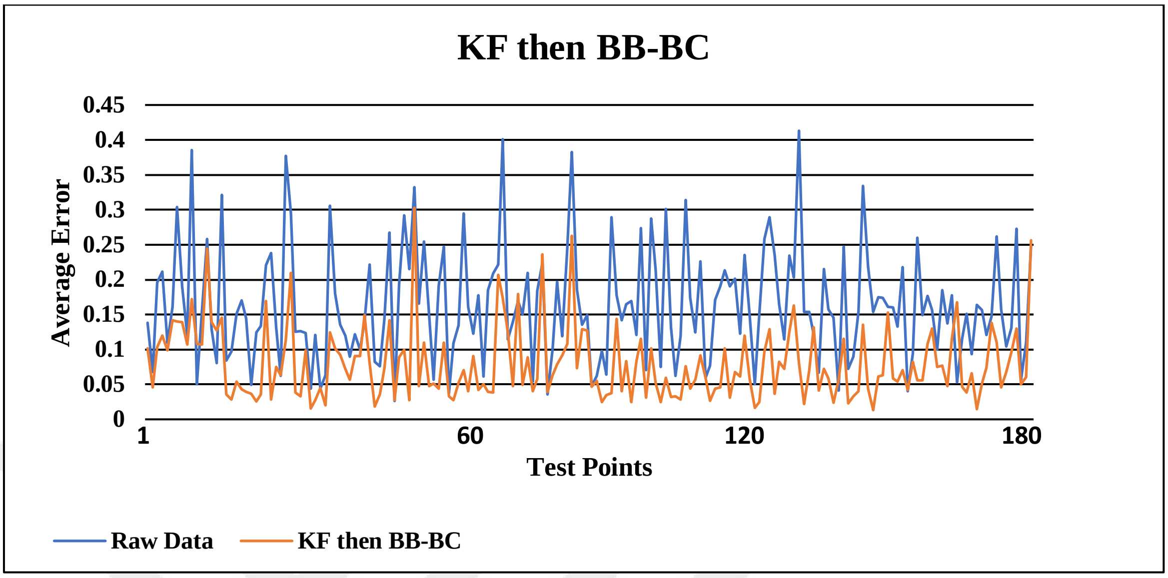

Figure 3.7 Kalman Filter then BB-BC algorithm.

3.2.3 BB-BC algorithm then Kalman Filter simulation

In the final simulation, the raw UWB test points for the training and test were used as input to BB-BC algorithm, then the Kalman Filter was applied to the output data for the training and test set. The results obtained from this simulation were the best result compared to the first and second simulations. In which the average location reduced by approximately 54.53 % (from 16.34 cm to 7.43 cm) for the test set. Figure 3.8 shows the improvement to the raw UWB test points for the test set. The comparison in computation time during the implementation of the mentioned simulations are presented in Table 3.3 Figure 3.9 show the results comparison among the implemented simulations for the test set.

Table 3.3 Computation time comparison of BB-BC simulations.

Simulation	Computation time in seconds
BB-BC	560.901
KF then BB-BC	914.345
BB-BC then KF	923.283

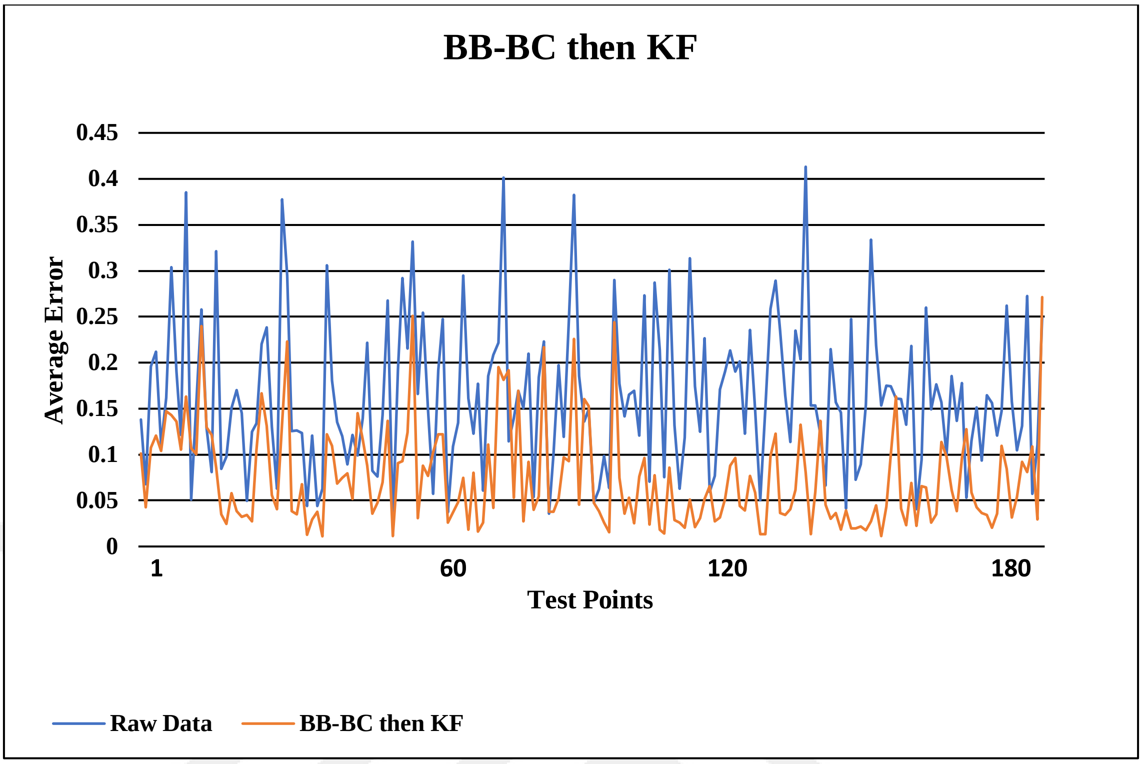

Figure 3.8 BB-BC algorithm then Kalman Filter.

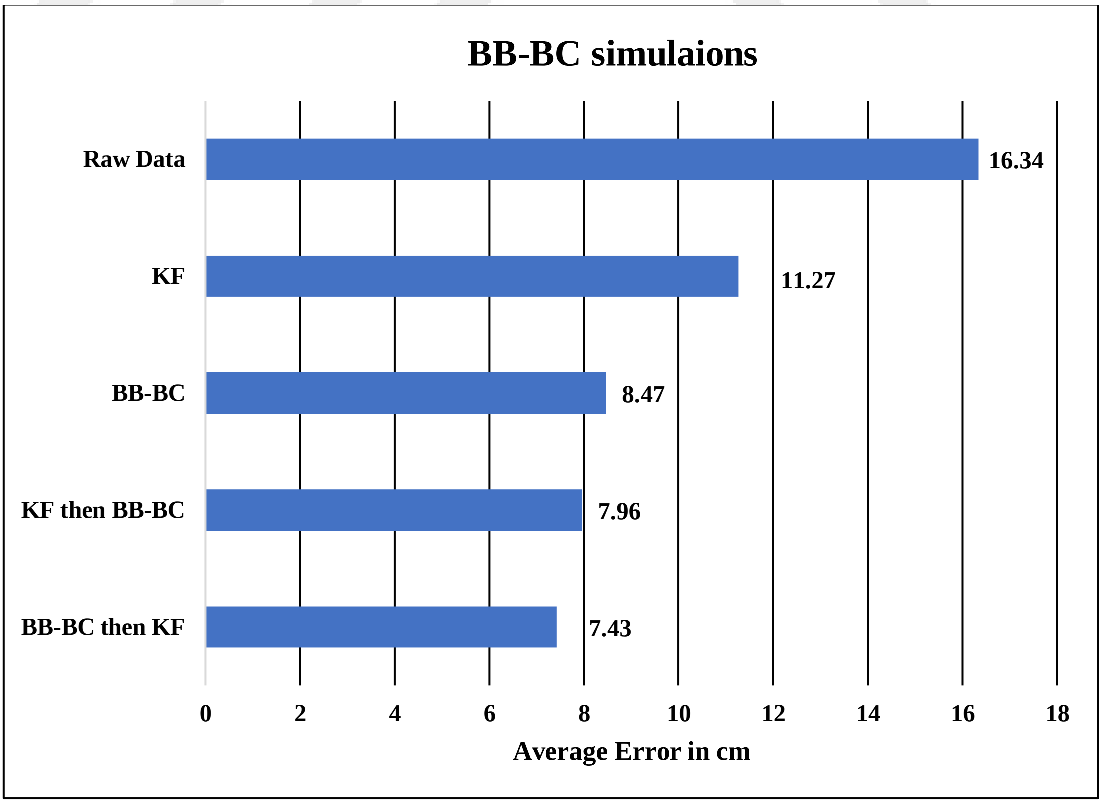

Figure 3.9 BB-BC simulations results.

3.3 Experimental Work and Evaluation of the Genetic Algorithm

The experiments were performed on Intel Core i7 dual core with 4 threads, using MATLAB R2018a Toolbox. The first stage in Genetic algorithm is to generate an initial population, the popular approach in generating the initial population, is the random generation of chromosomes. In which, each generation, the operators are implemented as follows: (1) Parent chromosomes are selected with respect to their fitness values; (2) Cross-over is implemented to the parent chromosomes and the new chromosomes are acquired; (3) The produced chromosomes are then evaluated by calculating the fitness values; (4) Finally, The GA process continue to the specified maximum number of generations. The fitness function used in Genetic Algorithm is defined in Eq. (3.1). The parameters values for the GA algorithm were chosen as shown in Table 3.3. Figure 3.10 show the flow chart of the proposed system for GA algorithm.

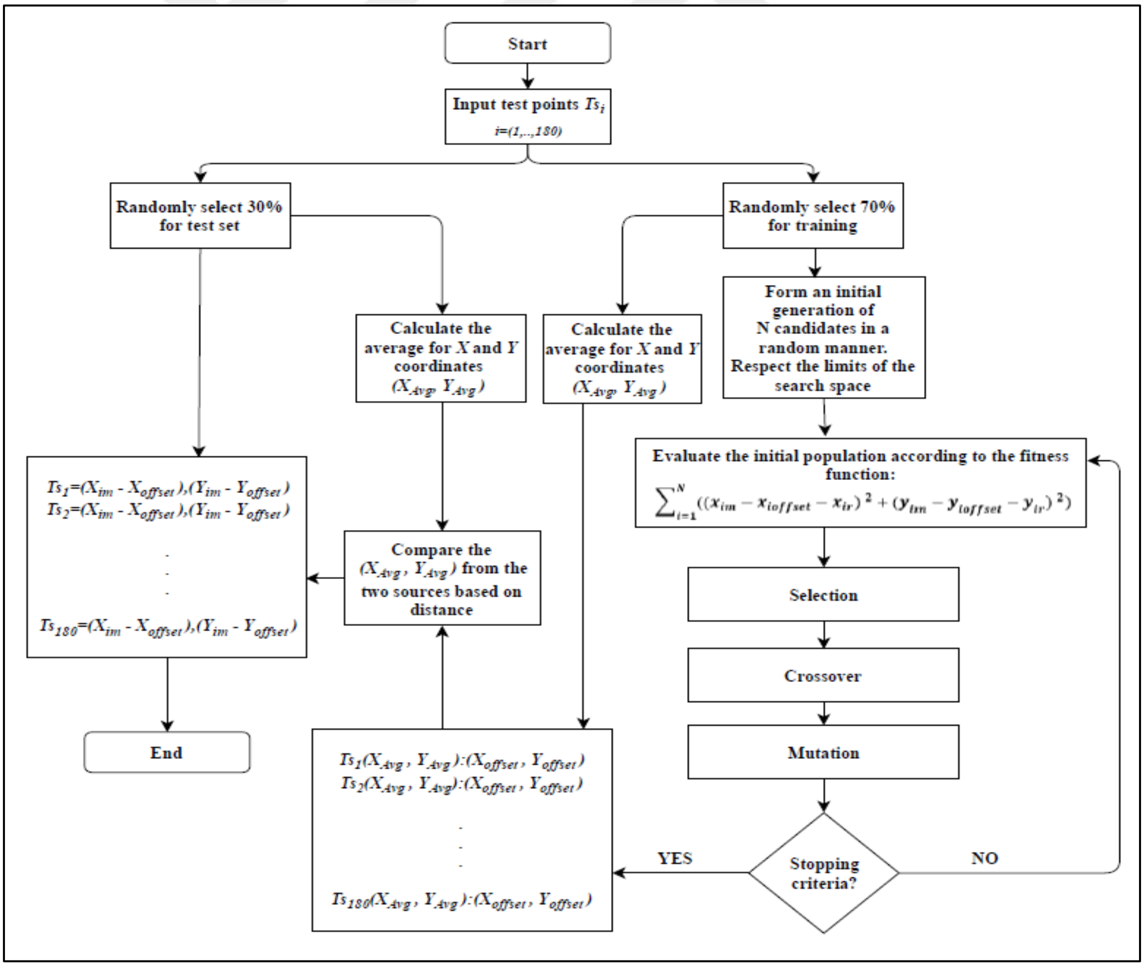

Figure 3.10 The proposed system for the GA algorithm implementation.

Table 3.4 Genetic Algorithm selected parameters.

Parameter	**Value**
Npop	Population size = 100
Generation	Maximum number of generations = 500 (Stopping criteria)
Fitness scaling	Rank
Crossover fraction	0.8
Reproduction (Selection)	Elite count = 0.05* Npop
Mutation rate	0.1

3.3.1 GA standalone simulation

In the first simulation, the Raw UWB training set were used as input to Genetic Algorithm in order to obtain the offset values to be used by the test set as shown in Figure 3.10. The acquired offset values form the implementation of GA are presented in Table 3.4. As a result, the average location error was reduced by 31.76 % (from 16.34 cm to 11.15 cm) for the test set. Figure 3.11 shows the improvement to the raw UWB test set.

Table 3.5 GA offset values.

P.	X_{Offset}	Y_{Offset}	**P.**	X_{Offset}	Y_{Offset}	**P.**	X_{Offset}	Y_{Offset}
1	0.007995591	0.013554168	**61**	-0.160061289	0.087259949	**121**	0.089432705	-0.023494573
2	0.002111152	0.023034086	**62**	0.017621977	-0.00460163	**122**	0.042009943	0.000765888
3	-0.001693977	0.088075078	**63**	-0.064688488	-0.074471237	**123**	0.026664627	4.50E-03
4	0.086677365	-0.045117969	**64**	-0.004367768	-0.112494848	**124**	0.025642652	-0.011860366
5	-0.006981793	0.071668384	**65**	-0.263243978	-0.026730104	**125**	-0.063700825	0.020722294
6	-0.052430109	0.102359224	**66**	-0.108551498	-0.111567405	**126**	-0.188428549	0.05162898
7	0.090139422	0.165215742	**67**	0.012406109	0.099859081	**127**	-0.111987934	0.021790454
8	0.102654079	0.036535477	**68**	0.016855183	0.020293017	**128**	0.007795127	0.014001176
9	0.026361532	0.009448725	**69**	0.014750483	0.005512197	**129**	0.01041955	0.039101405
10	-0.094288767	-0.381659561	**70**	-0.003621493	0.020707348	**130**	0.013937632	0.016830778
11	0.033409558	0.012570532	**71**	-0.001914178	0.03530327	**131**	0.020797987	0.037641154
12	-0.011287725	-0.14539248	**72**	-0.122720414	0.041093195	**132**	-0.085588931	0.03563343
13	0.13179761	-0.160707735	**73**	-0.228278713	0.242092688	**133**	0.073133656	0.124473226
14	-0.096755544	0.015903266	**74**	-0.037962652	-0.03793449	**134**	0.022243784	0.011541301
15	-0.064236844	-0.01620629	**75**	-0.061099766	0.113248438	**135**	0.012983621	0.015993433
16	0.011030451	0.075164129	**76**	-0.108095112	-0.108416185	**136**	-0.091200416	0.002452636
17	0.01372425	-0.000370396	**77**	0.022764051	0.028175667	**137**	0.02110593	0.035657112
18	0.013119995	0.010961089	**78**	-0.192490581	-0.011661365	**138**	0.016669831	0.022364795
19	0.043819921	-0.023962129	**79**	0.030968138	-0.018462149	**139**	0.028234267	-0.004062821
20	0.014846879	0.01296309	**80**	-0.161758753	-0.033155721	**140**	0.00435565	0.021784721
21	0.009618815	0.110343979	**81**	-0.155166384	-0.168166245	**141**	0.013539925	-0.03278835
22	0.012946312	0.01157686	**82**	-0.030706766	-0.003130598	**142**	0.007907831	0.024233864
23	0.007187166	0.051597446	**83**	0.077358991	0.00569887	**143**	0.015336175	0.017000345

24	0.025042779	0.016813819	**84**	-0.075499264	0.073823478	**144**	0.012653875	0.020464625
25	-0.034230711	0.075085827	**85**	-0.027563913	0.050171161	**145**	0.064270763	0.028473365
26	0.019504245	0.019412899	**86**	0.057338944	0.047154875	**146**	0.018982736	0.041414723
27	0.054339345	-0.121708319	**87**	0.101522827	0.151364221	**147**	0.007870882	0.01865221
28	0.019646721	0.032037351	**88**	0.017692377	-0.068844601	**148**	0.010289796	0.016247901
29	0.077212657	0.036834414	**89**	-0.102946999	0.018900112	**149**	0.024398334	0.022606746
30	0.097739684	0.033721863	**90**	-0.039399924	0.123843716	**150**	0.009250443	0.016676345
31	0.055663725	-0.010935012	**91**	0.000622066	-0.016586401	**151**	0.051482583	-0.104256946
32	0.010147549	0.065185209	**92**	-0.028459573	0.04213398	**152**	0.01756852	0.011414578
33	0.109124177	-0.022986714	**93**	0.015898849	-0.024271069	**153**	-0.001674982	0.016158662
34	0.006655227	0.006616704	**94**	0.005011729	-0.020882626	**154**	0.016144542	0.040864
35	0.012013661	0.018628413	**95**	-0.22320646	0.024881238	**155**	0.022789668	-0.016488843
36	0.018250059	-0.024607265	**96**	-0.026322923	0.034786351	**156**	-0.081581075	0.02568866
37	0.007455917	-0.042626926	**97**	0.013200939	0.015730911	**157**	0.042035882	-0.022363704
38	0.029836371	0.013547051	**98**	0.034924034	0.034523259	**158**	0.020543447	0.018318098
39	0.095596739	0.031689854	**99**	0.006078913	0.009286359	**159**	0.00904279	0.034517298
40	-0.037511234	0.036883423	**100**	0.024123687	5.84E-03	**160**	-0.094146883	0.07518581
41	0.044007226	0.07238993	**101**	0.174665415	0.066229128	**161**	-0.021057844	0.049872319
42	0.024029181	-0.025901758	**102**	0.015759606	0.005838669	**162**	0.013402402	0.015700358
43	0.012561389	0.012996784	**103**	0.028051647	0.042868024	**163**	-0.001472059	0.009540672
44	-0.017034957	0.058790991	**104**	-0.01993307	0.036342204	**164**	-0.011743328	0.01526819
45	-0.108949703	0.054111491	**105**	0.01735451	0.010954552	**165**	-0.00986535	-0.029596475
46	0.009437472	0.186463938	**106**	0.013176597	0.032754961	**166**	0.045297595	-0.0049333
47	0.005164041	0.003872865	**107**	0.062140399	-0.003995032	**167**	0.000200351	0.038346938
48	0.015905481	0.036266646	**108**	-0.005053316	0.009584471	**168**	0.082883374	-0.003514098
49	-0.03104545	-0.124527764	**109**	0.018222978	0.037038552	**169**	0.019428498	0.023890509
50	0.089062189	0.068302008	**110**	0.046789312	0.097991772	**170**	0.020092188	0.019515503
51	0.000856372	-0.018730517	**111**	0.012955998	0.020292777	**171**	-0.004163146	0.00680188
52	-0.056086025	0.08135287	**112**	0.011592398	0.010873019	**172**	0.067730941	-0.116224706
53	0.027631964	0.091947685	**113**	0.022357199	0.085284607	**173**	0.016211987	0.100878232
54	0.004692105	0.043184787	**114**	-0.007656371	-0.050865927	**174**	0.014849009	-0.002268364
55	-0.217671572	0.02399824	**115**	-0.027642601	0.011126746	**175**	0.012192779	0.024452171
56	0.011037376	0.035376717	**116**	0.047822795	0.012670909	**176**	-0.011117678	0.020504189
57	-0.095076638	0.215475771	**117**	-0.007900009	0.057338761	**177**	0.02961493	0.038273457
58	0.032063066	0.019901808	**118**	0.036413251	0.022502864	**178**	0.010995542	-0.020647512
59	0.023362187	0.015569395	**119**	0.021256411	0.007536819	**179**	0.013634667	-0.00930086
60	-0.009422622	0.010489067	**120**	0.060508213	0.027606005	**180**	-0.159537714	-0.172247015

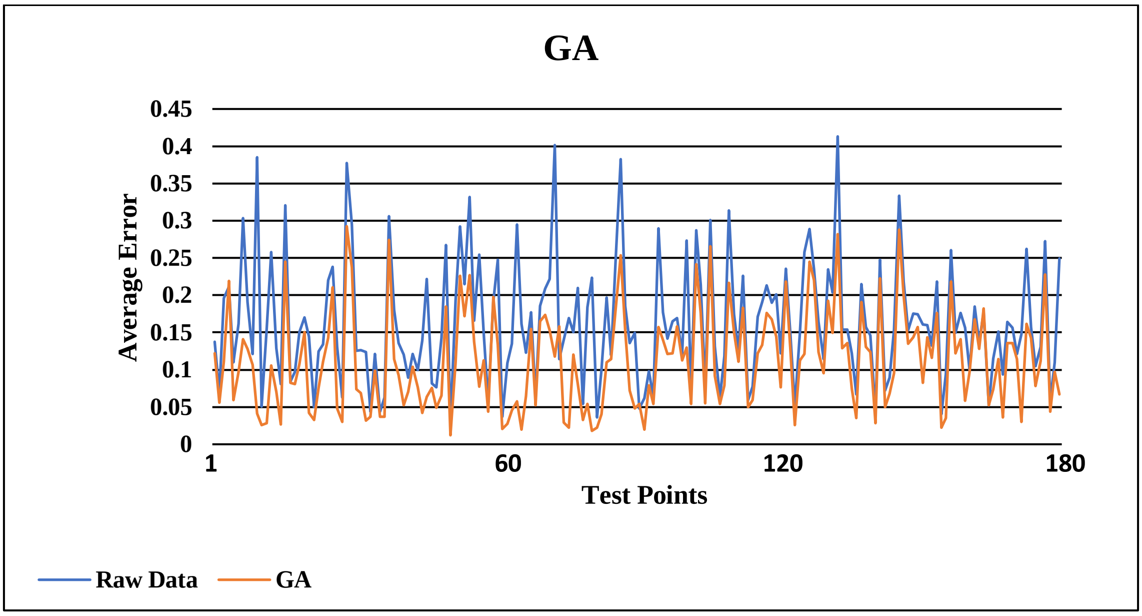

Figure 3.11 UWB test points location error when applying GA.

3.3.2 Kalman Filter then GA simulation

To increase the accuracy of Genetic algorithm, Kalman Filter was applied on UWB test points first for the training set and test set. In which the number of iterations the produce the best prediction was obtained from the training set and used in the test set. Then the Kalman Filtered UWB test points for the training and test set were used as an input to the prosed system for the GA. Table 3.5 show the GA obtained offset values for the Kalman Filtered UWB measurements. The improvement in location average error for the test set are shown in Figure 3.12. In which the average localization error was reduced by approximately 46.57 % (from 16.34 cm to 8.73 cm).

Table 3.6 GA offset values for the Kalman Filtered UWB.

P.	X_{Offset}	Y_{Offset}	P.	X_{Offset}	Y_{Offset}	P.	X_{Offset}	Y_{Offset}
1	-0.013934788	0.007090201	**61**	-0.127359075	0.027747356	**121**	0.067984035	-0.064132048
2	-0.003849929	0.015552505	**62**	0.0131404	-0.007296442	**122**	0.025111454	-0.042028436
3	-0.024995119	0.05518178	**63**	-0.044808024	-0.003053772	**123**	0.020896592	-0.016032553
4	-0.010112825	-0.10439578	**64**	0.036235616	-0.024436087	**124**	0.019095266	-0.000775138
5	0.002864392	0.092811756	**65**	-0.193971394	-0.016322981	**125**	-0.002000519	0.013274685
6	-0.071754817	0.128981901	**66**	-0.041373413	-0.010523608	**126**	-0.034487729	0.008787451
7	-0.004503454	0.116092415	**67**	-0.003265355	0.069924934	**127**	-0.152503411	-0.009374035
8	0.030597471	0.002252877	**68**	-0.009300766	0.010776766	**128**	-0.005184456	0.006979028
9	-0.014740059	0.005292973	**69**	0.000448096	-0.00347114	**129**	-0.011824721	0.02463667
10	-0.093761137	-0.230580001	**70**	-0.029543536	0.005894197	**130**	-0.007970264	0.006149619
11	0.005073229	0.043135309	**71**	-0.023934597	0.017451825	**131**	-0.00095569	0.01116418
12	0.00391877	-0.023367182	**72**	-0.118049452	0.018830005	**132**	-0.061595582	0.005606383
13	0.022616543	-0.062625624	**73**	-0.169834906	0.223369971	**133**	-0.063592695	0.075870888
14	-0.076643833	0.007544823	**74**	-0.102148421	-0.057268373	**134**	-0.006220672	-0.003462066
15	-0.037964348	0.022672466	**75**	-0.008116744	0.117595812	**135**	-0.02358464	-0.0027708
16	-0.023925787	0.03942252	**76**	-0.024265621	-0.042660493	**136**	-0.048551081	0.020980763
17	-0.007036946	-0.011396629	**77**	-0.013358007	-0.02249812	**137**	0.01079783	0.004010772
18	0.008288627	0.001945679	**78**	-0.031728689	0.012675152	**138**	-0.004473256	-0.000850677
19	-0.011663014	-0.040707356	**79**	0.021938672	-0.019715745	**139**	0.015176128	-0.054648141
20	-0.01141553	0.004430245	**80**	-0.042811374	-0.007124995	**140**	-0.01611384	0.010768822
21	-0.001397356	0.094502334	**81**	-0.014237544	-0.014231128	**141**	0.015222232	-0.022103148
22	0.006675903	0.006834756	**82**	-0.019907512	-0.00437135	**142**	-0.032936569	0.016900353
23	0.004689379	0.037345545	**83**	0.027872775	-0.036473458	**143**	0.008033093	0.009210534
24	0.00877366	-0.001163207	**84**	-0.04876592	0.063248862	**144**	0.007078347	0.009985511
25	-0.011510377	0.055768865	**85**	-0.060597445	0.066182714	**145**	0.009675723	-0.000922071
26	0.005753011	0.006338709	**86**	-0.019993047	0.029818475	**146**	-0.002282357	0.009792239
27	0.020501661	0.006088412	**87**	-0.012076547	0.114680305	**147**	-0.00804825	0.011242335
28	0.008360818	0.030321376	**88**	-0.010031713	-0.098001871	**148**	0.005723784	0.005316735
29	0.037314679	0.006965379	**89**	-0.099446854	0.015854897	**149**	-0.013834075	-0.021711594
30	-0.044760665	0.014732338	**90**	-0.05528483	0.113561496	**150**	-0.0082431	-0.006405085
31	0.040944864	0.00171904	**91**	-0.011012881	-0.017232762	**151**	0.036771611	-0.094949178
32	-0.03069109	0.038789945	**92**	-0.007777801	0.026764671	**152**	0.011933151	-0.010459084
33	0.046096506	-0.076831444	**93**	0.005881395	-0.008688931	**153**	0.006730858	-0.009258729
34	-0.008371799	-0.002641449	**94**	0.003370341	-0.003977832	**154**	0.010667788	0.020518526
35	0.001269653	0.009855255	**95**	-0.066175239	0.010952582	**155**	0.019310384	-0.007614826
36	0.032268562	-0.053298959	**96**	-0.01131462	0.024650482	**156**	-0.075943067	0.013033136
37	0.003233798	-0.015349348	**97**	-0.010945693	0.006565555	**157**	0.03134309	-0.124218084

38	0.002260211	-0.002319978	**98**	0.001223649	0.0271946	**158**	0.007711224	-0.012605225
39	-0.008408669	-0.008054201	**99**	-0.010045846	0.003662177	**159**	-0.041791576	0.016091539
40	-0.011062853	0.01819054	**100**	-0.020379872	-0.009789508	**160**	-0.056938514	0.058986604
41	0.00582209	0.047237242	**101**	0.044308072	0.014578788	**161**	-0.036529745	0.006536299
42	0.016999817	-0.004002961	**102**	0.01004127	-0.008354599	**162**	-0.005889787	-0.026836491
43	0.002906868	0.0045746	**103**	-0.019383673	0.022045585	**163**	-0.019558121	0.000496559
44	-0.068785961	0.101834194	**104**	-0.008206382	0.014101663	**164**	-0.044387618	-0.005994822
45	-0.091401534	0.104504005	**105**	0.000813832	0.008386149	**165**	-0.027401094	-0.03240899
46	0.006020905	0.065881976	**106**	0.004437847	0.000384135	**166**	0.033778437	-0.023700844
47	0.003231088	3.26E-05	**107**	0.042396038	-0.032712907	**167**	-0.001496098	-0.007653758
48	0.001141624	0.015193434	**108**	-0.001210222	-4.47E-07	**168**	0.07345151	-0.025582539
49	0.003394238	-0.005304607	**109**	-0.007512591	0.009153463	**169**	0.002579493	0.002715253
50	0.073490673	0.017594881	**110**	0.023160489	0.00710088	**170**	0.009385453	0.003003327
51	0.00011297	-0.009662935	**111**	0.006071986	0.008936988	**171**	0.001565709	0.005008488
52	-0.028411889	0.070081606	**112**	0.003496166	-0.007833795	**172**	0.073176424	-0.098199667
53	-0.009124592	0.04304501	**113**	0.00514385	0.027411807	**173**	-0.005639098	0.061548817
54	-0.008830053	0.029683652	**114**	-0.023115957	-0.037218623	**174**	0.006421807	-0.026426692
55	-0.117311772	0.001802938	**115**	-0.014212446	0.00772835	**175**	0.006084263	0.009450767
56	-0.014134608	0.024799082	**116**	0.010743223	-0.001161342	**176**	-0.013964356	-0.064371873
57	-0.009873537	0.181932464	**117**	0.005079194	0.043803446	**177**	-0.041455857	-0.01297556
58	-0.01889398	0.007553923	**118**	0.002653911	0.003965282	**178**	-0.002511223	-0.024094371
59	-0.018424427	0.03296092	**119**	-0.005832569	-0.009205445	**179**	0.006208235	-0.040564435
60	-0.0413811	0.007136433	**120**	0.00155236	0.011749308	**180**	-0.133875429	-0.199727971

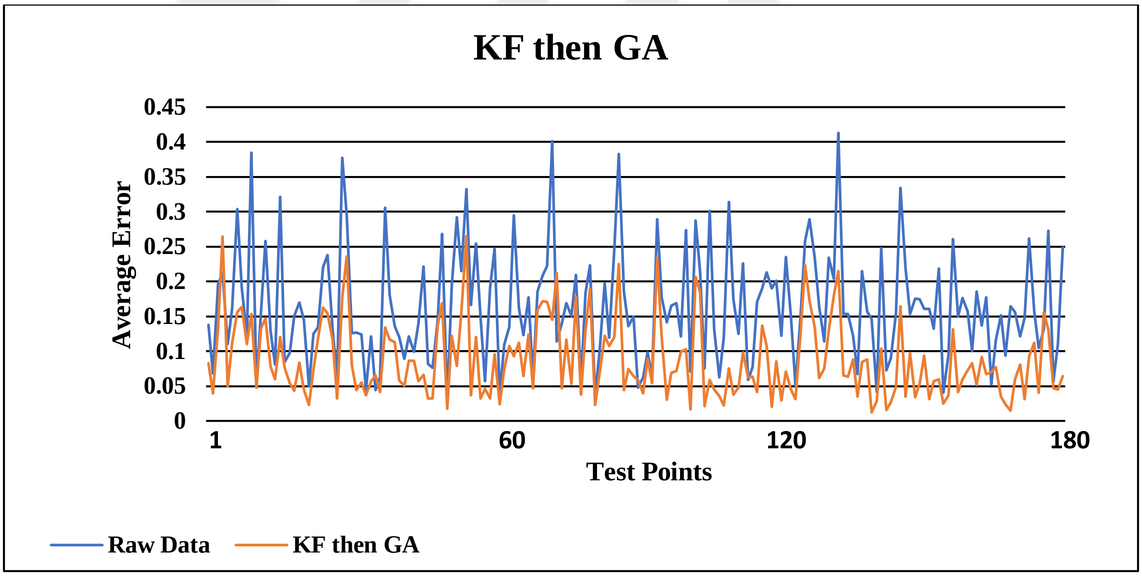

Figure 3.12 Kalman Filter then GA.

3.3.3 GA then Kalman Filter simulation

In the final simulation, the raw UWB training set and test set were used as input to the proposed system, the output training and testing set were used then as input to the Kalman Filter. The improvement in location average error for the test set are shown in Figure 3.13. In which the average localization error was reduced by approximately 52.08 % (from

16.34 cm to 7.83 cm). Which is the best result obtained when applying Genetic Algorithm. Figure 3.14 show the results comparison for the test set among the implemented simulations.

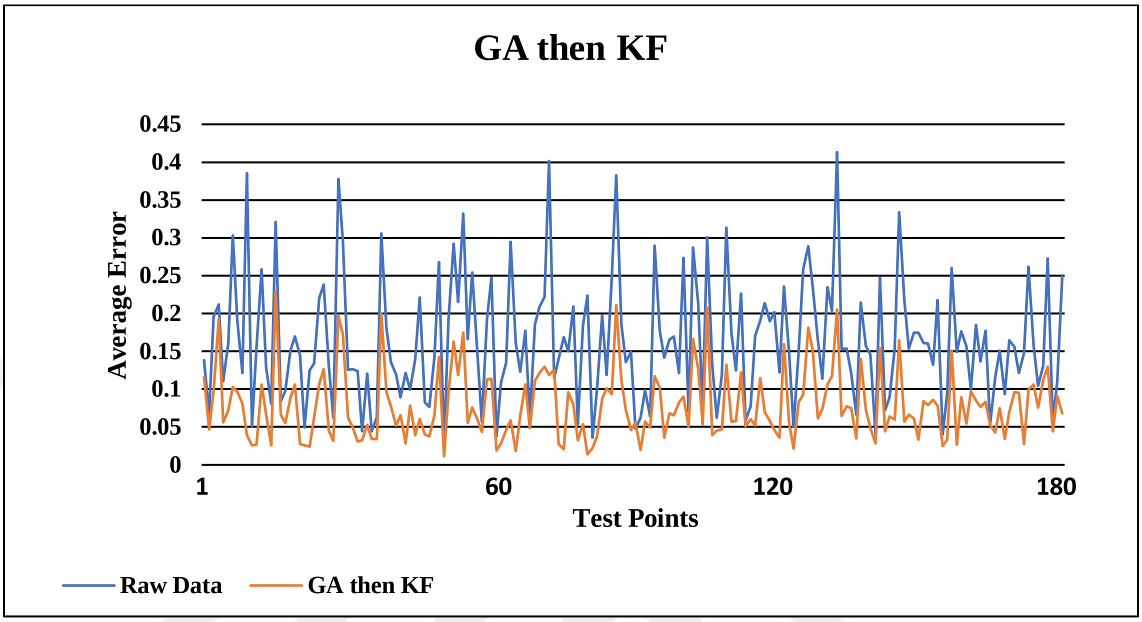

Figure 3.13 GA then Kalman Filter.

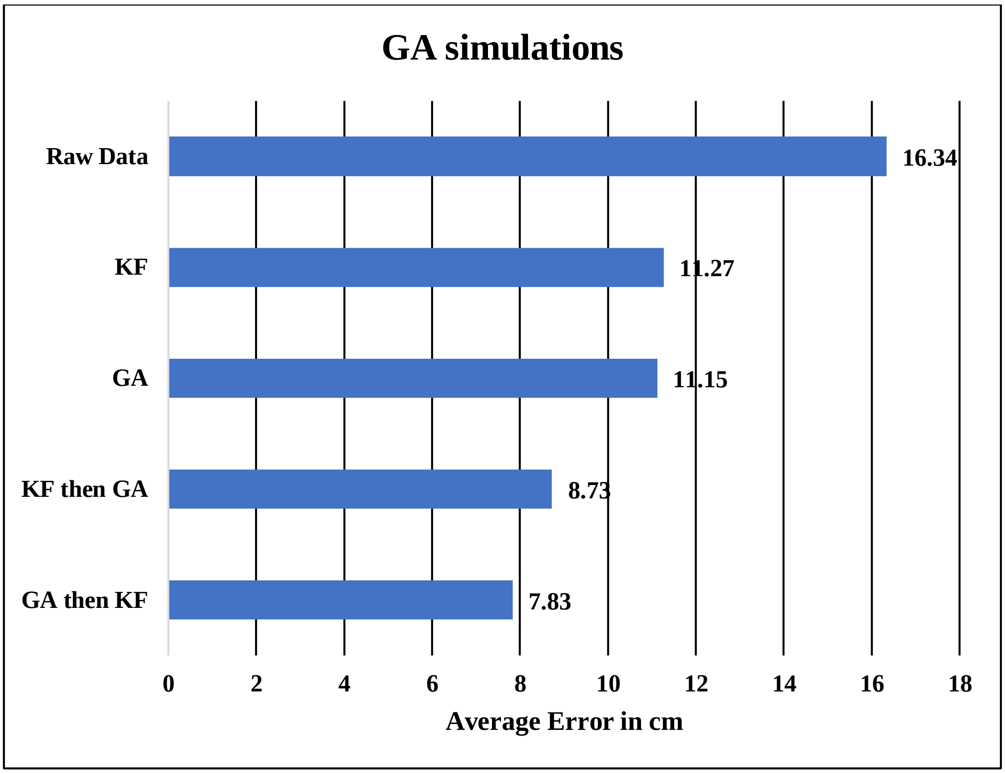

Figure 3.14 Results of GA simulations.

3.4 Results Summary of the Optimization Algorithms

The results of performing BB-BC and GA optimization algorithms for the training and test set are summarized in Table 3.6. In which the best result obtained when applying BB-BC algorithm, is when the data test points was used as input to the Big Bang-Big Crunch (BB-BC) then applying Kalman Filter on the output data. The best result obtained when applying genetic algorithm (GA), is also when the raw data test points were used as input to the GA algorithm, then applying Kalman Filter on the output data. When comparing both algorithms, the best result was obtained when applying BB-BC algorithm. In which the average location error was reduced by 54.53 % (from 16.34 cm to 7.43 cm).

Table 3.7 Results summary of the BB-BC and GA.

Input data	Raw Data	Raw Data	KF Data	Raw Data	Raw Data	KF Data	Raw Data
Algorithm	-	BB-BC	BB-BC	BB-BC **then** Kalman Filter	GA	GA	GA **then** Kalman Filter
Average Location Error (training)	16.3378 cm	8.42 cm	7.91 cm	7.37 cm	11.13 cm	8.69 cm	7.82 cm
Average Location Error (test)	16.3442 cm	8.47 cm	7.96 cm	7.43 cm	11.15 cm	8.73 cm	7.83 cm

Table 3.8 Computation time comparison of GA simulations.

Simulation	**Computation time in seconds**
GA	1300.898 s
KF then GA	1654.653 s
BB-BC then GA	1651.091 s

4. EXPERIMNTAL WORK AND EVALUATION OF THE MACHINE LEARNING AND HYBRID ALGORITHMS

4.1 Machine Learning Algorithms

Experiments were performed using ALC data set. The goal is focused on improving the accuracy of UWB indoor positioning system using machine learning methods. Accuracy were used as the performance metrics in the comparison among the clustering methods. The accuracy metric is based on the distance between the measured location to real location for a given point. It was calculated the distance using Euclidean distance:

$$d = \sqrt{(x_r - x_m)^2 + (y_r - y_m)^2} \tag{4.1}$$

where x_r, y_r are the coordinates of real location and x_m, y_m are the coordinates of the measured location. The ALC dataset has 180 test points location, and each test point has 150 samples. The dataset was partitioned randomly into training set and test set. In which the training set include 70% of the samples and test set has 30% of the samples. The proposed system for the clustering algorithms implementation is shown in Figure 4.2.

4.1.1 Standalone clustering algorithms

The proposed system is applicable for K-Means, FCM, and Mean Shift algorithms. However, when it comes to select the optimal number of clusters for each test point, K-Means and FCM algorithms are similar in term that we need to pre-defined the number of clusters, whereas Mean Shift, is non-parametric algorithm, which mean that we don't need to set the number of clusters. Thus, the average silhouette method was used to define the optimal number of clusters in K-Means and FCM algorithms for each test set point by varying k (number of clusters) from 2 to 6 clusters. For each k, the average silhouette c-was calculated using Eq. 2.22. Then, the number of clusters with the highest average silhouette coefficient was selected, for both the training set and test set. In order to understand the calculation of the average silhouette method, the following example will specify the optimal number of clusters obtained for the tenth test point (Ts10), for the test set which has 45 samples. The number of clusters is represented by C = {1, . . . ,6}.

Table 4.1 Silhouette coefficient values for the tenth test point.

Samples	**C=2**	**C=3**	**C=4**	**C=5**	**C=6**
1	0.8946	0.2468	0.3928	0.4363	0.4384
2	0.9095	0.3715	0.1627	0.5313	0.1833
3	0.9065	0.0860	0.3548	0.5160	0.3015
4	0.9190	0.5000	0.2619	0.5648	0.5216
5	0.8968	0.4641	0.3704	0.5119	0.4928
6	0.5563	0.5497	1	1	1
7	0.2049	0.1840	1	1	1
8	0.4813	0.1924	0.1776	1	1
9	0.8697	0.3864	0.2296	0.4514	0.3437
10	0.9190	0.5000	0.2619	0.5648	0.5216
11	0.9444	0.4866	0.4266	0.6086	0.1944
12	0.8676	0.3844	0.2429	0.3958	0.3611
13	0.8887	0.4367	0.2759	0.4484	0.4222
14	0.7884	0.2456	0.2753	0.2123	0.0394
15	0.9188	0.2500	0.3810	0.0902	0.3736
16	0.9352	0.3610	0.4397	0.3056	-0.0031
17	0.9075	0.3432	-0.0640	-0.0126	0.0806
18	0.9188	0.2500	0.3810	0.0902	0.3736
19	0.9286	0.4138	0.5907	0.4676	0.6759
20	0.9408	0.1055	0.5917	0.5833	0.5759
21	0.9408	0.1055	0.5917	0.5833	0.5759
22	0.9444	0.4866	0.4266	0.6086	0.1944
23	0.9408	0.1055	0.5917	0.5833	0.5759

24	0.9352	0.3610	0.4397	0.3056	-0.0031
25	0.9444	0.4866	0.4266	0.6086	0.1944
26	0.9444	0.4866	0.4266	0.6086	0.1944
27	0.9408	0.1055	0.5917	0.5833	0.5759
28	0.9332	0.5433	0.1176	0.5722	0.5185
29	0.9240	0.4421	-0.0256	0.3056	0.1691
30	0.9119	0.4965	0.2826	0.5093	0.4861
31	0.9048	0.4228	0.5215	0.1480	-0.1884
32	0.9318	-0.0467	0.5793	0.2901	0.3469
33	0.9352	0.3610	0.4397	0.3056	-0.0031
34	0.9408	0.1055	0.5917	0.5833	0.5759
35	0.9444	0.4866	0.4266	0.6086	0.1944
36	0.9444	0.4866	0.4266	0.6086	0.1944
37	0.9332	0.5433	0.1176	0.5722	0.5185
38	0.9309	0.4406	0.2359	0.5936	0.2735
39	0.9286	0.4138	0.5907	0.4676	0.6759
40	0.9153	0.3239	0.4685	0.4780	0.5288
41	0.9408	0.1055	0.5917	0.5833	0.5759
42	0.9190	0.5000	0.2619	0.5648	0.5216
43	0.9286	0.4138	0.5907	0.4676	0.6759
44	0.9201	0.4608	0.5805	0.1594	0.4286
45	0.9201	0.4608	0.5805	0.1594	0.4286
Mean Value →	**0.886546**	**0.352344**	**0.41389**	**0.480537**	**0.402796**
Max →	**0.886546**				

Based on the obtained mean values for each clustering, the maximum mean value is 0.886546, which is belong to cluster number of 2. Thus, the selected number of clusters to be used in tenth test point (Ts10) for K-Means algorithm is 2.

Figure 4.1 show the graphical silhouette values for each specified number of clusters for tenth test point in test set.

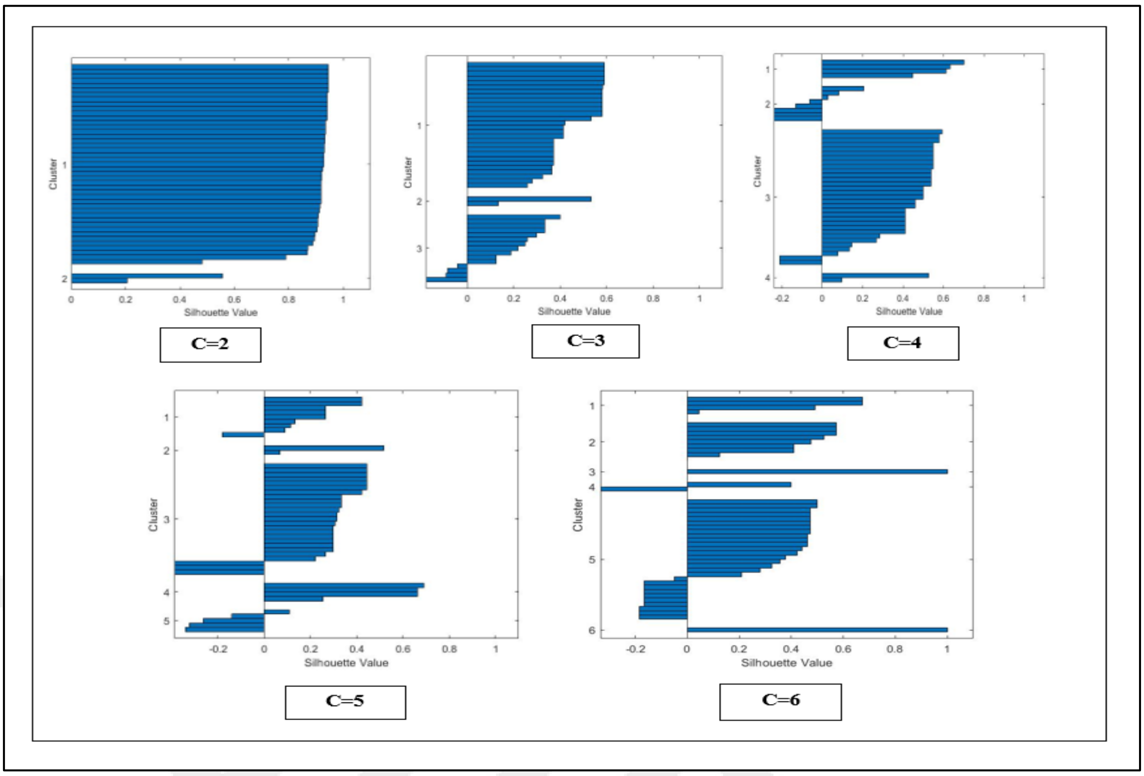

Figure 4.1 Silhouette values for the tenth test point in test set.

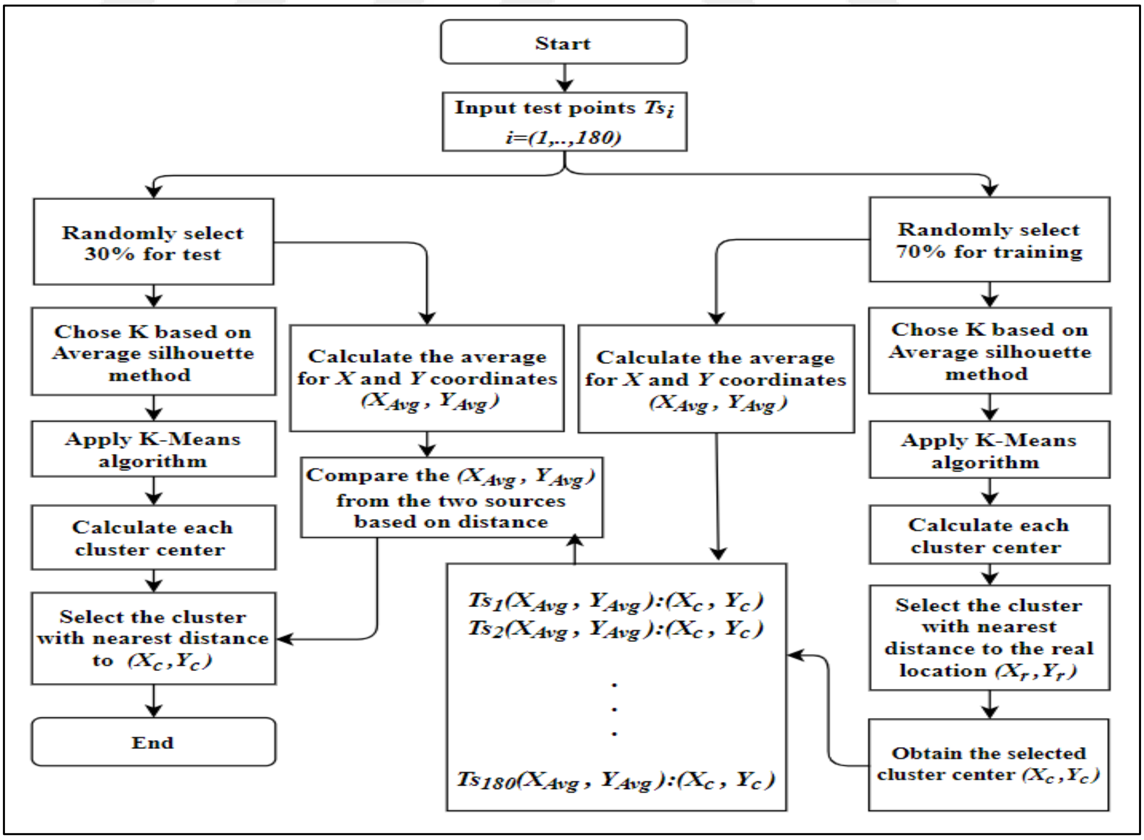

Figure 4.2 Flow chart of the proposed system for the clustering algorithm.

Figure 4.3 and Figure 4.4 shows the maximum average silhouette coefficient for K-Means and FCM for the training set, respectively. While Figure 4.5 and Figure 4.6 shows the maximum average silhouette coefficient for K-Means and FCM for the test set, respectively.

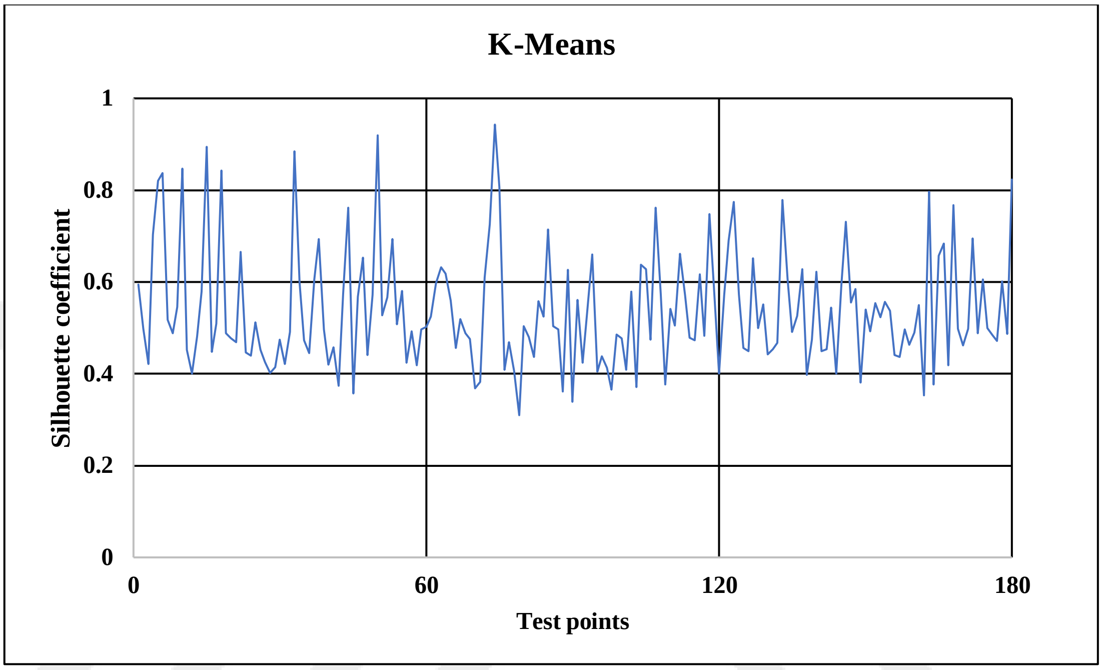

Figure 4.3 The maximum average silhouette coefficient in K-Means for the training set.

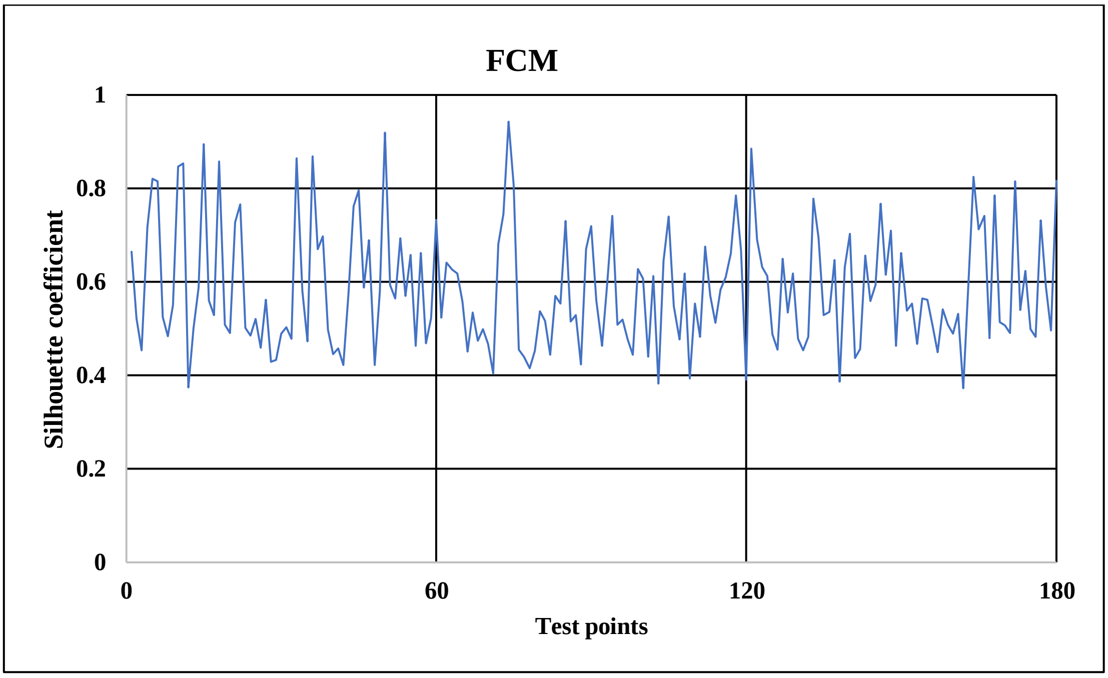

Figure 4.4 The maximum average silhouette coefficient in FCM for the training set.

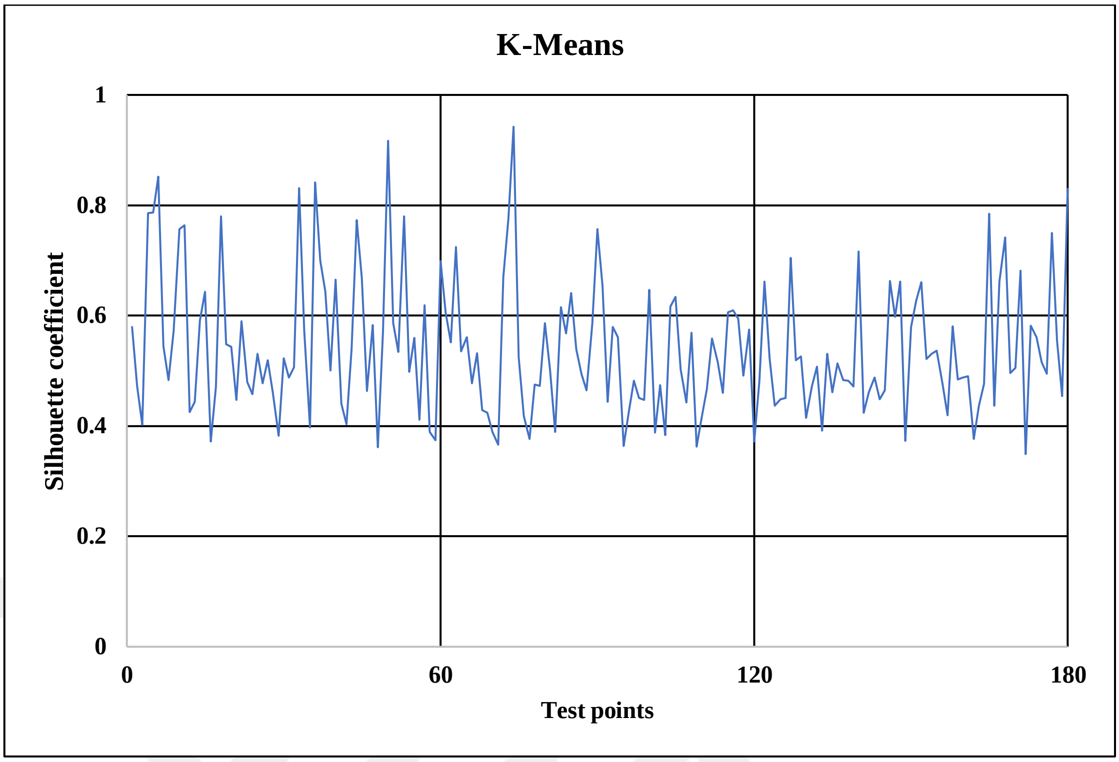

Figure 4.5 The maximum average silhouette coefficient in K-Means for the test set.

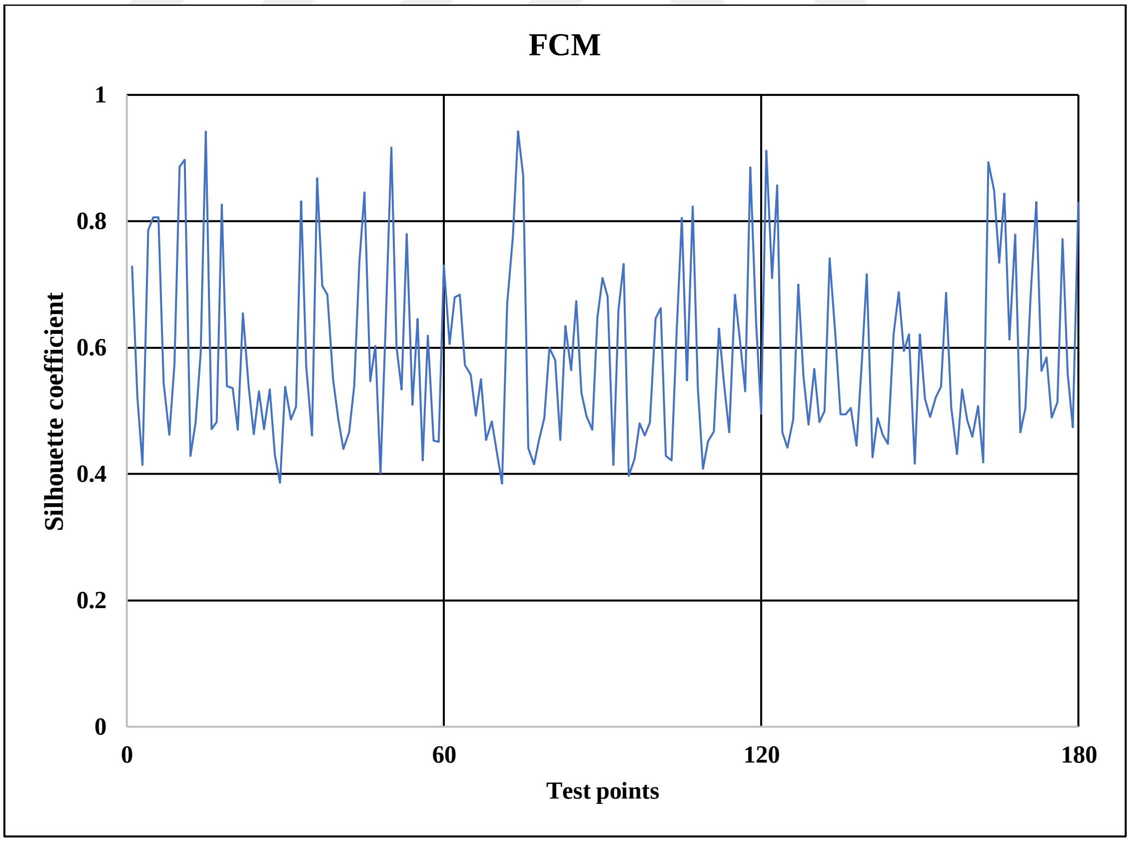

Figure 4.6 The maximum average silhouette coefficient in FCM for the test set.

Figure 4.7 shows the optimal distribution of the measured UWB test points (180 points) over clusters when applying the clustering algorithms for the training set. After setting the obtained number of clusters in all of the implemented algorithms for training set, one of the outcome clusters was chosen as a delegate based on its distance to the real location (X_r, Y_r) using Eq. 4.1. Then, the selected cluster center was calculated (X_c, Y_c). So, for each test point in the training set, we have the selected cluster center, which is coordinates dependent.

When it comes to the test set, the average silhouette method was also used to define the optimal number of clusters for K-Means and FCM algorithms. The optimal distribution of the test set over clusters is shown in Figure 4.8. One of the outcome clusters was chosen as a delegate based on its distance to (X_c, Y_c) for each test point. In order to identify which (X_c, Y_c). value belong to which test point in the test set, the average for each test point (X_{Avg}, Y_{Avg}) in both the training set and the test set were calculated. Then, the average of test point (X_{Avg}, Y_{Avg}) in the test set that has nearest distance to the test point (X_{Avg}, Y_{Avg}) in the training set, uses the corresponding (X_c, Y_c) value to select the delegate cluster. Table 4.2, Table 4.3, and Table 4.4 show the (X_c, Y_c) values that obtained from the training set for each test point, and to be used to select the delegate cluster from the test set for the K-Means, FCM, and Mean Shift algorithms, respectively. The average location error comparison for the training set and test are shown in Figure 4.9 and Figure 4.10, respectivly.

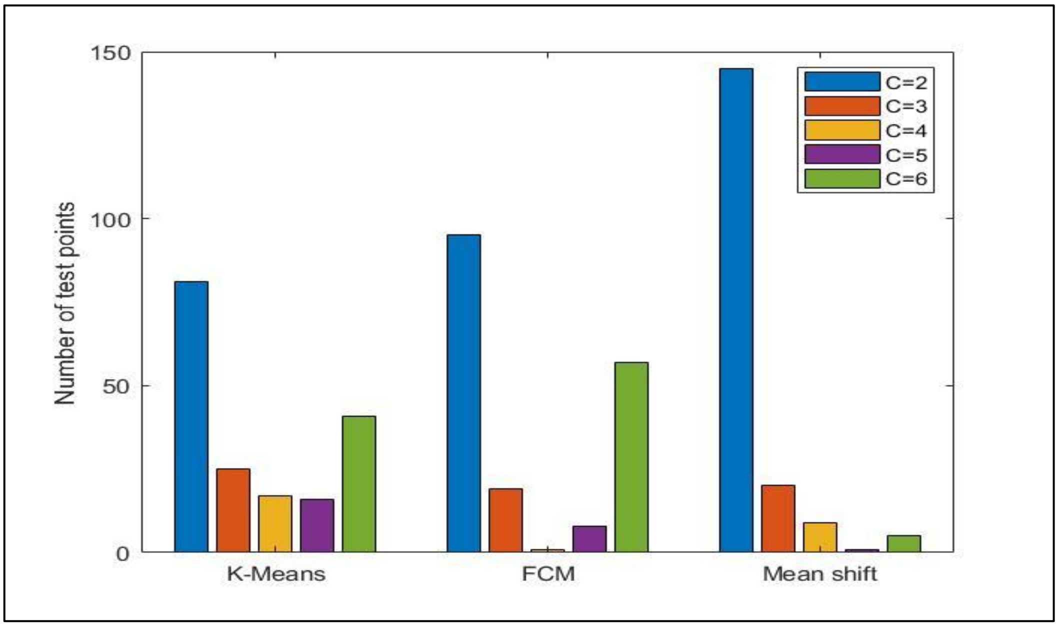

Figure 4.7 The distribution of UWB test points over clusters for the training set.

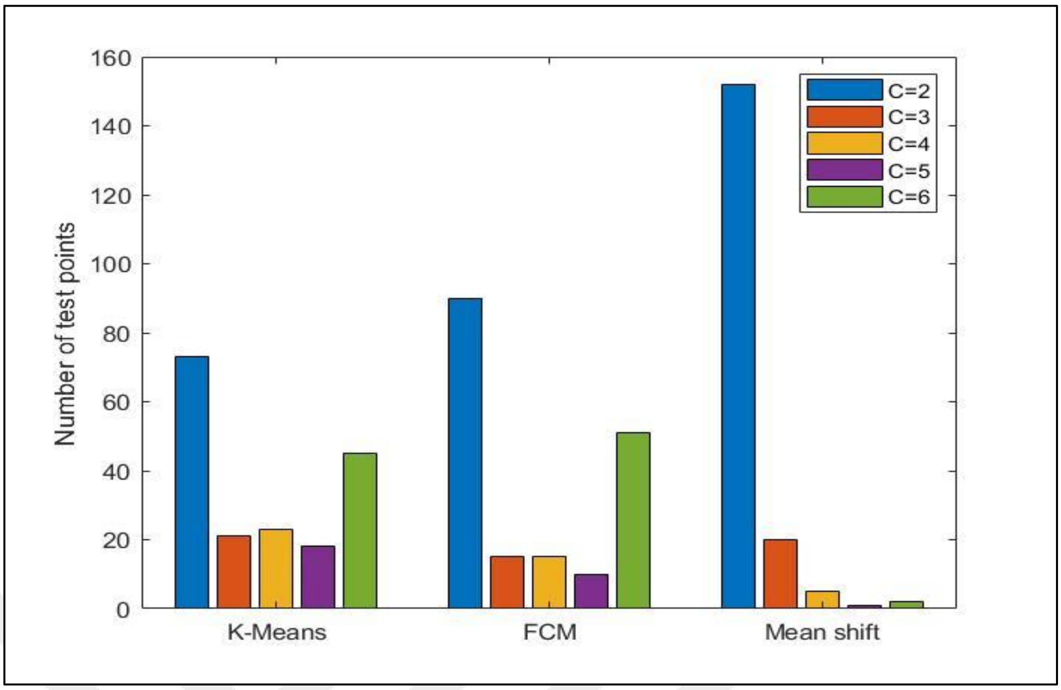

Figure 4.8 The distribution of UWB test points over clusters for the test set.

Table 4.2 Obtianed *(X_c,Y_c)* values in K-Means algorithm.

P.	***X_c***	***Y_c***	**P.**	***X_c***	***Y_c***	**P.**	***X_c***	***Y_c***
1	0.57	0.09429	**61**	0.38942	2.13096	**121**	0.58667	4.07
2	1.05375	0.005	**62**	1.02909	2.00273	**122**	1.11356	4.19724
3	1.51	0.08	**63**	1.43759	1.92862	**123**	1.60621	4.10017
4	2.11311	0.12156	**64**	2.01654	1.87436	**124**	2.037	3.992
5	2.49351	0.10521	**65**	2.27543	1.97087	**125**	2.4425	4.12438
6	2.98704	0.13898	**66**	2.95	1.885	**126**	2.82019	4.13148
7	3.63516	0.14742	**67**	3.52476	2.03857	**127**	3.4042	4.23275
8	4.06756	0.15133	**68**	4.03039	2.16779	**128**	4.03214	4.215
9	4.51063	0.07397	**69**	4.50545	2.05424	**129**	4.54512	4.14659
10	4.91029	-0.3669	**70**	5.01182	2.17424	**130**	5.01591	4.10591
11	5.53224	0.01224	**71**	5.5375	2.185	**131**	5.56945	4.20418
12	6.01471	-0.1206	**72**	6.10217	2.16217	**132**	5.91568	4.15123
13	6.6657	-0.1299	**73**	6.35067	2.14933	**133**	6.55636	4.32909
14	6.914	0.065	**74**	7.02908	1.99667	**134**	7.13	4.06
15	7.2851	-0.0254	**75**	7.30667	1.97	**135**	7.47828	4.05328
16	0.6	0.73	**76**	0.41333	2.43	**136**	0.4539	4.4478
17	1.01711	0.56868	**77**	1.00417	2.63375	**137**	1.04318	4.53716
18	1.55	0.56	**78**	1.33906	2.52672	**138**	1.54446	4.70446
19	2.11964	0.44782	**79**	2.02654	2.48962	**139**	2.1431	4.5269
20	2.61	0.592	**80**	2.37886	2.46971	**140**	2.56	4.62
21	3.02	0.47	**81**	2.876	2.327	**141**	3.00229	4.48458
22	3.53	0.52	**82**	3.485	2.49742	**142**	3.54485	4.73364
23	4.035	0.555	**83**	4.08535	2.51408	**143**	4.029	4.541
24	4.54727	0.60909	**84**	4.45821	2.67462	**144**	4.56692	4.548
25	4.97125	0.64438	**85**	5.03511	2.57389	**145**	5.06174	4.60826
26	5.6434	0.68925	**86**	5.56811	2.70324	**146**	5.64385	4.78923
27	6.06705	0.4375	**87**	6.08167	2.83024	**147**	6.146	4.6475
28	6.51545	0.53545	**88**	6.67122	2.43265	**148**	6.55	4.625
29	7.2764	0.6644	**89**	7.02533	2.57667	**149**	7.14833	4.58944
30	7.3787	0.75739	**90**	7.302	2.534	**150**	7.48378	4.59756
31	0.58389	0.94556	**91**	0.49911	3.02844	**151**	0.57321	4.965

32	1.02927	1.0978	**92**	0.97	2.98	**152**	1.05857	5.14873
33	1.475	1.1	**93**	1.58778	2.99333	**153**	1.49775	5.12596
34	2	1.02833	**94**	2.05478	2.98652	**154**	2.13548	5.15619
35	2.57473	1.08203	**95**	2.30741	3.18667	**155**	2.52935	4.99581
36	3.01929	1.00821	**96**	2.95	3.13	**156**	3.0625	5.02
37	3.54375	0.97375	**97**	3.59	3.10111	**157**	3.74581	5.00326
38	4.23696	1.1787	**98**	4.045	3.14684	**158**	4.11328	5.09688
39	4.6379	1.05597	**99**	4.57	3.13867	**159**	4.54375	5.13
40	4.98136	1.11559	**100**	5.048	3.09971	**160**	4.94391	5.11848
41	5.56571	1.03905	**101**	5.69337	3.15436	**161**	5.53054	5.03068
42	6.02625	1.0475	**102**	6.045	3.04	**162**	6.16	5.07045
43	6.54444	1.09889	**103**	6.52306	3.25163	**163**	6.52765	5.13314
44	7.02222	1.07078	**104**	6.986	3.201	**164**	7.13586	5.07207
45	7.41	1.07	**105**	7.3875	3.0425	**165**	7.37259	4.97035
46	0.52254	1.67718	**106**	0.60188	3.76688	**166**	0.57857	5.47571
47	1.05	1.55714	**107**	1.08842	3.56807	**167**	1.02229	5.54643
48	1.56118	1.53456	**108**	1.50747	3.54949	**168**	1.49333	5.46167
49	1.98463	1.40254	**109**	2.0225	3.57	**169**	2.026	5.56
50	2.6875	1.65261	**110**	2.732	3.686	**170**	2.525	5.54625
51	3.00429	1.49071	**111**	3.09481	3.62667	**171**	2.99688	5.52025
52	3.40966	1.62068	**112**	3.61	3.53714	**172**	3.55833	5.40167
53	4.24125	1.60125	**113**	4.19167	3.5275	**173**	4.06	5.5687
54	4.53333	1.67	**114**	4.51862	3.50759	**174**	4.64	5.41
55	4.8	1.7275	**115**	4.99333	3.565	**175**	5.06186	5.47763
56	5.635	1.56167	**116**	5.62038	3.56077	**176**	5.47268	5.47317
57	5.89333	1.45	**117**	5.92	3.61	**177**	6.08161	5.6225
58	6.62739	1.53848	**118**	6.61	3.67716	**178**	6.53878	5.39341
59	7.0292	1.5372	**119**	7.15929	3.57286	**179**	7.09353	5.40647
60	7.50754	1.55754	**120**	7.4375	3.6525	**180**	7.4	5.21

Table 4.3 Obtianed *(X_c, Y_c)* values in FCM algorithm.

P.	X_c	Y_c	**P.**	X_c	Y_c	**P.**	X_c	Y_c
1	0.57	0.09429	**61**	0.3887	2.13389	**121**	0.6098	4.04569
2	1.05192	0.01077	**62**	1.02909	2.00273	**122**	1.11356	4.19724
3	1.515	0.14125	**63**	1.43716	1.92875	**123**	1.59893	4.09714
4	2.11311	0.12156	**64**	2.01654	1.87436	**124**	2.03895	3.99421
5	2.49351	0.10521	**65**	2.27204	1.97082	**125**	2.43611	4.10944
6	2.98704	0.13898	**66**	2.902	1.89975	**126**	2.821	4.1292
7	3.63943	0.15571	**67**	3.52476	2.03857	**127**	3.40296	4.23408
8	4.04235	0.17353	**68**	4.04103	2.1631	**128**	4.03	4.21588
9	4.50185	0.07352	**69**	4.51552	2.04828	**129**	4.54909	4.13727
10	4.91029	-0.3669	**70**	5.056	2.164	**130**	5.01765	4.09706
11	5.54333	0.00412	**71**	5.54	2.18571	**131**	5.56509	4.20943
12	6.01667	-0.1193	**72**	6.10217	2.16217	**132**	5.91652	4.14188
13	6.66849	-0.1211	**73**	6.35067	2.14933	**133**	6.55636	4.32909
14	6.995	0.0925	**74**	7.02908	1.99667	**134**	7.1285	4.065
15	7.28195	-0.0199	**75**	7.30667	1.97	**135**	7.4731	4.0469
16	0.55618	0.81127	**76**	0.39833	2.42833	**136**	0.4519	4.44857
17	1.01711	0.56868	**77**	1.004	2.62267	**137**	1.0431	4.53655
18	1.55929	0.55857	**78**	1.35574	2.51957	**138**	1.545	4.694
19	2.08545	0.40727	**79**	2.02986	2.47743	**139**	2.14974	4.52167
20	2.62375	0.5925	**80**	2.34847	2.46972	**140**	2.54917	4.62708
21	3.01667	0.55556	**81**	2.86333	2.34333	**141**	3.00102	4.48633
22	3.53	0.52	**82**	3.49067	2.49933	**142**	3.53694	4.73429
23	4.04214	0.59179	**83**	4.08535	2.51408	**143**	4.03333	4.54
24	4.56143	0.60643	**84**	4.45821	2.67462	**144**	4.56	4.52857
25	4.96511	0.67787	**85**	5.03511	2.57389	**145**	5.06174	4.60826
26	5.6434	0.68925	**86**	5.5715	2.70175	**146**	5.64385	4.78923
27	6.06588	0.41635	**87**	5.90105	2.83816	**147**	6.14	4.64778

28	6.534	0.52267	**88**	6.66029	2.43103	**148**	6.55733	4.62733
29	7.28943	0.65914	**89**	7.00947	2.57684	**149**	7.14779	4.59029
30	7.3863	0.75704	**90**	7.31125	2.63958	**150**	7.478	4.6
31	0.58486	0.94622	**91**	0.50348	3.01	**151**	0.55579	4.94789
32	1.02927	1.0978	**92**	0.97436	3.04513	**152**	1.06529	5.13988
33	1.475	1.1	**93**	1.58765	2.98118	**153**	1.49974	5.12421
34	1.99909	1.02818	**94**	2.05	2.98333	**154**	2.13548	5.15619
35	2.57323	1.07815	**95**	2.30741	3.18667	**155**	2.52364	5.00364
36	3.01723	0.98475	**96**	2.968	3.164	**156**	3.0625	5.02
37	3.54357	0.97429	**97**	3.57792	3.10849	**157**	3.74636	5.00295
38	4.22941	1.18647	**98**	4.045	3.14684	**158**	4.11615	5.07692
39	4.63703	1.05844	**99**	4.58053	3.13526	**159**	4.53138	5.15793
40	4.94909	1.07909	**100**	5.01376	3.10835	**160**	4.95333	5.1175
41	5.5587	1.04087	**101**	5.62667	3.15778	**161**	5.5307	5.02775
42	6.0616	1.0168	**102**	6.05813	2.99875	**162**	6.15945	5.074
43	6.54604	1.09917	**103**	6.52	3.22083	**163**	6.55105	5.11895
44	7.02236	1.0727	**104**	6.98107	3.20679	**164**	7.1548	5.0748
45	7.24131	1.06949	**105**	7.39636	3.04	**165**	7.37505	4.97165
46	0.518	1.512	**106**	0.59867	3.76667	**166**	0.58333	5.47
47	1.06143	1.54	**107**	1.09743	3.56432	**167**	1.02229	5.54643
48	1.56118	1.53456	**108**	1.50818	3.54896	**168**	1.49333	5.46167
49	1.98463	1.40254	**109**	2.03457	3.59957	**169**	2.03684	5.55737
50	2.6875	1.65261	**110**	2.73	3.69737	**170**	2.525	5.53875
51	2.99655	1.48276	**111**	3.09759	3.62552	**171**	2.99628	5.51949
52	3.40931	1.61966	**112**	3.61	3.53714	**172**	3.51286	5.45
53	4.24125	1.60125	**113**	4.19091	3.52273	**173**	4.05862	5.61672
54	4.53375	1.6825	**114**	4.51903	3.50516	**174**	4.649	5.40867
55	4.792	1.737	**115**	4.98444	3.56278	**175**	5.06186	5.47763
56	5.63118	1.57353	**116**	5.62667	3.56091	**176**	5.47526	5.46868
57	5.90455	1.56727	**117**	5.95563	3.655	**177**	6.08102	5.64602
58	6.6124	1.5428	**118**	6.61	3.67716	**178**	6.53878	5.39341
59	7.02934	1.53658	**119**	7.15929	3.57286	**179**	7.096	5.411
60	7.52156	1.55667	**120**	7.45795	3.65282	**180**	7.20224	5.22092

Table 4.4 Obtianed (X_c,Y_c) values in Mean Shift algorithm.

P.	X_c	Y_c	**P.**	X_c	Y_c	**P.**	X_c	Y_c
1	0.58612	0.09959	**61**	0.40625	2.10313	**121**	0.60865	4.04712
2	1.05647	0.01529	**62**	1.02	2.02	**122**	1.11234	4.19714
3	1.65	0.12	**63**	1.515	1.9225	**123**	1.60108	4.1052
4	2.11311	0.12156	**64**	1.99088	1.87475	**124**	2.01	3.98
5	2.49351	0.10521	**65**	2.23425	1.97057	**125**	2.4303	4.12561
6	2.98	-0.1	**66**	2.95	1.885	**126**	2.81257	4.1603
7	3.618	0.1195	**67**	3.52538	2.01423	**127**	3.39789	4.23947
8	4.01	0.03	**68**	4.03063	2.17188	**128**	4.03306	4.22367
9	4.54908	0.07449	**69**	4.52098	2.055	**129**	4.55	4.11
10	4.91365	-0.3671	**70**	5.01058	2.18385	**130**	5.01621	4.10448
11	5.54333	0.00412	**71**	5.50455	2.20652	**131**	5.56684	4.19316
12	6.00038	-0.1396	**72**	6.11692	2.16538	**132**	5.91561	4.15207
13	6.67346	-0.0585	**73**	6.52	1.96	**133**	6.48	4.28
14	7.01	0.09333	**74**	7.02908	1.99667	**134**	7.1341	4.06385
15	7.2851	-0.0254	**75**	7.30667	1.97	**135**	7.46833	4.05125
16	0.6	0.73	**76**	0.43	2.45	**136**	0.46367	4.429
17	1.01056	0.56528	**77**	1.01077	2.63385	**137**	1.03548	4.54817
18	1.56727	0.56584	**78**	1.33338	2.52118	**138**	1.53887	4.70592
19	2.1492	0.4876	**79**	2.04038	2.48894	**139**	2.15088	4.52755
20	2.62902	0.60293	**80**	2.33808	2.46936	**140**	2.59	4.60667
21	3.016	0.561	**81**	2.86382	2.33971	**141**	3.01538	4.46808
22	3.53906	0.51969	**82**	3.49762	2.50286	**142**	3.595	4.725
23	4.035	0.555	**83**	4.08433	2.53788	**143**	4.04464	4.54988

24	4.548	0.61575	**84**	4.44333	2.6	**144**	4.56357	4.53429
25	4.96633	0.663	**85**	5.0389	2.57396	**145**	5.06926	4.61259
26	5.63974	0.68947	**86**	5.56933	2.69933	**146**	5.63982	4.79786
27	6.06592	0.40019	**87**	6.12	2.62	**147**	6.14526	4.64737
28	6.53313	0.52375	**88**	6.67107	2.4301	**148**	6.55606	4.6424
29	7.33062	0.65741	**89**	6.89767	2.57447	**149**	7.14926	4.58779
30	7.42536	0.7575	**90**	7.31125	2.63958	**150**	7.48512	4.59732
31	0.57929	0.93643	**91**	0.52183	2.98923	**151**	0.60273	4.89354
32	1.06333	1.06833	**92**	0.97	2.98	**152**	1.06392	5.14539
33	1.62275	0.98294	**93**	1.59533	2.99267	**153**	1.49476	5.12913
34	2.0095	1.0402	**94**	2.06	2.98262	**154**	2.13581	5.15387
35	2.57354	1.0857	**95**	2.365	3.18	**155**	2.51111	5.02778
36	3.01657	0.98608	**96**	2.95	3.13	**156**	3.0625	5.02
37	3.545	0.985	**97**	3.5931	3.09241	**157**	3.74241	5.01034
38	4.23923	1.1901	**98**	4.04	3.1	**158**	4.12268	5.08415
39	4.63955	1.06455	**99**	4.57563	3.14493	**159**	4.52667	5.1642
40	4.95891	1.11624	**100**	5.00849	3.11264	**160**	4.87	5.01
41	5.55171	1.07829	**101**	5.69186	3.15529	**161**	5.5401	5.06808
42	6.07938	0.98814	**102**	6.04	3.05	**162**	6.16814	5.06294
43	6.53524	1.09905	**103**	6.53275	3.2645	**163**	6.53627	5.12902
44	7.02571	1.07264	**104**	6.98345	3.21127	**164**	7.1548	5.0748
45	7.445	1.07	**105**	7.3875	3.0425	**165**	7.38515	4.97427
46	0.495	1.425	**106**	0.60364	3.78121	**166**	0.585	5.467
47	1.06175	1.55138	**107**	1.10624	3.56426	**167**	1.009	5.5574
48	1.535	1.46	**108**	1.49595	3.56243	**168**	1.58592	5.43379
49	2.01118	1.44118	**109**	2.01667	3.57	**169**	2.03222	5.55556
50	2.6875	1.65261	**110**	2.73163	3.71038	**170**	2.47667	5.55333
51	3.00231	1.47423	**111**	3.09948	3.63805	**171**	2.99465	5.52495
52	3.40148	1.61037	**112**	3.62026	3.53421	**172**	3.43333	5.47333
53	4.24125	1.60125	**113**	4.18814	3.57153	**173**	4.05831	5.63896
54	4.54	1.65	**114**	4.52577	3.45615	**174**	4.65467	5.43133
55	4.78243	1.75049	**115**	4.99333	3.565	**175**	5.05973	5.4824
56	5.63168	1.59871	**116**	5.6325	3.56075	**176**	5.49058	5.52835
57	5.89333	1.45	**117**	5.95824	3.66	**177**	6.08073	5.64417
58	6.63047	1.55791	**118**	6.6101	3.67673	**178**	6.52893	5.395
59	7.0427	1.47486	**119**	7.17169	3.57377	**179**	7.10832	5.40594
60	7.52806	1.5566	**120**	7.39	3.64	**180**	7.188	5.42

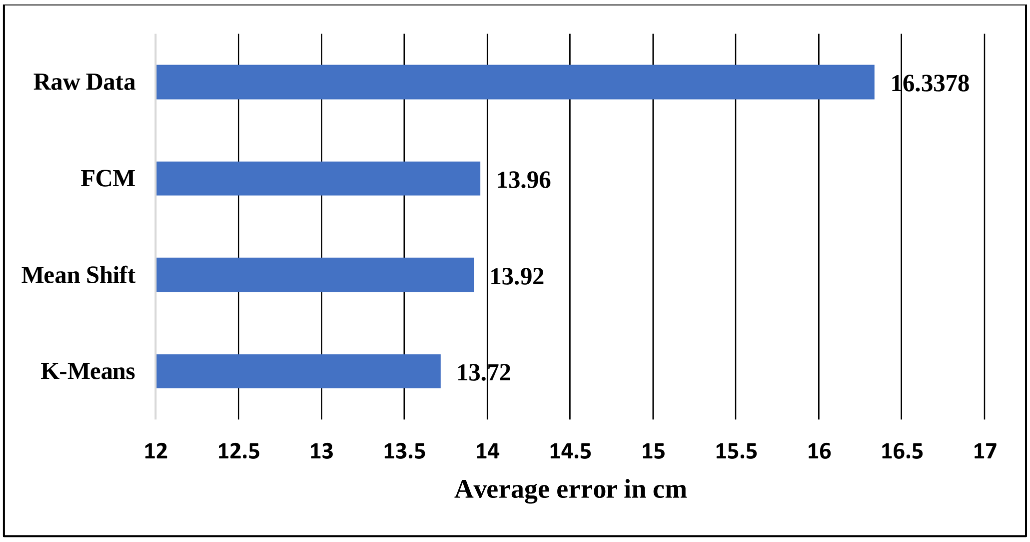

Figure 4.9 The average error comparison for the training set.

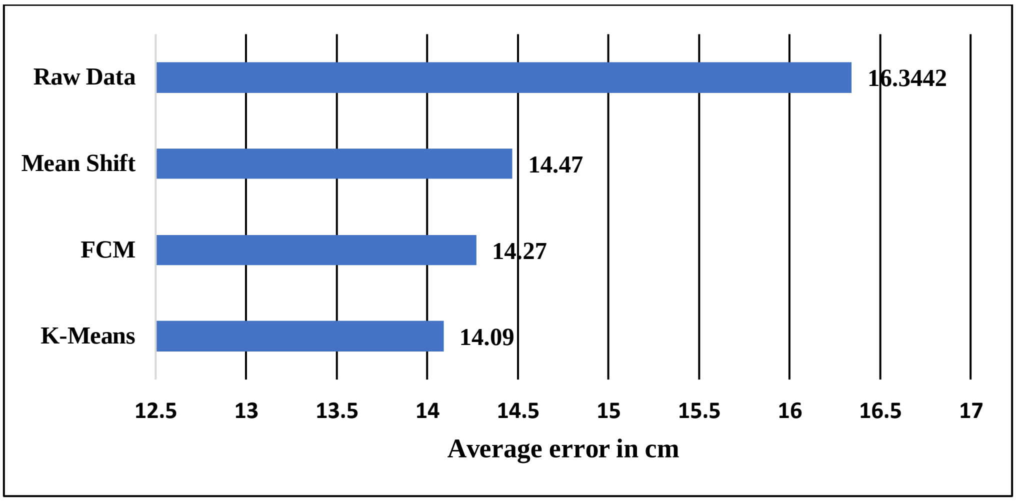

Figure 4.10 The average error comparison for the test set.

4.1.2 Clustering algorithms with Kalman Filter

To improve the accuracy of the clustering algorithms, in the second simulation the same simulation was repeated, but instead of using the row UWB measured test points, the Kalman filtered UWB test points were used as an input to the proposed system. Figure 4.11 and Figure 4.12 shows the maximum average silhouette coefficient when applying Kalman filter on training set for K-Means and FCM algorithms, respectively. The maximum average silhouette coefficient when applying Kalman filter on test set are shown in Figure 4.13 for the K-Means algorithm and Figure 4.14 for the FCM algorithm, respectively.

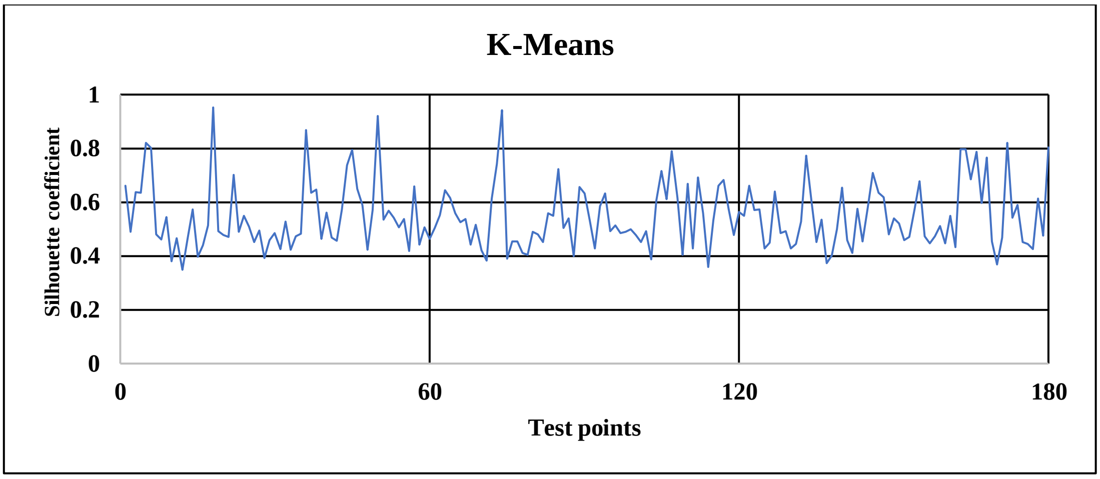

Figure 4.11 The maximum average silhouette coefficient in K-Means after applying Kalman filter for the training set.

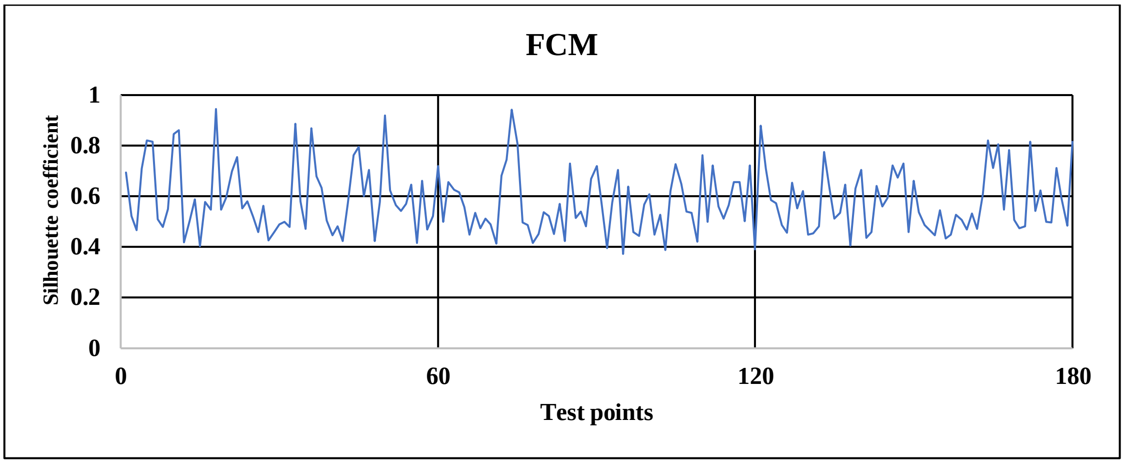

Figure 4.12 The maximum average silhouette coefficient in FCM after applying Kalman filter for the training set.

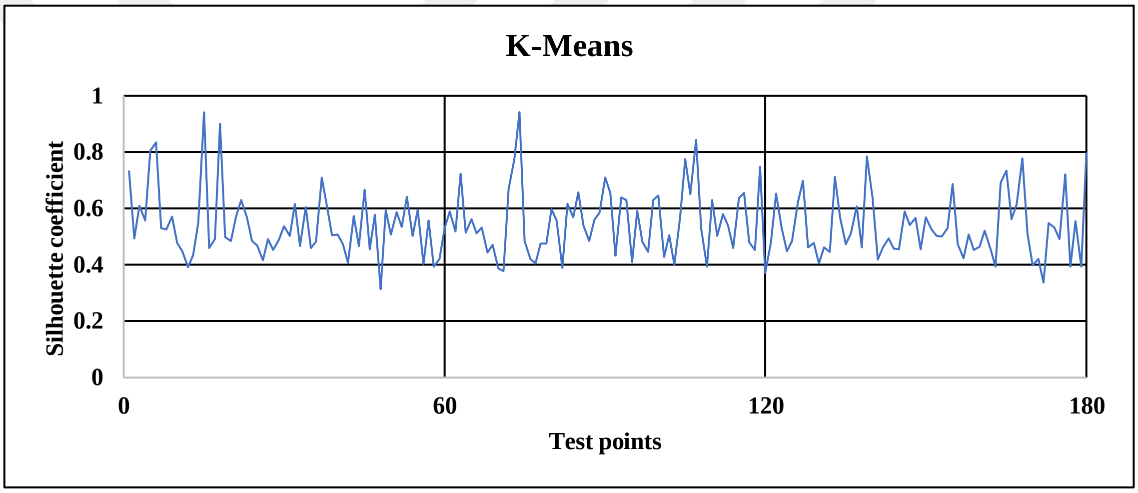

Figure 4.13 The maximum average silhouette coefficient in K-Means after applying Kalman filter for the test set.

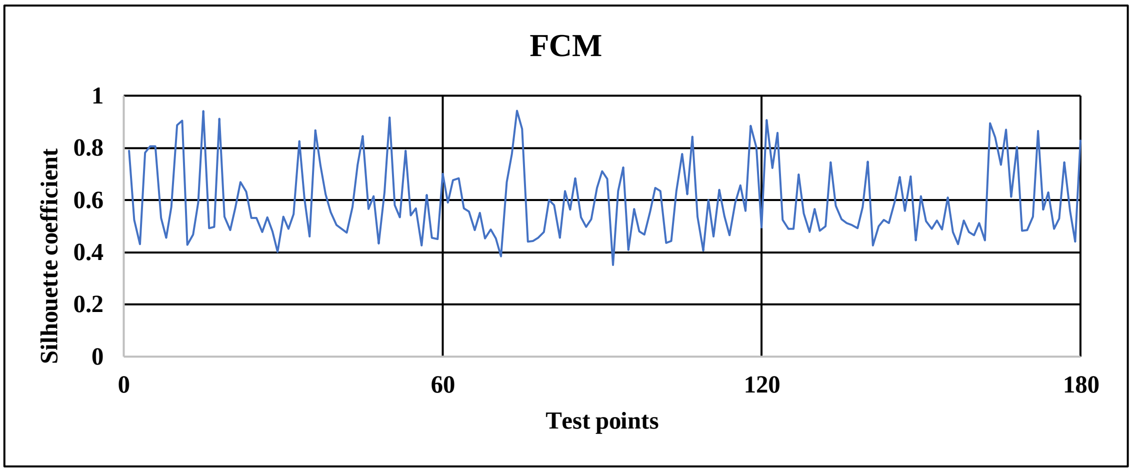

Figure 4.14 The maximum average silhouette coefficient in FCM after applying Kalman filter for test set.

The distribution of test points over clusters after applying Kalman filter for the training set and test set are shown in Figure 4.15 and Figure 4.16, respectively. Table 4.5, Table 4.6, and Table 4.7 present the *(X_c,Y_c)* values that obtained from the Kalman Filtered training set for each test point, and to be used to select the delegate cluster from the Kalman Filtered test set for the K-Means, FCM, and Mean Shift algorithms, respectively

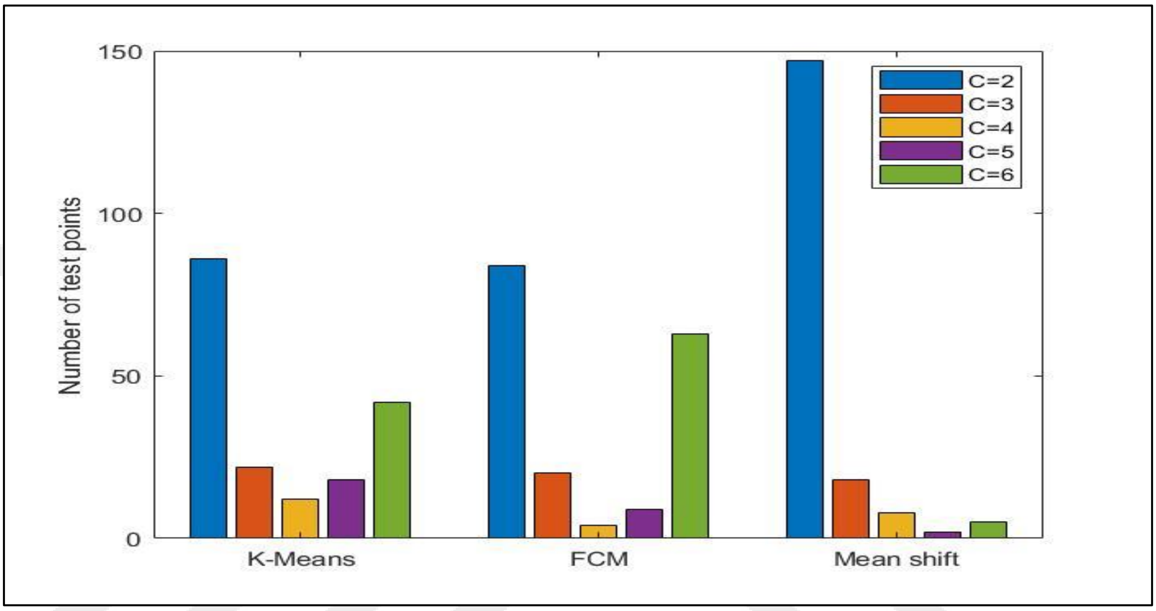

Figure 4.15 The distribution of test points over clusters after applying Kalman Filter for the training set.

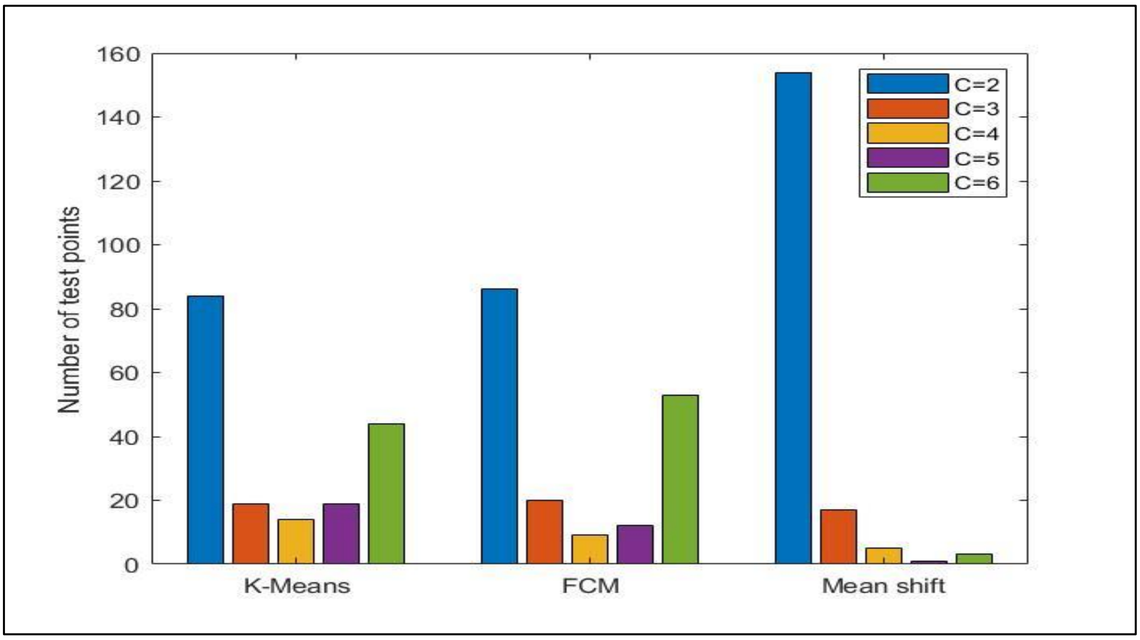

Figure 4.16 The distribution of test points over clusters after applying Kalman Filter for the test set.

Table 4.5 Obtianed *(X_c,Y_c)* values in KF K-Means algorithm.

P.	X_c	Y_c	P.	X_c	Y_c	P.	X_c	Y_c
1	0.475	0.07	**61**	0.36188	1.96708	**121**	0.54667	3.99667
2	1.02549	0.0202	**62**	1.02643	1.99143	**122**	0.98	3.96
3	1.59333	0.13333	**63**	1.43593	1.93868	**123**	1.54571	4.00429
4	1.9546	0.10747	**64**	2.02654	1.88423	**124**	2.01	3.99
5	2.50351	0.10521	**65**	2.28543	1.98087	**125**	2.41565	4.08174
6	2.99704	0.13898	**66**	2.96	1.895	**126**	2.80109	4.09652
7	3.37578	0.15822	**67**	3.49476	2.01857	**127**	3.3171	4.11823
8	4.03183	0.13817	**68**	3.94333	2.09333	**128**	3.93227	4.11273
9	4.49072	0.07398	**69**	4.48588	2.02912	**129**	4.44286	4.04762
10	4.92029	-0.3669	**70**	5.01333	2.14667	**130**	4.98444	4.06778
11	5.50364	0.00909	**71**	5.4975	2.165	**131**	5.41947	4.08711
12	6.025	-0.1083	**72**	5.95125	2.1625	**132**	5.92652	4.14188
13	6.4857	-0.1276	**73**	6.36067	2.14933	**133**	6.38091	4.21182
14	6.934	0.065	**74**	7.01908	1.99667	**134**	6.96136	3.95136
15	7.30176	-0.0147	**75**	7.325	1.9925	**135**	7.31737	3.94632
16	0.44762	0.60714	**76**	0.39833	2.43833	**136**	0.4539	4.4478
17	0.99636	0.55	**77**	0.96467	2.466	**137**	1.02794	4.50206
18	1.51	0.55	**78**	1.33906	2.52672	**138**	1.50814	4.57488
19	2.02214	0.48714	**79**	2.02986	2.48743	**139**	2.0775	4.44
20	2.4415	0.564	**80**	2.39367	2.48033	**140**	2.48333	4.5
21	3	0.47	**81**	2.87667	2.36167	**141**	2.99778	4.4975
22	3.50739	0.51913	**82**	3.50067	2.49933	**142**	3.45	4.60455
23	4.005	0.555	**83**	3.97433	2.46788	**143**	3.999	4.501
24	4.51	0.60941	**84**	4.46821	2.67462	**144**	4.51765	4.50647
25	4.97511	0.67787	**85**	4.99511	2.55389	**145**	4.96768	4.49524
26	5.51635	0.65423	**86**	5.49544	2.64412	**146**	5.4913	4.66674
27	6.01543	0.41864	**87**	6.06765	2.74603	**147**	5.98833	4.52583
28	6.527	0.516	**88**	6.50318	2.37091	**148**	6.50733	4.58733
29	7.10122	0.66061	**89**	7.03533	2.57667	**149**	6.95596	4.47263
30	7.34927	0.73455	**90**	7.312	2.534	**150**	7.30294	4.48588
31	0.56595	0.91838	**91**	0.50193	3.02246	**151**	0.55579	4.95789
32	0.94552	1.0303	**92**	0.9593	3.0257	**152**	1.03505	5.00161
33	1.45333	0.93833	**93**	1.58778	3.00333	**153**	1.45775	4.98596
34	1.9895	1.0302	**94**	2.03813	2.97	**154**	2.05619	5.05206
35	2.50473	1.05203	**95**	2.31741	3.19667	**155**	2.52962	5.00925
36	2.99723	0.97475	**96**	2.912	3.092	**156**	3.0625	5.01
37	3.51875	0.9675	**97**	3.49016	3.02111	**157**	3.64538	4.87564
38	3.93	1.09417	**98**	3.96291	3.07354	**158**	3.99818	4.97182
39	4.51377	1.03507	**99**	4.47	3.055	**159**	4.41	5.01905
40	4.97227	1.11546	**100**	4.97716	3.07642	**160**	4.94944	5.1075
41	5.50968	1.04129	**101**	5.483	3.078	**161**	5.49056	4.98775
42	6.00269	1.02538	**102**	6.00786	2.99786	**162**	5.99622	4.93892
43	6.49946	1.08786	**103**	6.34917	3.13083	**163**	6.50333	5.08455
44	6.98459	1.06255	**104**	6.92429	3.16714	**164**	7.00444	4.93333
45	7.26131	1.06949	**105**	7.3468	3.0196	**165**	7.37505	4.97165
46	0.48338	1.54706	**106**	0.54769	3.49923	**166**	0.57857	5.43571
47	1.02867	1.51	**107**	1.005	3.525	**167**	0.99013	5.39975
48	1.52118	1.49456	**108**	1.49747	3.51949	**168**	1.476	5.428
49	1.99463	1.40254	**109**	1.99	3.484	**169**	2	5.405
50	2.48739	1.52859	**110**	2.48	3.48	**170**	2.45306	5.42571
51	3.005	1.49	**111**	3.01941	3.52618	**171**	2.99	5.45833
52	3.41966	1.62068	**112**	3.51	3.455	**172**	3.509	5.422
53	3.926	1.5	**113**	4.0829	3.5371	**173**	3.94862	5.46672
54	4.42	1.61	**114**	4.5205	3.5045	**174**	4.61	5.43
55	4.79828	1.73793	**115**	4.99444	3.56278	**175**	5.02333	5.43262
56	5.485	1.54603	**116**	5.47667	3.46909	**176**	5.45058	5.48583
57	5.92381	1.56238	**117**	5.90563	3.625	**177**	5.915	5.431
58	6.49059	1.51706	**118**	6.43058	3.58165	**178**	6.52444	5.38361

59	7.00879	1.50364	**119**	7.0025	3.47667	**179**	7.0475	5.4525
60	7.31856	1.51667	**120**	7.28486	3.55541	**180**	7.22202	5.23283

Table 4.6 Obtianed *(X_c,Y_c)* values in KF FCM algorithm.

P.	X_c	Y_c	**P.**	X_c	Y_c	**P.**	X_c	Y_c
1	0.46375	0.07	**61**	0.35925	1.9734	**121**	0.58429	3.94804
2	1.02192	0.01077	**62**	1.01941	1.98	**122**	0.99941	3.93529
3	1.59333	0.13333	**63**	1.43716	1.93875	**123**	1.54533	4.006
4	1.95367	0.11111	**64**	2.02654	1.88423	**124**	2.03773	3.99682
5	2.50351	0.10521	**65**	2.28204	1.98082	**125**	2.41882	4.08
6	2.99704	0.13898	**66**	2.96	1.895	**126**	2.80109	4.09652
7	3.32833	0.07917	**67**	3.49476	2.01857	**127**	3.31443	4.124
8	4.03257	0.13829	**68**	3.93214	2.09571	**128**	3.93333	4.10667
9	4.46185	0.07352	**69**	4.49111	2.02889	**129**	4.44111	4.04556
10	4.92029	-0.3669	**70**	5.01333	2.14667	**130**	4.97688	4.0675
11	5.50235	0.00412	**71**	5.49857	2.16286	**131**	5.41945	4.09418
12	6.02471	-0.1206	**72**	6.11217	2.16217	**132**	5.92652	4.14188
13	6.4875	-0.1237	**73**	6.36067	2.14933	**133**	6.38091	4.21182
14	7.015	0.0925	**74**	7.01908	1.99667	**134**	6.96136	3.95136
15	7.3141	-0.0369	**75**	7.29848	2.09606	**135**	7.31737	3.94632
16	0.44636	0.60818	**76**	0.4	2.442	**136**	0.4519	4.44857
17	1.0019	0.5519	**77**	0.96467	2.466	**137**	1.0331	4.49655
18	1.52333	0.54	**78**	1.35574	2.51957	**138**	1.49421	4.56632
19	2.02625	0.50125	**79**	2.02986	2.48743	**139**	2.088	4.448
20	2.42902	0.55817	**80**	2.35847	2.47972	**140**	2.49038	4.50808
21	2.99667	0.55556	**81**	2.87333	2.35333	**141**	3.00102	4.49633
22	3.5	0.52	**82**	3.49765	2.49471	**142**	3.45	4.60455
23	4.01111	0.58444	**83**	3.96824	2.52324	**143**	3.999	4.501
24	4.51	0.60941	**84**	4.47923	2.67462	**144**	4.52	4.49111
25	4.97511	0.67787	**85**	4.99511	2.55389	**145**	4.9651	4.50255
26	5.51589	0.65536	**86**	5.49692	2.64554	**146**	5.5	4.66286
27	6.01588	0.41635	**87**	6.06765	2.74603	**147**	5.99091	4.52727
28	6.527	0.516	**88**	6.474	2.36875	**148**	6.50733	4.58733
29	7.13487	0.63256	**89**	7.01947	2.57684	**149**	6.95779	4.47029
30	7.29115	0.7325	**90**	7.32125	2.63958	**150**	7.30294	4.48588
31	0.56667	0.92051	**91**	0.50093	3.0163	**151**	0.55579	4.95789
32	0.95455	1.04864	**92**	0.97667	3.01833	**152**	1.02935	4.99484
33	1.50823	0.91532	**93**	1.58765	2.99118	**153**	1.45974	4.98421
34	1.97909	1.01818	**94**	2.03	2.96333	**154**	2.05784	5.04703
35	2.50323	1.04815	**95**	2.334	3.187	**155**	2.52364	5.01364
36	2.99723	0.97475	**96**	2.90444	3.08889	**156**	2.945	4.99333
37	3.51083	0.965	**97**	3.49846	3.00692	**157**	3.64636	4.87295
38	3.9225	1.08875	**98**	3.96397	3.07466	**158**	3.99579	4.96526
39	4.51358	1.03209	**99**	4.463	3.0565	**159**	4.41138	5.01793
40	4.955	1.07125	**100**	4.97376	3.07847	**160**	4.94302	5.10628
41	5.51304	1.03087	**101**	5.47667	3.076	**161**	5.49056	4.98775
42	6.0116	1.0068	**102**	6.00706	3.00294	**162**	5.98867	4.94467
43	6.49604	1.08917	**103**	6.35462	3.16231	**163**	6.50105	5.07895
44	6.96236	1.0627	**104**	6.93444	3.18	**164**	6.9648	4.9349
45	7.26131	1.06949	**105**	7.35105	3.01947	**165**	7.37505	4.97165
46	0.48246	1.56101	**106**	0.5475	3.49833	**166**	0.57857	5.43571
47	1.02867	1.51	**107**	1.005	3.525	**167**	0.99133	5.398
48	1.52118	1.49456	**108**	1.50067	3.51317	**168**	1.48333	5.42
49	1.99463	1.40254	**109**	1.99471	3.51647	**169**	2	5.41727
50	2.48739	1.52859	**110**	2.516	3.464	**170**	2.46042	5.41458
51	2.99655	1.48276	**111**	3.00923	3.52538	**171**	2.97681	5.47819
52	3.41931	1.61966	**112**	3.52045	3.46091	**172**	3.51286	5.46
53	3.926	1.5	**113**	4.0829	3.5371	**173**	3.94862	5.46672
54	4.41632	1.64421	**114**	4.51903	3.50516	**174**	4.612	5.405

55	4.802	1.737	**115**	4.9925	3.55917	**175**	5.02186	5.43763
56	5.47971	1.54086	**116**	5.48341	3.46854	**176**	5.43649	5.42703
57	5.92455	1.56727	**117**	5.90563	3.625	**177**	5.92093	5.49505
58	6.49059	1.51706	**118**	6.43653	3.57707	**178**	6.52878	5.38341
59	6.96934	1.52658	**119**	6.99524	3.48095	**179**	7.05333	5.39667
60	7.32095	1.51674	**120**	7.28706	3.54235	**180**	7.22224	5.23092

Table 4.7 Obtianed *(X_c,Y_c)* values in KF Mean Shift algorithm.

P.	***X_c***	***Y_c***	**P.**	***X_c***	***Y_c***	**P.**	***X_c***	***Y_c***
1	0.49	0.07	**61**	0.3792	1.936	**121**	0.58885	3.93712
2	1.02647	0.01529	**62**	1.01	2	**122**	1.00053	3.93
3	1.61	0.12	**63**	1.515	1.9325	**123**	1.56108	3.9952
4	1.95367	0.11111	**64**	1.99528	1.88444	**124**	2.01	3.99
5	2.50351	0.10521	**65**	2.24425	1.98057	**125**	2.4103	4.09561
6	2.99	-0.1	**66**	2.96	1.895	**126**	2.79257	4.1303
7	3.34421	0.10895	**67**	3.49538	1.99423	**127**	3.30789	4.12816
8	3.9	0.03	**68**	3.92063	2.11188	**128**	3.92306	4.11367
9	4.50908	0.07449	**69**	4.48098	2.035	**129**	4.43	4
10	4.92365	-0.3671	**70**	5.05	2.15	**130**	4.97621	4.07448
11	5.50235	0.00412	**71**	5.46455	2.18652	**131**	5.41684	4.08316
12	6.01038	-0.1396	**72**	6.12692	2.16538	**132**	5.92561	4.15207
13	6.49136	-0.0523	**73**	6.53	1.96	**133**	6.39333	4.21667
14	7.03	0.09333	**74**	7.01908	1.99667	**134**	6.9441	3.95385
15	7.3051	-0.0254	**75**	7.32667	1.97	**135**	7.2963	3.94543
16	0.44654	0.61577	**76**	0.43	2.46	**136**	0.46367	4.429
17	0.98742	0.55303	**77**	0.95359	2.45487	**137**	1.02548	4.50817
18	1.52	0.54	**78**	1.33338	2.52118	**138**	1.49875	4.57597
19	1.99203	0.45342	**79**	2.04038	2.49894	**139**	2.1	4.48
20	2.42919	0.55919	**80**	2.34808	2.47936	**140**	2.48373	4.515
21	3	0.495	**81**	2.87382	2.34971	**141**	3.01538	4.47808
22	3.50906	0.51969	**82**	3.50762	2.50286	**142**	3.495	4.595
23	4.005	0.555	**83**	3.97433	2.46788	**143**	4.01464	4.50988
24	4.50874	0.61476	**84**	4.45333	2.6	**144**	4.52357	4.49429
25	4.97633	0.663	**85**	4.9989	2.55396	**145**	4.96756	4.4941
26	5.51328	0.65746	**86**	5.48773	2.64467	**146**	5.48982	4.66786
27	6.01592	0.40019	**87**	5.96	2.55	**147**	5.9827	4.54
28	6.52313	0.52375	**88**	6.49107	2.36583	**148**	6.50606	4.6024
29	7.13313	0.63713	**89**	6.90767	2.57447	**149**	6.95926	4.46779
30	7.31143	0.7339	**90**	7.32125	2.63958	**150**	7.30294	4.48588
31	0.58571	0.94881	**91**	0.52183	2.98923	**151**	0.60273	4.90354
32	0.98333	0.98833	**92**	0.95779	3.03115	**152**	1.03392	5.00539
33	1.50284	0.91167	**93**	1.59533	3.00267	**153**	1.45476	4.98913
34	1.9895	1.0302	**94**	2.04	2.96262	**154**	2.05892	5.0477
35	2.50354	1.0557	**95**	2.375	3.19	**155**	2.53719	4.99156
36	2.99657	0.97608	**96**	2.87	3.05	**156**	3.0625	5.01
37	3.515	0.975	**97**	3.49	3.008	**157**	3.64241	4.88034
38	3.91923	1.1001	**98**	3.93	3.02	**158**	3.99739	4.96522
39	4.51727	1.03455	**99**	4.45563	3.06493	**159**	4.40667	5.0242
40	4.96891	1.11624	**100**	5.01769	3.07	**160**	4.87	5
41	5.50514	1.05457	**101**	5.54137	3.07363	**161**	5.49885	5.02808
42	6.02938	0.97814	**102**	6.0174	2.99173	**162**	6.00676	4.92608
43	6.48524	1.08905	**103**	6.35923	3.17436	**163**	6.48627	5.08902
44	6.96571	1.06264	**104**	6.92345	3.18127	**164**	6.9648	4.9349
45	7.25755	1.0717	**105**	7.34638	3.02652	**165**	7.38515	4.97427
46	0.47243	1.59146	**106**	0.5598	3.50111	**166**	0.585	5.427
47	1.03175	1.51138	**107**	1.005	3.525	**167**	0.9794	5.4074
48	1.51311	1.51	**108**	1.48595	3.53243	**168**	1.57592	5.39379
49	2.02118	1.44118	**109**	1.99583	3.48667	**169**	2.006	5.4036
50	2.48739	1.52859	**110**	2.48	3.48	**170**	2.45324	5.41422

51	3.00231	1.47423	**111**	3.01948	3.53805	**171**	2.97416	5.48564
52	3.41148	1.61037	**112**	3.51793	3.46069	**172**	3.43333	5.48333
53	3.92	1.47167	**113**	4.07814	3.47525	**173**	3.95	5.40765
54	4.42	1.61	**114**	4.52577	3.45615	**174**	4.61467	5.39133
55	4.79243	1.75049	**115**	5.00333	3.565	**175**	5.01973	5.4424
56	5.5175	1.5425	**116**	5.4825	3.469	**176**	5.45058	5.48583
57	5.91333	1.45	**117**	5.90824	3.63	**177**	5.92073	5.49417
58	6.46419	1.50371	**118**	6.4301	3.57673	**178**	6.51893	5.385
59	6.9827	1.46486	**119**	7.002	3.4772	**179**	7.0475	5.4525
60	7.32806	1.5166	**120**	7.27683	3.55394	**180**	7.208	5.43

The average error comparison after applying Kalman Filter for the training set and test set are shown in Figure 4.16 and Figure 4.17, respectively.

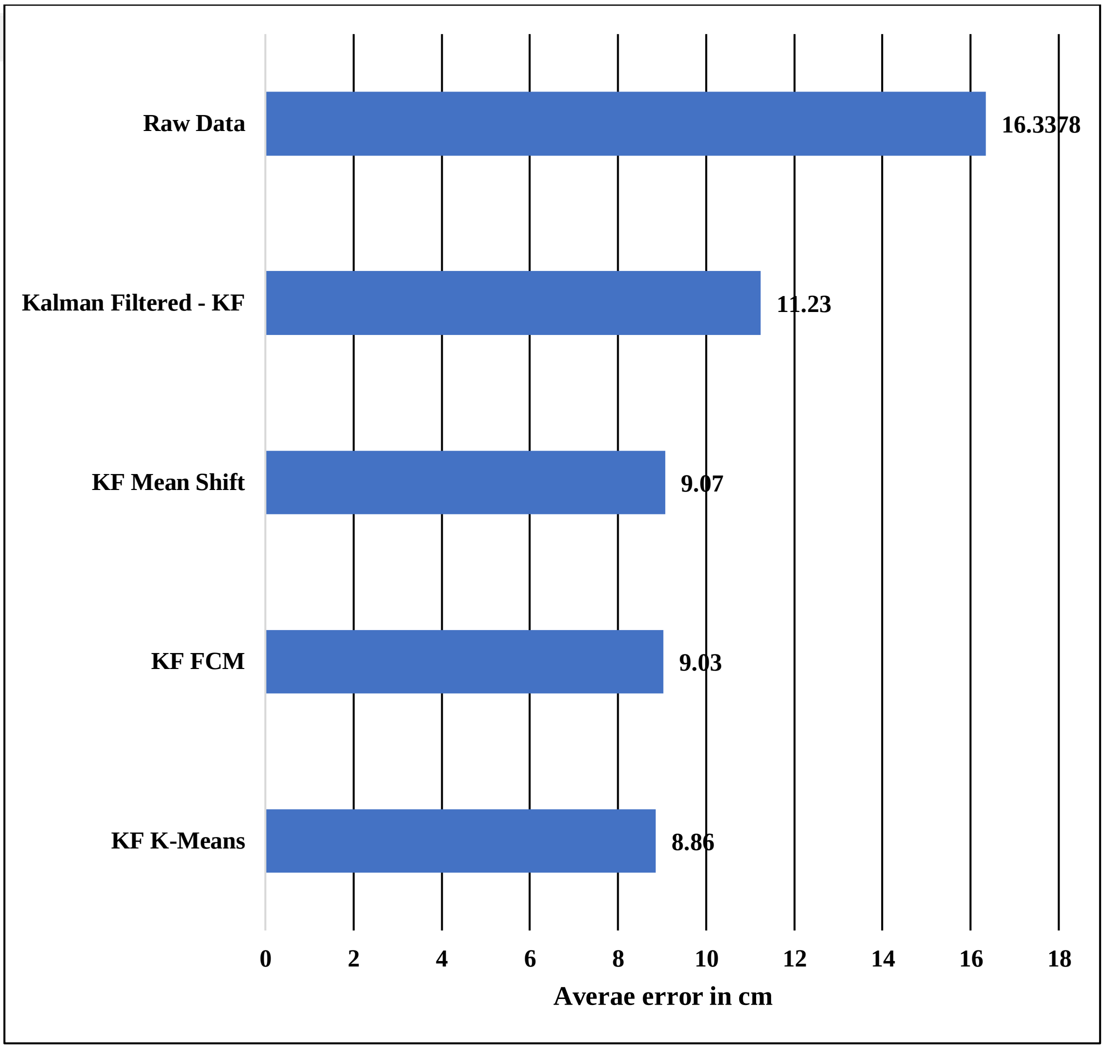

Figure 4.17 The average error comparison after applying Kalman Filter for the training set.

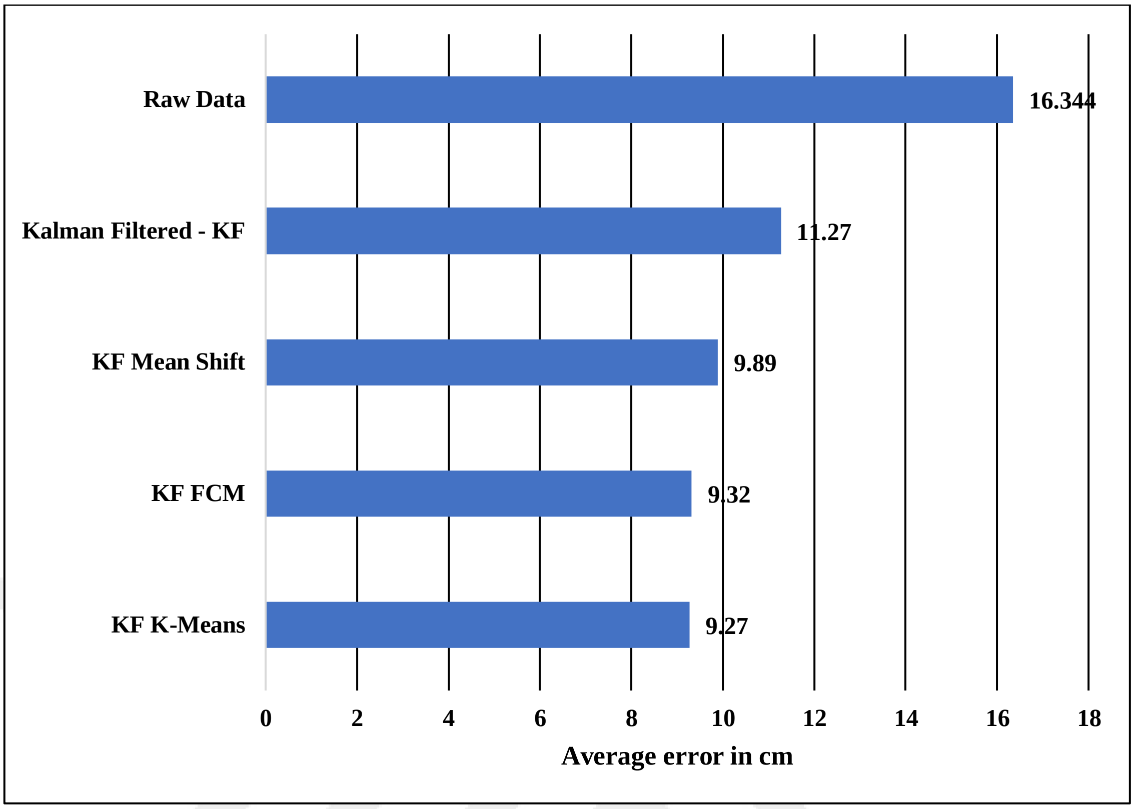

Figure 4.18 The average error comparison after applying Kalman Filter for the test set.

As shown in Figure 4.18 the results were significantly improved, and again, the K-Means algorithm outperform both FCM and Mean Shift algorithms.

In the final simulation, in which the Kalman Filter was implemented using the output data from the clustering algorithms as an input, produced very poor results. For example, when the Kalman Filter is implemented by using the output data that acquired from applying K-Means algorithm, the average location error increased from (16.34 cm to 16.5 cm).

Table 4.8 Computation time comparison of clustering simulations.

Simulation	**Computation time in seconds**
K-Means	844.907 for training
	861.48 for test
FCM	990.043 for training
	997.899 for test
Mean Shift	1081.651 for training
	1103.171 for test

Table 4.9 Computation time comparison of clustering simulations with K.F.

Simulation	Computation time in seconds
KF then K-Means	1182.085 for training
	1274.91 for test
KF then FCM	1296.662 for training
	1381.319 for test
KF then Mean Shift	1360.853 for training
	1388.911 for test

4.2 The Hybrid Algorithm

A Hybrid (BB-BC KF K-Means) algorithm was applied on UWB test points. Since, the best result obtained in term of optimization was obtained when applying BB-BC algorithm. In which, Kalman Filter was applied to the UWB optimized BB-BC test points. The average location error was reduced by 54.53 %. Whereas, the best result in term of clustering was obtained from performing K-Means algorithm. In which the average location error was reduced by 13.77 %. Thus, using such Hybrid algorithm, will reduced the average location error significantly.

As a result of using the Hybrid algorithm, the average location error was reduced by approximately 64.26 % (from 16.34 cm to 5.84 cm). Figure 4.19 and Figure 4.20 shows the maximum average silhouette coefficient when applying K-Means clustering algorithm on Kalman Filtered BB-BC optimized UWB test points for the training and test set, respectively. Figure 4.21 shows the optimal distribution of test points (180 test points) over clusters for the training and test set. Table 4.8 show the (X_c,Y_c) values that obtained from the training set for each test point, and to be used to select the delegate cluster from the test set. While Figure 4.22 and Figure 4.23 show the improvement in accuracy of the Hybrid algorithm over the best results obtained from the implementation of different simulations in both, the optimization and clustering algorithms for the training and test set, respectively.

Table 4.7 shows the computation time of implementing the hybrid algorithm for the training and test set.

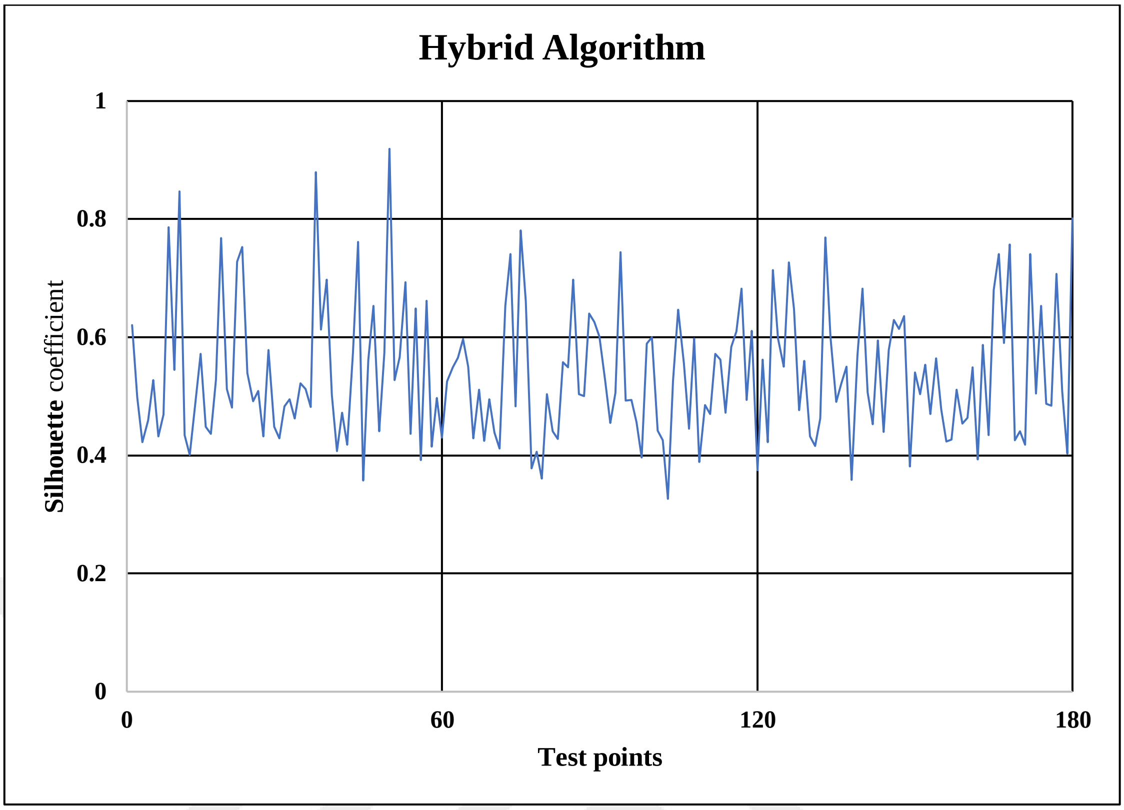

Figure 4.19 The maximum average silhouette coefficient in Hybrid Algorithm for the training set.

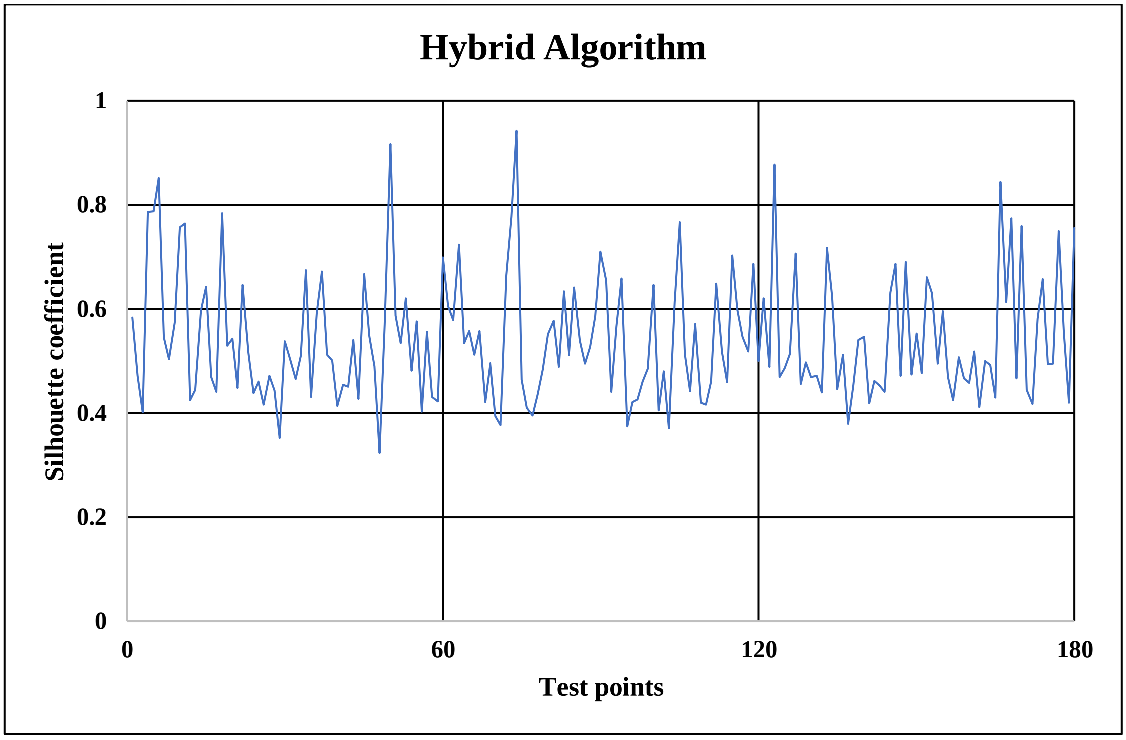

Figure 4.20 The maximum average silhouette coefficient in Hybrid Algorithm for the test set.

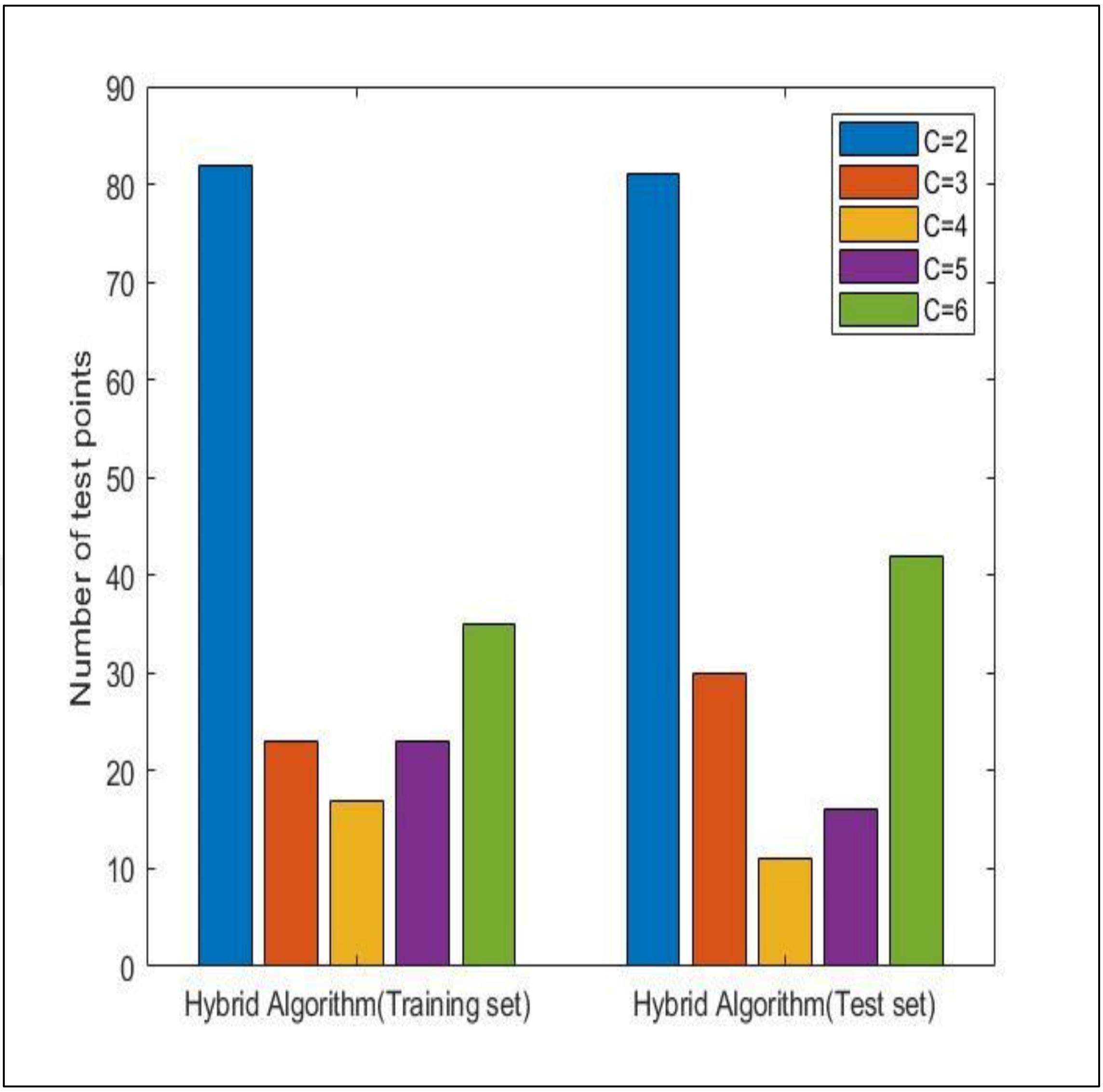

Figure 4.21 The distribution of test points over clusters in Hybrid algorithm.

Table 4.10 obtianed *(X_c,Y_c)* values in Hybrid algorithm.

P.	X_c	Y_c	**P.**	X_c	Y_c	**P.**	X_c	Y_c
1	0.5	0.08	**61**	0.4428	1.9928	**121**	0.49205128	3.971026
2	1.01375	0.005	**62**	0.99	1.987143	**122**	1.03714286	4.050714
3	1.48	-0.01	**63**	1.50229885	1.978966	**123**	1.525	3.98
4	1.97417722	0.050759	**64**	1.99653846	1.984359	**124**	2.00181818	3.998182
5	2.48351064	0.095213	**65**	2.51204082	1.980816	**125**	2.5	3.999655
6	2.96625	0.045	**66**	2.99560976	2.002683	**126**	2.96583333	3.945
7	3.41828571	0.061429	**67**	3.48463415	1.987073	**127**	3.41146667	4.0176
8	4.00647887	0.108451	**68**	3.99075472	1.999623	**128**	3.99	4.015
9	4.45072289	0.063976	**69**	4.5052	1.992	**129**	4.49111111	4.018889
10	5.12	0.083333	**70**	4.98	2.075714	**130**	4.98333333	4.022667
11	5.50444444	0.077778	**71**	5.47894737	1.993158	**131**	5.46083333	4.030278
12	6.00222222	0.075556	**72**	6.07217391	2.092174	**132**	6.01	4.075
13	6.32849315	0.158904	**73**	6.34366667	1.953444	**133**	6.43561798	4.031011
14	6.92016129	0.082581	**74**	6.87869565	1.93587	**134**	7	4.001579
15	7.38333333	-0.096667	**75**	7.32475248	1.966634	**135**	7.30804878	3.986341

16	0.44666667	0.495833	**76**	0.38111111	2.398444	**136**	0.44605263	4.388684
17	1.00068966	0.492069	**77**	0.98702703	2.503784	**137**	1.02793651	4.482063
18	1.5	0.49	**78**	1.5190625	2.466719	**138**	1.51448276	4.495172
19	1.99710526	0.525526	**79**	2.01774194	2.510968	**139**	1.9996	4.4968
20	2.49133333	0.517333	**80**	2.49424242	2.499697	**140**	2.50852941	4.503235
21	2.98895833	0.480521	**81**	2.87333333	2.343333	**141**	2.98745455	4.474182
22	3.49043478	0.48087	**82**	3.505	2.48625	**142**	3.50045455	4.494545
23	3.99719512	0.483902	**83**	3.99432692	2.507885	**143**	4.00210526	4.492895
24	4.50333333	0.581667	**84**	4.50820513	2.484615	**144**	4.50344828	4.486897
25	4.926	0.60225	**85**	4.95511111	2.543889	**145**	4.99768293	4.505244
26	5.49166667	0.61	**86**	5.43544118	2.514118	**146**	5.48285714	4.499286
27	5.96636364	0.504432	**87**	6.01382353	2.593235	**147**	5.98	4.516667
28	6.50076923	0.496154	**88**	6.47106796	2.480097	**148**	6.50137931	4.5
29	6.92882353	0.477647	**89**	6.97533333	2.536667	**149**	6.96833333	4.479444
30	7.32928571	0.666071	**90**	7.38333333	2.573333	**150**	7.26277778	4.506667
31	0.50043478	1.01029	**91**	0.50514706	3.006765	**151**	0.55321429	4.945
32	0.98791667	1.012222	**92**	0.96909091	3.006023	**152**	0.99428571	4.962619
33	1.53839286	0.963929	**93**	1.49769231	2.983846	**153**	1.49775281	4.985955
34	2	0.998889	**94**	2	2.997647	**154**	2.05619048	4.992063
35	2.51	1.030645	**95**	2.29740741	3.066667	**155**	2.5	5.002857
36	2.99722772	0.984752	**96**	2.9652381	3.050476	**156**	2.95712871	4.962673
37	3.50170732	0.998293	**97**	3.47595238	2.994286	**157**	3.45581395	4.973256
38	4.01	1.109	**98**	3.97162162	3.023243	**158**	4.01170213	5.000213
39	4.51955224	1.03209	**99**	4.49133333	3.004667	**159**	4.48	4.989048
40	4.97236364	0.988545	**100**	5.04866667	3.027333	**160**	4.94333333	4.9675
41	5.45327869	0.982787	**101**	5.51846154	3.040769	**161**	5.47870968	5.06
42	5.97578947	0.953947	**102**	6	3.003333	**162**	6.018	5.018
43	6.4705	0.991667	**103**	6.44777778	2.997037	**163**	6.48095238	4.99
44	7.11333333	1.016667	**104**	7.0025	2.996875	**164**	6.92142857	4.990714
45	7.42	0.99	**105**	7.34348837	2.999535	**165**	7.31928571	4.982143
46	0.45794118	1.516765	**106**	0.5575	3.458333	**166**	0.47541667	5.362083
47	0.98	1.49	**107**	1.00368421	3.492632	**167**	1.01845238	5.380952
48	1.4672973	1.493514	**108**	1.50746835	3.499494	**168**	1.51040816	5.384082
49	2.00462687	1.502537	**109**	2.00735294	3.505588	**169**	2.02659091	5.405455
50	2.5175	1.561522	**110**	2.504	3.488	**170**	2.50346535	5.394356
51	3.00347826	1.5	**111**	3.00481481	3.496667	**171**	2.97961538	5.405
52	3.47608696	1.517174	**112**	3.52	3.49	**172**	3.4494898	5.316735
53	4.00125	1.52125	**113**	4.00289855	3.527101	**173**	3.99888889	5.398889
54	4.48631579	1.602632	**114**	4.46862069	3.497586	**174**	4.5	5.41
55	4.79911111	1.632222	**115**	4.99	3.503333	**175**	4.976875	5.398125
56	5.48740741	1.499259	**116**	5.49683544	3.512152	**176**	5.43848485	5.402727
57	5.97638554	1.483735	**117**	5.95516854	3.480899	**177**	5.95092784	5.375052
58	6.45075	1.52725	**118**	6.43872549	3.557157	**178**	6.48434783	5.328261
59	6.89934211	1.526579	**119**	6.92222222	3.508889	**179**	6.99071429	5.403571
60	7.3225	1.545	**120**	7.31584906	3.500943	**180**	7.37	5.204

Table 4.11 Computation time of the Hybrid algorithm

Simulation	**Computation time in seconds**
Hybrid algorithm	1836.979 for training
	1876.08 for test

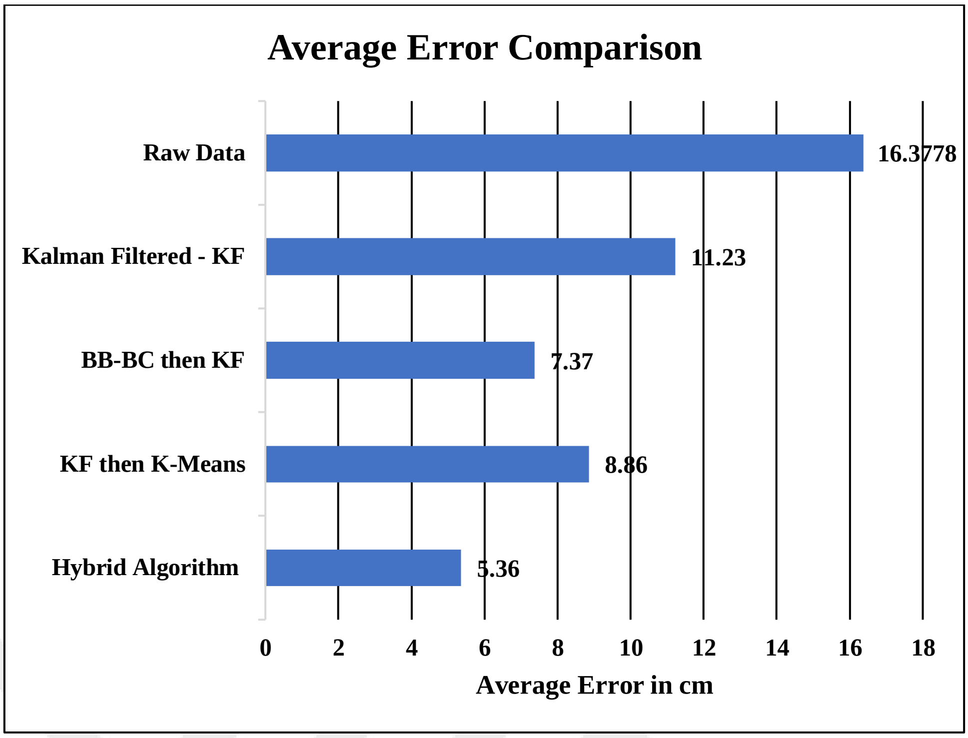

Figure 4.22 The Accuracy of Hybrid Algorithm for the training set.

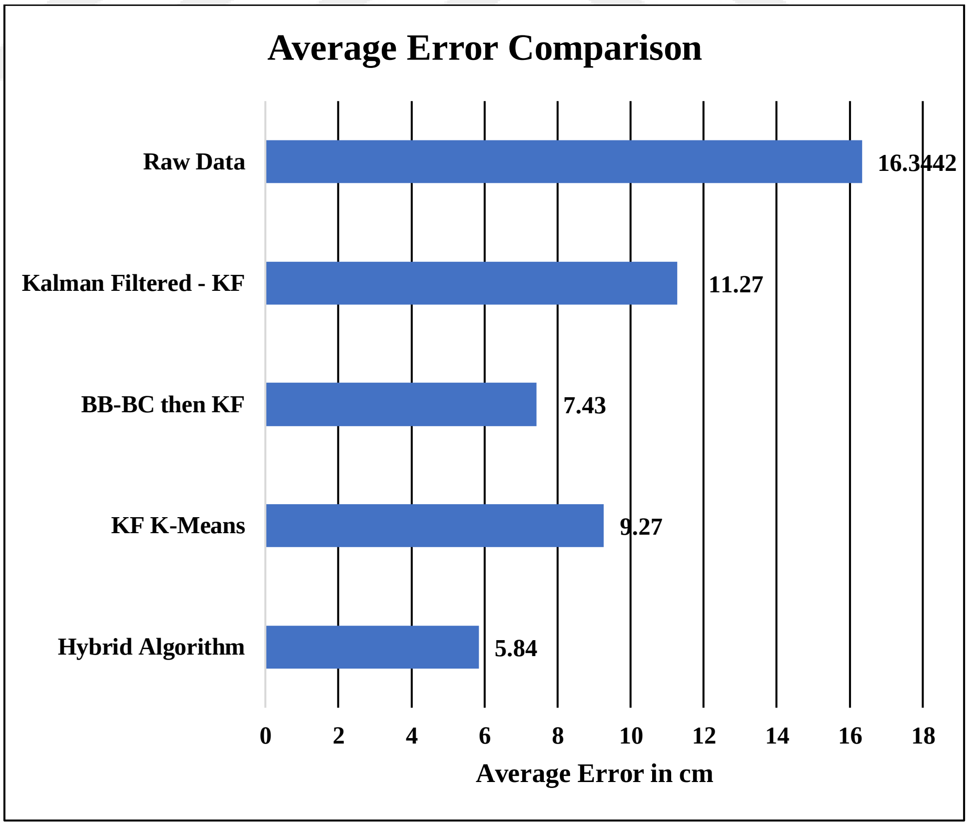

Figure 4.23 The Accuracy of Hybrid Algorithm for the test set.

5. CONCLUSION

5.1 Optimization Algorithms

Big bang big crunch (BB-BC) and Genetic algorithms were employed to increase the accuracy of UWB indoor positioning system, in which the ALC dataset was used for this purpose. As conclusion, the BB-BC algorithm outperform GA algorithm in all three performed simulations.

In the first simulation, the raw UWB test points were used an input to the BB-BC and GA algorithms. As a result, the BB-BC algorithm reduces the average location by 48.16 % (from 16.34 cm to 8.47 cm). Whereas, the GA algorithm manage to reduce the average location error by only 31.76 % (from 16.34 cm to 11.15 cm).

In the second simulation, the Kalman Filtered UWB test points were used as input to BB-BC and GA algorithms. As a result, the BB-BC algorithm was able to reduces the average location by 51.29 % (from 16.34 cm to 7.96 cm). While the GA algorithm reduced the average location error by 46.57 % (from 16.34 cm to 8.73 cm).

In the final simulation, just like the first simulation, the raw UWB test points were used an input to the BB-BC and GA algorithms, then, the optimized UWB test points were used as input to the Kalman Filter. As a result, the BB-BC algorithm was able to reduce the average location error by approximately 54.53 % (from 16.34 cm to 7.43 cm). Whereas the GA algorithm only reduced the average location error by approximately 52.08 % (from 16.34 cm to 7.83 cm).

The limitation of the GA algorithm is the slow convergence in term of reaching the optimal result, where in our case the optimal offset value. Whereas, the BB-BC overcome this drawback, since it offers speed convergence when reaching to the optimal value. Applying Kalman Filter reduced the average location and produce more optimized results. However, the Kalman Filter produce better result when its applied on UWB optimized test points, whether it’s been optimized by BB-BC or GA algorithms, when we compare it to the results obtained when applying Kalman Filter on the Raw UWB test points, then using the output data as input to the optimization algorithms.

5.2 Machine Learning Algorithms

Three machine learning clustering algorithms are compared in terms of accuracy using ALC dataset. The aim was to find the most appropriate clustering algorithm for indoor positioning problem via UWB in term of accuracy.

As a conclusion, The K-Means algorithm is superior to all other methods, with highest accuracy (14.09 cm) for the test set, especially when average silhouette method was utilized to determine the optimal number of clusters. Whereas Mean Shift algorithm has the lowest accuracy, (14.47 cm), when it's compared with K-Means and FCM algorithms, despite its advantage. The main advantages of Mean Shift algorithms rise from the nonparametric nature of the kernel density estimate (KDE) and the user need only to set one parameter, the bandwidth. Which is often more convenient than having to select the number of clusters explicitly or utilizing other methods to define the number of clusters such as the average silhouette or the Elbow methods.

FCM algorithm has accuracy of (14.27 cm), which is very close to the result that was obtained from K-Means algorithm. However, FCM algorithm tend to run slower when we compare it with K-Means, because more work is done during the processes. Where each data point is been evaluated with each cluster, and with each evaluation more operations are involved. FCM needs to do a full inverse-distance weighting, whereas K-Means just needs to do a distance calculation. Thus, K-Means is simpler and computationally faster.

The impact of using Kalman Filter on the measured UWB test points is also introduced when applying the clustering algorithms. As an advantage, the accuracy was enhanced significantly, where the average location error reduced by approximately 31.03%.

Finally, the Kalman Filtered UWB data were applied as input to the clustering algorithm, the best result was obtained from K-Means algorithm, in which the average error reduced by 43.27% (from 16.34 cm to 9.27 cm). Based on the obtained results from the clustering algorithms, K-Means were the most appropriate one for indoor positioning system, due to its high accuracy, simplicity and fast computations.

5.3 Comparison Between the Optimization and Machine Learning Algorithms

In term of compression among the applied method in this work, Figure 5.1 shows the average location error when applying the optimization algorithms and machine learning algorithms for the test set. In which the raw UWB test points were used as an input to the applied algorithms.

The best result obtained when using the BB-BC optimization algorithm, since it produces the highest accuracy (8.47 cm).

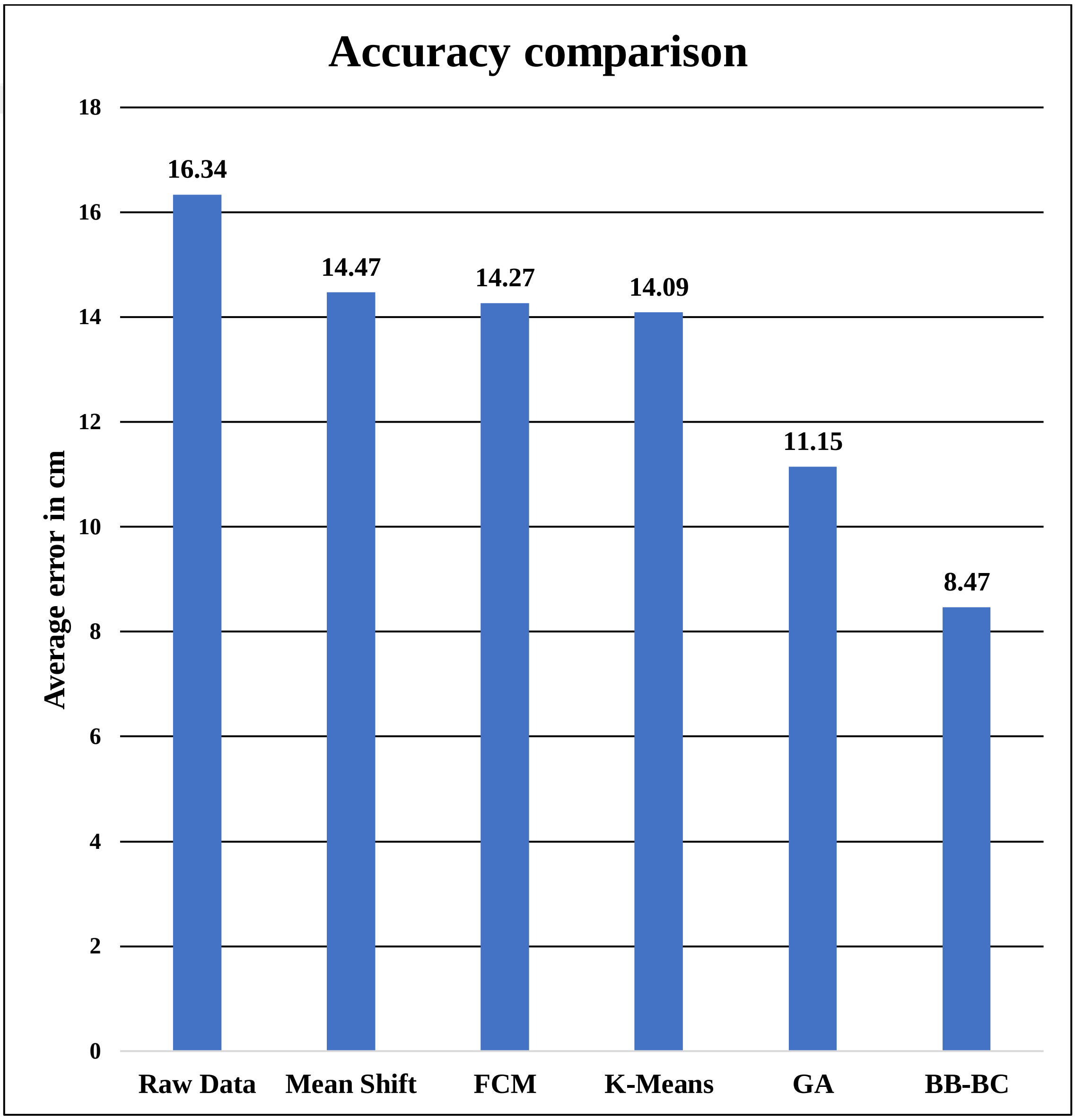

Figure 5.1 Accuracy Comparison of the optimization and Machine Learning algorithms using UWB test points for the test set.

Figure 5.2 shows the average location error when applying the optimization algorithms and machine learning algorithms. In which the Kalman Filtered UWB test points were used as an input to the applied algorithms for the test set.

The best result obtained when using the BB-BC optimization algorithm, since it has the highest accuracy (7.96 cm).

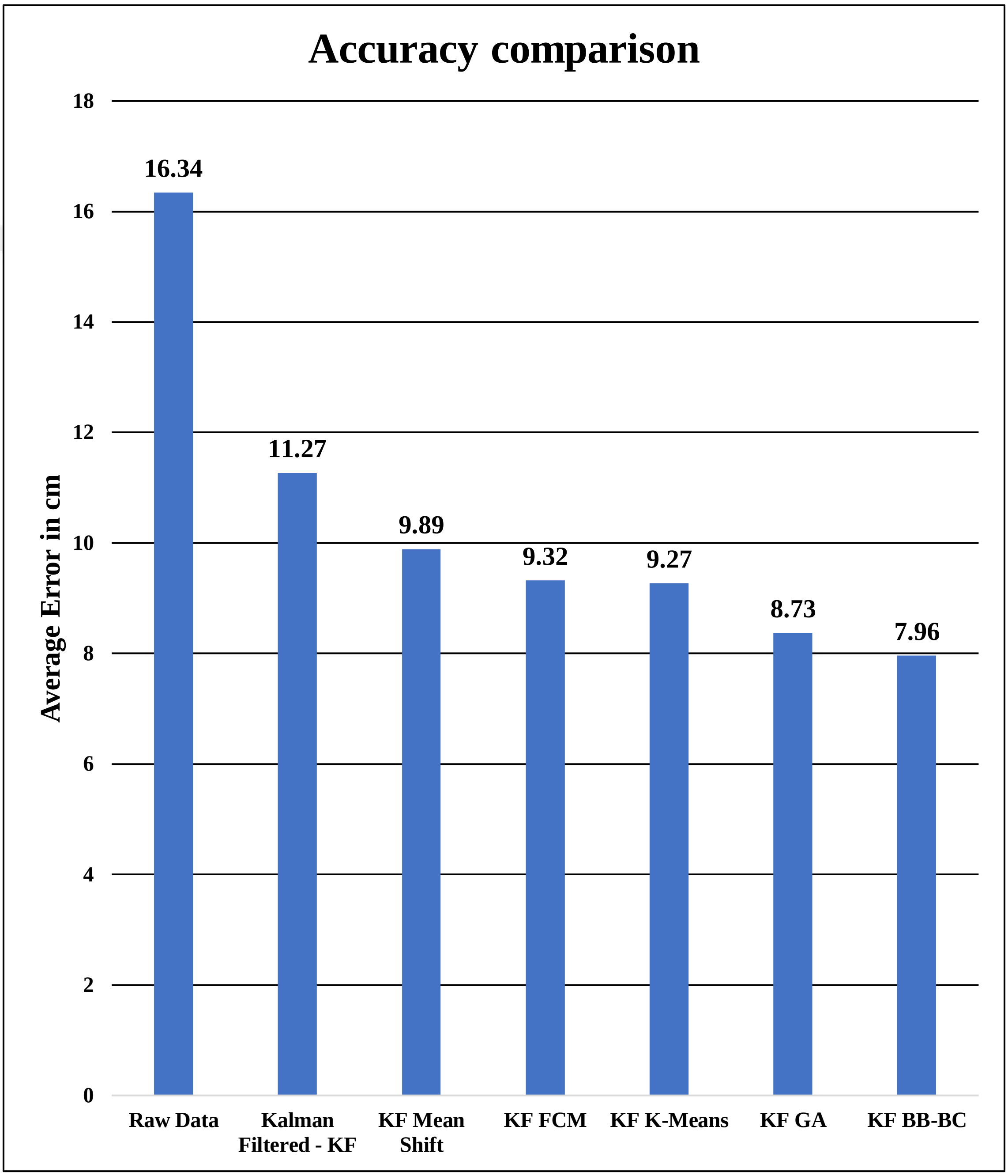

Figure 5.2 Accuracy Comparison of the optimization and Machine Learning algorithms using KF UWB test points for the test set.

Finally, a Hybrid (BB-BC KF K-Means) algorithm was implemented. In which the raw UWB test points go through three stages: (1) the implementation of BB-BC algorithm; (2) the implantation of Kalman Filter; (3) the implementation of K-Means algorithm. As expected, the results were significantly improved. In which the average location error was reduced by approximately 64.26 % (from 16.34 cm to 5.84 cm) for the test set.

5.4 Suggestions for Future Work

In order to develop the current implemented work, the following suggestions are presented:

(i) Investigate other methods to define the optimal number of clusters in clustering algorithms such as the Elbow method.

(ii) Implement classification methods such as K-Nearest Neighbor (KNN) and Support Vector Machine (SVM) to improve the classification after performing the clustering algorithms.

(iii) K-Means clustering algorithm randomly select N cluster centroids. By setting the clusters centroid manually, in which the clusters centroid is defined in advance, a better performance might produce out of the K-Means clustering algorithm.

REFERENCES

Alarifi, A., Al-Salman, A., Alsaleh, M., Alnafessah, A., Al-Hadhrami, A., Al-Ammar, M., Al-Khalifa, H. (2016). Ultrawide-band Indoor Positioning Technologies: Analysis and Recent Advances. Sensors, vol. 16, no. 5, p. 707.

Alata, M., Molhim, M., Ramini, A. (2008). Optimizing of Fuzzy C-Means Clustering A-lgorithm Using GA. World Academy of Science, Engineering and Technology International Journal of Computer and Information Engineering, vol:2, no:3.

Al-Jazzar, S., Al-Nimrat, A., Muchkaev, A. (2011). Low complexity and high accuracy angle of arrival estimation using eigenvalue decomposition with extension to 2D AOA and power estimation. EURASIP Journal on Wireless Communications and Networking, https://doi.org/10.1186/1687-1499-2011-123.

Arsan, T. (2018, J). Improvement of Accuracy of Bluetooth based Indoor Positioning S-ystem by Using Big Bang – Big Crunch Optimization Method. Süleyman Demirel: University Journal of Natural and Applied Sciences, Volume 22, Special Issue, 367-374.

Arsan, T. (2018, D). Improvement of Indoor Positioning Accuracy of Ultra-wide Band S-ensors by using Big Bang - Big Crunch Optimization Method. Pamukkale University: Journal of Engineering Sciences, 24(5), 921-928, DOI: 10.5505/pajes.2018.59365.

Arsan, T. and Kepez, O. (2017). Early Steps in Automated Behavior Mapping via Indoor Sensors. Sensors, 17, 2925, doi: 10.3390/s17122925.

Bekkali, A., Matsumoto, M., Sanson, H., (2007). RFID Indoor Positioning Based on Pro-babilistic RFID Map and Kalman Filtering. White Plains, NY: Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pp. 21-21.

Biradar, S. and Hote, Y. (2016). Accelerated modified big bang big crunch optimization based on evolution of universe. Roorkee: 2016 11th International Conference on Industrial and Information Systems (ICIIS), pp. 698-703.

Brena, R., García-Vázquez, J., Galván-Tejada, C., Muñoz-Rodriguez, D., Vargas Rosales , C., Jr., J. (2017). Evolution of Indoor Positioning Technologies: A Survey. Journal of Sensors, Vol. 2017.

Cai, Y., Chen, Y., Fang, L., Guan, W., Wu, Y., Xie, C. (2017). Indoor High Precision Th-ree-Dimensional Positioning System Based on Visible Light Communication Using Particle Swarm Optimization. IEEE Photonics journal, Vol 9, Number 6.

Carreira-Perpiñán, M. (2015).A review of mean-shift algorithms for clustering. e-print a-rXiv:1503.00687. p. arXiv:1503.00687.

Deb, K . (1999). An introduction to genetic algorithms. Sadhana, vol. 24, parts 4&5, pp-93-315.

Decawave, MDEK1001 Kit User Manual Module Development & Evaluation Kit for the DWM1001, Version 1.2, 2017, https://www.decawave.com/wp-content/uploads/2019/01/MDEK1001_System_User_Manual-1.1.pdf (accessed on 15 July 2018).

EL- Sawy, A., Hussein, M., Zaki, E., Mousa, A. (2014). An Introduction to Genetic Algorithms: A survey A practical Issues. International Journal of Scientific & Engineering Research, vol 5, issue 1.

Erol, O. and Eksin, I. (2006). A new optimization method: big bang–big crunch. Advances in Engineering Software, 37(2):106–111.

Gezici, S., Giannakis, G., Kobayashi, H., Poor, H., Sahinoglu, Z., Tian, Z. (2005). Localization via ultra-wideband radios: a look at positioning aspects for future sensor networks. IEEE Signal Processing Magazine, vol. 22, no. 4, pp. 70-84.

Ghavami, M., Michael, L., Kohno, R. (2006). Ultra-wide band Signals and Systems in C-ommunication Engineering, 2nd Edition. NJ, USA: John Wiley & Sons, Ltd.

Gigl, T., Dizdarevic, V., Irahhauten, Z., Janssen, G., Witrisal, K. (2007). Analysis of a U-WB Indoor Positioning System Based on Received Signal Strength. Hannover: 4th Workshop on Positioning, Navigation and Communication, pp. 97-101.

Gu, Y., Lo, A., Niemegeers, I. (2009). A survey of indoor positioning systems for wireless personal networks. IEEE Communication Surveys, 11, 13–32.

Guo, Q., Chang, X., Chu, H., (2007). Clustering Analysis Based on the Mean Shift. Harb-in, China: In 2007 International Conference on Mechatronics and Automation, pp. 309–313.

H. Zhou and N. N. Van (2014). Indoor Fingerprint Localization Based on Fuzzy C-Means Clustering. Zhangjiajie: In the Sixth International Conference on Measuring Technology and Mechatronics Automation, pp. 337-340.

Huang, H. and Gartner, G. (2010). A Survey of Mobile Indoor Navigation Systems. Heidelberg, Germany: In Cartography in Central and Eastern Europe, Gartner, G., Ortag, F., Eds., Springer: pp. 305–319.

Jekabsons , G., Kairish, V., Zuravlyov, V. (2011). An Analysis of Wi-Fi Based Indoor P -ositioning Accuracy. Scientific Journal of Riga Technical University, 44, 131–137.

Jiang, L., Hoe, L., Loon, L. (2010). Integrated UWB and GPS location sensing system in hospital environment. Taichung :5th IEEE Conference on Industrial Electronics and Applications, pp. 286-289.doi: 10.1109/ICIEA.2010.5516828.

Kalman, R. (1960). A new approach to linear filtering and prediction problems. Journal of Fluids Engineering, vol. 82, no. 1, pp. 35 -45.

Koyuncu, H. and Yang, S. (2010). A Survey of Indoor Positioning and Object Locating Systems. IJCSNS International Journal of Computer Science and Network Security, vol.10 no.5.

Labbi, Y. and Attous, D. (2010). Big bang-big crunch optimization algorithm for economic dispatch with valve-point effect. Journal of Theoretical and Applied Information Technology, vol.16, no. 1.

Lee, B., Chungg, W., Lee, Y. (2008). 3D navigation real time RSSI-based indoor tracking application. Journal of Ubiquitous Convergence Technology, 2. 67-77.

Li, Q., Dai, W., Ji, K., Li, R., (2015). Kalman Filter and Its Application. Tianjin: 8th International Conference on Intelligent Networks and Intelligent Systems (ICINIS), pp. 74-77.

Liu, H., Banerjee, B., Darabi, H., Liu, J., (2007). Survey of Wireless Indoor Positioning Techniques and Systems. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews), vol. 37, no. 6.

Mahfouz, S., Farah, J., Honeine, P., Mourad-Chehade, F., Snoussi, H. (2014). Target tracking using machine learning and Kalman Filter in wireless sensor networks. IEEE Sensors Journal, Institute of Electrical and Electronics Engineers, 14 (10), pp.3715 - 3725.

Mautz, R. (2012). Indoor Positioning Technologies. Zürich, Switzerland: Ph.D. Thesis.

Michalewicz, Z. (1996). Genetic Algorithms + Data Structures = Evolution Pro-grams, 3rd Edition. Springer-Verlag.

Namratha, M. and Prajwala, T. (2012). A Comprehensive Overview of Clustering Algorit -hms in Pattern Recognition. IOSR Journal of Computer Engineering (IOSRJCE), ISSN: 2278-0661 volume 4, issue 6.

Ni, M., Lau, Y., Liu, Y., Patil, A. (2004). LANDMARC: Indoor Location Sensing Using Active RFID. Wireless Networks, Volume 10, Issue 6, pp 701–710.

Paivinen, N. and Gronfors, T. (2006). Finding the Optimal Number of Clusters from Artifi -cial Datasets. Budapest: 2006 IEEE International Conference on Computational Cybernetics, pp. 1-6.

Reddy, N. and Sujatha, B. (2011). TDOA Computation Using Multicarrier Modulation f -or Sensor Networks. International Journal of Computer Science and Communication Network, 85–90.

Syed Ahmed and Yonghong Zeng (2017). UWB Positioning Accuracy and Enhancement. Malaysia: In Proc. of the 2017 IEEE Region 10 Conference (TENCON), 5-8.

Salmond, D. (2001). Target tracking: introduction and Kalman tracking filters. Enschede, Netherlands: IEE Target Tracking: Algorithms and Applications (Ref. No. 2001/174), pp. 1/1-1/16 vol.2.

Sedlacek, P., Kovac, D., Slanina, M.(2016). An Overview of Indoor and Outdoor Position -ning Technologies with Focus on their Precision. Elektrorevue, vol.18, no.6.

Shedthi, B., Shetty, S., Siddappa, M. (2017). Implementation and comparison of K-Mean -s and fuzzy C-Means algorithms for agricultural data. Coimbatore :2017 International Conference on Inventive Communication and Computational Technologies (ICICCT), pp. 105-108.

Singh, K., Malik, D., Sharma, N. (2011). Evolving limitations in K-Means algorithm in data mining and their removal. IJCEM International Journal of Computational Engineering & Management, vol. 12, 2230-7893.

Sonagara, D. and Badheka, S. (2014). Comparison of Basic Clustering Algorithms. Intern -ational Journal of Computer Science and Mobile Computing, vol.3 issue.10, pg. 58-61.

Suganya, R. and Shanthi, R. (2012). Fuzzy C- Means Algorithm- A Review. International Journal of Scientific and Research Publications, volume 2, issue 11.

Sunantasaengtong, P. and Chivapreecha, S. (2014). Mixed K-Means and GA-based weig-hted distance fingerprint algorithm for indoor localization system. Bangkok: TENCON 2014 - 2014 IEEE Region 10 Conference, pp. 1-5.

Suroso, D., Cherntanomwong, P., Sooraksa, P., Takada, J. (2011). Location fingerprint te -chnique using Fuzzy C-Means clustering algorithm for indoor localization. Bali: TENCON 2011-2011 IEEE Region 10 Conference, pp. 88-92.

Tekinay, S., Chao, E., Richton, R. (1998). Performance benchmarking for wireless location systems. In IEEE Communications Magazine, vol. 36, no. 4, pp. 72-76.

Tuncer, T. (2017). Intelligent Centroid Localization Based on Fuzzy Logic and Genetic Algorithm. International Journal of Computational Intelligence Systems, vol. 10, no. 1, pp. 1056–1065.

Wang, S. (2010). System implementation study on RSSI based positioning in UWB networks. York: 7th International Symposium on Wireless Communication Systems, pp. 36-40.doi: 10.1109/ISWCS.2010.5624355.

Want R., Falcao V., Gibbons J., Hopper A. (1992). The active badge location system. ACM Trans. Inf. System, 10, 91–102.

Ward, A., Hopper, A., Jones, A. (1997). A new location technique for the active office. IEEE Pers. Communication, 4, 42–47.

Woods, J. and Radewan, C. (1977). Kalman Filtering in two dimensions. Information Theory, IEEE Transactions, vol. 23, no. 4, pp. 473 – 482.

Yesilbudak, M. (2016). Clustering analysis of multidimensional wind speed data using k-Means approach. Birmingham: 2016 IEEE International Conference on Renewable Energy Research and Applications (ICRERA), pp. 961-965.

Zetik, R., Shen, G., Thomä R. (2010). Evaluation of Requirements for UWB Localization Systems in Home-Entertainment Applications. ETH Zurich, Switzerland: Proceedings of the 2010 International Conference on Indoor Positioning and Indoor Navigation (IPIN), pp. 1-8.

Zhong, Y., Dong, B., Wu, F., Zhang, J. (2016). Wi-Fi indoor localization based on K- Means. Shanghai: International Conference on Audio, Language and Image Processing (ICALIP), pp. 663-667.

Zuo, Z., Fang, Y., Liu, L., Zhang, L. (2018). Indoor Positioning Based on Bluetooth Low-Energy Beacons Adopting Graph Optimization. Sensors, V. 18(11): 3736, doi: 10.3390/s18113736.

CURRICULUM VITAE

Personal Information

Name Surname	: Mohammed Muwafaq Noori Hameez
Place and Date of Birth	: Iraq, Baghdad. 23/10/1990

Education

Undergraduate Education	: B.Sc. in Information and Communication Engineering
Graduate Education	: M.Sc. in Computer Engineering
Foreign Language Skills	: Arabic, English.

Work Experience

Name of Employer and Dates of Employment: Zain Telecom, 2016.

Contact:

Telephone	: 0 (536) 43 96 382
E-mail Address	: Mhd.hameez@gmail.com